

**Requirements
for an Electronic Calendar Tool**

**Version 0.8
17 November 2000**

Contents

1. Introduction	1
1.1. Problem Statement	1
1.2. System Personnel	1
1.3. Operational Setting	2
1.4. Impacts	2
1.5. Related Systems	2
1.6. Document Conventions	3
2. Functional Requirements	4
2.1. User Interface Overview	4
2.1.1. Regular User Interface	4
2.1.2. Administrative User Interface	8
2.1.3. General Features of the User Interface	9
2.1.4. Screen Map	10
2.2. Basic Appointment Scheduling	10
2.3. Viewing Calendars	17
2.3.1. Viewing the Calendar at Different Levels	17
2.3.1.1. Day View	17
2.3.1.2. Week Views	22
2.3.1.3. Month View	27
2.3.1.4. Year View	28
2.3.1.5. Scheduled Item View	29
2.3.2. Viewing Previous, Next, and Specific Calendar Dates	30
2.3.2.1. Next and Previous	31
2.3.2.2. Today	31
2.3.2.3. Goto Date	32
2.3.2.4. Upper and Lower Date Bounds	34
2.3.3. Viewing Lists of Scheduled Items	34
2.3.3.1. Appointment Lists	35
2.3.3.2. Meeting Lists	36
2.3.3.3. Task Lists	37
2.3.3.4. Event Lists	39
2.3.3.5. All Items Lists	40
2.3.3.6. Custom Lists	41
2.3.4. Filtered Viewing	47
2.3.4.1. Custom Filter Definition	50
2.3.4.1.1. Title	50
2.3.4.1.2. Earliest and Latest Date	54
2.3.4.1.3. Earliest and Latest Time	54
2.3.4.1.4. Duration	54
2.3.4.1.5. Recurring?	55
2.3.4.1.6. Category	55
2.3.4.1.7. Location	55
2.3.4.1.8. Security	55
2.3.4.1.9. Priority	55
2.3.4.1.10. Task Priority	55
2.3.4.1.11. Completed? and Completion Dates	56
2.3.4.1.12. Remind?	56
2.3.4.1.13. Details	56
2.3.4.1.14. Attendees	56

2.3.4.2. Using Custom Filters	56
2.3.4.3. Details of Filter Application	62
2.3.5. Viewing Other Users and Group Calendars	63
2.3.5.1. Viewing Other User's Calendars	63
2.3.5.2. Viewing Group Calendars	67
2.3.6. Viewing in Different Windows	68
2.3.6.1. Closing Windows	70
2.3.6.2. Windowing Mode	70
2.3.6.2.1. Per-Level Windowing Mode	71
2.3.6.2.2. Two-Window Mode	73
2.3.6.2.3. Multi-Window Mode	73
2.3.6.2.4. Windowing Mode Calendar Specificity	74
2.3.6.3. Magnetizing Windows	74
2.3.6.4. The Calendars List	79
2.3.6.5. Window Banner Syntax	81
2.4. More Scheduling	82
2.4.1. Scheduling Meetings	82
2.4.1.1. A Group Leader Scheduling Two Straightforward Meetings	83
2.4.1.2. A Group Leader Encountering Difficulties While Scheduling	93
2.4.1.3. A Super-Group Leader Scheduling a Meeting	99
2.4.1.4. A Non-Leader Scheduling a Meeting for Selected Individuals	102
2.4.1.5. Receiving and Accepting Meeting Notifications	106
2.4.1.6. Further Operational Details of Meeting Scheduling	112
2.4.1.6.1. The Contents of Group Calendars	112
2.4.1.6.2. Scheduling Over Private Items	113
2.4.1.6.3. The Status of Not-Yet-Accepted Meetings	113
2.4.1.6.4. Creation of New Categories	113
2.4.1.6.5. The Role of the Location Database	113
2.4.1.6.6. Non-Unique Scheduled Meetings	114
2.4.1.6.7. Notifications for Multiple Calendars from Multiple Central Hosts	115
2.4.1.6.8. Pending Notifications	115
2.4.1.6.9. Concurrent Scheduling	115
2.4.2. Scheduling Tasks	118
2.4.3. Scheduling Events	119
2.5. Finer Points of Viewing and Scheduling	123
2.5.1. Item-Level Viewing	123
2.5.1.1. Viewing Appointment Items	123
2.5.1.2. Viewing Meeting Items	128
2.5.1.3. Viewing Task Items	132
2.5.1.4. Viewing Event Items	133
2.5.1.5. Item-Level Viewing in Other Users' Calendars	133
2.5.2. Changing and Deleting Scheduled Items	133
2.5.2.1. Changing and Deleting Appointments	135
2.5.2.2. Scheduler Changing and Deleting Meetings	141
2.5.2.3. Meeting Change and Delete Notifications	151
2.5.2.4. Group Calendar Changes	158
2.5.2.5. Non-Scheduler Changing and Deleting Meetings	158
2.5.2.6. Changing and Deleting Tasks	161
2.5.2.7. Changing and Deleting Events	161
2.5.3. Details of Scheduling Recurring Items	164
2.5.3.1. Weekly/Biweekly Intervals	165
2.5.3.2. Monthly Intervals	165
2.5.3.3. Yearly Intervals	168
2.5.3.4. Recurring Meeting Requests	168

2.5.3.5. Recurring Items and Instances	170
2.5.4. Receiving Reminders	171
2.5.5. Editing Categories	172
2.5.6. Dynamic View Updating	176
2.6. Administrative Functions	177
2.6.1. Gaining Administrative Access	177
2.6.2. User Database	178
2.6.3. Group Database	185
2.6.4. Location Database	191
2.6.5. Other Admin Commands	197
2.6.5.1. Administrative Control of the Central Host	197
2.6.5.2. Listing Administrators	203
2.6.5.3. Changing an Administrator Passwords	204
2.6.5.4. Notifying Users	204
2.6.5.5. Purging Databases	204
2.6.5.6. Creating Calendar Tool Distributions	204
2.6.6. Regular User Access to Admin Commands	205
2.6.6.1. Database Access	205
2.6.6.2. Central Host Connections	205
2.6.6.3. Local Directory	206
2.6.7. Global Options	206
2.6.7.1. Administrative Options	206
2.6.7.1.1. Root System Administrator	206
2.6.7.1.2. Group Leader Privileges	206
2.6.7.1.3. Capping Calendar Sizes	207
2.6.7.1.4. Platform-Dependent Options	207
2.6.7.2. Summary of Super User Privileges	207
2.6.8. Establishing Administrative Privileges	207
2.6.9. Administrative Files	207
2.6.10. Administrative Notifications	207
2.6.11. User Communication with Administrators	208
2.7. Options	208
2.7.1. Times and Dates	213
2.7.2. Fonts	214
2.7.3. Scheduling Options	218
2.7.3.1. Scheduling Defaults Options	218
2.7.3.2. Scheduling Overlaps Options	224
2.7.3.3. Scheduling Meetings Options	224
2.7.3.4. Scheduling Reminders Options	226
2.7.4. Viewing Options	227
2.7.4.1. Viewing Lists Options	227
2.7.4.2. Viewing Windows Options	237
2.7.4.3. Viewing Miscellaneous Options	237
2.7.5. Administrative Options	237
2.7.6. Options in the Calendar Tool Administration Program	240
2.7.7. Operating Environment Conventions	241
2.8. Details of File Commands	241
2.8.1. New, Open, Close, and Close All	241
2.8.2. Save, Save As, Save All	242
2.8.3. Save Config	242
2.8.4. Print	243
2.8.5. Exit	243
2.8.6. Save As and Load in Editing Dialogs	244
2.8.7. Standard Files	248

2.8.8. File Types	248
2.8.9. Calendar File Contents	249
2.9. Details of Edit Commands	249
2.9.1. Undo, Redo, Repeat	249
2.9.2. Cut, Copy, Paste	249
2.9.3. Find	249
2.9.4. Text Command Interface	249
2.10. Help	249
2.10.1. About	250
2.10.2. Quick Help	250
2.10.3. Detailed Help	252
2.10.3.1. Detailed Help Contents	253
2.10.3.2. Detailed Help Index	254
2.10.3.3. Detailed Help Search	255
2.10.3.4. Help Presentation Details	257
2.11. Data Entry Details	257
2.11.1. Command Enabling and Disabling	257
2.11.2. Pressing OK (4sep02)	257
2.11.3. Comma-Separated Lists	257
2.11.4. Typed Input	257
2.11.5. Date Formats	258
2.11.6. Item Uniqueness Requirements	259
2.11.7. Scheduling Dialogs and Item-Level Displays	259
2.11.7.1. Meeting Scheduling Dialogs	259
2.11.7.2. Defaults	260
2.11.8. Required Fields	260
2.11.9. Selecting Viewing Targets	261
2.11.10. Deletion of Scheduled Items	262
2.11.11. Overlapping Items	262
2.11.12. Category Editing Details	262
2.11.13. String and Pattern Matching Rules	263
2.11.14. Regular Expression Filtering Patterns	263
2.11.15. Boolean Expression Filtering Patterns	263
2.11.16. System-Defined Events	263
2.11.17. Dialog State Details	263
2.11.17.1. Custom List Dialog	263
2.11.18. Combo-box menu text truncation and other behaviours	264
2.11.19. Default Names	264
2.11.20. Name Completion	264
2.11.21. Meeting Minutes	264
2.11.22. Merging Scheduler Changes	264
2.11.23. Picky Details about Change/Delete Button Enabling	264
2.11.24. Precise Details of Clear and Cancel	265
2.11.25. Add User and User Record Dialogs	265
2.11.26. Non-Modal, Calendar-Specific Dialogs	266
2.11.27. Units of Measure	269
2.11.28. Options Data	269
2.12. Error Conditions	269
2.12.1. Commands with No Effect	269
2.12.2. Searching for Unknown User	270
2.12.3. Scheduling Errors	270
2.12.3.1. Unknown Meeting Attendees	270
2.12.3.2. Meeting Scheduler Validation	270
2.12.3.3. Non-Unique Items	271

2.12.3.4. Changing a Meeting to an Unavailable Location	271
2.12.4. Errors from an External WWW Browser	271
2.12.5. Administrative Errors	271
2.12.5.1. Non-Admin Attempting to Login as Admin	271
2.12.6. Failure of Central Host Computer	271
2.12.6.1. Loss or Corruption of Central Host Data	272
2.12.7. Failure of Local User Computer or Invalidation of User Information	272
2.12.8. Missing or Corrupted Options File	272
2.12.9. File Errors	272
2.12.10. Zero-Instance Recurring Items	272
2.12.11. Exceeding Item Limits	272
2.13. Graphical User Interface Details	273
2.13.1. Screen Coordinates	273
2.13.2. Enabled and Disabled GUI States	273
2.13.3. Remembering Previously-Entered Values	273
2.13.4. 28jun01 Note	273
2.13.5. View Command Display Details	273
2.13.5.1. Day View Scrolling	273
2.13.5.2. Window Sizes	274
2.13.5.3. Details of Horizontal Overlap Display	275
2.13.6. Window Positioning for Next/Previous Commands	275
2.13.7. Apple Macintosh Menu Conventions	275
2.14. Multi-User Operating Environment	275
2.14.1. Invoking the Calendar Tool from the Operating Environment	276
2.14.2. The Central Calendar Repository	277
2.14.3. Messaging	277
2.14.4. Coordination	277
2.14.5. Implementation Considerations	278
2.14.6. Clock Synchronization Issues	278
2.14.7. Files	278
2.14.8. Administrative Purging of Excess Calendar Data	279
2.14.9. Security Considerations	279
2.14.10. Local/Remote Calendar Synchronization	279
2.15. System Installation and Operating Environment Interface	279
2.15.1. Regular User Installation	279
2.15.2. Implementor or Administrator Configuration of the Calendar Tool	279
2.15.3. User Installation of the Application	280
2.15.4. Administrator Installation of the Server	280
2.15.5. Location of Calendar Tool Files	280
2.15.6. Invoking the Calendar Once Installed	280
2.16. Future Enhancements	280
2.17. Work in Progress	280
2.17.1. Bug	280
2.17.2. The Term “Client”	280
2.17.3. Window “Banner” versus “Title”	281
2.17.4. Filtering Todo Items	281
2.17.5. Options Section Wording Fodder that Will Probably Be Nuked	281
2.17.6. Possible Fix Needed	281
2.17.7. Fix Links in Admin Section	281
2.17.8. Filtering Update Needed	281
2.17.9. Tree Viewer for Groups and Other Admin-Related Discussion	281
2.17.10. External Operating Environment Issues	281
2.17.11. Custom Filter Menu Options	282

2.17.12. Misc	282
2.17.13. Make Sure Button and Menu Item States are Accurate	282
2.17.14. Quick Note	282
2.17.15. Hmm, Centralized Server Architecture Rears Its Ugly Head	282
2.17.16. Nuked These Things and Think They Can Stay Nuked	283
2.17.17. Scripting Language Derived from UI Command	283
2.17.18. A Couple More Todo Items	283
2.17.19. Prev/Next Arrow Issue	284
2.17.20. Make Sure that Disabled (Greyed) State of All Buttons is Correct	284
2.17.21. Consider Where To Put Dialog State Details	284
2.17.22. Maybe for custom list section	285
2.17.23. Consider adding Emacs to list of related systems in section 1.5	285
2.17.24. Complete Clear Statement for Item Uniqueness Criteria	285
2.17.25. Check Consistency between Task Scheduling and Task List Sections	285
2.17.26. Default Task Priority	285
2.17.27. Color	285
2.17.28. Beware (potentially) of automatic rescheduling	285
2.17.29. General fixes and updates to do	286
2.17.30. General Constraints	286
2.17.31. Commands with No Effect	287
2.17.32. Trimming	287
2.17.33. Uniform Effect of the 'Cancel' Command	287
2.17.34. Explanatory Error and Confirmation Dialogs	287
2.17.35. Help	287
2.17.35.1. Possible Options	287
2.17.35.2. Paragraph (from Rolodex) about Quick Help Activation Area	287
3. Non-Functional Requirements	289
3.1. System	289
3.1.1. Performance	289
3.1.2. Operational Environment	289
3.1.3. General Characteristics	289
3.2. Process	289
3.3. Personnel	290
3.4. System Performance Constraints	291
3.5. Qualitative system characteristics	291
3.6. Development Goals and Constraints	291
4. Developer Overview	292
5. Formal Specification	293
5.1. Scheduling (schedule.rsl)	293
5.2. Calendar Database (caldb.rsl)	313
5.3. Viewing (view.rsl)	315
5.4. Administration (admin.rsl)	323
5.5. Options (options.rsl)	332
5.6. File Handling (file.rsl)	333
5.7. Data Editing (edit.rsl)	337
6. Rationale	341

6.1. Nuked from Admin; Presumably to Go In Some Form in Installation	341
6.2. Fodder from Winow Viewing Section, Ca. Aug 02	341
6.3. Meeting Notifi cations and Calendar-Host Association	341
6.4. Nuked from Options, Related to Overly-Complicated Home Dir Specs	342
6.5. Decision about User Control Over Notifi cations	342
6.6. Some Uncommonly To-The-Point Fodder from the Options Section	342
6.7. Nuked from Options Section	343
6.8. Hopefully Final Nukes for Options	343
6.9. Nice Simplifi cation to User Records	343
6.10. Overly Complicated Font Options Removed	344
6.11. Some Serious Rationale for Options	345
6.12. More Potential Rationale Fodder Cleansed from Rapidly Maturing Options Setion	346
6.13. Here’s an Important Bit that Finally Got Handled	346
6.14. The Real Deal for Options	346
6.15. Thinking about Options and Defaults	348
6.16. ScheduledItem Inheritance Structure	348
6.17. A Feature Removal (!)	349
6.18. More on Admin Decisions	349
6.19. Fodder Nuked from Admin Section	349
6.20. Detailed Explanation of Recurring Date Change Restrictions	350
6.21. Changing and Deleting Meetings	350
6.22. From the end of the change and delete section	350
6.23. From the task scheduling section	351
6.24. Possible fodder for schedule meeting	351
6.25. Possible fodder for meeting change/delete section rationale	351
6.26. Possible fodder from appt changing section	353
6.27. (Weakly) Possible fodder from meeting item viewing	353
6.28. Possible fodder from task item viewing	354
6.29. Possible fodder in the area of external fi le viewing	354
6.30. 2aug01 -- Nixing the notifi caiton enabling stuff (then not)	354
6.31. 30jul01	355
6.32. Axed from Meeting Scheduling	355
6.33. A Bit on Task Scheduling	356
6.34. Documentation Section Ordering	356
6.35. Relative Importance of Actual Tool versus Pedagogical Exmample Goals	356
6.36. Filter Dialog Layout	356
6.37. Refi nement Example	357
6.38. Alternatives for Multi-Window Mode Behavior	357
6.39. Precise Behavior of Next and Previous at the Item Level	357
6.40. Misc Ideas from Relatively Early On	358
6.41. Start/End versus Start/Duration, Revisited	358
6.42. Start/End versus Start/Duration (Older Ideas)	359

6.43. Big Issue about Factoring Options	359
6.44. Old Remarks from the List Viewing Section	359
6.45. Filtering Issues	360
6.46. Back and Forth with the Monthly Recurring Functionality	360
6.47. Maximum Date Range	360
6.48. Lists	361
6.49. Canonical Modeling Form	361
Appendix A. Users Manual	362
Appendix B. Command Language	363
Appendix C. Help Content	365
C.1. Quick Help Messages	365
C.2. Detailed Help Content	365
Appendix D. Feature Comparison Matrix	366

1. Introduction

This document defines requirements for an electronic Calendar Tool. The tool provides functions to manage calendars for individual users as well as for groups of users. Individuals use the Calendar Tool to schedule appointments and to view their calendar information in a variety of useful ways. Administrative users can schedule meetings for groups of users and perform other system administration functions.

In addition to providing useful functionality, the Calendar Tool serves as a pedagogical example of software engineering concepts and principles. The software development artifacts for the Calendar tool are considered part of the overall Calendar Tool package.

1.1. Problem Statement

The general problems to be solved by the Calendar Tool are the following:

- a. to provide a reliable and easy to use tool that manages an individual's calendar, in a manner comparable to non-electronic media such as day planners and date books
- b. to provide a secure group environment that manages a collection of individual calendars for the purposes of group scheduling and information sharing

While these problems have been solved by a number of commercially available and public domain products, the Calendar Tool provides a solution with certain functional improvements over existing tools. In addition, the development of the Calendar Tool focuses on two other specific problems not addressed in commercial tools:

- a. to provide a pedagogical example for use in the instruction of software engineering, including an example of the practical application of formal methods
- b. to provide fully documented public domain source code for use by others

1.2. System Personnel

The personnel involved in the Calendar Tool project are organized into the following groups and subgroups:

- a. end users
 - i. registered users
 - ii. user group leaders
 - iii. system administrators
 - iv. unregistered individual users
- b. system developers
- c. software engineering students
- d. outside parties

End users are those who use the Calendar Tool for its intended purpose. Registered end users have calendars that are stored in a Calendar Tool multi-user database, which is used for group scheduling purposes. Group leaders are designated registered users who may perform group scheduling operations which affect other users' calendars. Master system administrators perform overall system administration functions, including user and group leader registration. Finally, anyone can use the Calendar Tool as an unregistered user for the management of a strictly individual calendar.

The primary system developer is Gene Fisher. His efforts are based on the work of a number of software engineering students who have used the Calendar Tool as a class project in software engineering courses. Students whose work has been particularly helpful are Larry Bolef, Brent Smolinski, Rick Myers, Brandon Wallace, and Oliver Wallace.

As noted above, the Calendar Tool is intended to serve as an example for use in software engineering courses. The students who use the Calendar Tool for this purpose focus on the artifacts of its development rather than its use as a functioning tool.

The Calendar Tool is available as public domain software for use by outside parties. The project directory is located at

<http://www.csc.calpoly.edu/~gfisher/projects/calendar>

1.3. Operational Setting

There are two operational settings for the Calendar Tool: (1) normal use as a functioning software system; (2) use as an example in software engineering courses.

Since the Calendar Tool is designed as general public domain software, there is no specific operational setting in which it is intended to be installed. The setting for which it is appropriately suited is an office or organizational environment of some form, where users maintain their own calendars and need to schedule meetings with other users on a regular basis.

For use as a pedagogical example, the Calendar Tool has been designed to fit the curriculum used by Gene Fisher in undergraduate and graduate software engineering courses at Cal Poly University. These courses are two-quarter sequences that cover standard aspects of software engineering, with emphasis on the practical application of formal methods. For the most part, the concepts covered in these courses, and hence the concepts embodied in the Calendar Tool, are mainstream software engineering. Other instructors may therefore find the Calendar Tool and its development artifacts useful as course examples.

1.4. Impacts

The positive potential impacts of the Calendar Tool as a functioning system are increased convenience and efficiency in managing individual and group calendars. The positive impacts of the Calendar Tool as a course example are

- a. the presentation of a non-trivial software system that students can use as a guide for their own software development work;
- b. an illustration of how formal methods of can be put to practical use.

Potential negative impacts include those common to any user-oriented software system. Viz., if the system is poorly designed and implemented, it can be inconvenient to use and decrease rather than increase user productivity. More specific negative impacts relate to issues of privacy, reliability, and data integrity. If the system implementation is flawed, it may reveal private information that users do not wish to be revealed. If an organization of users relies on a flawed implementation of the system to schedule critical events, and such events are improperly or incompletely scheduled, the organization may suffer significant disruption of its operations. Significant disruption may also occur if the system loses or corrupts important calendar data.

As a course example, the use of the Calendar Tool has no significant negative impacts, unless its development methodology is considered weak or unrelated to the concepts being taught in a particular course. Such negative impacts can be easily avoided if instructors carefully examine the example before using it in a particular software engineering curriculum.

1.5. Related Systems

Widely used commercial systems that provide functionality comparable to the Calendar Tool include

- Microsoft Outlook,
- Netscape Calendar
- Qualcomm Now-Up-to-Date,
- FileMaker (formerly Claris) Organizer, and
- On Technology Meeting Maker.

Each of these systems provides a wide variety of calendar and scheduling functionality. In terms of such functionality, the goal for the Calendar Tool is to provide the important core features found in these tools, not the full range of features found in all of them. The Calendar Tool does provide modest functional improvements in three areas, when compared to most commercially available tools. These improvements are:

- a. more flexibility in viewing and selecting from lists of possible meeting times that are generated by the automatic scheduler;

- b. more flexibility in specifying recurring events;
- c. more extensive forms of view filtering.

The goals for the Calendar Tool relating to public domain source code and sound pedagogical development are not met by any of the above commercial systems.

1.6. Document Conventions

2. Functional Requirements

The Calendar Tool provides functionality to schedule appointments, meetings, events, and tasks. Calendars can be viewed in a variety of display formats, from single day to a full year. Calendar operations can be performed by individual users on their personal calendars, as well as by administrative users on a system-wide calendar. Administrative users can schedule meetings for user groups and perform general administrative operations for managing databases and setting global defaults.

Following an overview of the Calendar user interface, details of system use are presented in the following scenarios:

- basic appointment scheduling
- viewing calendars in a variety of ways
- scheduling the other types of calendar item
- finer points of viewing and scheduling
- performing administrative functions
- using options commands
- details of file-related commands
- details of edit commands
- help commands
- data entry details
- error conditions

Following the usage scenarios are three sections that summarize functional details in the areas of the multi-user operating environment, graphical user interface, and system installation. The functional requirements conclude with a discussion of possible future enhancements to the Calendar Tool. These are enhancements that may be incorporated into some future version of the system, beyond its initial release.

In preliminary versions of the requirements document there is section on work in progress, containing ideas that are not yet incorporated into an appropriate section of the functional requirements.

2.1. User Interface Overview

There are two separate user interfaces for the Calendar Tool, provided by two separate application programs. The primary interface is for regular users to browse and edit their calendars. A "regular user" is defined as a registered user without administrative privileges, or an administrator using the Calendar Tool for non-administrative purposes. The regular-user interface is provided by the program referred to as the "Calendar Tool".

A separate interface is provided for administrative users to maintain Calendar Tool databases and set default user options. This interface is provided by the program referred to as "Calendar Tool Administration".

2.1.1. Regular User Interface

When a registered user initially invokes the Calendar Tool, the screen appears as shown in Figure 1. The initial display consists of two windows: a command menubar and a monthly calendar view. The name of the current calendar is displayed in the banner of the menubar, the initial default being a new unnamed calendar. In the monthly display, the entry for the current date is highlighted with a widened border. The monthly display is the standard default view. The user can change this and other default settings using the option commands.

Figure 2 shows an expansion of the command menus for the regular user interface. The File menu contains typical commands for manipulating data files and performing other system-level functions. 'File New' creates a new empty calendar. 'File Open' opens an existing calendar from a previously saved file. 'File Close' closes the currently active calendar, offering to save if it has been modified since opening. 'File Close All' closes all open calendars, offering to save any that have been modified.

'File Save' saves the currently active calendar on the file from which it was opened, or on a new file if it was created from a new display. 'File Save As' allows the current calendar to be saved on a different file from which it was opened or most recently saved upon. 'File Save All' saves all currently open calendars.

Calendar: unnamed <input type="checkbox"/> <input type="checkbox"/>						
File Edit Schedule View Admin Options Help						
Monthly Agenda <input type="checkbox"/> <input type="checkbox"/>						
<input type="button" value="◀ Today ▶"/> September 1998						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
28	27	29	30			

Figure 1: Initial screen for a registered user.

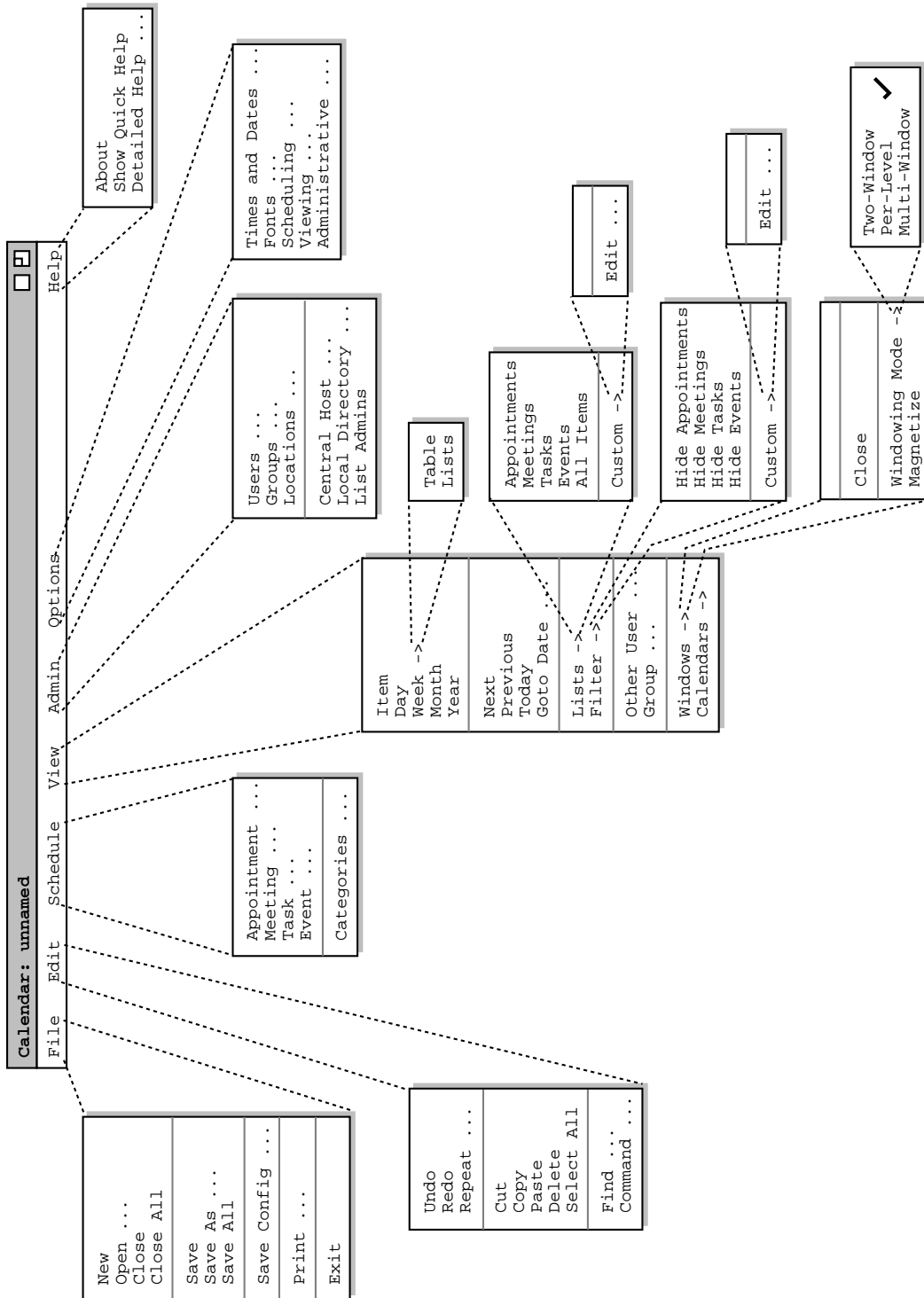


Figure 2: Expanded command menus for the regular user interface.

'File Save Config' saves the current screen configuration of all Calendar Tool windows. The saved configuration file can be used for a subsequent invocation that starts the Calendar tool in exactly the same configuration as was saved.

'File Print' prints the contents of the currently active display window in a form suitable for output to a printing device capable of displaying text and graphics. 'File Exit' exits the Calendar Tool, offering to save any

modified calendars if necessary.

The `Edit` menu contains commands for manipulating calendar data during editing. `Edit Undo` undoes the most recently completed editing commands. Repeated successive invocations of `Undo` undo successively earlier commands. `Edit Redo` redoes the most recently undone command. Repeated successive invocations of `Redo` redo the series of undone commands in reverse order. `Edit Repeat` displays a dialog that allows the most recently invoked command to be repeated one or more times.

`Edit Cut` removes and copies the currently selected datum in the current display. `Edit Copy` copies the currently selected datum without removing it. `Edit Paste` inserts the most recently cut or copied datum at the currently selected edit point in the current display. `Edit Delete` removes the currently selected datum without copying it. `Edit Select All` selects all editable data in the current display.

`Edit Find` performs a search for a text string. The search is performed in all open calendars. Successive invocations of search with an unchanged search string search for further occurrences of the string until all occurrences are found. `Edit Command` allows the user to enter any Calendar tool command in textual form, including sequences of commands to form scripts.

The `Schedule` menu contains commands to schedule four types of item on a calendar. An *Appointment* is an item scheduled for an individual user, containing a title, time, date, and other information. A *Meeting* is similar to an appointment, but may be scheduled for multiple users of the Calendar Tool. A *Task* is a form of "to do" item, with information similar to an appointment plus other task-related information. An *Event* is the simplest form of scheduled item, containing a subset of the information in an appointment. Events are useful for scheduling simple, one-time items.

`Schedule Categories` allows the user to view and edit the categories of scheduled items. A category is a descriptive, color-coded label used to help organize calendar items.

The `View` menu allows the user to browse through a calendar in a variety of ways. `View Item` displays the scheduling details for a selected scheduled item. `View Day` displays details of the currently selected calendar day. `View Week` displays the seven-day week in which the currently selected day appears, with less detail than the daily display. Weeks can be displayed in tabular or list format. `View Month` displays a large-grain view of the current month. `View Year` displays a very large-grain view of the current year. Each of these view commands can display its results in a separate window. With this feature, the user can create side-by-side displays to view any number of consecutive days, weeks, months, or years.

`View Next` and `View Previous` move forward and backward in the current display, based on the display granularity (i.e., item, day, week, month, or year). `View Goto Date` displays a dialog for choosing a specific date to become the current date in the active display.

`View Lists` displays lists of currently scheduled items of the four schedule types -- *Appointments*, *Meetings*, *Tasks*, and *Events*, or of all item types together. Custom lists can also be created, showing selected items only. `View Filter` allows the user to hide selected information from calendar displays. Any or all of the four types of scheduled items can be selectively hidden. The user can also construct custom viewing filters, based on a wide range of filtering criteria. Custom filters apply to both calendar and list views uniformly.

`View Other User` allows the current user to view the public portions of other users' or groups' calendars. Each individual user controls the visibility of scheduled items that other users may see.

`View Windows` allows the user to select for viewing from all active display windows. The `Close` command in the `View Windows` menu closes the current display window by removing it from the screen. `View Calendars` lists the names of all currently open calendars.

The `Windowing Mode ->` submenu provides three settings for the Calendar Tool windowing mode. In `Two-Window` mode, all views are shown in just two windows, one for calendars and the other for lists. In the default `Per-Level` mode, each level of the calendar, from individual item through full year, is shown in a separate level-specific window; each type of list is also shown in a separate window. In `Multi-Window` mode, the result of each and every view command is shown in a separate window, even for commands at the same level or the same type of list. Multi-window mode allows the user to create side-by-side displays of adjacent calendar periods.

The View Windows Magnetize command controls the "magnetic" property of display windows. When a window is magnetized, it "sticks to" an adjacent window when dragged next to it.

The Admin menu contains commands to access the three system databases -- users, groups, and locations. The *User* database contains a record for each officially registered user of the Calendar Tool. The *Group* database defines user groups, primarily for meeting scheduling purposes. The *Location* database defines rooms and other venues in which meetings can be held, including information about the facilities available in the locations. Regular users only have read access to the databases, that is they cannot add, change, or delete any database information.

The 'Admin Central Host' command displays information about the computer that is used to coordinate Calendar Tool registered users. 'List Admins ...' shows the Calendar Tool IDs of all system administrators. The 'Login' command allows a user with administrative privileges to launch the administrative interface from the regular user interface.

The 'Options menu has commands for setting various system options. The options are organized in five major areas: 'Times and Dates', 'Fonts', 'Scheduling', 'Viewing', and 'Administrative'.

The Help menu contains commands for viewing documentation of the Calendar Tool. 'Help About' displays a brief description of the tool, including information about the tool developers and how they can be contacted. 'Show Quick Help' activates brief help messages that appear when the user moves the mouse over various areas of the display screen. 'Detailed Help' displays an online version of the complete Calendar Tool users manual.

The preceding interface is that presented to registered Calendar Tool users. A user is registered by an administrator when a record is added to the Calendar Tool user database, as described in Section 2.6.2. The interface for unregistered users is a proper subset of the registered-user interface. Specifically, the following features are absent in the unregistered user interface:

- the 'Schedule Meeting' command
- the 'View Other User' and 'View Group' commands
- the 'Admin' command menu
- the 'Options Administrative' command

Description of these commands is absent in the help content.

2.1.2. Administrative User Interface

When an administrative user initially invokes the Calendar Tool Administration program and enters a valid administrator password, the system displays the interface shown in Figure 3. The initial display consists of a command menu

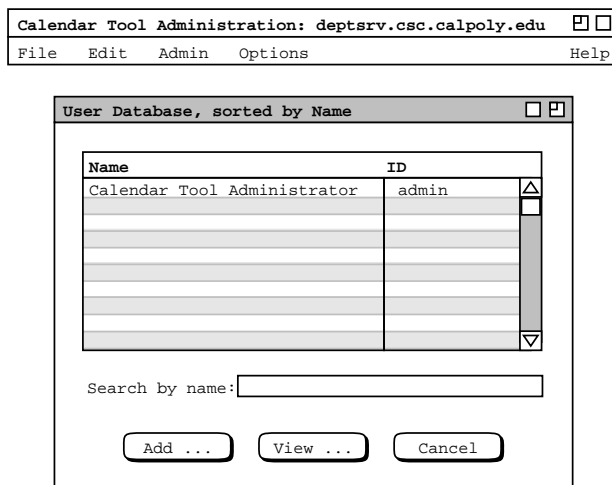


Figure 3: Initial screen for system administrator.

and a window for the Calendar Tool user database. The user database has one pre-defined entry for a system administrator. Other users are added with the administrative commands.

Figure 4 shows an expansion of the command menus for administrative access to the Calendar Tool databases. The 'File' menu has a reduced set of commands, since databases and system-wide options are stored in fixed files. The 'File Save' command saves all database changes performed by the administrative user. 'File Save Copy' saves a backup copy of the databases in a selected file directory. 'File Print' prints the contents of one or more databases in a form suitable for output to a printing device capable of displaying text and graphics. 'File Exit' exits Calendar Tool Administration, offering to save if necessary.

The 'Edit Menu' in the administrative interface is a proper subset that in the regular user interface. The edit commands that are present have the same functionality as in the regular interface.

The 'Admin' menu provides access to the three databases. Users with administrative privileges can add, change, and delete database information. The 'Central Host' command allows the administrator to view and edit information about the Calendar Tool central host computer. 'List Admins ...' shows the Calendar Tool IDs of all system administrators. The 'Password' command allows the administrative user to set her own administrative access password. 'Notify Users ...' provides the means to send a message to some or all Calendar Tool users. The 'Distribution' command is used to create pre-configured distribution copies of the Calendar Tool for downloading by users.

The 'Options' menu has the same commands as for the regular user. When an administrator sets option values, the values serve as the defaults for all regular users.

The 'Help' functionality is same as for regular users. Help content is modified accordingly to describe the commands that are particular to administrative users.

2.1.3. General Features of the User Interface

Overall, the Calendar Tool user interface has the following general features:

- It is presented in a form independent of any particular computer operating system or platform.
- The layout style of the interface is generally in separate independent windows, rather than one or a few windows with tiled or stacked information.
- Unless explicitly stated to the contrary, all interface dialogs are "non-modal", meaning that dialogs do *not* disable other components of the overall interface. When a non-modal dialog is running, the user may leave that dialog, perform any other interface commands, and then return to the dialog at some future point.

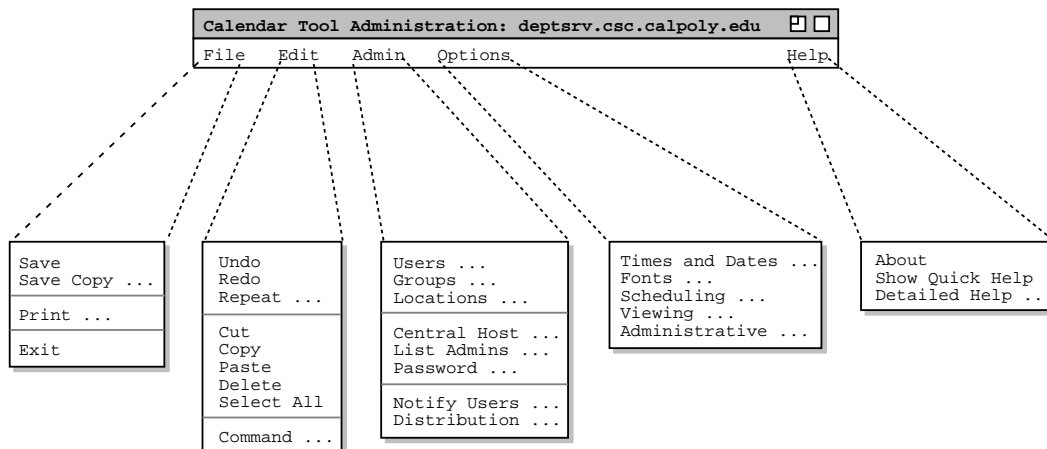


Figure 4: Expanded command menus for the administrative interface.

- d. The interface is presented in a plain and simple style, that focuses on functionality without extraneous aesthetic detail.

Further details of the interface presentation conventions are covered in the GUI conventions document used in the undergraduate and graduate software engineering courses taught by Gene Fisher at Cal Poly university.

2.1.4. Screen Map

Figure 5 shows a hierarchical map of the user interface screens for regular users. Figure 6 shows a screen map for administrative users.

2.2. Basic Appointment Scheduling

This introductory scenario shows an individual user scheduling two simple appointments. Appointment scheduling is one of the most commonly performed operations with the Calendar Tool. In this scenario, the user is assumed to have just been registered in the Calendar Tool. Hence the user has a brand new calendar with no entries and the standard default settings for all options. Details of user registration and default calendar creation are covered in Section 2.6.

To schedule an appointment, the user selects the 'Appointment' command in the Schedule menu. In response, the system displays the dialog shown in Figure 6. The figure shows the appointment scheduling dialog that contains a number of data-entry fields for the various components of the appointment. The 'Title' field is a one-line string that describes the appointment briefly. The 'Date' field contains the date on which the appointment is to occur. The 'End Date' field applies only to recurring appointments; it is initially disabled (physically greyed out) since the default for an appointment is non-recurring. The 'Start Time' and 'Duration' fields specify the time the appointment starts and how long it lasts.

The information in the fourth row of the appointment dialog is used to specify whether, and if so how often an appointment recurs. Immediately to the right of the 'Recurring?' label is a check box. If it is checked on, then information pertinent to a recurring appointment changes from disabled (greyed-out) to enabled (solid type). The recurring information is the 'End Date', the recurring 'Interval', and the seven daily check boxes to the right of the 'Interval' box. The 'Interval' selection is one of 'weekly', 'biweekly', 'monthly', or 'yearly'. Complete details on scheduling recurring items are covered in Section 2.5.3.

Below the recurring information are additional data entry boxes. The 'Category' entry is a color-coded text string that categorizes the appointment for listing and filtering purposes. The appointment 'Security' indicates what portions of a user's calendar other users may see. The four security levels are: 'public', 'title only', 'confidential', and 'private'. The meaning of the four security levels is as follows:

- public -- other users can see all details of the appointment
- title only -- other users can see the title, time, date, and duration of the appointment, but no other information
- confidential -- other users can see the time, date, and duration of the appointment, but no other information; the appointment title shows as "Unavailable" to other users
- private -- other users can see no information at all about the appointment

Details of viewing other users' calendars are discussed in Section 2.3.5.

The appointment 'Location' specifies where it will be held. The location value can be typed manually by the user or selected from a list of available locations, which are stored in the locations database. Further details of location definition and selection are covered in Sections 2.6.4 and 2.11.7 respectively.

The appointment 'Priority' indicates whether an appointment is a 'must' or 'optional'. The priority is used to indicate the general importance of an appointment to the user. The priority is also used in meeting scheduling, where optional appointments can be automatically scheduled over if necessary. Meeting scheduling is discussed further in Section 2.4.1.

The data-entry row starting with 'Remind?' is used to enable the sending of a reminder prior to the start of the appointment. When the 'Remind?' check box is on, the other reminder information changes from disabled to enabled. The two data-entry fields immediately to the right of the check box indicate how soon before the

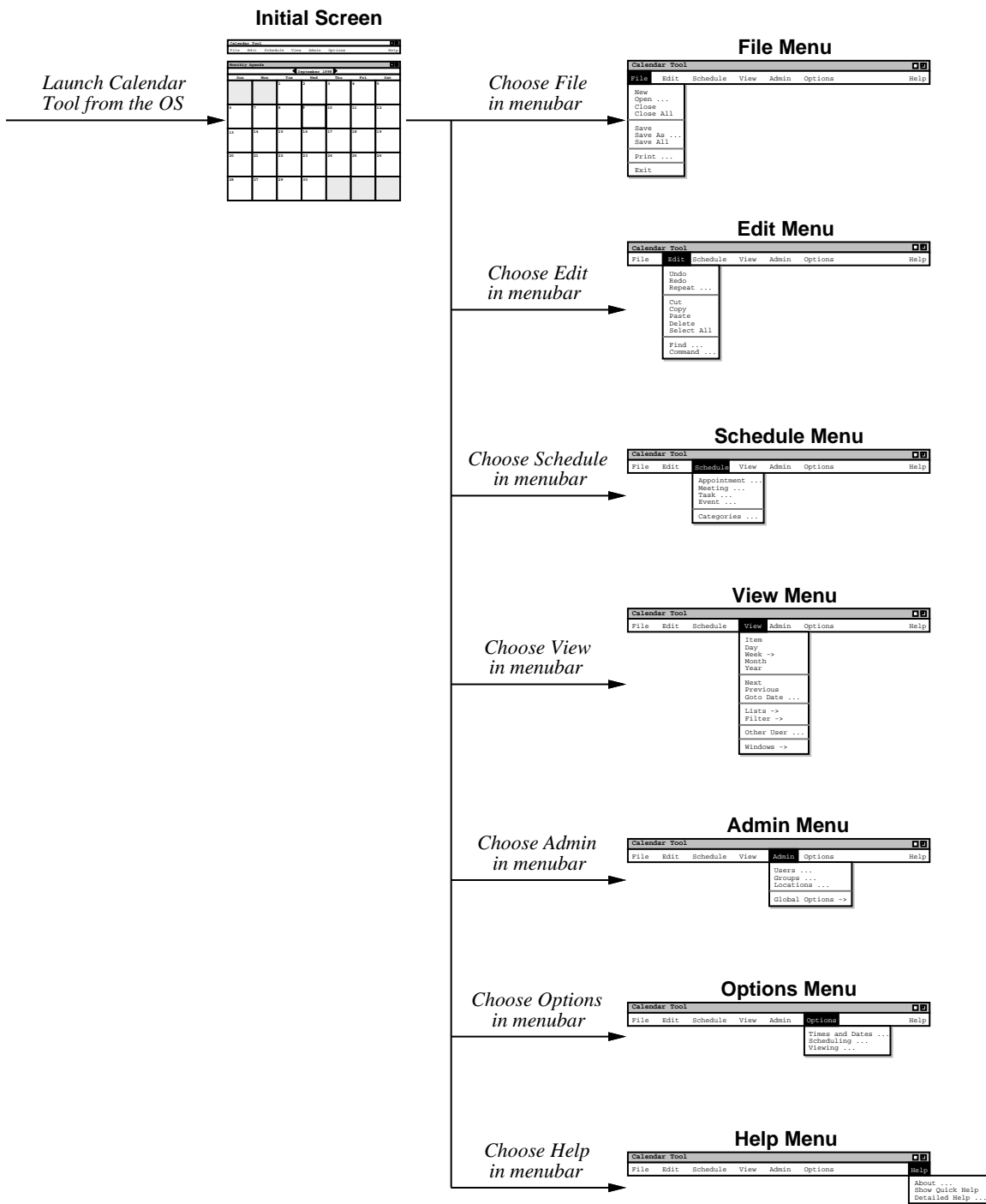


Figure 5: Top-level screen map.

Figure 6: Appointment scheduling dialog.

appointment the reminder is sent. The small box is a numeric value and the other box is one of the three alternatives ‘minutes before’, ‘hours before’, or ‘days before’. To the right of the time-before entry is a menu for selecting how the reminder should be sent. The choices are ‘on screen’ or ‘email’. Details of these reminder formats are covered in Section 2.5.4.

The bottommost data-entry area in the appointment dialog is for writing a detailed description of the appointment. The user may enter any amount of free-form textual information in the ‘Details’ field.

After the dialog appears on the screen, the user enters information by a combination of typing and selecting from pre-defined choices. Figure 7 shows the result of the user having filled in information for a new appointment titled "Dentist". The dentist appointment has a date, start time, and duration, but no end date since it is non-recurring. Since this is the user’s first time on the Calendar Tool, the only pre-defined appointment category is "Holiday". For the dentist appointment, the user decides to create a new category called ‘personal’.

To create a new appointment category, the user initially selects the down arrow next to the ‘Category’ display box. In response, the system displays the menu shown in Figure 8. Since the user wants to create a new category, she selects the ‘Edit . . .’ command at the bottom of the menu, whereupon the system displays the dialog shown in Figure 9. The user then selects the ‘Add’ button, which brings up the dialog in Figure 10. To create the new ‘personal’ category, the user types into the ‘Category Name:’ text box, and then selects ‘Purple’ from the color selection menu, as shown in Figure 11. The user then presses the ‘OK’ button to confirm the creation of the new category. To complete the edit, the user presses the ‘Done’ button in the still pending edit-categories dialog. At this point, the system adds the purple-colored text for the newly created category, as shown in Figure 12.

To confirm the scheduling of the dentist appointment, the user presses the ‘OK’ button in the main schedule-appointment dialog. At this point, the dentist appointment is added to the user’s personal calendar.

The user now proceeds to create another appointment, following the same procedure as above for the dentist appointment. The resulting filled-in appointment dialog is shown in Figure 13. In this case, the appointment is recurring, scheduled for each Tuesday and Thursday between the specified start and end dates. Note that when the recurring check box is selected, the ‘Date’ field label is changed to ‘Start Date’ to clarify the display.

Schedule an Appointment

Title: Dentist

Date: September 25, 1998 Start Time: 8 AM

End Date: Duration: 1 hr 30 min

Recurring? Interval: weekly S M T W Th F S

Category: Security: title only

Location: 1342 Sycamore Dr Priority: must

Remind? 1 day before on screen

Details:

OK Clear Cancel

Figure 7: Scheduling a dentist appointment.

Category:

- none
- holiday
- Edit ...

Figure 8: Initial categories menu.

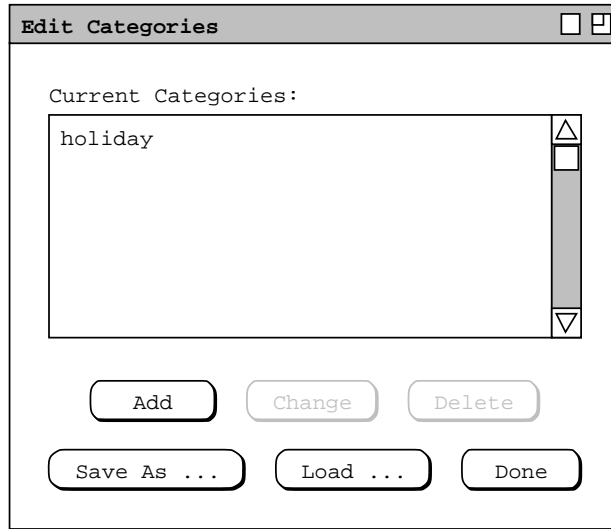


Figure 9: Edit categories dialog.

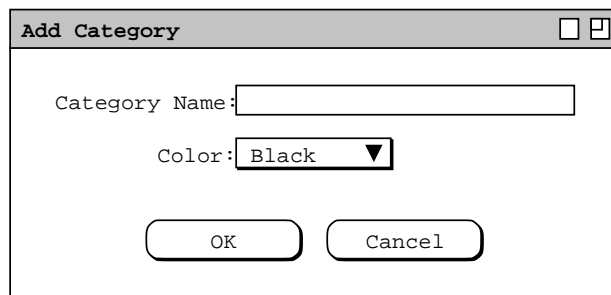


Figure 10: Add category dialog.

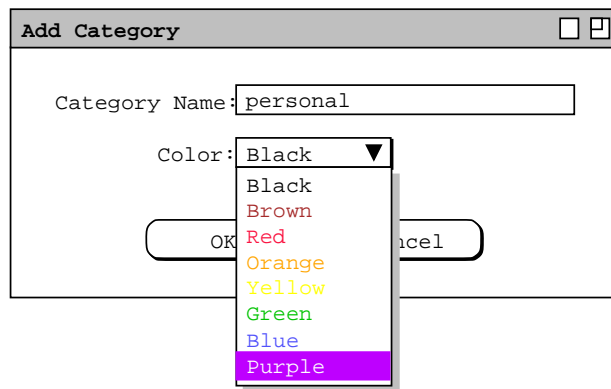


Figure 11: Creating a new category.

Schedule an Appointment

Title: Dentist

Date: September 25, 1998 Start Time: 8 AM

End Date: Duration: 1 hr 30 min

Recurring? Interval: weekly S M T W Th F S

Category: personal Security: title only

Location: 1342 Sycamore Dr Priority: must

Remind? 1 days before on screen

Details:

OK Clear Cancel

Figure 12: Dentist appointment with category filled in.

Schedule an Appointment

Title:

Start Date: Start Time:

End Date: Duration: hr min

Recurring? Interval: S M T W Th F S

Category: Security:

Location: Priority:

Remind?

Details:

Figure 13: Scheduling a second appointment.

The user has set the priority for the new appointment to 'optional' instead of 'must'. The operative effect of selecting an 'optional' priority relates to scheduling meetings, which is discussed further in Section 2.4.1. Aside from meeting scheduling, the meaning of 'must' versus 'optional' is purely informational, i.e., it indicates the relative importance of the appointment.

The information entered for a recurring appointment initially applies to all occurrences of the appointment. When the user confirms a recurring appointment with the 'OK' button, the appointment is entered into the user calendar on all of the specified dates. Subsequently, the user can change some or all of the information on a recurring appointment for some or all of the recurring dates. Details of how a recurring appointment is changed are covered in Section 2.5.2.1

To clear all information entered in an appointment dialog, the user presses the 'Clear' button. In response, the system clears all typing areas and sets all other data-entry fields to their default states. To cancel a scheduling command entirely, the user presses the 'Cancel' button. In response, the system clears the dialog and removes it from the screen, without performing any scheduling action. Details of the precise effects of 'Clear' and 'Cancel' are covered in Section 2.11.24.

After scheduling and confirming an appointment, the appointment data are entered in an online working copy of the user's calendar. In order for the appointments to be permanently saved in a calendar file, the user must invoke the 'File Save' command, as discussed in Section 2.8.2. The user may save calendars in as many different files as desired. However, only specifically designated calendar files are visible to other Calendar Tool users and hence available for scheduling group meetings. Details of meeting scheduling and access to other users' calendars are covered in upcoming sections of the requirements.

After the user presses the 'OK' button in the scheduling dialog, the dialog itself is removed from the screen. The complete information for a scheduled appointment can be viewed by selecting the 'Item' command in the 'View' menu, or by double clicking on the title of an appointment in a daily, weekly, or monthly calendar display. Details of these displays are covered in the next section of the requirements.

2.3. Viewing Calendars

The scenarios in this section show a user viewing her personal calendar in a variety of ways, in particular:

- viewing the calendar at different levels of detail
- moving backward, forward, and to specific dates
- viewing lists of scheduled items
- filtered viewing

There is also a scenario showing how a user views the public portions of another user's calendar, including a group's "virtual" calendar. The viewing section of the requirements concludes with a scenario showing how to control multiple viewing windows, including the creation of new windows for side-by-side displays.

2.3.1. Viewing the Calendar at Different Levels

A calendar can be displayed at five levels of detail. The levels, from the greatest detail to the least are: an individual scheduled item, a day, a week, a month, and a year. The commands to show these levels are the first five items in the 'View' menu (see Figure 2). In the discussion that follows, these are referred to as the *view-level* commands.

During normal use of the Calendar Tool, the user has one or more viewing windows displayed on the screen. The most recently selected window is called *current*. When the user wants to view the calendar at a different level than is displayed in the current window, she chooses one of the five view-level commands.

Normally, each of the five viewing levels is displayed in a separate level-specific window, sized appropriately to fit its contents. For example, if the user executes a 'View Day' command followed by a 'View Year' command, both a daily and yearly window are displayed on the screen.

By default, one separate window is used at each of the five view levels. Hence, there are up to five active windows for the view-level commands, each holding the result of the most recently executed command at a particular level. When the user changes the display at a particular level, that level's display is changed and brought to the front, without a new window being created. For example, if the user moves from one day to the next at the daily level, the day-level display window is changed, with no new window created.

The user may change the default display style by selecting *Two-Window* or *Multi-Window* viewing mode in the Windows menu. In two-window mode, the results of all viewing commands are shown in just two windows, one for calendars and the other for lists. In multi-window mode, the results of each and every view command are shown in a separate new window, even for commands at the same level. For example, if the user executes two successive 'View Month' commands in multi-window mode, the results are shown in two separate month-view windows. This allows the user to create side-by-side displays of different calendar periods. Complete details of two-window and multi-window viewing are described in Section 2.3.6.

By default, the target date for all view-level commands is today's date. The user can select another target by clicking on a desired date in the current window. For example, the user can click on a specific date in a month view and choose the 'View Day' menu item. In response, the system displays the day-level view for the selected date. Complete details of target date selection are covered in Section 2.11.9

2.3.1.1. Day View

To view the daily details for today's date, the user chooses the 'View Day' menu command. In response, the system displays a window of the form shown in Figure 14. To view the details of a day other than today, the user selects the name or date of the desired day in the current display, and then chooses the 'View Day' menu command. An interface shortcut to view a particular day is to double click on the name or date of the desired day in the current display.

In Figure 14, the user is viewing the details of Wednesday, September 23, which is today's date in this scenario. The items shown in Figure 14 reflect a number of scheduling commands having been performed by the user, subsequent to the commands illustrated in the scenario of Section 2.2. In particular, the user has scheduled some typical items such as meetings, lunch dates, and class lectures.

Daily Agenda	
< Today > Wednesday, September 23, 1998	
8 AM	Staff Meeting
9 AM	Office Hours
10 AM	Data Structures Lecture
11 AM	Data Structures Lab
12 PM	Lunch with Microsoft
1 PM	-----
2 PM	----- CAD Research Project Meeting
3 PM	-----
4 PM	-----
5 PM	

Figure 14: Day view.

The day view shows the scheduled items for a day in abbreviated form. The column on the left lists the starting times for scheduled items. The wider column on the right shows the item titles. The displayed title for one item is always a single line high; long titles are viewed by horizontal scrolling or enlarging the display window as necessary. The following are details of how information is presented in a day view:

- When an item starts and ends exactly on the hour, it is listed at the top of the row for the starting hour. For example, the "Staff Meeting" item shown in Figure 14 starts on the hour, and is thus displayed at the top of the 8AM to 9AM time slot.
- When an item starts at a time other than on the hour, the text for the item is placed proportionally down from the top of the hour, with a dashed line immediately above the text. For example, the "CAD Research Project Meeting" starts at half-past the hour, and is thus displayed half way down in the 2PM to 3PM time slot.
- When an item ends at a time other than on the hour, a dashed line is placed proportionally at the ending time. For example, the "CAD Research Project Meeting" ends at 4:30PM, and its ending time is thus displayed as a dashed line in the middle of the 4PM to 5PM time slot.
- When an item ends in an hour after the hour in which it begins, a down-pointing arrow is drawn to the ending time. This case is also illustrated by the "CAD Research Project Meeting".

Immediately below the window banner in the day view is a full description of the day of the week, month, and year. To the far left of the day description are three buttons -- a left-pointing arrow, right pointing arrow, and one labeled 'Today'. The arrow buttons are used to change the display to the previous or next calendar day. The 'Today' button is used to return to today's date. For example, when the user chooses the right-pointing arrow in the display of Figure 14, the display changes to that shown in Figure 15. Figure 15 shows a display when events and tasks are scheduled, in addition to appointments and meetings. When all four types of item are scheduled, the items are organized into three display areas:

- events are listed immediately below the full date label;
- tasks are listed below the events;
- following the tasks are the co-mingled appointments and meetings, listed in time slots.

Daily Agenda	
<div style="display: flex; justify-content: space-between; align-items: center;"> < Today > Thursday, September 24, 1998 </div>	
Events:	
Jim's Birthday	
Tasks:	
1. Prepare for Picnic Day Committee 2. Buy something for Jim	
8AM	Racket Ball
9AM	Picnic Day Committee
10AM	----- Software Engineering Colloquium
11AM	
12PM	
1PM	----- Office Hours
2PM	
3PM	Software Engineering Graduate Seminar
4PM	
5PM	

Figure 15: Result of pressing the next day arrow.

If more than one event is scheduled, the events are displayed in a multi-line list, one event title per line, sorted alphabetically by title. Event titles are shown in boldface font. Tasks are displayed in a numbered list, one task title per line, sorted first by due time, second by priority, and third by title. Tasks without due times appear at the end, sorted first by priority and second by title.

When no events or tasks are scheduled for a particular day, those display areas are absent. For example, Figure 14 shows the case where no events or tasks are scheduled for a day, whereas Figure 15 shows the case where both events and tasks are scheduled. If only events but not tasks are scheduled, or vice versa, the scheduled events or tasks are displayed, and the other display area is absent.

The following options are available to control the format of the day display:

- a. *Normal Time Range.* This option defines the starting and ending times that normally appear in the left column of the display. The default range of times is 8AM to 5PM.
- b. *Time Increment.* This option defines the time increment of the left column labels. The smallest allowable time increment is five minutes, the largest is twelve hours. When the time increment is less than one hour, the time labels are shown in hour and minute format; when the increment is one hour or greater, the labels are shown as hours with no minutes. The default increment is one hour.
- c. *Increment Height.* This option defines the default minimum number of text lines shown in each time increment. If there are more items starting in an increment than fit in the minimum number of lines, the system automatically increases the height for that increment to a size sufficient to display all items fully. The

default value is two lines per increment.

- d. *Events HeightP and Tasks Height.* These options define the maximum number of text lines shown in the event and task areas before vertical scrolling is invoked. If there are more tasks or events than fit in these heights, the system adds vertical scroll bars to one or both areas to allow the additional items to be viewed via scrolling. The default values are two lines for each, not including the labels at the top of each area.
- e. *Show/Hide 24 Hours.* This option toggles between showing a full 24 hours of display versus the normal time range only. Non-normal times are shown with a light grey background, to distinguish them from normal times. Setting this option to hide never hides scheduled items, only non-normal hours during which no items are scheduled. The default value for this option is to hide non-normal times.
- f. *Show/Hide Exact Time.* This option toggles between showing or not showing the starting and ending times for each scheduled item immediately to the left of the title. This option is useful when items do not typically start or end exactly on the hour. The default value for this option is to hide the exact time.
- g. *Show/Hide Dashed Lines.* This option toggles between showing and not showing the dashed lines that appear above and below items that do not start or end on the hour. The default is to show dashed lines.
- h. *Show/Hide Extension Arrows.* This option toggles between showing and not showing the down-pointing arrows for items with end times that extend beyond the hour of the start time. The default is to show extension arrows.
- i. *Proportional Spacing On/Off.* This option toggles between placing items proportionally down from the top of the hour based on starting time. When the option is off, the first (only) item in an hour is placed at the top of that hour and any other items starting in the hour are placed immediately below it. The default is proportional spacing on.
- j. *Display Overlaps Horizontally or Vertically.* This option controls how items with overlapping times are displayed. When displayed horizontally, overlapping items appear side-by-side in separate columns of the day view. When displayed vertically, overlapping items appear on separate lines in a single column, sorted by start time, duration, and title. The default is horizontal display.
- k. *Default Display Width and Default Display Height.* These options control the default width and height of the full daily display. If one or both of the default dimensions is too small to accommodate all information in the display, the information is truncated appropriately and scrollbars are added to the display window. Details of how this truncation is performed are covered in Section 2.13.5.

The user sets these options using the dialogs described in Section 2.7.4.

As an example of alternate option settings, Figure 16 shows a day view of the same information shown in Figure 14, with the following non-default option settings: normal time range 7AM to 6PM; time increment of 1/2 hour; increment height equal to one line; show 24-hour display; show exact time; hide dashed lines; and hide extension arrows. Figure 16 shows a fully expanded window, with all 24 hours of the day visible. If the user shrinks the window display vertically, a vertical scrollbar appears to allow viewing the non-visible hours. Further details on the effects of display resizing are covered in Section 2.13.5. Specific details on the effect of all the display options are presented in Section 2.7.4.

To illustrate the display of overlapping items, Figure 17 shows an updated version of the daily schedule shown in Figures 14 and 16. Here the user has added three new items, with times that overlap with previously scheduled items. The new items are a "TA Meeting" from 8 - 9:30 AM, a "College Meeting" from 2 - 3 PM, and a "Special Colloquium" from 2 - 3:30 PM. The precise definition of overlap is the following: Two or more scheduled items overlap if they are scheduled on the same day and any portion of their scheduled time periods overlaps. The scheduled time period of an item is its start time plus its duration.

The overlapping items are shown in two new columns next to the items with which they overlap. The left-to-right order of items is based first on start time (earliest first), second on duration (shortest first), and third on alphabetic order of the title. Note that this ordering has placed the "CAD Research Project Meeting" in the rightmost column, even though this item was scheduled by the user before the items with which it overlaps. That is, the order of overlapping items does not depend at all on the chronological order in which the user executes scheduling commands.

The layout of overlapping items may be further complicated when three or more items do not overlap fully, for example if the user were to add an appointment from 3-4PM to the schedule shown in Figure 17. The precise layout rules for such cases are covered in Section 2.13.5.3.

Daily Agenda □ □	
◀ Today ▶ Wednesday, September 23, 1998	
12:00 AM	
12:30 AM	
1:00 AM	
1:30 AM	
2:00 AM	
2:30 AM	
3:00 AM	
3:30 AM	
4:00 AM	
4:30 AM	
5:00 AM	
5:30 AM	
6:00 AM	
6:30 AM	
7:00 AM	
7:30 AM	
8:00 AM	8-9 AM Staff Meeting
8:30 AM	
9:00 AM	9-10 Office Hours
9:30 AM	
10:00 AM	10-11 AM Data Structures Lecture
10:30 AM	
11:00 AM	11 AM-12 PM Data Structures Lab
11:30 AM	
12:00 PM	12-1:30 PM Lunch with Microsoft
12:30 PM	
1:00 PM	
1:30 PM	
2:00 PM	
2:30 PM	2:30-4:30 PM CAD Research Project Meeting
3:00 PM	
3:30 PM	
4:00 PM	
4:30 PM	
5:00 PM	
5:30 PM	
6:00 PM	
6:30 PM	
7:00 PM	
7:30 PM	
8:00 PM	
8:30 PM	
9:00 PM	
9:30 PM	
10:00 PM	
10:30 PM	
11:00 PM	
11:30 PM	

Figure 16: Day display with non-default option settings.

Daily Agenda			
◀ Today ▶ Wednesday, September 23, 1998			
8 AM	Staff Meeting	TA Meeting	
9 AM	Office Hours	-----	
10 AM	Data Structures Lectur		
11 AM	Data Structures Lab		
12 PM	Lunch with Microsoft		
1 PM	-----		
2 PM	College Meeting	Special Colloquium	CAD Research Project M
3 PM		-----	-----
4 PM			-----
5 PM			

Figure 17: Day view with overlapping times.

In displaying overlapping items horizontally, the system attempts to display the full width of all items in one screen. If there is insufficient physical screen width to display all items in full width, the system truncates the display, leaving overlap columns of equal width. If the display window is resized, each column is resized proportionally and equally. Further exact details of truncation and display resizing are covered in Section 2.13.5.2.

Figure 18 shows the same information as Figure 17, after the user has changed the overlapping display option from *horizontal* (the default) to *vertical*. When the user selects the vertical display option for overlapping items, the system automatically turns on the *Show Exact Times* option and turns off the *Dashed Lines*, *Extension Arrows*, and *Proportional Spacing* options. Turning off these three options is necessary since they are generally incompatible with the vertical display of overlaps. (The reader is invited to think this through clearly.) If the user subsequently sets the overlapping-item display option to horizontal, the system restores the settings of all automatically-set options to their values prior to their automatic setting.

Of note in Figure 18 is the automatic resizing of the 2PM time slot from two to three rows, even though the current setting of the *Increment Height* option is 2. Whenever the number of items in a time increment is greater than the current increment height, the system automatically increases the height of the affected increment to show all items that start in that increment. This automatic height increase is performed as necessary for any time increment in a day view, not just for increments with overlapping items.

2.3.1.2. Week Views

After the day view, the next larger-grain level of viewing is a week. There are two choices for the format of the week view: table and lists. These choices are available on a submenu when the user chooses the 'Week' command from the 'View' menu. When the user chooses the table-style week view, the screen appears as shown in Figure 19.

When the user chooses either style of week view, the displayed week is the one that contains the most recently selected day. Here for example, the user is viewing the week that contains the most recently viewed day of Wednesday, September 23. The current date, that is today's date, is highlighted with a darkened border in the week display. In this scenario, for example, the current date is Wednesday, September 23.

Daily Agenda □ ▢	
◀ Today ▶ Wednesday, September 23, 1998	
8 AM	8-9 AM Staff Meeting 8-9:30 AM TA Meeting
9 AM	9-10 AM Office Hours
10 AM	10-11 AM Data Structures Lecture
11 AM	11 AM-12 PM Data Structures Lab
12 PM	12-1:30 PM Lunch with Microsoft
1 PM	
2 PM	2-3 PM College Meeting 2-3:30 PM Special Colloquium 2:30-4:30 PM CAD Research Project Meeting
3 PM	
4 PM	
5 PM	

Figure 18: Day view with overlapping times, displayed vertically.

Weekly Agenda							
◀ Today ▶		Week of September 20-26, 1998					
	Sun	Mon	Tue	Wed	Thu	Fri	Sat
Events:		Autumnal			Jim's Bir		
Tasks:		1. Colloq			1. Prepar 2. Buy so		
8 AM			Racket ba	Staff M ▶	Racket ba	Dentist ▶	
9 AM		Office ho	Research	Office ho	Picnic da	↓	
10 AM		Data Stru		Data Stru	↓	Data Stru	
11 AM		Data Stru		Data Stru	↓	Data Stru	
12 PM				Lunch wit			
1 PM		Faculty M		↓	Office ho		
2 PM				College ▶	↓		
3 PM		Software			↓	Software	
4 PM		↓			↓		
5 PM							

Figure 19: Table-style week view.

The table-style weekly view shows items in the same format as the day view, except with narrower columns to accommodate all the days of the week. Each item is shown with as much of the title as fits horizontally within its display area. The columns of the weekly table view are all of equal width. If the display window is resized, each column is resized proportionally and equally. Immediately below the window banner in the weekly view is the exact description of the week. To the left of the week description are arrow buttons to go to the previous and next weeks, and a 'Today' button to return to the week containing today's date. As with the day view, if there are no events or tasks scheduled for the entire week, then the event and/or task display areas are absent.

A notable aspect of the weekly table view is the display format for days that contain overlapping items. In the example shown in Figure 19, there are three overlapping appointment times. These overlaps are indicated by solid right-pointing arrows wherever overlap occurs. When the user presses a right-pointing arrow, the display for the column scrolls to the left to allow overlapping items to be viewed.

Figure 20 further illustrates the display of overlapping weekly table items. The figure shows four excerpts of the weekly table view for the week of September 20, illustrating four stages of viewing overlapping items. In Figure 20a, three right-pointing arrows appear at Wednesday 8 AM, Wednesday 2 PM, and Friday 8 AM. Each arrow is positioned vertically at the start time of an overlapping item.

When the user clicks on the right-pointing arrow at Wednesday 2 PM, the list of 2 PM overlapping items scrolls to the left. The result is shown in Figure 20b, with the changed item circled. Figure 20c shows the result of the user clicking on the right-pointing arrow at Wednesday 8 AM. The display change is again circled. Of note in Figure 20c is that the Office Hour item at 9 AM has been removed from the display. When overlap scrolling in a weekly view, the system displays only complete overlapping items, no partial items. That is, the system removes all items from the display with which a scrolled-to item fully or partially overlaps.

Figure 20d shows the result of the user clicking on the right arrow at Friday 8 AM. This change again illustrates how only one overlapping item is displayed at a time. That is, the Dentist appointment at 8 AM is not displayed at

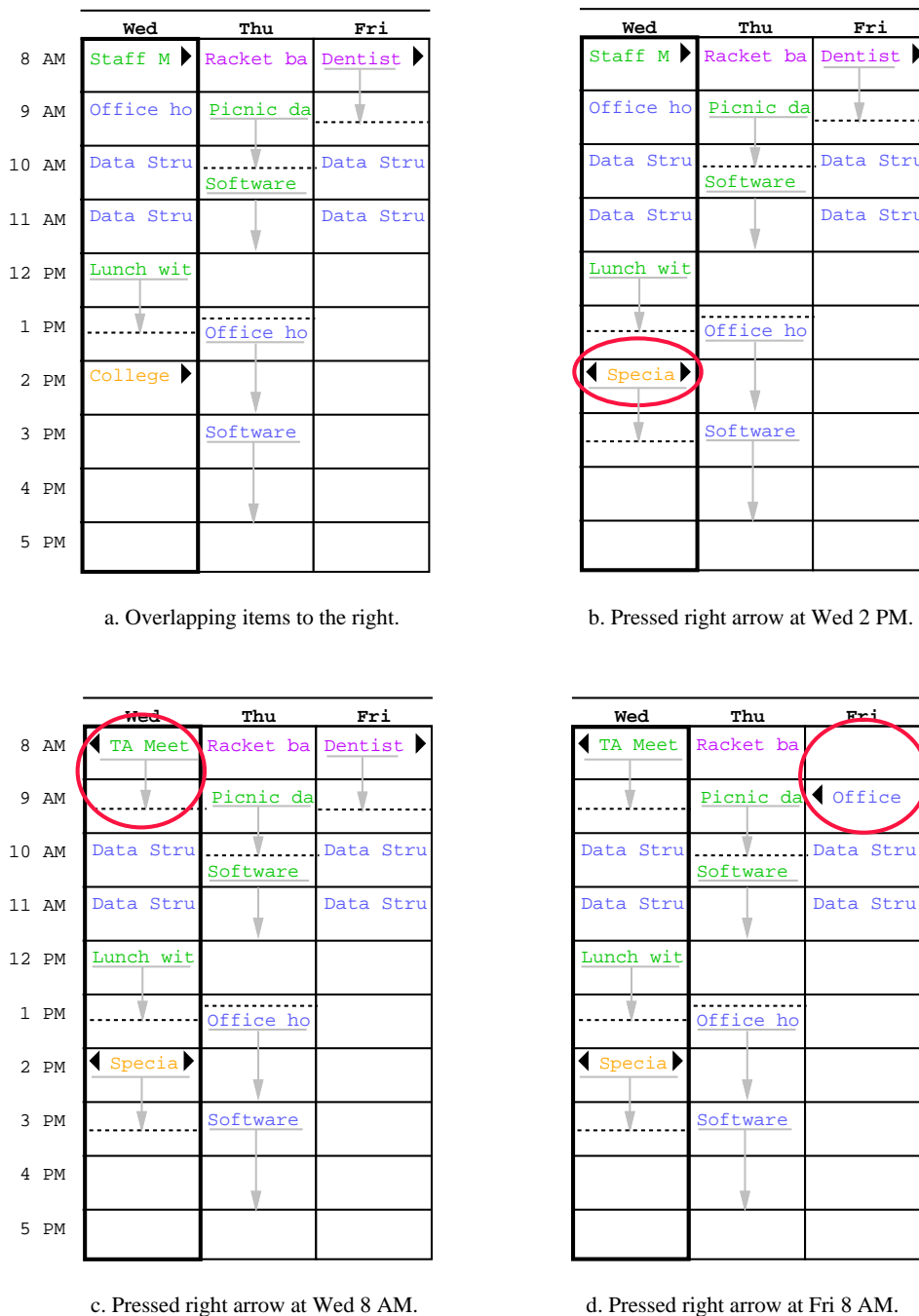


Figure 20: Clarification of overlap in a weekly table view.

all in 20d, even though it would fit partially in the display. Figure 20d also illustrates that the left-pointing arrow always appears at the top of the item that is displayed. Hence, the left-pointing arrow in 20d appears next to the 9 AM office hour appointment, not in the empty 8AM slot where the Dentist appointment previously appeared.

When necessary, a left-pointing arrow appears to allow scrolling back. Examples of left-pointing arrows are shown in Figures 20b, 20c, and 20d. The 2 PM item in Figures 20b-d has both left- and right-pointing arrows, since there are overlapping items on both sides. When there are no further overlapping items, the left- or right-pointing arrow disappears, as appropriate. Disappearance of the right-pointing arrow is illustrated in the Wednesday 8 AM item in

Figure 20c and the Friday 9 AM item in Figure 20d.

All of the display options available for the day view are also available for the table-style week view. In addition, the following display options are available for the week view:

- a. *Show/Hide Title*. This option toggles between showing or not showing the title for each scheduled item. When the user selects to hide item titles, the system automatically turns on the *Show Exact Times* option, if it is not on already. This prevents both the exact time and title from being hidden simultaneously. The default value for the show/hide title option is to show the title.
- b. *Normal Day Range*. This option defines the starting and ending days that appear along the top of the display. The default range is Sunday through Saturday, inclusive.
- c. *Show/Hide 7 Days*. This option toggles between showing all 7 days of the week versus the range specified as the normal day range, if it is different. Non-normal day columns are shown with a light grey background, to distinguish them from normal days. The default value for this option is to show non-normal times.

The display of a list-style weekly view is shown in Figure 21. The Figure shows the same week as in the table-style view in Figure 19. In the list-style weekly view, the exact time for each item is always displayed, since there is no times column as in the table-style view. The display sizing constraints for the list-style view are the same as for the table-style view. Specifically, each item is shown with as much of the time and title as fit horizontally within its display area. The columns of the list-style view are all of equal width. If the display window is resized, each column is resized proportionally and equally.

For each day of the week, the items for that day are listed consecutively in the column for that day. The list starts with bold-faced event titles, if any, sorted alphabetically by title. Following the events are the scheduled tasks, if any. Tasks are numbered starting from 1, sorted first by due time, second by priority, and third by title. Tasks without due times appear at the end, sorted first by priority and second by title. After the tasks are the appointments and meetings, sorted first by start time (earliest first), second by duration (shortest first), and third by alphabetic order of the title. No further sorting criteria are necessary for appointments and meetings, since the system disallows multiple scheduled items with exactly the same values for date, start time, duration, and title. A subset of the table-style display options are applicable to the list-style weekly view. These options are:

- a. *Show/Hide Title*.

Weekly Agenda □ ▢						
◀ Today ▶ Week of September 20-26, 1998						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
20	21	22	23	24	25	26
	1. Colloq 9-10 AM O 10-11 AM 11 AM-12 1-2 PM Fa 3-5 PM So	Autumnal 8-9 AM Ra 9 AM-5 PM	8-9 AM St 8-9:30 AM 9-10 AM O 10-11 AM 11 AM-12 12-1:30 P 2-3 PM Co 2-3:30 PM 2:30-4:30	Jim's Bir 1. Prepar 2. Buy so 8-9 AM Ra 9-10:30 A 10:30 AM- 1:15-2 PM 3-5 PM So	8-9:30 AM 9-10 AM O 10-11 AM 11 AM-12	

Figure 21: List-style week view.

b. *Normal Day Range.*

c. *Default Display Width and Default Display Height.*

A single set of display option values applies to both the table-style and list-style weekly views.

2.3.1.3. Month View

An example of the month view was shown originally in the initial screen configuration (Figure 1). since the month view is the standard default. It is chosen explicitly with the 'Month' command from the 'View' menu. Figure 22 shows the result of the user making this selection for the month of September 1998. The Figure shows the scheduled items that have been added during the course of the scenarios up to this point. The current date is highlighted with a darkened border. Immediately below the window banner is the name of the month, with arrow buttons to go to the previous and next months, and a 'Today' button.

In the month view, the display format for an individual day is the same as the list-style weekly view. That is, the time and title for each item are displayed in a vertically-formatted list, sorted by start time, duration, and title. Since the height of each day in the month view is considerably shorter than in the week view, a transient vertical scrollbar is provided when the number of items in a day is greater than fits vertically in the display box for a day. "Transient" for the scrollbar means that it only appears when the user clicks within a day for which scrolling is necessary. For example, if the user clicks within the display box for Monday September 21 (in Figure 22), the scrollbar appears as shown in Figure 23. The scrollbar appears since there are more items scheduled for September 21 than fit in the current size of the September 21 box. If the user clicks within a day where scrolling is not necessary, such as September 22, no scrollbar appears. A displayed transient scrollbar for any day disappears when the user clicks anywhere outside of the day in which the scrollbar appears.

Monthly Agenda						
◀ Today ▶		September 1998				
Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1 8-9 AM Ra 11 AM-2 P	2	3 8-9 AM Ra	4	5
6	7 Labor Day	8 8-9 AM Ra 11 AM-2 P	9	10 8-9 AM Ra	11	12
13	14	15 8-9 AM Ra 11 AM-2 P	16 1. Send c	17 8-9 AM Ra	18	19
20	21 1. Colloq 9-10 AM O 10-11 AM 11 AM-12	22 Autumnal 8-9 AM Ra 9 AM-5 PM	23 8-9 AM St 8-9:30 AM 9-10 AM O 10-11 AM	24 Jim's Bir 1. Prepar 2. Buy so 8-9 AM Ra	25 8-9:30 AM 9-10 AM O 10-11 AM 11 AM-12	26
28	27 1. Colloq 9-10 AM O 10-11 AM 11 AM-12	29 8-9 AM Ra 9 AM-5 PM	30 9-10 AM O 10-11 AM 11 AM-12 2:30-4:30			

Figure 22: Month view.

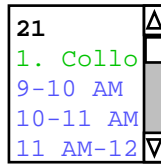


Figure 23: Transient scrolling in a month view.

A subset of weekly list-style display options are applicable to the month view. These options are:

- a. *Show/Hide Title.*
- b. *Default Display Width and Default Display Height.*

These options have the same meaning and defaults as for the list-style weekly view.

2.3.1.4. Year View

The largest-grain calendar view is a year, as shown in Figure 24. The current date is highlighted with a rectangular

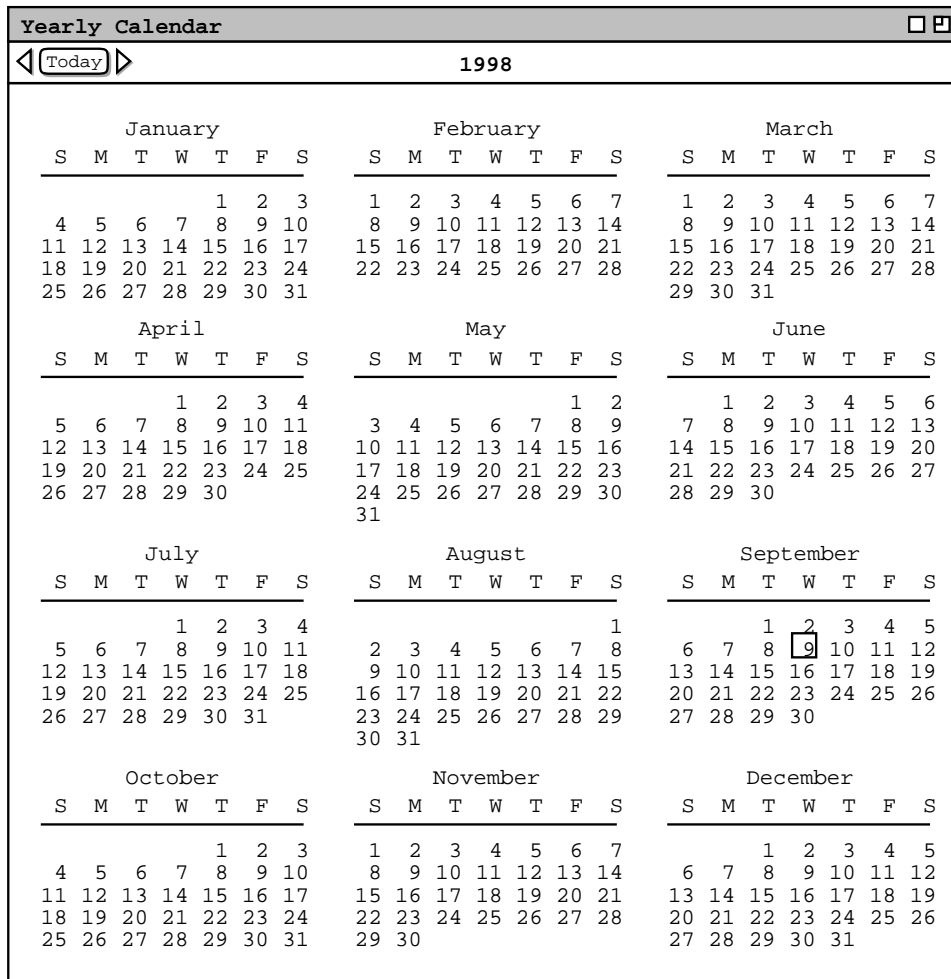


Figure 24: Year view.

border. Immediately below the window banner the year number, with arrow buttons to go to the previous and next years, and a 'Today' button.

Unlike the smaller-grain views, the yearly view shows no information about scheduled items. The user may select any date as the target of another view command to show item details. The only display option that applies to the yearly view is *Default Display Width* and *Default Display Height*.

2.3.1.5. Scheduled Item View

The user may view the full scheduling details for an item shown in any day, week, or month display. To do so, the user selects the item time or title and then chooses the 'View Item' menu command. A short cut to view item details is to double click on the item time or title in any larger-grain view.

Figure 25 shows the user having selected the item for 8 AM September 25 in a monthly view. When the user clicks anywhere in the item time or title, the system highlights the entire item. When the user then selects the 'Item' command in the 'View' menu, the system displays the item level view in Figure 26. This is the item-level view for an appointment. Details of item-level viewing for meetings, tasks, and events are presented later in Section 2.5.1.

The information displayed for an item is editable by the user. This editability distinguishes the item-level view from the larger-grain daily, weekly, and monthly views. That is, none of the larger-grain displays allows the user to edit any of the displayed item information.

Monthly Agenda □ □						
◀ Today ▶		September 1998				
Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1 8-9 AM Ra 11 AM-2 P	2	3 8-9 AM Ra	4	5
6	7 Labor Day	8 8-9 AM Ra 11 AM-2 P	9	10 8-9 AM Ra	11	12
13	14	15 8-9 AM Ra 11 AM-2 P	16 1. Send c	17 8-9 AM Ra	18	19
20	21 1. Colloq 9-10 AM O 10-11 AM 11 AM-12	22 Autumnal 8-9 AM Ra 9 AM-5 PM	23 8-9 AM St 8-9:30 AM 9-10 AM O 10-11 AM	24 Jim's Bir 1. Prepar 2. Buy so 8-9 AM Ra	25 8-9:30 AM 9-10 AM O 10-11 AM 11 AM-12	26
28	27 1. Colloq 9-10 AM O 10-11 AM 11 AM-12	29 8-9 AM Ra 9 AM-5 PM	30 9-10 AM O 10-11 AM 11 AM-12 2:30-4:30			

Figure 25: Item selected in monthly view.

The screenshot shows a window titled "Scheduled Appointment" with a close button. Below the title bar is a navigation area with a left arrow, a "Today" button, and a right arrow, followed by the text "8 AM Friday, September 25, 1998". The main area contains several input fields and controls:

- Title: Dentist
- Date: September 25, 1998
- Start Time: 8 AM
- End Date: (empty dashed box)
- Duration: 1 hr, 30 min
- Recurring? Interval: weekly
- Day selection: S M T W Th F S (all boxes are empty)
- Category: personal
- Security: title only
- Location: 1342 Sycamore Dr
- Priority: must
- Remind? 1 days before
- on screen
- Details: (empty text area with scroll bars)

At the bottom are three buttons: "Change", "Delete", and "Clear".

Figure 26: Details of selected scheduled item.

The appointment item display in Figure 26 has the same data fields as the appointment scheduling dialog shown in Figure 6. The item display differs from the scheduling dialog as follows:

- there is a time and date summary in the item display, just below the window banner;
- the command buttons at the bottom of the item display are different than in the scheduling dialog.

The left and right arrow buttons at the top of the display are used to view the previous and next scheduled item, as described further in Section 2.3.2. The most important difference between the item-level display and the corresponding scheduling dialog are the command buttons along the bottom of the display window. Specifically, the scheduling dialog has 'OK', 'Clear', and 'Cancel' buttons. In the item display, the buttons are 'Change', 'Delete', and 'Clear'.

To change the displayed item, the user makes one or more changes in the item data fields and then presses the 'Change' button. As soon as the user changes one or more data fields, the 'Change' and 'Clear' buttons become enabled, and the 'Delete' button becomes disabled. To delete the displayed item (prior to editing any data fields), the user presses 'Delete'. The system responds with a confirmation dialog. If the user confirms affirmatively, the system deletes the scheduled item and removes the display from the screen. To clear any edits made since the dialog was initially displayed, the user presses 'Clear'. This disables the 'Change' and 'Clear' buttons, and re-enables 'Delete'.

The effects of the change and delete commands are potentially wide-ranging, particularly for recurring items or meetings with many attendees. Complete details of item change and deletion are covered in Section 2.5.2

2.3.2. Viewing Previous, Next, and Specific Calendar Dates

Immediately below the 'Year' on the 'View' menu are the three commands 'Next', 'Previous', and 'Goto Date'. These commands provide a quick means to go to another calendar date than is in the current display.

2.3.2.1. Next and Previous

The effect of ‘View Next’ and ‘View Previous’ is based on the level of the current display. At the day, week, month, and year levels, the ‘View Next’ command changes the display to the next calendar unit at that level. At the item level, ‘View Next’ changes the display to the chronologically next scheduled item, if any. The ‘View Previous’ command operates analogously in the backward-time direction. An example of the effect of ‘View Next’ at the day level is illustrated in Section 2.3.1.1 in the transition from Figure 14 to Figure 15.

The right and left arrow buttons at the top of the view-level displays are shortcuts for the ‘View Next’ and ‘View Previous’ commands, respectively. These arrows appear to the far left of the date headers, as shown in Figures 14 , 19 , 22 , 24 , and 26 . Pressing one of these arrows has precisely the same effect as the corresponding menu command. Using the shortcut arrow keys is most effective when multi-window mode is *off*. With multi-window mode off, the user can leave the mouse in a fixed position on the screen while repeatedly pressing a next or previous arrow. This permits the user to move more rapidly in the next and previous directions. Further details of the multi-window display option are covered in Section 2.3.6.

At the item-level of display (Figure 26), moving in the next and previous directions is fundamentally different than at the day through year levels. Specifically, movement at the item level is non-uniform through time. At the day through year level, Next and Previous move exactly one unit of time at the given level. At the item level, Next and Previous move a variable amount of time, based on when the next or previous scheduled item appears in the calendar.

By default, ‘Next’ and ‘Previous’ at the item level move through the complete list of all scheduled items of all four item types, i.e., appointments, meetings, tasks, and events. The day-to-day order of the item list is conveniently visualized in a list-style week view, as shown in Figure 21. Within a given day, the order of scheduled items is as follows:

- a. all events come first, sorted alphabetically by title;
- b. tasks follow the events, sorted first due time, second by priority, and third by title;
- c. the combined list of appointments and meetings come next, sorted first by start time, second by duration, and third by title.

Within a particular day, ‘Next’ and ‘Previous’ move through the items in this order. When a ‘Next’ command is executed at the last item of a day, the display is changed to the first item of the next day that has an item, if any. Similarly, when a ‘Previous’ command is executed at the first item of a particular day, the display is changed to the last item of the previous day that has an item, if any.

The default item-level behavior of ‘Next’ and ‘Previous’ can be altered with the ‘View Filter’ command, described in Section 2.3.4. By setting various filtering options, the user can change which items appear in the next/previous traversal, as well as the beginning and ending time ranges of the traversal.

If the current display window is not a calendar view or if there is no current window, the ‘Next’ and ‘Previous’ commands are disabled in the ‘View’ menu. A calendar view is a display created by one of the five view-level commands, i.e., ‘Item’, ‘Day’, ‘Week’, ‘Month’, or ‘Year’.

2.3.2.2. Today

There are two ways to execute the command that displays information for today’s date. The ‘Today’ menu command is immediately below the ‘Previous’ command in the ‘View’ menu. There is also the small ‘Today’ command button between the left and right arrow buttons at the top left of all view-level displays. Both of these commands have exactly the same effect.

When the user selects a ‘Today’ command, the system changes the current or most recent calendar display to show the information for today’s date. If there is no calendar display, the system displays a new window at the default start-up level. The exact behavior of the ‘Today’ command is the same as when the user enters today’s date for the ‘Goto Date’ command, as described in the immediately following section of the requirements.

2.3.2.3. Goto Date

When the user selects the 'Goto Date' command from the 'View' menu, the system displays the dialog shown in Figure 27. The user enters the desired date and presses the 'OK' button to proceed. For example, Figure 28 shows the user having entered the date 31 October 1998. In response to 'OK', the system removes the dialog from the screen and updates the current display based on the entered date. The exact manner of display update is based on the level of the current display and the setting of the windowing mode.

When per-level mode is on, the system responds to 'Goto Date' by changing the current display, or by creating a new display window if there is no applicable current display. Specifically:

- a. If the current display is at the month or year level, the system changes the display to the month or year that contains the selected date, and highlights the date number. For example, if the current display is at the month level when the user presses 'OK' in Figure 28, the system updates the display as shown in Figure 29.
- b. If the current display is at the week level, the system changes the display to the week containing the selected date and highlights the day name of that date. For example, if the current display is a list-style week view when the user presses 'OK' in Figure 28, the system updates the display as shown in Figure 30.
- c. If the current display is at the day level, the system simply updates the display to the selected date, with no component of the display highlighted.
- d. If the current display is at the item level or not a calendar view, the system brings to the front the most recently active display window at the day, week, month, or year level, if any. This newly fronted window is then updated per the immediately preceding rule a, b, or c, whichever is applicable.
- e. If there is no calendar view window on the screen, the system creates a new window at the standard start-up

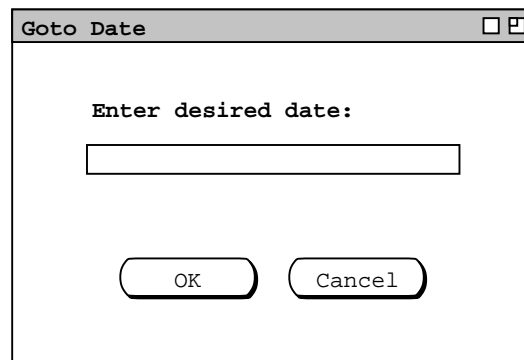


Figure 27: Goto date dialog.

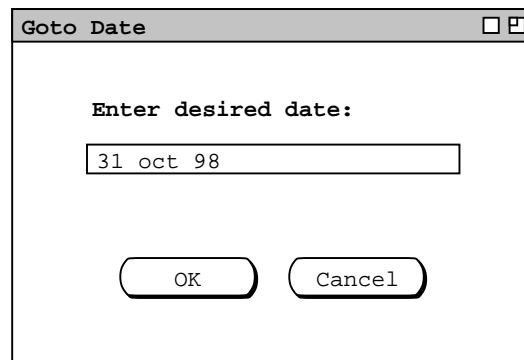


Figure 28: Goto date 31 Oct 98.

Monthly Agenda □ □						
◀ Today ▶ October 1998						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
				1 8-9 AM Ra 10:30-12 1:15-2 PM 3-5 PM So	2 9-10 AM O 10-11 AM 11 AM-12	3
4	5 1. Colloq 9-10 AM O 10-11 AM 11 AM-12	6 8-9 AM Ra 9 AM-5 PM	7 9-10 AM O 10-11 AM 11 AM-12 2:30-4:30	8 8-9 AM Ra 10:30-12 1:15-2 PM 3-5 PM So	9 9-10 AM O 10-11 AM 11 AM-12	10
11	12 1. Colloq 9-10 AM O 10-11 AM 11 AM-12	13 8-9 AM Ra 9 AM-5 PM	14 9-10 AM O 10-11 AM 11 AM-12 2:30-4:30	15 8-9 AM Ra 10:30-12 1:15-2 PM 3-5 PM So	16 9-10 AM O 10-11 AM 11 AM-12	17
18	19 1. Colloq 9-10 AM O 10-11 AM 11 AM-12	20 8-9 AM Ra 9 AM-5 PM	21 9-10 AM O 10-11 AM 11 AM-12 2:30-4:30	22 8-9 AM Ra 10:30-12 1:15-2 PM 3-5 PM So	23 9-10 AM O 10-11 AM 11 AM-12	24
25 Daylight	26 1. Colloq 9-10 AM O 10-11 AM 11 AM-12	28 8-9 AM Ra 9 AM-5 PM	27 9-10 AM O 10-11 AM 11 AM-12 2:30-4:30	29 8-9 AM Ra 10:30-12 1:15-2 PM 3-5 PM So	30 9-10 AM O 10-11 AM 11 AM-12	31 Halloween

Figure 29: Goto date 31 Oct 98 at the Month Level.

Weekly Agenda □ □						
◀ Today ▶ Week of October 25 - 31, 1998						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
25 Daylight	26 1. Colloq 9-10 AM O 10-11 AM 11 AM-12 3-5 PM So	28 8-9 AM Ra 9 AM-5 PM	27 9-10 AM O 10-11 AM 11 AM-12 2:30-4:30	29 8-9 AM Ra 10:30-12 1:15-2 PM 3-5 PM So	30 9-10 AM O 10-11 AM 11 AM-12	31 Halloween

Figure 30: Goto date 31 Oct 98 at the Week Level.

level and updates that window per the immediately preceding rule a, b, or c, whichever is applicable. The default start-up level is the month view; this default setting can be changed via the option settings commands described in Section 2.7.4.

If multi-window mode is on, the system follows these same rules, but instead of updating the current or most recently active window at a given level, it creates a new window at that level. The new window having been created, the system updates the window per rule a, b, or c, whichever is applicable.

The user may enter dates in a variety of common formats. Details of these formats are covered in Section 2.11.5. If the user enters an improperly formatted date string, the system highlights the entry, whereupon the user may retype the date or press the ‘Cancel’ button to cancel the operation entirely. Further details of this and other error conditions are covered in Section 2.12.

2.3.2.4. Upper and Lower Date Bounds

The earliest and latest possible dates in a calendar are 1 January 0000 and 31 December 9999, respectively. These are the operative boundaries of the ‘View Next’, ‘View Previous’, and ‘Goto Date’ commands at the day through year levels. At the item level, the operative boundaries of ‘View Next’ and ‘View Previous’ are the latest and earliest scheduled items, respectively.

When user issues ‘View Next’ (or one of its shortcuts) at the upper end of the operative boundary, the command has no effect. Similarly, when the user issues ‘View Previous’ (or one of its shortcuts) at the lower end of the operative boundary, the command has no effect. Specific details of how the system indicates a no-effect command are covered in Section 2.17.31.

2.3.3. Viewing Lists of Scheduled Items

The calendar views described in Section 2.3.1 are organized in standard calendar units of days through years. The user may also view scheduled items in the alternate form of ordered lists. To view a list of scheduled items, the user chooses the ‘View Lists’ menu command. In response, the system displays the submenu shown in Figure 31.

The first four commands are used to list the four different types of scheduled item. The fifth ‘All Items’ command displays a list of all four types of scheduled item. The last command displays a menu of custom views created by the user.

By default, one separate window is used for each of the six subcommands. For example, if the user executes ‘View Lists Appointments’ followed by ‘View Lists Meetings’, both an appointments list and meetings list window are displayed on the screen. Hence, there are up to six list windows, each holding the result of the most recently executed listing command. When the user re-executes a particular ‘View Lists’ command, that command’s display is changed (if necessary) and brought to the front, without a new window being created. For example, if the user executes ‘View Lists Appointments’, then adds two new appointments, and re-executes ‘View Lists Appointments’, the appointments-list display window is changed, with no new window created.

As for the view-level commands of Section 2.3.1, the user may change the default display style by enabling *two-window* or *multi-window* viewing mode. In two-window mode, the results of view commands are shown in just two windows, one for calendars and the other for lists. In multi-window mode, the results of each and every viewing



Figure 31: View lists submenu.

command are shown in a separate new window, even for lists of the same type. For example, if the user executes two successive ‘View Lists Meetings’ commands in multi-window mode, the results are shown in two separate meeting list windows. Complete details of multi-window viewing are described in Section 2.3.6.

Multi-window lists are particularly useful when applied in conjunction with filtered viewing. For example, the user can display a list of two different categories of appointments in two separate side-by-side windows. Complete details of filtered viewing are presented in Section 2.3.4.

2.3.3.1. Appointment Lists

Figure 32 shows the response to the user’s selection of the ‘View Lists Appointments’ command.

The figure shows a list of appointments created by the user in the scenarios presented thus far. By default, the list is sorted primarily by date, secondarily by time, thirdly by duration, and fourthly by title. The default date range of the list is the three-week period starting one week prior to today’s date. The default physical height of the list display window is twenty items. These default settings can be changed using the options commands described in Section 2.7.4.1.

For the ongoing scenario, the current date is September 23. This means that the three-week period covers the dates September 16 through October 6, inclusive. Since there are more than twenty appointments scheduled in this period, the list window can be scrolled or resized to view the additional items.

The items in an appointments list are a somewhat abbreviated version of the information for a scheduled appointment. The precise extent of the abbreviations is as follows:

- A list item has only a start date, with no end date. This is because there is a separate list item for every instance of a recurring appointment.
- A list item has only a yes/no indication of whether the appointment recurs, not the full recurring information.
- A list item contains no remind information or details.

The user can change the default order of the list by clicking on any of the column labels that appear in the display. For example, if the user clicks on the **Title** column label in Figure 32 the system updates the display to that shown in Figure 33, where the list items are sorted alphabetically by title. The precise sorting rules for each of columns are as follows:

Appointments, sorted by Date									<input type="checkbox"/>	<input type="checkbox"/>
Title	Date	Time	Duration	Recurs?	Category	Location	Security	Priority		
Racket Ball	Sep 17, 1998	8:00 AM	1 hr	yes	personal	Rec Cente	title	optional		▲
Office Hours	Sep 21, 1998	9:00 AM	1 hr	yes	classes	14-210	public	must		
Data Structure	Sep 21, 1998	10:00 AM	1 hr	yes	classes	14-256	public	must		
Data Structure	Sep 21, 1998	11:00 AM	1 hr	yes	classes	14-232	public	must		
Software Engin	Sep 21, 1998	3:00 PM	2 hrs	yes	classes	14-301	public	must		
Racket Ball	Sep 22, 1998	8:00 AM	1 hr	yes	personal	Rec cente	title	optional		
Research	Sep 22, 1998	9:00 AM	8 hrs	yes	research	14-210	public	must		
Office Hours	Sep 23, 1998	9:00 AM	1 hr	yes	classes	14-210	public	must		
Data Structure	Sep 23, 1998	10:00 AM	1 hr	yes	classes	14-256	public	must		
Data Structure	Sep 23, 1998	11:00 AM	1 hr	yes	classes	14-232	public	must		
Lunch with Mic	Sep 23, 1998	12:00 PM	1 hr 30	no	departmen	Staff din	public	optional		
Racket Ball	Sep 24, 1998	8:00 AM	1 hr	yes	personal	Rec cente	title	optional		
Office Hours	Sep 24, 1998	1:15 PM	1 hr 45	yes	classes	14-210	public	must		
Software Engin	Sep 24, 1998	3:00 PM	2 hrs	yes	classes	14-301	public	must		
Dentist	Sep 25, 1998	8:00 AM	1 hr 30	no	personal	1342 Syca	title	must		
Office Hours	Sep 25, 1998	9:00 AM	1 hr	yes	classes	14-210	public	must		
Data Structure	Sep 25, 1998	10:00 AM	1 hr	yes	classes	14-256	public	must		
Data Structure	Sep 25, 1998	11:00 AM	1 hr	yes	classes	14-232	public	must		
Office Hours	Sep 28, 1998	9:00 AM	1 hr	yes	classes	14-210	public	must		
Data Structure	Sep 28, 1998	10:00 AM	1 hr	yes	classes	14-256	public	must		▼

Figure 32: Basic Appointments List.

Appointments, sorted by Date								
Title	Date	Time	Duration	Recurs?	Category	Location	Security	Priority
CAD Research P	Sep 9, 1998	2:30 PM	2 hr	yes	research	Rec Cente	title	optional
College Meetin	Sep 9, 1998	2:00 PM	1 hr	no	campus	14-210	public	must
Data Structure	Sep 7, 1998	11:00 AM	1 hr	yes	classes	14-256	public	must
Data Structure	Sep 9, 1998	11:00 AM	1 hr	yes	classes	14=232	public	must
Data Structure	Sep 11, 1998	11:00 AM	1 hr	yes	classes	14-301	public	must
Data Structure	Sep 14, 1998	11:00 AM	1 hr	yes	classes	Rec cente	title	optional
Data Structure	Sep 16, 1998	11:00 AM	1 hr	yes	classes	14=210	public	must
Data Structure	Sep 18, 1998	11:00 AM	1 hr	yes	classes	14-238	public	must
Data Structure	Sep 21, 1998	11:00 AM	1 hr	yes	classes	14=238b	public	optional
Data Structure	Sep 7, 1998	10:00 AM	1 hr	yes	classes	14-210	public	must
Data Structure	Sep 9, 1998	10:00 AM	1 hr	yes	classes	14-256	public	must
Data Structure	Sep 11, 1998	10:00 AM	1 hr	yes	classes	14-232	public	must
Data Structure	Sep 14, 1998	10:00 AM	1 hr	yes	classes	Staff din	public	optional
Data Structure	Sep 16, 1998	10:00 AM	1 hr	yes	classes	3-206	public	optional
Data Structure	Sep 18, 1998	10:00 AM	1 hr	yes	classes	20-118	public	optional
Data Structure	Sep 21, 1998	10:00 AM	1 hr	yes	departmen	14-301	public	must
Data Structure	Sep 9, 1998	12:00 PM	1 hr 30	no	classes	Rec cente	title	optional
Office Hours	Sep 7, 1998	9:00 AM	1 hr	yes	classes	14-238	public	must
Office Hours	Sep 9, 1998	9:00 AM	1 hr	yes	classes	21-139	public	optional
Office Hours	Sep 10, 1998	1:15 PM	1 hr 45	yes	classes	14-210	public	must

Figure 33: Appointments list sorted by title.

- The Title column is sorted in ascending lexical order of title string, ignoring differences in the case of letter characters. Lexical ordering is defined precisely as the case-insensitive characterwise comparison of strings based on the ASCII collating sequence, or comparable collating sequence in standard use in a particular operating environment.
- Date is sorted in ascending chronological order.
- Time is sorted in ascending time-of-day order.
- Duration is sorted in ascending order from shortest to longest duration.
- Recurs is sorted with all 'yes' values before all 'no' values.
- Category is sorted lexically in the same manner as Title. Items with an empty category value appear after all items with non-empty category values.
- Location is sorted in the same manner as Category.
- Security is sorted in the order public, title, protected, private.
- Priority is sorted with must values before optional values.

Since no single field value in an appointment is necessarily unique, the primary sorting order must be augmented with secondary, tertiary, quaternary and in some cases quintinary orders. Table 1 specifies the precise subordering for each of the column labels used as the primary sorting key.

Figure 33 shows how the banner of a list window is altered to indicate that a sorting order other than the default is in use. The format of the banner for all list displays is the following:

type-of-list, sorted by field-name

where *type-of-list* is one of **Appointments, Meetings, Tasks, Events**, or **All Items**; *field-name* is the name of item field used as the primary sorting key.

2.3.3.2. Meeting Lists

To view a list of scheduled meetings, the user selects the 'View Lists Meetings' menu command. Figure 34 shows the result of the user so doing. The list contains eleven entries, which reflects that fact that eleven meetings are scheduled for the default listing period of September 16 through October 6. The format of meetings lists is the same as appointments, with the addition of a 'Sched By' column. This new column contains the ID of the Calendar Tool user who scheduled the meeting.

2.3.3.5. All Items Lists

The final listing form contains all four types of scheduled item together in one list. To select this form, the user executes the 'View Lists All Items' command, in response to which the system displays a window of the form shown in Figure 37.

By default, the list is sorted primarily by date. Among items of the same type, the precise sorting rules for each list column are as stated above in Sections 2.3.3.1, 2.3.3.2, 2.3.3.3, and 2.3.3.4 above.

Table 4 gives the subordering for all possible primary keys, of which there are eleven in the all-items list. The 'Time' field for appointments and meetings represents the start time of the item. The 'Time' for tasks represents the time the task is due. The single 'Time' column in the all items list is used for both the start time and due time values, depending on the type of item listed in a particular row. The sorting rules for the 'Time' column are defined so that the order among appointments, meetings, and tasks is consistent with the order among these items in calendar views.

The following assumptions are made for sorting lists with different types of items, in which one or more of the sub-order keys are missing:

All Items, sorted by Date										
Title	Date	Time	Duration	Recurs?	Category	Location	Security	Priority	Sched By	Completed
Send class mat	Sep 16, 1998	8:00 AM		no	classes		public	10		Sep 17, 1998
Racket Ball	Sep 17, 1998	8:00 AM	1 hr	yes	personal	rec centr	title	optional		
Colloquium ann	Sep 21, 1998	9:00 AM		yes	departmen		public	8		Sep 21, 1998
Office Hours	Sep 21, 1998	9:00 AM	1 hr	yes	classes	14-210	public	must		
Data Structure	Sep 21, 1998	10:00 AM	1 hr	yes	classes	14-256	public	must		
Data Structure	Sep 21, 1998	11:00 AM	1 hr	yes	classes	14-232	public	must		
Faculty Meetin	Sep 21, 1998	1:00 PM	1 hr	yes	departmen	14-256	public	must	hlandis	
Software Engin	Sep 21, 1998	3:00 PM	2 hrs	yes	classes	14-301	public	must		
Autumnal Equin	Sep 22, 1998				holiday		public			
Racket Ball	Sep 22, 1998	8:00 AM	1 hr	yes	personal	rec centr	title	optional		
Research	Sep 22, 1998	9:00 AM	8 hrs	yes	research	14-210	public	must		
Staff Meeting	Sep 23, 1998	8:00 AM	1 hr	no	departmen	14-238	public	must	estier	
TA Meeting	Sep 23, 1998	8:00 AM	1 hr 30	no	departmen	14-238b	public	optional	estier	
Office Hours	Sep 23, 1998	9:00 AM	1 hr	yes	classes	14-210	public	must		
Data Structure	Sep 23, 1998	10:00 AM	1 hr	yes	classes	14-256	public	must		
Data Structure	Sep 23, 1998	11:00 AM	1 hr	yes	classes	14-232	public	must		
Lunch with Mic	Sep 23, 1998	12:00 PM	1 hr 30	no	departmen	staff din	public	optional		
College Meetin	Sep 23, 1998	2:00 PM	1 hr	no	campus	3-206	public	must	wsmith	
Special Colloq	Sep 23, 1998	2:00 PM	1 hr 30	no	campus	20-118	public	optional	hpaulsen	
CAD Research P	Sep 23, 1998	2:30 PM	2 hrs	yes	research	14-301	public	must		

Figure 37: All items list.

Primary	Secondary	Tertiary	Quaternary	Quintinary
Title	Date	Time	Duration	
Date	Time	Duration	Title	
Time	Date	Duration	Title	
Duration	Date	Time	Title	
Recurs	Date	Time	Duration	Title
Category	Date	Time	Duration	Title
Location	Date	Time	Duration	Title
Security	Date	Time	Duration	Title
Priority	Date	Time	Duration	Title
Sched By	Date	Time	Duration	Title
Completed	Date	Time	Duration	Title

Table 4: Sorting suborders for all-item lists.

- a. The 'Time' value for any event is considered to be earlier than that of any appointment, meeting, or task; this means that for any given day, events precede tasks, appointments, and meetings when 'Time' is used as the sort key.
- b. For the purposes of sorting an all-items list, the 'Time' value for any task is considered to be later than that of any event and earlier than that of any appointment or meeting; this means that tasks follow events and precede appointments and meetings when 'Time' is used as the sort key. This is the case even if a task due time is after the start time of some appointments or meetings of the same day. This sorting rule is defined as it is so that the relative list order of tasks compared to appointments and meetings is the same as it is in the day, week, and month calendar views, wherein all tasks precede all appointments and meetings for any given day.
- c. The 'Duration' of any event is considered to be shorter than that of any appointment meeting, or task; this means that events precede all other types of item when 'Duration' is the sort key.
- d. The 'Duration' of any task is considered to be longer than that of any event, shorter than any appointment or meeting; this means that tasks follow events and precede appointments and meetings when 'Duration' is used as the sort key.
- e. The 'Recurs' value for events is considered to be 'no'; this means that events follow all other types of item when 'Recurs' is used as the sort key.
- f. The 'Location' value for events is considered to be empty; this means that events follow all other types of item when 'Location' is used as the sort key.
- g. The 'Priority' value for events is considered to be higher than any appointment, meeting, or task; this means that events precede all other types of item when 'Priority' is used as the sort key.
- h. The 'Sched By' value for appointments, tasks, and events is considered to be greater than that for any meeting; this means that meetings precede all other types of item when 'Sched By' is used as the sort key.
- i. The 'Completed' value for appointments, meetings, and events is considered to be greater than any task; this means that tasks (whether completed or not) precede all other types of item when 'Completed' is used as the sort key.

2.3.3.6. Custom Lists

The custom lists feature of the Calendar Tool allows the user to define specialized forms of lists. In particular, the user can specify which types of items appear in a list, which item fields appear as list columns, and what range of dates the list covers. For example, the user can define a list that shows only the title and date of appointments and meetings for the next six months.

By default, there are no pre-defined custom lists. Hence, when the user initially selects the 'View Lists Custom' menu command, the system displays the submenu as shown in Figure 38. The blank space above the 'Edit ...' command is where the names of custom lists appear once they are defined by the user.

To create a new custom list, the user selects the 'Edit ...' command, whereupon the system displays the custom list dialog shown in Figure 39. The 'List Name' text box contains the name of the list being defined. The initial default name is "New List". The user enters a new list name by typing in the text box. The name for each custom list must be unique among all other defined lists.

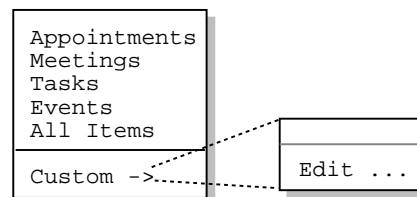


Figure 38: Initial custom lists menu.

Figure 39: Custom list dialog.

The check boxes under the heading ‘Show these items’ specify which types of item appear in a custom list. At least one of the boxes must be checked. The check boxes under the heading ‘Show these fields’ specify which item fields appear as column headings in the list. At least one of the boxes must be checked. The ‘Primary sort key’ is a menu containing each of the fields listed under ‘Show these fields’. The default primary sort key for custom lists is the ‘Date’ field.

The ‘Start Date’ and ‘End Date’ data entry fields specify the starting and ending dates of the custom list. As shown in the figure, the default start date is 1 week ago, the default end date 2 weeks from now. The small data-entry box immediately to the right of the label is a typeable numeric field in which the user can enter a value between 1 and 99. Immediately to the right of the numeric field is a menu-augmented text box containing the date range information. Immediately below the range information is a two-element menu containing the selections ‘ago’ and ‘from now’. Figure 40 shows an expansion of both the pulldown menus. If ‘any’, ‘today’, or ‘specific date’ is selected, the numeric text field is disabled, since it does not apply in these cases. Also in these cases, the date direction menu is disabled (physically greyed out). When ‘specific date’ is selected, the user types a specific date value into the text field. When any of the other menu entries is selected, typing is disabled in the text field.

Figure 41 shows the result of the user having defined a new list named “Year’s events”. The list applies only to events, leaves out the security column, and extends over the date range of six months before and after today. The figure illustrates how the check boxes for some fields are selectively disabled when a subset of item check boxes are on. The field check boxes that are disabled correspond to those fields that do not apply to any of the selected types of item. Table 5 defines specifically which fields apply to which types of item. A check box under the ‘Show these fields’ heading is enabled (physically solid type) when one or more of the types to which it applies is checked under the ‘Show these items’ heading. The state of the check boxes is reflected in the ‘Primary sort key’ pulldown menu. Specifically, the active menu items correspond to the check-box selected fields. For

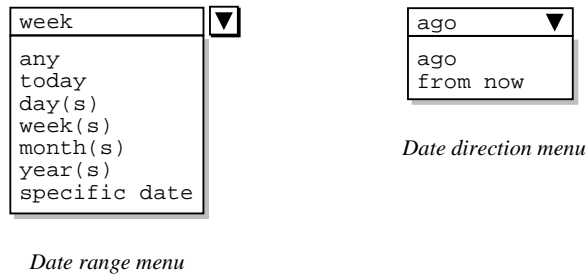


Figure 40: Custom list date menus.

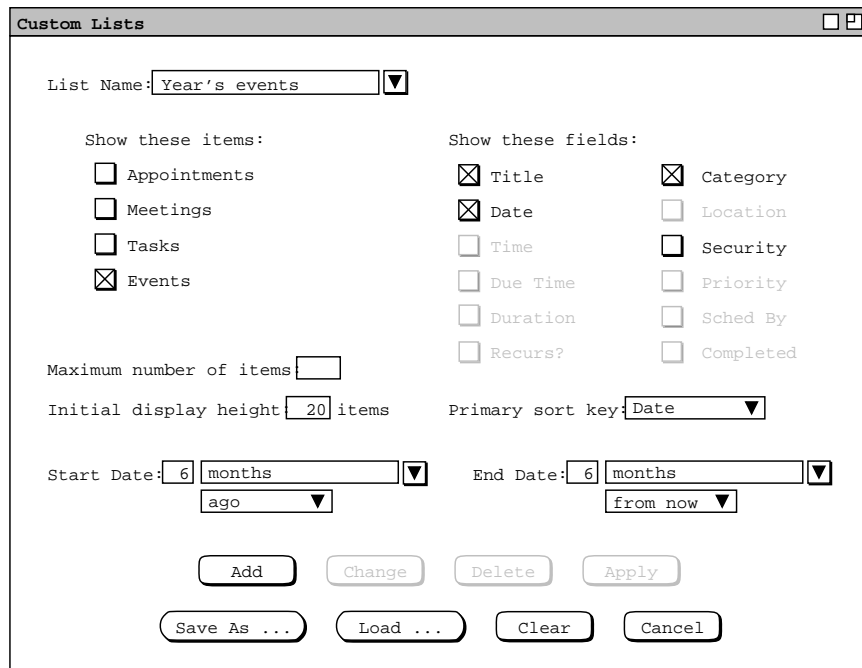


Figure 41: Year's events custom list.

Data field	Applicable to these types of item
Title	all
Date	all
Time	appointments, meetings
Due Time	tasks
Duration	appointments, meetings
Recurs?	appointments, meetings, tasks
Category	all
Location	appointments, meetings, events
Security	all
Priority	appointments, meetings, tasks (‘must’ or ‘optional’ for appointments and meetings; number between 0 and 10 for tasks)
Sched By	meetings
Completed	tasks

Table 5: Fields applicable to each type of scheduled item.

example, Figure 42 shows the state of the ‘Primary sort key’ menu in the context of Figure 41. The enabled menu items correspond to the three selected check boxes. The disabled items in the menu cannot be selected as the primary sort key. When ‘Date’ is not selected in the check boxes, the default primary sort key is chosen as the first checked key in top-to-bottom order of check boxes in the dialog (which is the order shown in Table 4).

Sorting suborder for custom lists is based on Table 4 with the following additional provisions. When a field that would otherwise be used as a suborder key is missing from the custom list, the next field in the subordering takes the place of the missing key. When one or more of the subordering keys in Table 4 is missing, additional subordering is defined by the following order among the remaining keys: ‘Due Time’, ‘Recurs’, ‘Category’, ‘Location’, ‘Security’, ‘Priority’, ‘Sched By’, and ‘Completed’. For example, if the user defines a custom list that contains only ‘Location’, ‘Priority’, and ‘Sched By’, those keys are used as the primary, secondary, and tertiary sorting keys, respectively.

To complete the definition of a new list, the user presses the ‘Add’ button at the bottom of the custom lists dialog. In response, the system adds the list to those that are available for use. To define additional new lists, the user types the new names into the ‘List Name’ text box, selects the desired settings, and presses the ‘Add’ button for each

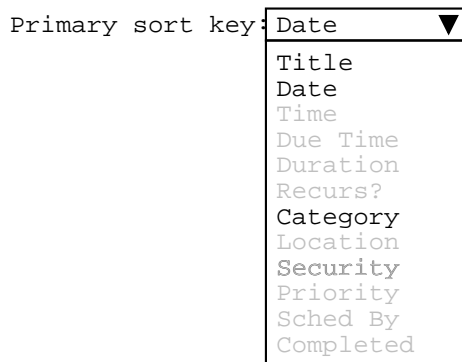


Figure 42: Primary sort key menu in Year’s events custom list.

new list to be defined.

The names of available lists appear in two contexts:

- a. the 'View Lists Custom' submenu, accessible from the main command menubar;
- b. the 'List Name' menu, in the custom lists dialog.

These contexts are illustrated in Figure 43. The figure reflects the user having defined two additional custom lists named "Appt titles only" and "Mtgs next 2 months". The list names are sorted alphabetically in the menus. In the custom list dialog, the name in the 'List Name' text box is that most recently selected or typed by the user.

Figure 44 shows the result of the user selecting the 'Year's events' list in the 'View Lists Custom' submenu. The banner of a custom list display contains the name of the custom list.

In the context of the scenarios thus far, "Jim's Birthday" is the only user-scheduled event. The other events in the 'Year's Events' list are all pre-defined holidays, as discussed further in Section 2.11.16.

The user can change or delete defined lists in the custom list dialog. To change an existing list, the user selects the desired list in the 'List Name' menu, or types its name in the text box and presses the Enter key. The user then performs the desired changes and presses the Change button. For example, Figure 45 shows the user having edited the previously defined 'Year's events' list. The edits involve changing the date range to specific dates instead of the six month range before and after today's date. To perform these edits, the user selects 'specific date' in

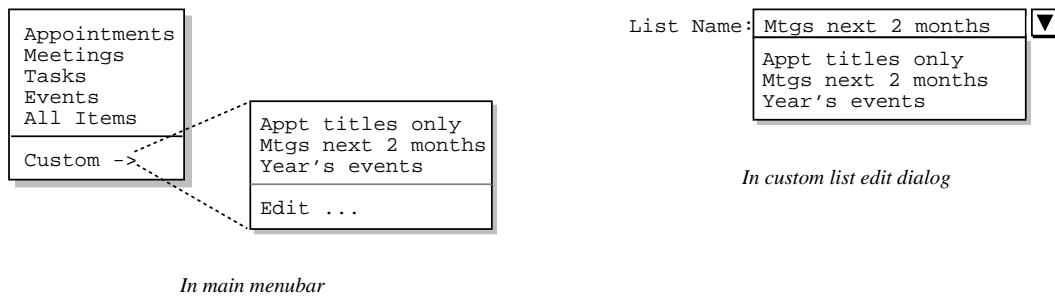


Figure 43: Menu contexts in which list names appear.

Year's events, sorted by Date		
Title	Date	Category
April Fool's D	Apr 1, 1998	holiday
Daylight Savin	Apr 5, 1998	holiday
Easter Sunday	Apr 12, 1998	holiday
Mother's Day	May 10, 1998	holiday
Memorial Day	May 25, 1998	holiday
Flag Day	Jun 14, 1998	holiday
Summer Solstic	Jun 21, 1998	holiday
Father's Day	Jun 21, 1998	holiday
Independence D	Jul 4, 1998	holiday
Labor Day	Sep 7, 1998	holiday
Autumnal Equin	Sep 22, 1998	holiday
Jim's Birthday	Sep 23, 1998	special e
Daylight Savin	Oct 25, 1998	holiday
Halloween	Oct 31, 1998	holiday
Veteran's Day	Nov 11, 1998	holiday
Thanksgiving	Nov 26, 1998	holiday
Hanukkah	Dec 14, 1998	holiday
Ramadan Begins	Dec 20, 1998	holiday
Winter Solstic	Dec 21, 1998	holiday
Christmas	Dec 25, 1998	holiday

Figure 44: Year's events list display.

Custom Lists

List Name: Year's events ▼

Show these items:

- Appointments
- Meetings
- Tasks
- Events

Show these fields:

- Title
- Date
- Time
- Due Time
- Duration
- Recurs?
- Category
- Location
- Security
- Priority
- Sched By
- Completed

Maximum number of items:

Initial display height: items

Primary sort key: ▼

Start Date: ▼ ago ▼

End Date: ▼ from now ▼

Buttons: Add, Change, Delete, Apply, Save As ..., Load ..., Clear, Cancel

Figure 45: Changed years-events list.

both the 'Start Date' and 'End Date' menus, and then types the desired date range of January 1, 1999 to December 31, 1999. To confirm these changes, the user presses the 'Change' button at the bottom of the dialog.

To delete an existing list, the user selects its name from the 'List Name' menu and presses the 'Delete' button. In response, the system removes the list from those that are defined, removes it from the 'List Name' menu, and clears the 'List Name' text box. The 'Apply' button is used to apply the currently selected list, as if its name had been chosen from 'View Lists Custom' submenu.

The 'Save As ...' and 'Load ...' buttons allow the user to save and load copies of custom list definitions to and from files. Details of saving and loading auxiliary data files are covered in Section 2.8.6

To clear all entered information in the custom list dialog, the user presses the 'Clear' button. In response, the system restores the dialog to the state it was in when it was most recently displayed on the screen. The 'Cancel' button removes the custom lists dialog from the screen. Only the explicit selection of 'Cancel' (or an environment-specific window closing button) removes the dialog from the screen. Execution of any of the other commands ('Add' through 'Clear') leaves the dialog on the screen.

Custom list definitions are calendar-specific. That is, each calendar has its own set of custom list definitions. When more than one calendar is open, the custom definitions that appear in menus, displays, and dialogs are those of the current calendar. The specific contexts in which custom list definitions appear are the following:

- a. the banners of custom list displays
- b. the 'View Lists Custom' command menu
- c. custom list editing dialogs

The system provides a separate custom list editing dialog for each calendar. Whenever the user executes the 'View Lists Custom Edit ...' command, the system (re)displays a custom list dialog for the current calendar.

Each calendar uses a single custom list dialog. This means that up to N custom list dialogs can be displayed at any time, for N equal to the number of open calendars. When more than one calendar is open, the calendar name appears by default in the banner of each custom list dialog. The appearance or non-appearance of the calendar name in window banners is controlled by an option setting, as described in Section 2.7.4.2.

2.3.4. Filtered Viewing

When the user selects the ‘Filter ->’ item in the View menu, the system displays the submenu shown in Figure 46. The first four commands in the menu allow the user to filter in or out the four types of scheduled item. The menu commands act as show/hide toggles. When the ‘Hide ...’ form is visible in the menu, selecting it hides the selected type of scheduled item and toggles the menu label to ‘Show ...’. When the ‘Show ...’ form of the command is selected, items that were hidden are made visible again, and the menu is toggled back to the ‘Hide ...’ form.

Consider for example the month view shown earlier in Figure 22. If the user selects ‘Filter Hide Appointments’ in the context of that view, the system updates the display as shown in Figure 47. If the user then selects

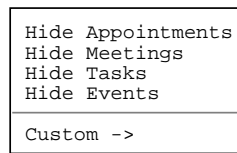


Figure 46: Filter submenu.

Monthly Agenda, hide A						
September 1998						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1 11 AM-2 P	2	3	4	5
6	7 Labor Day	8 11 AM-2 P	9	10	11	12
13	14	15 11 AM-2 P	16 1. Send c	17	18	19
20	21 1. Colloq 1-2PM Fa	22 Autumnal	23 8-9 AM St 8-9:30 AM 2-3 PM Co 2-3:30 PM	24 Jim's Bir 1. Prepar 2. Buy so 9-10:30 A	25	26
28	27 1. Colloq	29	30 2:30-4:30			

Figure 47: Month view with appointments filtered out.

'Hide Meetings', the system updates the display to that shown in Figure 48. Since most of the items scheduled for the month of September were appointments and meetings, only a few items remain in the display.

Having hidden appointments and meetings, a subsequent invocation of the 'Filter' menu appears as shown in Figure 49. If the user selects 'Show Appointments' and 'Show Meetings', the month display is restored to the original state shown in Figure 22.

Showing and hiding tasks and events is analogous to meetings and appointments. At the day level and weekly table level, the separate areas of the display are removed when tasks and/or events are hidden. For example, Figures 50 and 51 show the results of hiding both events and tasks in the context of the daily and weekly views shown originally in Figures 15 and 19.

Any or all of the commands can be in either the show or hide state. When all four commands are in the show state, all scheduled items are visible. When all four commands are in the hide state, no scheduled items are visible.

Monthly Agenda, hide A,M						
◀ Today ▶		September 1998				
Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5
6	7 Labor Day	8	9	10	11	12
13	14	15	16 1. Send c	17	18	19
20	21 1. Colloq	22 Autumnal	23	24 Jim's Bir 1. Prepar 2. Buy so	25	26
28	27 1. Colloq	29	30			

Figure 48: Month view with meetings filtered out.

Show Appointments
Show Meetings
Hide Tasks
Hide Events
Custom ->

Figure 49: Filter menu in 'Show Appointments', 'Show Meetings' state.

Daily Agenda, hide T,E	
◀ Today ▶ Thursday, September 24, 1998	
8AM	Racket Ball
9AM	Picnic Day Committee
10AM	Software Engineering Colloquium
11AM	
12PM	
1PM	Office Hours
2PM	
3PM	Software Engineering Graduate Seminar
4PM	
5PM	

Figure 50: Events and tasks filtered out of a day view.

Weekly Agenda, hide T,E							
◀ Today ▶ Week of September 20-26, 1998							
	Sun	Mon	Tue	Wed	Thu	Fri	Sat
8 AM			Racket ba	Staff M ▶	Racket ba	Dentist ▶	
9 AM		Office ho	Research	Office ho	Picnic da		
10 AM		Data Stru		Data Stru	Software	Data Stru	
11 AM		Data Stru		Data Stru		Data Stru	
12 PM				Lunch wit			
1 PM					Office ho		
2 PM				College ▶			
3 PM		Software			Software		
4 PM							
5 PM							

Figure 51: Events and tasks filtered out of a table-style week view.

Hmm, I think we need to be clearer up front about the fact fi ltering applies to the item level differently than to the day, week, and month levels. Viz., item-level fi ltering determines whether an entire item-level display is visible, but does not fi lter the guts of any item-level display.

The four show/hide fi ltering commands apply only to calendar views at the item, day, week, and month levels. Show/hide does not apply to year-level views since there are no items visible at that level. Show/hide fi ltering also does not apply to list views, since lists are already specialized to each kind of item. The custom fi lters described in the next section apply to calendar views as well as to lists. Operational details of show/hide and custom fi ltering used in conjunction are covered in Section 2.3.4.3.

2.3.4.1. Custom Filter Definition

The custom fi ltering feature of the Calendar Tool allows the user to defi ne fi ner-grain fi lters than simply all items of a particular type. For example, the user can choose to fi lter out all but a particular category of item that is scheduled between some particular dates.

By default, there are no custom fi lters defi ned. Hence, when the user initially selects the ‘View Filters Custom’ menu item, the system displays the submenu as shown in Figure 52. The blank space above the ‘Edit ...’ command is where the names of custom fi lters appear once they are defi ned by the user.

To create a new custom fi lter, the user selects the ‘Edit ...’ command, whereupon the system displays the fi lter defi nition dialog shown in Figure 53.

The ‘Filter Name’ text box contains the name of the fi lter being defi ned. The initial default name is "New Filter". The user enters a chosen fi lter name by typing in the text box. The name for each defi ned fi lter must be unique among all other fi lters.

The editing fi elds below the fi lter name allow the user to defi ne detailed fi ltering criteria for the display of items in calendars and lists. The criteria apply to all types of items that are currently visible, i.e., items for which the show state is on in the show/hide fi lter commands. The criteria apply uniformly to all calendar and list displays.

The fi elds in the fi lter defi nition constitute the union of components from all four types of scheduled item. Given this, each fi eld is not necessarily applicable to all types of scheduled item. Table 6 defi nes specifcally which fi elds apply to which item types. (This table is a superset of Table 5 defi ned for custom lists.)

There are a number of pulldown menus associated with the fi elds of the fi lter defi nition. Figure 54 shows an the expansion of these menus. Descriptions of each fi ltering fi eld follow. Section 2.11.13 defi nes the precise pattern matching rules that apply to fi ltering

2.3.4.1.1. Title

The ‘Title’ fi eld in the fi lter defi nition defi nes which items are visible based on the value of their titles. Immediately to the right of the ‘Title’ label is a text box containing a pattern string for item titles. The pattern is used in conjunction with the selection in the pulldown menu below the ‘Title’ label. Specifcally:

- a. When ‘any’ is selected in the menu, items with any title are visible. Typing in the ‘Title’ text box is disabled when ‘any’ is selected.

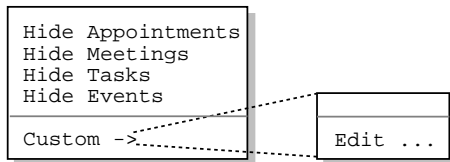


Figure 52: Initial custom fi lters menu.

Custom Filters □ □

Filter Name: ▼

Title:
 any ▼

Earliest Date: today ▼ ago ▼

Latest Date: 5 weeks ▼ from now ▼

Earliest Time: 8 AM Latest Time: 5 PM

Duration: hr min expression
 any ▼

Recurring? yes no either

Category:
 any ▼

Location:
 any ▼

Security:
 any ▼

Priority: must optional either

Task Priority: expression
 any ▼

Completed? yes no either

Earliest Completion Date: 1 week ▼ ago ▼

Latest Completion Date: today ▼ from now ▼

Remind? yes no either

Details:
 any ▼

Attendees:
 any ▼

Figure 53: Filter definition dialog.

Data field	Applicable to these types of item
Title	all
Earliest Date	all
Latest Date	all
Earliest Time	appointments, meetings, tasks
Latest Time	appointments, meetings, tasks
Duration	appointments, meetings
Recurring?	appointments, meetings, tasks
Category	all
Location	appointments, meetings, events
Security	all
Priority	appointments, meetings
Task Priority	tasks
Completed	tasks
Earliest Completion Date	tasks
Latest Completion Date	tasks
Remind	appointments, meetings, tasks
Details	appointments, meetings, tasks
Attendees	meetings

Table 6: Filter fields applicable to each type of scheduled item.

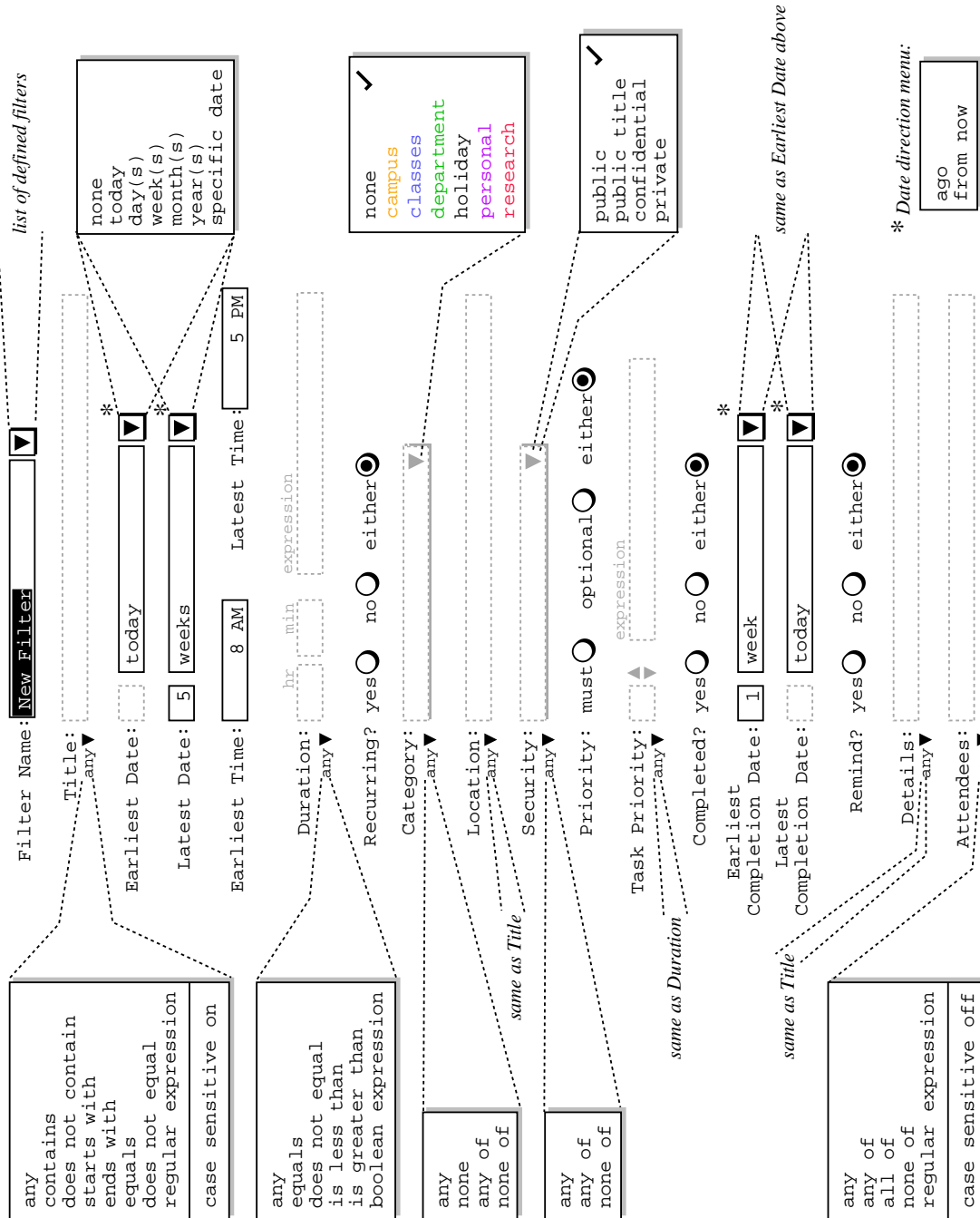


Figure 54: Expansion of filter definition menus.

- When 'contains' is selected, items are visible when they have a title that contains the pattern string.
- When 'does not contain' is selected, items are visible when they have a title that does not contain the pattern string.
- When 'starts with' is selected, items are visible when they have a title that starts with the pattern string.
- When 'ends with' is selected, items are visible when they have a title that ends with the pattern string.
- When 'equals' is selected, items are visible when they have a title that exactly equals the pattern string.
- When 'does not equal' is selected, items are visible when they have a title that does not equal the

pattern string.

- h. When 'regular expression' is selected, items are visible when they match a regular expression pattern. The pattern is typed into the text box. Details of regular expression syntax are covered in Section 2.11.14.

The last item in the 'Title' field menu specifies the case sensitivity of the pattern string matching. The menu item is an on/off toggle. By default, case sensitive matching is on for 'Title' matching.

2.3.4.1.2. Earliest and Latest Date

The 'Earliest Date' and 'Latest Date' fields define the earliest and latest scheduled dates for which items are visible. The small text box immediately to the right of the field label is used for a numeric value in the range 1 through 99. To the right of the numeric box is a combination text box and pulldown menu that is used to define the date range. Specifically:

- a. When 'none' is selected, items are visible all the way to the system-enforced earliest and latest dates, which are, respectively, 1 January 0000 and 31 December 9999. Typing in the numeric box is disabled when 'none' is selected.
- b. When 'today' is selected, the earliest (latest) date for visible items is today's date. Typing in the numeric box is disabled when 'today' is selected.
- c. The next four menu selections define how far before or after today's date items are visible, in terms of days, weeks, months, or years. These items are defined in conjunction with a specific numeric value, the default being 1.
- d. When 'specific date' is selected, items are visible up to and including a specific date before or after today's date. The date is entered by typing in the text box. Typing in the numeric box is disabled when 'specific date' is selected.

At the far right of the date entry fields is a two-element menu containing the selections 'ago' and 'from now'. These selections specify whether the selected date pattern comes before or after today's date.

2.3.4.1.3. Earliest and Latest Time

The 'Earliest Time' and 'Latest Time' fields define the earliest and latest scheduled times for which items are visible. For appointments and meetings, the scheduled time is the start time; for tasks, the scheduled time is the due time. Both fields contain time values. The default values are 8 AM for earliest and 5 PM for latest. If the 'Earliest Time' field is empty, it is assumed to be 12:00 AM; if 'Latest Time' is empty, it is assumed to be 11:59 PM. If both fields are non-empty, the earliest time must be less than or equal to the latest time.

2.3.4.1.4. Duration

The 'Duration' field defines item visibility based on duration. A duration value is entered in hours and minutes in the two text edit boxes. If the user enters no duration value, the default is 1 hour and no minutes. The duration value is used in conjunction with a selection in the pulldown menu below the 'Duration' label. Specifically:

- a. When 'any' is selected, items with any duration are visible. Typing in the hour and minute text boxes is disabled when 'any' is selected.
- b. When 'equals' is selected, items are visible when they have a duration that exactly equals the specified hour/minute value.
- c. When 'not equals' is selected, items are visible when they have a duration that does not equal the specified hour/minute value.
- d. When 'is less than' is selected, items are visible when they have a duration that is less than the specified hour/minute value.
- e. When 'is greater than' is selected, items are visible when they have a duration that is greater than the specified hour/minute value.
- f. When 'boolean expression' is selected, items are visible when they equal a boolean expression. The expression is typed in the text box labeled 'expression' immediately to the right of the minute-value text box. Details of boolean expression syntax are covered in Section 2.11.15.

2.3.4.1.5. Recurring?

The 'Recurring?' field defines visibility based on whether a scheduled item is recurring. Specifically:

- a. If `yes` is selected, items are visible if they are recurring.
- b. If `no` is selected, items are visible if they are not recurring.
- c. If `either` is selected, items are visible whether or not they are recurring.

2.3.4.1.6. Category

The 'Category' field defines visibility based on the category of scheduled items. The pulldown menu to the right of the Category label allows the user to select one or more category values. A selected category is indicated with a check mark to the right of its name. The category-value selection is used in conjunction with the selection in the pulldown menu below the Category label. Specifically:

- a. When 'any' is selected, items with any category, including none, are visible. All items in the categories menu are unchecked when 'any' is selected.
- b. When 'none' is selected, only items with no category are visible. All items in the categories menu are unchecked when 'none' is selected.
- c. When 'any of' is selected, scheduled items are visible if they have a category value that is any of the values selected in the menu.
- d. When 'none of' is selected, scheduled items are visible if they have a category value that is not any of the values selected in the menu.

If exactly one category is checked in the right-side menu, then its color-coded name appears in the text area. If two or more categories are checked, then the name of the first category appears in the text area, suffixed with the string "+ ...".

2.3.4.1.7. Location

The 'Location' field is the same form of text box and menu combination as the 'Title' field. The entered pattern defines which items are visible based on the value of their locations.

2.3.4.1.8. Security

The 'Security' field defines visibility based on the security of scheduled items. The pulldown menu to the right of the field label allows the user to select one or more of the four security values. The security-value selection is used in conjunction with the selection in the pulldown menu below the label. Specifically:

- a. When 'any' is selected, items with any security are visible. All items in the security menu are unchecked when 'any' is selected.
- b. When 'any of' is selected, scheduled items are visible if they have a security value that is any of the values selected in the menu.
- c. When 'none of' is selected, scheduled items are visible if they have a security value that is not any of the values selected in the menu.

2.3.4.1.9. Priority

The 'Priority' field defines visibility based on the priority of scheduled items. Specifically:

- a. If 'must' is selected, items are visible if their priority is 'must'.
- b. If 'optional' is selected, items are visible if their priority is 'optional'.
- c. If 'either' is selected, items are visible with either priority.

2.3.4.1.10. Task Priority

The 'Task Priority' field defines the visibility of tasks only, based on their numeric priority. The small text box immediately to the right of the field label is used for a numeric value in the range 0 through 10. To the right of

the text box are up- and down-pointing arrows that are used to increment and decrement the numeric value. A value can also be typed directly into the text box. The priority value is used in conjunction with the pulldown menu below the 'Task Priority' label. The items in the menu have the same meanings as described above for the 'Duration' field.

2.3.4.1.11. Completed? and Completion Dates

The 'Completed?', 'Earliest Completion Date', and 'Latest Completion Date' fields define the visibility of tasks based on if and when tasks are completed. The 'Completed?' field has the same three alternatives as the 'Recurring' field described above. The completion date fields have the same text box and menu combinations as the 'Earliest Date' and 'Latest Date' fields described above.

2.3.4.1.12. Remind?

The 'Remind?' field defines visibility based on whether a scheduled item has a reminder. The field has the same three alternatives as the 'Recurring?' field.

2.3.4.1.13. Details

The 'Details' field is the same form of text box and menu combination as the 'Title' and 'Location' fields. The entered pattern defines which items are visible based on the value of their Details field.

2.3.4.1.14. Attendees

The 'Attendees' field defines the visibility of meetings based on their attendees lists. Immediately to the right of the 'Attendees' label is a text box containing a pattern string. The pattern is in the form of a comma-separated list of calendar user IDs or a regular expression. The pattern is used in conjunction with the selection in the pulldown menu below the 'Attendees' label. Specifically:

- a. When 'any' is selected in the menu, items with any attendees are visible. Typing in the text box is disabled when 'any' is selected.
- b. When 'any of' is selected, meetings are visible if they have an attendees list that contains any of the listed user IDs.
- c. When 'all of' is selected, meetings are visible if they have an attendees list that contains all of the listed user IDs.
- d. When 'none of' is selected, meetings are visible if they have an attendees list that contains none of the listed user IDs.
- e. When 'regular expression' is selected, meetings are visible when the IDs of all attendees match the regular expression pattern. The regular expression is typed into the text box; details of regular expression syntax are covered in Section 2.11.14.

The last item in the 'Attendees' menu specifies case sensitivity in the same manner as in the 'Title' menu. The default for attendees filtering is case sensitive off.

2.3.4.2. Using Custom Filters

Figure 55 shows the result of the user having defined a filter named "Fall classes" with the following criteria:

- earliest date is September 21
- latest date is December 4
- category is 'classes'

To complete the definition of this new filter, the user presses the 'Add' button at the bottom of the custom filter dialog. In response, the system adds the filter to those that are available for use. To define additional new filters, the user types new names into the 'Filter Name' text box, selects the desired settings, and presses the 'Add' button for each new filter to be defined.

The image shows a 'Custom Filters' dialog box with the following fields and options:

- Filter Name:** Fall classes
- Title:** any
- Earliest Date:** September 21, 1998 ago
- Latest Date:** December 4, 1998 from now
- Earliest Time:** 8 AM
- Latest Time:** 5 PM
- Duration:** hr min expression
- Recurring?:** yes no either
- Category:** classes
- Location:** any
- Security:** any
- Priority:** must optional either
- Task Priority:** any expression
- Completed?:** yes no either
- Earliest Completion Date:** 1 week ago
- Latest Completion Date:** today from now
- Remind?:** yes no either
- Details:** any
- Attendees:** any

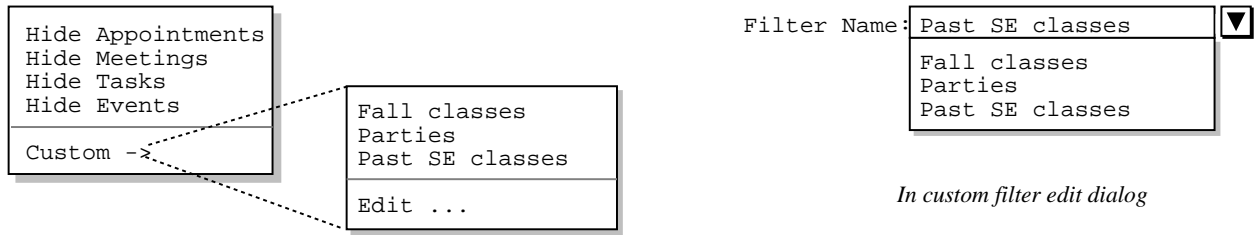
Buttons at the bottom: Add, Change, Delete, Apply, Save As ..., Load ..., Clear, Cancel.

Figure 55: Fall classes custom filter.

The names of available filters appear in two contexts:

- the 'View Filters Custom' submenu, accessible from the main command menubar;
- the 'Filter Name' menu, in the custom filters dialog.

These contexts are illustrated in Figure 56. The figure reflects the user having defined two additional custom filters named "Parties" and "Past SE classes". The filter names are sorted alphabetically in the menus. In the



In main menubar

In custom filter edit dialog

Figure 56: Menu contexts in which filter names appear.

custom filter dialog, the name in the 'Filter Name' text box is that most recently selected or typed by the user.

Consider again the example month view shown in Figure 22. Figure 57 shows the result of the user performing the following filtering commands:

- a. selecting the 'Hide' filtering command for all types of item except appointment;
- b. selecting the 'Fall classes' filter from the 'View Filters Custom' submenu.

The resulting display shows only class-category items scheduled for the Fall quarter. The figure shows how the banner of a calendar viewing window is altered to indicate that filtering is active. The format of the banner for calendar filtering is the following:

Monthly Agenda, hide M,T,E, filter Fall classes						
◀ Today ▶		September 1998				
Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21 9-10 AM O 10-11 AM 11 AM-12 3-5 PM So	22	23 9-10 AM O 10-11 AM 11 AM-12	24 1:15-3 PM 3-5 PM So	25 9-10 AM O 10-11 AM 11 AM-12	26
28	27 9-10 AM O 10-11 AM 11 AM-12 3-5 PM So	29	30 9-10 AM O 10-11 AM 11 AM-12			

Figure 57: Month view with Fall classes filter applied.

, **hide** *hide-list* , **filter** *custom-fi lter-name*

where *hide-list* is a list of one-letter abbreviations indicating which of the four types of items is (are) hidden; *custom-fi lter-name* is the name of the active custom fi lter, if any. If no items are hidden, the *hide-list* (including the trailing comma) is missing. If no custom fi lter is active, the *custom-fi lter-name* (including the leading " , **filter**" prefix) is missing.

To clarify that a fi lter is active, its name is checked in the custom fi lter menu. For example, Figure 58 shows the state of the fi lter menus as established by the user at this point in the scenarios.

When a fi lter is selected from the list, it is applied automatically to all active display windows. For example, consider the appointments list shown originally in Figure 32. Activating the "Fall classes" fi lter when this display is visible causes the system to update its display to that shown Figure 59. This fi ltered list contains only appointments of the **classes** category, per the defi nition of the 'Fall classes' fi lter.

Since the show/hide fi ltering commands do not apply to lists, the format of the banner string is different for a list display. Viz., the fi ltering banner for a list display contains only a custom fi lter name, with no hide-list.

The user can change or delete defi ned fi lters in the custom fi lter dialog. To change an existing fi lter, the user selects the desired fi lter in the 'Filter Name' menu, or types its name in the text box and presses the Enter key. The user then performs the desired changes and presses the Change button. For example, Figure 60 shows the user having edited the previously defi ned 'Fall classes' fi lter. The edits involve changing the 'Title' fi ltering fi eld to the case-insensitive pattern "'does not contain' office hours". To perform these edits, the user selects 'does not contain' in the 'Title' menu, and then types the "office hours" string in the text box. The

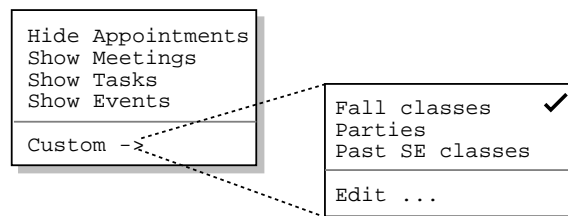


Figure 58: State of the fi lter menus with fi ltering applied.

Appointments, sorted by Date, filter Fall classes									<input type="checkbox"/>	<input type="checkbox"/>
Title	Date	Time	Duration	Recur?	Category	Location	Security	Priority		
Office Hours	Sep 21, 1998	9:00 AM	1 hr	yes	classes	14-210	public	must	△	
Data Structure	Sep 21, 1998	10:00 AM	1 hr	yes	classes	14-256	public	must		
Data Structure	Sep 21, 1998	11:00 AM	1 hr	yes	classes	14-232	public	must		
Software Engin	Sep 21, 1998	3:00 PM	2 hr	yes	classes	14-301	public	must		
Office Hours	Sep 23, 1998	9:00 AM	1 hr	yes	classes	14-210	public	must		
Data Structure	Sep 23, 1998	10:00 AM	1 hr	yes	classes	14-256	public	must		
Data Structure	Sep 23, 1998	11:00 AM	1 hr	yes	classes	14-232	public	must		
Office Hours	Sep 24, 1998	1:15 PM	1 hr 45	yes	classes	14-210	public	must		
Software Engin	Sep 24, 1998	3:00 PM	2 hr	yes	classes	14-301	public	must		
Office Hours	Sep 25, 1998	9:00 AM	1 hr	yes	classes	14-210	public	must		
Data Structure	Sep 25, 1998	10:00 AM	1 hr	yes	classes	14-256	public	must		
Data Structure	Sep 25, 1998	11:00 AM	1 hr	yes	classes	14-232	public	must		
Office Hours	Sep 28, 1998	9:00 AM	1 hr	yes	classes	14-210	public	must		
Data Structure	Sep 28, 1998	10:00 AM	1 hr	yes	classes	14-256	public	must		
Data Structure	Sep 28, 1998	11:00 AM	1 hr	yes	classes	14-232	public	must		
Software Engin	Sep 28, 1998	3:00 PM	2 hr	yes	classes	14-301	public	must		
Office Hours	Sep 30, 1998	9:00 AM	1 hr	yes	classes	14-210	public	must		
Data Structure	Sep 30, 1998	10:00 AM	1 hr	yes	classes	14-256	public	must		
Data Structure	Sep 30, 1998	11:00 AM	1 hr	yes	classes	14-232	public	must		
Office Hours	Oct 1, 1998	1:15 PM	1 hr 45	yes	classes	14-210	public	must	▽	

Figure 59: Appointments list with Fall classes fi lter applied.

Custom Filters □ □

Filter Name: ▼

Title:
 does not contain ▼

Earliest Date: ▼ ▼

Latest Date: ▼ ▼

Earliest Time: Latest Time:

Duration:
 any ▼

Recurring? yes no either

Category: ▼
 any of ▼

Location:
 any ▼

Security:
 any ▼

Priority: must optional either

Task Priority:
 any ▼

Completed? yes no either

Earliest Completion Date: ▼ ▼

Latest Completion Date: ▼ ▼

Remind? yes no either

Details:
 any ▼

Attendees:
 any ▼

Figure 60: Changed years-events filter.

'case sensitive off' menu selection is also made. To confirm these changes, the user presses the 'Change' button at the bottom of the dialog.

If the user applies this changed version of the 'Fall classes' filter in the context of Figure 57, the system updates the display as shown in Figure 61, where the appointments with "Office hours" in the title have been filtered out.

Monthly Agenda, hide M,T,E, filter Fall classes						
◀ Today ▶		September 1998				
Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21 10-11 AM 11 AM-12 3-5 PM So	22	23 10-11 AM 11 AM-12	24 3-5 PM So	25 10-11 AM 11 AM-12	26
28	27 10-11 AM 11 AM-12 3-5 PM So	29	30 10-11 AM 11 AM-12			

Figure 61: Changed Fall classes filter applied, filtering out office hours.

To delete an existing filter, the user selects its name from the 'Filter Name' menu and presses the 'Delete' button. In response, the system removes the filter from those that are defined, removes it from the 'Filter Name' menu, and clears the 'Filter Name' text box. If a deleted filter is applied at the time of its deletion, it is unapplied just prior to deletion.

The 'Apply' button is used to apply the currently selected filter, as if its name had been chosen from 'View Filters Custom' submenu. When the 'Apply' button is pressed, its text changes to 'Unapply' so a subsequent press will deactivate the filter. Hence the button acts as an apply/unapply toggle. The application of the 'Apply/Unapply' button is fully linked to the selection of a filter by name from the 'View Filters' submenu. Specifically, when 'Apply' is executed, the filter name is checked in the menu; when 'Unapply' is executed, the filter name is unchecked in the menu. Similarly, when a filter name is selected in the menu, the button label for that filter is changed to 'Unapply'; when the filter is deselected from the menu, the button name is changed to 'Apply'.

The 'Save As ...' and 'Load ...' buttons in the custom filters dialog allow the user to save and load copies of filter definitions to and from files. Details of saving and loading auxiliary data files are covered in Section 2.8.6

To clear all entered information in the custom filter dialog, the user presses the 'Clear' button. In response, the system restores the dialog to the state it was in when it was most recently displayed on the screen. The 'Cancel' button removes the custom filters dialog from the screen. Only the explicit selection of 'Cancel' (or an environment-specific window closing button) removes the dialog from the screen. Execution of any of the other commands ('Add' through 'Clear') leaves the dialog on the screen.

To deactivate a custom filter, the user reselects its name in the 'View Filters Custom' submenu. In response, the system restores all filtered displays to their unfiltered states. In addition, the filter name is removed from all display window banners and the filter name is unchecked in the 'View Filters Custom' submenu.

2.3.4.3. Details of Filter Application

As noted in the introduction, the four show/hide filter commands apply only to calendar views, not to lists. Custom filters apply uniformly to calendar and list views.

When a filter is applied, its effects are reflected in all display windows generated by the following commands:

need to enumerate these explicitly, the list is doubtless short, presumably just the top five display commands, but also need to say that filtering never effects the guts of a displayed item-level view, only whether or not an entire item-level view is displayed (see todo item 1 above)

For as long as the filter is active, all display windows are filtered, including new windows that become visible in response to new view command execution. When a filter is unapplied by reselecting its name, all windows are restored to their unfiltered states.

A custom filter is applied in conjunction with the current settings of the four show/hide filter commands. Specifically, the four show/hide commands are applied first, then any items remaining in the display are further filtered based on the definition of the currently active custom filter, if any. It is possible for a custom filter alone to filter out all items of a particular type, even if show/hide filtering is in the show state for that type of item. For example, if all meetings were defined to be of a particular category, then a custom filter that hides that category effectively hides all meetings.

For list views, a custom filter is applied in conjunction with the current settings of the list viewing options described in Section 2.7.4.1. Specifically, the list display is first configured per the option settings and then the custom filter is applied. In this way, a date range defined in a custom filter overrides any date range specified for a list particular list.

When a custom filter has the effect of filtering out all events and/or tasks in a daily or table-style weekly view, the event and/or task areas of the display are removed. This kind of filtering effect is illustrated in Figures 50 and 51. Again, a custom filter can have such an effect on its own, even if show/hide filtering for events and tasks is in the show state.

Only one custom filter can be active at any given time. The reason is to avoid the possibly conflicting effects of simultaneously applying two or more custom filters. When one custom filter is active, selecting another custom filter automatically deactivates the first filter, then activates the newly-selected filter.

As noted in Section 2.3.2.1, filtering affects the behavior of the 'Next' and 'Previous' view commands at the scheduled-item viewing level. Specifically, when filtering is active and the user executes a 'View Next' or 'View Previous' command, the system only displays items that are visible through the currently active filtering. For example, suppose the user has set filtering active as defined in Figure 58 above. In this filtering state, the 'View Next' and 'View Previous' commands display only items that meet the filtering criteria, i.e., appointments that match the 'Fall classes' filter. Items that do not meet these filtering criteria are skipped in the next/previous traversal. When a filter specifies an earliest and/or latest date, the next/previous traversal ends at those dates. For example, with the 'Fall classes' filter applied, the 'Previous' traversal stops at September 21, 1998; the 'Next' traversal stops at December 4, 1998.

When filtering is applied, all active item-level windows for the current calendar are subject to filtering by making the windows invisible. For example, if a task item window is active when the user executes 'View Hide Tasks', that window is removed from the screen. When the user subsequently executes the 'Show Tasks' command, the removed window is restored to visibility in its original X,Y position on the screen. The front-to-back position of restored windows is the same as it was at the time of filter activation, relative to all windows that were active at the time of filter activation the front-to-back positions of which did not change while filtering was active. Restored windows are placed behind all windows made newly visible or moved to the front during the time the filter is active.

Filter settings, custom filter definitions, and filter application are all calendar-specific. That is, each calendar has its own show/hide filter settings and its own custom filter definitions. When more than one calendar is open, filter application affects only the displays of the current calendar. Also, the filter settings and definitions that appear in menus, displays, and dialogs are those of the current calendar. The specific contexts in which filter settings and definitions appear are the following:

- a. the banners of filtered display windows

- b. the ‘View Filters’ command menu
- c. custom filter editing dialogs

The system provides a separate custom filter editing dialog for each calendar. Whenever the user executes the ‘View Filter Custom Edit...’ command, the system (re)displays a custom filter dialog for the current calendar. Each calendar uses a single custom filter dialog. This means that up to N custom filter dialogs can be displayed at any time, for N equal to the number of open calendars. When more than one calendar is open, the calendar name appears by default in the banner of each custom filter dialog. The appearance or non-appearance of the calendar name in window banners is controlled by an option setting, as described in Section 2.7.4.2.

2.3.5. Viewing Other Users and Group Calendars

The Calendar Tool provides cross-user access to the calendars it manages. It also provides access to the group calendars that contain the meetings scheduled for user groups.

2.3.5.1. Viewing Other User’s Calendars

To gain access to the calendar of another registered user, the current user selects ‘Other User ...’ from the ‘View’ menu. In response, the system displays the dialog shown in Figure 62. The users listed in the dialog reflect additions to the user database made by one or more system administrators. Details of user database operations are covered in Section 2.6.2.

The scrolling area at the top of the dialog lists all existing users by name and Calendar Tool ID. The default sorting order of the user list is case-insensitive by name. Since names are listed with last name first, the sorting order is primarily by last name. Secondary, tertiary, and quaternary orders are first name, middle name, and ID, respectively. The sorting order can be changed to ID by pressing the ‘ID’ column heading. The order for the ‘ID’ column is alphabetic, case-insensitive. Since IDs are unique, no secondary order is necessary when ID is the primary sort key.

To select a user for viewing, the current user clicks on the name or ID in any row of the display. The ‘Search by’ field allows the current user to type a name or ID to be searched for. As characters are typed, the list scrolls to the earliest listed name or ID with a leading prefix that matches the typed characters. Whether the search is by name or ID is based on the selected sorting order. When the list is sorted by the ‘Name’ field, the search field is labeled “Search by name”. The search field is labeled “Search by ID” when the list is sorted by ‘ID’. The current user can enter a regular expression in the search field. Details of regular expression search and syntax are covered in Section 2.11.13.

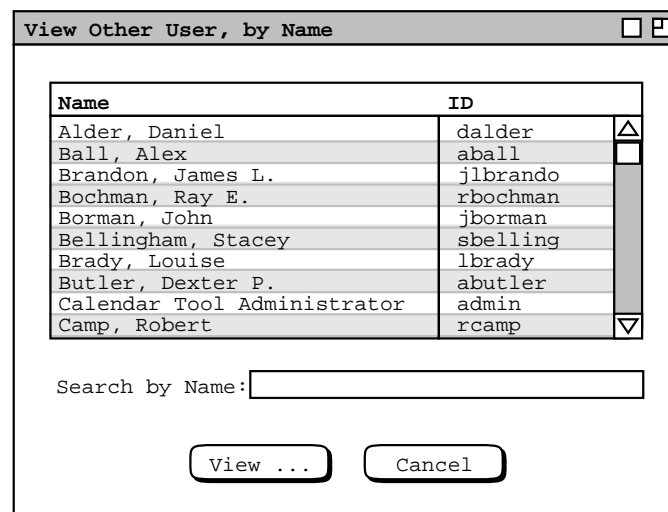


Figure 62: View other user dialog.

Figure 63 shows the user having selected user James L. Brandon for viewing. When the user presses the 'View ...' button, the system responds with a new display window as shown in Figure 64. Other users' calendars may be viewed with all of the commands available in the View menu. For example, Figure 65 shows the effect a View Day command on the September 22 entry in Figure 64. Figure 66 shows the daily view of September 23.

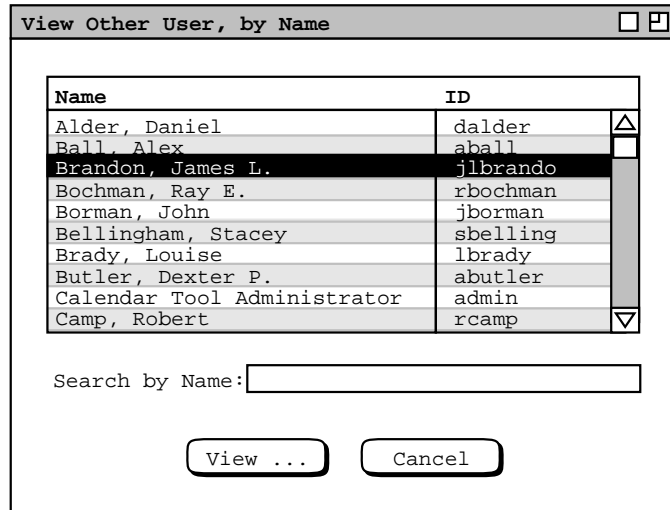


Figure 63: Selecting user for viewing.

Monthly Agenda, user James L. Brandon						
◀ Today ▶						
September 1998						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1 8-9 AM Un	2	3 8-9 AM Un	4	5
6	7 Labor Day	8 8-9 AM Un	9	10 8-9 AM Un	11	12
13	14	15 8-9 AM Un	16	17 8-9 AM Un	18	19
20	21 1-2 PM Co 3-4 PM 10 4-5 PM 10 1. prepar	22 Autumnal 8-9 AM Un 11 AM-12 1. prepar	23 all-day m 12-1:30 P 1:30-2:30 3-4 PM 10	24 8-9 AM Un	25 3-4 PM 10 4-5 PM 10	26
28	27 3-4 PM 10 4-5 PM 10 1. review	29 8-9 AM Un 11 AM-12 1. prepar	30 3-4 PM 10 4-5 PM 10			

Figure 64: Monthly view for user James L. Brandon.

Daily Agenda, user James L. Brandon	
<< Today >> Tuesday, September 22, 1998	
Events: Autumnal equinox	
Tasks: 1. prepare for grad seminar 2. prepare for Microsoft visit	
8AM	Unavailable
9AM	
10AM	
11AM	grad seminar
12PM	
1PM	
2PM	
3PM	
4PM	
5PM	

Figure 65: Day view in the James L. Brandon calendar.

Daily Agenda, user James L. Brandon	
<div style="display: flex; justify-content: space-between; align-items: center;"> ◀ Today ▶ Wednesday, September 23, 1998 </div>	
Events: all-day Microsoft visit	
8AM	
9AM	
10AM	
11AM	
12PM	Microsoft lunch
1PM	----- Microsoft rep
2PM	-----
3PM	101 lecture
4PM	101 lab
5PM	

Figure 66: Another day view in the James L. Brandon calendar.

Execution of the 'View Other User' command is independent of the windowing mode in that the system always opens a new window for each different user being viewed. Once the initial window for another user is open, the windowing mode affects subsequent viewing commands in the normal way, as described in Section 2.3.6.2

The visibility of items in other users' calendars is controlled by the security settings established by each user. As introduced originally in Section 2.2, the four security levels are the following:

- public -- other users can see all details of a scheduled item
- title only -- other users can see the title, time, date, and duration of the item, but no other information
- confidential -- other users can see the time, date, and duration of the item, but no other information; the item title shows as "Unavailable" to other users
- private -- other users can see no information at all about the item

The operational effect of these security settings is detailed in the following paragraphs. In these paragraphs, the term "current user" refers to the user who is viewing another user's calendar; the term "other user" refers to the user whose calendar is being viewed.

To the current user, public items appear in displays as they appear to the other user, except:

- a. The other user's items are not editable in item-level views. This means that the change, delete, and clear buttons that normally appear in an item-level view are absent when viewing another user at the item level.
- b. All display option settings are those of the current user; the other user's option settings have no effect from the current user's perspective. This means that the initial view of the other user's calendar is based on the setting of the current user's initial view option (see Section 2.7.4.3).
- c. Filtering is based on the current user's filter settings, if any; the filter definitions of the other user have no effect from the current user's perspective.

The category definitions of public items, including their coloring, are those of the other user. The current user's category definitions have no effect when viewing another user's calendar. Category-based filtering has no effect whatsoever when viewing another user's calendar.

'Title only' and 'confidential' security apply only to appointments and meetings, not to tasks or events. Other users' appointments and meetings with either of these securities are viewable at the month, week, and day levels, not at the item level. For 'title only' security, the title of the item appears at its scheduled date, time, and duration. For confidential security, the title "Unavailable" appears. Since the category is not visible for 'title only' and 'confidential' items, the text of their titles appear in black type only. At the item level of calendar viewing, 'title only' and 'confidential' items are skipped in the next/previous traversal (see Section 2.3.2.1).

To the current user, the precise effect of 'private' security in another user's item is to filter the item out of view, in exactly the same manner as if it had been filtered out with a custom filter definition. At the day, week, and month level, this means that private items are not visible at all. At the item viewing level, private items are not visible and are skipped in the next/previous traversal.

Per the operational details just described, the contents of Figures 64, 65, and 66 reflect the following settings in the calendar of James L. Brandon:

- a. There is a category named "school" colored red.
- b. There is a category named "industry" colored blue.
- c. There are public recurring appointments titled "101 lecture" and "101 lab", scheduled MWF 3-4 and 4-5, starting September 21, of the "school" category.
- d. There is a public recurring appointment titled "grad seminar", scheduled weekly Tu 11-12, starting September 22, of the "school" category.
- e. There is a non-recurring public appointment titled "Microsoft lunch", scheduled 12-1:30 September 23, category "industry".
- f. There is a public recurring meeting titled "Computer Science Faculty Meeting", scheduled biweekly M 1-2, starting September 21 (appearing penciled-in because it is not yet accepted by the other user)
- g. There is a title-only meeting "Microsoft rep", scheduled 1:30-2:30 September 23, category "industry" (but appears in black type because it is title-only).
- h. There is a recurring confidential appointment titled "racket ball", scheduled TuTh 8-9 AM, starting September 1 (but appears as "Unavailable" because it is confidential).
- i. There is a public recurring task titled "prepare for faculty meeting as necessary", scheduled biweekly Mon 9 AM, category "school".
- j. There is a public recurring task titled "review lecture notes", scheduled Mon 11 AM, category "school".
- k. There is a public recurring task titled "prepare for grad seminar", scheduled weekly Tu 10 AM, category "school".
- l. There is a public non-recurring task titled "prepare for Microsoft visit", Tu, no due time, September 22, category "industry".
- m. There is a private event titled "my birthday", scheduled September 24 (but does not appear because it is private).

The figures show how the banner of the display window is altered to indicate that the calendar of another user is displayed. The format of the banner is the following:

, user *full-user-name*

where *full-user-name* is the full name of the other user as registered in the Calendar Tool user database (see Section 2.6.2).

2.3.5.2. Viewing Group Calendars

When the user selects the 'Group ...' item in the 'View menu', the system displays the dialog shown in Figure 67. This dialog has the same format as the one for viewing users, shown in Figure 62. The groups listed in Figure

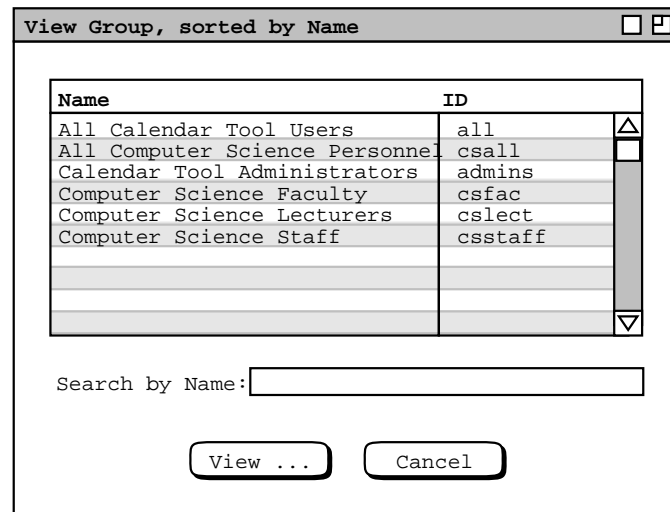


Figure 67: View group dialog.

209 reflect additions having been made by system administrators. Details of group database operations are covered in Section 2.6.3. The default sorting order for the user list is by name, which can be changed to ID by clicking on the 'ID' column heading. Since names need not be unique, the secondary sorting order is by ID when Name is the primary sort key. The 'Search by' field operates in the same manner as in the user database dialog.

To choose a group for viewing, the user selects its name or ID in the group list and presses 'View ...'. In response, the system displays the monthly calendar for the selected group, for the month containing today's date. For example, Figure 68 shows the response to the user selecting the *csfac* group for viewing. The figure shows two meetings: a recurring "Faculty Meeting", scheduled 1-2PM on alternate Mondays; a recurring "Software Engineering Colloquium", scheduled 10:30-12PM on Thursdays.

All of the rules described in the preceding section for viewing other users' calendars apply to viewing group calendars. In addition, the following rules apply specifically to viewing group calendars:

- a. only meetings appear in a group calendar;
- b. the only meetings that appear are those scheduled by a leader of the group

Section 2.4.1.6.1 discusses further the contents of group calendars.

Figure 68 shows how the banner of the display window is altered to indicate that the calendar of a group is displayed. The format of the banner is the following:

, group *full-group-name*

where *full-group-name* is the full name of the group as registered in the Calendar Tool group database (see Section 2.6.3).

2.3.6. Viewing in Different Windows

When the user selects the 'Windows ->' command in the 'View' menu, the system displays a submenu of the form shown in Figure 69. The top segment of the menu is a list of all active windows for all open calendars. An active window is any window displayed by the system, including view windows as well as data-entry dialog windows. An iconified (i.e., minimized) window is considered active. The order of the windows is chronological, starting with the most recent at the top of the list. "Most recent" is the window most recently made current by the user selecting it or by the system displaying it. "Least recent" is the window least recently made current or displayed. The window at the top of the list is *the* current window. The sample contents of the menu in Figure 69 represent the chronology of windows as they have been displayed in the requirements scenarios up to this point, assuming all scheduling dialogs have been closed and per-level windowing mode has been in effect (see Section 2.3.6.2.1).

Monthly Agenda, group Computer Science Faculty						
◀ Today ▶		September 1998				
Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22 1-2 PM Fa	23	24 10:30 AM-	25	26
28	27	29	30			

Figure 68: Monthly view for group Computer Science Faculty.

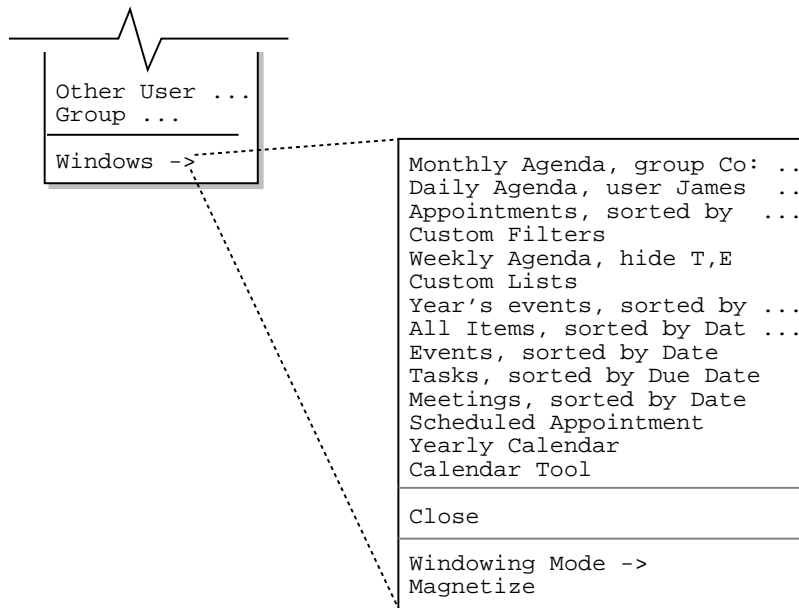


Figure 69: Windows submenu with chronological list of active windows.

The windows are listed by the strings that appear in the window banners. When the banner is longer than the default menu width, the " . . ." suffix is added. The window listed as "Calendar Tool" is the command menubar. It is listed explicitly because it is a physically separate window, however it always appears at the bottom of the list since it is never considered current for the purposes of window selection and closing.

To make any window current, the user selects its name in the 'Windows' menu. In response, the system performs the following actions:

- a. deiconifies the window if necessary;
- b. moves the window to the front on the screen, so it is in front all other windows;
- c. moves the window's title to the top of the windows menu list, unless the selected window is the command menubar;
- d. defines the calendar associated with the selected window as current, per the rules below for calendar currency, unless the selected window is the command menubar.

The same actions are performed when the user physically selects a window through the normal means of the underlying operating environment, such as clicking on its banner, border, or body.

2.3.6.1. Closing Windows

Immediately below the window list is the 'Close' window command. When the user selects this command, the system removes the current window from the screen and removes its name from the windows list. If there is a pending action in the window being closed, the action is canceled without warning. For example, if the window being closed is a scheduling dialog in which the user has made unconfirmed edits, closing the dialog window cancels all edits and does not perform any scheduling operation. In this way, closing a window has exactly the same effect as pressing the 'Cancel' button in any window that has a 'Cancel' button. After the current window is closed, the next window in the list becomes current. Any window except the command menubar can be closed with the 'Close' menu command.

The action performed by the 'Close' command is different from the action typically referred to as "iconify" or "minimize" in window-based operating environments. The iconify action does not completely remove a window, but rather reduces it to an iconic form that can be re-opened. In contrast, the close operation fully removes a window from the screen, leaving no icon or minimized item of any form. An underlying operating environment may provide both iconify and close operations. If it provides close, the Calendar Tool detects closure of one of its windows, removes the name of the closed window from the windows list and makes the next window on the list current.

In operating environments that provide an iconify or minimize operation, the effect of this operation in the Calendar tool is to change the window title in the 'Windows' menu to greyed italic type. The position of an iconified window does not change in the list, but it cannot be the current window. Therefore, if current window is iconified, its name is greyed in the list, and the current window becomes the first in the list that is not iconified. For example, Figure 70 shows the state of the 'Windows' menu in Figure 69 after the top two windows have been iconified, the 'Scheduled Appointment' window has been iconified, and the list windows have been closed. At this point, the third window in the list is current. If the user selects an iconified window in the list, that window is uniconified and made current. The same happens if the user uniconifies a window using a command of the underlying operating environment.

As noted above, the command menubar window cannot be closed using the Calendar Tool 'Close' command. If the operating environment provides a means to close a window, closing the menubar window has the same effect as executing the 'File Exit' command, described in Section 2.8.5.

When the user closes the last active window for a calendar, the system executes the 'File Close' command for that calendar, as described in Section 2.8.1.

2.3.6.2. Windowing Mode

As introduced in earlier scenarios, the user may choose from three different windowing modes. To do so, the user selects the 'Windowing Mode' command, in response to which the system displays the submenu shown in Figure 71. The windowing mode determines how the system displays the results of certain commands in the 'View' menu.

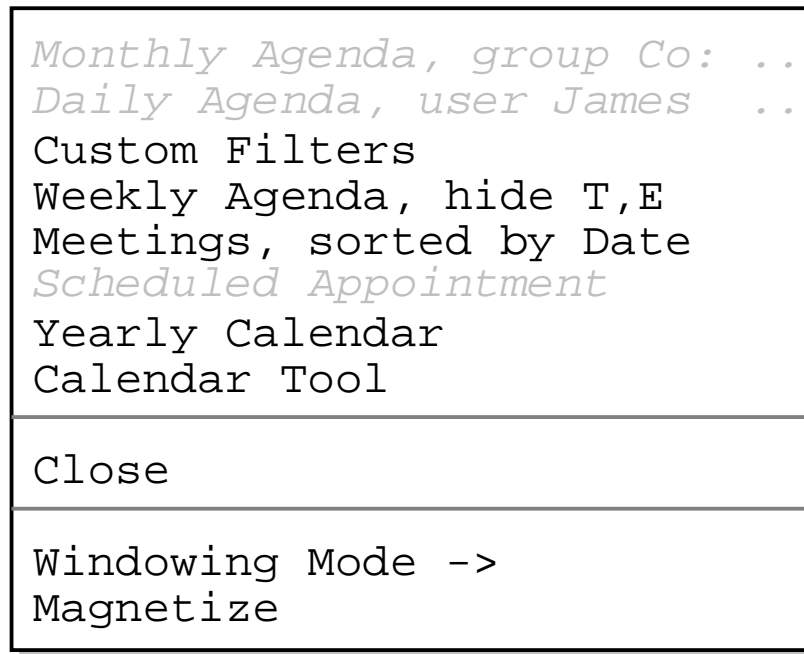


Figure 70: Iconified names in the windows menu.

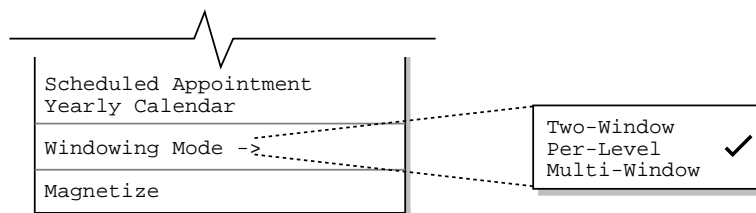


Figure 71: Windowing mode submenu.

Specifically, Figure 72 shows precisely which commands are affected by the windowing mode setting. The commands are those that display items in calendar and list format.

2.3.6.2.1. Per-Level Windowing Mode

The default windowing mode is 'Per-Level'. The windows list in Figure 69 reflects the state of windows based on per-level mode having been active during the scenarios presented to that point. In per-level mode, there are up to eleven separate viewing windows active on the screen. Table 7 describes each of these windows. For item views, the *item-type* is one of "Appointment", "Meeting", "Task", or "Event". For week views, the results of 'View Week Table' and 'View Week Lists', go in the same window. For custom lists, the *custom list name* is the name of one of the custom lists defined by the user.

When one of the eleven mode-affected view commands is executed in per-level mode, the system performs the following display actions:

- a. If there is no window currently displayed for that command, one is created and displayed on the screen.
- b. If there is an existing window for that command, the contents of the window are updated to reflect the results of the latest command execution and the window is moved to the front of all other windows on the screen.

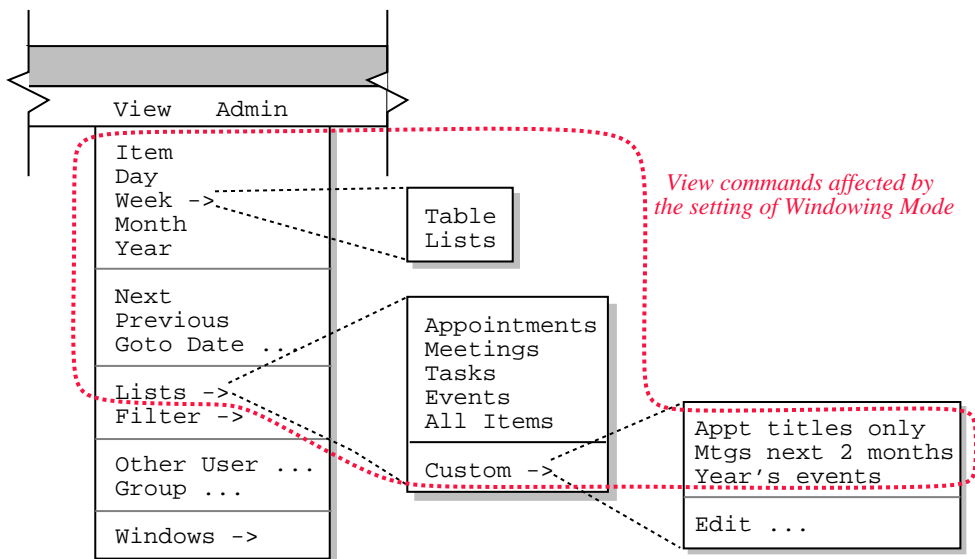


Figure 72: Commands affected by the setting of Windowing Mode.

Window Title	Command that Generates Window
"Scheduled <i>item-type</i> "	View Item
"Daily Agenda"	View Day
"Weekly Agenda"	View Week
"Monthly Agenda"	View Month
"Yearly Calendar"	View Year
"Appointments, sorted by ..."	View Lists Appointments
"Meetings, sorted by ..."	View Lists Meetings
"Tasks, sorted by ..."	View Lists Tasks
"Events, sorted by ..."	View Lists Events
"All Items, sorted by ..."	View Lists All Items
<i>custom list name</i>	View Lists Custom <i>custom list name</i>

Table 7: Windows used in per-level windowing mode.

The 'View Next' and 'View Previous' commands never create a new window in per-level mode. This is because these two commands must be executed in the context of an existing display. Therefore, 'View Next' and 'View Previous' always change the current display when executed in per-level mode.

The 'View Goto Date' command updates an existing applicable window if one exists. Otherwise it creates a new window at the default viewing level. Further operational details of 'View Goto Date' are given in Section 2.3.2.3.

In some cases, the size of an existing window may change when a command is executed in per-level mode. For example, at the item-level of viewing, the size of an appointment display is larger than an event display. In such cases, the window is positioned so that its upper left corner stays in the same screen location.

2.3.6.2.2. Two-Window Mode

In two-window mode, a single window is used to display all calendar viewing commands; another single window is used to display all list commands. Hence there up to two active display windows in two-window mode. The calendar viewing commands are the first five in the 'View' menu, and the first five listed in Table 7. The list viewing commands are those in the 'View Lists' submenu, and the last six listed in Table 7.

When the user selects two-window mode, the mode transition does not itself change the state of the screen. Rather, the transition establishes how windows are displayed subsequently. Hence, the transition to two-window mode from per-level or multi-window modes does not remove any windows from the screen.

When one of the five calendar view commands is executed in two-window mode, the system performs the following actions:

- a. If there is no calendar view window currently displayed, one is created and displayed on the screen.
- b. If there is an existing calendar view window, the contents of the window are updated to reflect the results of the latest command execution and the window is moved to the front of all other windows on the screen.

The same two actions apply to the six list viewing commands for the single list-viewing window.

Three or more windows can be displayed while in two-window mode, but only two of them are actively used for command display. If the user changes the active calendar window while in two-window mode, the newly current window becomes that in which all subsequent single-window calendar display occurs. The same applies to a change in the current list view window.

The size of an existing window may frequently change when a command is executed in two-window mode. In such cases, the window is positioned so that its upper left corner stays in the same screen location.

2.3.6.2.3. Multi-Window Mode

In multi-window mode, a separate window is used for each and every execution of a mode-affected view command, even for commands at the same calendar level and for list commands of the same type. The purpose of multi-window mode is to allow the user to create side-by-side displays of consecutive calendar periods, or other useful combinations of multi-window information. For example, Figure 73 shows a side-by-side three-month display, which the user creates by performing the following actions:

- a. execute the 'View Windowing Mode Multi-Window' command
- b. execute 'View Month' for the month of September 1998
- c. execute 'View Previous' and 'View Next' from the September display
- d. resize the monthly agenda windows to smaller sizes and move them next to each other on the screen

Monthly Agenda <input type="checkbox"/> <input type="checkbox"/>							Monthly Agenda <input type="checkbox"/> <input type="checkbox"/>							Monthly Agenda <input type="checkbox"/> <input type="checkbox"/>						
<< Today >> August 1998							<< Today >> September 1998							<< Today >> October 1998						
Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
						1			1 8-9	2	3 8-9	4	5					1 8-9	2 9-10	3
2	3	4 11-2	5	6	7	8	6	7 Labo	8 8-9	9	10 8-9	11	12	4	5 1. C	6 8-9	7 9-10	8 8-9	9 9-10	10
9	10	11 11-2	12	13	14	15	13	14	15 8-9	16 1. S	17 8-9	18	19	11	12 1. C	13 8-9	14 9-10	15 8-9	16 9-10	17
16	17	18 11-2	19	20	21	22	20	21 1. C	22 Autu	23 8-9	24 Jim'	25 8-9:	26	18	19 1. C	20 8-9	21 9-10	22 8-9	23 9-10	24
23	24	25 11-2	26	28	27	29	28	27 1. C	29 8-9	30 9-10				25 Dayl	28 1. C	26 8-9	27 9-10	29 8-9a	30 9-10	31 Hall

Figure 73: Side-by-side three-month display.

In multi-window mode, or when multiple calendars are open, it is possible for multiple windows to have exactly the same banner string. For example, Figure 73 shows three windows with the same banner. When this occurs, the window banners are not disambiguated in any way. This means that when the windows are listed in the 'View Windows' submenu, their names are identical.

2.3.6.2.4. Windowing Mode Calendar Specificity

The windowing mode is calendar-specific. That is, each calendar has its own windowing mode setting. When more than one calendar is open, the windowing mode shown in the 'Windowing Mode' menu is that for the current calendar and applies only to the current calendar. In per-level and two-window modes, the maximum number of displayed windows is computed on a per-calendar basis. For example, if three calendars are open and two-window mode is on for all three, then up to six windows can be displayed -- two windows for each calendar.

2.3.6.3. Magnetizing Windows

The window magnetizing feature of the Calendar Tool helps the user manage multi-window displays. When a window is magnetized, it "sticks to" other windows when it is moved in close proximity to them. "Close proximity" is defined by default as four screen pixels, where a pixel is the smallest granularity of screen measurement available on a particular physical display. The user can change this value with an option command, as described in Section 2.7.4.2.

To magnetize any window, the user selects that window and then chooses the 'Magnetize' command in the 'View Windows' submenu. In response, the system turns on the magnetic property of the window. The magnetizing command acts as an on/off toggle, switching between the states 'Magnetize' and 'Demagnetize'. The state of the command label reflects the magnetized state of the current window. Specifically, if the current window is not magnetized, then the command label is 'Magnetize'. If the current window is magnetized, then the command label is 'Demagnetize'.

When the user moves the side of a magnetized window in close proximity to the side of another window, the two sides are snapped into alignment and the windows become magnetically connected. The magnetic effect between two windows is the same if either or both are magnetized. Hence, moving an unmagnetized window in proximity to a magnetized one, or vice versa, has the same effect.

When two or more windows are magnetically connected, moving any one of the windows moves all connected windows along with it, so that the relative positions of the windows is maintained. A magnetized window can be moved independently by holding down the shift key while moving, which temporarily demagnetizes the window during the move. A window can be moved independently without the shift key whenever it is demagnetized and not connected to any other magnetized windows.

Figures 74 through 80 illustrate the series of actions taken by the user to produce the side-by-side display in Figure 73. Figure 74 is the initial state of the three windows as created by the 'View Month', 'View Previous', and 'View Next' commands. The windows are positioned per an assumed policy of the underlying operating environment, which in this case positions new windows down and to the right relative to earlier windows. In Figure 74, the user has set the magnetize property on for all three windows. In Figure 75, the user moves the August window so its right side is in close proximity to the left side of the September window. In response, the system snaps the August window into left-to-right alignment with the September window, as shown in Figure 76. The default left-to-right spacing is one screen pixel, which value can be changed with an option setting (Section 2.7.4.2). When the system snaps two windows into alignment, the window being moved by the user is that which changes position. The position of the other window, that is the one the user is not moving, remains fixed.

In Figure 77 the user moves the August window up while holding down the shift key, so that the top of the August window is in close proximity to the top of the September window. If the user did not use the shift key during the move, both the August and September windows would move together, since they are now magnetically attached. During the August move, the system maintains the left-to-right alignment of the two windows as long as the user does not move the mouse more than four pixels to the left or right from its initial left-to-right position at the time the move began. In response to the upward movement of the August window, the system snaps the top of the August window into alignment with the top of the September window, as shown in Figure 78.

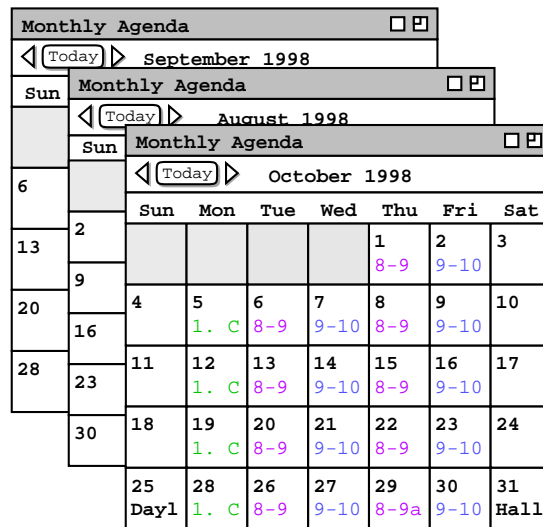


Figure 74: Initial unaligned positions of three magnetized windows.

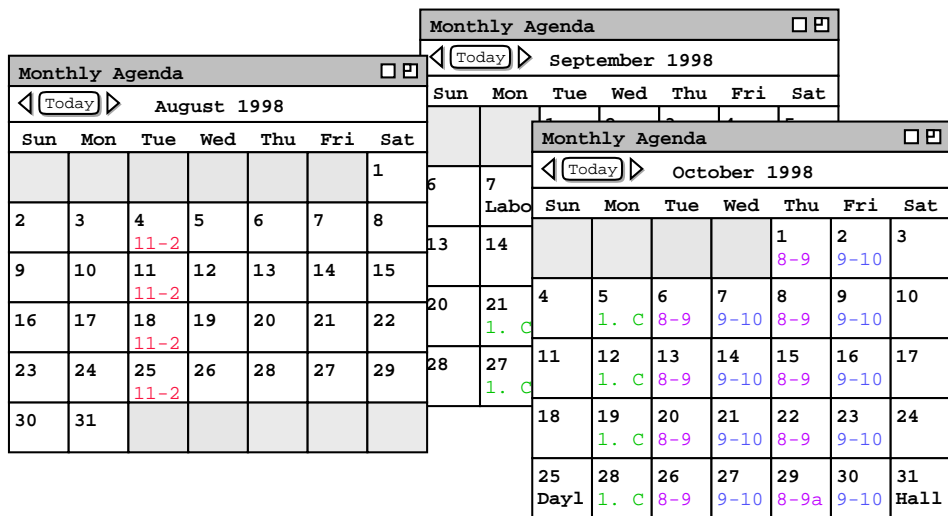


Figure 75: User moves right side of August window near left side of September window.

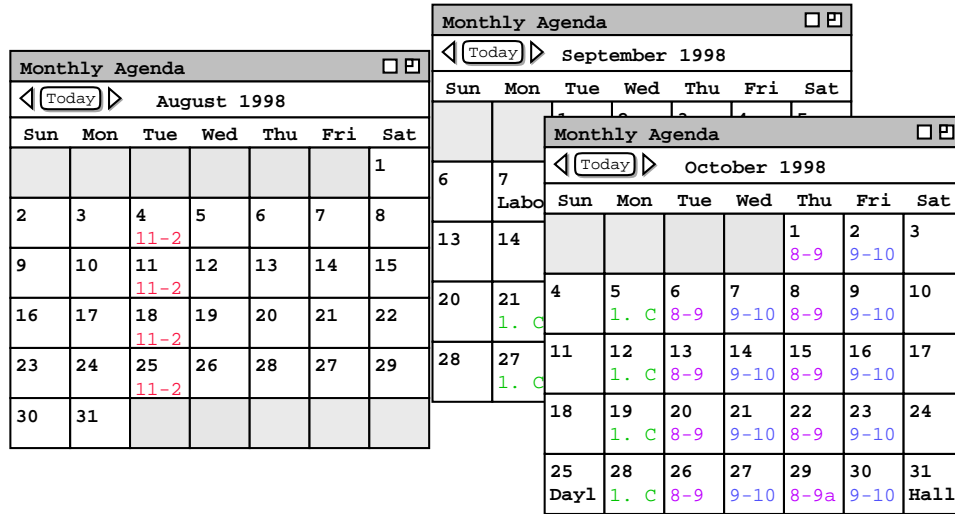


Figure 76: System snaps windows into left-to-right alignment.

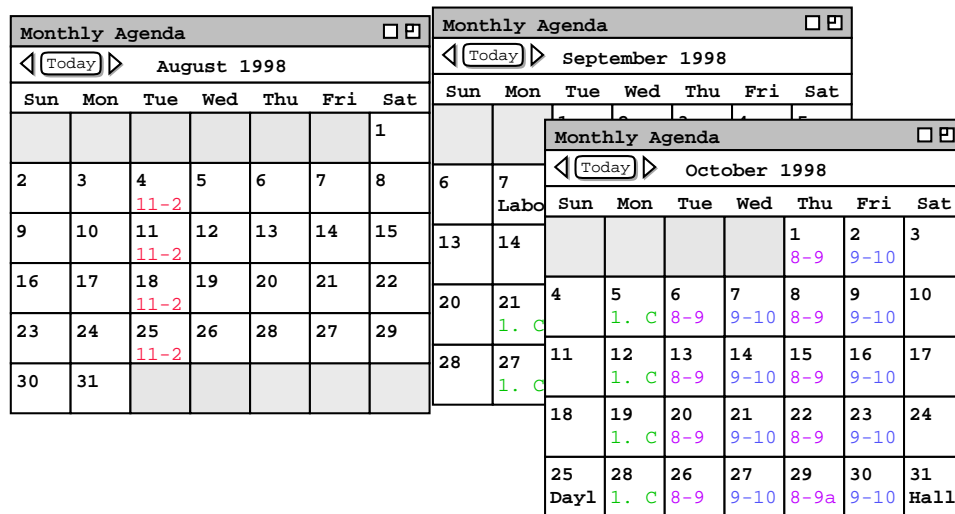


Figure 77: Holding shift key, user moves top of August window near top of September window.

Monthly Agenda <input type="checkbox"/> <input type="checkbox"/>							Monthly Agenda <input type="checkbox"/> <input type="checkbox"/>								
◀ Today ▶ August 1998							◀ Today ▶ September 1998								
Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat		
						1									
2	3	4 11-2	5	6	7	8	6	7 Labo							
9	10	11 11-2	12	13	14	15	13	14			1 8-9	2 9-10	3		
16	17	18 11-2	19	20	21	22	20	21 1. C	4	5 1. C	6 8-9	7 9-10	8 8-9	9 9-10	10
23	24	25 11-2	26	28	27	29	28	27 1. C	11	12 1. C	13 8-9	14 9-10	15 8-9	16 9-10	17
30	31								18	19 1. C	20 8-9	21 9-10	22 8-9	23 9-10	24
									25 Dayl	28 1. C	26 8-9	27 9-10	29 8-9a	30 9-10	31 Hall

Figure 78: System snaps windows into top alignment.

Monthly Agenda <input type="checkbox"/> <input type="checkbox"/>							Monthly Agenda <input type="checkbox"/> <input type="checkbox"/>							Monthly Agenda <input type="checkbox"/> <input type="checkbox"/>							
◀ Today ▶ August 1998							◀ Today ▶ September 1998							◀ Today ▶ October 1998							
Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	
						1			1 8-9	2	3 8-9	4	5					1 8-9	2 9-10	3	
2	3	4 11-2	5	6	7	8	6	7 Labo	8 8-9	9	10 8-9	11	12	4	5 1. C	6 8-9	7 9-10	8 8-9	9 9-10	10	
9	10	11 11-2	12	13	14	15	13	14	15 8-9	16 1. S	17 8-9	18	19	11	12 1. C	13 8-9	14 9-10	15 8-9	16 9-10	17	
16	17	18 11-2	19	20	21	22	20	21 1. C	22 Autu	23 8-9	24 Jim' 8-9:	25	26	18	19 1. C	20 8-9	21 9-10	22 8-9	23 9-10	24	
23	24	25 11-2	26	28	27	29	28	27 1. C	29 8-9	30 9-10				25	28 Dayl	26 8-9	27 9-10	29 8-9a	30 9-10	31 Hall	
30	31																				

Figure 79: Without shift, user moves top and right of August/September pair near top and right of October.

Monthly Agenda <input type="checkbox"/> <input type="checkbox"/>							Monthly Agenda <input type="checkbox"/> <input type="checkbox"/>							Monthly Agenda <input type="checkbox"/> <input type="checkbox"/>							
◀ Today ▶ August 1998							◀ Today ▶ September 1998							◀ Today ▶ October 1998							
Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	
						1			1 8-9	2	3 8-9	4	5					1 8-9	2 9-10	3	
2	3	4 11-2	5	6	7	8	6	7 Labo	8 8-9	9	10 8-9	11	12	4	5 1. C	6 8-9	7 9-10	8 8-9	9 9-10	10	
9	10	11 11-2	12	13	14	15	13	14	15 8-9	16 1. S	17 8-9	18	19	11	12 1. C	13 8-9	14 9-10	15 8-9	16 9-10	17	
16	17	18 11-2	19	20	21	22	20	21 1. C	22 Autu	23 8-9	24 Jim' 8-9:	25	26	18	19 1. C	20 8-9	21 9-10	22 8-9	23 9-10	24	
23	24	25 11-2	26	28	27	29	28	27 1. C	29 8-9	30 9-10				25	28 Dayl	26 8-9	27 9-10	29 8-9a	30 9-10	31 Hall	
30	31																				

Figure 80: System snaps windows into top and left-to-right alignment.

To complete the side-by-side configuration, the user moves the connected August/September window pair so its top and right sides are in close proximity to the top and left sides of the October window, respectively, as shown in Figure 79. During this move, the user does not hold down the shift key so that the window pair moves together. This can be accomplished by moving either of the two connected windows; the other window remains connected during the move. In response to this final move, the system snaps the August/September pair into both top and left-to-right alignment with the October window, as shown in Figure 80

Magnetic connection is only made between the outside edges of windows, not with inside edges. This means that magnetically connected windows never overlap with each other, although they can overlap with other unconnected windows. When the side of a moving window comes into proximity with the side of an overlapping window such that the outside edges can be connected without overlap, the system snaps the windows into alignment. This case is illustrated in Figure 75 above.

In total, there are six possible types of window alignment, as described in Table 8. The top-to-bottom alignment type is considered the same as what could be called bottom-to-top alignment. Similarly, left-to-right alignment is considered the same as right-to-left alignment.

The magnetic connection between two windows takes effect as soon as the windows are in close proximity. This means that the system snaps the moving window up to the proximate window as soon and as long as the two windows are in close proximity, including while the user continues to move the original moving window. Movement is considered to be continuing until the user performs whatever environment-specific action is required to discontinue a move, such as releasing a mouse button. At that point, the windows are connected, so that they move together on all subsequent moves that are not augmented with the shift key or until all connected windows are demagnetized.

When the magnetic effect is initially established, the moving window continues to move independently until the user discontinues the move. As long as the user does not move out of proximity with the snapped-to window, the moving window stays aligned along the axis of connection. This is the same form of alignment-constrained movement as when the user employs the shift key while moving a magnetically-connected window, as illustrated in Figure 77. With a left-to-right connection, alignment-constrained movement in the vertical directions occurs freely, but there is "resistance" to horizontal movement. That is, when the user moves the mouse horizontally, the window itself does not move until the distance moved by the mouse is greater than the proximity distance. With top-to-bottom alignment, the same form of free and resisted movement occurs, but in the horizontal and vertical directions, respectively.

Type of Alignment	Description
top	The top sides of two windows are at the same vertical position on the screen.
bottom	The bottom sides of two windows are at the same vertical position on the screen.
left	The left sides of two windows are at the same horizontal position on the screen.
right	The right sides of two windows are at the same horizontal position on the screen.
top-to-bottom	The top side of one window is one pixel below the bottom side of the other window.
left-to-right	The left side of one window is one pixel to the right of the right side of the other window.

Table 8: Types of alignment between windows.

If the user moves a magnetized window in close proximity to the aligned sides of two or more other windows, all windows become magnetically connected. If the user moves a magnetized window in close proximity to the *unaligned* sides of two or more windows, the system establishes the magnetic connection with only the closest proximate window. As explained above, a moving window stays connected with a proximate window until the moving window goes out of proximity. Given this, the magnetic effect that would normally occur between two windows may not take effect if the sides of two or more windows are out of alignment by less than the proximity distance. Figure 81 illustrates such a case, assuming that the default values for proximity distance and magnetic spacing are in effect, which are four pixels and one pixel respectively. In Figure 81a, window 0 is magnetized and about to be moved by the user. The left sides of windows 1 through 4 are out of horizontal alignment by only one pixel. That is, the left side of window 2 is one pixel to the right of the left side of window 1, window 3 is two pixels to the right of window 1, and window 4 is three pixels to the right. If the user moves window 0 to the right, the first window with which it comes into proximity is window 1, as shown in Figure 81b. At this point, the system snaps window 0 into alignment with window 1, as shown in Figure 81c. If the user continues to move window 0 to the right, the movement does not take effect until the mouse has moved a distance of five pixels, as shown in Figure 81d. This behavior is due to the magnetic effect not being overcome until window 0 moves more than the proximity distance away from window 1. Given this, no magnetic connection can be established with windows 2 or 3 while moving in this direction. From the position in Figure 81d, the system follows the normal rule for magnetic connection by snapping window 0 into left-right alignment with window 4, as shown in Figure 81e.

Connected windows are treated separately for the purposes of current window selection. That is, when the user selects one window of a connected set of windows, the selected window alone becomes current, and that window alone is the one to which the 'Magnetize'/'Demagnetize' command applies. Hence, the 'Magnetize'/'Demagnetize' command affects exactly one window per execution. If the current window on the screen is not a calendar or list display window, e.g., it is a scheduling dialog, then the 'Magnetize'/'Demagnetize' command is disabled. In order for a window to be moved independently without using the shift key, it and all windows to which it is magnetically connected must be demagnetized.

Magnetization is not transferred from one window to another via magnetic connection. For example, suppose in Figure 79 that only the August window is magnetized, with the September and October windows not magnetized. In this case, the system would not perform the magnetic attachment to produce the Figure 80 configuration. This is because neither the September nor October windows is magnetized, and the magnetization of the August window is not transferred to the September window. If the user moved the August window into close proximity of the October window, the system would connect the windows, since only one of a window pair need be magnetized for the connection to be made.

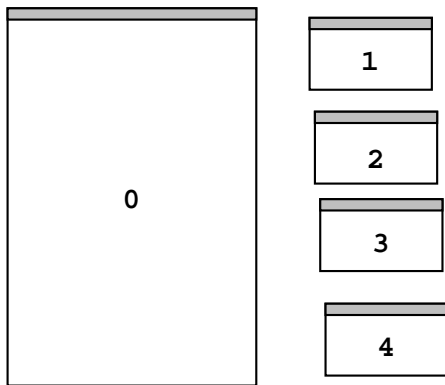
2.3.6.4. The Calendars List

Zero or more calendars can be open in a given Calendar Tool session. The names of open calendars are displayed using the 'Calendars ->' item at the bottom of the 'View' menu. When the user selects this item, the system displays a submenu of the form shown in Figure 82. The submenu shows the names of all open calendars. The menu list is sorted chronologically, with the current calendar at the top of the list and the least recently current calendar at the bottom of the list. In this example, the user has opened five different calendars.

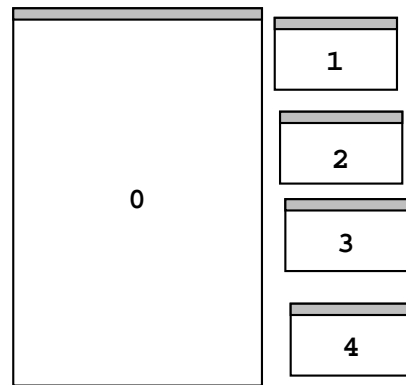
If a calendar has unsaved changes, its name in the calendars list is marked with a boldface asterisk character. The calendars of other users and groups are shown in grey italic type.

When a calendar has been saved to a file, the calendar name is synonymous with its root file name, where "root" is the file name without any leading file path and without any file extension. If a calendar has not yet been saved to a file, its name is "unnamed". If more than one calendar of the same name is open, then the names of the second calendar and beyond are suffixed with [*i*], for *i* equal 2 to the number of calendars with that same name. Two or more calendars can have the same name if they have the same root name in different file directories, or they are an "unnamed" calendar.

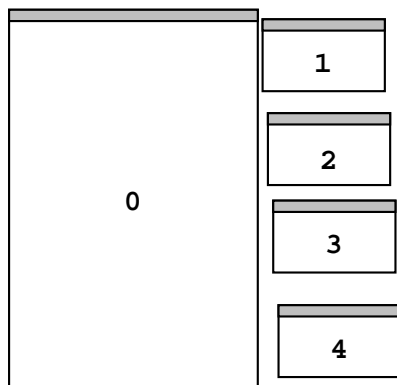
Other-user and group calendars cannot be saved to a file. Therefore, the names of these calendars consist of the keyword "user" or "group", followed by the name of the user or group who owns the calendar. Other-user and group calendars are not editable, therefore their names are never shown with an asterisk in the Calendars list.



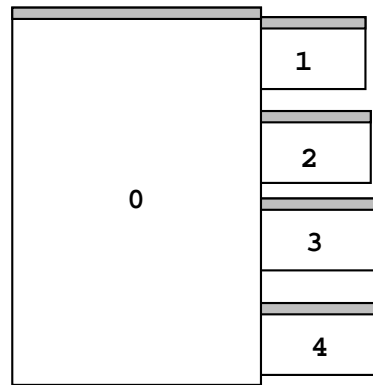
a. Windows 1- 4 are slightly out of alignment.



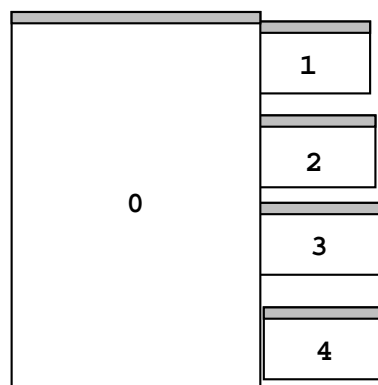
b. User moves window 0 into proximity of window 1



c. System snaps window 0 into left-to-right alignment with window 1.



d. User moves window 0 five pixels to the right..



e. System snaps window 0 into left-to-right alignment with window 4, skipping windows 2 and 3.

Figure 81: Moving a magnetized window near other slightly unaligned windows.

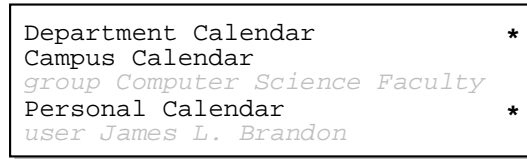


Figure 82: Calendars submenu.

The user can change the current calendar by selecting its name in the list. In response, the system performs the following actions:

- a. defines the selected calendar as current for the purposes of all subsequent Calendar Tool operations that reference the current calendar;
- b. moves the calendar's name to the top of the 'Calendars' menu list
- c. makes current the most recently active window of the selected calendar, per the above rules for window currency
- d. updates the custom lists menu to reflect the custom lists defined for the selected calendar (see Section 2.3.3.6 for explanation of the custom lists menu)
- e. updates the custom filters menu to reflect the custom filters defined for the selected calendar (see Section 2.3.4.2 for explanation of the custom filters menu)
- f. updates the windowing mode setting to that of the selected calendar (see Section 2.3.6.2.4)

A calendar is also made current whenever the user selects one of its windows in the 'Windows' list.

An other-user or group calendar can be current but is not editable. When an other-user or group calendar is current, all calendar modification commands are disabled. Section 2.11.1 covers details of command enabling and disabling.

The association between a calendar and its windows is based the commands that display the windows and the order in which the commands are executed. From the time the Calendar Tool is initially invoked, calendar-to-window association is established as follows:

- a. An initial view window is displayed for a calendar when the user executes one of the following commands: 'File->New', 'File->Open', 'View Other User,' or 'View Group'; the calendar displayed in the initial window becomes the current calendar.
- b. While a calendar remains current, all windows created as a result of non-administrative command execution are associated with that calendar.
- c. When the user creates, opens, or makes current another calendar, all subsequent non-administrative windows are associated with that calendar.
- d. The current calendar remains so until the user executes a 'New', 'Open', 'Other User', 'or Group' command, or until the user makes current a different calendar by selecting one of its windows or selecting its name in the 'Calendars' menu.

Administrative commands, that is commands in the 'Admin' menu, are not calendar-specific. When an administrative command window is current, the current calendar is that most recently current prior to display of the administrative window(s). At all times, the name of the current calendar is displayed in the banner of the command menubar and is listed at the top of the 'Calendars' menu.

2.3.6.5. Window Banner Syntax

As described throughout the functional requirements, the banners of Calendar Tool windows contain text that is appropriate to the window contents. The banner of the command menubar window has a format that is distinct from all other windows. For the Calendar Tool regular user interface, the menubar banner format is the following:

Calendar: *calendar name* [, **Host:** *host name*] [, *filtering*]

The *calendar name* is that of the current calendar, as it appears at the top of the *Calendars* list. If no calendar is currently open, the calendar name is "none". The *host name* is the name of the Calendar Tool central host computer to which the user is currently connected, as described in Section 2.6.6.2. If the user is not connected to a central host,

the **Host** component of the banner is absent. The *filtering* component of the banner is formatted as described in Section 2.3.4. If no filtering is active, the *filtering* component is absent.

The following is the banner format for all Calendar Tool windows except the menubar:

content description [, *filtering*] [, *calendar name*]

The *content description* is the context-specific string appropriate to each window. Details of content descriptions are described throughout the functional requirements. The appearance of *filtering* and *calendar name* are controlled by option settings, as described in Section 2.7.4.2. If present, the *filtering* and *calendar name* banner components have the same format as in the menubar window.

For the Calendar Tool Administration program, the menubar banner format is the following:

Calendar Tool Administration: *host name*

where *host name* is the name of the central host computer on which the administration program is running. For all other Calendar Tool Administration windows, the banner contains only a context-specific content description. Since calendars are not open within the Calendar Tool Administration program, the *filtering* and *calendar name* components do not apply.

2.4. More Scheduling

The scenario in Section 2.2 focused on appointment scheduling. The scenarios in this section describe the scheduling of the three other types of scheduleable item:

- meetings
- tasks
- plain events

2.4.1. Scheduling Meetings

The scenarios in this section show how meetings are scheduled for Calendar Tool users. A scheduled meeting has all the information that an appointment has, plus the following additional information:

- a list of the meeting attendees
- the ID of the user of who schedules the meeting
- the date and time scheduling takes place
- a reference to the minutes of the meeting, once it has taken place

Meetings are scheduled among registered users of the Calendar Tool system. For convenience, the Calendar Tool provides administrative functions to define groups of users among whom meetings may be regularly scheduled. Complete details of defining user groups are covered in Section 2.6.3. For the purposes of meeting scheduling, a group is viewed simply as a collection of Calendar Tool users with a common group name. When the group name is listed in the attendees list for a meeting, all members of the group are considered attendees.

User groups have leaders, who are themselves Calendar Tool users. When a group leader schedules a meeting for a group, the system automatically notifies group members about the meeting and adds the meeting to the group calendar. Meetings can also be scheduled by users who are not group leaders, with individual users as attendees. Administratively, scheduling a meeting for a group as a whole is restricted to leaders of the group.

Meeting scheduling takes place in the context of a multi-user operating environment. Complete details of the environment are covered in Section 2.14. The following paragraph summarizes the environment features that are relevant to the meeting-scheduling scenarios presented in this section.

The Calendar Tool multi-user operating environment consists of a central host computer and user client computers. Calendar Tool users, including meeting schedulers, run the Calendar Tool on their local client computers and connect to a common central host computer via some form of computer network. User calendars are stored in files on their local computers. In addition, the central host has a copy of the public portions of each user's calendar. When a meeting is scheduled, a not-yet-accepted version of the meeting is added to the central host calendars of all attending users. A meeting notification is then sent to the local computer of each attending user. If the user responds to the

notification by accepting the meeting, it is marked as accepted on the central host and copied to the user's local calendar. If the user declines the meeting, the not-yet-accepted copy is deleted from the central host calendar and no change is made to the user's local calendar.

In the scenarios that follow, meeting schedulers and meeting attendees are all assumed to be connected to the same central host computer. Details of establishing a central host connection are presented in Section 2.6.6.2.

2.4.1.1. A Group Leader Scheduling Two Straightforward Meetings

The scenario in this section presents a group leader scheduling both a non-recurring and recurring meeting, with some variations in scheduling details. In this scenario and the meeting scenarios to follow, "the scheduler" refers to the user performing the scheduling.

To begin the meeting scheduling process, the scheduler selects the 'Meeting' command in the 'Schedule' menu. In response, the system displays the dialog shown in Figure 83. Scheduling a meeting is much like scheduling an appointment, as described in Section 2.2. The principle difference between meeting versus appointment scheduling is that a meeting dialog defines a range of possible dates and times, rather than a single date and time as for an appointment. Hence, a meeting dialog holds a form of scheduling *request*. Key elements of the request are a date/time range, a list of attending users, and an optional meeting location. When the dialog is filled in, the system computes all the dates and times that satisfy the request. The scheduler then chooses a specific desired meeting time from those computed.

The following fields in the scheduling request dialog hold the same form of information as the appointment scheduling dialog in Figure 6:

- Title
- Duration
- Category
- Security
- Location
- Priority
- Remind, and its accompanying fields
- Details

The scheduler enters information for these fields in the same manner as for an appointment, as described in Section 2.2.

The date and time fields for a scheduling a meeting are notably different than for an appointment. In the case of a meeting, there are earliest and latest possible dates, and earliest and latest possible start times. These date and time ranges are used by the scheduler to indicate the range of acceptable dates and times in which the meeting can be scheduled. For a recurring meeting, the earliest and latest end dates are enabled. These fields indicate the acceptable range of dates for the last occurrence of a recurring meeting.

The Recurring checkbox and Interval menu are the same for meetings as they are for appointments. The recurring details are different for a meeting request, as discussed further below.

The Attendees field contains a comma-separated list of attendee IDs. The IDs are those of individual calendar tool users or user groups. The ID of a user or group is the unique identifier by which the user or group is known in the Calendar Tool system. IDs are defined and searched for as described in Section 2.6. The IDs in the Attendees list need not be unique. The system ignores duplicates in the attendees list in that only a single meeting notification is sent to each attending user, even if a user is listed more than once in the attendees list. Duplication of users in the attendees list can arise when a single user is a member of two or more groups appearing in the attendees list.

The 'Minutes' text field specifies the location of the meeting minutes as a WWW URL. In the case of a recurring meeting, the scheduler may specify the URL as a directory rather than as an individual file. When a directory URL is entered, the scheduler must subsequently enter the name for each individual minutes file by editing individual meeting occurrences, as described in Section 2.5.2.2. Details of how users view meeting minutes are covered in Section 2.5.1.2.

Figure 83: Meeting scheduling request dialog.

Figure 84 shows the scheduler having entered information for a non-recurring meeting of the computer science staff. The earliest and latest possible dates are the Monday and Friday of the week of September 20. The earliest and latest start times are 8 AM to 6 PM. The category is blank and there is no reminder. To proceed with the scheduling operation, the user presses the 'List Times ...' button. In response, the system displays a list of possible meeting times, as shown in Figure 85. The scheduling request dialog remains on the screen, but all of its data-entry fields and command buttons are disabled until the meeting confirmation process concludes.

The list in Figure 85 contains all the meeting times that conform to the values entered in the scheduling request dialog. Each item in the list consists of a time range, day of the week, and date. If the listed items span to or more years, date includes the year. The precise meaning of "conform to" is as follows:

- a. All meeting dates in the possible list are between the earliest and latest possible dates entered in the scheduling request dialog.

Schedule a Meeting □ □

Title:

Earliest Possible Date: Earliest Start Time:

Latest Possible Date: Latest Start Time:

Earliest End Date: Duration: ^{hr} ^{min}

Latest End Date:

Recurring? Interval:

Category: Security:

Location: Priority:

Remind?

Attendees:

Details:

Agenda:

Minutes:

Figure 84: Scheduling a staff meeting.

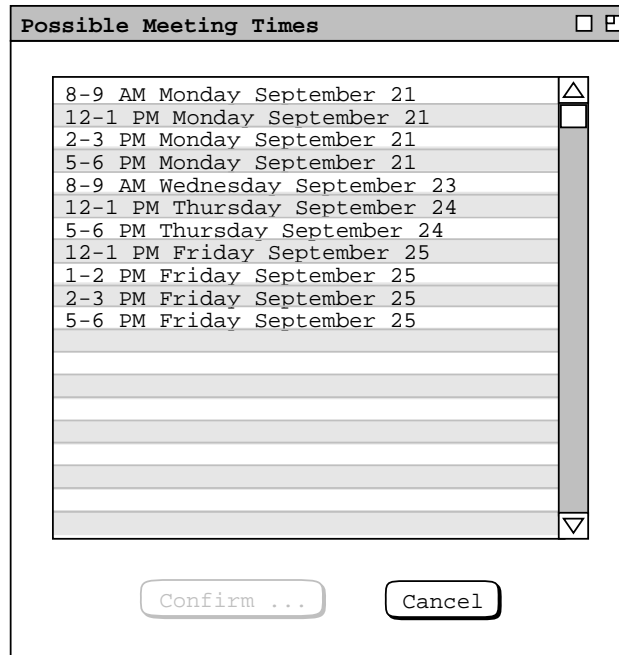


Figure 85: List of possible meeting times for the csstaff meeting.

- b. All meeting start times are between the earliest and latest start time range entered in the scheduling request dialog.
- c. All meeting durations are the value entered in the scheduling request dialog.
- d. For all attendees, there is no non-private scheduled item for the attendee that overlaps with any listed meeting time.
- e. If the location in the request dialog is in the location database (Section 2.6.4), then the location is available at the scheduled time and date. A location not in the database is assumed to be available at any time.

The list of possible times is sorted first by date and second by time, from earliest to latest. The default maximum number of listed items is twenty, which value can be changed as an option setting. The minimum number of items in the list is one. When there are no conforming possible times, the system displays an alert message instead of the list, as discussed in the next meeting scheduling scenario.

To select one of the possible meeting times, the user clicks on the desired item in the list and presses the 'Confirm ...' button, which becomes enabled when the user makes a list selection. For example, Figure 86 shows the scheduler having selected the 8-9 AM time slot on September 23.

When the scheduler presses 'Confirm ...' in Figure 86, the system responds with the meeting confirmation dialog in Figure 87. The confirmation dialog contains the information entered in the initial scheduling request dialog, with the date and time ranges replaced with the single date and time selected from the list of possible times. There are four additional data fields below 'Minutes'. The 'Scheduled By', 'On', and 'Host' fields are read-only text. The values of these three fields are set by the system to the ID of the user performing the scheduling, the date/time on which the scheduling occurs, and the Calendar Tool central host computer on which the scheduling occurs.

The scheduled 'On' field is formatted as

ddmmmyy hh:mm:ss

where dd is the date, mmm is the short month name, yy is the year, hh is the hour in 24-hour clock time, mm is the minute, and ss is the second. Section 2.6.5.1 describes the format of Calendar Tool host computer names.

Below the 'Host' field is a check box indicating whether or not to send email notification to all attendees. All attendees are automatically notified of a confirmed meeting with a pop-up dialog, as explained in Section 2.4.1.5.

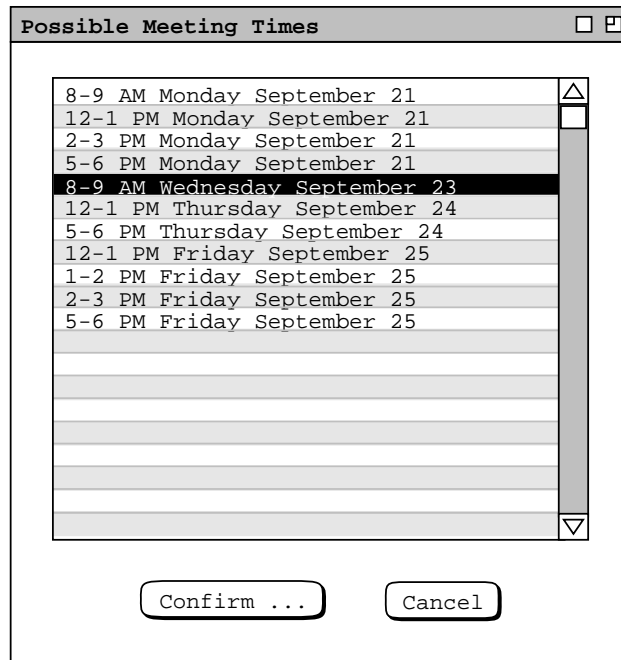


Figure 86: Selection in the possible meeting times list.

Confirm a Meeting

Title:

Date: Start Time:

End Date: Duration: hr min

Recurring? Interval: S M T W Th F S

Category: Security:

Location: Priority:

Remind? minutes before

Attendees:

Details:

Agenda:

1. Duty assignments for Fall quarter
2. News of office remodeling project
3. Introduce new staff members
4. Promotions
5. Initial prep for accreditation visits

Minutes:

Scheduled By: On:

Host:

Send email notification?

Figure 87: Staff meeting confirmation dialog.

An email message can be sent in addition to the Calendar Tool dialog notification.

The scheduler can edit some of the data-entry fields in the meeting confirmation. The uneditable fields are those that affect how the list of possible times is computed, as well as the 'Scheduled By', 'On', and 'Host' fields. These fields have greyed and dashed borders to indicate their disabled state. The complete information for a meeting is that entered initially in the scheduling request dialog (Figure 83) and subsequently in the meeting confirmation dialog (Figure 87). Information that is not necessary for the possible-times computation can be left blank in the initial request dialog. For example, if the meeting title is left blank in the scheduling request dialog, it can be entered subsequently in the confirmation dialog. The exact data entry rules for the request and confirmation dialogs are given in Section 2.11.7.1.

While the meeting confirmation dialog is displayed, the original request dialog and possible times list remain on the screen for reference purposes. However, all of the data-entry fields of these earlier two dialogs remain inactive while the confirmation dialog is displayed. Changes made in the confirmation dialog do not affect any of the values in the

original scheduling dialog. For example, if the scheduler edits the title in the confirmation dialog, the title remains as initially entered in the scheduling request dialog.

Figure 88 shows the scheduler having edited the meeting confirmation dialog by adding a reminder and correcting a spelling error in the original 'Details' data field.

After completing the edits in the meeting confirmation dialog, the scheduler confirms the meeting by pressing the 'OK' button. In response, the system schedules the meeting. On the screen, the system removes the three pending dialog windows -- the possible meeting times, the confirmation dialog, and the original scheduling request dialog.

The precise effects of scheduling a meeting are the following:

Confirm a Meeting

Title: Computer Science Staff Meeting

Date: September 23, 1998 Start Time: 8 AM

End Date: Duration: 1 hr min

Recurring? Interval: weekly S M T W Th F S

Category: Security: public

Location: 14-238 Priority: must

Remind? 15 minutes before on screen

Attendees:
csstaff

Details:
Agenda:
1. Duty assignments for Fall quarter
2. News of office remodeling project
3. Introduce new staff members
4. Promotions
5. Initial prep for accreditation visits

Minutes: http://www.csc.calpoly.edu/~estier/departme

Scheduled By: estier On: 15sep98 08:24:01
Host: csc.calpoly.edu

Send email notification?

OK Clear Cancel

Figure 88: Meeting confirmation dialog with some edits.

- a. The system adds the meeting to each attendee's calendar on the central host computer and marks each meeting as "not-yet-accepted"
- b. The system notifies each attendee except the scheduler, as discussed in Section 2.4.1.5.
- c. The system adds the meeting to the scheduler's central host calendar as well as to scheduler's local calendar, whether or not the scheduler is an attendee. The meeting item in the scheduler's calendar is a regular item, i.e., it is not marked as not-yet-accepted.
- d. The system adds the meeting to the calendar of all groups that appear in the attendees list. Details on the contents of group calendars are covered in Section 2.4.1.6.1.
- e. If the specified location is in the location database, the location is marked as booked for the duration of the meeting. If the location is not in the database, then no location-related action is taken.

Only a group leader can schedule a meeting for a group. If a non-leader attempts to do so, the system responds with an error message, as described further in Section 2.12.3.

In the case of the staff meeting in Figure 88, only the single `csstaff` group appears in the attendees list. In this scenario, the scheduler is assumed to be a leader of that group, as well as a member. Therefore, when the scheduler confirms the staff meeting, the system:

- a. adds a not-yet-accepted meeting to the central host calendars of all members of `csstaff`, except the scheduler;
- b. notifies all members of `csstaff`, except the scheduler;
- c. adds the meeting to the scheduler's central host and local calendars;
- d. adds the meeting to the `csstaff` group calendar;
- e. marks the 14-238 location as booked for the duration of the meeting.

The scheduler now proceeds to create another meeting, following the same procedure as for the preceding staff meeting. The resulting filled-in request dialog is shown in Figure 89. This is a recurring meeting for computer science lecturers. The earliest and latest start dates are the same as for the staff meeting. Since the meeting is recurring, the latest and earliest end date fields are enabled. Also, the word "Possible" in the two start date labels is changed to "Start" to clarify the display as that of a recurring meeting.

The recurring interval of the meeting is 'weekly', which is the default. In this case, the scheduler does not specify any further recurring details. This means that the system will consider all days of the week in the normal day range for which times are available. The default normal day range is Monday through Friday, which can be changed using 'Options ...' as explained in the next meeting-scheduling scenario. The 'Details ...' aspect of recurring meetings is covered in the next scenario, as well as in Section 2.5.3.4.

When the scheduler presses the 'List Times ...' button for the lecturer meeting, the system responds with the list of possible meeting times shown in Figure 90. For a weekly recurring meeting, each item in the list consists of a time range, day(s) of the week, start date, and end date. The scheduler selects the third item in Figure 90 and presses 'Confirm ...', whereupon the system displays the confirmation dialog in Figure 91. The scheduler accepts the meeting without further edits by pressing 'OK' in the confirmation dialog. In this scenario, the scheduler is assumed to be a leader but not a member of the `cslect` group. Therefore, when the scheduler confirms the lecturer meeting, the system:

- a. adds a not-yet-accepted meeting to the central host calendars of all members of `cslect`, which does not include the scheduler;
- b. notifies all members of `cslect`, which does not include the scheduler;
- c. adds the meeting to the scheduler's calendar, even though the scheduler is not an attendee;
- d. adds the meeting to the `cslect` group calendar;
- e. marks the 14-238 location as booked for the duration of the meeting.

Before a meeting is confirmed, the 'Clear' button in the confirmation dialog clears all changes made since the system initially displayed the dialog. For example, if the user presses 'Clear' in the dialog of Figure 88 the system restores the dialog state to that shown Figure 87. The 'Cancel' button in the meeting confirmation dialog removes the dialog from the screen and re-activates the possible-times dialog. Pressing 'Cancel' in the possible-times dialog

Schedule a Meeting □ □

Title:

Earliest Start Date: Earliest Start Time:

Latest Start Date: Latest Start Time:

Earliest End Date: Duration: ^{hr} ^{min}

Latest End Date:

Recurring? Interval:

Category: Security:

Location: Priority:

Remind?

Attendees:

Details:

Minutes:

Figure 89: Scheduling another meeting.

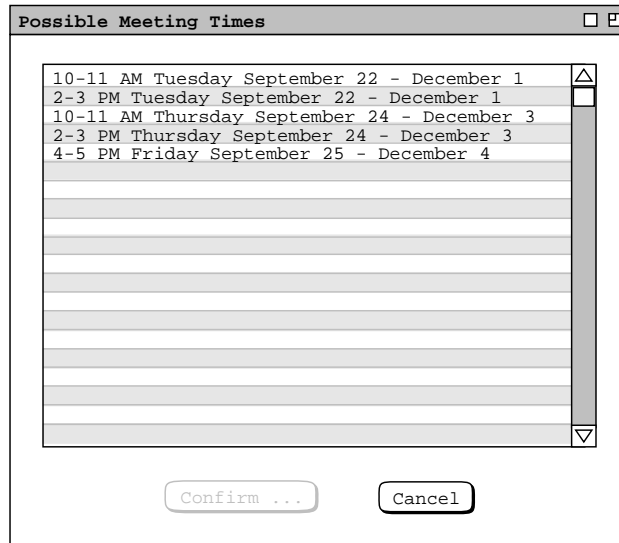


Figure 90: List of possible meeting times for the cslect meeting.

Confirm a Meeting

Title:

Date: Start Time:

End Date: Duration: hr min

Recurring? Interval: S M T W Th F S

Category: Security:

Location: Priority:

Remind?

Attendees:

Details:

Minutes:

Scheduled By: On:

Host:

Send email notification?

Figure 91: Lecturer meeting confirmation dialog.

removes it from the screen and re-activates the original scheduling request dialog. ‘Cancel’ there removes the dialog, thereby canceling the scheduling operation entirely. Hence at the level of the meeting confirmation dialog, cancelling the scheduling operation is a three-step process: first the confirmation is cancelled, then the possibilities list, and finally the original scheduling request dialog.

Once a meeting is confirmed, the item in the scheduler’s local calendar is that through which attendee-wide changes or cancellation are performed. This is the reason that a scheduled meeting is added to the scheduler’s calendar even if the scheduler is not an attendee. Changing and cancelling scheduled meetings are discussed in Section 2.5.2.2.

2.4.1.2. A Group Leader Encountering Difficulties While Scheduling

This section presents a scenario where the scheduler must adjust scheduling options in order to arrive at an acceptable meeting time. The scheduler in this scenario is a different Calendar Tool user than in the preceding scenario.

To begin, the scheduler selects 'Schedule Meeting' in the command menu and fills in the dialog as shown in Figure 92. In this case, the scheduler wants to schedule a recurring meeting on a specific day and time. The single specific time is chosen by entering the same 1 PM value for both the earliest and latest start times. To select a specific day of the week, the scheduler presses the 'Details ...' button to the right of the recurring interval. In response the system displays the dialog shown in Figure 93. The first field in the dialog specifies the number of days per week on which to schedule a recurring meeting. The default value is one; the maximum possible value is 7. The second field specifies the allowable days for the meeting. The default days are Monday through Friday. The number of allowable days must be greater than or equal to the days per week, and at least one allowable day must be checked.

Schedule a Meeting

Title:

Earliest Start Date: Earliest Start Time:

Latest Start Date: Latest Start Time:

Earliest End Date: Duration: hr min

Latest End Date:

Recurring? Interval:

Category: Security:

Location: Priority:

Remind?

Attendees:

Details:

Minutes:

Figure 92: Scheduling a Faculty Meeting.

Details for Weekly Recurring Meeting Request

Days per week:

Allowable days: S M T W Th F S

Minimum number of days between each occurrence

Maximum number of days between each occurrence

OK Clear Cancel

Figure 93: Details of a weekly recurring schedule.

The third and fourth fields specify, respectively, the minimum and maximum number of days between each meeting occurrence. These last two fields are enabled if the 'Days per week' value is greater than the number of allowable days. As an example, consider the days per week set to 2, allowable days set to the default, and the last two fields set to 1 and 2 respectively. These settings specify a meeting scheduled two days per week, with each meeting one or two days apart. This allows Monday/Wednesday, Tuesday/Thursday, Monday/Thursday, or Tuesday/Friday meeting times.

In the current scenario, the scheduler wants to schedule the faculty meeting for Mondays at 1PM, and so fills in the details dialog as shown in Figure 94. where only the Monday checkbox is selected. In this scenario, the scheduler is acting on an assumed agreement made among the meeting attendees. Specifically, the computer science faculty attendees have been asked in advance to leave the 1-2 PM MWF time slot free on their calendars. This request is assumed to have been made through some means outside of the Calendar tool, such as in-person contact or other communication. Unfortunately, not all faculty have honored the request. Hence, when the scheduler presses the 'List Times ...' button, the system responds with the message shown in Figure 95. This message indicates that the system is unable to find any meeting times that conform to the values entered in the scheduling request dialog, by the definition of "conform to" given above.

Details for Weekly Recurring Meeting Request

Days per week:

Allowable days: S M T W Th F S

Minimum number of days between each occurrence

Maximum number of days between each occurrence

OK Clear Cancel

Figure 94: Weekly recurring details filled in.

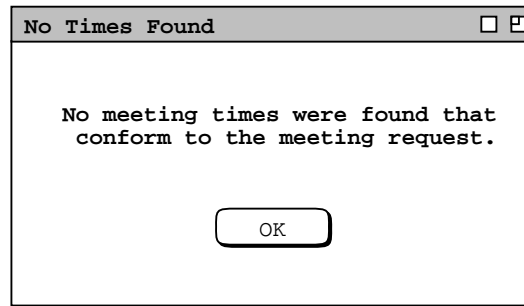


Figure 95: No times found dialog.

To address the outcome of no possible meeting times being found, the scheduler can change one or more of the values entered in the scheduling request dialog and re-execute the 'List Times ...' command. For example, the scheduler can change the range of possible times, dates, or days on which the meeting can be scheduled. In this scenario, the scheduler decides to leave the date and time range the same, but expand the possible days from Monday only to Monday, Wednesday, or Friday. To do so, the scheduler reselects the recurring 'Details ...' command and changes the values as shown in Figure 96. Here the scheduler has checked the Wednesday and Friday check boxes in addition to the Monday check box. Having made the desired changes, the scheduler presses 'OK' in the details dialog and 'List Times ...' in the scheduling request dialog. In response, the system again displays the no-times-found dialog of Figure 95, meaning that the system is still unable to find any conforming meeting times. It must be the case that one or more faculty have scheduled appointments or meetings in the 1-2 PM MWF time slots.

The other way to deal with the no-times outcome is to adjust one or more scheduling option values. To do so, the scheduler presses the 'Options ...' button at the bottom of the scheduling request dialog. In response, the system displays the dialog in Figure 97. The first option setting is the length of the possible times list, the default value for which is 20. The second and third options specify whether a meeting can be scheduled at the same time as items with 'optional' or 'must' priority on attendee calendars. If the default value of 'no' is selected for both options, the system will not attempt to schedule a meeting at the same time as any non-private scheduled items. If 'yes' is selected for either or both options, then the system will allow a meeting to be scheduled at times that overlap with optional and/or must items. Such items will appear in the possible times list, flagged with the IDs of the affected

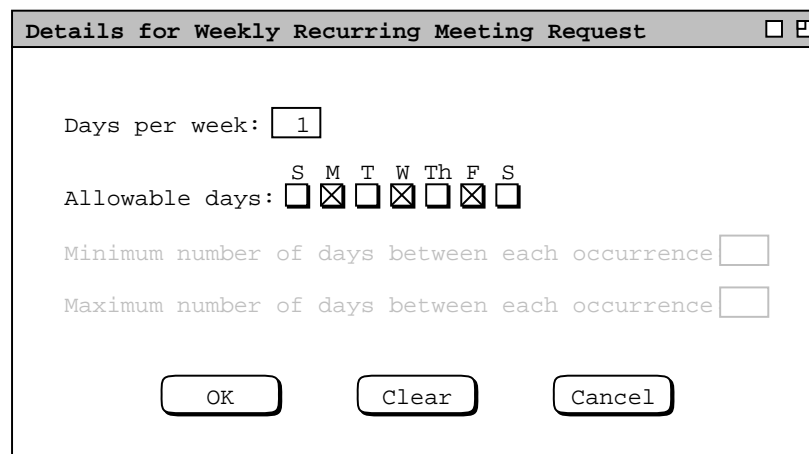


Figure 96: Weekly recurring details modified.

Meeting Scheduling Options

Maximum length of possible times list:

Schedule at same time as optional items? no yes

Schedule at same time as must items? no yes

Allowable days for non-recurring meetings: S M T W Th F S

Start on: full hour
 half hour
 quarter hour
 any time

OK Clear Cancel

Figure 97: Meeting scheduling options.

attendees.

The fourth scheduling option specifies the specific allowable days on which non-recurring meetings can be scheduled. The default days are Monday through Friday, inclusive. This option applies only to non-recurring meetings, since the ‘Details’ dialogs contain more specific settings for the allowable days of recurring meetings. Recurring details for weekly meetings are shown in Figure 93. Recurring details for other intervals are described in Section 2.5.3.4.

The final scheduling option indicates the time increment on which meetings can start. The default is for meetings to start on the full hour. The other selections are on the half hour, quarter hour, or at any time. For the ‘any time’ selection, the smallest possible increment is five minutes.

The scheduler proceeds to adjust the scheduling options as shown in Figure 98, where the second and third options are both changed to ‘yes’. Having made the desired option changes, the scheduler presses ‘OK’ in the options dialog and ‘List Times ...’ in the scheduling request dialog. This time the system responds with the dialog shown in Figure 99. The new list of possible times has two additional columns containing the user IDs of attendees with items scheduled at conflicting times. The columns appear in the list when the scheduler selects the ‘yes’ setting for the second and/or third scheduling option, as is done in Figure 98. A user is listed under ‘Optional Conflicts’ for each possible time where the user has an optional, non-private scheduled item that overlaps with the possible time. A user is listed under ‘Must Conflicts’ for each possible time where the user has a must, non-private scheduled item that overlaps with the possible time.

The default sorting order for a list with overlap columns is the same as without the columns -- first by start date and second by time, from earliest to latest. The user can click on either of the overlap column headings to change the default sorting order. For both columns, the primary order is from the smallest to largest number of users with overlapping times; the secondary, tertiary, and quaternary orders are date, time, and alphabetic by user ID. Clicking on the ‘Possible Times’ column label restores the default sorting order. The sorting order for the names in any conflicts row is alphabetic.

The ‘View User’ button at the bottom of the list allows the scheduler to view the daily calendar of any user listed in one of the overlap columns. To do so, the user selects one of the user IDs and then presses the ‘View User’ button, which becomes enabled upon selection of an ID. This command is discussed further in Section 2.4.1.3

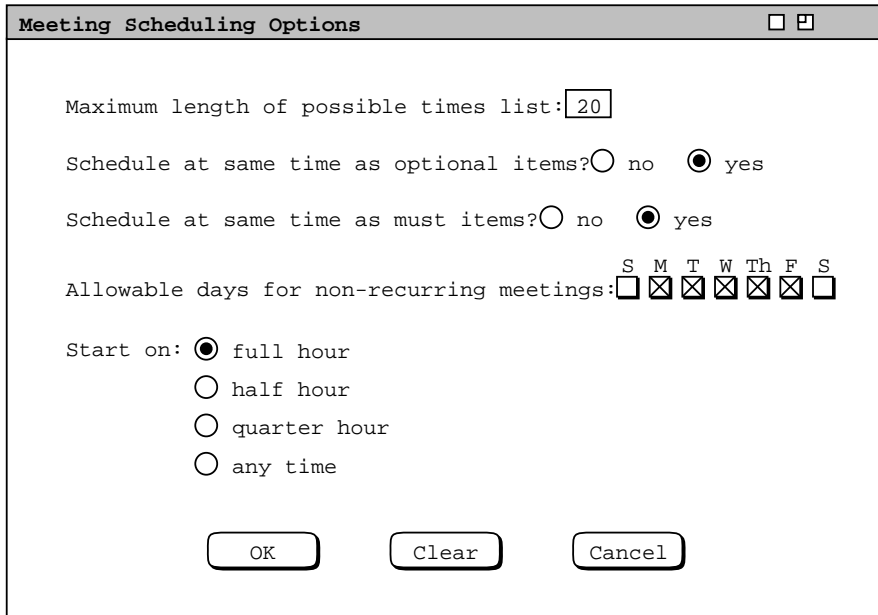


Figure 98: Modified meeting scheduling options.

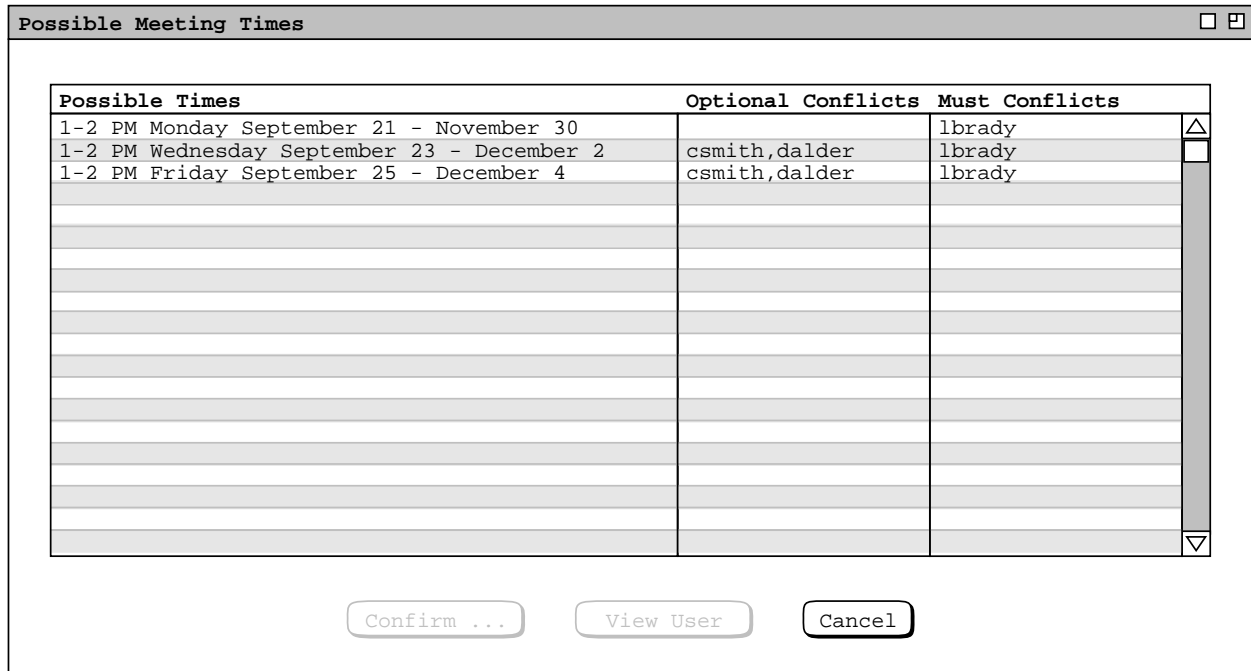


Figure 99: Possible meeting times with one or two faculty unable to attend.

For this scenario, the scheduler elects to choose the 1-2 PM Monday meeting time as originally planned, as shown in Figure 100. The scheduler presses 'Confirm ...' in the possible times list and 'OK' in subsequent confirmation dialog. The system then notifies all attendees, including those with overlapping items. The notification message for the attendees with overlapping items makes note of the condition. Details of attendee notification are covered in Section 2.4.1.5.

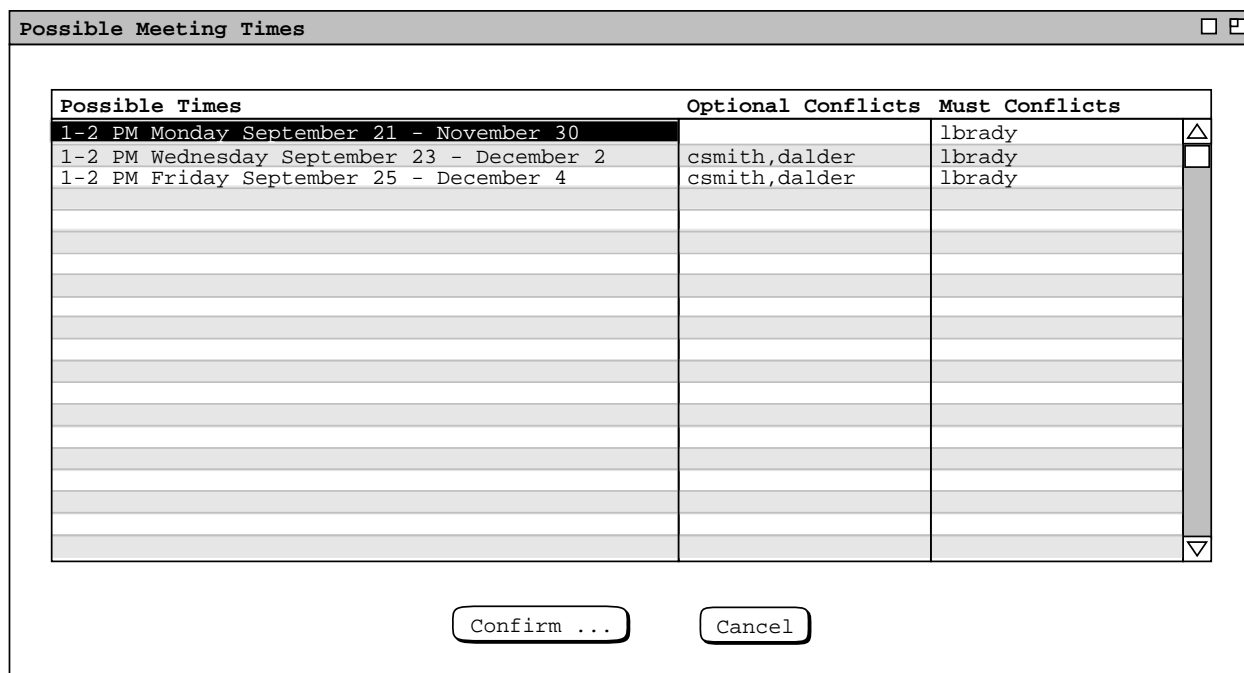


Figure 100: Selected faculty meeting time.

2.4.1.3. A Super-Group Leader Scheduling a Meeting

This section presents further details of meeting scheduling. Here the scheduler is the leader of a Calendar Tool *supergroup*, which is a group comprised of other groups.

Figure 101 shows a filled-in meeting request dialog for a computer science special colloquium; Figure 102 shows the options settings for this request. It is a non-recurring meeting, to be held on October 29 or 30, between 1:30 and 4 PM; the category is set to ‘CS Special’. The attendees are the ‘csall’ user group, and three additional individual users. The options allow the meeting to be scheduled at the same time as both optional and must items. The increment is changed from ‘full hour’ to ‘half hour’.

As defined in Section 2.6.3, *csall* is a Calendar Tool supergroup, comprised of the subgroups *csfaculty*, *cslect*, and *csstaff*. These groups are in turn comprised of individual Calendar Tool users. When a supergroup is listed among the attendees of a meeting, individual attendees are all members of all subgroups. The complete set of attendees is comprised of all group members plus all individual users appearing in the attendees list.

When the scheduler presses ‘List Times ...’ in Figure 101, the system responds with the possible times list shown in Figure 103. Given the large number of requested attendees, there are a large number of time conflicts listed in the conflicts columns. To determine the most acceptable time, the scheduler can use the ‘View User’ command to examine the calendars of one or more users listed in the conflicts columns. For example, Figure 104 shows the scheduler having selected user *mhutchen* for viewing. When the scheduler clicks anywhere in a user name, the system highlights the entire name and enables the ‘View User’ button. When the scheduler selects ‘View User’ in Figure 104, the system displays the daily agenda for *mhutchen* for Thursday October 29. Viewing such conflicting user calendars may assist the scheduler in choosing a meeting time that overlaps with relatively less important scheduled items.

After viewing some user calendar information, the scheduler decides to choose the 4-5:30 time slot on Thursday October 29, and confirm the meeting. In response, the system notifies all members of *csall* plus the three individually-listed attendees; the system adds the meeting to the *csall* group calendar, as well as to each subgroup calendar. The scheduling of meeting for a supergroup has a fully transitive effect on all sub(...sub)groups. That is, when a meeting is scheduled for a supergroup, it is also scheduled for all its sub(...sub)groups, including adding the

Schedule a Meeting □ □

Title:

Earliest Possible Date: Earliest Start Time:

Latest Possible Date: Latest Start Time:

Earliest End Date: Duration: ^{hr} ^{min}

Latest End Date:

Recurring? Interval:

Category: Security:

Location: Priority:

Remind?

Attendees:

Details:

Minutes:

Figure 101: Scheduling a special colloquium.

Meeting Scheduling Options

Maximum length of possible times list:

Schedule at same time as optional items? no yes

Schedule at same time as must items? no yes

Allowable days for non-recurring meetings: S M T W Th F S

Start on: full hour
 half hour
 quarter hour
 any time

Figure 102: Special colloquium options settings.

Possible Meeting Times

Possible Times	Optional Conflicts	Must Conflicts
1:30-3 PM Thursday October 29		arowland, chdana, jbc
2-3:30 PM Thursday October 29	lsimms	arowland, chdana, jbc
2:30-3 PM Thursday October 29		arowland, chdana, jpc
3-4:30 PM Thursday October 29		dbutler, mhutchen, rb
3:30-5 PM Thursday October 29		dbutler, mhutchen, rb
4-5:30 PM Thursday October 29	dbutler	mhutchen, rboche
1:30-3 PM Friday October 30		babrams, lmyers, pnic
2-3:30 PM Friday October 30		ekeller, hpaulson, lm
2:30-3 PM Friday October 30		ekeller, hpaulson, lm
3-4:30 PM Friday October 30		jconnely, pnic, rboc
3:30-5 PM Friday October 30		jconnely, pnic, rboc
4-5:30 PM Friday October 30	dbutler	jdalbey, mliu

Figure 103: Possible times for the special colloquium.

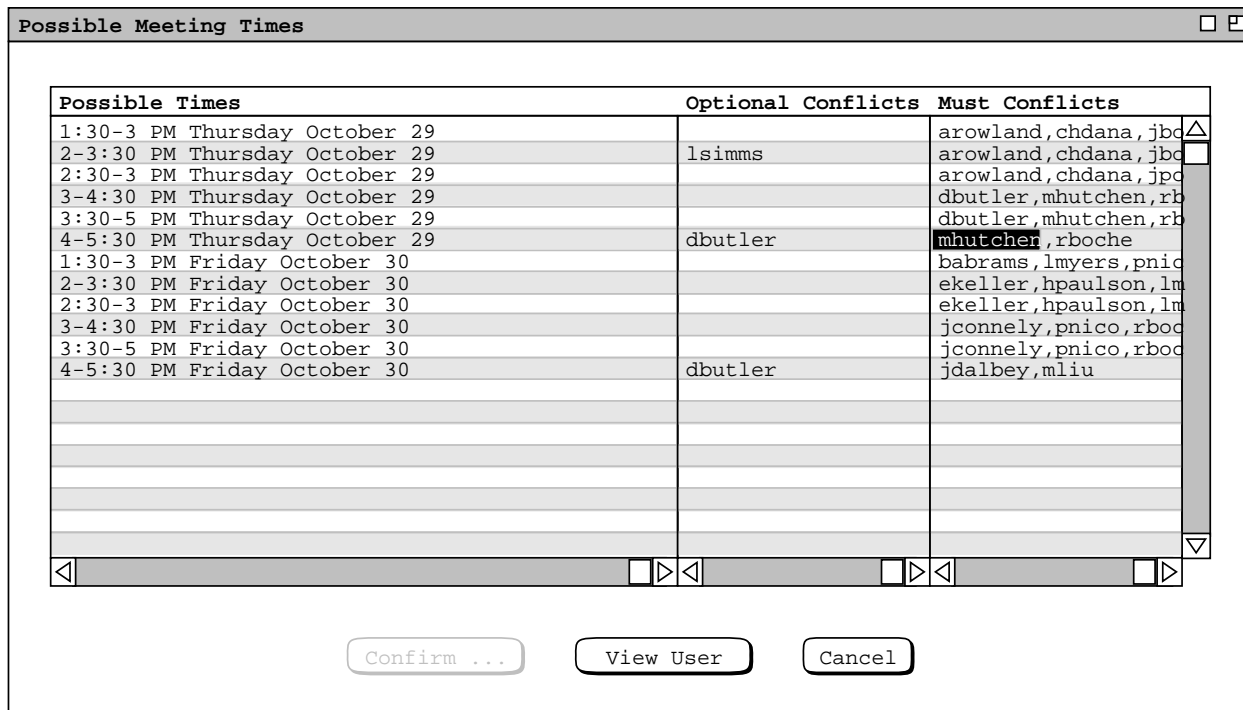


Figure 104: Selecting a conflicting user for the purpose of viewing that user’s calendar.

meeting to the sub(...sub)group calendars.

2.4.1.4. A Non-Leader Scheduling a Meeting for Selected Individuals

The final meeting scheduling scenario presents an individual user scheduling for a selected number of individual users. The attending users are not members of a group that could be used for the meeting at hand. The scheduler begins by entering the request information shown in Figure 105. The meeting is non-recurring, confidential security, for six individual attendees. In this scenario, the entered value for the meeting location is assumed not to be in the location database. This means that location has no effect on the possible times computation. The same would be true if the location field were empty.

When the scheduler selects ‘List Times ...’, the system responds with the no-times dialog of Figure 95. In this case, the scheduler deals with the situation by expanding the date and time range for the meeting, as shown in Figure 106. The scheduler also changes the last two scheduling options, as shown in Figure 107. Saturday is added to the allowable days and the ‘Start on’ interval is set to ‘any time’. When the scheduler again presses ‘List Times ...’, the system responds with the result in Figure 108. In this case, the scheduler has been a tad over-zealous in date/time and options settings, since the possibles list contains a large number of times. As might be typical among Calendar Tool users, many time slots on Saturday are available among the selected attendees. Hence combining a ‘Start on’ setting of ‘any time’ with Saturday as an allowable day results in a large number of possible times. Rather than adjust the date/time ranges or options further, the scheduler scrolls through the list of possible times to find one that is acceptable and selects it, as shown in Figure 109. The scheduler then proceeds to confirm the meeting.

Of note in Figures 108 and 109 is how the possible times list contains all allowable times, including ones that overlap such as 8-9 AM and 8:05-9:05 AM. The list contains all distinct meeting times that individually conform to the scheduling request values and option settings. Two possible times are *distinct* if they differ in at least one value of start time, end time, or date.

Schedule a Meeting [] [x]

Title:

Earliest Possible Date: Earliest Start Time:

Latest Possible Date: Latest Start Time:

Earliest End Date: Duration: ^{hr} ^{min}

Latest End Date:

Recurring? Interval:

Category: Security:

Location: Priority:

Remind?

Attendees:
 [] [x] [v]

Details:
 [] [x] [v]

Minutes:

Figure 105: Scheduling a tenure review meeting.

Schedule a Meeting [] []

Title:

Earliest Possible Date: Earliest Start Time:

Latest Possible Date: Latest Start Time:

Earliest End Date:

Duration: hr min

Latest End Date:

Recurring? Interval:

Category: Security:

Location: Priority:

Remind? hours before

Attendees:

Details:

Minutes:

Figure 106: Modified tenure review meeting request.

Meeting Scheduling Options

Maximum length of possible times list:

Schedule at same time as optional items? no yes

Schedule at same time as must items? no yes

Allowable days for non-recurring meetings: S M T W Th F S

Start on: full hour
 half hour
 quarter hour
 any time

Figure 107: Tenure review meeting options.

Possible Meeting Times

8-9 AM Saturday September 19	▲
8:05-9:05 AM Saturday September 19	
8:10-9:10 AM Saturday September 19	
8:15-9:15 AM Saturday September 19	
8:20-9:20 AM Saturday September 19	
8:25-9:25 AM Saturday September 19	
8:30-9:30 AM Saturday September 19	
8:35-9:35 AM Saturday September 19	
8:40-9:40 AM Saturday September 19	
8:45-9:45 AM Saturday September 19	
8:50-9:50 AM Saturday September 19	
8:55-9:55 AM Saturday September 19	
9-10 AM Saturday September 19	
9:05-10:05 AM Saturday September 19	
9:10-10:10 AM Saturday September 19	
9:15-10:15 AM Saturday September 19	
9:20-10:20 AM Saturday September 19	
9:25-10:25 AM Saturday September 19	
9:30-10:30 AM Saturday September 19	
9:35-10:35 AM Saturday September 19	▼

Figure 108: Very long possible times list.

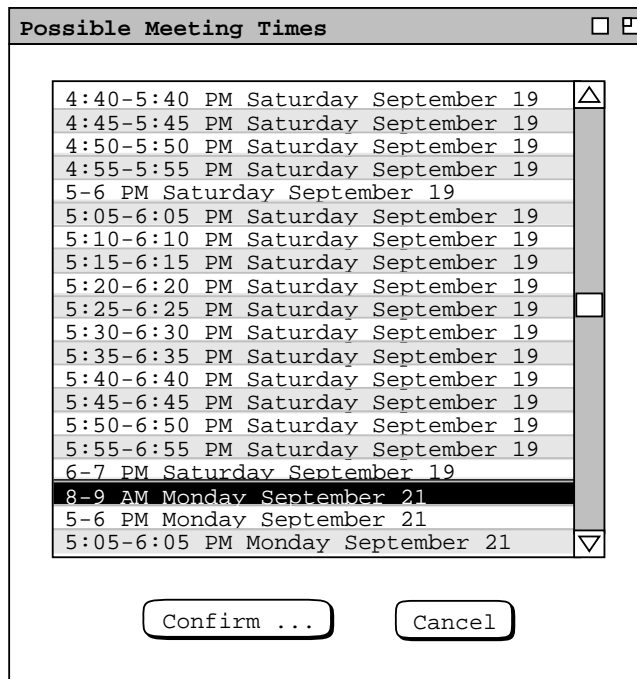


Figure 109: Long possible times list scrolled to an acceptable time.

2.4.1.5. Receiving and Accepting Meeting Notifications

When a meeting scheduler confirms a meeting, the system notifies each attendee, except the scheduler, with a dialog of the form shown in Figure 110. This is a notification dialog for the non-recurring staff meeting as scheduled in Section 2.4.1.1. The message indicates the scheduler's ID plus the time and date of the meeting. To notify a user of a recurring meeting, the system displays a dialog of the form shown in Figure 111. The message indicates the scheduler's ID plus the time and date of the first occurrence of the meeting.

When a meeting overlaps with one or more existing scheduled items in the user's calendar, the notification message has the following line appended:

NOTE: this meeting conflicts with *N* existing item(s).

where *N* is the number of scheduled items that have overlapping times with the requested meeting and "item(s)" is singular if *N* = 1, plural otherwise.

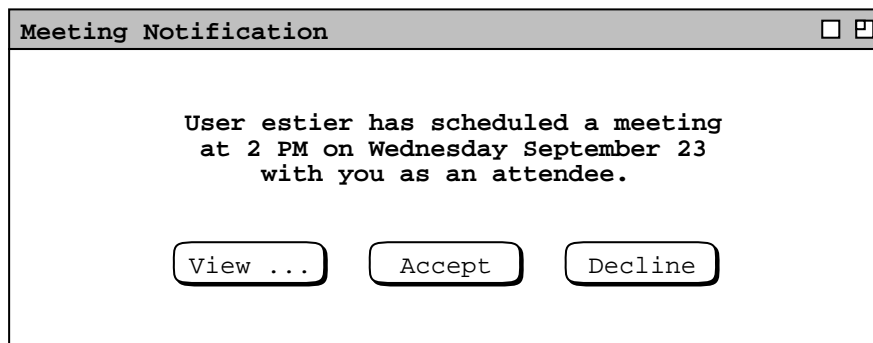


Figure 110: Meeting notification dialog.

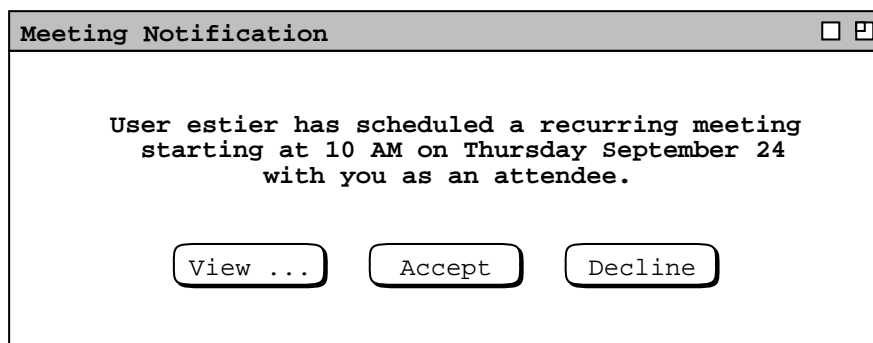


Figure 111: Recurring meeting notification dialog.

The user can view the complete details of the meeting by pressing the ‘View ...’ button in the initial notification dialog. For example, if the user presses ‘View ...’ in Figure 110, the system responds with the meeting information shown in Figure 112. This display has the same information as in the meeting confirmation dialog accepted by the scheduler (Figure 88). The user can edit any of the data fields in the notification dialog except for ‘Scheduled By’, ‘On’, and ‘Host’. Complete details of meeting editing are presented in Section 2.5.2.5

The user can respond to a meeting notification in one of three ways: accept the meeting, decline the meeting, or delay a decision. Both the initial meeting notification dialog (Figure 110) and the larger details dialog (Figure 112) have ‘Accept’ and ‘Decline’ buttons. If the user chooses ‘Accept’, the meeting is added to the user’s local calendar as a scheduled item. If the user chooses ‘Decline’, the meeting is not added to the user’s calendar.

The precise effects of accepting a meeting are the following:

- a. The system removes the not-yet-accepted marking from the meeting in the user’s central host calendar.
- b. The system copies the meeting to the user’s local calendar.
- c. The system removes the notification dialog from the screen.

The precise effects of declining a meeting are:

- a. The system deletes the not-yet-accepted meeting from user’s the central host calendar, without changing the user’s local calendar
- b. The system removes the notification dialog from the screen.

To delay the accept/decline decision, the user can close the notification dialog window, using the ‘View Windows Close’ command. A user cannot delay an accept/decline decision beyond the date and start time of a non-recurring meeting, or beyond the date and start time of the last occurrence of a recurring meeting. If a user fails to accept or decline within this time, the system automatically executes decline on the user’s behalf. As a result, the declined item is removed from all affected displays, including the complete removal of any item-level view for the declined item.

In the case of Figure 112, the notified user decides to edit the ‘Title’ and ‘Category’ fields, and turn off the reminder, as shown in Figure 113. The user then presses ‘Accept’, whereupon the system adds the meeting to the user’s local calendar as a regular scheduled item. To clear any edits made after the confirmation dialog is initially displayed, the user can press ‘Clear’. As illustrated in Figures 112 and 113 the ‘Clear’ button is disabled until the user performs some editing.

When the detailed notification dialog is displayed, the initial smaller notification dialog remains active. The user can press ‘Accept’ or ‘Decline’ in either dialog. When the user does so, the active dialog is removed from the screen, and the other dialog, if displayed, is also removed.

The item for an accepted meeting is stored as an individual copy in each accepting user’s local calendar. In this way, any changes that a user makes to an accepted meeting are local to that user and do not affect any of the other attendees. Changes made by the meeting scheduler can be announced to all attendees, if the scheduler chooses to do so. Further details of changing and deleting meetings are covered in Sections 2.5.2.2 and 2.5.2.5.

Meeting Notification Detailed View

Title:

Date: Start Time:

End Date: Duration: hr min

Recurring? Interval: S M T W Th F S

Category: Security:

Location: Priority:

Remind? minutes before

Attendees:

Details:

Agenda:

1. Duty assignments for Fall quarter
2. News of office remodeling project
3. Introduce new staff members
4. Promotions
5. Initial prep for accreditation visits

Minutes:

Scheduled By: On:

Host:

Send email notification?

Figure 112: Viewing details of a meeting notification.

Meeting Notification Detailed View
□ □

Title:

Date: Start Time:

End Date: Duration: hr min

Recurring? Interval: S M T W Th F S
□ □ □ □ □ □ □

Category: Security:

Location: Priority:

Remind?

Attendees:
 ▲
□
▼

Details:
 ▲
□
▼

Minutes:

Scheduled By: On:

Host:

Send email notification?

Figure 113: Notifi cation details dialog edited.

Prior to an accept/decline decision, a not-yet-accepted meeting appears on a user’s calendar in *penciled-in* form. In terms of screen display, a penciled-in meeting appears in light-grey italic type in daily, weekly, and monthly calendar views, as well as in all list views. For example, Figure 114 shows a daily agenda where two meetings have been confirmed by the scheduler, but not yet accepted by the user. The meetings are in the 8 AM and 2:30 PM time slots. Once a penciled-in meeting is accepted, it appears normally in all calendar and list views. For example, if the user selects the ‘View Day’ menu command after accepting the staff meeting in Figure 113, the system displays the daily agenda shown in Figure 115. This figure shows an updated version of the daily agenda from Figure 114 where the previously penciled-in staff meeting is now shown as a regularly scheduled item, with its edited title and new category color.

In item-level calendar views, not-yet-accepted meetings are indicated by different command buttons at the bottom of the display. Details of this are presented in Section 2.5.1.2.

Daily Agenda	
< Today > Wednesday, September 23, 1998	
8 AM	<i>Computer Science Staff Meeting</i>
9 AM	Office Hours
10 AM	Data Structures Lecture
11 AM	Data Structures Lab
12 PM	Lunch with Microsoft
1 PM
2 PM
3 PM	<i>CAD Research Project Meeting</i>
4 PM
5 PM	

Figure 114: Two meetings displayed in penciled-in form.

Daily Agenda	
< Today > Wednesday, September 23, 1998	
8 AM	Staff Meeting
9 AM	Office Hours
10 AM	Data Structures Lecture
11 AM	Data Structures Lab
12 PM	Lunch with Microsoft
1 PM
2 PM
3 PM	<i>CAD Research Project Meeting</i>
4 PM
5 PM	

Figure 115: Accepted meeting changed from penciled in form.

From the perspective of a meeting's scheduler, notification is sent to the attendees immediately upon confirmation of the meeting. At any point between scheduler confirmation and attendee acceptance, the meeting appears in penciled-in form on the calendars of all attendees who permit penciling in.

From the perspective of an attending user, the appearance of notification dialogs and not-yet-accepted items is controlled with two option settings. The option for when to show notification dialogs has one of the following values:

- 'any time' -- meeting notifications appear any time the user's Calendar Tool local host computer is running, whether or not the Calendar Tool program is running; the definition of a user's local host computer is discussed in Section 2.6.2; 'any time' is the default setting for the option
- 'during Calendar Tool execution' -- notifications appear any time the Calendar Tool is running on the user's local computer
- 'at start-up only' -- notifications appear only when the user initially starts up the Calendar Tool from the external operating environment
- 'at exit only' -- notifications appear only when the user exits from the Calendar Tool program
- 'never' -- meeting notification dialogs never appear; in this case, the only indication of a meeting having been scheduled is its appearance in penciled-in form in a calendar or list view

When the 'any time' setting is on and a notification is received while the Calendar Tool program is not running, receipt of the notification causes the Calendar Tool to be started, after which the notification dialog is displayed. If the user accepts the meeting, the system opens the calendar that is associated with the host from which the notification originates, as described in Section 2.4.1.6.7. Once the calendar is open, the system proceeds to process the meeting acceptance as described above.

The system also provides an option to enable or disable the appearance of not-yet-accepted items. The option is a show/hide toggle, with the default setting of 'show'. When the 'show' setting is selected, all not-yet-accepted meetings are shown in penciled-in form in daily, weekly, and monthly calendar views, as well as in all list views. At the item level, not-yet-accepted meetings appear with different command buttons, as described in Section 2.5.1.2. The system dynamically updates all active displays that are affected by a meeting being scheduled. An "affected" display is one in which the penciled-in item should appear. Details of dynamic view updating are covered in Section 2.5.6. When the 'hide' setting is selected, not-yet-accepted meetings do not appear at all in calendar or list views. This includes item-level views, which do not appear in next/previous item traversals (Section 2.3.2.1.) when not-yet-accepted meetings are hidden.

Given the two preceding options, it is possible for a user to disable entirely any form of meeting notification. Viz., if the user chooses the 'never' setting for notification dialogs and the 'hide' setting for not-yet-accepted items, meetings scheduled by other users are invisible via the Calendar Tool. The prudence of this combination of option settings is questionable, but left to users' discretion.

The system displays the notification dialog for any given meeting at most one time. The notification dialog appears as soon as the state of the user's local host computer and the setting of the notification option allow. After the user makes the accept, decline, or delay decision, the notification dialog is never redisplayed. If the start date and time of the meeting pass before the dialog is allowed to appear, the system never displays it.

To be clear, the notification options control only the *visibility* of not-yet-accepted items, not their existence on the central host computer. Whenever a scheduler confirms a meeting, a copy of the meeting is added to the central host calendars of all attendees. The only way the user can delete the central host copy of a meeting is to accept meeting and then delete it, or decline the meeting.

When viewing another user's calendar, not-yet-accepted meetings *always* appear in penciled-in form, independent of the current user's or other user's setting of the show/hide penciled-in option. Hence, a user can hide penciled-in items for her or his own calendar, but not for the calendars of other users. Hiding penciled-in items also has no affect on meeting scheduling. For the purposes of meeting scheduling, all not-yet-accepted meetings are considered scheduled, as discussed further in Section 2.4.1.6.3.

A meeting scheduler may optionally elect to send email notification to all attendees. Email notification is sent in addition to the standard dialog form of notification. The body of an email notification contains the text that appears in the initial notification dialog, plus an additional explanatory sentence. The user ID that appears in the notification

dialog is replaced with the user's full name, as defined in the Calendar Tool user database (Section 2.6.2). The sender of the message is the meeting's scheduler. The message subject is "Meeting notification". For example, had the scheduler chosen to send email notification for the meeting notified in Figure 110, the email message sent to user `dcollins` would appear as shown in Figure 116. The sender's email address and the email addresses of the attendees are those defined in the Calendar Tool user database. The body of the message is left justified in 72 character columns.

The Calendar Tool sends email notifications using the electronic mail facilities of the underlying operating environment. Once an email notification has been sent, its further handling and delivery are beyond the control of the Calendar Tool. The option for when to show meeting notifications has no effect on email notifications. Email meeting notifications are sent unconditionally when the meeting scheduler selects the email notification option. To disable receipt of email meeting notifications, a Calendar Tool user may use commands of the email handling system of the underlying operating environment, if such commands are available. Use of such commands is beyond the control of the Calendar Tool.

2.4.1.6. Further Operational Details of Meeting Scheduling

A meeting differs from the three other types of scheduled item in that meetings involve multiple users. The multi-user aspects of scheduled meetings give rise to operational details that must be carefully considered.

2.4.1.6.1. The Contents of Group Calendars

A group calendar contains scheduled meetings in which the group is among the attendees. A group is considered to be among meeting attendees in the following cases:

- a. the group ID appears in the attendees list
- b. the ID of a supergroup of which the group is a member appears in the attendees list

If the group ID is not listed directly or by subgroup inclusion in the attendees list, then the group is never considered an attendee, even if all members of the group are listed individually as attendees,

As noted above, only a group leader can schedule a meeting for a group. The group calendar is not itself editable. Any changes made to a meeting in the scheduler's calendar are automatically reflected in the group calendar for that meeting. Section 2.5.2.2 presents complete details of changing and deleting scheduled group meetings.

A group calendar has its own category definitions. The set of group calendar categories is comprised of the categories that appear in the scheduled meetings within the calendar. Category changes made by group meeting schedulers are reflected in the group calendar.

Group calendars exist only for explicitly defined groups in the Calendar Tool group database, as defined in Section 2.6.3. There are no group calendars for meetings with lists of individual users, such as the meeting scheduled in Section 2.4.1.4. In terms of physical storage, a group calendar is stored only on the central host computer where the group is administratively defined. Group calendars are not copied to any local user computers.

```
From: estier@calpoly.edu
To: dcollins@calpoly.edu
Subject: Meeting notification
```

```
Ellen M. Stier has scheduled a meeting at 2 PM on Monday September 21
with you as an attendee. Please use the Calendar Tool to accept or
decline this meeting.
```

Figure 116: Content of email meeting notification.

2.4.1.6.2. Scheduling Over Private Items

Scheduled items with 'private' security are by definition invisible to all other users, where "other" is a user different than the user on who's calendar the private item is scheduled. For the purposes of meeting scheduling, the scheduler is always considered an other user, even if the scheduler is listed among the attendees. Therefore, all private items are entirely invisible when the list of possible times is computed, including private items on the scheduler's own calendar. This means that the times during which any private item is scheduled are considered available without conflict for meeting scheduling.

2.4.1.6.3. The Status of Not-Yet-Accepted Meetings

As explained above, not-yet-accepted meetings are those that have been confirmed by the scheduler but not yet accepted by an attendee. These are the items that appear in penciled-in form on the calendars of users who have not yet accepted. For the purposes of possible times computation, a not-yet-accepted meeting is considered a scheduled item. This means that the times of not-yet-accepted meetings are considered UNavailable for meeting scheduling. An individual user's setting of the show/hide option for penciled-in items has no affect on the status of the items for scheduling purposes. Specifically, if a user chooses to hide penciled-in items, their times are still considered unavailable for meeting scheduling.

2.4.1.6.4. Creation of New Categories

A scheduler may specify a category when scheduling a meeting. That category must be defined in the scheduler's calendar, but may not be defined in one or more attendees' calendars. When a notified user accepts a meeting with a category that is undefined in the accepting user's calendar, the system prompts the user with a dialog of the form shown in Figure 117. This figure corresponds to the special colloquium meeting shown in Figure 101 for an attendee on whose calendar 'CS Special' is undefined. A meeting category is considered undefined in an attending user's calendar if there is no category of same name and color in the user's calendar. Comparison of category names is made on a case-sensitive, character-by-character basis.

As shown in Figure 113, an accepting user can edit the meeting category prior to accepting it. This means that a user can change the category assigned by the scheduler to empty or a category already defined in the user's calendar. If such a change is made, the system does not display a category definition dialog of the form shown in Figure 117.

2.4.1.6.5. The Role of the Location Database

The role of the location database is strictly limited in scheduling a meeting. Specifically, when the scheduler enters a 'Location' value that is in the location database, the system excludes from the possibles list any times during which that location is unavailable.

The meeting operation performs no "room finding" computation, as in finding a room that is available and of a proper size for a meeting. When the 'Location' field is empty, the system assumes that the meeting can be held at

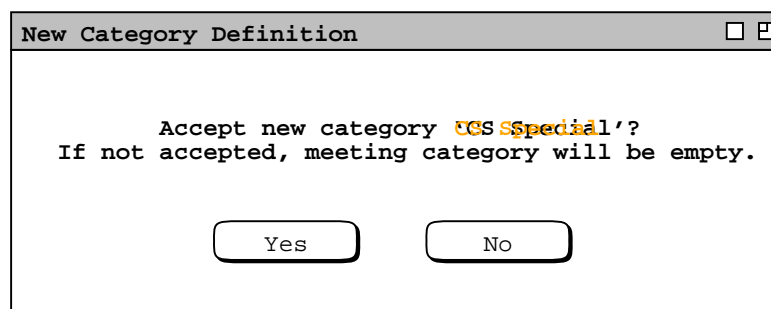


Figure 117: Prompting dialog for new category definition.

any location. In this case, it is the responsibility of the scheduler to inform attendees of the location by some means external to the Calendar Tool. When the 'Location' value is non-empty but not in the location database, the system assumes that the location is available at any time. In either of the latter two cases (empty location or non-database location), the location database has no role at all during the possible times computation.

2.4.1.6.6. Non-Unique Scheduled Meetings

As discussed in Section 2.11.6, the Calendar Tool disallows multiple scheduled items with exactly the same values for date, start time, duration, and title. This uniqueness condition can be violated for one or more attendees when a scheduler allows a meeting to be scheduled at the same time as optional or must items, as in Figure 102, for example. The uniqueness condition can also be violated for private items on attendees' calendars.

The uniqueness condition is not checked at the time the scheduler confirms the meeting, but rather at the time an attending user accepts the meeting. Specifically, if an attending user accepts a meeting that violates the uniqueness condition, the system prompts the user with a dialog of the form shown in Figure 118. As the dialog explains, the user presses 'Retitle' to have the system change the title by adding a suffix. The suffix added is the string "[n]", where *n* is a unique integer starting at 2 and incremented as necessary to guarantee unique titles. If the user presses 'Edit', the system (re)displays the detailed notification dialog, in which the user can edit the title. The user could satisfy the uniqueness condition by editing the date, start time, or duration instead of the title. Title editing is presumably the most typical.

For either 'Retitle' or 'Edit' selection, the system removes the non-unique warning dialog from the screen. In the case of 'Edit', the user is returned to the state where the meeting can be accepted, declined, or delayed. If the user presses 'Accept' without editing the title (or other field) to meet the uniqueness condition, the system redisplay the non-unique warning dialog.

Prior to acceptance, penciled-in meetings that violate the item uniqueness condition overlap with one or more extant calendar items. As described in Sections Section 2.3.1 and 2.3.2, overlapping items must be ordered for the purposes of display and next/previous traversal. For items that meet the uniqueness condition, unambiguous ordering is based on start time, duration, and title. In the case of non-unique penciled in items, the ordering must be determined using additional fields of the meeting. Specifically, the ordering of non-unique penciled in items is defined as follows:

- a. a penciled-in item with the same start time, duration, and title as an extant item goes after the extant item
- b. in the presumably rare case of two or more penciled-in items with the same start time, duration, and title, the ordering of these items is based first on the 'On' value, second on 'Scheduled By', and third on 'Host'.

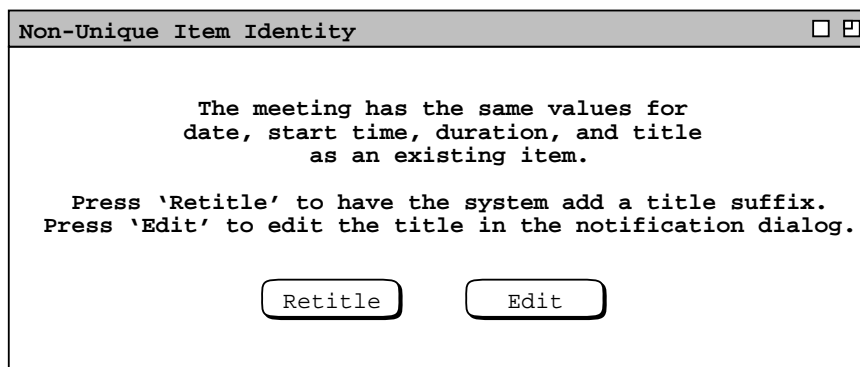


Figure 118: Accepted meeting with non-unique identity.

2.4.1.6.7. Notifications for Multiple Calendars from Multiple Central Hosts

As described in Section 2.6.6.2, the user can be connected to more than one central host with more than one calendar. The association of calendar to central host is unique. Specifically, a calendar is associated with at most one central host. This means that meeting notifications from a particular host are always directed to one specific calendar. That a notification is "directed to" a calendar means that when the user accepts a notification, the accepted meeting is added to the calendar that is associated with the host from which the meeting originates. Specifically, when the user accepts a meeting that originates from host X, the system performs the following actions:

- a. opens the calendar associated with host X, if the calendar is not open
- b. makes the calendar associated with host X current, per the calendar currency rules specified in Section 2.3.6.4.
- c. completes the accepting actions described in Section 2.4.1.5.

As explained in Section 2.7.3.3, the options for accepting meeting notifications are calendar-specific. This means that a user sees or does not see notifications based on the option settings for each calendar individually.

2.4.1.6.8. Pending Notifications

Depending on the state of an attendee's Calendar Tool computer and the setting of notification options, it is possible for meeting notifications to be sent by a scheduler at one time and seen by an attendee at a later time. The system maintains a queue of pending meeting notifications for each requested attendee. Notifications are maintained in the queue in confirmation-time order, from earliest to latest confirmation time. Confirmation time is defined as the time at which the meeting scheduler presses the 'OK' button in the meeting confirmation dialog.

As soon as conditions arise for a user to see meeting notifications, the system displays one dialog for each entry in the user's notification queue. The dialogs are displayed successively in queue order, with the system waiting for the accept/decline/delay decision between each display. For example, if a user's notification queue contains four entries, the system successively displays four separate notification dialogs. When the user makes the accept/decline/delay decision in one notification dialog, the system displays a dialog for the next entry in the notification queue.

As explained in Section 2.11.26, most Calendar Tool dialogs are "non-modal", including meeting notification dialogs. This means that the user is not required to respond to meeting notifications prior to performing other Calendar Tool operations. Section 2.11.26 discusses this issue in detail.

By default, the system removes pending notifications from users' queues after the time for the notified meeting has past. The user can change this default behavior with an option setting.

Notifications are not queued when a user selects the 'never' setting for the option that controls when to show notifications. During the period that the 'never' setting is in force, notifications are discarded. If the user sets the option to some value other than 'never', notifications are displayed or queued subsequently. Hence, a user cannot "catch up" on missed notifications that were sent while the notification option was set to 'never'.

Since notification options are calendar-specific, pending notifications are therefore calendar-specific. As explained in the preceding section, all notifications are associated with a specific calendar. Given this, notifications are entered into and removed out of the pending queue based on the option settings for each calendar individually. For example, suppose the user has three calendars with notification options set to 'any time', 'at start-up', and 'never' respectively. Notifications for the 'any time' calendar are only queued when the user's local host computer is not running; as soon as the computer comes up, the notifications are displayed. For the 'at start-up' calendar, notifications are queued between each invocation of the Calendar Tool, and displayed immediately upon invocation. Notifications for the 'never' calendar are never queued and never displayed.

2.4.1.6.9. Concurrent Scheduling

It is possible for two or more Calendar Tool users to execute the 'Schedule Meeting' command concurrently. In such circumstances, the system imposes synchronization restrictions in order to ensure consistent scheduling behavior. Specifically, two or more users may perform scheduling concurrently up to the point of the first 'List Times' command in a scheduling request dialog (such as Figure 83). Upon execution of 'List Times' by the

first user to do so, all other users are blocked from executing 'List Times' until the first user confirms a meeting or cancels both the confirmation and possible times dialogs (such as Figures 85 and 87). During this period of blockage, when a second user executes 'List Times', the system displays a dialog of the form shown in Figure 119. As the dialog explains, the user is informed that another user is currently in the process of scheduling. If the second user presses 'Wait', the system adds the user to a waiting queue. When the user presses 'Wait', that user may perform other Calendar Tool operations while waiting. When a waiting user is allowed to proceed, the system notifies the user as discussed below. If the user presses 'Cancel' in the blocking dialog, the scheduling operation is canceled entirely, as if the second user had pressed 'Cancel' in the scheduling request dialog.

When a third user and beyond execute 'List Times', the system continues to respond with the blocking dialog, incrementing the number of users in the waiting queue each time an additional user presses 'Wait'. For example, when a fourth user beyond *estier* executes 'List Times', the system displays the dialog in Figure 120. The number of users shown in the queue does not include the currently active scheduler, e.g., *estier* in this case.

If a user fails to take action in the blocking dialog, other users may proceed ahead of the non-acting user. As this occurs, the system dynamically increments the number of users in the queue while the blocking dialog remains on the screen. As other users complete their operation, the system decrements the queue length. When all other users complete their work, the blocking dialog is removed from the screen and the 'List Times' operation for the non-acting user proceeds. In this way, a user who takes no action in the blocking dialog will be able to proceed after all other users have completed scheduling. After a user presses 'Continue' in the blocking dialog, the user can remove herself from the waiting queue by pressing 'Cancel' in the suspended scheduling request dialog.

When the active scheduler confirms or cancels the meeting operation, the system removes the first user from the waiting queue. If this user is the only one waiting, then the system proceeds with the 'List Times' operation that was originally blocked for the waiting user. Proceeding with the 'List Times' operation means that the system displays the possible times list as the top-most display window on the screen, interrupting any Calendar Tool operation the waiting user may have been performing. The system issues an audible alert as an additional cue that the

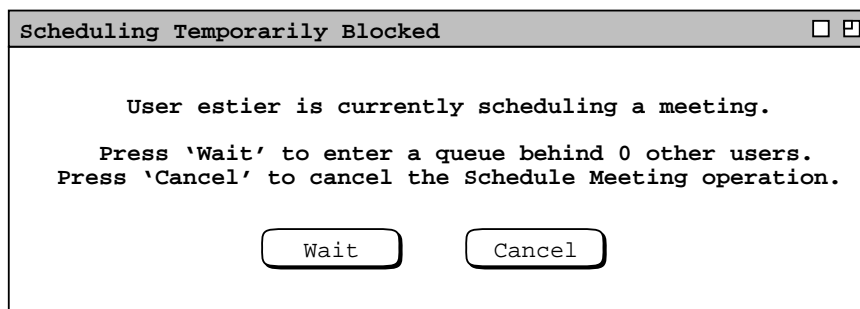


Figure 119: Scheduling temporarily blocked.

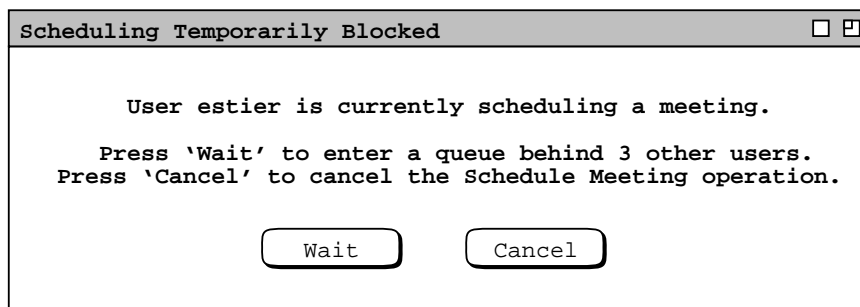


Figure 120: Fourth blocked user.

'List Times' operation may proceed.

If there are two or more users waiting in the queue, the system displays a dialog of the form shown in Figure 121, accompanied with an audible alert. As the dialog explains, the user has 30 seconds to respond before being put back at the end of the waiting queue. If the user does respond within 30 seconds, the system proceeds with the blocked 'List Times' operation for that user, and that user thereby becomes the active scheduler.

To promote fair service among users, the system monitors the amount of time a scheduler takes to complete or cancel a meeting scheduling operation while other users are waiting. The default time-to-complete increment is two minutes, which value can be set as an administrative option. If the waiting queue is non-empty, a completion timer starts when the possible times dialog is displayed for the current scheduler. If the waiting queue is empty when the possible times dialog is displayed, the completion timer starts when another user enters the waiting queue.

When a scheduler has spent two minutes in the scheduling operation while one or more users are waiting, the system displays a time warning dialog of the form shown in Figure 122. As the dialog explains, the scheduler is offered two additional minutes to complete the scheduling operation. If the scheduler presses 'Continue', the system removes the warning dialog from the screen, allowing the scheduler to continue with meeting scheduling. The system continues to redisplay the dialog at two-minute intervals until the scheduler completes or cancels the scheduling operation, or fails to respond within fifteen seconds to the warning. Cancellation of the scheduling operation, either by explicit button press or 15-second timer expiration, is equivalent to the user pressing the 'Cancel' button in the scheduling request dialog.

There is no limit to the number of times that a scheduler may respond with 'Continue' in the time warning dialog. In theory, this allows a user to block indefinitely all other users from scheduling. Such a circumstance is presumably

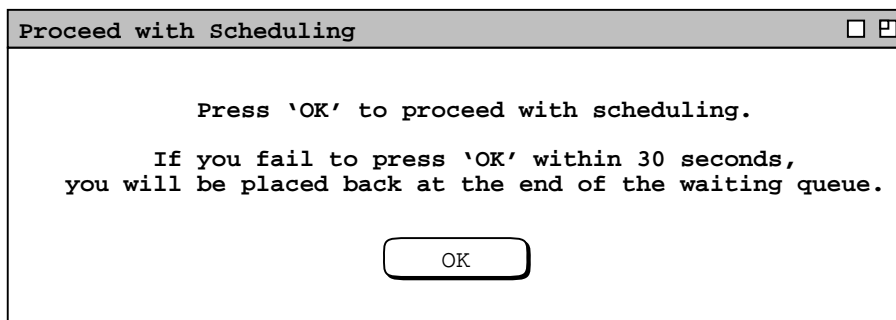


Figure 121: Notification to proceed with scheduling.

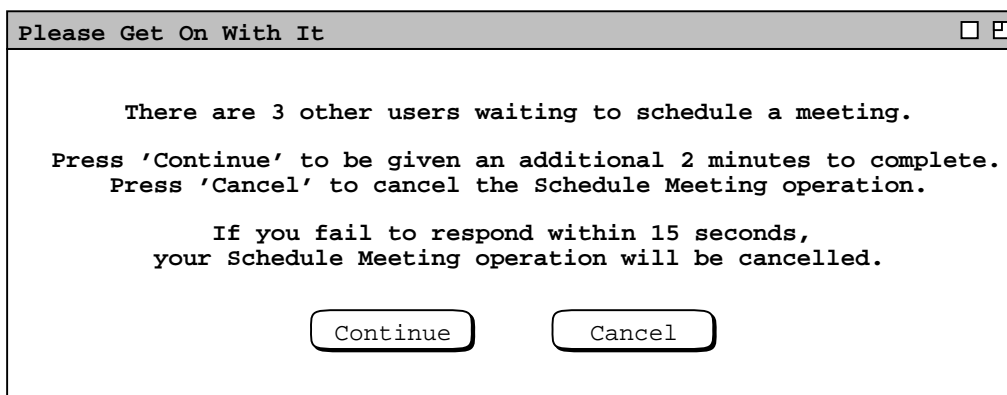


Figure 122: Scheduling time warning.

unlikely and must be remedied by means outside of the Calendar Tool system.

2.4.2. Scheduling Tasks

A task is a "to-do" item in a calendar. To schedule a task, the user selects the 'Task' command in the 'Schedule' menu, in response to which the system displays the dialog shown in Figure 123.

The following fields in the task scheduling dialog hold the same form of information as the appointment scheduling dialog in Figure 6:

- Title
- Recurring, and its accompanying fields
- Category
- Remind, and its accompanying fields
- Details

The user enters information for these fields in the same manner as for an appointment, as described in Section 2.2.

The date and time fields of a task specify when the task is due. A task has no specific start time or duration. The 'End Date' field applies to recurring tasks; it is initially disabled since the default for tasks is non-recurring. The 'Security' for a task is one of two levels: 'Public' or 'Private'. The 'TitleOnly' and 'Confidential' security levels do not apply to tasks, since these levels are only meaningful for scheduled items that have a start time and duration, i.e., appointments and meetings.

The 'Priority' for a task is fundamentally different than for the other types of scheduled item. For a task, priority is a numeric value that defines the relative importance among tasks. The value is between 0 and 10, with a higher value indicating a higher priority. Figure 124 shows the task priority pulldown menu. For documentation purposes, the menu lists parenthetically that 0 is the lowest and 10 the highest priority. These documentary words do not

Figure 123: Task scheduling dialog.

appear in the data field itself for the 0 and 10 values.

Figures 125 and 126 show two filled-in task scheduling dialogs for tasks that have appeared in the figures of earlier scenarios. In the recurring task, the 'Due Date' label is changed to 'Start Date' to clarify the display. As with the appointment scheduling dialog, the user presses the 'OK' button to add the task to her calendar, 'Clear' to clear all data-entry values to the defaults, and 'Cancel' to cancel the scheduling operation. The system removes the dialog window from the screen when the user presses 'OK' or 'Cancel'.

2.4.3. Scheduling Events

An event is the simplest form of scheduled item in a calendar. When the user selects Event in the Schedule menu, the system displays the dialog shown in Figure 127. The Title and Category fields contain the same information as in an appointment, meeting, and task. Events have a start date and an optional end date, but no time. Since

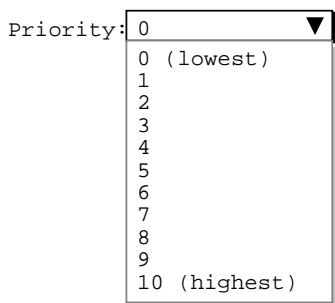


Figure 124: Task priority menu.

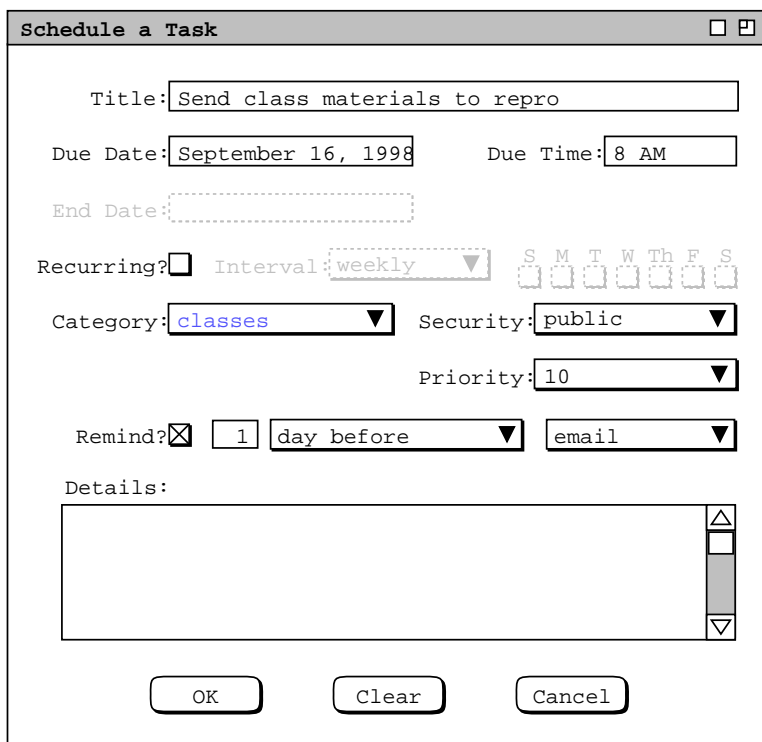


Figure 125: Scheduling a non-recurring task.

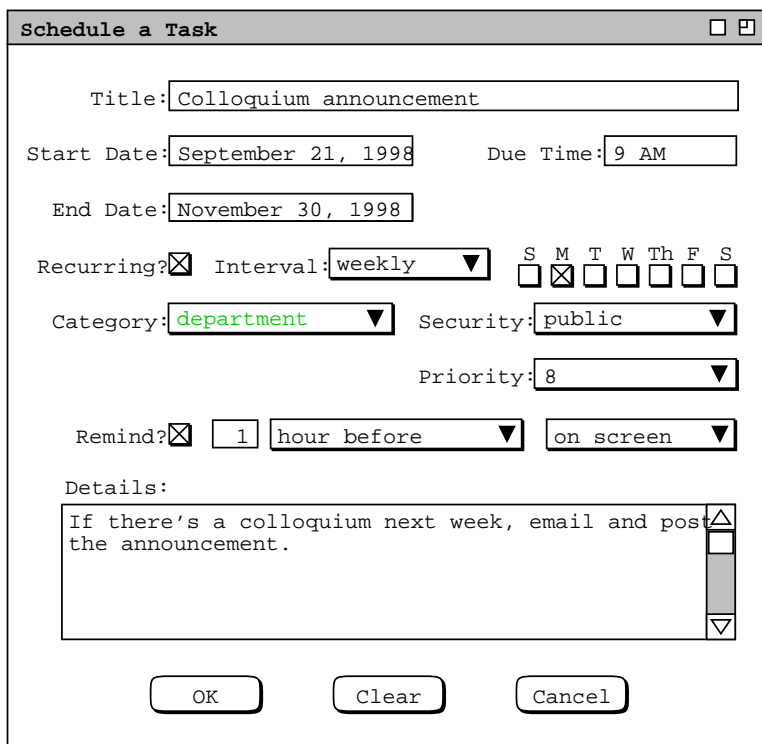


Figure 126: Scheduling a recurring task.

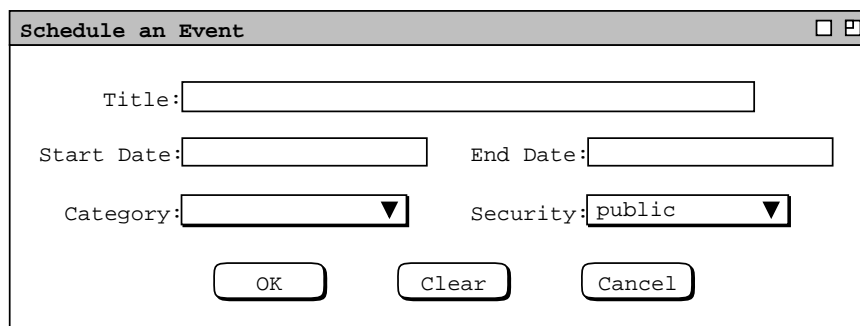


Figure 127: Event scheduling dialog.

events cannot be recurring, the end date is used to define a multi-day event. A multi-day event is a single item in the calendar, but it appears on multiple days in calendar and list views. If specified, the end date must be one or more days after the start date. If the end date is empty, the event is for a single day. As for a task, security is one of 'public' or 'private'.

Figures 128 and 129 show two filled-in event scheduling dialogs, for a single-day and multi-day event, respectively. When the user presses 'OK', the system adds the event to the user's calendar. 'Clear' clears any entered data values; 'Cancel' cancels the event scheduling operation. The system removes the dialog window from the screen when the user presses 'OK' or 'Cancel'.

Figures 130 and 131 show how a multi-day event appears in calendar and list views. The figures show how each day of a multi-day event is displayed as a separate item in both calendar and list views. The list view contains the "Year's events" custom list, shown originally in Figure 44. Here the "Ski vacation" event has been added and the list is scrolled down to show all its occurrences.

The screenshot shows a dialog box titled "Schedule an Event". It contains the following fields and controls:

- Title: Jim's Birthday
- Start Date: September 23, 1998
- End Date: (empty)
- Category: special event (dropdown menu)
- Security: private (dropdown menu)
- Buttons: OK, Clear, Cancel

Figure 128: Jim's birthday event.

The screenshot shows a dialog box titled "Schedule an Event". It contains the following fields and controls:

- Title: Ski vacation
- Start Date: December 13, 1998
- End Date: December 19, 1998
- Category: personal (dropdown menu)
- Security: public (dropdown menu)
- Buttons: OK, Clear, Cancel

Figure 129: Multi-day vacation event.

Monthly Agenda □ □						
◀ Today ▶		December 1998				
Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1 8-9 AM Ra 9 AM-5 PM	2 9-10 AM O 10-11 AM 11 AM-12 2:30-4:30	3 8-9 AM Ra 10:30-12 1:15-3 PM 3-5 PM So	4 9-10 AM O 10-11 AM 11 AM-12	5
6	7 9-10 AM O	8 8-9 AM Ra 9 AM-5 PM	9 9-10 AM O 2:30-4:30	10 8-9 AM Ra 1:15-3 PM	11 9-10 AM O	12
13 Ski vacat	14 Hanukkah Ski vacat	15 Ski vacat 8-9 AM Ra	16 Ski vacat	17 Ski vacat 8-9 AM Ra	18 Ski vacat	19 Ski vacat
20 Ramadan B	21 Winter So	22 Christmas 8-9 AM Ra	23	24 8-9 AM Ra	25	26
28	27	29 8-9 AM Ra	30	31 8-9 AM Ra		

Figure 130: Month view showing a multi-day event.

Year's events, sorted by Date □ □		
Title	Date	Category
Labor Day	Sep 7, 1998	holiday
Autumnal Equinox	Sep 22, 1998	holiday
Jim's Birthday	Sep 23, 1998	special e
Daylight Saving	Oct 25, 1998	holiday
Halloween	Oct 31, 1998	holiday
Veteran's Day	Nov 11, 1998	holiday
Thanksgiving	Nov 26, 1998	holiday
Ski vacation	Dec 13, 1998	personal
Hanukkah	Dec 14, 1998	holiday
Ski vacation	Dec 14, 1998	personal
Ski vacation	Dec 14, 1998	personal
Ski vacation	Dec 14, 1998	personal
Ski vacation	Dec 14, 1998	personal
Ski vacation	Dec 14, 1998	personal
Ski vacation	Dec 14, 1998	personal
Ramadan Begins	Dec 20, 1998	holiday
Winter Solstice	Dec 21, 1998	holiday
Christmas	Dec 25, 1998	holiday
New Year's Day	Jan 1, 1999	holiday
Martin Luther	Jan 18, 1999	holiday

Figure 131: Event list showing a multi-day event.

The Calendar Tool defines a number of built-in events for standard holidays and other event-worthy dates. The complete list of system-defined events is given in Section 2.11.16.

2.5. Finer Points of Viewing and Scheduling

This section of functional requirements covers the remaining details of viewing and scheduling calendar items. The topics covered are:

- viewing at the individual item level
- changing and deleting scheduled items
- details of recurring items
- scheduling and receiving reminders
- editing categories
- dynamic view updating

2.5.1. Item-Level Viewing

Item-level viewing for appointments was introduced in Section 2.3.1.5. This section covers further details of appointment viewing, as well as item views for meetings, tasks, and events. Independent of item type, the user views an item by selecting its time or title in any larger-grain view, and then executing the ‘View Item’ menu command. This series of interactions is illustrated in Figures 25 and 26. A short cut to view an item is to double click on the item time or title. Complete details of target item selection are covered in Section 2.11.9.

Common to all item views are change, delete, and clear buttons at the bottom of the display. In the initial viewing state, the change and clear buttons are disabled, and the delete button is enabled. If the user edits one or more data fields, the button states are reversed, with change and clear enabled, delete disabled. The scenarios of this section focus on viewing scheduled items prior to any editing by the user. Hence all item-level displays are shown in the initial change-disabled state. Complete details of changing and deleting items are covered in Section 2.5.2

2.5.1.1. Viewing Appointment Items

Figure 132 shows the result of the user viewing the item for 8 AM September 25.

The item-level appointment display has the same data fields as the appointment scheduling dialog shown in Figure 6. The item display differs from the scheduling dialog as follows:

- a. there is a time and date header in the item display, just below the window banner;
- b. the command buttons at the bottom of the item display are ‘Change’, ‘Delete’, and ‘Clear’ instead of ‘OK’, ‘Clear’, and ‘Cancel’.

At the far left of the time/date header are the Previous/Today/Next buttons that appear in all calendar views. These buttons are used to navigate among scheduled items, as described in Section 2.3.2. The ‘Change’, ‘Delete’, and ‘Clear’ commands are covered in Section 2.5.2.1.

Figure 133 shows the item-level view of a recurring appointment. The view shows the October 8 instance of the racket ball appointment scheduled in Figure 13.

The primary difference in a recurring versus non-recurring display is the addition of the three radio buttons above the command buttons. The radio button setting indicates how many recurring instances the ‘Change’ and ‘Delete’ commands apply to. For viewing purposes, the setting of these radio buttons affects the format of the display.

When ‘this instance’ is selected, the first date field is labeled ‘Instance Date’. This indicates that a change or delete affects this one instance only. The value for the ‘End Date’ field is shown in the display, but typing in the text field is disabled, as indicated by the greyed and dashed border. This is the case because changing the end date is not meaningful when only a single instance is in view. The values for the recurring information are shown, but the data edit fields are disabled. Again, this is because changing the recurring information is not meaningful when only a single instance is in view.

Figure 134 shows the October 8 item display when the user selects the ‘all instances’ radio button. In this case, the first date field is labeled ‘Start Date’, as in the appointment scheduling dialog (Figure 13). The value in the start date field is that entered when the appointment was scheduled, which is September 1 in this case. The

Scheduled Appointment
□ □

Today
8 AM Friday, September 25, 1998

Title:

Date: Start Time:

End Date: Duration: hr min

Recurring? Interval: S M T W Th F S

Category: Security:

Location: Priority:

Remind?

Details:

Figure 132: Item-level appointment view.

Scheduled Appointment [] []

◀ Today ▶ 8 AM Thursday, October 8, 1998

Title:

Instance
Date: Start Time:

End Date: Duration: hr min

Recurring? Interval: S M T W Th F S

Category: Security:

Location: Priority:

Remind?

Details:

▲
□
▼

Apply Change and Delete to:
 this instance all instances future instances

Figure 133: Item-level view of a recurring appointment.

The screenshot shows a window titled "Scheduled Appointment" with a navigation bar at the top containing a "Today" button and the text "8 AM Thursday, October 8, 1998". The main area contains several input fields and controls:

- Title:** Racket Ball
- Start Date:** September 1, 1998
- Start Time:** 8 AM
- End Date:** December 31, 1998
- Duration:** 1 hr
- Recurring?** Interval: weekly
- Day Selection:** S M T W Th F S
- Category:** personal
- Security:** title only
- Location:** rec centre
- Priority:** optional
- Remind?** 15 minutes before on screen
- Details:** Call Jim if you need to cancel, (with a scrollable text area)
- Apply Change and Delete to:**
 - this instance
 - all instances
 - future instances
- Buttons:** Change, Delete, Clear

Figure 134: Item-level view of a recurring appointment, all instances setting.

'End Date' field is enabled for editing, as is all of the recurring information except for the recurring checkbox. The disabled recurring checkbox means that a recurring appointment cannot be changed to non-recurring. This is the case because changing from recurring to non-recurring could potentially create a large number of "unbundled" appointment instances that could be cumbersome to change or delete.

Figure 135 shows the October 8 item display when the user selects the 'future instances' radio button. Here the dialog state above the radio buttons is the same as for the 'this instance' selection. Namely, the first date field is labeled 'Instance Date', the end date is shown but uneditable, and the recurring information is shown but uneditable. Changes apply to all instances starting from the instance date through to the end date, inclusive.

Prior to any changes in a recurring appointment, the same data values appear for all instances, except for the differences between instance date versus start date described above. When the 'all instances' radio button is selected, the data values for each and every unchanged instance are identical. The only display distinction between two different instances is the time and date header. For example, Figure 136 shows the racket ball racket instance for December 15. The only difference between this display and Figure 134 is the December 15 versus October 8 date header.

Of note for recurring items is that the 'Next' and 'Previous' commands do not specifically lead to the next and previous recurring instances. Rather, 'Next' and 'Previous' move consistently among all scheduled items, including not-yet-accepted meetings, in the order described in Section 2.3.2.1.

Scheduled Appointment
□ □

◀ Today ▶
8 AM Thursday, October 8, 1998

Title:

Instance Date: Start Time:

End Date: Duration: hr min

Recurring? Interval: S M T W Th F S

Category: Security:

Location: Priority:

Remind?

Details:

Call Jim if you need to cancel,

Apply Change and Delete to:

this instance
 all instances
 future instances

Figure 135: Item-level view of a recurring appointment, future instances setting.

The screenshot shows a window titled "Scheduled Appointment" with a date/time header "8 AM Tuesday, December 15, 1998". The appointment details are as follows:

- Title: Racket Ball
- Start Date: September 1, 1998 | Start Time: 8 AM
- End Date: December 31, 1998 | Duration: 1 hr
- Recurring? Interval: weekly | Days: S M T W Th F S
- Category: personal | Security: title only
- Location: rec centre | Priority: optional
- Remind? 15 minutes before | on screen
- Details: Call Jim if you need to cancel,
- Apply Change and Delete to: this instance all instances future instances
- Buttons: Change, Delete, Clear

Figure 136: Item-level view of another recurring instance..

2.5.1.2. Viewing Meeting Items

Figure 137 shows the result of the user viewing the item for 8 AM September 23. This is a view of the staff meeting scheduled in the scenario of Section 2.4.1.1. In particular, it is the view of the meeting after it has been accepted by the user in the notification dialog of Figure 113. The meeting view contains mostly the same information as in the notification dialog, with the following specific differences:

- There is a time and date summary in the viewing display, just below the window banner.
- There is a 'View' button to the right of the 'Minutes' field in the viewing display.
- There is no checkbox for sending an email notification in the viewing display.
- The command buttons at the bottom of the viewing display are 'Change', 'Delete', and 'Clear'.

At the far left of the time/date header are the Previous/Today/Next buttons, as described above for the appointment item. The 'Change', 'Delete', and 'Clear' buttons provide editing functionality, as covered in Sections 2.5.2.2 and 2.5.2.5.

As introduced in Section 2.4.1.1, the value entered in the 'Minutes' text field is a WWW URL. If the minutes text field is non-empty, then the 'View' button is enabled. If the minutes field is empty, then there are no available meeting minutes and the 'View' button is disabled. When the user presses an enabled 'View' button, the system displays the minutes using an external WWW browser, open on the given URL. The name and location of the external browser program are set as a user option, as described in Section 2.7.4.3. The Calendar Tool system performs no validation of the URL entered in the minutes field. If the URL is invalid in any way, the error condition is handled entirely by the external browser, as discussed in Section 2.12.4.

Scheduled Meeting [] [x]

◀ Today ▶ 8 AM Friday, September 23, 1998

Title:

Date: Start Time:

End Date: Duration: hr min

Recurring? Interval: S M T W Th F S

Category: Security:

Location: Priority:

Remind?

Attendees:

Details:

Agenda:

1. Duty assignments for Fall quarter
2. News of office remodeling project
3. Introduce new staff members
4. Promotions
5. Initial prep for accreditation visits

Minutes:

Scheduled By: On:

Host:

Figure 137: Item-level meeting view.

The scheduler may initially enter the URL in the form of a file directory, and subsequently edit the minutes field to specify an individual file. If the scheduler does not change a directory URL to an individual file, the system opens the external browser on the entire directory. If the server on which the URL resides does not support directory browsing, then the WWW browser may signal an error, as discussed in Section 2.12.4. In any case, it is the responsibility of the meeting scheduler to specify a valid URL for the meeting minutes. If the scheduler leaves the minutes field as a directory, it is also the scheduler's responsibility to name the files in the directory such that users can understand which files correspond to which meeting occurrences. The Calendar Tool does not enforce any constraints on the naming or structure of meeting minutes directories or files.

By definition, meeting minutes contain meaningful information after a meeting has taken place. If the scheduler specifies a meeting minutes file in advance of a meeting being held, it is the scheduler's responsibility to define appropriate pre-meeting content for the file, such as an indication that the minutes are to appear. The Calendar Tool does not enforce any constraints on the content of meeting minutes files.

When the user views a not-yet-accepted meeting at the item level, the command buttons at the bottom of the display are different. Specifically, the 'Change', and 'Delete' buttons are replaced with 'Accept' and 'Decline' buttons. For example, Figure 138 shows the item-level view of the staff meeting of Figure 137, before it has been accepted by the viewing user. This is the view of the not-yet-accepted meeting as it appears in the initial notification dialog of Figure 112. The 'Accept' and 'Decline' buttons in the item-level view have the same affect as in the notification dialog. If the user presses 'Accept', the meeting becomes a regular scheduled item, and the buttons in the item-level display become 'Change', 'Delete, and Clear'. If the user presses 'Decline', the meeting is not scheduled and the item-level display is removed from the screen. The 'Clear' button is used to clear any edits made in the dialog after it is initially displayed.

As noted in Section 2.4.1.5, the user cannot delay the accept/decline decision beyond the start date and time of a meeting. If the user fails to accept or decline within this time, the system automatically executes decline on the user's behalf, thereby deleting the meeting for the user.

Not-Yet-Accepted Meeting

◀ Today ▶ 8 AM Friday, September 23, 1998

Title: Computer Science Staff Meeting

Date: September 23, 1998 Start Time: 8 AM

End Date: Duration: 1 hr min

Recurring? Interval: weekly S M T W Th F S

Category: public Security: public

Location: 14-238 Priority: must

Remind? 15 minutes before on screen

Attendees:

csstaff

Details:

Agenda:

1. Duty assignments for Fall quarter
2. News of office remodeling project
3. Introduce new staff members
4. Promotions
5. Initial prep for accreditation visits

Minutes: <http://www.csc.calpoly.edu/~e> View ...

Scheduled By: estier On: 15sep98 08:24:01

Host: csc.calpoly.edu

Accept Decline Clear

Figure 138: Item-level view of a not-yet-accepted meeting.

Unless the user selects the option to hide penciled-in (i.e., not-yet-accepted) meetings, such meetings appear at all viewing levels, including the item level. This means that not-yet-accepted meetings appear at the appropriate points in the next/previous item traversals described in Section 2.3.2.1.

Of note in this scenario is how the state of Figure 138 reflects the unedited state of the initial notification dialog in Figure 112. In contrast, the state of Figure 137 reflects the state of the edited notification dialog in Figure 113. The figure pairings represent two different phases of a scheduled meeting. Figures 112 and 138 represent an earlier phase, before the meeting has been accepted. Figures 113 and 137 represent a later phase, after the meeting has been accepted.

Figure 139 shows the item-level display of the recurring meeting scheduled in Section 2.4.1.1. The view shows a

Scheduled Meeting
□ □

◀ Today ▶
10 AM Thursday, October 1, 1998

Title:

Instance Date: Start Time:

End Date: Duration: hr min

Recurring? Interval:

Category: Security:

Location: Priority:

Remind? minutes before

Attendees:

▲
□
▼

Details:

▲
□
▼

Minutes:

Scheduled By: On:

Host:

Apply Change and Delete to:

this instance all instances future instances

Figure 139: Item-level view of a recurring meeting.

particular instance of the recurring meeting, for the date of October 1. The change/delete radio buttons have exactly the same functionality as in the appointment view, including the same effects on the display format.

2.5.1.3. Viewing Task Items

When the user selects 'View Item' for a task, the system displays a view of the form shown in Figure 140. This view shows the first occurrence of the recurring task scheduled in Figure 126. The item-level view of a task contains mostly the same information as the task scheduling dialog (Section 2.4.2), with the following differences:

- there is a time and date summary in the item display, just below the window banner;
- there are two additional data fields in the item display: the 'Completed?' checkbox and the 'Completion Date' text field.
- the command buttons at the bottom of the item display are 'Change', 'Delete, and Clear' instead of 'OK', 'Clear', and Cancel';
- if the task is recurring, there are instance-selection radio buttons above the command buttons.

The 'Completed?' and 'Completion Date' data fields indicate the completion status of a task or task instance. Details of how these fields are set are presented in Section 2.5.2.6. The change/delete radio buttons have exactly the same functionality as in the appointment view, including the same effects on the display format.

The screenshot shows a window titled "Scheduled Task" with a close button. Below the title bar is a navigation area with a left arrow, a "Today" button, and a right arrow, followed by the text "9 AM Friday, September 21, 1998".

The main form area contains the following fields and controls:

- Title:
- Instance Date: Due Time:
- End Date:
- Recurring? Interval:
- Category: Security:
- Priority:
- Remind?
- Details:
- Completed? Completion Date:
- Apply Change and Delete to:
 - this instance
 - all instances
 - future instances
- Buttons:

Figure 140: Item-level task view.

2.5.1.4. Viewing Event Items

Figure 141 shows the result of the user executing ‘View Item’ for the event scheduled in Figure 128. The item-level view of an event contains mostly the same information as the event scheduling dialog (Section 2.4.3), with the following differences:

- a. there is a time and date summary in the item display, just below the window banner;
- b. the command buttons at the bottom of the item display are ‘Change’, ‘Delete’, and ‘Clear’ instead of ‘OK’, ‘Clear’, and ‘Cancel’.

The event item display has no additional data fields than those in the event scheduling dialog. Details of event change and deletion are covered in Section 2.5.2.7.

2.5.1.5. Item-Level Viewing in Other Users’ Calendars

Figure 142 shows the user viewing another user’s calendar at the item level. This is the view of the faculty meeting in the calendar of James L. Brandon, prior to the meeting having been accepted by that user. Access to other users’ items is gained through the ‘View Other User’ command, detailed in Section 2.3.5.1. When viewing other users at the item level, none of the data fields is editable. Hence, all data fields are disabled. Also, the command buttons at the bottom of the display are absent, as are the associated radio buttons in a recurring item.

In a user’s own calendar, the item view of not-yet-accepted meetings has ‘Accept’ and ‘Decline’ command buttons instead of ‘Change’ and ‘Delete’. This distinction in buttons indicates that an item is not yet accepted. Since the command buttons are missing in the view of another user’s items, the title text of another user’s not-yet-accepted item is shown in light-grey italics.

2.5.2. Changing and Deleting Scheduled Items

Changing and deleting scheduled items is performed in item-level displays. As illustrated in the preceding section of the requirements, all item-level displays have ‘Change’ and ‘Delete’ command buttons to perform these operations. The enabled/disabled state of these buttons is consistent in all four types of item. Specifically, when the system initially displays an item-level view, the ‘Change’ and ‘Clear’ buttons are disabled and the ‘Delete’ button is enabled. This is illustrated in the display excerpt shown in Figure 143. What this means is that ‘Change’ and ‘Clear’ are not applicable until the user edits one or more data fields. Furthermore, the ‘Delete’ command is only applicable prior to any data field editing, or after the user presses ‘Clear’ to remove the edits.

As soon as the user performs an edit to some data field, the initial state of the command buttons is reversed. Namely, the ‘Change’ and ‘Clear’ buttons are enabled and the ‘Delete’ button is disabled. This is illustrated in Figure 144. The command buttons remain in the Figure 144 state until the user executes ‘Change’ or ‘Clear’, or re-edits all edited data fields back to their original values. Executing the ‘Change’ command changes the item in the

The screenshot shows a window titled "Scheduled Event" with a close button in the top right corner. Below the title bar is a navigation bar with a left arrow, a "Today" button, and a right arrow, followed by the date "September 23, 1998". The main area contains the following fields:

- Title: Jim's Birthday
- Start Date: September 23, 1998
- End Date: (empty)
- Category: special event (dropdown menu)
- Security: private (dropdown menu)

At the bottom of the dialog are three buttons: "Change", "Delete", and "Clear". The "Delete" button is highlighted with a dark border, indicating it is the active or enabled button.

Figure 141: Item-level event view.

Not-Yet-Accepted Meeting, user James L. Brandon

Today 1 PM Monday, September 21, 1998

Title: Computer Science Faculty Meeting

Start Date: September 21, 1998 Start Time: 1 PM

End Date: November 30, 1998 Duration: 1 hr 0 min

Recurring? Interval: biweekly S M T W Th F S

Category: Security: public

Location: 14-256 Priority: must

Remind? 15 minutes before on screen

Attendees: csfac

Details: Agenda TBA.

Minutes: <http://www.csc.calpoly.edu/~e> View ...

Scheduled By: estier On: 15Sep98 08:36:52

Host: csc.calpoly.edu

Figure 142: Item-level view of another user’s calendar.



Figure 143: Initial state of command buttons in all item-level displays.



Figure 144: State of command buttons after user edits one or more data fi elds.

affected calendars and all affected displays. Executing the ‘Clear’ command clears all edits, restoring the data fi elds to their original values when the item-level view was initially displayed, or mos recently changed. No calendar changes are made when ‘Clear’ is executed. When either ‘Change’ or ‘Clear’ is executed, the command button state returns to that of Figure 143. The command buttons also return to this state when the user re-edits all edited data fi elds back to their original values.

When the user executes and confirms 'Delete', the system removes the item from the affected calendars, removes the item-level display from the screen, and updates any affected display windows.

Since the different types of scheduled item share many data fields in common, many aspects of changing items are common to all types. Aspects of item deletion are also common among the different types of item, such as the general form of the confirmation dialogs. The most wide-ranging effects of change and delete are made to meetings by meeting schedulers. Such changes and deletions potentially affect meeting attendees. As noted in Section 2.4.1.5, the system maintains a separate copy of a meeting item for the scheduler and for each attendee. Further, the system allows schedulers as well as attendees to change or delete their own individual copies. This multi-user access to meeting items leads to functional requirement details that must be carefully considered. The requirements presented in Sections 2.5.2.2 through Section 2.5.2.5 address these details.

2.5.2.1. Changing and Deleting Appointments

To change a scheduled appointment, the user edits one or more data fields in the item-level display and then presses the 'Change' button. For example, Figure 145 shows the user having edited the date and time fields of the dentist appointment scheduled in Figure 7. The change to the date is from September 25 to 29; the change to the start time is from 8 AM to 8:30. When the user presses 'Change' in this display, the system responds with the confirmation dialog shown in Figure 146. When the user presses 'OK', the system changes the item per the edits made to the data fields. If the user presses 'Cancel', the changes are not made. In either case, the confirmation dialog is removed from the screen and the item-level display again becomes current. In the 'OK' case, the command button state is returned to its initial configuration (Figure 143). In the 'Cancel' case, the buttons remain in the edits-performed state (Figure 144).

Figure 145: Date and time changes to the dentist appointment.

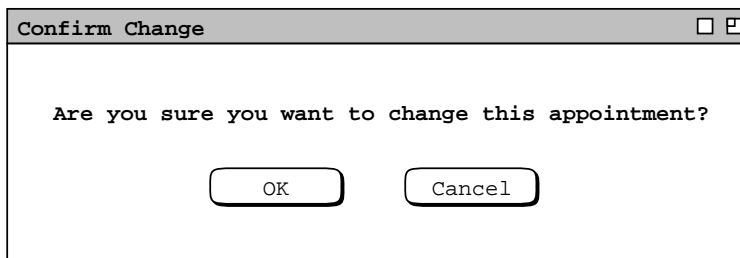


Figure 146: Change confirmation dialog.

In this scenario, the user selects 'OK' in the confirmation dialog, whereupon the system proceeds with the appointment change. After the change is confirmed, the system updates all display windows that are affected by the change. For example, the month view shown in Figure 22 is changed to that shown in Figure 147, where the dentist appointment has been moved from 8-9:30 on the 25th to 8:30-10 on the 29th. All other display windows that contain the dentist appointment are updated to reflect the confirmed change. Further details of dynamic view updating are covered in Section 2.5.6

All data fields may be edited in the item-level view of an appointment. The same data-entry rules apply when changing an appointment as when initially scheduling it. These rules are covered in detail in Section 2.11.7. In addition to these data-entry rules, there are the following restrictions on changing a recurring appointment:

- a. a recurring appointment cannot be changed to non-recurring;

Monthly Agenda						
◀ Today ▶		September 1998				
Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1 8-9 AM Ra 11 AM-2 P	2	3 8-9 AM Ra	4	5
6	7 Labor Day	8 8-9 AM Ra 11 AM-2 P	9	10 8-9 AM Ra	11	12
13	14	15 8-9 AM Ra 11 AM-2 P	16 1. Send c	17 8-9 AM Ra	18	19
20	21 1. Colloq 9-10 AM O 10-11 AM 11 AM-12	22 Autumnal 8-9 AM Ra 9 AM-5 PM	23 8-9 AM St 8-9:30 AM 9-10 AM O 10-11 AM	24 Jim's Bir 1. Prepar 2. Buy so 8-9 AM Ra	25 9-10 AM O 10-11 AM 11 AM-12	26
28	27 1. Colloq 9-10 AM O 10-11 AM 11 AM-12	29 8-9 AM Ra 8:30-10 A 9 AM-5 PM	30 9-10 AM O 10-11 AM 11 AM-12 2:30-4:30			

Figure 147: Updated view after confirmed appointment change.

- b. changes made to any of the following data fields in a recurring appointment apply to all instances of the appointment:
 - Start Date (as opposed to Instance Date)
 - End Date
 - Recurring Interval and the associated recurring details

Figure 148 shows the user having edited an instance of a recurring appointment. Here the start time for the September 29th instance of the racket ball appointment has been changed from 8-9 AM to 7-8 AM. When the user presses 'Change', the system responds with the change confirmation dialog in Figure 149. The user selects 'OK' whereupon the system proceeds with the single-instance change and all necessary view updates. For example, the month view in Figure 147 is updated to that in Figure 150. The figure shows the time change to the September 29th instance of the racket ball appointment. The times of the other instances on the 1st, 8th, 15th, and 22nd remain the same.

The user decides to perform another change to the racket ball appointment, by deleting the last three weeks from the recurring schedule. To do so, the user moves to the October 1 instance and performs the edit shown in Figure 151. Here the user has changed the end date from December 31 to December 10. Of note is that the user moved to an instance other than the edited September 29. If the user had remained at September 29 to make the all-instance change to the end date, then the September 29 start-time change would be made to all instances as well. Since this is not what the user wants, the end-date change is made in another instance. Even after an individual instance has

The screenshot shows a window titled "Scheduled Appointment" with a navigation bar at the top showing "Today" and "8 AM Tuesday, September 29, 1998". The main form contains the following fields and controls:

- Title: Racket Ball
- Instance Date: September 29, 1998
- Start Time: 7 AM
- End Date: December 31, 1998
- Duration: 1 hr
- Recurring: checked, Interval: weekly
- Day selection: S, M, T, W, Th, F, S (T and W are selected)
- Category: personal
- Security: title only
- Location: rec centre
- Priority: optional
- Remind: unchecked, 15 minutes before, on screen
- Details: Call Jim if you need to cancel,
- Apply Change and Delete to:
 - this instance
 - all instances
 - future instances
- Buttons: Change, Delete, Clear

Figure 148: Edits made to a recurring appointment.

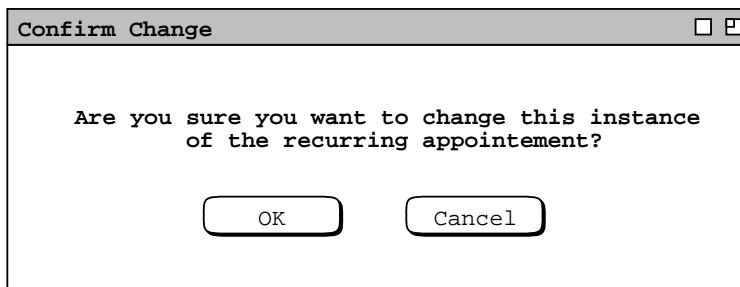


Figure 149: Confirmation dialog for changing one instance of a recurring appointment.

Monthly Agenda						
September 1998						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1 8-9 AM Ra 11 AM-2 P	2	3 8-9 AM Ra	4	5
6	7 Labor Day	8 8-9 AM Ra 11 AM-2 P	9	10 8-9 AM Ra	11	12
13	14	15 8-9 AM Ra 11 AM-2 P	16 1. Send c	17 8-9 AM Ra	18	19
20	21 1. Colloq 9-10 AM O 10-11 AM 11 AM-12	22 Autumnal 8-9 AM Ra 9 AM-5 PM	23 8-9 AM St 8-9:30 AM 9-10 AM O 10-11 AM	24 Jim's Bir 1. Prepar 2. Buy so 8-9 AM Ra	25 9-10 AM O 10-11 AM 11 AM-12	26
28	27 1. Colloq 9-10 AM O 10-11 AM 11 AM-12	29 7-8 AM Ra 8:30-10 A 9 AM-5 PM	30 9-10 AM O 10-11 AM 11 AM-12 2:30-4:30			

Figure 150: Updated view after another appointment change.

Scheduled Appointment [] [x]

< Today > 8 AM Thursday, October 1, 1998

Title: Racket Ball

Start Date: September 1, 1998 Start Time: 8 AM

End Date: December 10, 1998 Duration: 1 hr min

Recurring? Interval: weekly S M T W Th F S

Category: personal Security: title only

Location: rec centre Priority: optional

Remind? 15 minutes before on screen

Details:
Call Jim if you need to cancel,

Apply Change and Delete to:
this instance all instances future instances

Change Delete Clear

Figure 151: Additional edits made to a recurring appointment.

changed, it remains an instance of the same recurring appointment. Therefore, it may be used to effect changes to all other or all future instances. This issue is discussed further in Section 2.5.3.5.

The user proceeds to press 'Change' in Figure 151, in response to which the system displays the confirmation dialog in Figure 152. When the user presses 'OK', the system performs the change, which the user then inspects by opening the month view for December, shown in Figure 153. The previously scheduled racket ball instances for December 15 through 31 are now gone.

Confirm Change [] [x]

Are you sure you want to change all instances of the recurring appointment?

OK Cancel

Figure 152: Confirmation dialog for changing all instances of a recurring appointment.

Monthly Agenda						
◀ Today ▶		December 1998				
Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1 8-9 AM Ra 9 AM-5 PM	2 9-10 AM O 10-11 AM 11 AM-12 2:30-4:30	3 8-9 AM Ra 10:30-12 1:15-3 PM 3-5 PM So	4 9-10 AM O 10-11 AM 11 AM-12	5
6	7 9-10 AM O	8 8-9 AM Ra 9 AM-5 PM	9 9-10 AM O 2:30-4:30	10 8-9 AM Ra 1:15-3 PM	11 9-10 AM O	12
13 Ski vacat	14 Hanukkah Ski vacat	15 Ski vacat	16 Ski vacat	17 Ski vacat	18 Ski vacat	19 Ski vacat
20 Ramadan B	21 Winter So	22 Christmas	23	24	25	26
28	27	29	30	31		

Figure 153: Updated view after further appointment change.

If the user changes an appointment with ‘all future’ selected, the system displays the dialog of Figure 154. The dialog contains the appropriate wording change to reflect the ‘all future’ selection.

To delete a non-recurring appointment, the user presses ‘Delete’ in the item-level display. In response, the system displays the confirmation dialog of Figure 155. The user presses ‘OK’ to proceed with the delete, in response to which the system deletes the appointment from the user’s calendar and removes both the confirmation dialog and item-level display from the screen. The system also updates any affected display windows by removing the deleted item. If the user selects ‘Cancel’, the system performs no delete and removes the confirmation dialog, leaving the item-level display in its state prior to ‘Delete’ having been executed.

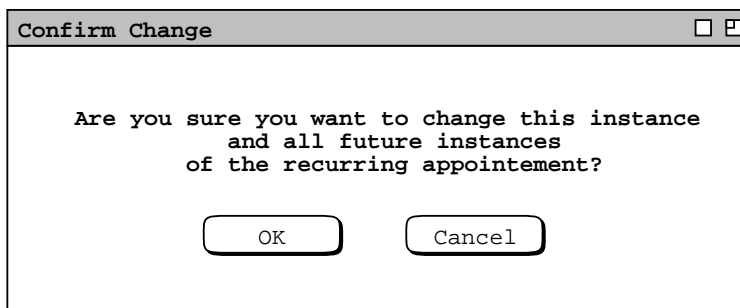


Figure 154: Confirmation dialog for changing all future instances of a recurring appointment.

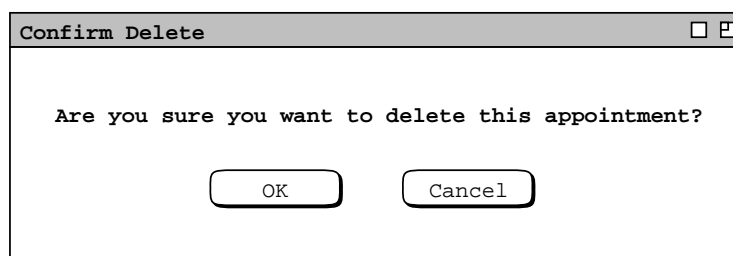


Figure 155: Delete confirmation dialog.

When the user executes 'Delete' for a recurring appointment, the system displays one of the three confirmation dialogs shown in Figure 156, based on the setting of the change/delete radio buttons. When the user presses 'OK', the system deletes the one or more instances as selected, removes the item display from the screen, and updates all affected display windows by removing the deleted item(s). If the user selects 'Cancel', no deletions are made and the item-level dialog is made current. In either case, the confirmation dialog is removed from the screen.

2.5.2.2. Scheduler Changing and Deleting Meetings

The change and delete commands for meetings behave much the same as they do for appointments. The significant difference is when a scheduler changes a critical meeting data field or deletes a meeting, all attendees are notified. Scheduler deletion of a meeting is the means by which the meeting is effectively cancelled.

As described in Section 2.4.1, some of the data fields of a scheduled meeting are computed by the system, based on the initial scheduling request. This means there are scheduler restrictions on changing some meeting data fields. Furthermore, some data fields are sufficiently critical to the meeting that all attendees must be notified if one or more such fields are changed. Table 9 summarizes what changes, if any, are allowed by the scheduler and which field changes require attendee notification. The table entries listed as "any" are still subject to the general data entry rules described in Section 2.11

Figure 157 shows edits made by the scheduler to the staff meeting confirmed in Figure 88. The changes are to increase the duration from 1 hour to 1.5 hours and to update the agenda. When the scheduler presses 'Change', the system responds with the confirmation dialog in Figure 158. The contents of this dialog reflect two conditions of the change:

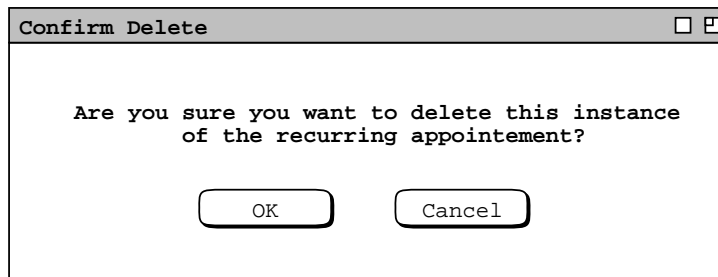
- a. One or more critical data fields have been changed, Duration in this case; this requires that all attendees be notified.
- b. There is a time conflict for one or more attendees; the scheduler is informed of this condition in the dialog.

If the scheduler chooses 'OK', the system proceeds with the change on the scheduler's calendar and notifies all attendees of the change, except for the scheduler. If the scheduler chooses 'Cancel', the system performs no changes. In either the 'OK' or 'Cancel' case, the confirmation dialog is removed from the screen and the item-level meeting display becomes current.

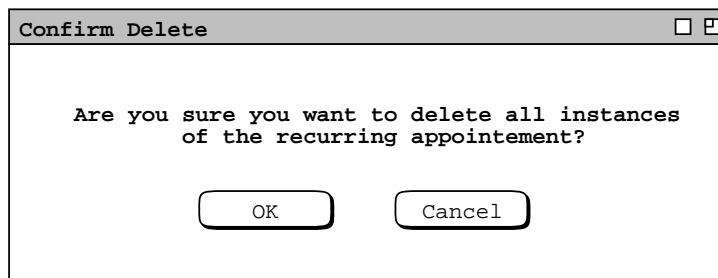
In this scenario, the scheduler chooses 'OK'. The scheduler's calendar is changed and all affected display windows are changed on the scheduler's display screen. In this case the scheduler is among the attendees and also appears in the conflicts list. The scheduler receives no notification, since the dialog itself serves this purpose. The fact that the meeting conflicts with existing item(s) on the scheduler's calendar is handled in the normal way, i.e., the item is subsequently displayed in overlapping format, as illustrated in Section 2.3.1.

When the scheduler confirms the change, all attendees are notified. Complete details of attendee change notification are covered in Section 2.5.2.3

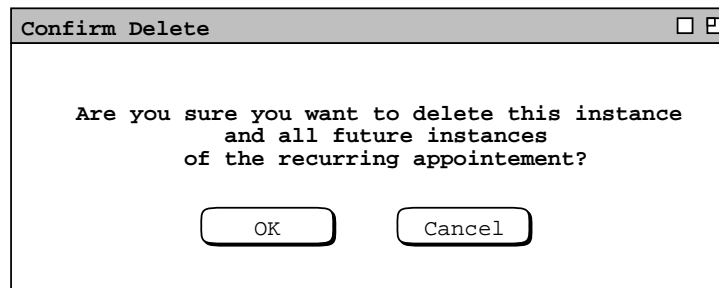
The definition of time conflict when changing a meeting is the same as when the meeting is initially scheduled. Viz., a conflicting item on an attendee's calendar is any non-private item that overlaps with the changed meeting. Such a conflict can arise when the scheduler makes a change to one or more of the following meeting fields:



Delete confirmation for one instance of a recurring appointment.



Delete confirmation for all instances of a recurring appointment.



Delete confirmation for all future instances of a recurring appointment.

Figure 156: Delete confirmation dialogs for a recurring appointment.

Data Field	Changes Allowed	Notification Required
Title	any except empty	no
(Start) Date	any future date, with warning if conflicts detected	yes
End Date	same as Start Date	yes
Start Time	any, with warning if conflicts detected	yes
Duration	same as Start Time	yes
Recurring	any, with warning if conflicts detected	yes
Category	any	no
Security	any	no
Location	any available at time(s) of meeting	yes
Priority	any	no
Remind	any	no
Attendees	none	
Details	any	no
Minutes	any	no
Scheduled By	none	
On	none	
Host	none	

Table 9: Summary of allowable changes to meeting data fields by the scheduler.

Scheduled Meeting □ □

◀ Today ▶ **8 AM Friday, September 23, 1998**

Title:

Date: Start Time:

End Date: Duration: ^{hr} ^{min}

Recurring? Interval: S M T W Th F S

Category: Security:

Location: Priority:

Remind?

Attendees:

Details:

Minutes:

Scheduled By: On:

Host:

Figure 157: Duration and details changes to the staff meeting.

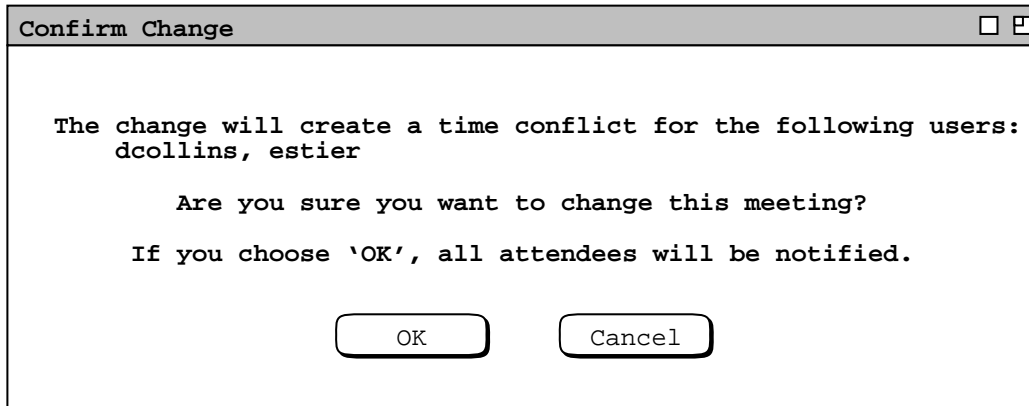


Figure 158: Confirmation dialog when scheduler changes critical meeting field, with conflicts.

- (Start) Date
- End Date
- Start Time
- Duration
- Recurring

The system provides no means to recompute a list of possible meeting times via the ‘Change’ operation. All that the system does is detect and report attendee time conflicts. If the scheduler wants to reschedule the meeting at time when there are no conflicts, the scheduler must re-execute the ‘Schedule Meeting’ operation to view a list of non-conflicting possible times. The scheduler can then choose a time from that list for the change, using copy and paste. Alternatively, the scheduler can delete the original meeting and re-execute the ‘Schedule Meeting’ operation from scratch.

Figure 159 shows the scheduler having edited an instance of the recurring lecturer meeting scheduled in Figure 91. The change is to the ‘Minutes’ field for a single meeting instance. In this scenario, the current date is assumed to be after the first occurrence of the lecturer meeting, so the change to the ‘Minutes’ field is considered a normal operation. When the scheduler presses ‘Change’ in Figure 159, the system displays the dialog in Figure 160. Because no critical data field has changed, the dialog provides the option of whether or not to notify all attendees. In this case, the scheduler leaves the default ‘Notify’ setting on and presses ‘OK’, whereupon the system updates the scheduler’s calendar, updates any affected displays on the scheduler’s screen, and notifies all attendees of the change.

Figure 161 shows another change to the lecture meeting, this time to future instances, at and beyond the October 1 instance. The figure reflects the scheduler having navigated to the October 1 instance, i.e., the scheduler did not change the instance date. The changes are to the ‘Location’ and ‘Details’ fields. Location is a critical field that requires attendee notification. The ‘Details’ field has a textual note indicating the location change. This is an example of using ‘Details’ for a slightly different purpose than may be typical, viz., a note about a change. Since this change will appear in all future instances, the scheduler may later perform individual edits to the ‘Details’ field.

When the scheduler presses ‘Change’ in Figure 161, the system displays the dialog of Figure 162. When the scheduler confirms the Location change, the system checks that the new location is available at all meeting times, that is, the times of all future instances. If it is not available, an error message is issued, as discussed in Section 2.12.3.4 The system also performs the appropriate calendar and display changes.

Figures 158, 160, and 162 show three examples of the confirmation dialog for changes made by a meeting scheduler. There are sixteen possible configurations for this dialog, based on whether there are attendee conflicts, whether the meeting recurs, and whether attendee notification is to happen. The following is a description of all possible configurations:

- If there are conflicts, the following text appears at the top of the dialog:

Scheduled Meeting □ □

◀ Today ▶ **10 AM Thursday, September 24, 1998**

Title:

Instance Date: Start Time:

End Date: Duration: ^{hr} ^{min}

Recurring? Interval: S M T W Th F S

Category: Security:

Location: Priority:

Remind?

Attendees:

△
□
▽

Details:

△
□
▽

Minutes:

Scheduled By: On:

Host:

Apply Change and Delete to:

this instance all instances future instances

Figure 159: Scheduler edits a recurring meeting.

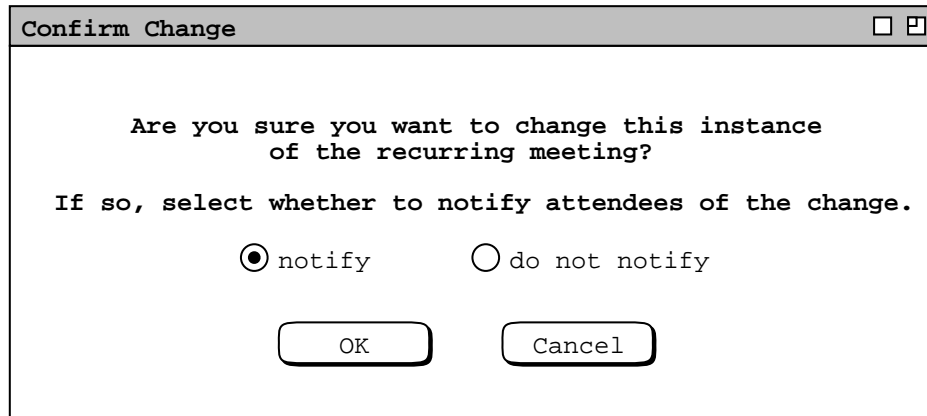


Figure 160: Confirmation dialog when scheduler changes non-critical meeting field in a recurring meeting.

Scheduled Meeting
□ □

◀ Today ▶
10 AM Thursday, October 1, 1998

Title:

Instance Date: Start Time:

End Date: Duration: hr min

Recurring? Interval: S M T W Th F S

Category: Security:

Location: Priority:

Remind?

Attendees:

cslect ▲
□
▼

Details:

NOTE room change for future meetings. ▲
□
▼

Minutes:

Scheduled By: On:

Host:

Apply Change and Delete to:

this instance
 all instances
 future instances

Figure 161: Location change to future lecturer meetings.

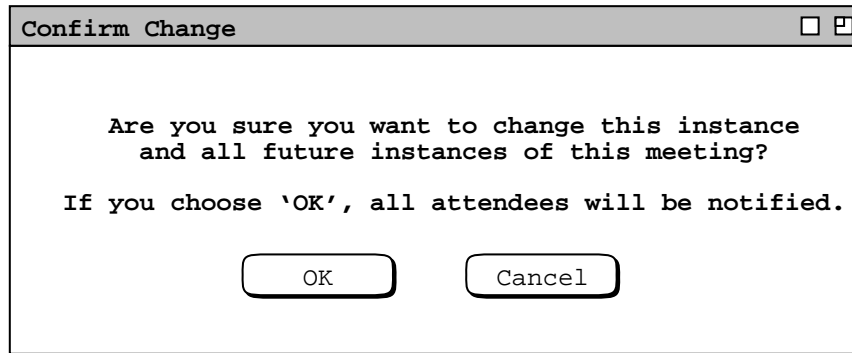


Figure 162: Confirmation dialog when scheduler changes critical field for future instances.

"The change will create a time conflict for the following users:"

with the IDs of all affected users listed below the message.

- The confirmation message appears below the conflict message, or at the top of the dialog if there are no conflicts. The confirmation message text is one of the following four alternatives, based on whether the meeting is recurring, and if so, which of the change/delete radio buttons is selected:

"Are you sure you want to change this meeting?"

"Are you sure you want to change this instance of the recurring meeting?"

"Are you sure you want to change all instances of the recurring meeting?"

"Are you sure you want to change this instance and all future instances of the recurring meeting?"

- Below the confirmation message is the indication or choice of attendee notification. If one or more critical fields are changed, the notification text is

"If you choose 'OK', all attendees will be notified."

If no critical field is changed, the notification text is

"If so, select whether to notify all attendees of the change."

below which are the 'notify' and 'do not notify' radio buttons.

- At the bottom of the dialog are 'OK' and 'Cancel' buttons.

To delete a non-recurring meeting, the scheduler presses 'Delete' in the item display. In response, the system displays the dialog of Figure 163. Scheduler deletion of a meeting always requires attendee notification. Therefore the delete confirmation provides no notification option.

The scheduler presses 'OK' to proceed with the delete, in response to which the system deletes the meeting from the scheduler's calendar, deletes the meeting from all attending group calendar(s), and notifies all attending users. The system also updates any affected display windows on the scheduler's screen by removing the deleted item from those displays. Updates to attendee screens are performed as described in Section 2.5.2.3. If the scheduler selects 'Cancel', the system performs no delete and removes the confirmation dialog, leaving the item-level display in its state prior to 'Delete' having been executed.

Deletion of a recurring meeting applies to all instances selected by the change/delete radio buttons, with the same effects for each instance as non-recurring deletion. The confirmation dialogs are shown in Figure 164.

When the scheduler cancels a meeting with 'Delete', the meeting is permanently removed from the scheduler's calendar, and from the calendars of all accepting attendees. If the scheduler wants to retain a record of a cancelled

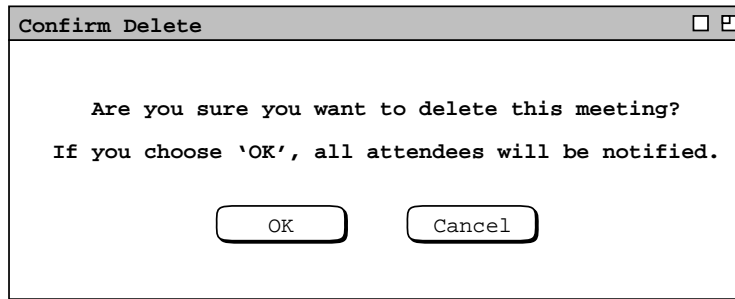
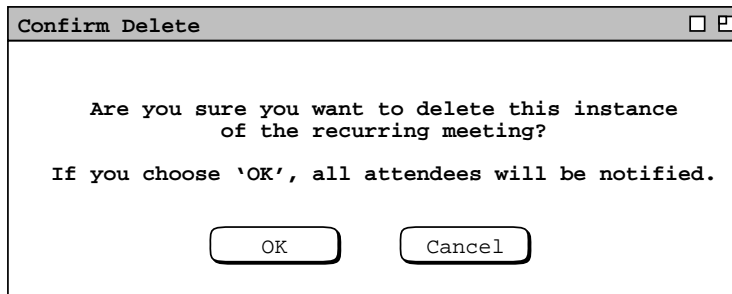
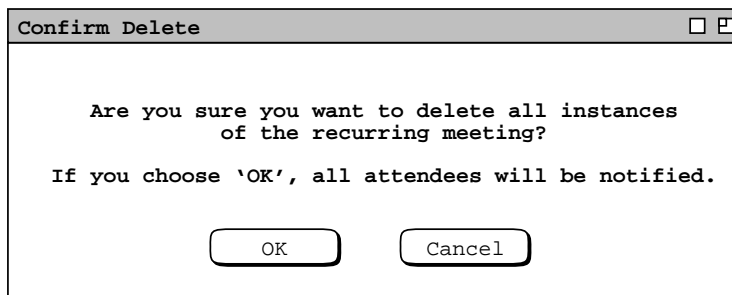


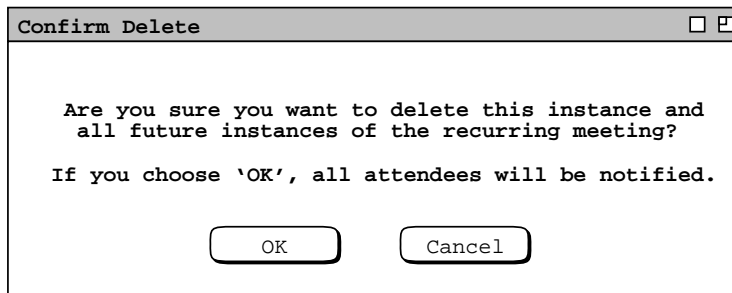
Figure 163: Confirmation dialog for scheduler-deleted meeting.



Delete confirmation for one instance of a recurring appointment.



Delete confirmation for all instances of a recurring appointment.



Delete confirmation for all future instances of a recurring appointment.

Figure 164: Confirmation dialogs for scheduler-deleted recurring meeting.

meeting, the scheduler can perform cancellation with ‘Change’ instead of ‘Delete’. To do so, the scheduler can type an appropriate cancellation text in the ‘Title’ and/or ‘Details’ fields. The cancellation text will be visible to users in bold red type in the notification details dialog, as described in the next section.

2.5.2.3. Meeting Change and Delete Notifications

When the scheduler changes a meeting such that a notification is required or selected, the system notifies each attendee (except the scheduler) with a dialog of the form shown in Figure 165. This is a notification for the change made in Figure 157. The message indicates the scheduler’s ID plus the time and date of the changed meeting.

When a scheduler change causes a conflict with one or more existing items in the notified user’s calendar, the notification message has the following line appended:

NOTE: this change conflicts with *N* existing item(s).

where *N* is the number of scheduled items that have overlapping times with the changed meeting and "item(s)" is

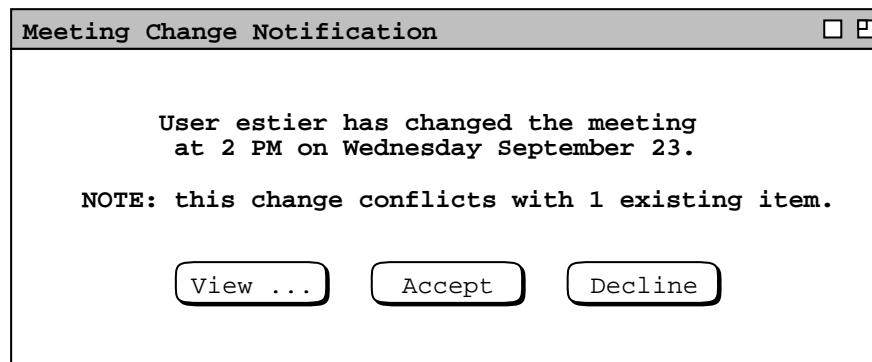


Figure 165: Meeting change notification.

singular if $N = 1$, plural otherwise.

The user can view the complete details of the change by pressing the 'View ...' button in the initial change notification dialog. For example, when the user presses 'View ...' in Figure 165, the system responds by displaying the notification details dialog in Figure 166. The changes made by the scheduler (in Figure 157), are highlighted in boldface red font. If the scheduler has changed the 'Category' field, the category name is highlighted in boldface font, but in the normal color of the category. (There is no category change in this scenario.) For all other changed data fields, the entire field value is displayed in bold red, not just the individual characters or lines to which edits were made. For example in Figure 166, The entire text of the 'Details' field is shown in bold red, even though only a single line of the text has been changed by the scheduler.

There are 'Accept' and 'Decline' buttons at the bottom of the display. To the left of the buttons is a label explaining that they apply to changes made by the scheduler. The user can edit any data fields except 'Scheduled By', 'On', and 'Host'.

The user can respond to a meeting change notification in one of three ways: accept the change, decline the change, or delay a decision. Both the smaller change notification dialog (Figure 165) and the larger details dialog (Figure 166) have 'Accept' and 'Decline' buttons. If the user chooses 'Accept', the changes are merged into the scheduled item in the user's calendar. If the user chooses 'Decline', the changes are not made.

To delay the accept/decline decision, the user can close the notification dialog window, using the 'View Windows Close' command. A user cannot delay an accept/decline decision beyond the date and start time of a non-recurring meeting, or beyond the date and start time of the last occurrence of a recurring meeting. If a user fails to accept or decline within this time, the system automatically executes decline on the user's behalf.

In the case of Figure 166, the notified user decides to edit the 'Duration' field from the scheduler changed value of 1.5 hours back to the original value of 1 hour, as shown in Figure 167. In this scenario, the viewing user is one of those for whom the scheduler's change has created a conflict, as indicated in the small notification dialog in Figure 165. Therefore, the user re-edits the duration to avoid the conflict. This is an example where a user legitimately edits a scheduler-generated data value. The issue of what constitutes a "legitimate" user edit of a meeting is discussed further in Section 2.5.2.5.

When the user presses 'Accept' in Figure 167, the system merges the scheduler's changes into the user's copy of the meeting item. Change merging is accomplished by retaining the most recently entered value for each meeting data field. For this staff meeting, the user has changed the 'Title', 'Duration', 'Category', and 'Remind' fields more recently than the scheduler. Therefore, the user's values for these fields are retained in the changed version. In the case of 'Title', 'Category', and 'Remind', the user's edits were made prior to the original acceptance of the scheduled meeting (Figure 113). The 'Duration' field was edited by the scheduler (Figure 157), but then subsequently edited by the user in the notification dialog (Figure 167). Therefore, the user's edit to 'Duration' is the most recent, and it is retained in the changed version. The scheduler's change to the 'Details' field is most recent, and therefore appears in the accepted version. All of the other data fields are

Meeting Change Notification Detailed View

Title:

Date: Start Time:

End Date: Duration: hr min

Recurring? Interval: S M T W Th F S

Category: Security:

Location: Priority:

Remind? minutes before

Attendees:

Details:

UPDATED Agenda:

1. Presentation by campus safety coordinator
2. Duty assignments for Fall quarter
3. News of office remodeling project
4. Introduce new staff members
5. Promotions

Minutes:

Scheduled By: On:

Host:

Scheduler Changes:

Figure 166: Viewing details of a meeting change notification.

Meeting Change Notification Detailed View
□ □

Title:

Date: Start Time:

End Date: Duration: hr min

Recurring? Interval: S M T W Th F S

Category: Security:

Location: Priority:

Remind?

Attendees:

Details:

UPDATED Agenda:

1. Presentation by campus safety coordinator
2. Duty assignments for Fall quarter
3. News of office remodeling project
4. Introduce new staff members
5. Promotions

Minutes:

Scheduled By: On:

Host:

Scheduler Changes:

Figure 167: Change notification details dialog edited.

unchanged by either the scheduler or user from their original scheduled values, and hence are retained as they are.

The detailed notification dialog reflects change merging when it is initially displayed. Hence, the ‘Title’, ‘Category’, and ‘Remind’ fields in Figure 166 contain the values most recently entered by the user, not the older values that appeared in the scheduler’s view in Figure 157. Any user edits made in the notification dialog are reflected in the ultimately accepted changes. In this case, ‘Duration’ is so edited. Hence, when the user executes ‘View Item’ after pressing ‘Accept’ in Figure 167, the system displays the item-level view shown in Figure 168. This item-level view contains exactly the same data values as Figure 167. The item view has a full date description at the top, with ‘Change’ and ‘Delete’ command buttons instead of ‘Accept’ and ‘Decline’.

The policy to merge only the most recent changes may not be optimal in all circumstances. For example, if the user changes a data field and the scheduler subsequently changes the same field, the user might prefer her own less-recent

Scheduled Meeting [] [x]

◀ Today ▶ 8 AM Friday, September 23, 1998

Title:

Date: Start Time:

End Date: Duration: hr min

Recurring? Interval: S M T W Th F S

Category: Security:

Location: Priority:

Remind? minutes before

Attendees:

Details:

UPDATED Agenda:

1. Presentation by campus safety coordinator
2. Duty assignments for Fall quarter
3. News of office remodeling project
4. Introduce new staff members
5. Promotions

Minutes:

Scheduled By: On:

Host:

Figure 168: Scheduler changes merged into meeting.

field value. Such circumstances are presumed to be uncommon, so the system provides no option to select a different merge policy. The user can view the change notification and item displays side by side to inspect the differences between scheduler and user versions of a changed meeting. The user can then edit any scheduler-changed fields in the notification dialog.

To be clear, there are two separate copies of a not-yet-accepted meeting for each user who has not yet accepted or declined scheduler changes. The copy shown in the item-level display is that stored in the user’s local calendar, without any not-yet-accepted scheduler changes. The copy shown in the change notification detail is that of the scheduler, reflecting any scheduler-initiated changes that may have been made but not yet accepted. The scheduler-changed copy is stored in the Calendar Tool central repository. Details of the central repository are covered in Section 2.14.

To undo any edits made after the change confirmation dialog is initially displayed, the user can press ‘Clear’. As illustrated in Figures 166 and 167, the ‘Clear’ button is disabled until the user performs some editing.

When the detailed notification dialog is displayed, the initial smaller notification dialog remains active. The user can press 'Accept' or 'Decline' in either dialog. When the user does so, the active dialog is removed from the screen, and the other dialog, if displayed, is also removed.

When a scheduler changes a recurring meeting, the notification dialog text is one of the following, based on the scheduler's setting of how many instances to change:

**"User *user ID* has changed one instance
of the recurring meeting on *date*"**

**"User *user ID* has changed all instances
of the recurring meeting on *date*"**

**"User *user ID* has changed all future instances
of the recurring meeting starting on *date*"**

The *user ID* is the ID of the scheduling user. In the one-instance and future-instance messages, *date* is the instance date. For the all- instances message, *date* is the start date.

When the scheduler deletes a meeting, attending users are notified with a dialog of the form shown in Figure 169. This is the notification for the meeting scheduled in Section 2.4.1.4. For deletion of a recurring meeting, the dialog text is changed in the same way as for a change notification, with the word "changed" replaced with "deleted".

To view details of the scheduler-deleted meeting, the user presses 'View ...' in Figure 169. In response, the system displays the dialog in Figure 170. To the left of the command buttons is a label explaining that the buttons apply to a deletion performed by the scheduler. The user may edit any data fields except 'Scheduled By', 'On', or 'Host'. Performing edits is only meaningful if the user plans to decline the deletion. The user may choose to do so if she wants to retain a copy of a cancelled meeting.

If the user presses 'Accept', the system deletes the meeting from the user's calendar, updates all affected displays, and removes the delete notification dialog(s) from the screen. If the user presses 'Decline', no deletion is performed. If the user has edited one or more data fields prior to pressing 'Decline', the system displays a change confirmation of the form shown in Figure 172 of Section 2.5.2.5. If the user confirms the change, the system removes the notification from the screen. If the user cancels the change, the notification dialog remains, with no action having been taken.

Prior to an accept/decline decision for a scheduler changed or deleted meeting, the meeting appears in penciled-in form. As discussed in Section 2.4.1.5, the penciled-in format appears in daily, weekly, and monthly calendar views, as well as in all list views. The format is illustrated in Figure 114. In item-level calendar views, meetings with not-yet-accepted scheduler changes or deletion are indicated by additional command buttons at the bottom of the display. Details of this are presented in Section 2.5.2.5

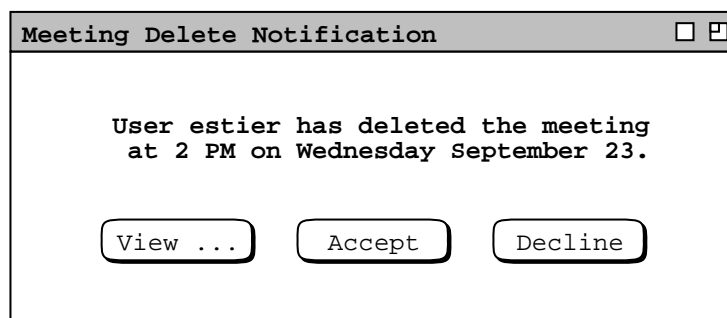


Figure 169: Delete notification dialog.

Meeting Delete Notification Detailed View □ □

Title:

Date: Start Time:

End Date: Duration: hr min

Recurring? Interval: S M T W Th F S
□ □ □ □ □ □ □

Category: Security:

Location: Priority:

Remind?

Attendees:
 △ □ ▾

Details:
 △ □ ▾

Minutes:

Scheduled By: On:
Host:

Scheduler Deletion:

Figure 170: Viewing details of a meeting delete notification.

The appearance of change/delete notifications is controlled by the setting of the meeting notification option described in Section 2.4.1.5. Also described in that section is the option that controls the appearance of penciled-in items. The system dynamically updates all active displays that are affected by a scheduler meeting change. An "affected" display is any attending user's display that shows the item prior to being changed. In all such displays, the item is dynamically changed from normal format to penciled-in format. In item-level displays, scheduler changes are indicated with the dynamic appearance of extra command buttons. The format of these item-level displays is described in Section 2.5.2.5. Further details of dynamic view updating are covered in Section 2.5.6. When the 'hide' setting is selected, meeting items changed by the scheduler appear in normal form, not penciled-in.

It is possible for a scheduler to produce multiple change or delete notifications before an attending user accepts or declines any of them. The system retains only a single change/delete notification for any given meeting. The retained notification corresponds to the most recently confirmed change or delete executed by the meeting's scheduler. To be clear, there is a single notification for any particular meeting, corresponding to the most recent change or delete operation. There is not a separate notification for change and for delete. By the nature of the delete operation, once confirmed it is the last notification produced by the scheduler. Hence, once a delete notification is sent, no further notifications will follow. The definition of a "particular meeting" is based on the unique identification provided by its 'Scheduled By', 'On', and 'Host' values.

Pending change and delete notifications are stored in the same queue as initial scheduling notifications, as described in Section 2.4.1.6.8. Change and delete notifications are commingled with initial scheduling notifications, in confirmation time order. For a change or delete operation, the confirmation time is defined as the time at which the scheduler presses the 'OK' button in the change or delete confirmation dialog. For any given meeting, the system removes any existing notification from the queue before entering a later notification. If a notification for a particular meeting is sent to a user at the same time the user is viewing a previous notification for that meeting, the newer notification is not added to the queue until the user makes the accept, decline, or delay decision for the earlier notification. If the circumstance arises where a scheduler sends two or more notifications while a user continues to view a previous notification, the system maintains the policy that at most one change/delete notification appears in the queue for any given meeting. When the user ultimately acts on an earlier notification, the system adds the single most recent notification to the user's queue.

After the user makes the accept, decline, or delay decision in any notification dialog, the notification is never redisplayed. If the start date and time of the meeting pass before the dialog is allowed to appear, the notification is removed from the queue and the system never displays it.

2.5.2.4. Group Calendar Changes

When a meeting scheduler confirms a change or delete, the effects are reflected immediately in all attending group calendars. As explained in Section 2.4.1.6.1, only a group leader can schedule a meeting for a group. Accordingly, only changes made by a group leader affect the group calendar.

The set of categories defined for a group calendar consists of the union of the categories for all items appearing in the calendar. When a scheduler changes or deletes the category for a meeting, the change or deletion is reflected in the set of defined categories as necessary. Specifically,

- a. if a category change leads to the definition of a new category, that category is added to the group calendar;
- b. if a meeting deletion is for the last group calendar item that contains a particular category, that category is deleted from the group calendar.

Group calendars are "virtual" in the sense that they are not directly editable. Change and delete on group calendars is effected by a group leader when the leader changes or deletes on the leader's own calendar. Calendar Tool system administrators can purge items from a group calendar, as described in Section 2.6.7.1.3.

2.5.2.5. Non-Scheduler Changing and Deleting Meetings

Non-scheduler change or delete of a meeting is performed in basically the same manner as appointment change or delete. Non-scheduler change or delete affects only an individual user's local calendar and no attendee notifications are sent. As noted earlier, the item for any meeting is stored as an individual copy in each user's local calendar. This means that scheduler and non-scheduler changes are performed on separate copies of the same meeting. The scheduler and non-scheduler copies are merged whenever an attending user accepts a scheduler's changes. What it means to have separate copies of the "same meeting" is that values of the 'Scheduled By', 'On', 'Host' fields are the same. The set values for these three fields constitutes the unique identification of a meeting on all users' calendars. This is the reason why these values cannot be changed after a meeting has been scheduled.

The only meeting fields that a non-scheduler can never edit are 'Scheduled By' and 'On'. All other fields are editable, subject to the following conditions:

- a. the general data-entry rules in Section 2.11.7 must be followed

- b. the edit restrictions for recurring appointments in Section 2.5.2.1 must be followed;
- c. if an item has not-yet-accepted scheduler changes, the data fields that the scheduler has changed are not editable until the user accepts or declines the scheduler changes.

Of note is that the scheduler editing restrictions in Table 9 do *not* apply to non-schedulers.

Given the reasonably liberal policy for non-scheduler edits, it is possible for a user to make imprudent changes to a meeting. For example, changing the date or location is typically not prudent, unless the user has information of the change from outside the context of the Calendar Tool. When a non-scheduler executes the change operation, the system displays a general warning in the confirmation dialog, but imposes no further restrictions on the change. The alternative to this liberal change policy would involve a potentially complicated definition of what constitutes "legitimate" change to a meeting by a non-scheduler. The developers of the Calendar Tool requirements decided to avoid the complications of such a definition.

Change and delete operations are executed from the item-level view of a meeting. Examples of these views are shown in Section 2.5.1.2. The examples in that section show regularly scheduled as well as not-yet-accepted meetings. There are two additional forms of item view for meetings with not-yet-accepted scheduler changes or deletion. Items with not-yet-accepted scheduler changes appear in the form illustrated in Figure 171. This is the item-level view of the changed staff meeting before the changes have been accepted by the user, and before any changes have been made in the change notification dialog. That is, it is the result of a 'View Item' command for the meeting changed by the scheduler in Figure 157, before the user has performed any action in the change notification dialog.

The display in Figure 171 contains the same information as the regular item-level view in Figure 137, with the addition of accept/decline command buttons at the bottom. For added clarity, the banner of the display is augmented with the "**SCHEDULER CHANGED**" text. The additional command buttons have the same functionality as in a change notification dialog, such as Figure 166. That is, the buttons allow the user to accept, decline, or view details of the scheduler's changes.

The data values in Figure 171 are those most recently entered by the user, independent of the scheduler changes. Hence, the not-yet-accepted changes to the 'Duration' and 'Details' fields do not appear. To see these changes, the user can select the 'View ...' command, in response to which the system displays the detailed change notification dialog of Figure 166.

To prohibit conflicting changes, the system disables editing in data fields that have scheduler changes not yet accepted by the user. Hence, the 'Duration' and 'Details' fields are disabled in Figure 171. These are the same fields shown in bold red type in the unedited notification dialog of Figure 166. The user can edit and confirm changes to any of the other editable data fields in Figure 171. The purpose of disabling scheduler-changed data fields in the item view is to force the user to edit such fields in the notification dialog, where they are clearly indicated as having been changed by the scheduler.

When the user presses 'Change' in the item view of a non-recurring meeting, the system displays the confirmation dialog in Figure 172. The dialog informs the user that changing an individual copy of a meeting is likely to result in differences with other users, namely the scheduler and other attendees. It is "likely" in the sense that the user's version will differ from other users unless all of them make exactly the same changes. When the user presses 'OK', the system proceeds with the change on the user's local copy and updates all affected displays. Pressing 'Cancel' in the confirmation dialog results in no calendar or display changes.

The change confirmation dialogs for recurring meetings differ in the first sentence of the dialog text. The first sentences are:

**"Are you sure you want to change this instance
of the recurring meeting?"**

**"Are you sure you want to change all instances
of the recurring meeting?"**

**"Are you sure you want to change this instance and
all future instances of the recurring meeting?"**

the appropriate one of which appears based on the user's setting of how many instances to change.

Scheduled Meeting -- SCHEDULER CHANGED
□ □

◀ Today ▶
8 AM Friday, September 23, 1998

Title:

Date: Start Time:

End Date: Duration: hr min

Recurring? Interval: S M T W Th F S

Category: Security:

Location: Priority:

Remind?

Attendees:

▲
□
▼

Details:

Agenda:

1. Duty assignments for Fall quarter
2. News of office remodeling project
3. Introduce new staff members
4. Promotions
5. Initial prep for accreditation visits

▲
□
▼

Minutes:

Scheduled By: On:

Host:

Scheduler Changes:

Figure 171: Item-level view of meeting with not-yet-accepted scheduler changes.

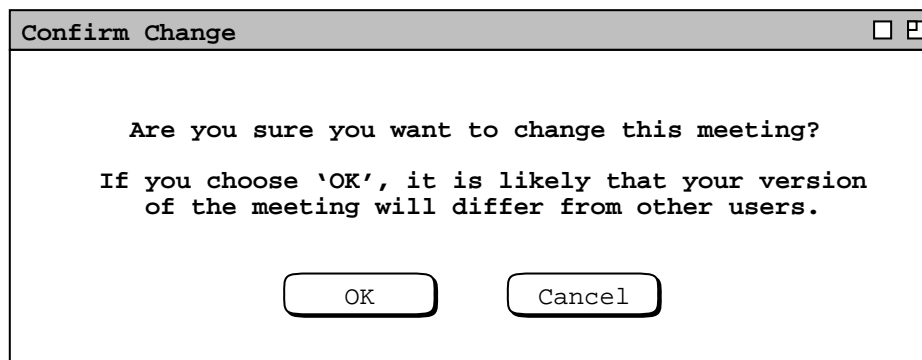


Figure 172: Non-scheduler change confirmation dialog.

When a user views an item with not-yet-accepted scheduler deletion, the display appears as shown in Figure 173. As in Figure 171, there are both change/delete and accept/decline command buttons. In this case, the label beside the accept/decline buttons indicates that they apply to a not-yet-accepted scheduler deletion. For added clarity, the banner of the display is augmented with the "**SCHEDULER DELETED**" text.

The two groups of command buttons in Figures 171 and 173 correspond to the two different copies of a meeting with not-yet-accepted scheduler changes or deletion. The change/delete buttons operate on the pre-accept copy. The accept/decline buttons operate on a copy reflecting the scheduler's not-yet-accepted changes.

Meeting deletion by a non-scheduler is processed in the same manner as appointment deletion, as explained in Section 2.5.2.1. The only difference is that the word "appointment" in the confirmation dialogs of Figures 155 and 156 is replaced with the word "meeting".

After a non-scheduler deletes a meeting, that user will not receive any subsequent scheduler change or delete notifications for the meeting.

2.5.2.6. Changing and Deleting Tasks

Changing and deleting tasks is much the same as appointments. The task fields that differ from an appointment are:

- a. 'Priority', which is numeric for a task
- b. 'Completed' and 'Completion Date', which are not present in appointments

Editing functionality for all other fields is the same as for appointments. Figure 174 shows the user having edited an instance of the recurring task shown in Figure 140. Here the 'Completed' check box has been turned on by the user. When the user turns 'Completed' on, the system responds immediately by entering today's date in the 'Completion Date' field. The user may edit the completion date to any legal, non-empty date value. The system imposes no range restrictions on the completion date, meaning a task can be completed any time before, on, or after its due date. In the case of Figure 174, the user changes the system-entered value of September 23 to September 21 and presses 'Change'. If the user turns 'Completed' off after having turned it on, the system clears and disables the 'Completion Date' field.

The change and delete confirmation dialogs for tasks are the same as they are for appointments, with the word "appointment" replaced with "task". When a task change or delete is confirmed, the calendar and display affects are as described for appointments.

2.5.2.7. Changing and Deleting Events

As with the other types of item, change and delete of an event is executed in the item-level display. For example, Figure 175 shows the user having edited the event scheduled in Figure 141. The edits are a change of the start date from September 23 to September 25, and a change from private to public security. When the user presses 'Change', the system displays the dialog in Figure 176. The event delete confirmation dialog is the same as that for change, with the word "change" replaced with "delete". Since events do not recur, there is only a single form of

Scheduled Meeting -- SCHEDULER DELETED

Title:

Date: Start Time:

End Date: Duration: ^{hr} ^{min}

Recurring? Interval: S M T W Th F S

Category: Security:

Location: Priority:

Remind?

Attendees:

Details:

Minutes:

Scheduled By: On:

Host:

Scheduler Deletion:

Figure 173: Item-level view of meeting with not-yet-accepted scheduler deletion.

Scheduled Task [] []

◀ Today ▶ 9 AM Friday, September 21, 1998

Title: Colloquium announcement

Instance Date: September 21, 1998 Due Time: 9 AM

End Date: November 30, 1998

Recurring? Interval: weekly S M T W Th F S

Category: department Security: public
 Priority: 8

Remind? 1 hour before on screen

Details:
 If there's a colloquium next week, email and post the announcement.

Completed? Completion Date: September 23, 1998

Apply Change and Delete to:
 this instance all instances future instances

Change Delete Clear

Figure 174: Edits made to a recurring task.

Scheduled Event [] []

◀ Today ▶ September 23, 1998

Title: Jim's Birthday

Start Date: September 25, 1998 End Date:

Category: special event Security: public

Change Delete Clear

Figure 175: Birthday event changed.

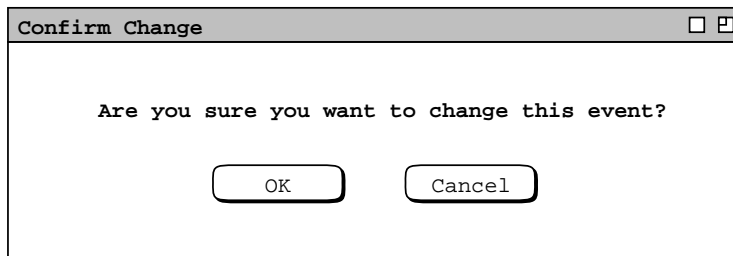


Figure 176: Event change confirmation dialog.

change and delete confirmation dialog. There is no distinction between single-day versus multi-day events in either confirmation.

When the user confirms an event change or delete, the user’s calendar and active display windows are updated accordingly. Change of a multi-day event affects all of its days. Deletion of a multi-day event deletes all of its days. The effect of deleting one or more days of a multi-day event is achieved by changing the start or end date values. The start and end dates of a multi-day event always represent a contiguous range of days. Hence, the system provides no means to delete one or more middle days of an event, between but not including a start or end date.

The user cannot change or delete the system-defined events presented in Section 2.11.16 There is an option to hide these events if the user does not want to see them.

2.5.3. Details of Scheduling Recurring Items

The scheduling of weekly and biweekly recurring items has been presented in preceding scenarios. This section presents the remaining details of monthly and yearly recurring intervals.

When the user checks the *Recurring* box in a scheduling dialog or item-level view, the data-entry boxes to the right become active. Figure 177 shows the general format. The ‘Interval’ menu in Figure 178 shows the four available selections: weekly, biweekly, monthly, and yearly. The details area to the right of the interval menu varies based on the menu selection and the scheduling context. In all contexts except a meeting request, the details area has three possible configurations:

- a. for weekly and biweekly intervals there are check boxes for the days of the week;
- b. for the monthly interval there is a single ‘Details ...’ button;
- c. for the yearly interval the details area is empty.

In the scheduling request dialog for a meeting, the details area always contains the ‘Details ...’ button,



Figure 177: Information for a recurring item.

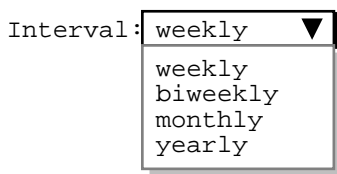


Figure 178: Interval selections.

whatever the interval selection. Descriptions of the different details configurations and dialogs follow.

2.5.3.1. Weekly/Biweekly Intervals

The default recurring interval is weekly. When either the weekly or biweekly interval is selected, the interval details area contains seven check boxes, one for each day of the week. When the user checks one or more of the day boxes, the system schedules the item to recur on the selected day(s). For a weekly interval, the item recurs each week on the selected days. For biweekly, the item recurs every other week on the selected days. At least one of the day boxes must be checked. Examples of weekly and biweekly scheduling appear in Sections 2.2, 2.4, and 2.5.

2.5.3.2. Monthly Intervals

When the user selects the monthly interval, the system changes the recurring information display as shown in Figure 179. When the user presses ‘Details ...’, the system displays the dialog in Figure 180. The dialog allows the user to schedule a recurring item on one or more months of the year, selected via the check boxes. Below the month selections are two alternatives for choosing the days on which the item recurs. The ‘Days of the week’

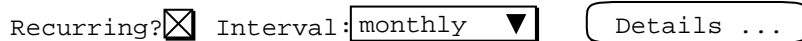


Figure 179: Monthly interval selected.

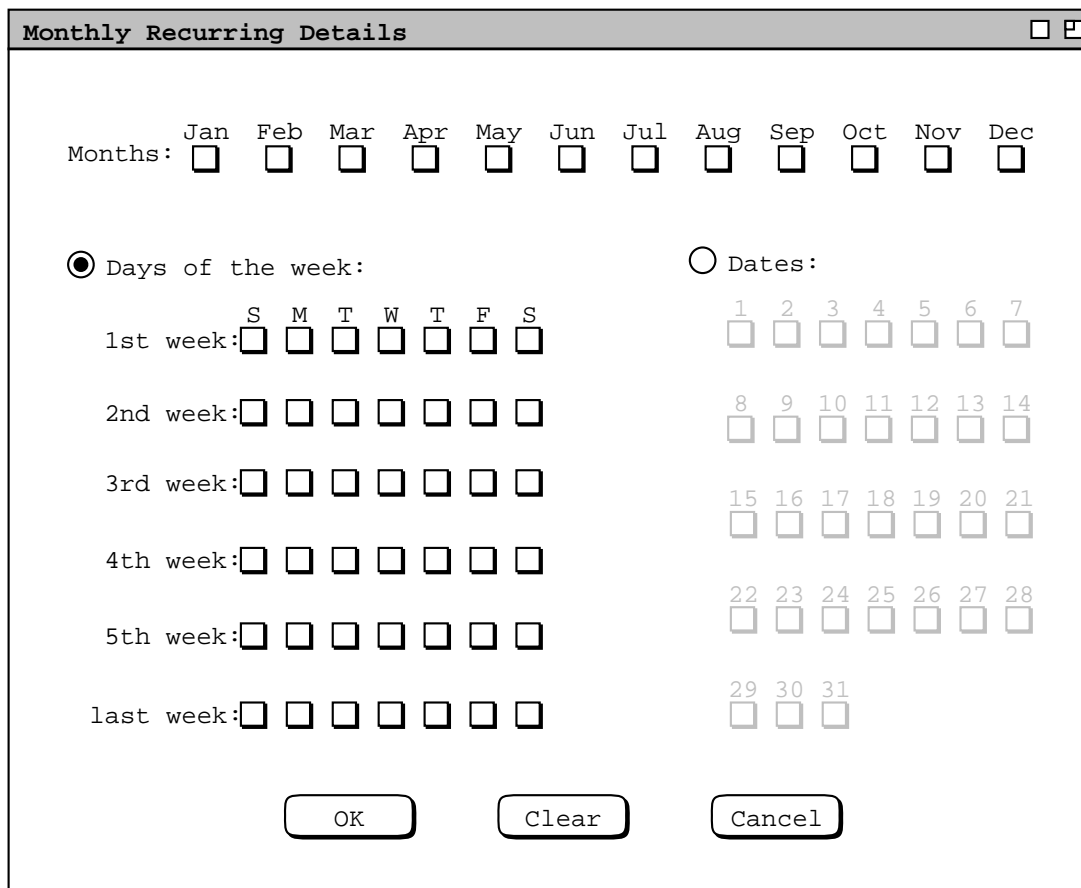


Figure 180: Monthly interval scheduling dialog.

alternative allows the user to select one or more days within the different weeks of the month. The 'Date' alternative allows the user to select one or more numeric dates. Only one of the two alternatives is active at one time. For convenience, check box selections made in either alternative remain checked when the other alternative is activated.

Figure 181 shows settings for an item that recurs on the first Monday of every other month, starting in January. Figure 182 shows an item that recurs on the first and third Friday of every month. An item scheduled in this way recurs at most two times per month. In particular, it does not occur on any fifth Fridays, when they happen. Figure 183 uses the 'Dates' selection to schedule a meeting on the 15th of every third month, starting in January.

The monthly details dialog contains generic information, not applicable in all cases. The following restrictions apply to the 'Days of the week' settings:

- a. For months with four weeks only, the specification for the 5th week does not apply, and the specification for the last week, if present, overrides any specification for the 4th week, if present.
- b. For months with five weeks, the specification for the last week, if present, overrides any specification for the 5th week.

Since the number of weeks per month varies dynamically from year to year, it is not possible to consistently disable any particular row of check boxes under 'Days of the week'. Hence, all day boxes are enabled whenever 'Days of the week' is selected. The two override rules are applied dynamically when an item is scheduled.

Under 'Dates', the check boxes for 30 and 31 are selectively disabled based on which months are selected. Specifically, both 30 and 31 are disabled if only 'Feb' is selected. 31 is disabled if one or more of 'Apr', 'Jun', 'Sep',

Monthly Recurring Details

Months: Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

Days of the week: Dates:

1st week:	S <input type="checkbox"/>	M <input checked="" type="checkbox"/>	T <input type="checkbox"/>	W <input type="checkbox"/>	T <input type="checkbox"/>	F <input type="checkbox"/>	S <input type="checkbox"/>	1 <input type="checkbox"/>	2 <input type="checkbox"/>	3 <input type="checkbox"/>	4 <input type="checkbox"/>	5 <input type="checkbox"/>	6 <input type="checkbox"/>	7 <input type="checkbox"/>
2nd week:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	8 <input type="checkbox"/>	9 <input type="checkbox"/>	10 <input type="checkbox"/>	11 <input type="checkbox"/>	12 <input type="checkbox"/>	13 <input type="checkbox"/>	14 <input type="checkbox"/>
3rd week:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	15 <input type="checkbox"/>	16 <input type="checkbox"/>	17 <input type="checkbox"/>	18 <input type="checkbox"/>	19 <input type="checkbox"/>	20 <input type="checkbox"/>	21 <input type="checkbox"/>
4th week:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	22 <input type="checkbox"/>	23 <input type="checkbox"/>	24 <input type="checkbox"/>	25 <input type="checkbox"/>	26 <input type="checkbox"/>	27 <input type="checkbox"/>	28 <input type="checkbox"/>
5th week:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	29 <input type="checkbox"/>	30 <input type="checkbox"/>	31 <input type="checkbox"/>				
last week:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>							

OK Clear Cancel

Figure 181: Item recurs on the first Monday of every other month.

Monthly Recurring Details □ □

Months: Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

Days of the week: Dates:

1st week: S <input type="checkbox"/> M <input type="checkbox"/> T <input type="checkbox"/> W <input type="checkbox"/> T <input type="checkbox"/> F <input checked="" type="checkbox"/> S <input type="checkbox"/>	1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5 <input type="checkbox"/> 6 <input type="checkbox"/> 7 <input type="checkbox"/>
2nd week: <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	8 <input type="checkbox"/> 9 <input type="checkbox"/> 10 <input type="checkbox"/> 11 <input type="checkbox"/> 12 <input type="checkbox"/> 13 <input type="checkbox"/> 14 <input type="checkbox"/>
3rd week: <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> F <input checked="" type="checkbox"/> S <input type="checkbox"/>	15 <input type="checkbox"/> 16 <input type="checkbox"/> 17 <input type="checkbox"/> 18 <input type="checkbox"/> 19 <input type="checkbox"/> 20 <input type="checkbox"/> 21 <input type="checkbox"/>
4th week: <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	22 <input type="checkbox"/> 23 <input type="checkbox"/> 24 <input type="checkbox"/> 25 <input type="checkbox"/> 26 <input type="checkbox"/> 27 <input type="checkbox"/> 28 <input type="checkbox"/>
5th week: <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	29 <input type="checkbox"/> 30 <input type="checkbox"/> 31 <input type="checkbox"/>
last week: <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	

Figure 182: Item recurs on the first and third Fridays of every month.

Monthly Recurring Details

Months: Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

Days of the week:

1st week: S M T W T F S

2nd week:

3rd week:

4th week:

5th week:

last week:

Dates:

1 2 3 4 5 6 7

8 9 10 11 12 13 14

15 16 17 18 19 20 21

22 23 24 25 26 27 28

29 30 31

OK Clear Cancel

Figure 183: Item recurs quarterly on the 15th.

or 'Nov' are selected. The 29 selection applies to 'Feb' only on leap years.

2.5.3.3. Yearly Intervals

The yearly interval setting means that an item recurs once every year on the same date. For example, Figure 184 shows an appointment that recurs yearly on August 27, from 1998 through 2048. The 'Start Date' and 'End Date' must be the same month and date. The recurring details area is empty. Although yearly recurrence can be specified using monthly details, the yearly form is provided as a simple convenience.

2.5.3.4. Recurring Meeting Requests

As discussed in Section 2.4.1, a meeting request specifies a range of possible times and dates on which a meeting can be held. In this context, the recurring details specify *allowable* days on which the meeting can occur, rather than specific days on which the meeting does occur.

For weekly and biweekly intervals, recurring meeting requests are covered in Section 2.4.1.2. For the monthly interval, the recurring details dialog is shown in Figure 185. In the top row of the dialog, the user selects one of three alternatives for granularity of recurrence. At the far right of the first row is the value for the selected granularity. The default granularity is 1 day per month.

The 'Allowable months' check boxes are used to specify the months on which a meeting is allowed to occur. The default is all months.

'Allowable days' and 'Allowable dates' are two alternatives for setting the specific days or dates on which the meeting is allowed to occur. The use of these alternatives is similar to the regular monthly recurring

Schedule an Appointment □ □

Title:

Start Date: Start Time:

End Date: Duration: hr min

Recurring? Interval: ▼

Category: ▼ Security: ▼

Location: ▼ Priority: ▼

Remind? ▼ ▼

Details:

Forget this and you're toast.

Figure 184: Anniversary appointment.

Details for Monthly Recurring Meeting Request □ □

Days per month:
 Weeks per month:
 Months per year:

Allowable months:
 Jan
 Feb
 Mar
 Apr
 May
 Jun
 Jul
 Aug
 Sep
 Oct
 Nov
 Dec

Allowable days:
 Allowable dates:

1st week:	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2nd week:	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3rd week:	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4th week:	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5th week:	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
last week:	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				

Minimum number of days between each occurrence:

Maximum number of days between each occurrence:

Figure 185: Monthly recurring details for a meeting request.

dialog in Figure 181. In the case of the meeting request, the settings select *possible* days or dates rather than actual days or dates.

The last two lines in Figure 185 define the minimum and maximum span between meeting occurrences. The use of these settings is analogous to the last two settings in the weekly recurring request shown in Figure 93. The occurrence granularity changes based on the setting in the first-row radio buttons. The figure shows the default case of "days". If the 'Weeks per month' selection is on, the string "days" is replaced with "weeks" in both the minimum and maximum labels. If 'Months per year' is on, the string "days" is replaced with "months" in both labels. The minimum and maximum fields are enabled if the numeric value in the first row is greater than the number of allowable days, weeks, or months, as appropriate. The number of allowable days is defined as the number of day or date check boxes that are selected. The number of allowable weeks is defined as the number of rows under 'Allowable days' or 'Allowable dates' that have one or more days or dates selected. The number of allowable months is the number of month check boxes selected.

2.5.3.5. Recurring Items and Instances

When a recurring item is scheduled, it generates a collection of *instances* in the user's calendar. The instances occur on the dates specified by the recurring information. The system keeps track of the instance collections as a whole,

for the purposes of change and delete. Each instance that is generated from a particular recurring item is marked as part of that item's collection. In this way, when the user changes or deletes all instances of a recurring item, the system knows precisely which calendar items are affected.

Since a recurring item cannot be changed to non-recurring, an instance always remains part of its recurring collection, unless the instance is deleted. Hence, when an individual recurring instance is changed, it still remains part of its recurring collection. This means that a changed instance can continue to be used for changing or deleting all other or all future instances. The cumulative effect of individual instance changes is reflected whenever an instance is used to change all other or future instances. For example, suppose the user confirms two separate change operations to an individual item, then selects 'all instances' and confirms a third change. In this case, the results of all three changes are made to all instances, not just the result of the third change after 'all instances' is selected.

To maintain a reasonable degree of consistency for recurring items, all instances in the same collection must have the same start date, end date, and recurring information. For this reason, the start date, end date, and recurring information can only be changed when the 'all instances' radio button is selected in an item-level view.

When the user changes the start date, end date, or recurring details for all instances, the effect can be to delete one or more instances. Specifically, the following forms of change result in instance deletion:

- a. changing the start date forward, to after the existing start date
- b. changing the end date backward, to before the existing end date
- c. removing one or more days from the recurring details

When the user presses the 'Change' button in an item-level display after making one or more such changes, the system displays a confirmation dialog of the form shown Figure 186, where N is the number instances that will be deleted if the change is confirmed. If the user confirms with 'OK', the effects of the change are reflected in all affected display windows, including the removal of the item-level display for the changed instance if the change results in its deletion.

2.5.4. Receiving Reminders

The user receives a reminder for each item with 'Remind' turned on. Reminders appear in one of two forms: on-screen or email. On-screen reminders appear anytime the Calendar Tool host computer is running, whether or not the Calendar Tool itself is running. For example, when it is time for the dentist appointment reminder (Figure 145), the system displays the on-screen dialog shown in Figure 187. The reminder text indicates the start time, end time, and title of the scheduled appointment. An audible alert accompanies the on-screen display. When the user responds by pressing 'OK', the system removes the dialog from the screen and the reminder will not appear again. For recurring items, a reminder appears for each instance in which 'Remind' is turned on.

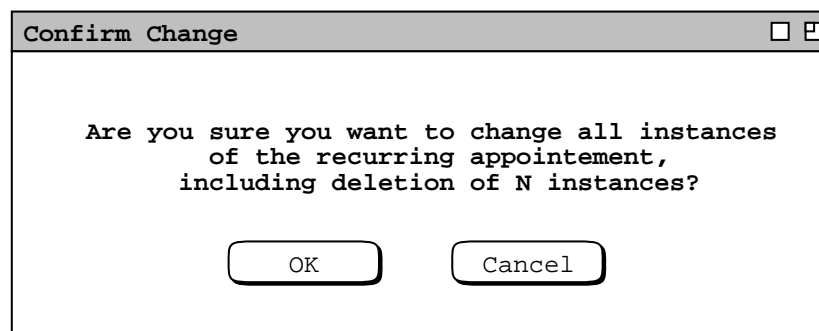


Figure 186: Confirmation dialog for deleting instances by date or recurring details changes.

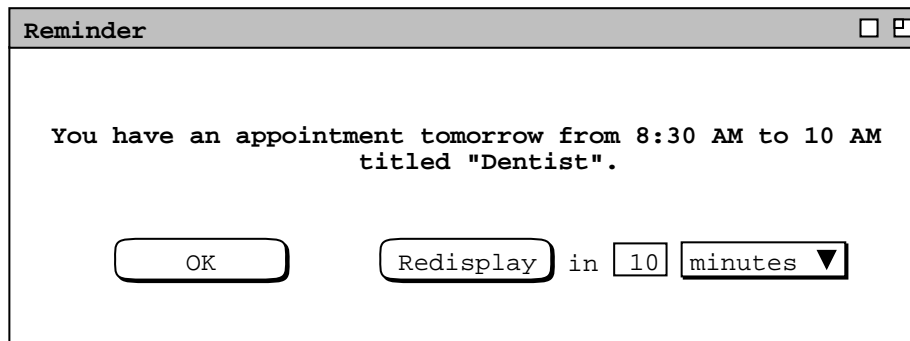


Figure 187: Dentist appointment on-screen reminder.

The reminder text varies depending on how soon before the appointment the reminder is to appear. Specifically,

- a. If the reminder appears on the same date as the appointment is scheduled, the word "today" appears before the time.
- b. If the reminder appears on the day before the appointment is scheduled, the word "tomorrow" appears before the time. This is the case in Figure 187, because the reminder has been set to appear 24 hours before the appointment.
- c. If the reminder appears two or more days before the appointment is scheduled, the month name and numeric date appear before the time.

If the user responds with 'Redisplay', the system removes the reminder dialog from the screen and redisplay it in the indicated amount of time. The default time is 10 minutes, which the user can change to a value between 1 minute and 99 hours. The menu to the right of the numeric data field contains the selections 'minutes' and 'hours'. The default value of 10 minutes can be set as an option, as described in Section 2.7.3

The same general form of reminder is sent for all types of scheduled item. The difference is in leading several words of the reminder text. The four text forms are:

"You have an appointment ..."

"You have a meeting ..."

"You have a task due ..."

"There is an event ..."

The ellipses are the same in all cases, consisting of the today/tomorrow/date indicator, start time, end time, and title.

When the 'email' setting is selected for a reminder, the system sends an electronic mail message to the user at the time the reminder is scheduled to appear. The subject of the message is "Calendar Tool Reminder". The body of the message is the same text that appears in an on-screen reminder, left justified in 72 character columns.

The Calendar Tool sends email reminders using the electronic mail facilities of the underlying operating environment. Once an email reminder has been sent, its further handling and delivery are beyond the control of the Calendar Tool.

2.5.5. Editing Categories

As introduced in Section 2.2, Calendar Tool categories are used to provide additional descriptive information for scheduled items. Categories apply uniformly to all four types of item. Any category can be applied to any number of appointments, meetings, tasks, or events.

Category definitions are accessible from scheduling dialogs, notification dialogs, and item-level displays. The category menu in all such contexts appears the same, as illustrated in Figure 188. If the category is 'none' is selected, the text area above the menu is empty. If any other category is selected, the text area contains that category's name, as shown in Figure 189. When the user selects 'Edit ...' in the categories menu, the system displays the dialog of Figure 190. This same dialog is also accessible from the 'Categories' item in the 'Schedule' menu.

Adding a new category was introduced in Section 2.2. The categories shown in Figures 188 through 190 reflect category additions having been made during the course of the item-scheduling operations of the preceding scenarios. When the user adds a new category, it must be different in name from any existing category. Names are compared on a character-by-character, case sensitive basis.

The user can change and delete existing categories. To change a category, the user selects its name in the 'Current Categories' list and presses 'Change'. For example, Figure 191 shows the user having selected the 'special event' category. The figure illustrates that 'Change' and 'Delete' become enabled when the user selects a category name. When the user presses 'Change', the system responds with the dialog in Figure 192. The user can edit the category name to any non-empty string that is not the same as an existing category name; the color can be changed to any available in the color menu. In Figure 193, for example, the user has shortened the category name and changed the color to red. When the user presses 'OK', the system responds with the confirmation dialog in Figure 194. The confirmation indicates the total number of items that the category change will affect, that is, the number of items that are defined with this category. In computing the item count, each instance of a recurring item is counted separately. When the user presses 'OK', the system proceeds to change the category in all affected items. The system also updates all active display windows in which the old category appears, replacing the old category name and/or color with the new name and/or color.

To delete an existing category, the user selects its name and presses 'Delete'. In response, the system displays a dialog of the form shown in Figure 195. Here the user has chosen to delete a category that is used in 15 scheduled

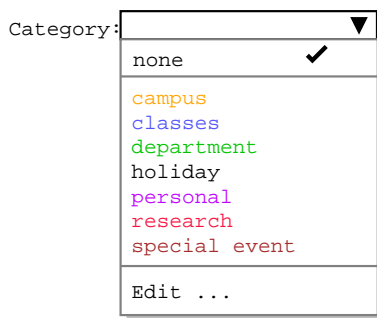


Figure 188: Category display menu.

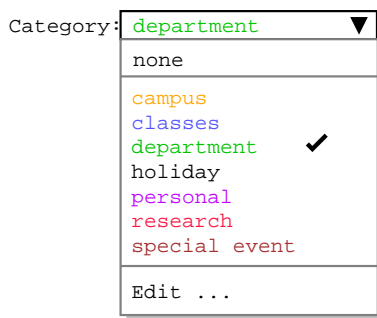


Figure 189: Category menu with non-empty category selected.

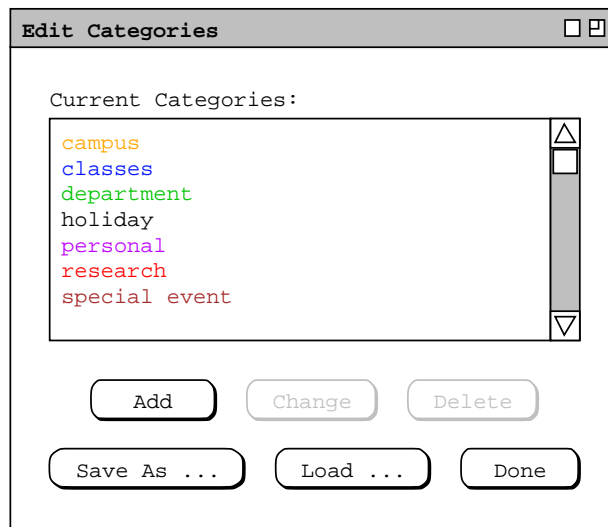


Figure 190: Edit categories dialog with several user-defined categories.

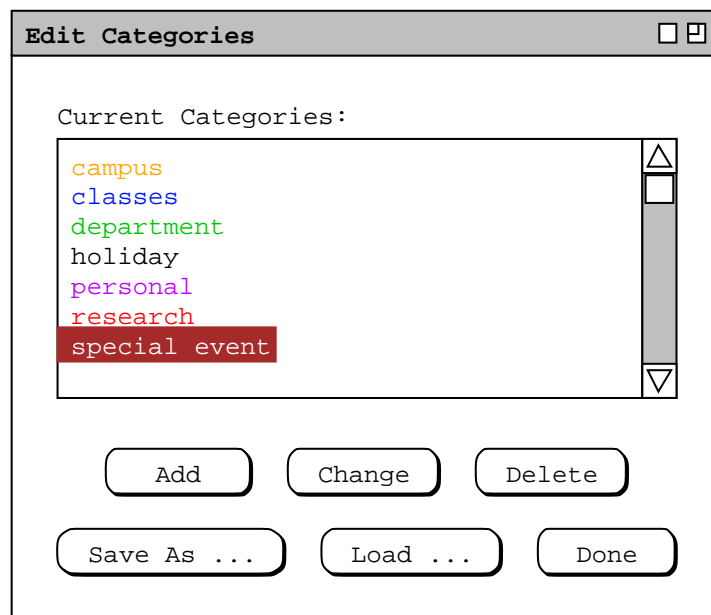


Figure 191: Special event category selected for editing.

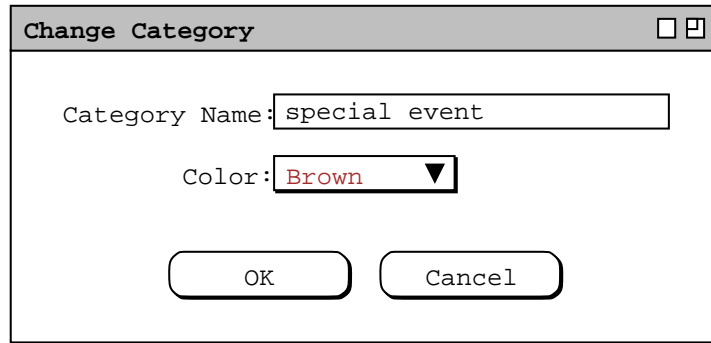


Figure 192: Change category dialog.

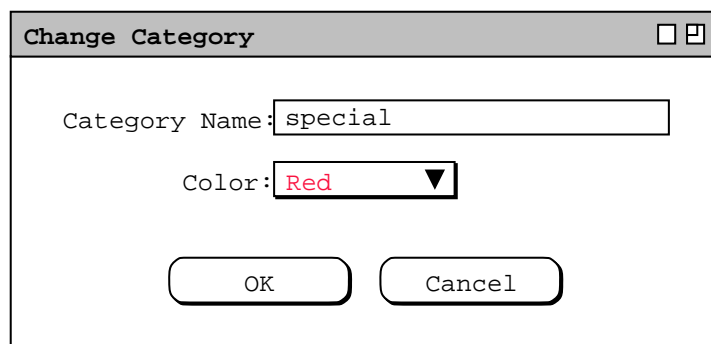


Figure 193: Change category dialog, edited.

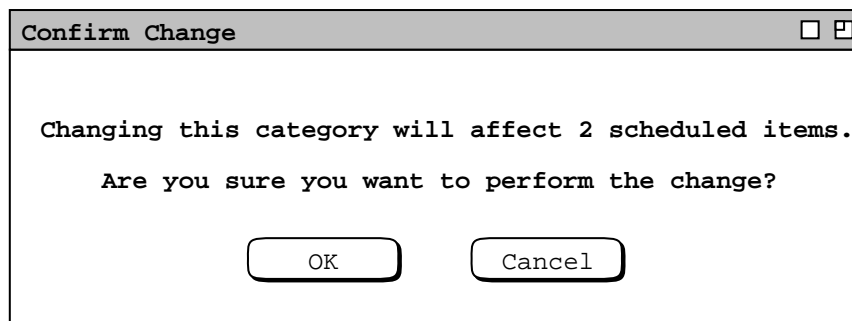


Figure 194: Category change confirmation dialog.

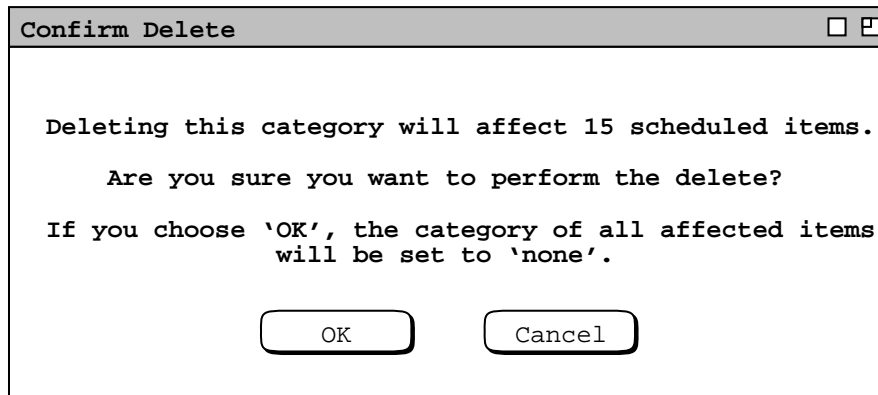


Figure 195: Category delete confirmation dialog.

items. As the dialog explains, the affected items will all have their categories set to 'none' if the delete is confirmed. When the user presses 'OK', the system deletes the category from the defined list, removes it from all affected items, and updates all active displays. The display updates entail changing the color of all affected items to black and removing the category name wherever it appears. Blank text appears in all displays where the category name appeared prior to deletion.

The 'Save As ...' and 'Load ...' buttons in the categories dialog allow the user to save and load copies of category definitions to and from files. Details of saving and loading auxiliary data files are covered in Section 2.8.6

Category definitions are calendar-specific. That is, each calendar has its own set of category definitions. When more than one calendar is open, the category definitions that appear in menus, displays, and dialogs are those of the current calendar. The specific contexts in which category definitions appear are the following:

- a. all four types of scheduling dialogs
- b. all four types of item-level displays
- c. the coloring in all calendar and list views
- d. meeting notification dialogs
- e. category editing dialogs

The system provides a separate category editing dialog for each calendar. Whenever the user executes the 'Schedule Categories ...' command, the system (re)displays a category dialog for the current calendar. Each calendar uses a single category dialog. This means that up to N category dialogs can be displayed at any time, for N equal to the number of open calendars. When more than one calendar is open, the calendar name appears by default in the banner of each category dialog. The appearance or non-appearance of the calendar name in window banners is controlled by an option setting, as described in Section 2.7.4.2.

2.5.6. Dynamic View Updating

Whenever a user confirms execution of a 'Schedule' or 'View' command, the visible effects of the command, if any, are immediately reflected in all affected display windows for that user, unless filter or option settings limit the visibility of the effects. The definition of "affected display window" is any active window the contents of which are altered by the command execution. The scenarios in the preceding sections describe the specific effects of commands in specific contexts.

When a meeting scheduler confirms an operation to schedule, change, or delete a meeting, the notification dialog and penciled-in changes appear immediately on the displays of all active attending users whose filter and option settings permit. If user option settings limit when the notification dialog can appear, it appears as soon as the option setting allows or is changed to allow. When a user hides penciled-in items, such items appear immediately after the user changes the pencil-in option to 'show'.

Display updates to other user and group calendars do not appear dynamically. For example, if user *A* is viewing user *B*, changes made by user *B* do not appear on user *A*'s screen until user *A* re-executes the command that generates the view of user *B*'s calendar. Similarly, if user *A* is viewing group calendar *G*, changes made by a leader of *G* do not appear on *A*'s screen until *A* re-executes the command that generates the changed group calendar view.

When an administrative user confirms execution of an 'Admin' command, display updates appear dynamically on the administrative user's display of the Calendar Tool Administration program. Also, any administrator-generated notifications appear dynamically on the affected users' displays. Administrative notifications are generated for adding, changing, and deleting user records, and by the 'Notify Users' command.

Aside from administrative notifications, changes to administrative data do *not* appear dynamically on affected users' screens administrator performing the change-effecting command. For example, if a user is viewing a record in the user database, changes made by an administrator to that record do not appear on the user's screen until the user re-executes the command that generates the user-record display. Complete details of administrator database changes are presented in Section 2.6.

When the user confirms category changes or applies option changes, the effects are reflected immediately in all affected displays. Details of category and option change effects are covered in Sections 2.5.5 and 2.7 respectively.

2.6. Administrative Functions

The scenarios in this section cover administrative commands. The commands provide access to the three Calendar Tool databases: users, groups, and locations. Regular users can access the databases with viewing commands. Administrative users can modify the databases with add, change, and delete commands. In addition to database access, there are other administrative commands for performing small managerial functions.

As described in Section 2.1, there are separate interfaces for two types of registered user -- regular and administrative. An administrative user is one with designated administrative privileges. Both regular and administrative users can run the Calendar Tool through the regular user interface shown in Figures 1 and 2. Only users with administrative privileges can run through the administrator interface shown Figures 3 and 4.

Privileged administrator access is covered in Sections 2.6.1 through 2.6.5 Access to administrative commands by regular users is covered in Section 2.6.6. In the scenarios, the term "administrator" refers to a user with administrative privileges; the term "user" refers to a regular user without administrative privileges. Also, the term "privileged Admin menu" refers to the command menu in the privileged administrator's interface; the term "Admin menu" refers to the menu accessible from the regular, non-privileged interface.

2.6.1. Gaining Administrative Access

When a user invokes the Calendar Tool Administration program, the system presents the password dialog shown in Figure 196. The user must enter a legal password in order to be confirmed as an administrator. If the user does so,

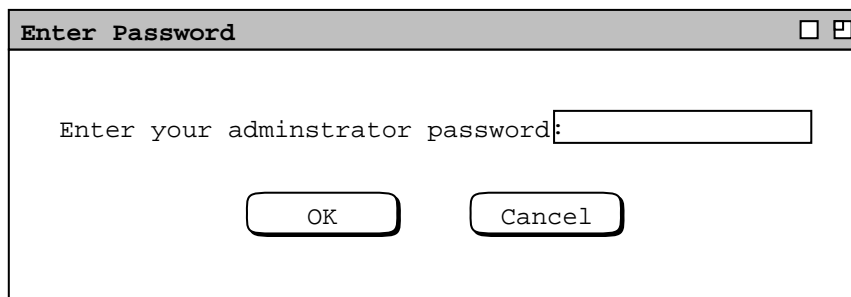


Figure 196: Password entry dialog.

the system displays the administrative interface of Figure 3. If the user fails to enter the correct password, the system denies access by presenting the dialog of Figure 197. The system displays this dialog for up to three additional attempts, after which the dialog is removed and the user is not permitted administrative access. As the password is typed, it is echoed as asterisk characters, not as the actual characters typed.

Each administrative user has a separate password, which is set with the password command described in Section 2.6.5. The Calendar Tool Administration program is installed with one pre-defined administrative user. The user name is "Calendar Tool Administrator", with ID "admin" and initial password "admin". This pre-defined administrator cannot be deleted, but its password can and should be changed as soon as the program is initially invoked.

Whenever an administrator invokes the Calendar Tool Administration program, the computer on which the invocation takes place is considered to be a Calendar Tool central host. The name of the computer is displayed in the banner of the Calendar Tool Administration menu bar, as shown in Figure 4. Further details on the role of central host computers in the overall Calendar Tool operating environment are covered in Section 2.14.

2.6.2. User Database

As shown in Figure 3, the default administrative interface has a dialog for accessing the Calendar Tool user database. The same dialog is displayed when the administrator selects the 'Users ...' item in the privileged 'Admin' menu. Figure 198 shows the state of the user database after a number of user records have been added. The scrolling area and search field operate in the same manner as described in Section 2.3.5.1 for viewing other users.

To add a new user to the database, the administrator presses the 'Add ...' button. In response, the system displays the dialog shown in Figure 199. The dialog contains all of the information in a user record. The 'Name' is entered separately as 'last', 'first', and 'middle'. Only 'last' is required. If 'middle' is non-empty, 'first' must be non-empty. The names appearing in the database list of Figure 198 are formatted as follows:

last, first middle

The comma and spaces are absent if there are no first and middle names. The last name can itself contain space characters. For example, the last name "Calendar Tool Administrator" is that of the initial pre-defined user in the user database.

The 'Calendar Tool ID' is the identifier by which a user is known in the Calendar Tool system. The ID must be unique among all Calendar Tool users and groups. The area code and phone number fields are optional information that may be useful for communicating with users outside the context of the Calendar Tool.

'Email' is required information for all users. It is the electronic mail address of the user for mailing meeting notifications and reminders. 'Local Host Computers' is a list of unique identifiers for the computer(s) on which the user runs the Calendar Tool. The format of the local host identifier is platform-dependent, based on the type of

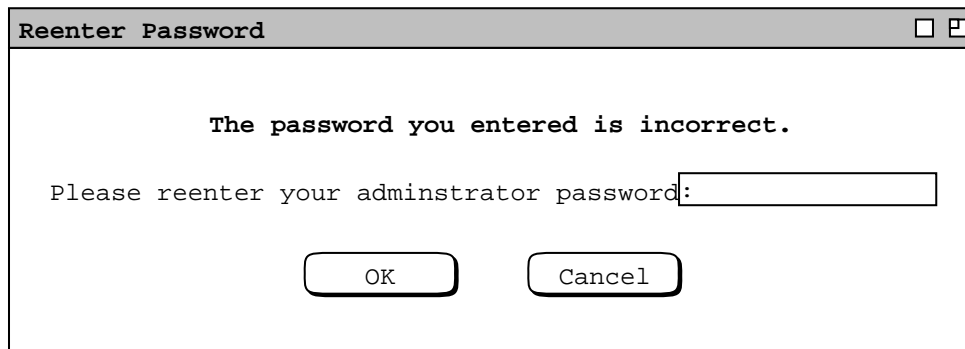


Figure 197: Password re-entry dialog.

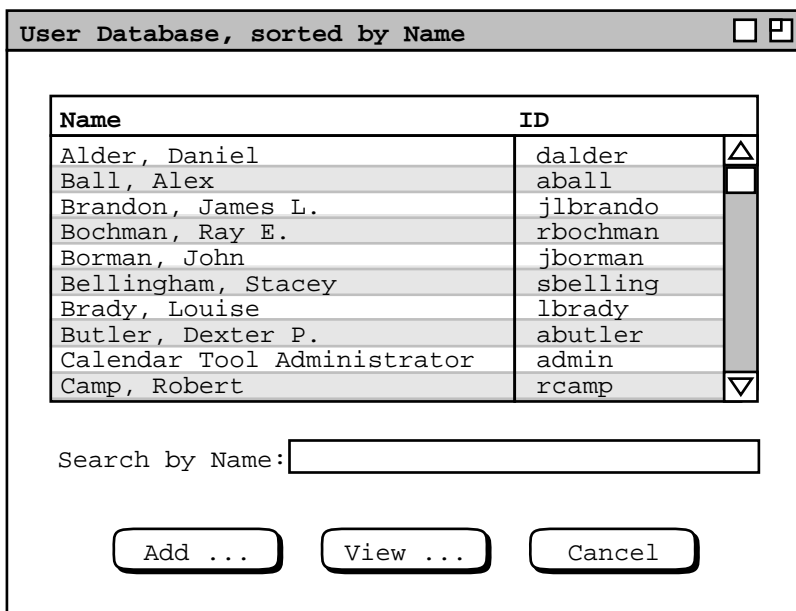


Figure 198: User database dialog.

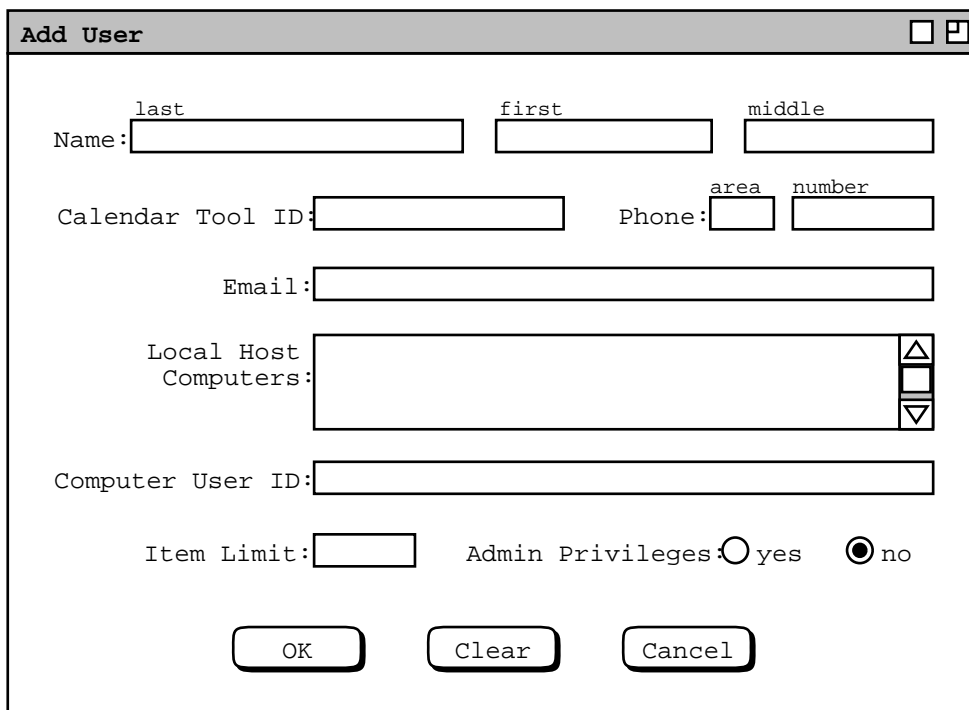


Figure 199: Add user dialog.

network to which computers are attached. IP address or qualified host name are typical formats. 'Computer User ID' is the user's unique identification on the host computer(s). If more than one local host is listed, the user's computer ID must be the same on all of them.

The 'Item Limit' field specifies the maximum number of scheduled items that can be stored for the user in the Calendar Tool central repository. The limit value is optional. If it is absent, there is no limit. The item limit can be zero, in which case the user has an empty calendar in the central repository. Complete details on the central calendar repository are covered in Section 2.14.

The 'Admin Privileges' radio buttons indicate if the user has administrative privileges. The default is 'no'.

Figure 200 shows the administrator having entered values for a new user. As may be typical, values for several of the fields are similar. Here for example, the 'Calendar Tool ID', 'Email' prefix, and 'Computer User ID' are the same. This is not required, but is likely to be a common practice.

When the administrator presses 'OK' in the add-user dialog, the system performs the following actions:

- a. validates the user information that is external to Calendar Tool administrative control, as described in Section 2.11.25
- b. adds the new user record to the user database
- c. updates the list of names in the user database dialog
- d. adds the new user's ID to the All Users group, as described further in Section 2.6.3.
- e. notifies the new user of the addition
- f. If the 'yes' option is selected for 'Admin Privileges', the system
 - i. grants the affected user administrative access privileges
 - ii. adds the user to the 'admins' group
 - iii. sets the value of the user's administrative password to the user's ID

When an administrator changes the 'Admin Privileges' setting from 'yes' to 'no', the system ungrants administrative access and removes the user from the 'admins' group.

The 'Add User' dialog box contains the following fields and controls:

- Name:** last: Anderson, first: Richard, middle: (empty)
- Calendar Tool ID:** randerso
- Phone:** area: (empty), number: 756-1843
- Email:** randerso@calpoly.edu
- Local Host Computers:** falcon.csc.calpoly.edu
- Computer User ID:** randerso
- Item Limit:** (empty)
- Admin Privileges:** yes no

Buttons at the bottom: OK, Clear, Cancel.

Figure 200: Adding a new user.

The new-user notification message is of the form shown in Figure 201. [NO to the following, per the latest revelation the no admin data are ever stored on local user machines see below : In conjunction with the notification message, the Calendar Tool on the new user's host computer is reconfigured with the ID of the Calendar Tool central host.]

Details of central host configuration are covered in Sections

and 2.6.6.

Pressing **Clear** in the add-user dialog erases any edits made since the dialog was displayed. **Cancel** cancels the add operation without adding to the database.

To view the database record of an existing user, the administrator selects the user name or ID in the list and presses the **View ...** button. Clicking on either the name or ID highlights both. For example, Figure 202 shows the user having selected James L. Brandon for view. Note that the figure shows the newly added user Richard Anderson. When the administrator presses **View ...** in Figure 202, the system displays the dialog of Figure 203. A shortcut to view a user's record is to double click on the user name or ID in the scrolling list.

The user record dialog has all of the same data fields as the dialog for adding a new user. In addition, there are two scrollable lists that show the Calendar Tool groups of which the user is a member and a leader. The administrator can change all data fields except **Calendar Tool ID**, **Member of**, and **Leader of**. Once an ID is

You have been added as a new registered user in the Calendar Tool server
running on host computer *host name*.

Your registered name and ID are "Richard Anderson" and "randerso".

To view your complete user record,
connect to this host using the **Admin->Central Host** command,
then select the **Admin->Users** command,
and then select your name or ID in the list of registered users.

Figure 201: New user notification message.

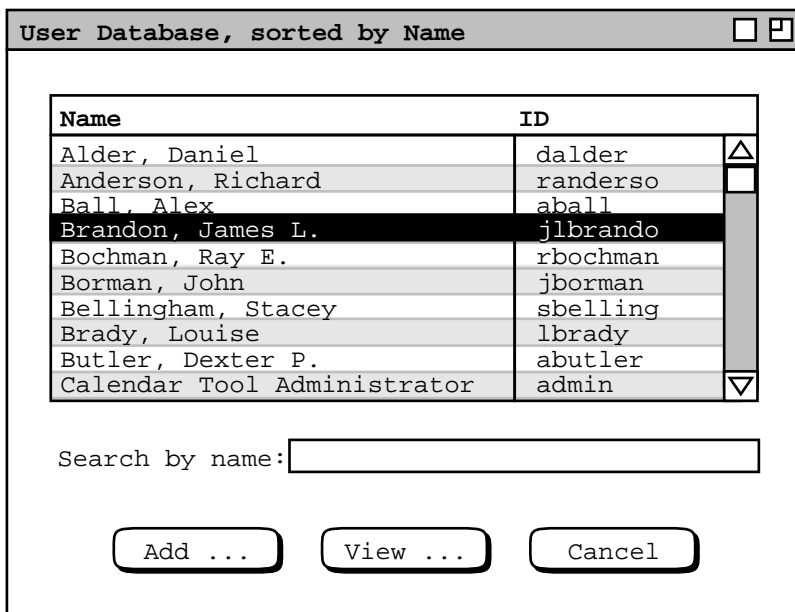


Figure 202: Brandon selected in users list.

The screenshot shows a window titled "User Record" with the following fields and controls:

- Name:** last: Anderson, first: Richard, middle: (empty)
- Calendar Tool ID:** randerso
- Phone:** area: 805, number: 756-1843
- Email:** randerso@calpoly.edu
- Local Host Computers:** falcon.csc.calpoly.edu (with up/down arrows)
- Computer User ID:** randerso
- Item Limit:** (empty text box)
- Admin Privileges:** yes, no
- Member of:** cadrc, ceng, csfac (with up/down arrows)
- Leader of:** cadrc (with up/down arrows)
- Buttons:** Validate, Change, Delete, Clear, Cancel

Figure 203: User database record for James L. Brandon.

assigned for a particular user, it cannot be changed. Group membership and leadership are changed with the group database commands described in Section 2.6.3.

The 'Validate' button is used to validate user information that is external to Calendar Tool administrative control. This is the information entered in the 'Email' through 'Computer User ID' fields. Details of the validation are covered in Section 2.11.25

Figure 204 shows the results of the administrator having performed edits to the user record for James L. Brandon. The edits are to the 'Phone', 'Email', and 'Local Host Computers' fields. When the administrator presses 'Change', the system displays the confirmation dialog in Figure 205. The administrator confirms the change with 'OK' or cancels it with 'Cancel'. Upon confirmation, the system performs the following actions:

- updates the user record in the database
- notifies the affected user of the change
- deletes items from the user's calendar in the central repository, if necessary to satisfy a changed item limit value; Section 2.14.2 has further details on central repository item limits
- terminates Calendar Tool Administration if 'Admin Privileges' was set to 'no' and the user is executing it at the time the change is confirmed, including if it is the administrator who has performed the change

User Record

Name: last first middle

Calendar Tool ID: Phone: area number

Email:

Local Host Computers:

Computer User ID:

Item Limit: Admin Privileges: yes no

Member of:

Leader of:

Figure 204: User record for James L. Brandon edited.

Confirm Change

Are you sure you want to change the user record for James L. Brandon?

Figure 205: Confirmation dialog for changing a user record.

In this example, neither action c nor d applies.

The change notification appears on the affected user’s screen, in the form shown in Figure 206. The notification message lists the specific details of the change. The email address at the end of the message is that of the administrator who performed the change. This email address is defined in the user record for the administrator.

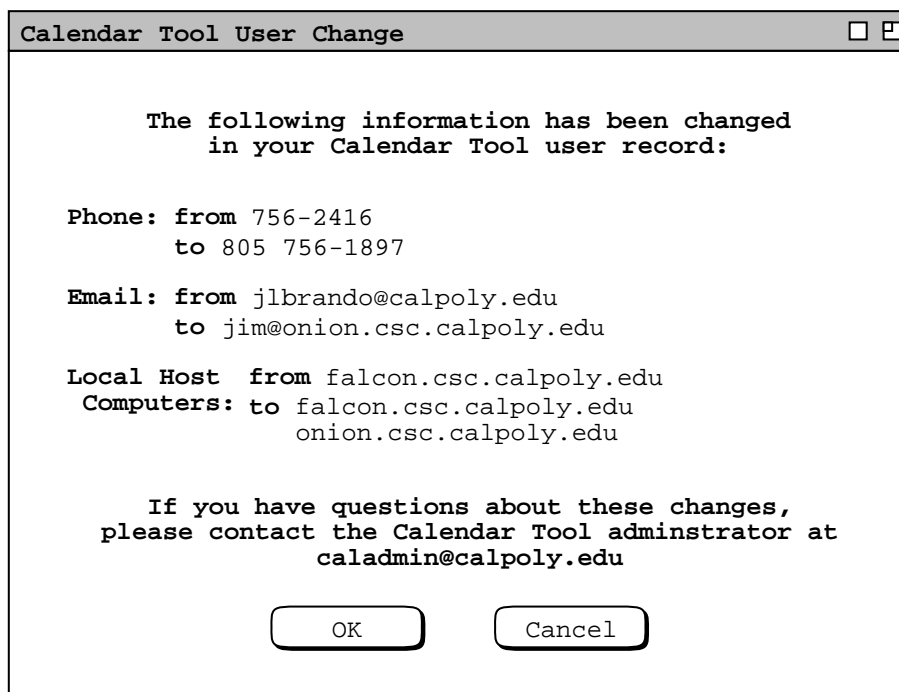


Figure 206: Notification of change to James L. Brandon user record.

To delete a user, the administrator presses 'Delete' in the user record display. In response, the system displays a confirmation dialog of the form shown in Figure 207. When the administrator confirms the deletion with 'OK', the system notifies the deleted user with an on-screen notification of the form shown in Figure 208. This is the notification sent to user Frank D. Smith. The notification text explains the delete action.

Deleting a user has the following effects on system data:

- a. removes the user's record from the user database
- b. removes the user from all groups of which the user is a member and/or leader, including the All Users group, and if appropriate the Administrators group
- c. removes the user's calendar from the central repository
- d. cancels any meeting scheduling operation that the deleted user is executing at the time the deletion is confirmed

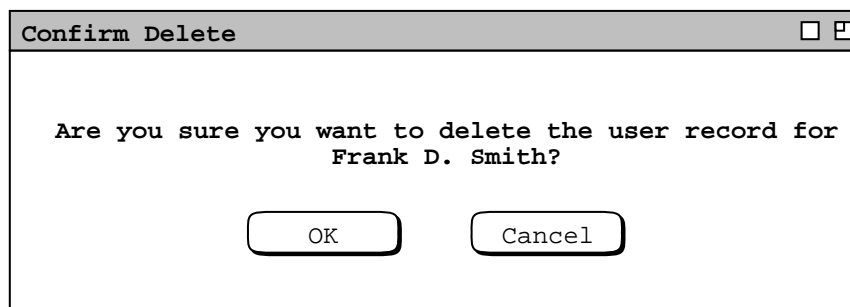


Figure 207: Confirmation dialog for deleting a user record.

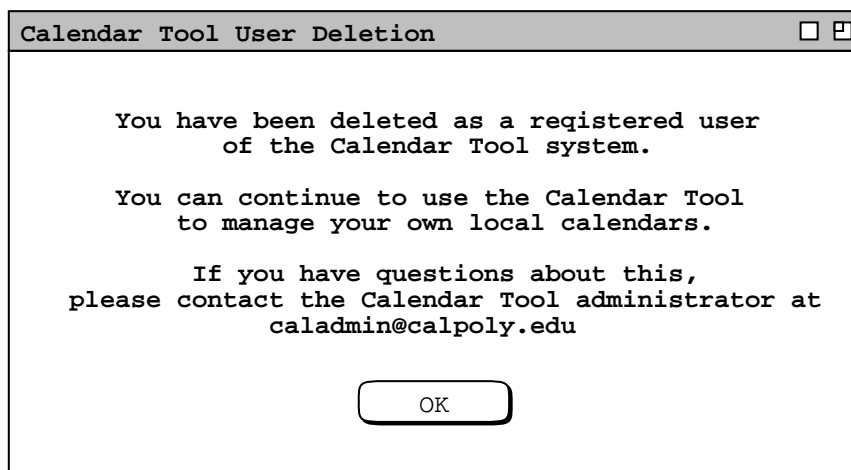


Figure 208: Notification sent to deleted user.

- e. terminates the Calendar Tool Administration if the deleted user is executing it at the time the deletion is confirmed, including if it is the administrator who has performed the deletion

Deleting a user does not delete the user's ID from any meeting attendees lists on which it appears. Once a meeting is scheduled, its attendees list remains unchanged, except for local edits performed by individual (non-scheduler) users.

There is a rare but possible condition where an administrator deletes a user while another user is scheduling a meeting. Under this condition, the user deletion could affect the possible times or time conflict lists that are already displayed at the time of user deletion. In such cases, the system does *not* dynamically update the affected lists. Therefore, the in-progress scheduling operation proceeds without reflecting the user deletion. All meeting scheduling that begins subsequent to the user deletion is performed as it should be in the context of the deleted user.

NOTE: Somewhere in this section (or conceivably 2.14) should be examples of Admin Priveleges set to yes and Item Limit set to something non-empty.

2.6.3. Group Database

User groups are a convenient means to define a named list of related users. In comparison to an individual user, a group can be considered a virtual entity. No single user owns the group; a group has no email address and no host computer information. As explained in Section 2.4.1, meetings scheduled by group leaders are recorded in group calendars, as are meeting changes made by leaders. However, none of the group leaders owns the group calendar explicitly. In particular, if all group leaders are deleted, the group calendar persists.

To access the group database, the administrator selects the 'Groups . . .' command in the privileged 'Admin' menu. In response, the system displays the dialog in Figure 209. This dialog has the same format as that for the user database in Figure 198. Two groups are built-in to the Calendar Tool: all and admins. These groups consist of all registered users and all administrative users, respectively.

To add a new group, the administrator presses 'Add . . .', whereupon the system displays the dialog of Figure 210. The group 'Name' is free-form text for the full name of the group. The 'ID' is the unique group identifier. The 'ID' must be unique among all Calendar Tool users and groups. The 'Item Limit' specifies the maximum number of scheduled items that can be stored for the group in the central repository.

The 'Members' and 'Leaders' lists contain the user IDs of the group members and leaders. Both of these lists can have zero or more entries. A group with no members or leaders is considered dormant. A group with members but no leaders cannot have meetings scheduled for it. The utility of dormant and leaderless groups is limited, but the system allows them to exist at the discretion of system administrators. A group leader need not be a member. This

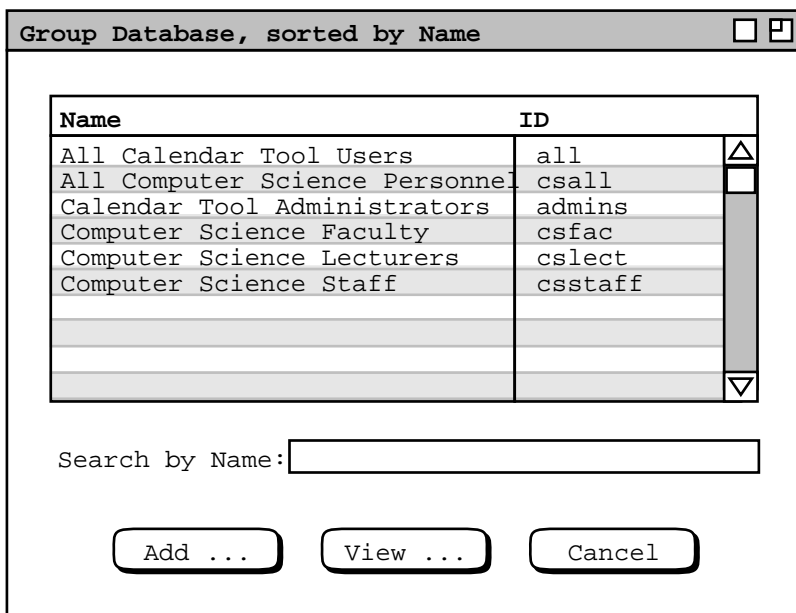


Figure 209: Group database dialog.

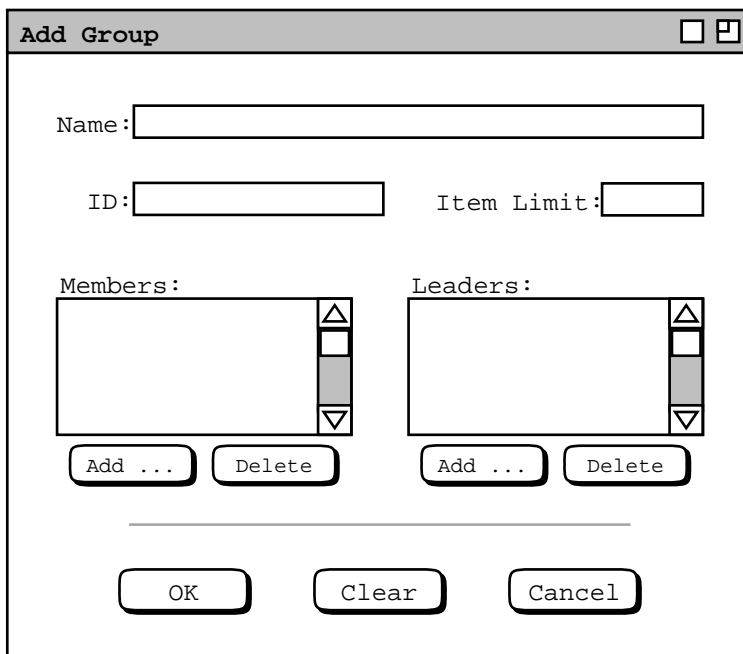


Figure 210: Add group dialog.

allows a leader to schedule a meeting at which the leader is not an attendee.

Figure 211 shows the administrator having entered values for a new group. Immediately below the member and leader lists are small 'Add ...' and 'Delete' buttons. These are used to add and delete users from the lists. For example, when the administrator presses the 'Add ...' button below the 'Members' list, the system displays the dialog of Figure 212. The dialog allows the administrator to enter one or more IDs for group members. The entered IDs must be of current registered users and/or defined groups. To facilitate entry, ID completion is operative in the

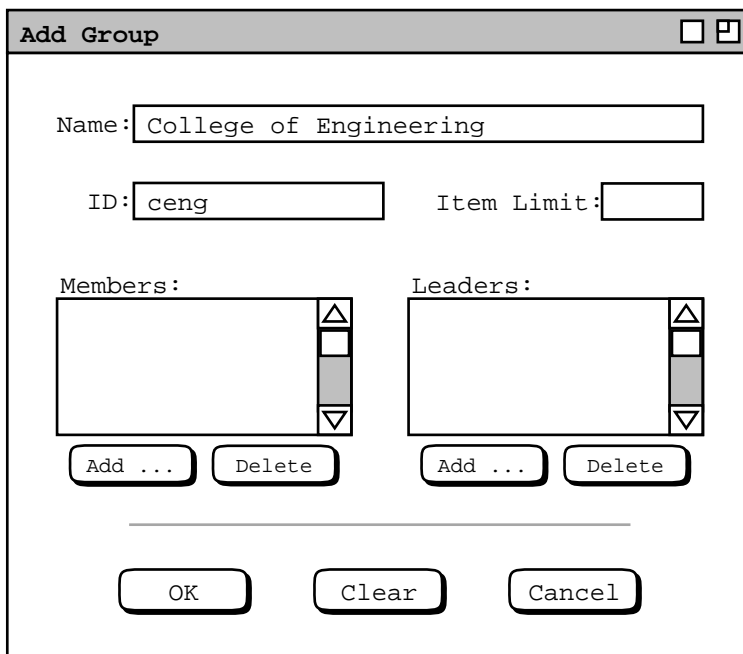


Figure 211: Adding a new group.

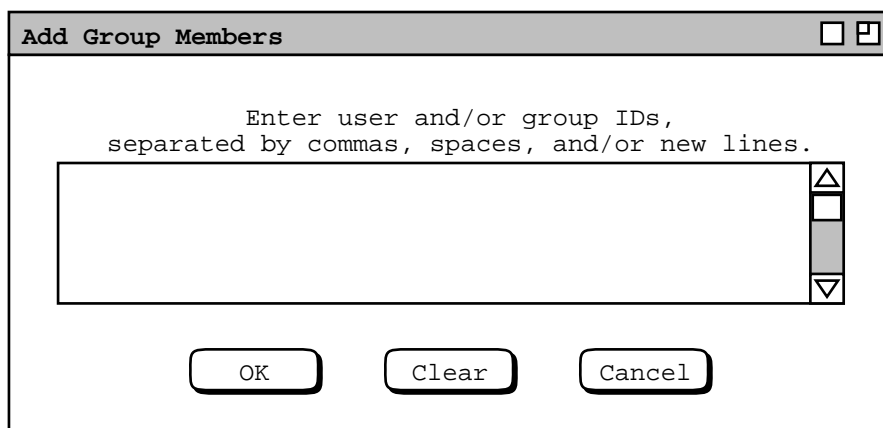


Figure 212: Add group member dialog.

dialog. Specifically, the administrator can type the leading prefix for an ID and press the TAB key to have the ID completed. Details of name and ID completion are covered in Section 2.11.20 The administrator can view the list of all available user and group IDs in the database dialogs of Figures 198 and 209. Allowing multiple IDs to be entered facilitates the definition of groups with a large number of users. If desired, the administrator can use the 'Edit Paste' command to enter names from an outside text file, assuming inter-program copy/paste operations are supported by the underlying operating environment.

Figure 213 shows the administrator having entered IDs for the `ceng` group. The first row of IDs are for individual users. The IDs that follow are for other groups. As introduced in Section 2.4.1.3, a group that contains other groups as members is a *supergroup*. There is no limit on the depth of the supergroup hierarchy. That is, a supergroup can have one or more supergroups as members. In terms of individual users, supergroup membership is defined as the set of all users appearing throughout the supergroup hierarchy. As explained in Section 2.4.1.1, user duplication in a group membership hierarchy is ignored in the context of scheduling meetings.

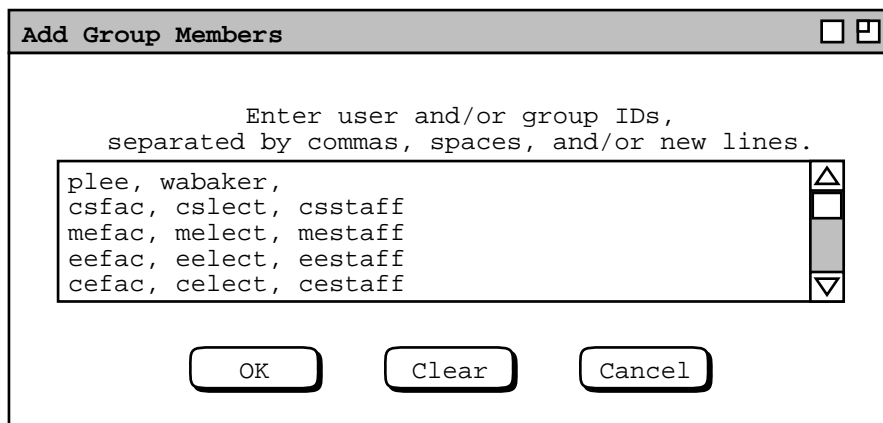


Figure 213: Add group member dialog filled in.

When the user presses ‘OK’ in Figure 212, the system validates all entered IDs. If one or more IDs are invalid, the system issues an appropriate error message, as described in Section 2.12.5. If all IDs are valid, the system enters them in alphabetic order into the members list for the group. If the same ID is entered more than once in the add-member dialog, the name appears only once in the completed members list. Figure 214 shows the results of adding the ‘ceng’ group members. The added members are listed alphabetically, one per line in the ‘Members’ viewing list.

To add one or more group leaders, the administrator follows mostly the same procedure as for adding members. The only difference is that the IDs of group leaders must be individual users, not groups.

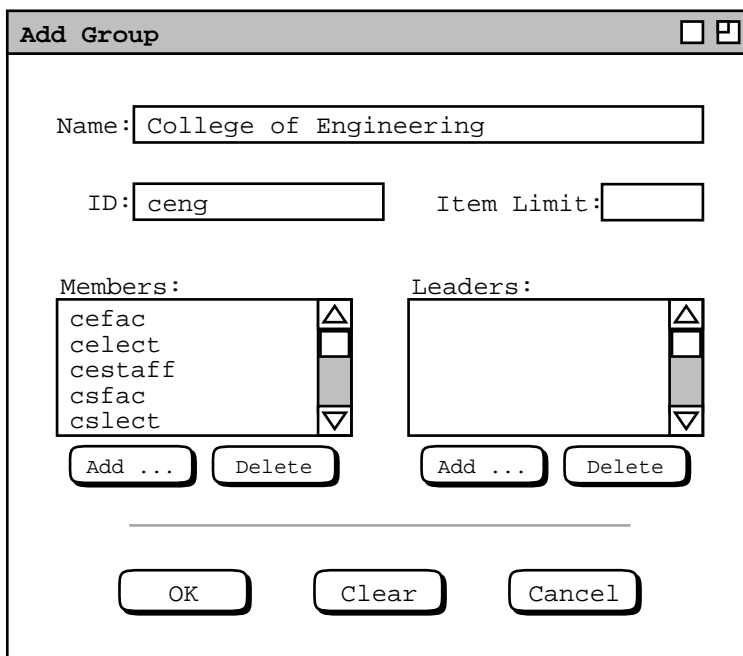


Figure 214: Add group dialog with members added.

At the bottom of the add-group dialog are 'OK', 'Clear', and 'Cancel' buttons. These apply to the group as a whole. When the user presses 'OK', the system adds the group to the database. 'Clear' and 'Cancel' have the normal functionality.

To view an existing group, the administrator selects the group name or ID in the group database dialog of Figure 209. and then presses the 'View ...' button. For example, Figure 215 shows the result of the user viewing the csfac group. The administrator can change all fields except 'ID'. As for a user, once an 'ID' is assigned to a group in cannot be changed. The 'Add' buttons immediately below the members and leaders lists operate in exactly the same manner as in the add-group dialog of Figure 210.

To delete one or more group members, the administrator selects the name(s) in the 'Members' list and presses the 'Delete' button immediately below the list. To select multiple names for deletion, the user holds down the SHIFT key as each name is selected in the list. In response to executing 'Members: Delete', the system displays a confirmation dialog of the form shown in Figure 216. Here the confirmation is in response to the administrator selecting two users for deletion from the csfac group. 'OK' confirms the deletion, 'Cancel' cancels it. In response to

The screenshot shows a window titled "Group Record" with a close button in the top right corner. Inside the window, there are several input fields and lists:

- Name:** A text box containing "Computer Science Faculty".
- ID:** A text box containing "csfac".
- Item Limit:** An empty text box.
- Members:** A list box containing the names "aball", "arichard", "cdaniels", "dalder", and "dbutler". Below the list are "Add ..." and "Delete" buttons.
- Leaders:** A list box containing the names "estier" and "hlandis". Below the list are "Add ..." and "Delete" buttons.

At the bottom of the window, there are four buttons: "Change", "Delete", "Clear", and "Cancel".

Figure 215: Group database record for the Computer Science Faculty.

The screenshot shows a window titled "Confirm Member Deletion" with a close button in the top right corner. The main text inside the window reads:

Are you sure you want to delete the following members from the csfac group?

cdaniels, wbucknel

At the bottom of the window, there are two buttons: "OK" and "Cancel".

Figure 216: Group member delete confirmation.

confirming the delete, the system removes the names from the members list, as shown in Figure 217. Since the members list is longer than fits in the scrolling area, only the deletion of 'cdaniels' is immediately visible.

To delete one or more group leaders, The administrator follows the same procedure as for member deletion. In the leader-deletion confirmation dialog, the word "member" is replaced by "leader".

The four command buttons at the very bottom of group dialog correspond to the group record as a whole. When the user presses 'Change', the system displays a confirmation dialog of the form shown in Figure 218. When the administrator confirms the change with 'OK', the system performs the following actions:

- a. updates the group record in the group database
- b. updates the members and leaders lists in the group records of all affected users
- c. notifies all added or deleted users of the change
- d. notifies all added or deleted leaders of the change

If the same user is added as both a member and leader, only one notification is sent. The notification takes the following form:

Figure 217: Members deleted in group record dialog.

Figure 218: Confirmation dialog for changing a group record.

**You have been *actioned*
the *group* Calendar Tool *group*.**

**If you have questions about this
please contact the Calendar Tool administrator at
*admin***

The *actioned* string is one of the following, as appropriate to the change(s) made:

"added as a member of"

"added as a leader of"

"added as a member and leader of"

"deleted as a member of"

"deleted as a leader of"

"deleted as a member and leader of"

The *group* string is the group ID. The *admin* string is the email address of the administrator performing the operation.

When the user executes 'Delete' for a group record, the system displays the same form of confirmation dialog as shown in Figure 218, with the word "**change**" replaced with "**delete**" in both the dialog banner and message text. When the delete is confirmed, the system

- a. removes the record from the group database
- b. removes the group calendar from the central repository
- c. removes the group from the members and leaders lists of all affected users
- d. notifies all affected users

The notification takes the following form:

**The Calendar Tool *group* group,
of which you were a *affiliation*,
has been deleted.**

The *group* string is the group ID. The *affiliation* is one of the following, as appropriate: "**member**", "**leader**", or "**member and leader**"

Change or deletion of a group record has no effect on any user calendars. In particular, no meetings scheduled with the group as an attendee are changed or deleted. Since deleted group IDs remain in scheduled items, system administrators must exercise good judgment in reusing the ID of a deleted group. If a deleted group ID is reused for a group with different membership, the extant appearances of the ID in user calendars will be misleading in terms of the users for whom the meeting was originally scheduled.

The Calendar Tool provides no commands for a user to request addition to or removal from a group. Such requests must be made to an administrative user by some means outside of the Calendar Tool, such as electronic mail. The user can view a list of all administrators using the 'List Admins' command. The email address for an administrator is available in the administrator's user database record.

2.6.4. Location Database

The location database contains information records for rooms and other locations in which meetings can be scheduled. To access the location database, the administrator selects the 'Locations . . .' command in the privileged 'Admin' menu. In response, the system displays the dialog in Figure 219. This dialog has a very similar format to the user and group database dialogs. Here, locations are listed by name and number. The default sorting order is by name, which can be changed to number by clicking on the 'Number' column heading. Since names need not be

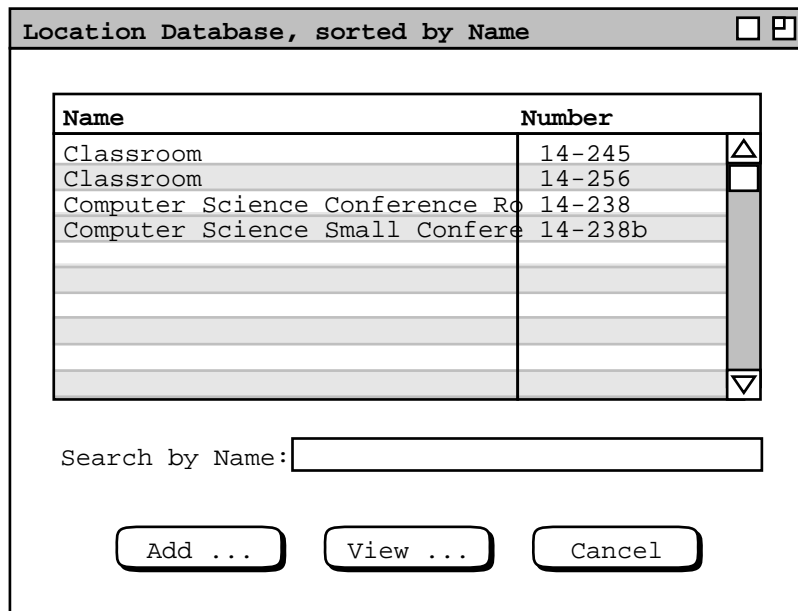


Figure 219: Location database dialog.

unique, secondary order is by number when name is the primary sort key. The 'Search by' field operates in the same manner as in the user and group database dialogs.

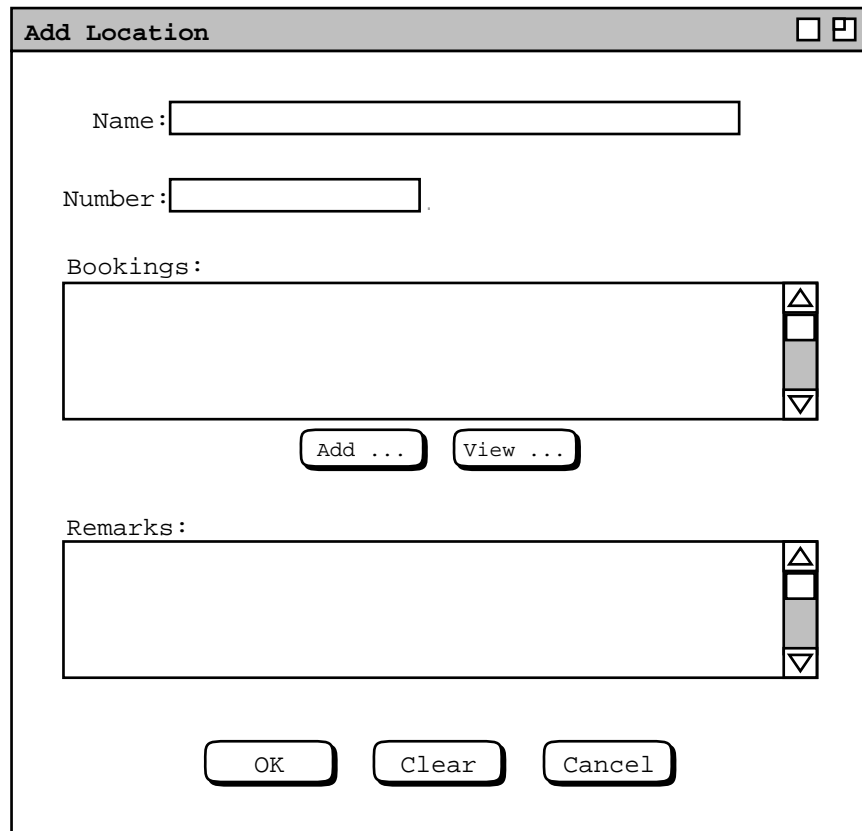
To add a new location, the administrator presses 'Add ...', whereupon the system displays the dialog of Figure 220. The 'Name' field is a free-form text string. 'Number' is the unique location identifier, typically in the form of a room number, but not restricted to any specific format. The scrollable 'Bookings' field contains a list of time slots during which the room is booked for meetings or other activities. The entries in the 'Bookings' list are of two forms:

- a. the title of a scheduled meeting
- b. the description of some other usage

The meeting titles are for meetings scheduled using the Calendar Tool, as described in Section 2.4.1. The other usage descriptions are added by the administrator for times during which the location is known to be in use, meaning that meetings cannot be scheduled at those times. Immediately below the bookings list are command buttons to add and view bookings. The 'Remarks' field contains optional explanatory information about the location, such as equipment or facilities it contains.

Figure 221 shows the administrator having entered values for a new location record. The 'Name', 'Number', and 'Remarks' values are entered by normal typing. To add a booking, the administrator presses the 'Add ...' button below the bookings list, in response to which the system displays the dialog of Figure 222. The 'Use' field describes the use for which the location is booked. The 'Start Time' and 'Duration' specify the time slot of the booking. The 'Start Date' and 'End Date' specify the date range. The dates may be empty, which means that the location is booked indefinitely for the specified use. If the 'Start Date' is specified but the 'End Date' is empty, the start date is taken as a single date. The time and date values must follow the normal data entry rules described in Section 2.11. Pressing 'OK' adds the booking to the 'Bookings' list, if it does not overlap in time or date with any existing booking. The 'Clear' and 'Cancel' commands have the standard functionality.

Figure 223 is an example of a location booking. In this example, the booking is the first to be added, so no overlap occurs. If the administrator attempts to add a booking that does overlap with an existing booking, either meeting or other usage, the system issues an appropriate error message as described in Section 2.12.5. A booking overlap arises when the time and date range of any existing booking overlaps with any part of the time and date range for a new booking to be added.



The image shows a dialog box titled "Add Location". It contains the following elements:

- Name:** A text input field.
- Number:** A text input field.
- Bookings:** A list box with a vertical scrollbar on the right side, containing three items. Below the list box are two buttons: "Add ..." and "View ...".
- Remarks:** A text area with a vertical scrollbar on the right side.
- Buttons:** At the bottom of the dialog are three buttons: "OK", "Clear", and "Cancel".

Figure 220: Add location dialog.

Location Record

Name:

Number:

Bookings:

Remarks:

Figure 221: Location database record.

Book a Location

Use:

Start Time: Duration: hr min

Days: S M T W Th F S

Start Date: End Date:

Figure 222: Book location dialog.

Book a Location

Use:

Start Time: Duration: hr min

Days: S M T W Th F S

Start Date: End Date:

Figure 223: A location booking for the Grad Student Room.

To view an existing location record in the database, the administrator selects a location name or number in Figure 219 and presses 'View ...'. For example, Figure 224 shows the result of the administrator viewing the second listed location in Figure 219. The location record dialog has exactly the same data fields as the add-location dialog

Location Record for Classroom

Name:

Number:

Bookings:

- Afternoon Classes
- Computer Science Faculty Meeting
- Morning Classes

Remarks:

Used for classes except 1-2 PM.
Seats 35.
Has computer hookup and projection system.

Figure 224: Viewing a location record.

shown in Figures 220 and 221. In place of the 'OK' button there are 'Change' and 'Delete' buttons. The sorting order for the 'Bookings' list is alphabetic, case-insensitive. If a meeting title and other-usage booking are the same string, the meeting title appears first in the list.

The 'Add ...' booking command is described above. To view an existing booking, the administrator selects the booking description in the list and presses the 'View ...' button immediately below the list. When the selected booking is for a meeting, the system displays an item-level view for the version of the meeting on the scheduler's calendar. If it is a recurring meeting, the displayed instance is for the start date. When the viewed booking is for a non-meeting usage, the system displays a booking dialog. For example, Figure 225 shows the result of 'View ...' for the second booking listed in Figure 224; Figure 225 shows the result of 'View ...' for the third listed booking. Since an outside-usage booking is for a contiguous time period, it takes multiple bookings to define

Scheduled Meeting, user Ellen M. Stier
□ □

◀ Today ▶ **1 PM Monday, September 21, 1998**

Title:

Start Date: Start Time:

hr min

End Date: Duration:

Recurring? Interval:

Category: Security:

Location: Priority:

Remind?

Attendees:

Details:

Minutes:

Scheduled By: On:

Host:

Figure 225: Viewing a meeting from the location record dialog.

Figure 226: Viewing an other-usage booking from the location record dialog.

discontinuous usages. For example, the 'Morning Classes' booking in Figure 224 covers the 8 AM to 1 PM time slot. The 'Afternoon Classes' booking covers 2 PM to 6 PM. These two definitions allow meetings to be scheduled between 1 and 2 PM.

The administrator can change or delete a booking using the command buttons at the bottom of the booking dialog. To change, the administrator edits one or more data fields and presses 'Change'. To delete, the administrator presses 'Delete'. 'Clear' removes any edits made after initial dialog display. 'Cancel' removes the dialog with no change or deletion made.

An other-usage booking is not a scheduled item, so it does not appear on any calendar. The effect that a booking has on scheduling is to prohibit a meeting from being scheduled in the location during any of the booked times and dates. An other-usage booking expires after its end date or single start date, if present. The system does not automatically delete expired bookings, but an expired booking no longer affects meeting scheduling. Deletion of expired bookings is the responsibility of system administrators.

2.6.5. Other Admin Commands

In addition to maintaining Calendar Tool databases, the administrator can perform commands to configure the central host computer, list administrative users, change administrator passwords, and notify calendar tool users. These commands are described in the subsections that follow.

2.6.5.1. Administrative Control of the Central Host

21jul02 Update: Add/merge the following with the following following:

- The user can choose the host in the central-host dialog shown in Figure 227.

To connect or disconnect to a host, the user selects the desired row in the list and presses 'Connect' or 'Disconnect'. When the designated file for a particular host is opened, the system attempts to establish a connection with the central host for that file; when the file is closed, the system disconnects. (See multi-user-envir for more info.) A host must have a designated calendar, i.e., the 'Calendar' column cannot be empty and must contain the name of a valid readable/writable calendar file.

OK, OK, one more go round about allowing multiple hosts on the same calendar. These need to be the rules:

- a. When the user saves, the cal goes to *all* designated central hosts.
- b. When a notification arrives from *any* designated host, it goes to the same calendar.

Central Host	Calendar	Connected
19.62.146.83	Personal Calendar	
csc.calpoly.edu	Work Calendar	X
falcon.csc.calpoly.edu	Work Calendar	
onion.csc.calpoly.edu	Lab Calendar	X

Connect Disconnect

Add Change Delete Clear Cancel

OLD:

Central Host Computer ID: host name, with admin-settable default

Status: connected/disconnected

Connect Disconnect Clear Cancel

Figure 227: Regular user central host dialog.

And here's a summary of why multiple calendars on one host is a bad idea:

- a. When the user saves one cal, then another, previous save gets clobbered on the central host. This is seemingly not a good idea, since the cal on a particular host is the user's advertised public calendar. Having it bounce back and forth between different calendars doesn't seem like that great an idea.
- b. When a notification arrives, the system cannot determine without asking to which calendar the notification applies, unless the rule is that it goes to all calendars. This would mean that all calendars are opened upon notification.
- c. I cannot think of any case like that described under "Rebuttal" just below for the multi-cal-to-host case. And here's the crux, I believe, of why not -- the way updates are made from host-to-cal versus cal-to-host is asymmetric. Viz., in the case of host-to-cal, a notification arrives and when the user accepts, a meeting is *incrementally added* to the current (host-associated) calendar. In the case of cal-to-host, updating is *not (necessarily) incremental*, but rather happens for the full calendar whenever save is executed. Now, I don't really think that we can quibble about this asymmetry, unless we go to some form of transparent file-less data model on the local host. This is because in order for updates to the central host to be incremental in the way they are on the local host, we'd have to have some model of both a local and central workspace calendar, which I think is way overkill. Without such a model, either file-less or added central-host workspace, the effects of an add, change, or delete operation cannot be incrementally transmitted to the central host.

OK, thinking about screwing the next paragraph (but see previous paragraph, which nixes the screwing, which means we're going with allowing many-to-one host to calendar associations) and requiring the host-to-cal mapping be unique. Need to think just a bit more if this is a potentially annoying constraint, but I don't think it is. Here's basically the deal on why I don't think it's annoying. The reasons for allowing more than one host per calendar are: (1) In case a host goes down, it's easy to switch to a backup host, without having to enter its name and calendar in the list. (2) The user may want to have the same calendar actively associated with more than one host, e.g., a work host and a home host. The argument against both of these goes like this: (1) Hosts going down is considered a reasonably rare event, therefore having a "hot backup" config in the host list is not a big deal; i.e., typing in the name of the new host is not going to happen very often, and therefore it's not a significant convenience for the user to be able to do this in advance. (2) Even if it's not all that confusing to allow a user to dynamically switch back and forth between hosts with the same calendar, it's not clear that it's something the system should encourage all that much, particularly by making it easy in the hosts lists. If the user wants to do this, she can do it easily enough simply by changing the host for a particular calendar. Therefore, let's call it a done deal and say that there's a one-for-one

correspondence between a calendar and a host, with neither field allowed to be empty. Further the way the host list is populated is based simply on the currently open calendars, with each calendar having a host component stored in it. (Rebuttal: Ah, but the problem with this is clarified by the following scenario. I'm at Poly and there's a campus-wide host and departmental host. I don't really want to have a separate calendar for the campus-wide host, preferring instead to have campus meeting notifications go to my main campus calendar. And further, I'm fine with having my dept calendar and campus calendar be the same. Therefore, I'd like to have both the campus and dept hosts connected to the same calendar. In this case it would definitely be a pain to have to manually switch between campus and dept calendars. Any notifications that come from the campus host would have to be manually copied to the dept calendar -- I think this is the biggest pain. One solution to this might be to allow dynamic calendar selection from the meeting notification dialog, but I think this would be a potentially serious complication to things. I.e., we want the user to just press 'Accept' in the normal simple cases, without having to worry about to which calendar the meeting is going, or even have any kind of GUI fodder about this in his face, and/or to have to deal with some kind of emacs-esque prompt-for-calendar-on-meeting-accept option. Or how about a list-valued option var `auto-accept-meeting-notifications-from-hosts-when-mapped`, which lists the hosts from which meeting notifications are automatically accepted as long as the host is associated with a unique calendar. The last digressing sentence is further justification for the one-cal-per-host(s) requirement, actually.

Even with the unique host-to-cal association, it is possible to have a calendar open without being connected to a host, however the converse is not true. I.e., the user cannot be connected to a host without having the calendar for that host open. The open-but-unconnected case allows the user to work "offline" with a calendar, e.g., when the user is on a laptop that is not currently connected to a network, or for whatever other reason the user may choose to do so. Allowing connected-but-not-open case would be problematic in the way the user designated (and therefore knew) which host was current. I.e., there'd have to be a way to make a host current independent of making a calendar current. This probably isn't all that bad, but it seems like a pain and potentially confusing. The reason the system always needs to know which if any host is current is for the purposes of the admin DB commands, which are clearly host-specific.

So the bottom line is this. A particular calendar is the user's "portal" to a particular host. Whenever the user saves the calendar, the central calendar on its associated host is updated, not the calendar on some other host. Whenever the user accepts a meeting notification from a particular host, the meeting is added to the calendar associated with the host. One slight inconvenience here is that the user cannot view the administrative information or another user's calendar on a particular host without having the user's own calendar for that host open. (*Think this through wrt the no-windowless-open-calendars rule.*)

Probably NOT: As can be seen in the sample table, the same calendar may be the designated file for more than one host. When this happens, that single calendar is one to which accepted meetings are added from all hosts for which the calendar is designated. The converse is not true, i.e., a single host may not have more than one designated calendar. This is because when the user receives and accepts a meeting notification from a particular host, the system needs to know which calendar the accepted meeting goes to, without having to ask the user. In terms of the UI, the 'Central Host' column must be unique where as the 'Calendar' column need not be.

The user must be aware of the consequences of changing the designated file for a particular host, as well as designating the same file for two or more hosts. See the details in the multi-user-envir section.

To add a new host, press 'Add' to which the system responds by bringing up an add dialog, which if confirmed adds the new host in its lexically sorted position in the list. To change, select item and press 'Change' to which the system responds by bringing up a change dialog, which if confirmed makes the change, including resorting if necessary. [This version of change doesn't work cause it'd be hard to tell if select meant place edit bar or select for delete: To change, edit a row and press 'Change' (the all host and cal columns of all extant rows are always editable).] To delete, select a row and press 'Delete', in response to which system brings up delete confirmation dialog. Both add and change enforce unique-host constraint.

OLD: When the user presses 'Connect', the system verifies that the selected host computer is operational, has an active Calendar Tool server running, and that the user is registered with the Calendar Tool on that host. If the verification is successful, the system establishes communication with the chosen host and sends the user's community calendar file there.

If communication cannot be established with the selected host, the first line is changed to

Communication cannot be established with the selected host computer.

If communication can be established, but the host is not running a Calendar Tool server, the message is

The selected host computer is not currently running a Calendar Tool server.

If host is running a Calendar Tool server but the user is not registered there, the message is

The selected host computer is running a Calendar Tool server, but you are not a registered user on it.

If the user

I have the feeling that much if not nearly all of the sketch that follows belongs in 2.14 instead of here. This needs to be ironed out (obviously).

OK, try this overall model, including what it looks like from individual user side:

o The user can choose a server explicitly; the server ID is stored in a **KNOWN** location, a la `.netscape/bookmarks.html`.

o As we've figured all along, there exactly one standard calendar file known to the central host, in a **KNOWN** location, viz., in a standard file name in the user's home directory as specified in the user's record in the currently-selected host.

o When a regular user starts up the cal tool or confirms a change to the server ID, the cal tool performs server validation as follows:

- if the user-specified server is empty, then validation is trivially satisfied;
- if the server is unknown or down, then an appropriate error is displayed;
- if the server computer is up but not running as a cal tool central host, an appropriate error is displayed;
- if the server is up as a cal tool central host, but the user is not registered there, an appropriate error is displayed;
- if the server is up as a cal tool central host and the user is registered there, the validation is satisfied.

o The rule for what's stored in the central host, and therefore what's known about one user to other users, is defined as follows:

- What's stored on the central host is the version of the known cal file the last time it was saved *when the host was up*.
- If the user does a save of the standard cal file when the current host is down, then this information is recorded *as part of the standard cal file itself*, so that if the user saves when the host is down, leaves, and comes back when host is up again, the cal tool will notify the user that the host was down the last time the user tried to save the standard cal file, so the standard cal file is being (has been) (re)written to the host.
- Or, if the user does one or more saves while the host is down, and then the host comes back up, the same notification about now saving to the host appears.

o If the user confirms a change of central host, and the change validates, then the user is notified (not asked) that the (non-private parts of the) current standard cal file are being written to the newly-selected host.

o As shown in the current sketch of the admin version of the 'Central Host' dialog, a host may have a *designated backup* host, which is maintained in "hot backup" state whenever it's up.

o BUT, in contrast to the current narrative following the sketch, let's screw the autostart of hot backup idea; when a host goes down, that's it until it comes back up or until the admin explicitly brings up another host and explicitly notifies users to change if they want to join that host.

- As noted above, if the user saves to the standard cal file while the current host is down, when it comes back up, the user is notified that the unsaved-on-the-central-host changes are being written to the central host; this only happens if the user has not changed the central host himself.
- If a host is going to stay down for a while, the admin can send an explicit notification that the host is going to change until further notice.

- If there's been hot backup, then switching is easy.
- Otherwise, the admin needs to use whatever backup source is available for the cal admin files that were most recently on the now currently-down host.

o Here's how things typically go "out of the box":

- An admin chooses a central host and backup, fires up cal admin tool on the host, and starts the server there. In terms of low-level server invocation details, this means that the admin tool *essentially is* the server, in that when the admin presses the 'Start Server' button, the server is launched on whatever host machine the cal admin tool is running. That is, the cal admin tool is not equipped to launch a server on some machine other than the machine on which cal admin is running. As far as the backup server goes,
- The admin registers a bunch of users, under the general assumption that the registration is happening before the users fire up the cal tool themselves.
- Through communication outside of the cal tool, the admin lets users know what host to select, and/or installs that file for them (as part of an installation script); the idea is that as part of the normal installation process, the site-specific default central host is automatically set for the user in the known file, so that the first time the user fires up the cal tool, the central host is set to the default.
- Now brand new users fire up the cal tool, each seeing immediately the notification that they've been added as registered users.
- The users proceed about their business, in many if not most cases never changing the default host setting, unless notified to do so by an admin.
- Also, we can probably assume that most users won't change where they save their calendar, i.e., keeping it saved continuously in the standard cal file in its fully known location.
- If users do save their standard cal file, it's most likely of the "save a copy" variety of saving, where they're just saving a personal backup copy; when they do do this, they need to be careful to do save as again back to the standard cal file, so it's kept up to date in the central calendar repository.
- If users happen to start up the cal tool before the admin is done adding the user records, that should be just fine, since they'll see whatever appropriate error message appears regarding the system not being able to validate them on a particular host, most likely the default host.
- As an alternative to having a built-in (i.e., pre-configured) central host name, the admin could simply instruct everyone (offline from cal tool) what host to type in, once all the users have been added to that host. This means that the default initial host would be empty.

o By allowing the user to dynamically select a host, the user can play in different host spaces; such playing may be tricky and is "caveat user".

- The governor to such playing can be readily controlled by an admin by not having a user registered in more than one active host at a time.
- As noted above, even if a user chooses a particular host, that user won't be able to "play" there unless she's registered there.
- So, "playing" is to a large extent controlled by the admin; he needs to be careful not to have the following situation arise:
 - Host A goes down and the admin advises all users (who are listening) to go to Host B, which has been kept in hot back up from A.
 - Most users hang out fine on B.
 - Host A eventually comes back up, with auto start of the cal tool server, but all listening users are still being served by host B.
 - Some user who wasn't listening earlier fires up the cal tool with A still selected as host, and ends up being in an out-of-date host A environment.
 - BUT (!), I think we can safely say that if the non-listening user stays with A, it's his own stupid fault, since when he fires up cal tool in the preceding step, he'll see the original (pending) notification that advised all users to go to B. If there's been a subsequent notification to shift back to A, then the non-listening user will

see that one too, and everything will be fine, as long as the user *does* listen now.

Describe that the Central Host command allows the admin user to set unique identification for the Calendar Tool central host computer. Describe functions of central host, viz,

- a. facilitates communication among computers of Calendar Tool users
- b. stores calendar information and notifications when necessary (i.e., when user computers are not operating or accessible in the network)
- c. is the computer on which the Calendar Tool database files are located

Figure 228 shows the 'Central Host' dialog.

The 'Host Server ID' field is the unique identifier of the Calendar Tool central host computer on which the Calendar-Tool administration program is running. The format of the identifier is platform-dependent, based on the type of network to which host computers are attached. IP address is a typical format. The reason this field is read-only is that it must be synchronized with the physical host computer which is actually running the server. *Cite later section 2.14, or wherever it is.*

The 'Backup Server ID' specifies an optional backup host computer. If specified, the backup computer is maintained in a "hot backup" state, such that all changes made to Calendar Tool administrative files on the active host are immediately copied to mirror files on the backup host.

The last six data-entry fields specify the absolute file locations of Calendar Tool administrative data files. These file locations are typically set only one time when the Calendar Tool Administration program is initially configured (*explain this more clearly, by being clear about when a screen shot is the initial version of the dialog, versus a subsequent version (i.e., show two separate screen shots for these two dialog states)*). Whenever any of these file names is changed, the contents of the current file are copied to the newly-named file. The User, Group, and Location database files contain the database information described in Sections 2.6.2, 2.6.3, and 2.6.4 above. The general database contains global options settings and other administrative information. The central calendar file contains copies of non-private scheduled items for all registered users, as described further in Section 2.14.2. A separate 'User Calendar' file exists for each user, also as described in Section 2.14.2. **NOTE: Perhaps we should make setting the name of user cal file appear in some other dialog.**

When the location of the central host computer changes and the server is started there, all registered users are notified of the change. The notification dialog is of the form shown in Figure 229 The change notification appears immediately for all users currently running the Calendar Tool, or when a user next invokes the Calendar Tool if it is not currently running.

```

Host Computer ID: (read-only name of current host)
Backup Server ID: changeable name of hot backup -- NUKE THIS
  User Calendars: directory name
    User Database: file name, with reasonable default name
    Group Database: ditto
Location Database: ditto
General Database: ditto ?what the farg is this?

```

```
Status: running, n users connected/not running
```

```

Start      Stop      Install
Server    Server    Data Files

```

Figure 228: Dialog for entering central host computer information.

The identity of the Calendar Tool central host computer has changed
from falcon.csc.calpoly.edu
to bluebird.csc.calpoly.edu

This notification is for information purposes only;
the change has been processed fully automatically by the Calendar Tool.

OK

Figure 229: Notification of change of central host computer.

In conjunction with the change notification, the Calendar Tools of all registered users are reconfigured to reflect the central host change. The effect of this reconfiguration is that any command that requires access to the Calendar Tool central host will subsequently use the newly changed host. The commands that require such access are:

- o File Save, Save As, or Save All
- o the 'List Times' command during meeting scheduling
- o View Other User or Group
- o any Admin command

If possible in a given operating environment, the backup central host immediately becomes the active central host in the event that the original central host stops operating.

Also need to scenarioize if/how the user is notified for other Figure 228 info changes.

2.6.5.2. Listing Administrators

Describe that 'List Admins' shows who all the admins are, Figure 230.

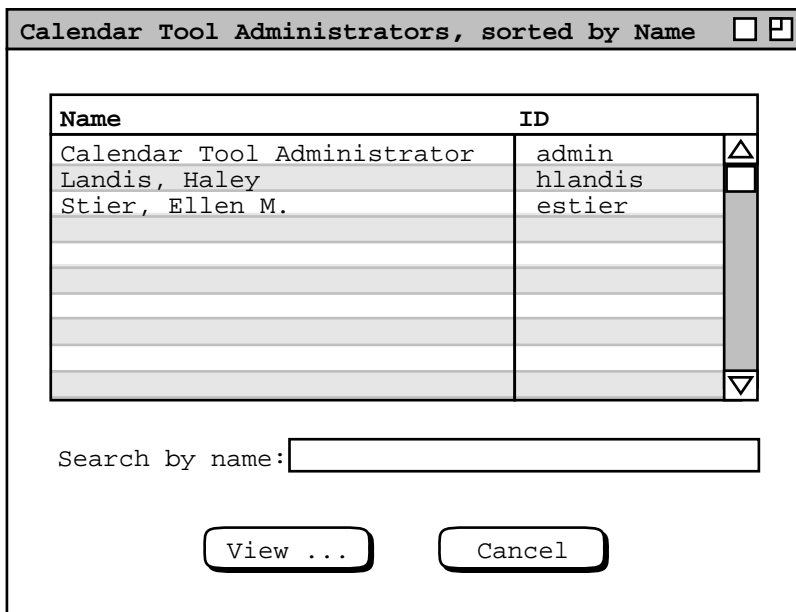


Figure 230: List of Calendar Tool administrators.

Since there's a lot of server-side pushing going on, administrators need to take care not to confuse users. E.g., starting two different servers with overlapping but not identical user dbs, could leave some users stranded or isolated in weird-ass user spaces. *Think this through!*

2.6.5.3. Changing an Administrator Passwords

Describe that 'Password' lets the admin change her password or the password of any other system administrator. Clearly, administrators should use good judgment when changing passwords, particularly for other administrators.

2.6.5.4. Notifying Users

If the administrator uses the 'Admin Notify Users' command to send a message to all registered Calendar Tool users. When the administrator selects this command, the system displays the dialog shown in Figure 231. For example, Figure 232 shows the administrator sending a notification that the main Calendar Tool host computer is changing. The notification is to be delivered both on the screen and via email.

2.6.5.5. Purging Databases

-- THIS NEEDS TO BE ADDED TO THE COMMAND MENU. --

For large installations and/or for central hosts with limited storage resources, Calendar Tool administrators can purge scheduled items from the central host calendars of some or all users. Sketch of functionality:

- o select user by ID or with radio button 'all'
- o select before and after dates

It's probably not worth it or prudent or even particularly smart to give any selective purging control based on other item content beside the date, e.g., security or category. Since we talking about data owned by the admin, not any local data, the admin has full discussion. Clearly, nuking everything, or lots of really current stuff is most likely a bad idea.

Hmm, if we follow the latest functionality for local-to-central cal copying, it may not do any permanent long-term good to purge without lowering limits as well. So, it looks like we need some kind of coordination with purging and item-limit setting. Shit, one more fucking thing to work out.

2.6.5.6. Creating Calendar Tool Distributions

The administrator can create distribution copies of the Calendar Tool using the 'Admin Distribution ...' command. When the administrator selects this command, the system displays the dialog shown in Figure 233

Message Text:

Deliver How: on screen email

OK Clear Cancel

Figure 231: Notify users dialog.

Figure 232: A notification about change of central host computer.

```

User Distribution: file name
                Create

Adminstrator Distribution: file name
                Create

OK      Clear      Cancel

```

Figure 233: Distribution dialog.

2.6.6. Regular User Access to Admin Commands

Explain that 'List Admins' works exactly the same for regular users as for admins.

2.6.6.1. Database Access

User must be connected to a central host, as described in next section. If not, db commands are disabled. When db commands are enabled, they are the same as for admins, except regular users cannot add, change, or delete any data.

2.6.6.2. Central Host Connections

The 'Central Host' dialog looks like Figure 234. -- NO, it's been updated here. If the status of the central host is down, then all of the Admin menu items except 'Central Host' are disabled. If there is no central host at all, then all of the Admin menu items are disabled.

The central host ID is read-only to a regular user. The ID is set automatically when an administrator updates the user database or changes the central host. Specifically:

- a. when a user record is added or changed, the central host for that user is set to that listed in the user's record
- b. when a user record is deleted, the central host for that user is set to empty
- c. when an administrator changes the central host ID, the central host for all users is set to the newly changed ID

If the Calendar Tool is running at the time a host-changing command is confirmed by the administrator, the change takes effect immediately. If the Calendar Tool is not currently running on a user's host computer, then the central-host change takes effect when the user next invokes the Calendar Tool.

A regular user cannot change any of the values in a user database information record. If a user needs one or more information fields to be changed, the user must request the changes be made by a system administrator. There is no specific Calendar Tool command to make such change requests. An electronic mail request is presumably typical. A user can look up the electronic mail address of any administrator using the 'List Admins' and 'Users' commands in the 'Admin' menu. Figure 235 shows a typical mail message sent by a user requesting an administrator to make some changes to the user's Calendar Tool user information.

OLD: As noted in the introduction, there is a single calendar file that is known to the Calendar Tool system for each user. This is the file named 'Calendar' stored in the user's Calendar Tool directory listed in the user's database record. Only items scheduled in this calendar for are visible to other users of the Calendar Tool system.

```

Host Computer ID: (read only)      Status: (up or down)

```

Figure 234: Central host dialog for a regular user.

To: caltool_admin@csc.calpoly.edu
 From: rich@ailab.calpoly.edu
 Subject: Changes to calendar tool user information

Dear calendar tool admin:

I have changed my office computer and email address. Please make the following changes to my calendar tool user information:

```
email: rich@ailab.calpoly.edu
host computer: acorn.ailab.calpoly.edu
user id: rich
caltool direcotry: c:\home\calendar
```

Thanks.

Figure 235: User requesting calendar tool information changes.

NEW: The 'Local Directory' command allows the user to set the directory on the local computer where the Calendar Tool looks for standard files. Cite section the describes the files and say that a normal file choose comes up with the message "Enter the name of the directory where the Calendar Tool looks for standard files:". Also, explain that this file is set at initial installation.

2.6.6.3. Local Directory

2.6.7. Global Options

Explain how the first three kind of options are ?all? settable by user and are therefore explained in the next section. The difference between what the admin does versus what a regular user does is that under 'Admin->Global Options', the admin is setting the global defaults for all users, some or all of which may be changed by users on an individual basis. I think it's also worth mentioning that an admin can be a regular user if she wants to be. I.e., there need not be a special user account that is "the" admin; there can be if that's how a group of users wants to set things up, but there need not be. If an admin wants to be are regular user, she needs to be aware that functions performed under the 'Admin' menu have potentially global effects, and must exercise according care in the execution of the functions.

The Admin options are viewable by regular users but settable only by system admins, so they're described here in this section of the requirements.

2.6.7.1. Administrative Options

2.6.7.1.1. Root System Administrator

2.6.7.1.2. Group Leader Privileges

A bit of rationale: Since we're nixing pencil in, there was a bit of thought just now that group leaders may not be all that necessary. However, there are other useful reasons to have them. Also, I think it's reasonable to say that group leaders are the only ones who can schedule meetings for which online notification will be sent by the tool. So in effect what we've done is demote pencil power in to notify power.

With that bit of rationale, here are the group leader privileges as we currently see them:

- a. Meeting notifications sent to attendees.
- b. 3jul02 NOTE: The following option is obsolete, given the details described in the latest changing-and-deleting section, in particular the fact that a leader cancels a meeting without deleting it by using the change operation instead of delete.

Cancellation notice sent when group leader deletes a meeting; deleted for non-recurring, marked as CANCELLED for recurring (maybe, or probably better just to nuke it or maybe make this an option).

- c. Can add and remove members for group.

2.6.7.1.3. Capping Calendar Sizes

For any or all users and groups, can cap the number of items for that user that are stored in the central calendar. When the cap is reached, items are removed such that an equal number of items, +/-1, appear before and after today's date. Notification is sent to the user when the cap is reached, including an annoying message every time it changes, which means the user either has to do something to clear out old junk or ask the admin to increase the cap.

Changing the cap size to a value smaller than the current number of items effectively executes a purge operation.

2.6.7.1.4. Platform-Dependent Options

As of 23jul02, I'm saying a firm (as possible with me) NO to this.

Although we'd like to avoid it, it may be necessary to at least plan for platform-dependent options that may be needed for networking connections. If at all possible, I'd like to avoid these. They sound like a cop out to me.

2.6.7.2. Summary of Super User Privileges

- a. *Maybe on this one.* Can cancel any leader-scheduled meeting. We need to clarify that the super user can see only the meeting in the group calendar, not in any individual user's calendar.
- b. Can delete items from group calendars.
- c. Can delete any user. The user's calendar wont be affected, but the user will no longer participate the way a registered user does.
- d. *Cannot* see any more of any user calendar than any other user can; cannot change or delete anything from any user's calendar.
- e. Can cap the number of items that appear in the central Cal Tool repository for any particular user or group.

2.6.8. Establishing Administrative Privileges

2.6.9. Administrative Files

Describe all of the central host and local admin files, including their naming conventions where they have fixed names. Note that this subsection might go better in the File section; we'll see.

Also, mention the kinds of central and local errors that can happen, including invalidation of user information on a local computer, and cite the appropriate subsection of the errors section, e.g., Section 2.12.7.

2.6.10. Administrative Notifications

Unlike meeting notifications, there is no option to disable the notifications sent by administrators to registered users. The on-screen notifications appear immediately for all affected users who are running the Calendar Tool, or the next time an affected user invokes the Calendar Tool if it is not running at the time of notification. This policy applies to all administrator-generated notifications, specifically

- o* notifications for adding, changing, or deleting a user record
- o* notifications for adding, changing, or deleting a group record
- o* notifications sent with the 'Notify Users' command

Describe how notifications are queued up and appear en masse when the user starts the Calendar Tool.

2.6.11. User Communication with Administrators

To avoid the need for message-handling functionality in the Calendar Tool Administration program, there is no means for a regular user to communicate with an administrator via the Calendar Tool. To do so, the user must look up the administrators' email addresses using the 'Admin->List Admins' command and communicate via email, look up the phone number, or otherwise communicate with the admin outside of the context of the Calendar Tool.

2.7. Options

Preceding sections of the requirements have described options for controlling various aspects of Calendar Tool operation. The scenarios in this section cover the complete details of all Calendar Tool options.

When the user selects the 'Times and Dates' item in the 'Options' menu, the system displays the dialog shown in Figure 236. The items in the 'Options' menu correspond one-for-one with the tabs in the options dialog. When the user selects one of the menu items, the dialog is displayed with the corresponding tab selected. Details of each options tab are covered in the scenario subsections that follow, including explanation of the precise effects of option settings on displays and other aspects of Calendar Tool operation.

At the bottom of the options dialog is a row of buttons that is common to all of the tabs, meaning that the effect of the buttons applies to all tabs simultaneously. The 'Apply' button is used to apply any option changes entered in

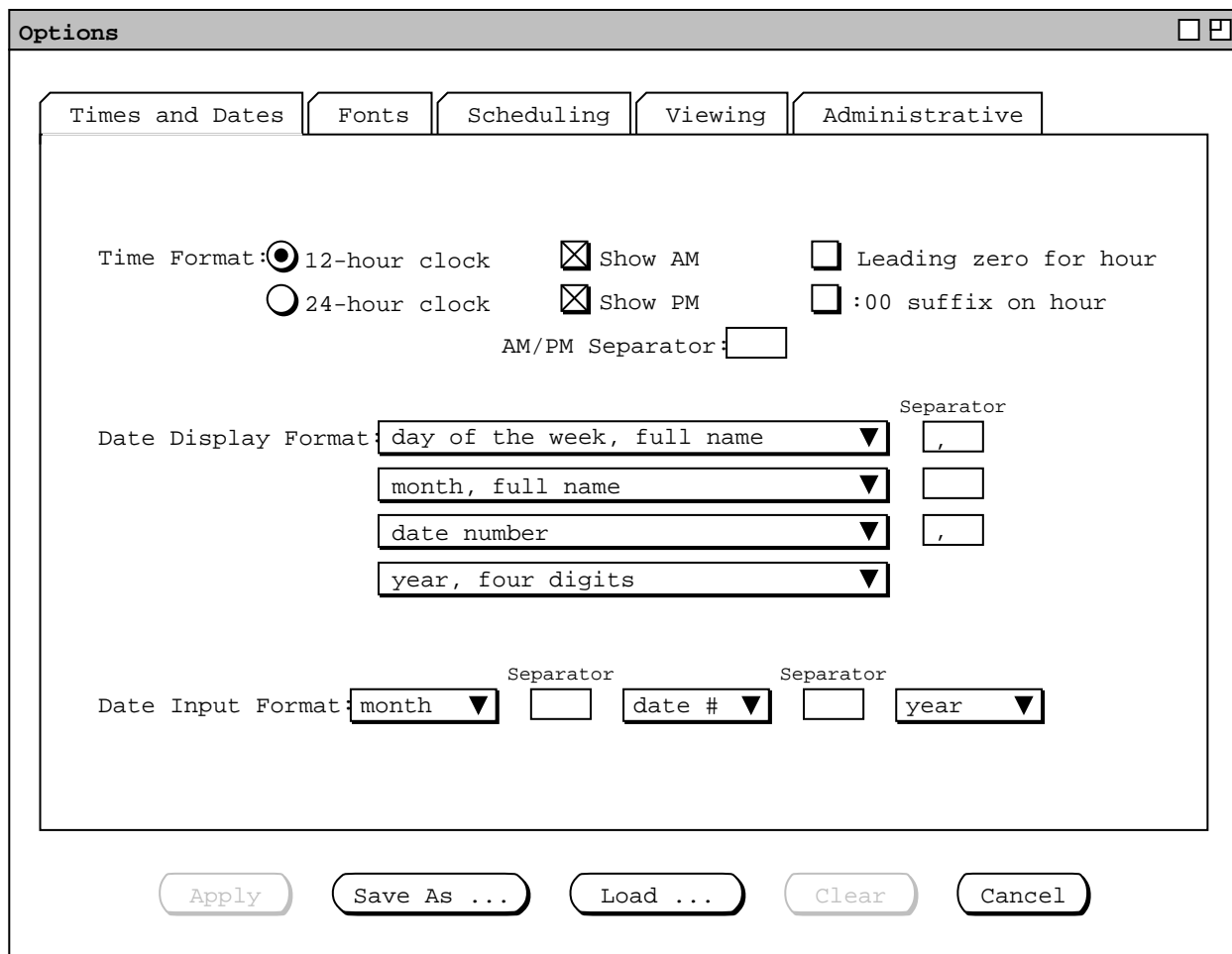


Figure 236: Options dialog.

any of the tabs. 'Apply' is initially disabled, becoming enabled when the user changes the options settings in one or more data-entry fields in one or more tabs. When the user presses 'Apply', the system applies the changed settings to the current calendar, updating all open display windows for the current calendar.

Since the command buttons apply across all dialog tabs, pressing 'Apply' has the affect of applying all changes made in any of the tabs, whether or not a changed tab is currently visible. Furthermore, switching between tabs does not automatically apply the changes in a tab that was previously displayed. All option changes remain unapplied until the user presses the 'Apply' button. As soon as the changed settings are applied, the 'Apply' button is reset to the disabled state. The 'Apply' button remains disabled until the user changes one or more settings.

The 'Save As ...' and 'Load ...' buttons allow the user to save and load copies of option settings to and from files. Details of saving and loading auxiliary data files are covered in Section 2.8.6 The 'Clear' button is used to clear any unapplied changes. When the user presses 'Clear', the system restores all dialog settings in all tabs to their most recently-applied values. The enabled/disabled state of the 'Clear' button coincides exactly with the 'Apply' button. Executing 'Clear' disables 'Apply', and vice versa. Pressing 'Cancel' in the option dialog clears any unapplied changes and removes the dialog from the screen, without applying any changes.

Option settings are calendar-specific. That is, each calendar has its own option settings. When more than one calendar is open, the system provides a separate options dialog for each calendar. Whenever the user executes the 'Options' menu command, the system (re)displays an options dialog for the current calendar. Each calendar uses a single options dialog. This means that up to N options dialogs can be displayed at any time, for N equal to the number of open calendars. When more than one calendar is open, the calendar name appears by default in the banner of each options dialog. The appearance or non-appearance of the calendar name in window banners is controlled by an option setting, as described in Section 2.7.4.2.

Hierarchical Overview:

Times and Dates

Time Display Format: one of

12-hour

24-hour

Date Display Format

Prefix

Position 1, Separator

Position 2, Separator

Position 3, Separator

Position 4, Separator

where each Position i , $1 \leq i \leq 4$, is one of

day of week, full name

day of week, three characters

day of week, one or two characters

date number

month, full name

month, short name

month number

year, four digits

year, two digits

and where Prefix and each Separator is a 3-char-wide text box

and a sample is shown at the bottom to clarify the look

Date Input Format:

month-number separator day-number separator year: on/off

day-number separator month-number separator year: on/off

year separator month-number separator day-number: on/off

where the last three are mutually exclusive;

in any of the formats, one input is treated as dd and

two inputs as dd and mm, in the appropriate order

and any number of spaces is always a legal separator

Fonts

Scheduled Items:

Font Name: pulldown of font names
 Size: pulldown of environment-specific ints
 All Other Text
 -- *same as scheduled-item text* --

Viewing

Item

Default display width and height for each of A,M,T,E
 Day (see structural viewing section for range and default values)
 Normal time range
 Time increment
 Increment height
 Events height
 Tasks height
 Show/hide 24 hours
 Show/hide exact time
 Show/hide dashed lines
 Show/hide extension arrows
 Proportional spacing on/off
 Display overlaps horizontally or vertically
 Default display width and height
 Week (only title, day range, and width/height apply to list style)
 -- all for day (put width/height last, after next two) --
 Show/hide title
 Normal day range
 Show/hide 7 days

Month

Show/hide title
 Default display width and height

Year

Default display width and height

Lists (pretty similar to custom list dialog, q.v.)

Appointments

Show these fields: (a la custom dialog, but for applicable fields only)
 Primary sort key: (a la custom dialog)
 Maximum number of items: (a la custom dialog)
 Initial Display Height: (a la custom dialog)
 Start Date: (a la custom dialog)
 End Date: (a la custom dialog)

Meetings

-- *same as Appointments* --

Tasks

-- *same as Appointments* --

Events

-- *same as Appointments* --

All Items

-- *same as Appointments* --

Windows

Initial Windowing Mode: one of the three menu selections

Magnetization:

Proximity: (default 4 pixels)

Spacing: (default 1 pixel)

Window Positioning: one of Calendar Tool Policy OS policy

(where Cal Tool policy is relative to menubar, and staggered down and to the right in multi-window mode)

Show current calendar name in window titles: one of

show in menubar title only

show in all window titles, when more than one calendar is open (default)

(where "all windows" means all windows of the current user, not other users or groups, which is consistent with the most recent clarification of the specialness of other user and group windows)

show in all window titles, always
 Show active filter name in window banners: one of
 show in menubar banner only
 show in all affected window banners (default)

Miscellaneous

Initial View:
 Current Calendar: one of
 item (first of the day, or empty if none)
 day
 week, table
 week, list
 month
 year
 All Calendars: OUT
 -- *same as current calendar* --
 ?button: Set All Calendars to Default?
 Show/Hide system-defined events (holidays, etc.): show/hide toggle
 External WWW Browser: application file chooser

Scheduling

Defaults (NOTE: banner of default-setting window must say "DEFAULTS" in it.)

NOTE: things are significantly muddled here, most particularly because having previous values be used for initial values don't make no sense. Also the notion that a default field could be "unset" is overkill and confusing. So, look in the actual scenario section for how things really work, considering all of the explanatory detail just below as historical fodder.

label: Click on a button to bring up a normal scheduling dialog. Then, to set a default value for one or more dialog fields, click on that field. The values entered (including empty values) are the defaults for those fields. To unset a field default, click again on the field (and it goes grey and cleared, where "cleared" is as defined in Section). Note that the initial settings in these default-setting dialogs are as defined by the admin in the admin-controlled options commands described in Section .

buttons: Appts ... Mtgs ... Tasks ... Events ...

I believe the following three are obsolete (see "And it works this way" below):

If the "All Previous Values" radio button is on, then the default value for all fields will be the value most recently entered for that field (thus ignoring any explicitly set defaults).

If the "Undefaulted Previous Values" radio button is on, then the default value for all undefaulted fields will be the value most recently entered for that field.

If the "No Previous Values" radio button is on, then the default value for all fields will be the value most recently entered for that field (thus ignoring any explicitly set defaults).

buttons: Use Previous Values (one of): all undefaulted none
 -- Or Better: --

Initial Values (one of): as entered the last time the dialog was displayed
 as specified in the default option settings
 where "Initial" means the first time the dialog is displayed and each

subsequent time it is displayed after having been closed.

And it works this way: 'previous' means use values as previously entered the most recent time the dialog was last displayed; 'defaults' means use default setting values where defined, previously-entered otherwise. Note that the 'previous' setting completely overrides any specified default values. To obtain the case where defaults completely override any previously-entered values, the user must explicitly specify a default value for every field, i.e., leaving no field in the default-setting dialog greyed out.

Overlaps: one of
 disallow entirely
 allow with warning
 allow with no warning

Meetings:

Show/Hide not-yet-accepted meetings: show/hide toggle (default show)
 When to Show Meeting Notifications: one of
 any time (default)
 while Calendar Tool is running
 at start-up only
 at exit only
 never
 How to Show Meeting Notifications: any of
 on screen, screen pos
 email
 Accept meeting notifications from Leaders of these Groups: one of
 all (default)
 list of group IDs
 none
 Accept meeting notifications from these Individuals: one of
 all (default)
 list of user IDs
 none
 Automatically remove expired meeting notifications: yes/no (default yes)
 Maximum number of entries in possible meeting times list: int (default 20)

Reminders

label: these settings are the default for all four types of scheduled item;
 settings can be made separately for each type of item in the
 defaults dialogs above

Remind? row that is common to all dialogs (thereby allowing default redisplay time to be se
 ?button: Set All Items to this Default?

Ignore All: yes/no

Reminder window placement: screen pos

Categories:

-- On/Off: Make current calendar's categories the default for new calendars
 default off

Admin

User calendar item limit: integer text field, with empty default

Allow leaderless groups: yes/no (if no, last ID in leaders list cannot be del'd)

Time to Complete Meeting Scheduling While Others Are Waiting: int minutes (default 2)

Defaults

buttons: User ... Group ... Location ...

Initial Values (one of): as entered the last time the dialog was displayed
 as specified in the default option settings

top-level buttons: -- see figure --

2.7.1. Times and Dates

The time and date options provide formatting control for the times and dates that appear in various Calendar Tool displays. When the user selects the ‘Times and Dates’ tab in the options dialog, the system updates the display to that shown in Figure 236. The ‘Time Format’ settings allow the user to select between 12-hour and 24-hour time, whether to show the "AM" and/or "PM" suffix, and what separation if any goes between the time and AM/PM suffix. The user can also set whether or not a leading zero character appears in front of single-digit hours and if the suffix ":00" appears in on-the-hour times. The default values for these settings are 12-hour time, show both "AM" and "PM" suffixes, one space of suffix separation, leading zero off, and ":00" suffix off. The ‘Separator’ may be from zero to three characters. When 24-hour clock is selected, the show AM/PM and separator fields are disabled, but not cleared.

The settings under ‘Date Display Format’ specify the format and relative order of the components in the display of dates. A full date string consists of four components:

- a. the day of the week, Sunday through Saturday
- b. the month name, January through December
- c. the date number, 1 through 28, 29, 30, or 31
- d. the year number, in two or four digits

Date strings appear in a number of calendar contexts, but not always with all four components. Table 10 indicates in which contexts dates are displayed and which date components appear in those contexts. For date strings with fewer than four components, the date display settings specify the format and relative order of those components that are present.

The four date-display list boxes have the same list of items, which is shown in Figure 237. The list items specify the

Date Context	Components
Date text fields in all dialogs and displays	date, month, year
Date label at the top of an item view	day, date, month, year
Date label at the top of a day view	day, date, month, year
Date label at the top of a week view	date range, month, year
Date label at the top of a month view	month, year
Month name within year view	month
Date columns in list views	date, month (short or number), year

Table 10: Date display contexts.

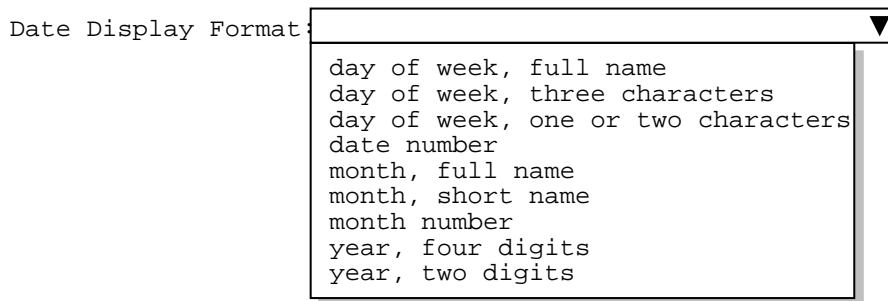


Figure 237: Date format combo box lists.

possible formats for each of the four components of a full date. The first three items specify the format of the day of the week. Table 11 shows the three alternatives. In all cases, the day name is capitalized. The fourth list item is for the date number, which is always an integer between 1 and 31. The fifth through seventh items specify the month name. The short name is the first three letters of the month, capitalized; the month number is between 1 and 12. The last two items specify the year as four or two digits. When the two-digit year format is selected, the two digits correspond to the current century or the previous century, whichever century contains the decade nearest the specified decade, where the decade is that specified by the first of the two digits.

Each of the four components of a date must specify exactly one of the day, date, month, and year. The separators must be at least one character each. The separator defaults are ", " ", and ", " .

In some displays there are restrictions on date formats, such that the options apply only partially or not at all. The format restrictions are the following:

- a. In list views, the month always appears in short form, either number or short name. If full-name format is selected for the month, the short name is used in list views.
- b. In week and month views, the day-name column headings are always three characters, independent of the option setting for day name.
- c. In year views, the date label at the top is always four digits, even if two-digit format is selected.
- d. In year views, months always appear by name, never as a number. If number format is selected for the month, the short name is used in year views.
- e. In meeting dialogs and item-level meeting displays, the time and date value in the 'On' field is always formatted as described in Section 2.4.1.1, independent of all time and date option settings.

The settings under 'Date Input Format' specify the order and separation of date components when input by the user. Each of the three list boxes contains the list of 'date #', 'month', and 'year'. Both separators must be the same single character, with the default a space character. The system provides the user a good deal of flexibility when inputting dates, in order to reduce the amount of typing the user must do. Once dates are input, they are confirmed for validity and converted to the format specified in the 'Date Display Format' settings. Complete details of date input rules, including the precise interpretation of input format settings, are covered in Section 2.11.5.

Figure 238 shows the user having changed the time and date option settings. The 'Apply' and 'Clear' buttons are now enabled since the user has made some option changes. Figure 239 shows the result of applying these changes on selected display excerpts. These excerpts cover all of the date contexts listed in Table 10.

2.7.2. Fonts

The fonts options provide control over the text fonts used in all Calendar Tool windows. When the user selects the 'Fonts' tab in the options dialog, the system updates the display as shown in Figure 240. The user can set font

full name	three chars	one or two chars
Sunday	Sun	S
Monday	Mon	M
Tuesday	Tue	Tu
Wednesday	Wed	W
Thursday	Thu	Th
Friday	Fri	F
Saturday	Sat	S

Table 11: Day of the week formats.

Options

Times and Dates | Fonts | Scheduling | Viewing | Administrative

Time Format: 12-hour clock Show AM Leading zero for hour
 24-hour clock Show PM :00 suffix on hour
AM/PM Separator:

Date Display Format:

Date Input Format:

Apply Save As ... Load ... Clear Cancel

Figure 238: Changes to time and date options.

Schedule an Appointment

Title:

Start Date: Start Time:

End Date: Duration: ^{hr} ^{min}

a. In a scheduling dialog.

Scheduled Appointment

◀ Today ▶ 08:00 9/25/98, Fri

b. In an item-level view (same format in day view).

Weekly Agenda

◀ Today ▶ Week of 9/20-28/98

c. In a week view.

Monthly Agenda

◀ Today ▶ 9/98

d. In a month view.

Yearly Calendar

◀ Today ▶ 1998

Jan							Feb							Mar						
S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S
				1	2	3	1	2	3	4	5	6	7	1	2	3	4	5	6	7

e. In a year view.

Appointments, sorted by Date

Title	Date	Time	Duration	Recurs?	Category	Location	Security	Priority
Racket Ball	9/17/98	08:00	1 hr	yes	personal	Rec Cente	title	optional
Office Hours	9/21/98	09:00	1 hr	yes	classes	14-210	public	must
Data Structure	9/21/98	10:00	1 hr	yes	classes	14-256	public	must
Data Structure	9/21/98	11:00	1 hr	yes	classes	14-232	public	must
Software Engin	9/21/98	15:00	2 hrs	yes	classes	14-301	public	must
Racket Ball	9/22/98	08:00	1 hr	yes	personal	Rec cente	title	optional
Research	9/22/98	09:00	8 hrs	yes	research	14-210	public	must
Office Hours	9/23/98	09:00	1 hr	yes	classes	14-210	public	must

f. In a list view.

Figure 239: Effects of time and date option changes on different displays.

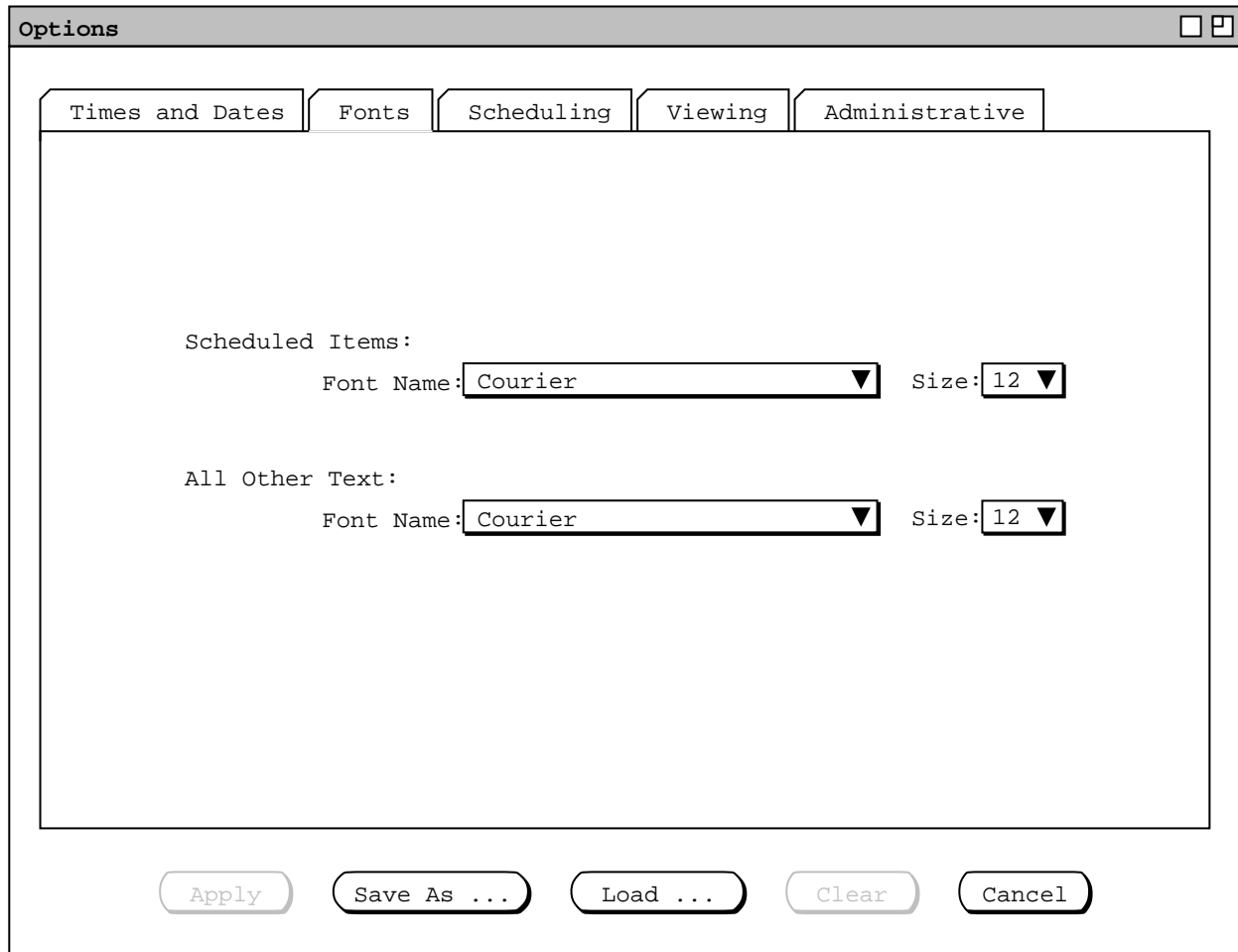


Figure 240: Fonts options tab.

options for two forms of text: scheduled items and all other text. The font settings under ‘Scheduled Items’ apply to the text of scheduled items as it appears in day, week, month, and list views. The settings under ‘All Other Text’ apply to all other text appearing in Calendar Tool display windows, including user-entered text as well as permanent system-generated text.

The same two data-entry fields are available for both forms of text. The ‘Name’ and ‘Size’ menus are platform-specific lists of available fonts and their sizes. With only a few exceptions, the same relative font size is used throughout all Calendar Tool display windows. The exceptions are the following:

- a. The font of the ‘Today’ button is two sizes smaller than other text.
- b. The font of secondary labels appearing above text-entry fields is two sizes smaller than other text, for example, the secondary hour and minute labels for the ‘Duration’ field in scheduling dialogs.
- c. The pulldown menu names in the custom filter dialog are two sizes smaller.

When the user changes the font size for ‘All Other Text’, the system maintains the two-sizes smaller relative size of these texts. In addition, the system performs some resizing of display window components when the user changes font sizes. Details of window resizing are covered in Section 2.13.5.2

The font options tab provides no control of font style or color. The normal, boldface, or italic style of text is controlled entirely by the system. The user can change the text color of scheduled items using the category editing features described in Section 2.5.5. No other control of text coloring is available.

The font options settings do not apply to the text in the banners of display windows. The format of banner text is considered to be under the control of the underlying operating environment, and is therefore not settable from within the Calendar Tool.

Figure 241 shows the user having changed font option settings. Figure 242 shows the result of applying these changes on the monthly view shown originally in Figure 22. The scheduled item text is displayed in Helvetica-14 and all other text is in Times-Roman-18. Font changes may involve size changes to some or all of the components in a display window. Details of window (re)sizing are covered in Section 2.13.5.2.

2.7.3. Scheduling Options

The scheduling options provide control over the commands in the 'Schedule' menu. When the user selects the 'Scheduling' tab in the options dialog, the system updates the display as shown in Figure 243. The second tier of scheduling tabs provide option settings for scheduling defaults, overlapping items, meeting scheduling, and reminders. Details of these options follow in the next four subsections.

2.7.3.1. Scheduling Defaults Options

The 'Defaults' tab allows the user to control the default values that appear in scheduling dialogs when they are initially displayed and when the user presses 'Clear'. To set the default values for a scheduling dialog, the user

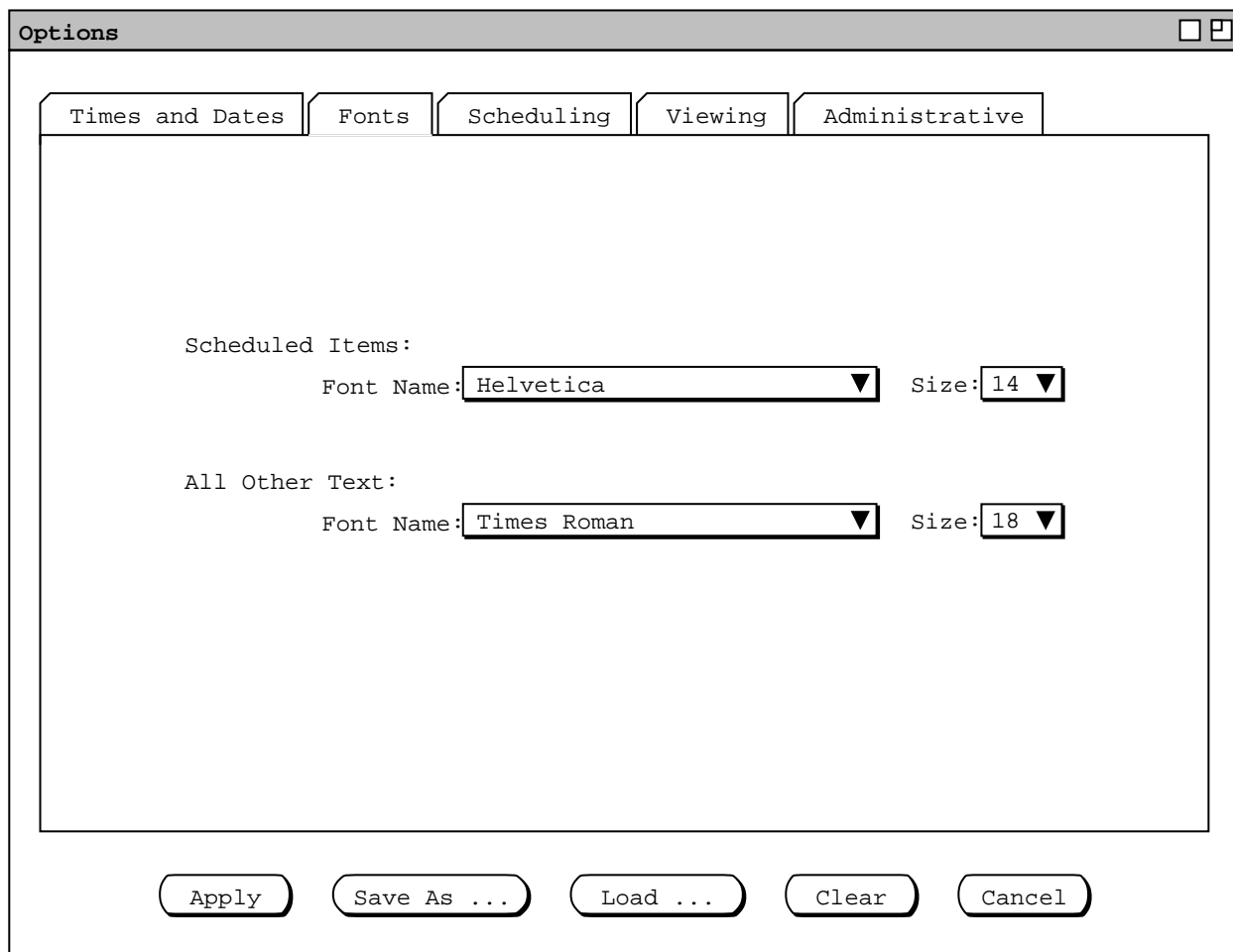


Figure 241: Changes to font options.

Monthly Agenda						
< Today > September 1998						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1 8-9 AM Rac 11 AM-2 PM	2	3 8-9 AM Rac	4	5
6	7 Labor Day	8 8-9 AM Rac 11 AM-2 PM	9	10 8-9 AM Rac	11	12
13	14	15 8-9 AM Rac 11 AM-2 PM	16 . Send class	17 8-9 AM Rac	18	19
20	21 1. Colloquium	22 Autumnal Equinox 8-9 AM Rac 9-10 AM Off 10-11 AM D	23 8-9 AM Sta 8-9:30 AM T 9-10 AM Off	24 Jim's Birthd 1. Prepare for 2. Buy some	25 8-9:30 AM 9-10 AM Off 10-11 AM D	26
28	27 1. Colloquium 9-10 AM Off 10-11 AM D	29 8-9 AM Rac 9 AM-5 PM	30 9-10 AM Off 10-11 AM D 11 AM-12 D			

Figure 242: Monthly view with font changed applied.

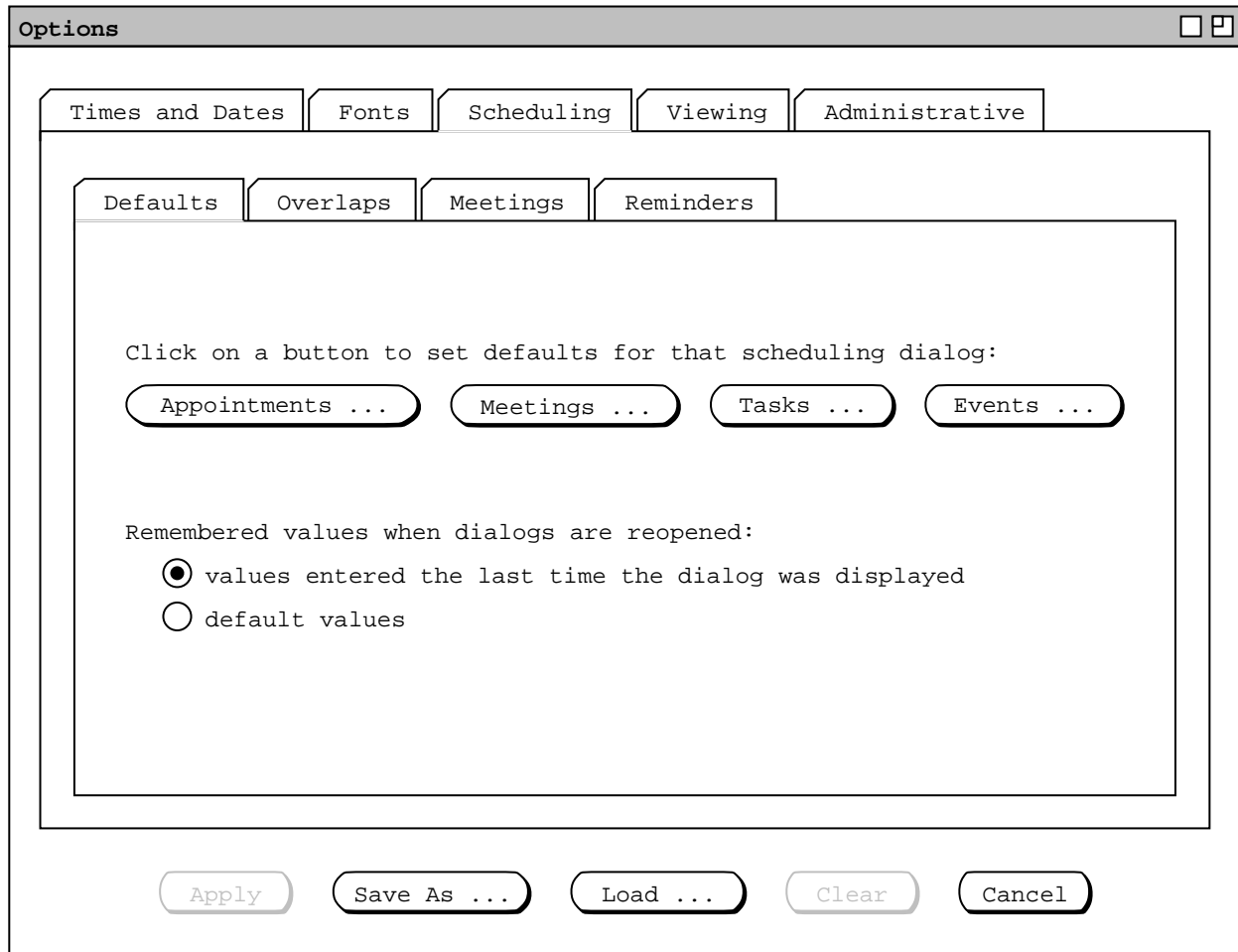


Figure 243: Scheduling options tab.

presses one of the buttons to bring the dialog up. When the user does so, the system displays a normal scheduling dialog that is augmented for the purposes of setting default values. For example, when the user presses the 'Appointments ...' default-setting button, the system displays the dialog shown in Figure 244. This dialog has exactly the same format as the regular appointment scheduling dialog shown in Figure 6. The banner of the dialog is prefixed with "Defaults for" to indicate that it is being used set dialog defaults, not actually to schedule an item.

Operationally, the regular scheduling dialog and defaults-setting dialog differ as follows:

- a. The word "today" can appear in date fields of a default-setting dialog, meaning the default value is dynamically replaced with the current date whenever the scheduling dialog is opened.
- b. Date fields in the defaults-setting dialog may have a suffix of the form "+ n", indicating a date increment in days.
- c. Data-entry fields are never disabled in a defaults-setting dialog.

Figure 245 shows the result of the user setting some appointment dialog defaults. The "today" value in the 'Date' field defaults to today's date. If the appointment is recurring, the 'End Date' value of "today + 14" defaults to the today's date plus 14 days. The duration default is set to 1 hour. The recurring checkbox is off, but the 'M', 'W', and 'F' checkboxes are set. This means that the default value for an appointment is non-recurring, but that if the user turns recurring on in the regular scheduling dialog, the default days are pre-set to 'M', 'W', and 'F'. The default security is 'title only', the default priority is 'optional', and 'Remind' is turned on with default values of '1 hour before' and 'on screen'.

Defaults for Scheduling an Appointment

Title:

Date: Start Time:

End Date: Duration: hr min

Recurring? Interval: S M T W Th F S

Category: Security:

Location: Priority:

Remind?

Details:

OK Clear Cancel

Figure 244: Appointments scheduling defaults dialog.

Defaults for Scheduling an Appointment

Title:

Date: Start Time:

End Date: Duration: hr min

Recurring? Interval: S M T W Th F S

Category: Security:

Location: Priority:

Remind?

Details:

OK Clear Cancel

Figure 245: Appointments scheduling defaults dialog with values filled in.

When the user presses 'OK' in Figure 245, the system records the entered values and removes the default-setting dialog from the screen; the options dialog remains on the screen. The recorded default values become part of the changed option settings that the user can apply by pressing the 'Apply' button in the options dialog. When the user executes 'Apply', the default settings come into effect for all subsequent scheduling operations. For example, Figure 246 shows the default state of the appointments dialog after the user has applied the settings in Figure 245. The default settings of "today" and "today + 14" have been replaced with specific dates. The 'Recurring' checkbox is off with the 'Interval' information therefore disabled, but the 'M', 'W', and 'F' day boxes are on. Since 'Recurring' is disabled, the 'End Date' is also disabled, but with a default date value entered. The disabled default values become active if the user turns 'Recurring' on. Figure 246 also reflects the changes to date format options made in Figure 238.

The initial default values for all scheduling dialogs are defined when the calendar tool is configured for installation (See Section 2.15.). Table 12 shows the default values for all fields in all four types of scheduled item. Figures 6, 83, 123, and 127 illustrate the initial default settings for each type of scheduled item.

The two radio buttons in the defaults tab are used to select the values that appear in scheduling dialogs when they are reopened after having been opened one or more times previously. The first selection specifies that values in reopened dialogs are those that were entered by the user the last time the dialog was open. The second radio button selection specifies that values in reopened dialogs are the dialog defaults, independent of the most-recently entered values. A scheduling dialog is considered to be "re"opened when the user re-executes a scheduling command after having executed 'Cancel' or 'Close' in the previously-open dialog. Deiconifying or moving an already open dialog to the front on the screen does not constitute reopening in this context.

Schedule an Appointment

Title:

Date: Start Time:

End Date: Duration:

Recurring? Interval:

Category: Security:

Location: Priority:

Remind?

Details:

Figure 246: Appointments dialog with updated default settings.

Data field	Initial default value
Title	empty
All Date fields	empty
All Time fields	empty
Duration	empty
Recurring?	off, weekly, all days off
Category	empty (none)
Location	empty
Security	public
Priority	must for appointments and meetings; 0 for tasks
Remind?	off, 15 minutes before, on screen
Attendees	empty
Details	empty
Minutes	empty
Completed	off

Table 12: Default field values for all four types of scheduled item.

2.7.3.2. Scheduling Overlaps Options

When the user selects the ‘Overlaps’ tab, the system updates the options dialog as shown in Figure 247. Overlap control is one of three settings: ‘allow’, ‘allow with warning’, and ‘disallow’. When ‘allow’ is selected, the user may freely schedule items with overlapping time periods, as long as rules for item uniqueness in Section 2.11.6 are met. When ‘allow with warning’ is selected, the user may schedule overlapping items, but when the user confirms the scheduling of such an item, the system displays a dialog of the form shown in Figure 248. The number N is the number of existing scheduled items with which the new item overlaps, and the text "**item(s)**" is singular if $N = 1$, plural otherwise.

If ‘disallow’ is selected for overlapping items, then the system treats the scheduling of such items as an error. The details of scheduling error messages are covered in Section 2.12.3 Complete details on the effects of allowing overlapping items are covered in Sections 2.3.1. and 2.13.5.3

The overlap option applies only to items scheduled by the current user, not to meetings scheduled by another user. Details of overlaps in meeting scheduling are covered in Sections 2.4.1 and Section 2.5.2.3.

2.7.3.3. Scheduling Meetings Options

When the user selects the ‘Meetings’ tab, the system updates the options dialog as shown in Figure 249. The show/hide setting for ‘Not-yet-accepted Meetings’ controls the display of meetings that a scheduler has created but the user has not yet accepted. The settings for ‘Accept Notifications’ specify when and if meeting notifications are accepted.

The third and fourth meeting-scheduling options control from whom meeting notifications are accepted. The options separately control notifications from group leaders and individuals who are not group leaders. The user can select to receive notifications from all, none, or selected leaders or individuals. Group leaders are listed by the IDs of the groups they lead. Individual users are listed by user ID. Both lists are comma-separated, per the syntax described in Section 2.11.3.

Complete details on the use and effects of meeting acceptance options are covered in Section 2.4.1.5. The options apply only to notifications for scheduled meetings, not for the administrative notifications discussed in Section 2.6. Administrative notifications are sent unconditionally; the user cannot choose to decline administrative notifications.

The fifth meeting-scheduling option controls where meeting notifications appear. They may appear on the screen at a particular position and/or via email, the default being on-screen. At least one of the checkboxes must be on, even

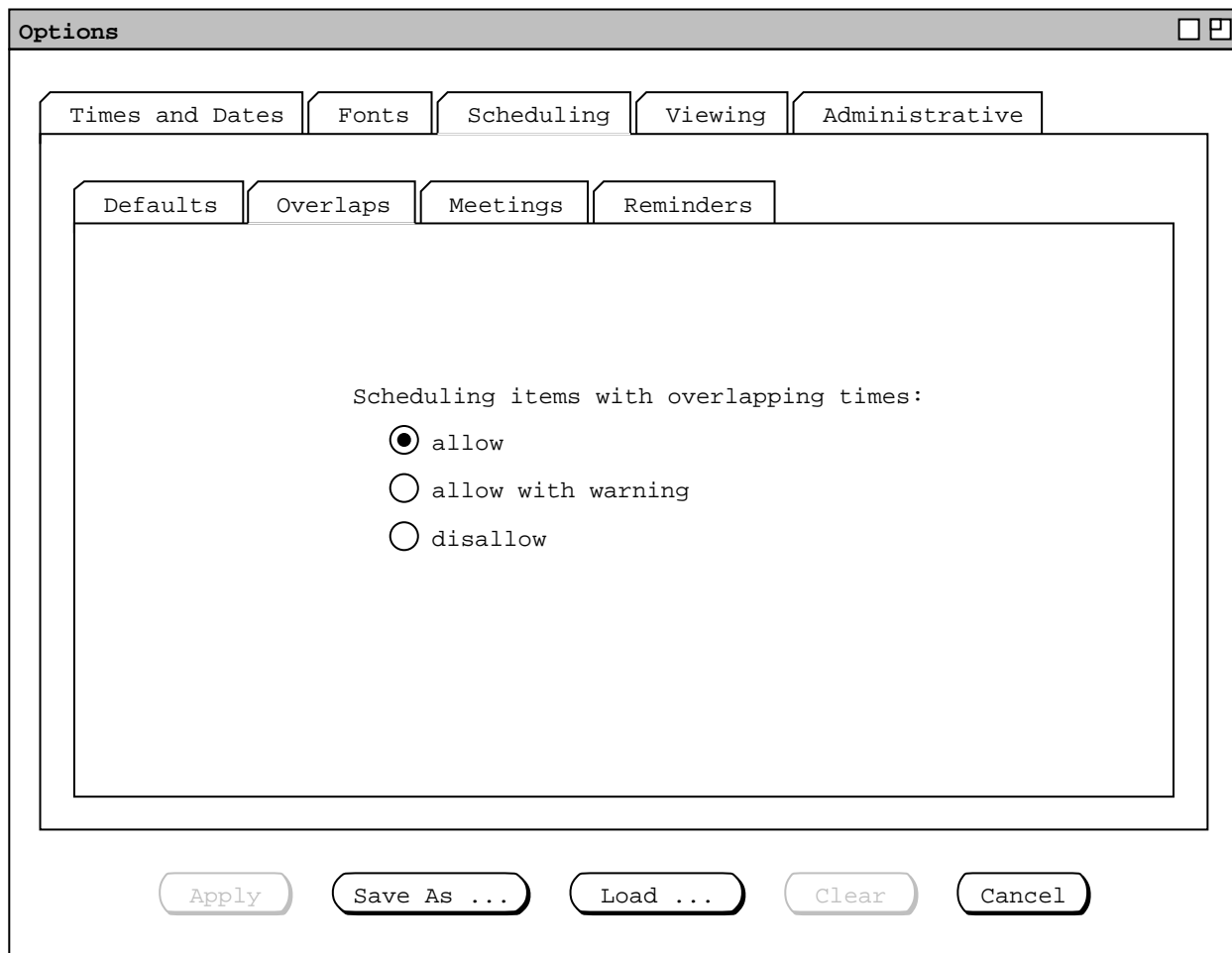


Figure 247: Scheduling overlaps options tab.

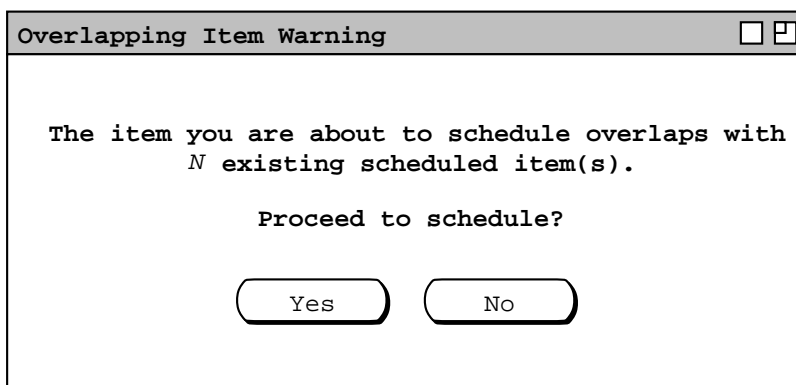


Figure 248: Overlapping item warning.

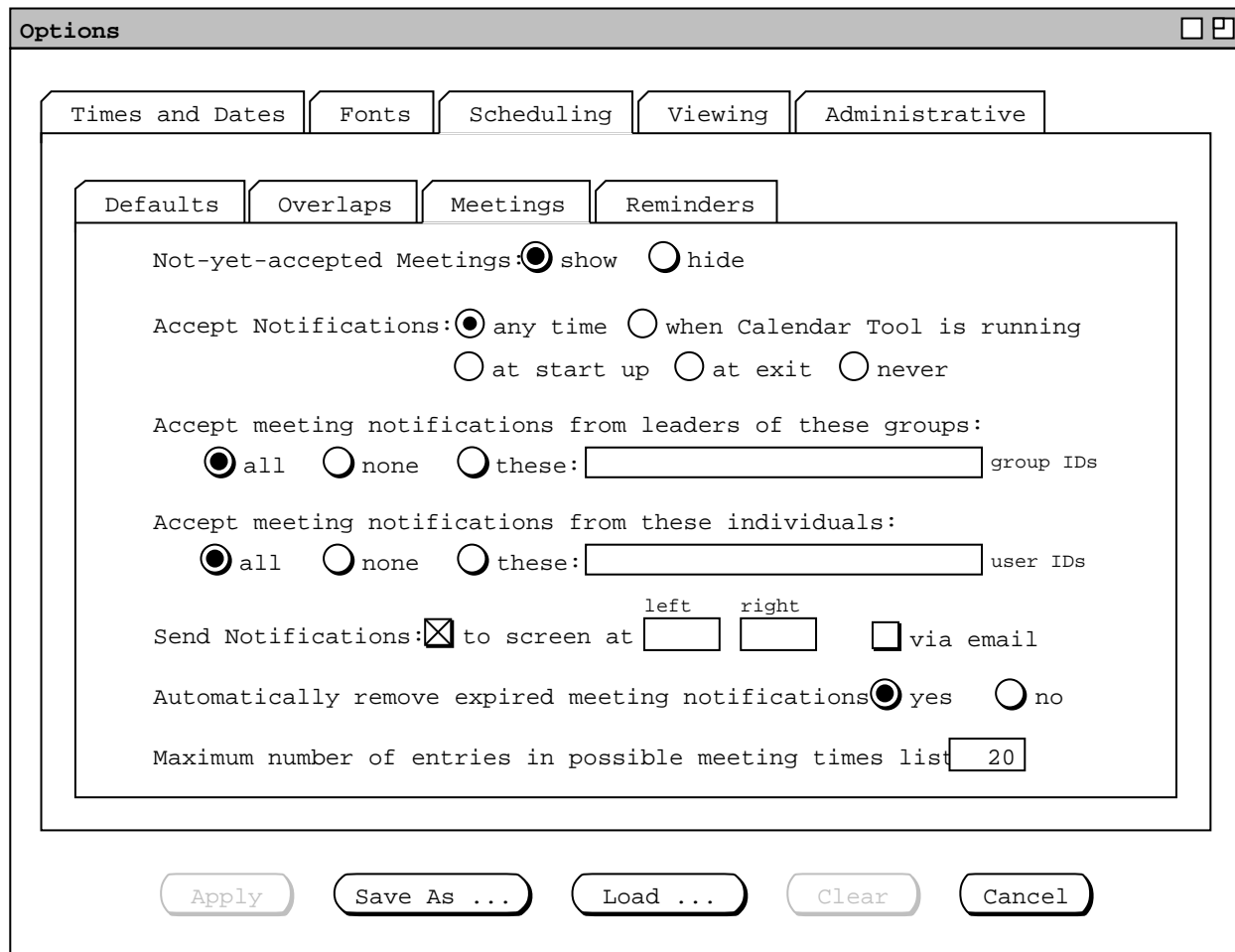


Figure 249: Scheduling meetings options tab.

if the preceding two options are set to 'hide' and 'never'. If the screen position is blank (the default), the system places the window per the current setting of the window positioning option, as described in Section 2.7.4.2. If the user enters specific values for the left and right screen positions, the system places the window according to the screen coordinate conventions given in Section 2.13.1.

The option to 'Automatically remove expired meeting notifications' controls whether meeting notifications are retained after the time for a meeting has past, as explained in Section 2.4.1.6.8. The last meeting-scheduling option specifies the maximum number of entries in a possible meeting times list, as described in Section 2.4.1.1.

2.7.3.4. Scheduling Reminders Options

When the user selects the 'Reminders' tab, the system updates the options dialog as shown in Figure 250. The 'Default Redisplay Settings' specify how soon a reminder is redisplayed if the user presses the 'Redisplay' button as described in Section 2.5.4. The setting to 'Ignore All Reminders' controls whether or not to show any reminders. If the default of 'no' is selected, reminders appear for all items in which the 'Remind' checkbox is turned on. If the user selects 'yes', then no reminders appear at all, even for items with 'Remind' individually turned on. During the time that reminders are ignored, the system discards any reminders that would have appeared. If the user subsequently sets 'Ignore All Reminders' to 'no', all future reminders will appear, but reminders with times already past will not appear. Hence, the user cannot "catch up" on missed reminders that would have appeared while all reminders were ignored.

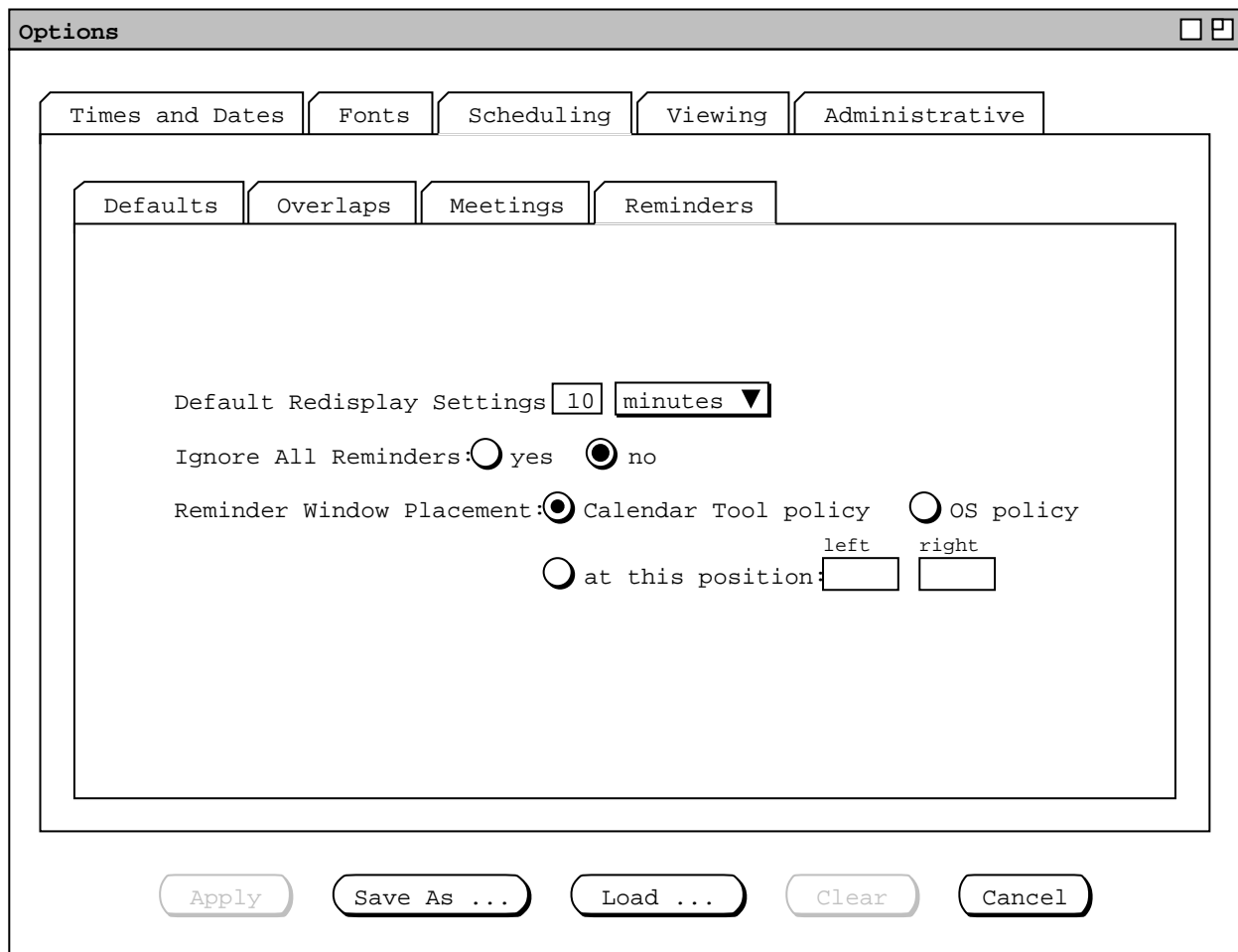


Figure 250: Scheduling reminders options tab.

The ‘Reminder Window Placement’ option controls where on the screen the reminder window appears. Details of screen positioning are covered in Sections 2.7.4.2 and 2.13.1.

2.7.4. Viewing Options

The viewing options provide control over the commands in the ‘View’ menu. When the user selects the ‘Viewing’ tab in the options dialog, the system updates the display as shown in Figure 251. The second tier of viewing tabs provide option settings for each each of the five calendar viewing levels, for lists, windows, and some miscellaneous options. The ‘Item’ tab has settings for the default dimensions of the item view for each type of scheduled item. The default values are as computed by the window sizing calculations given in Section 2.13.

The viewing options for the day through year levels are described and illustrated in Sections 2.3.1.1 through 2.3.1.4, respectively. Figures 252 through 255 show the presentation of these settings in the the options dialog.

Details of list, window, and miscellaneous options follow in the next three subsections.

2.7.4.1. Viewing Lists Options

Explain how the format of all five third-tier tabs is similar. Also explain that these options work a heck of a lot like custom lists. In fact, the only fundamentally new functionality in a custom list is the ability to have two or three types of items in the same list. Also, the ability to name custom lists is handy, since there’s only a single default

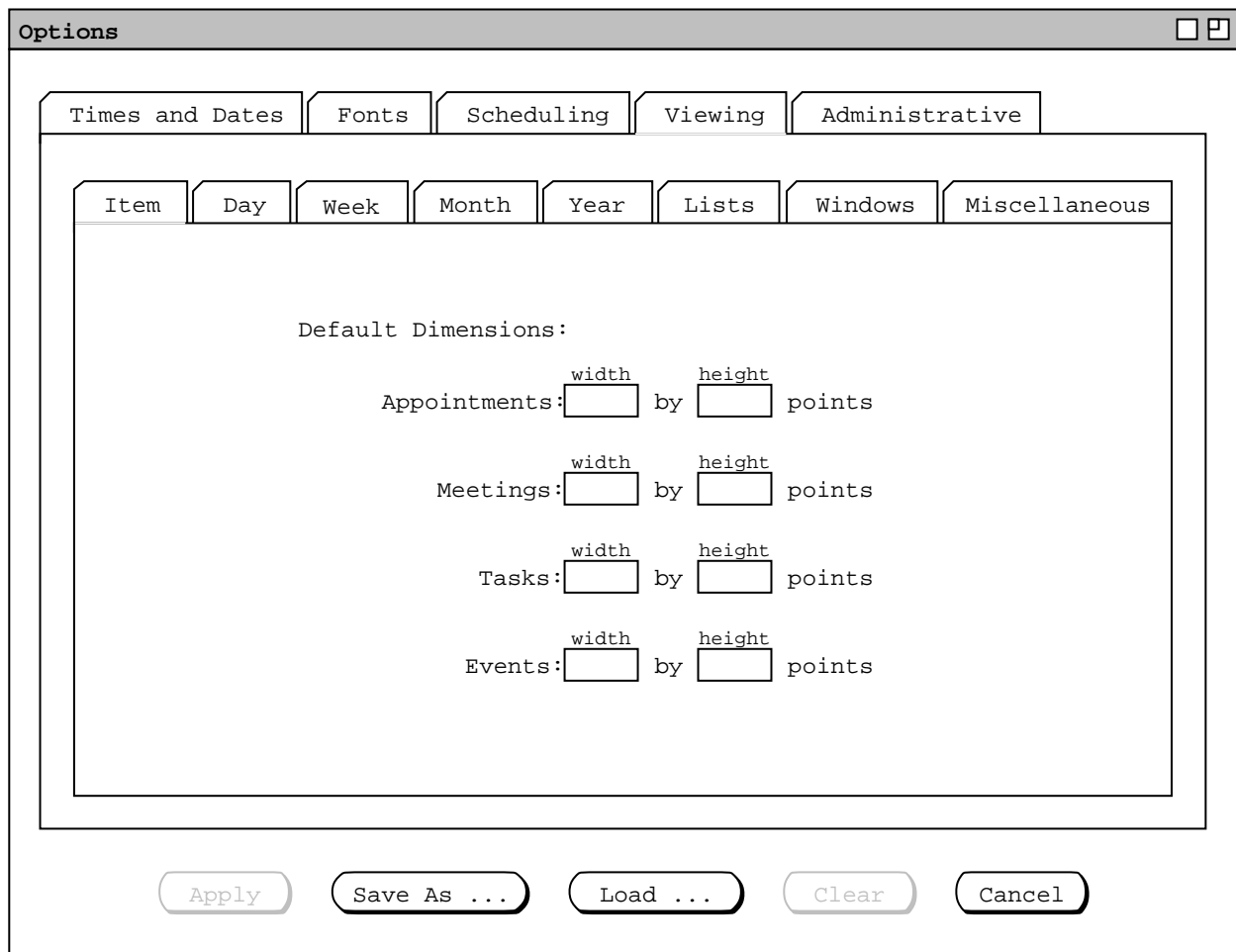


Figure 251: Viewing options tab.

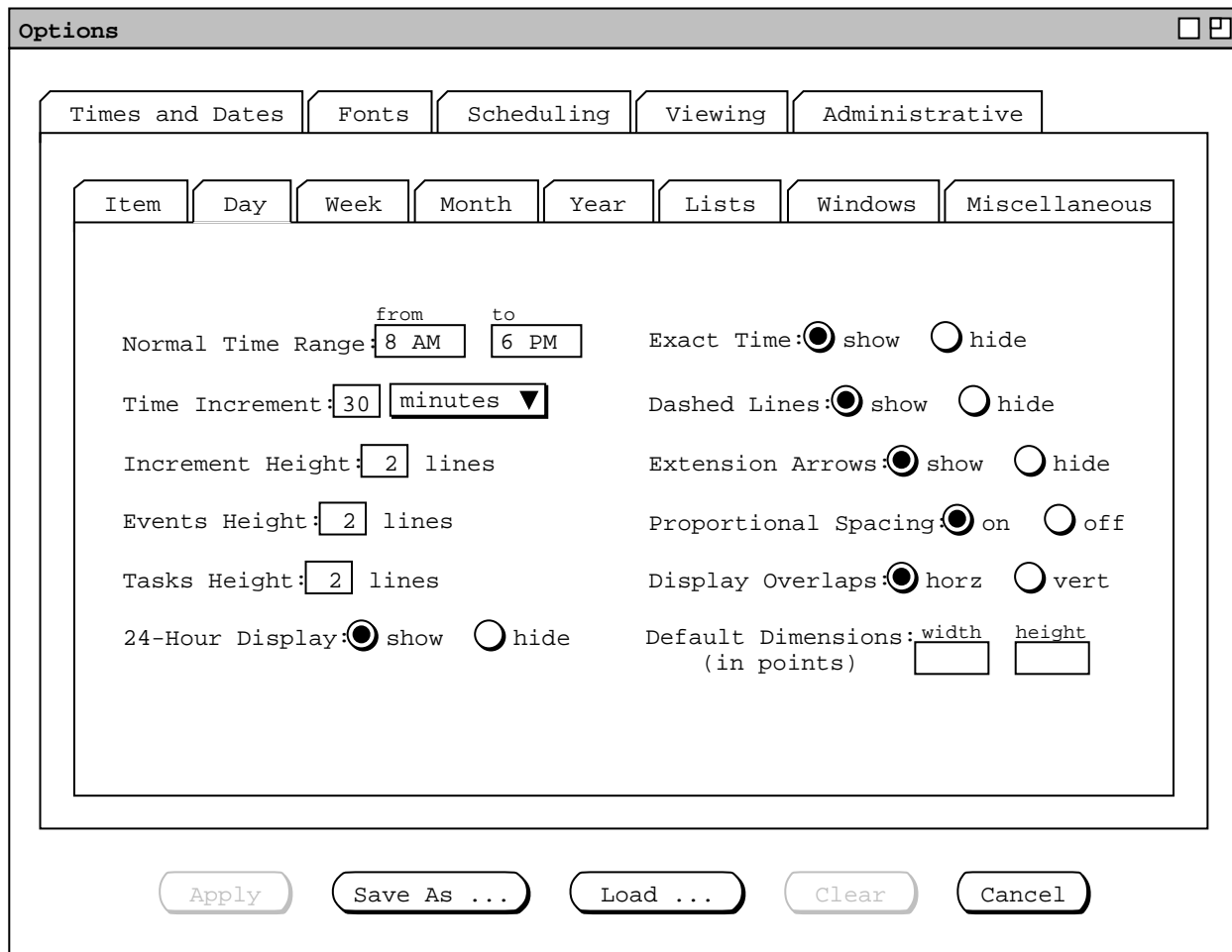


Figure 252: Viewing day options tab.

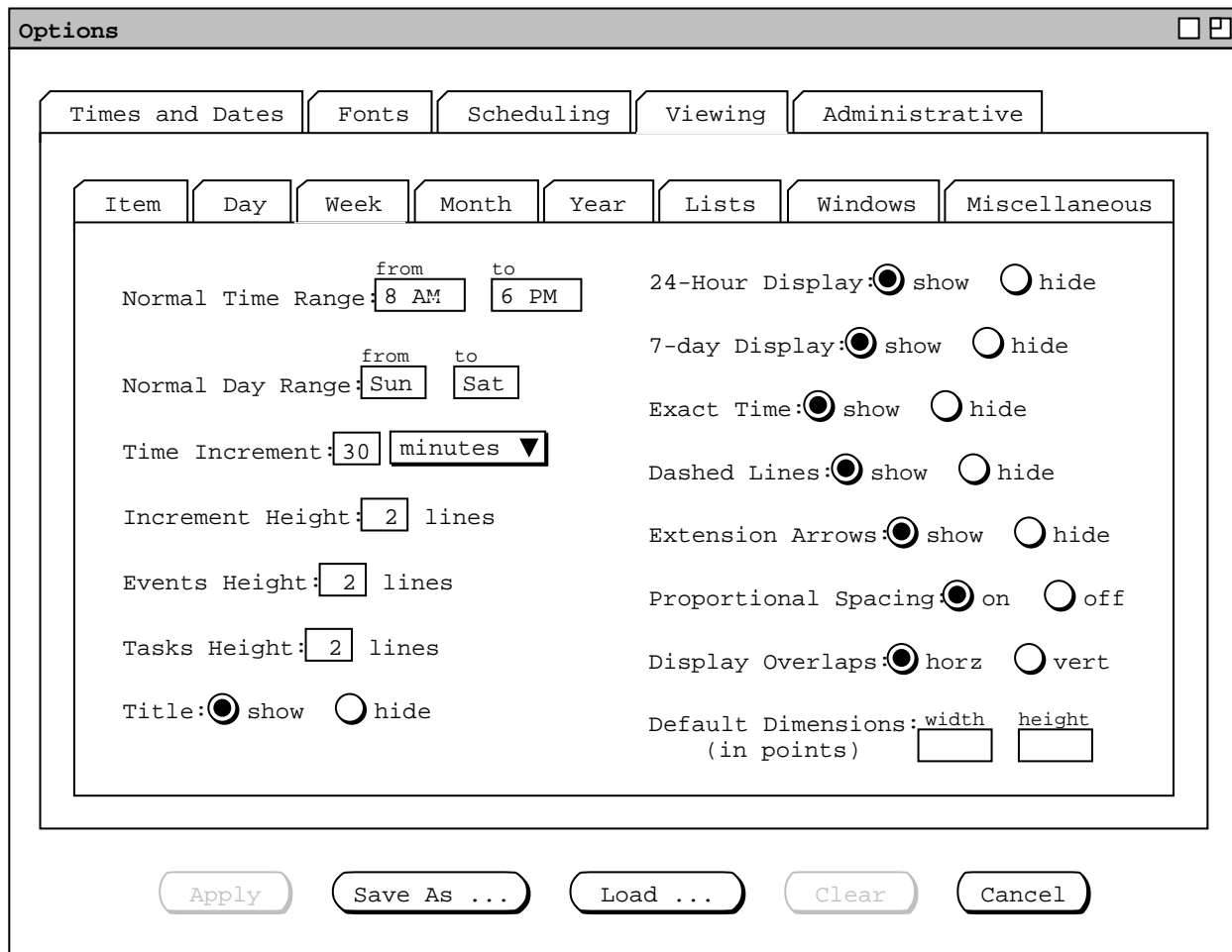


Figure 253: Viewing week options tab.

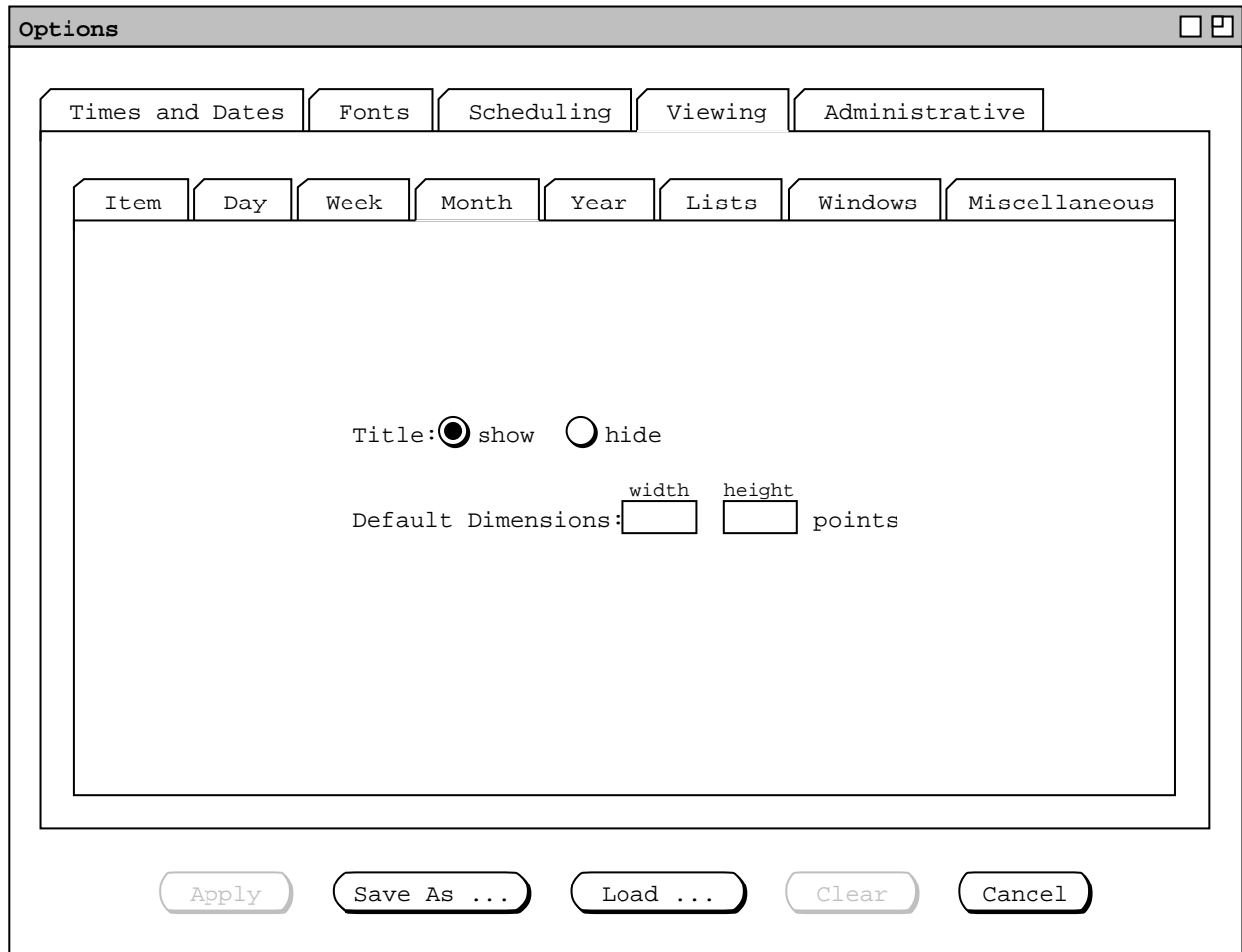


Figure 254: Viewing month options tab.

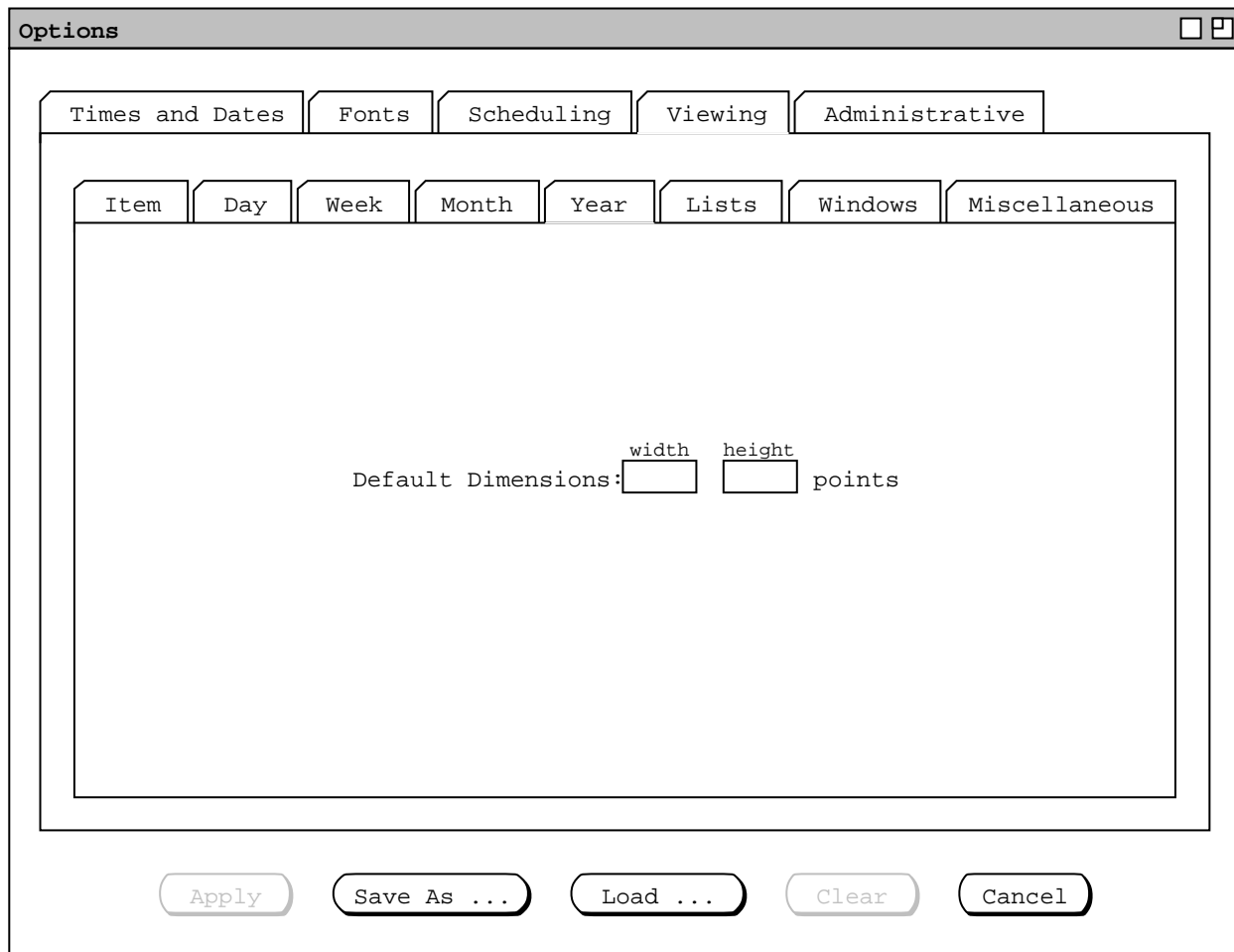


Figure 255: Viewing year options tab.

behavior for each type of list, but with custom list naming we can have multiple selectable listing behaviors. So, bottom line, the custom list functionality is not a huge add-on above list options, but it's fine since custom lists and options all work nicely alike.

OLD FODDER (from here to end of subsection):

The length of each of the four categories of list is controlled by the dialog shown in figure 261. The dialog controls allow the user to specify the number of items before and after a specified date. For example, the dialog settings shown in Figure 261 show the user having specified the following format for the display of item lists:

- o the appointments list is 100 elements long, starting with today's date
- o the meetings list is 75 elements long, starting at seven days before today
- o the length of the task list is unlimited, i.e., all tasks currently scheduled in the calendar are displayed in the task list
- o the length of the events list is 25 elements long, starting with today's date

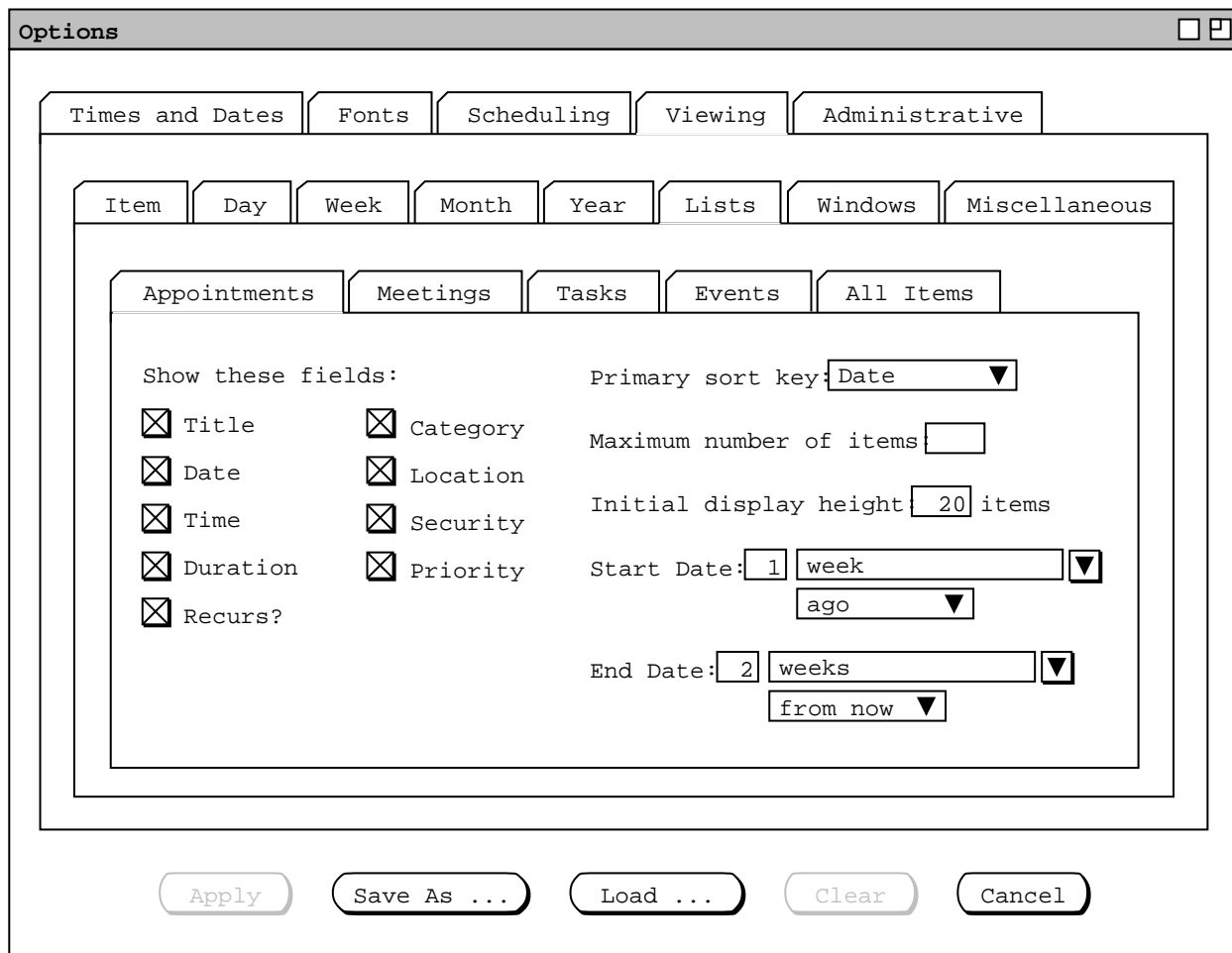


Figure 256: Viewing lists options tab.

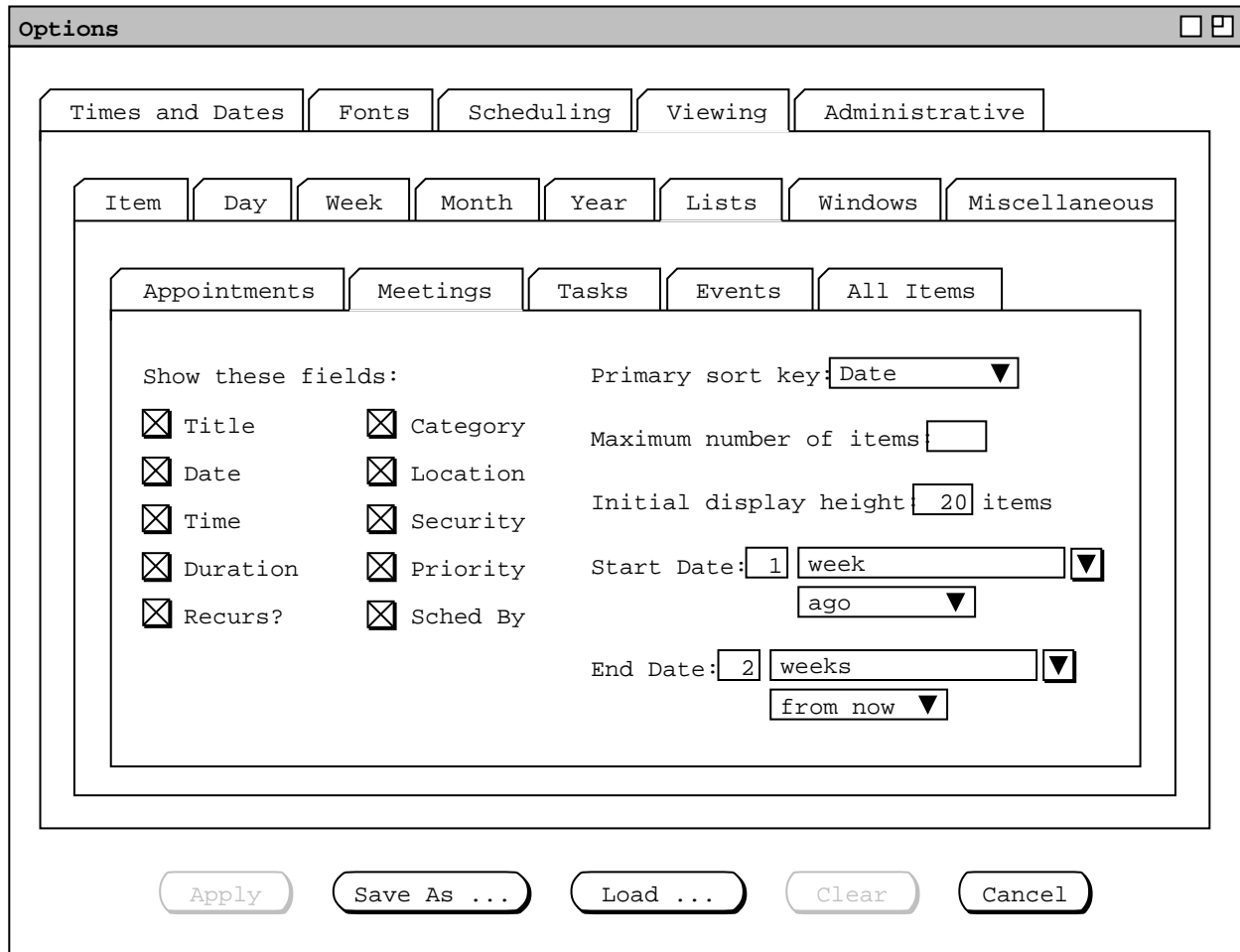


Figure 257: Viewing meetings lists options tab.

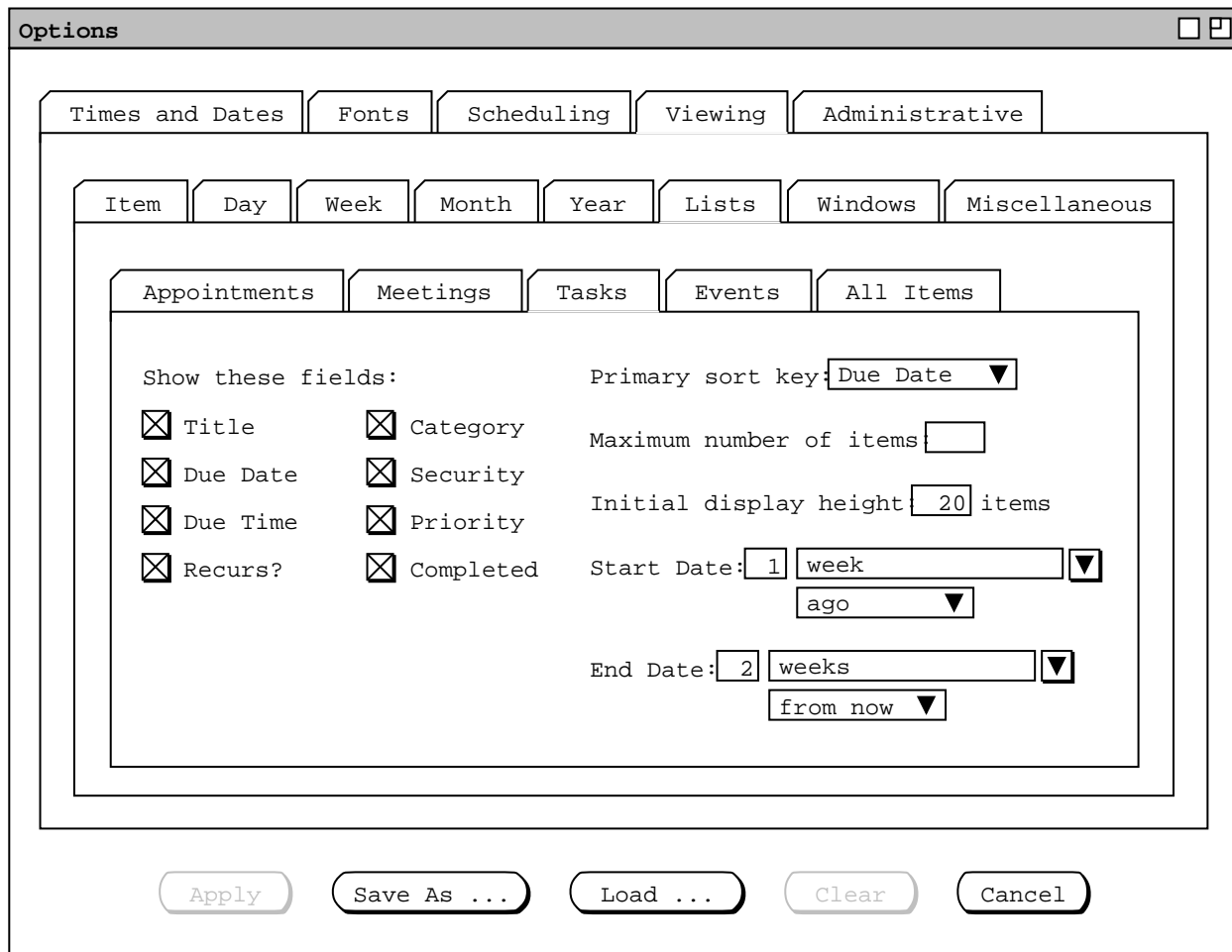


Figure 258: Viewing tasks lists options tab.

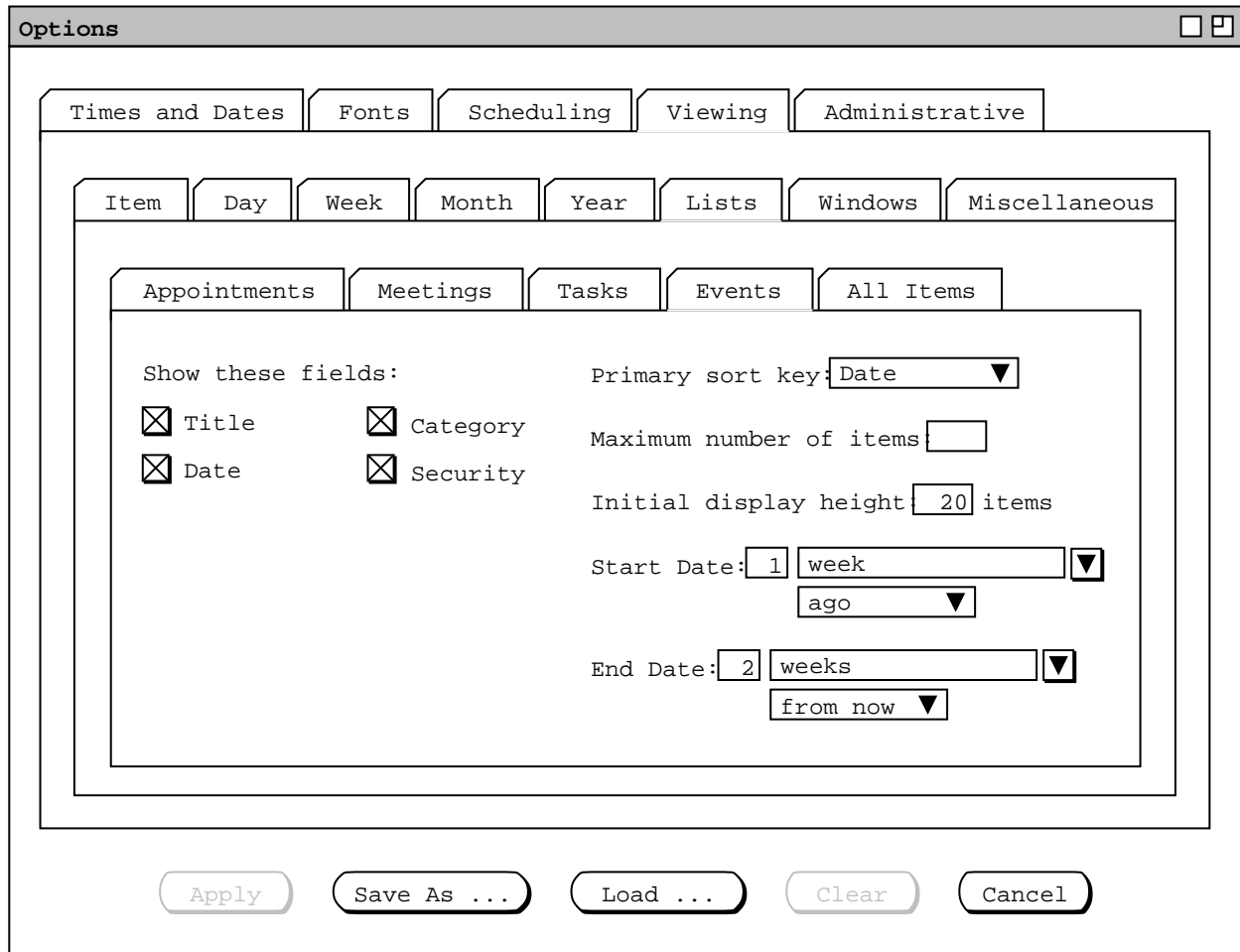


Figure 259: Viewing events lists options tab.

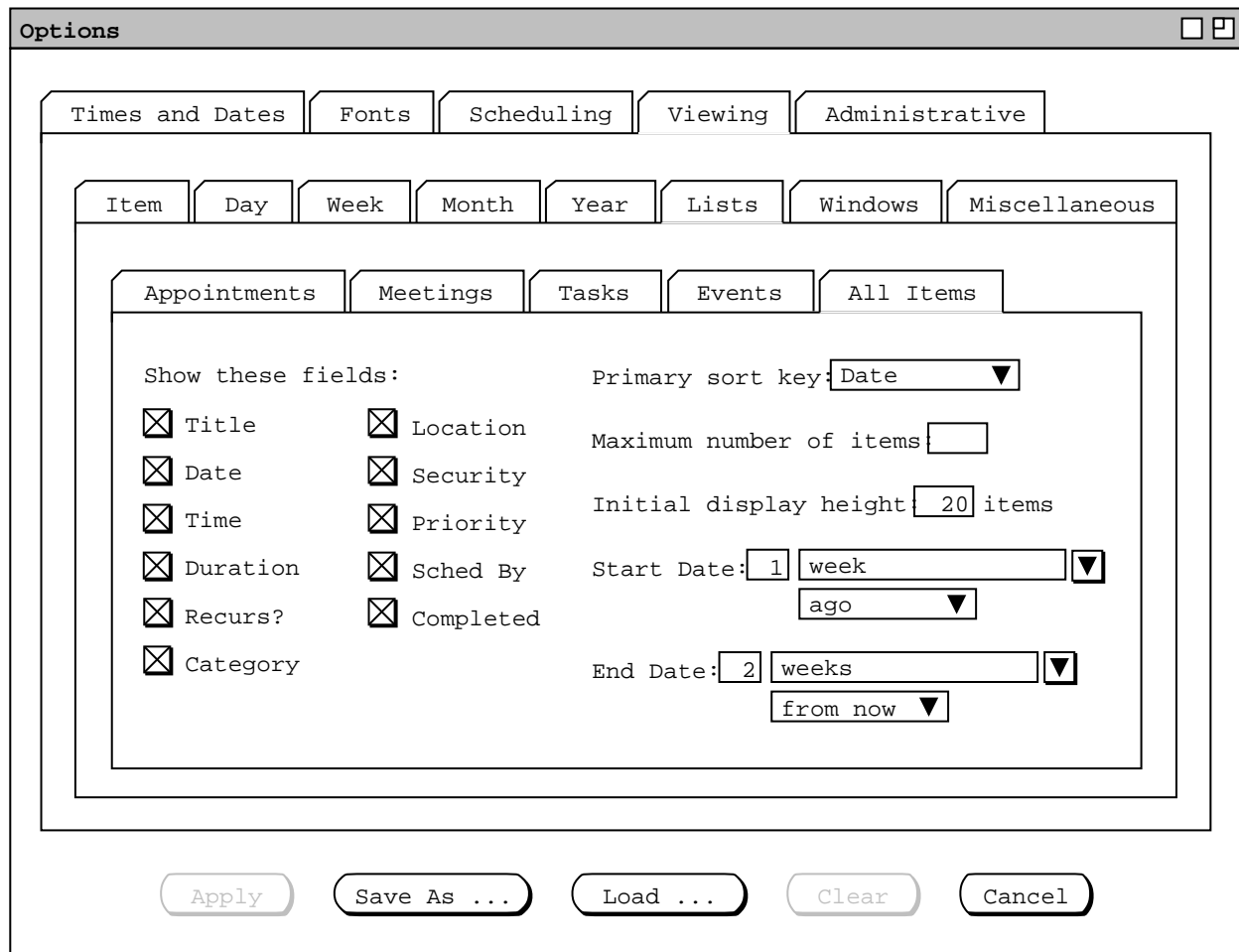


Figure 260: Viewing all-items lists options tab.

Hmm ... now that I'm doing this, I'm thinking that maybe we should just make this part of the regular filtering section, so that we maintain orthogonality. We need to think about this. An immediate question in this regard is do we really need to provide both a list-length feature as well as start/end date means to specify the number of items in a list. Curiously, this is pretty much the same concept as specifying start/end versus start/duration item lengths.

2.7.4.2. Viewing Windows Options

The 'in menubar banner only' setting does not apply to view windows for other-user and group calendars, since the banners of these windows unconditionally contain the user or group name.

2.7.4.3. Viewing Miscellaneous Options

Re. 'Initial View' level, to be consistent with the way things are explained in specific-date-viewing, the default level cannot be at the item level. Nor do I think it makes particularly good sense to have it be some list. Therefore, the default initial view level is one of four day through year levels, and perhaps a "no window" option. Remember also the bit about the initialization file and how it deals with the default view level always happening, including how a "no window" option value affects things initialization-filewise.

2.7.5. Administrative Options

When the user selects the 'Administrative' tab in the options dialog, the system updates the display as shown in Figure 263. The 'Auto-connect on file open' option controls whether the system attempts to establish



Figure 261: Viewing windows options tab.

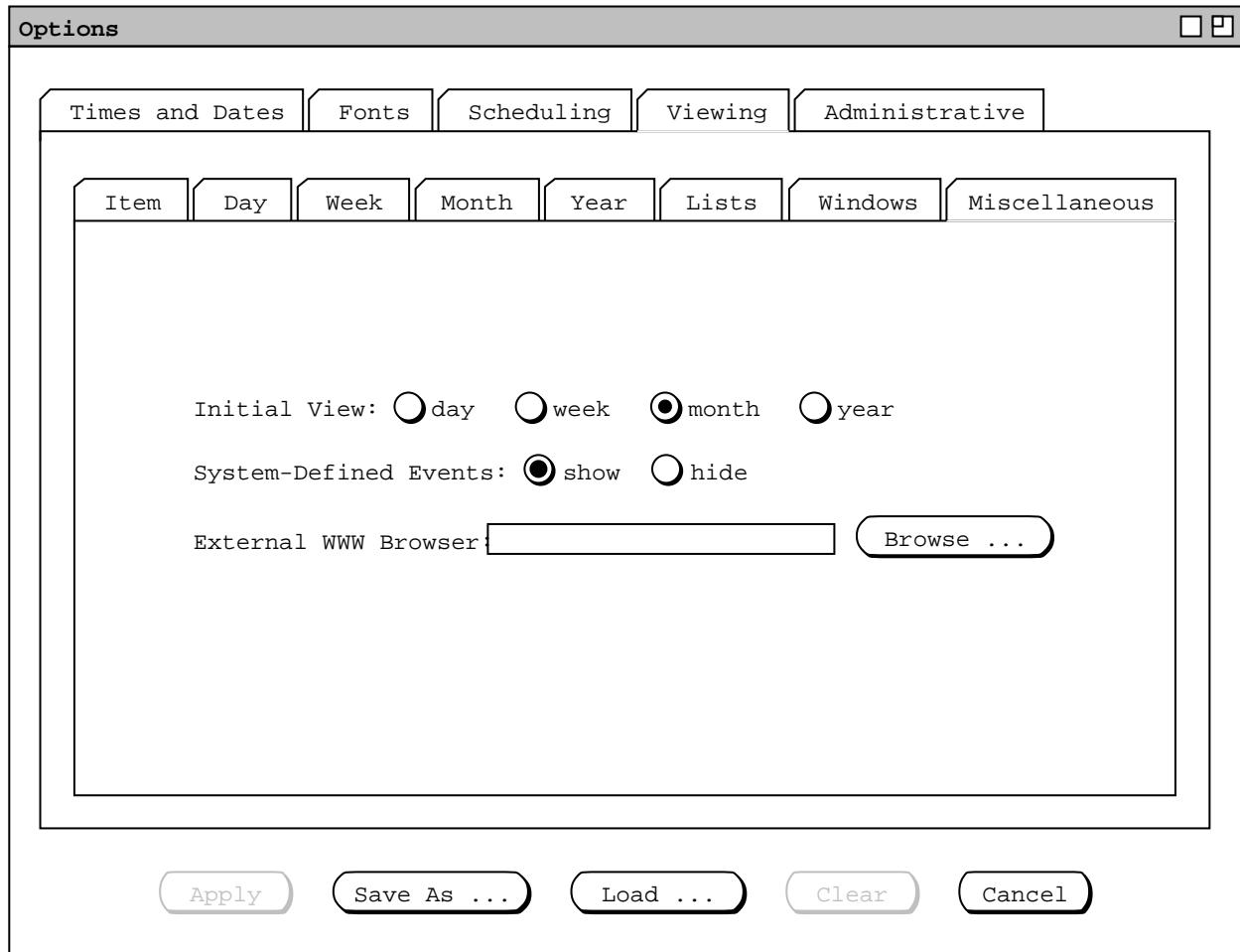


Figure 262: Viewing miscellaneous options tab.



Figure 263: Administrative options tab.

a connection with the calendar's central host computer whenever the calendar is open. When the default setting of 'yes' is selected, every time the user executes 'File Open' for the current calendar, the system tries to connect with the associated central host. Details of host association and connection are covered in Section 2.6.6.2. If the user selects 'no' for the auto-connect option, then the system does not establish a host connection when the calendar is open, requiring that the user establish the connection explicitly with the host 'Connect' command.

For a regular user running the Calendar Tool, there are no options for the administrative database commands, since these commands operate in read-only mode for regular users. For administrative users running the Calendar Tool Administration program, there are additional administrative options, as described in the immediately following section of the requirements.

2.7.6. Options in the Calendar Tool Administration Program

The options dialog that runs under Calendar Tool Administration has the same overall format as for regular users, except that the buttons along the bottom of the dialog are limited to, Save, Clear, and Cancel.

There separate dialog 'Save' and there is no 'Apply' button. Since there are no calendar viewing commands in the Calendar Tool Administration program, setting options does not apply to a calendar being viewed, but to the initial default values for those options for Calendar Tool users, as stored in the Options file.

NOTE: This paragraph needs to be fixed per the latest and greatest options set up. The default option values set by the administrator are available to users in two ways. When running the Calendar Tool, the user can select the 'Defaults' button in the 'Options' dialog. This allows the user to set options to the default values set by an administrator, as described above.

The other way that administrator-set options values are made available to regular users is through a Calendar Tool program distribution created by the administrator. That is, when an administrator creates a program distribution using the 'Admin Distribution' command, the option values that have been set by the administrator are installed in the distribution program as the defaults for that program.

Here's the deal for "inheritance" of options:

- a. First global options are set.
- b. Then user-specific options are set, possibly overriding admin options.
- c. Finally, calendar-specific options are set, possibly overriding either of the previous levels.

2.7.7. Operating Environment Conventions

Some operating environments may have specific conventions for the organization of the functionality described in Section 2.7 of the requirements. For example, this functionality is classified as "Preferences" in some environments, with conventions for the placement and layout of the Preference user interface. Implementors of the Calendar Tool may follow such conventions, as long as the hierarchical organization and content of the options functionality is maintained as described here.

2.8. Details of File Commands

Explain the deal with calendars, files, and their names up here.

2.8.1. New, Open, Close, and Close All

OK, OK -- it looks like we've covered a lot of the ground about "current" in the new(ish) Section 2.3.6.4.

Also need firm definition of "current", to wit:

- a. current "calendar" is the listed at the top of the 'View Calendars' list, as determined
- b. Question: If the current window is not a calendar, then is there still a current calendar? If so, it is presumably the calendar, if any, who's window is the topmost in the windows list. But what of the banner in the menubar in this case? Does it, potentially confusingly, list the name of the calendar most recently open? OK, the next item addresses these questions.
- c. OK, here's the deal about banner labeling, I think:
 - i. The menubar banner always contains the name of the calendar with the active or most recently active window (as defined by the order of windows in the 'Windows' menu.
 - ii. When calendar has been saved to a file, the calendar name is the same as its file name.
 - iii. When a (new) calendar has not been saved to a file, its name is "no name" or "no name [i]" for $i = 2$ to the number of unsaved saved calendars.
 - iv. When the
- d. Some more terminology: windows are "active"; calendars and files are "open".

Zero or more calendars may be open at any time. Need to be clear about the distinction between an open calendar versus an open calendar file. I'm pretty sure it's pretty much the same distinction as a buffer and file in emacs, however it seems that with the requirements we've developed this far, a calendar doesn't get a name until it's stored to a file, i.e., the calendar's name is synonymous (and co-existent with) its file name. This may actually point to a problem we've not yet (I don't think) adequately addressed, viz., having more than one unsaved calendar open at the same time, i.e., as the result of executing 'File New' more than once without executing any saves for the new calendars.

The name of the currently active file is listed in the banner of the command menubar. When no calendar file is open, the file name in the banner is listed as "none" (or perhaps, per musing above, something like "no name" is better).

If the active calendar is that of another user, that user's name is displayed in place of the file name, in the form "OTHER USER: *user name*".

When exactly one calendar file is open, its name is listed only in the banner of the command menubar, not in any other display window. When two or more calendar files are open, the banner of every display window is augmented with the name of the file on which the displayed data are or will be stored.

The effect of multiple files on the windowing modes is as follows. Every window says what file it's associated with. The current file is defined as the file associated with the current window. Whenever a different window becomes active, the current file is changed to the file of that window. The banner of the menubar is changed as necessary dynamically. The default per-level windowing mode applies separately to each

22aug01 Note: The problem with every window having the file ID in the banner is that with the newish listing and filtering banner paraphenalia, the banners may get too crowded. So, let's leave the filename out of the individual window banners and have it just be in the menubar banner, updating as noted in the preceding paragraph.

This paragraph is now WRONG. Note that File Close and Window Close are different commands. The former closes a file and all windows containing views of that file's data. The latter closes only the current window and does not close the file (ever), even if the window is the last or only window open for (i.e., belonging to) a particular calendar.

Mention that the solid/grey type deal applies to options loading, but instead of calendar file types shown in solid type, calendar option files are shown in solid type.

Some 'File Close' details:

- o* closing an file-less calendar with unsaved changes (items or other) does offer-to-save-as

- o* closing an filed calendar with unsaved changes (items or other) does offer-to-save

where "or other" is changes to categories, custom lists, custom filters, options, or in any pending scheduling dialog (maybe for the last of these, or maybe just have all pending scheduling quietly canceled).

2.8.2. Save, Save As, Save All

Note that 'Save' and 'Save As' are disabled when the current window is that of another user's or group's calendar.

NOTE: Given the new unified view of options as just another type of "other data", there should not be a specific offer-to-save for options that is fundamentally different that offer to save other unsaved changes, such as categories or filters. Therefore, as noted somewhere abouts, we can mention, but not really make special cases for, the kinds of unsaved changes that exist when the offer-to-save dialog comes up.

If the user executes 'File Save' for a calendar with unapplied aux data, the file-save dialog contains the message and radio buttons:

```
"There are X changes for this calendar."
(x) Apply these changes before saving
( ) Do not apply these changes before saving
```

If the user chooses the first radio option, the options are saved and the state of the options for the saved calendar go from unsaved to saved, which will be reflected in the options dialog. If the user chooses the second radio option,

Some 'Save As' Details:

- o* dir names shown in regular black type

- o* all file names shown in grey, but still selectable

- o* warning if OK pressed and extant file would be overwritten

2.8.3. Save Config

It's more than just the window config since it includes what files are open. I think the easiest way to define exactly what save config does is as a script file. This way we can deal with the tricky issue of having the display look the

same, but having the displayed content all be relative to today's date. If for some reason the user would like to have a config that is absolute datewise, of relative some other date than today, then she can tweak the generated script from 'Save Config', or of course write her own start-up script from scratch.

An issue we need to deal with is the relationship between the option setting of what we call "the standard default view" in the ui-overview. Here's what I think could work well: There is an option called 'Start-Up View Level' that specifies the default viewing level from item to year. If there is no init file in the user's home dir (per OS-specific definition of "home dir" and as discussed further in Admin section), then the Cal Tool comes up as described in the initial screen config in the ui-overview. Viz., there's a menubar and zero or one view window at the level specified by the setting of 'Start-Up View Level', open on today's date. If there is an init file or the tool is invoked with an start-up file in the command line or from a start-up file icon, then we could do one of two things:

- a. The system displays *no* windows automatically, thus ignoring entirely the setting of `default-init-view-level`; this means that the mere existence of an init file means there's no default display, and I don't think I like this; so let's try thing b ...
- b. The system proceeds to put up the initial window per `default-init-view-level`, and then goes about executing the init file. If the user really wants to get rid of the default initial window(s), she puts this code at the beginning of the init file, which code closes the initial window:

```
ViewWindows(1);
FileClose();
```

This works because of the rule that says the default window is put up before the init file is run, which means therefore that it's window 1 in the windows list, hence the arg of 1 to `ViewWindows` (assuming lists start from 1 like humans think, not from 0 like geeks think). But given this, here's an even simpler bit of code to do the same thing:

```
FileCloseAll();
```

which works for hopefully obvious reasons.

This all seems pretty darn clean. To be consistent, it seems with this scheme that the first like of the init file saved with 'Save Config' will in fact be the preceding chunk of code to close the default window, so that the config file starts with a blank slate.

2.8.4. Print

Sketch of some ideas: The print dialog should include at least the following options:

- o start date*
- o end date*
- o apply active filter?*

But woe nelly, I just had a look at all of the features in Claris, particularly for the "book" option, including editing its format. I think this will be an excellent place to draw the line and say that fancy printing will be part of future work.

So, here's the deal with printing. We'll keep it brutally simple by printing out in the plain text form of the current example-items file. In fact, we can use this file pretty much as is, except for the parenthetical remarks about edits made. We'll also need to add some kind of indication of individual instance changes to recurring items, which will be done in the same brutally simple form.

2.8.5. Exit

If the user exits or closes a calendar with unsaved options, then in addition to whatever offer-to-save file messages there may be, there are also offer-to-save options messages. Viz., there is one such message for each active calendar for which options have been applied but not saved. The text of these offer-to-save messages should most likely be integrated into the file offer-to-save dialogs, so two dialogs don't appear if both the file data and options need to be saved for a particular calendar.

2.8.6. Save As and Load in Editing Dialogs

NOTE: Adjust the following fodder pulled from the options section to apply uniformly to all four types of aux data. What might be nice is to use this options-specific scenario as the concrete example for all five types of aux data. See also the semi-epiphany in Section 2.11.26.

The 'Save As ...' button is used to save a copy of the current option settings in a selected file. Saving options in different files allows the user to create different option configurations that can be loaded using the options 'Load' command. When the user presses 'Save As ...', the system displays a dialog of the form shown in Figure 264. The default save-as directory is the user's Calendar Tool directory, the default name of which is ".CalendarTool". The four listed files are the standard files created when the Calendar Tool is installed, as described in Section 2.15.5. Section 2.8.2. covers complete details of the format and use of file saving dialogs, including the directory browsing list and the warning issued when an extant file would be overwritten by the save.

To save options to a file, the user enters the filename in the save-as dialog and presses 'OK'. For example, Figure 265 shows the user having entered the file name "PersonalOptions" as the location to which options are saved. The 'Save As' command does not execute 'Apply' or 'Save', it only saves a copy of the current options settings on the selected file. The applied and saved states of options are unchanged after execution of 'Save As'.

The 'Load ...' button is used to load a previously-saved options file. When the user presses 'Load ...', and there are no unapplied or unsaved options, the system displays the dialog in Figure 266. The names of valid options files appear in normal typeface in the directory list; all other types of files appear in greyed typeface. Section 2.8.1 covers further details on the format and use of file opening (i.e., loading) dialogs, including the directory browsing list and file-type validation.

To load options from a local file, the user selects or enters a filename in the 'File:' text field. The user can alternatively load the administrator-defined options file from the Calendar Tool central host computer, if the user is currently connected to an operational central host (see Section). If the user is not so connected, the 'Load from central host' checkbox is disabled. Figure 267 shows the user having selected to load from the "Personal Options" file. Figure 268 shows the case where the user selects to load from the central host. When the central host checkbox is selected, the File and Directory data fields are cleared and disabled. The user confirms the

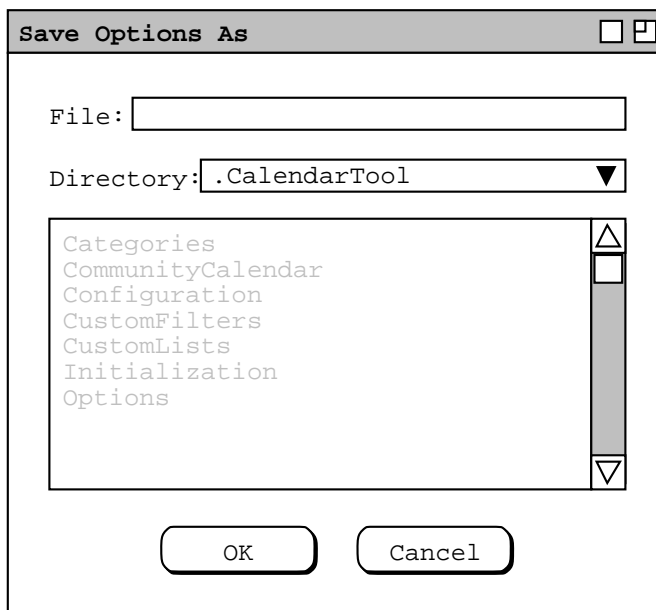


Figure 264: Options save-as dialog.

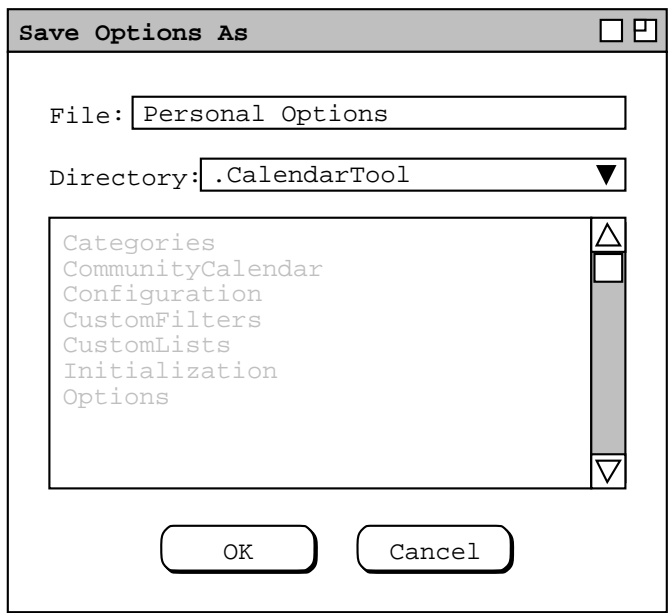


Figure 265: Saving options to a PersonalOptions file.

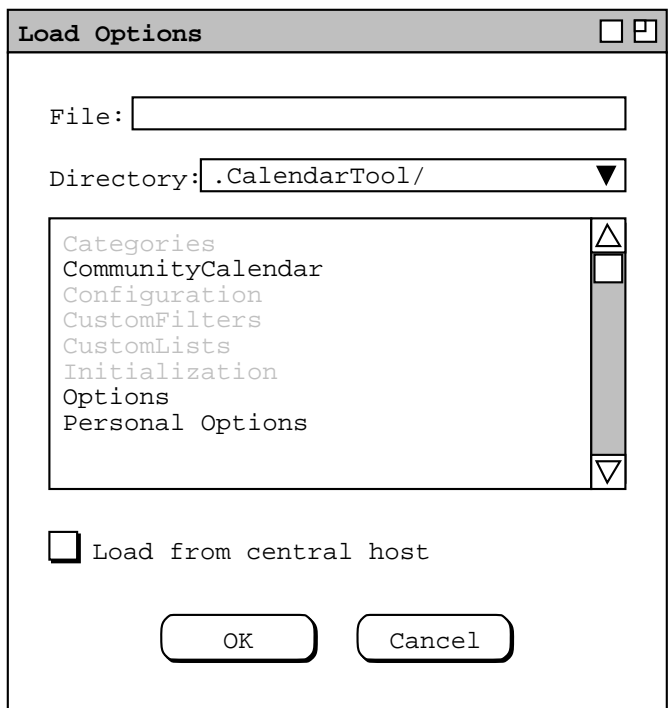


Figure 266: Options load dialog.

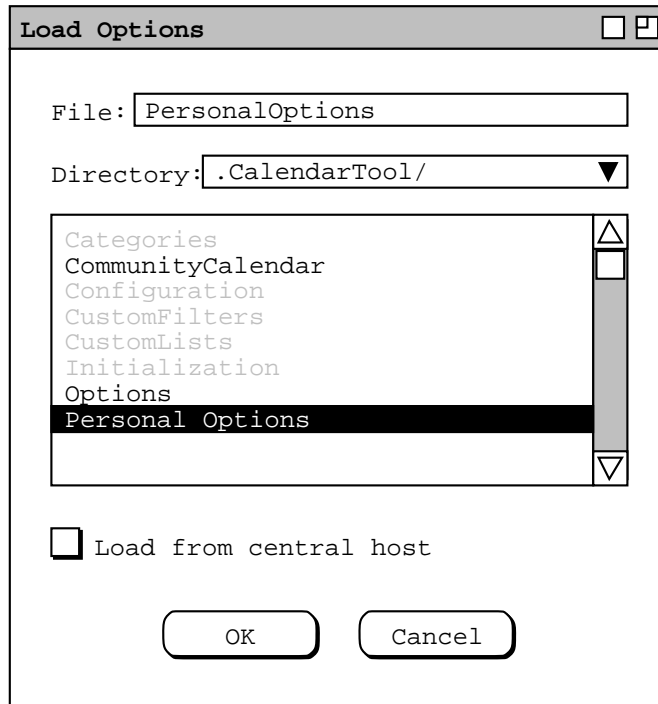


Figure 267: Loading from the Personal Options file.

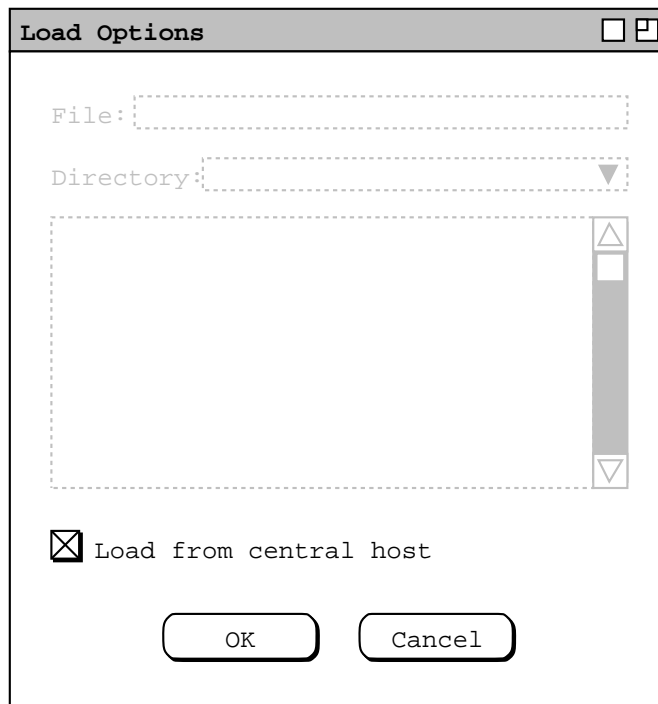


Figure 268: Loading options from the Calendar Tool central host.

load by pressing 'OK', whereupon the option settings defined in the loaded file become the current unapplied and unsaved settings in the options dialog. (To restore the Calendar Tool to the global default options configuration, the user can save the options settings loaded from the central host to the local `Options` file.)

If there are unapplied or unsaved option changes when the user presses 'Load . . .', the system issues the warning shown in Figure 269, where *affected* is one of "unapplied" or "unsaved", as appropriate. Unapplied changes are also unsaved, given that saving automatically applies. The user presses 'Yes' to proceed, 'No' to cancel the 'Load' command.

The 'Load' command does not perform 'Apply' or 'Save'. Once options are loaded, they are both unapplied and unsaved. Hence, loading has precisely the same effect as if the user had manually entered all of the loaded values in the options dialog.

Leaning towards fixing the following paragraph to allow load from (and maybe save to) two sources -- options files and (portions of) calendar files. If this happens, we need to fix above too. See Section 2.8.8 for an overview. So if we allow load from calendar file but not save-as to calendar file, here's how we accomplish save-into: open the calendar file, load, save (as opposed to save-as).

Even though calendar-specific options are stored within calendar files, the 'Save As' and 'Load' option commands operate only on separate options files, not on the options portion of calendar files. The system considers calendar files and saved option files as two different file types (see Section 2.8.8). When an extant calendar file is selected in the 'Save As' file-selection dialog, saving to that file replaces the previous file contents entirely, in just the same way as saving to any other extant file would do. The 'Load' options command can only be used to load previously saved options files, not the options portion of a calendar file. Sections 2.8.1 and 2.8.2 describe how file dialogs display inapplicable file types and the details of overwriting extant files.

Since options are calendar-specific, the 'Apply' and 'Save' commands operate only on one calendar at a time, namely the current calendar. Switching to a different current calendar always makes the options for the newly current calendar become active, without applying or saving options for the previously current calendar. Given this behavior, there is no direct way to save or apply the same option settings to multiple calendars en masse. To apply or save the same option settings to more than one calendar, the user must save the options to a file, make a new calendar current, load the saved options, and then apply or save the loaded options to the newly current calendar.

Any applied but unsaved changes are not saved by 'Cancel'. If the options dialog is subsequently opened on a calendar with unsaved option settings, the unsaved state of the options is reflected by the 'Save' button being enabled. If user closes a calendar with unsaved options or exits the Calendar Tool when one or more calendars have unsaved options, the system warns the user of the condition, as described in Sections 2.3.6.1 and 2.8.5.

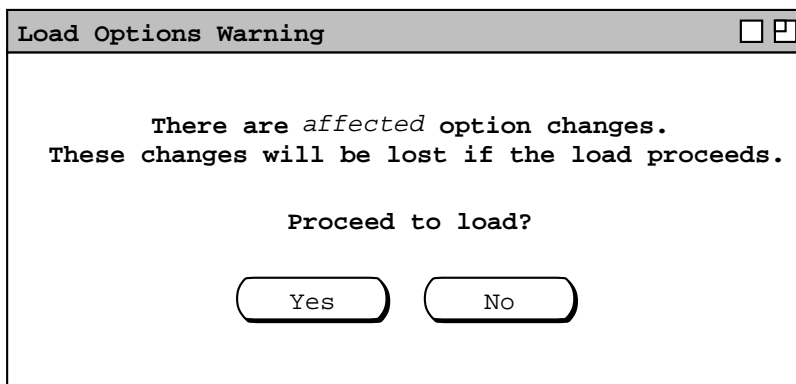


Figure 269: .

2.8.7. Standard Files

There are four standard fixed-name files used by the CalendarTool:

- a. "CommunityCalendar": [THIS IS WRONG NOW, need to decide if we want to keep CommunityCal around anyway as the "standard", or more likely "standard sample" normal calendar] the place where the central host expects to find the calendar that the user shares with the Calendar Tool community
- b. "Options": the standard options file that is used to set the initial options for all new calendars
- c. "Configuration": the file where the results of the 'Save Config' command go, that says what windows are up initially and where they are
- d. "Initialization": the command file read in by the CalendarTool whenever it starts up

Here's the order in which files are handled on start up:

- a. Any command-line files that do not exist are created as new Calendar Tool files, with the contents of the Options file copied into them.
- b. Any command-line files that exist as valid Calendar Tool files are opened, and their default view windows are opened at window-manager-controlled positions.
- c. Any command-line files that are not valid Calendar Tool files or not otherwise readable are ignored, with a warning dialog to this effect displayed on the screen.
- d. If there's a Configuration file and it requires opening any of the command-line files, these files are considered already opened. Even if multi-window mode is on in a file's options, there is at most one default view window opened for each file. If the positions of any of the command-line file default view windows are specified in the Configuration file, the windows are placed at those positions. The system then proceeds to process the rest of the configuration file. In this way, the start-up configuration is the union of the command-line files and the Configuration file.
- e. If there's an Initialization file, it is executed last. This means that the Initialization file can override any of the configuration performed as a result of command-line file specification and Configuration file processing. E.g., the initialization file could contain the command `CloseAll` to undo all of the command-line and Configuration file processing. Since there are no window placement commands in the Calendar Tool command language, there's no way to duplicate the window-placement functionality of Configuration file.

Given the mixing of `CommunityCalendar` as *the* global calendar, the following paragraph is WRONG. If the `CommunityCalendar` file is deleted or corrupted, it's as if the user sets 'Options.Meetings.When to Show Meeting Notifications' to 'never', but otherwise things are as normal. Note that deleting (or renaming) `CommunityCalendar` on the user's local computer *does not* delete any of the user's community calendar information on any central host on which the user is registered. In order to do this, the user must contact the system administrator(s) on the host(s) from the user wants to be unsubscribed. There need not be any explicitly identifiable `CommunityCalendar` file on the central host. It is up to the user to be prudent about saving backup copies of calendar files, including the `CommunityCalendar`. There is no Calendar Tool command to download the contents of a `CommunityCalendar` file from a Calendar Tool central host.

2.8.8. File Types

Discuss the different file types and the ramifications thereof, including that file types are used to do the filename greying thing in open/load dialogs, and that file typing may vary in different operating environments.

OK, try this: (See the semi-epiphany in options.me.) Also, as a non-annoying emacs-like behavior, we'll list all file names greyed out in all save-as dialogs, but all the user to select them. This way in contexts like the options save-as dialog, we can let the user select the standard Options file in the chooser for overwriting. Given this behavior, we may well want to specialize the overwrite message into two categories: (1) overwriting an extant file of the correct type (e.g., overwriting Options from the options save-as dialog or an extant cal file from File->Save As) (2) overwriting an extant file of the wrong type (e.g., overwriting Categories the options save-as dialog or an extant file of some non-cal-tool type from the File->Save As dialog. It think is is all pretty farging sweet.

2.8.9. Calendar File Contents

- a. scheduled items
- b. categories
- c. custom lists
- d. filter settings and custom filters
- e. windowing mode
- f. options

2.9. Details of Edit Commands

Organizationally, this section will probably need to be broken up into several files.

2.9.1. Undo, Redo, Repeat

2.9.2. Cut, Copy, Paste

2.9.3. Find

[Sketch: Among the more important items to define clearly in this section are the details of how Edit.Find works. We should check out the details of how it works in other systems.]

2.9.4. Text Command Interface

2.10. Help

For bulk and as horse pepples, the Rolodex help file is herewith dumped wholesale. The Rolodex help was left simple so we could get on with life. A feature that (may) need(s) to go here that is not in Rolodex help is the +/- explorer-style navigation of the contents window, and probably of the index if the index goes to greater than one level. Also, there (maybe) should be a Troubleshooting section at the end of the Users Manual. Finally, we may want to have indexed help allow multiple occurrences of the index phrase, most likely with prev/next buttons to move between them.

```
Search For: <reg expr>

Start Date:                               End Date:

X All Items      _ Visible Items Only
```

Figure 270: Find dialog.

```
Execute commands:
  -- multi-line command interpreter --

Load command file:
  -- file chooser --
```

Figure 271: Command dialog.

Here be the Rolohelp ...

As with the File and Edit commands, processing of Help varies in different operating environments. In particular, some operating environments have specific standards or conventions for the delivery of help information. The requirements here (and in Appendix C) are generic and operating-environment independent. What is specified here is the required content for the help information. This content may be reformatted and possibly reorganized, based on environment-specific conventions.

2.10.1. About

When the user selects About from the Help menu, the system displays the pop-up dialog shown in Figure 272. The specific contents of the About message are subject to change, based on the latest version, release date, and repository address. To remove the about message from the screen, the user presses the OK button at the bottom of the About dialog window.

2.10.2. Quick Help

When the user selects Show Quick Help from the Help menu, the system enters *Quick Help Mode*. In this mode, the system displays a small pop-up message when the user moves the mouse over elements of the display for which quick help is available. For example, Figure 273 shows the result of the user selecting Show Quick Help and then moving the mouse over the File menu. In quick help mode, the help messages appear with or without any mouse button depressed, whenever the mouse is moved over a display element for which quick help is available. In this way, the user can view quick help for display elements that may appear only when the mouse is depressed, such as menu items. When the user moves the mouse off of an element for which quick help is displayed, the quick help message is removed from the screen.

Figures 274 and 275 show two additional quick help examples, when the user has moved the mouse over the File->New menu item and over the OK button in the add-card dialog. Whenever quick help is displayed, the help message box appears immediately below the element to which it applies, with the left side of the message box aligned with the horizontal center of the element. Each element for which help is available has a *quick help activation area* defined. This is the area within which the mouse is moved to activate quick help for the element. Geometrically, the quick help activation area is the smallest enclosing rectangle around the screen element for which quick help is available.

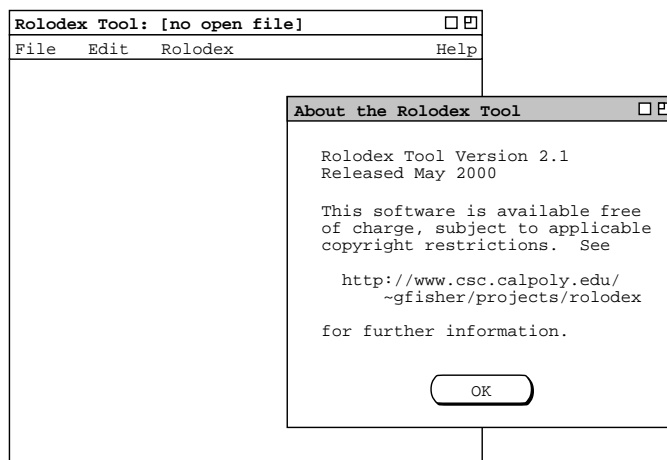


Figure 272: About the rolodex tool.

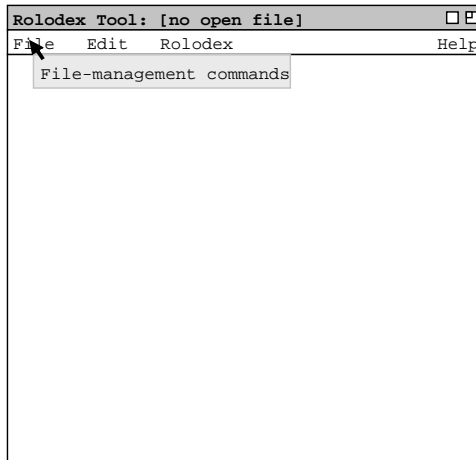


Figure 273: Quick help for the File menu.

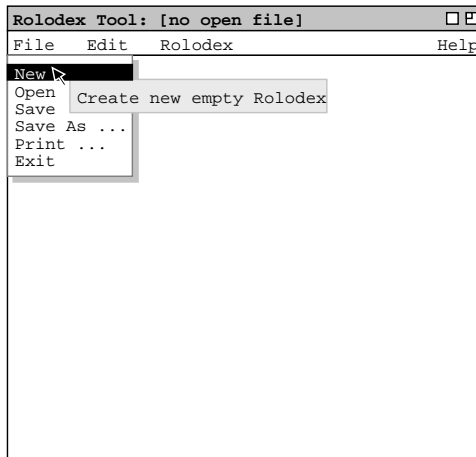


Figure 274: Quick help for File New.

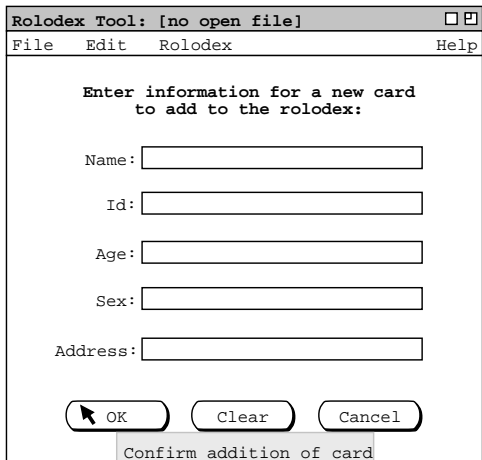


Figure 275: Quick help for add card OK.

After entering quick help mode, the system changes the Show Quick Help menu item to Hide Quick Help. When the user subsequently reselects the Help menu, it appears as shown in Figure 276. When the user selects Hide Quick Help, the system exits quick help mode so that quick help messages no longer appear. Each successive selection of the quick-help menu item toggles the menu item text between Show Quick Help and Hide Quick Help.

The complete list of display elements for which quick help is available is given in Appendix C.1

2.10.3. Detailed Help

When the user selects Detailed Help from the Help menu, the screen appears as shown in Figure 277. Detailed help is displayed in a pop-up window, separate from the main Rolodex Tool window. The detailed help window is divided into two horizontal panes. The narrower pane on the left has controls for selecting sections of help information. The wider pane on the right displays the selected information.

The detailed help control panel consists of three tabs for selecting information. The Contents tab displays a table of contents for the major topics of information. The Index tab displays an alphabetic subject listing. The Search

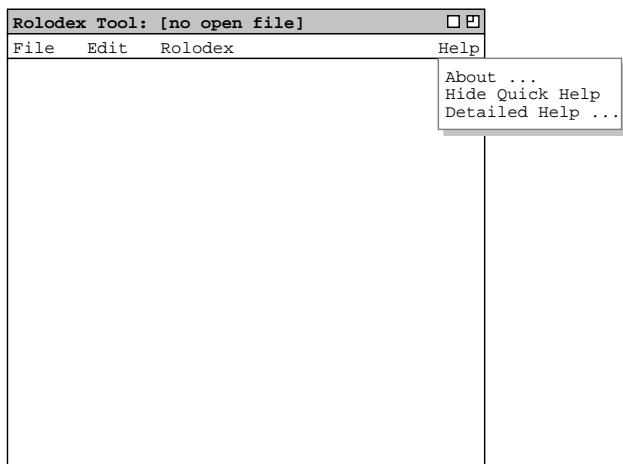


Figure 276: Hide quick help.

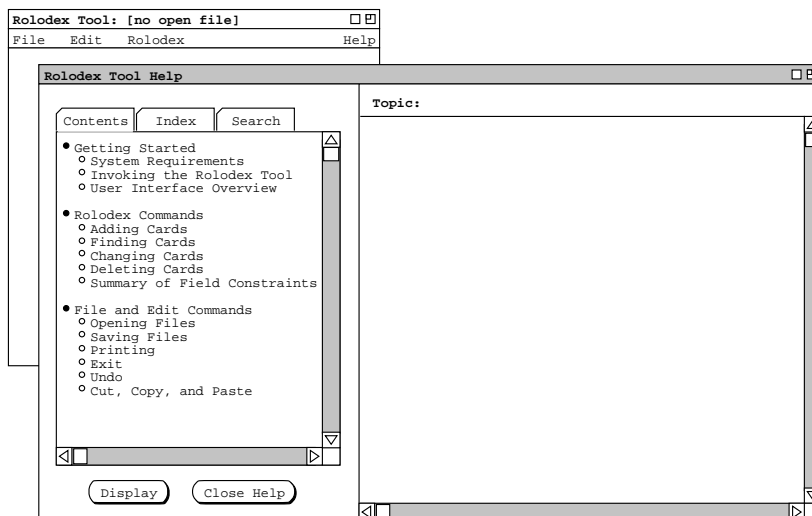


Figure 277: Initial detailed help screen.

tab allows the user to search for a keyword or string that appears anywhere in the help information.

The content of detailed help is a reformatted version of the same content that appears in the Rolodex Tool Users Manual in Appendix A. Details of the manual-to-help reformatting are given in Appendix C.2.

2.10.3.1. Detailed Help Contents

The default tab for the detailed help control panel is Contents, as shown in Figure 277. The Contents pane shows a table of contents for the help information. Each line in the table is a separate topic of information that can be displayed. To select a particular topic for viewing, the user clicks on a single line in the Contents pane and presses the Display button. For example, Figure 278 shows the result of the user selecting the topic "User Interface Overview" in the Contents pane, and then pressing Display. The selected topic in the

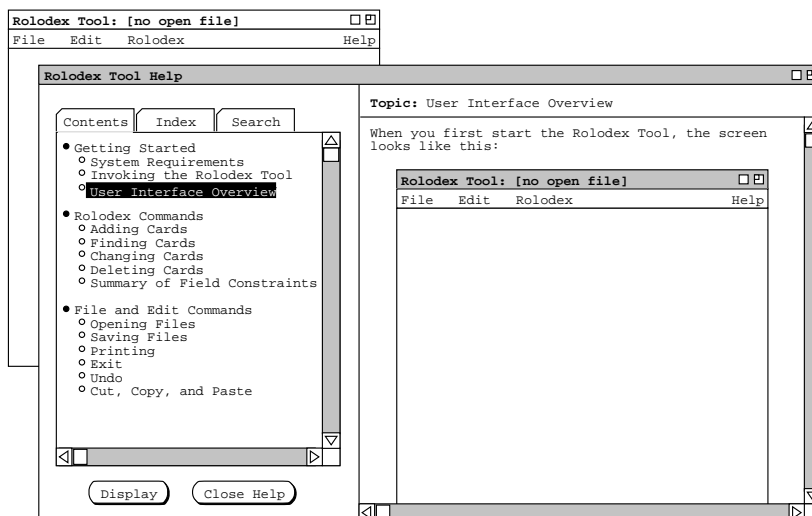


Figure 278: Detailed help for a selected topic.

Contents pane is highlighted and the associated information for that topic is displayed in the information pane on the right. To view another topic, the user selects it in the Contents pane and presses Display again. In response, the system unhighlights the previously selected topic, highlights the newly selected topic, and displays the information for the newly selected topic. Only a single topic can be selected and displayed at one time. Hence, the extent of the information displayed in the information pane is exactly one topic (if the pane is not empty).

2.10.3.2. Detailed Help Index

When the user selects the Index tab in the help control panel, the system updates the help window as shown in Figure 279. The Index pane shows an alphabetic list of index terms that appear in the body of the help information. Each line of the Index pane is a term consisting of a keyword or key phrase that is described in the help information. To select a term for viewing, the user clicks on a single line in the Index pane and presses the Display button. For example, Figure 280 shows the result of the user selecting the term "card field constraints" and then pressing Display. In response, the system displays the information for the topic in which the index term appears, with the textual context of the term highlighted. If necessary, the display is scrolled to the text location where the term appears. The scrolling is performed so that the first line of the paragraph in which the term appears is at the top of the display window.

The information pane for a displayed term contains the information for exactly one topic, where the available topics are those listed under the Contents tab. Even if an index term appears more than once in a topic (or topics), the term is linked to a single textual location. Hence, displaying an index term always leads to the same single location in the same topic.

In the Index pane of the control panel, the user can scroll through the terms in the normal way using the scroll bar on the right of the pane. The user can also scroll through the index terms by pressing one of the twenty-six small buttons labeled with the letters of the alphabet. When the user presses one of these buttons, the system scrolls the term list so that the first term starting with the selected letter appears at the top of the pane. If there is no index term starting with the selected letter, the system scrolls the list so that the term at the top of the display is the last term starting with the earliest preceding letter for which there is at least one term defined. If there is no such term, the display is unchanged.

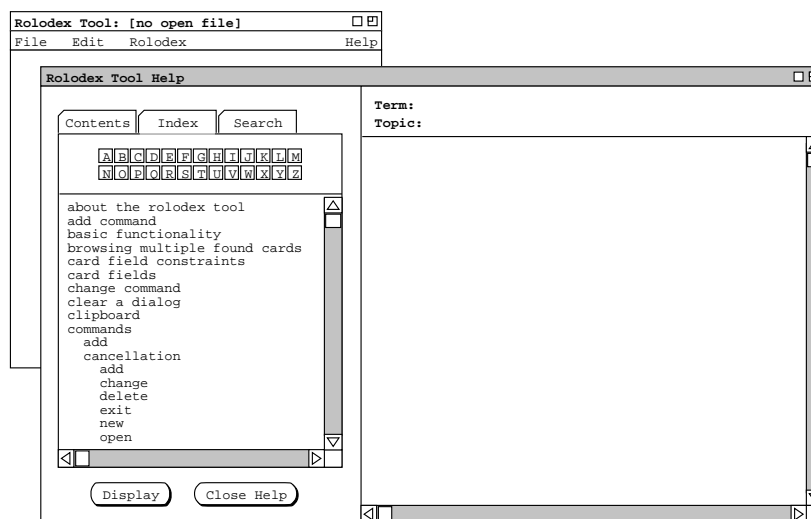


Figure 279: Detailed help index.

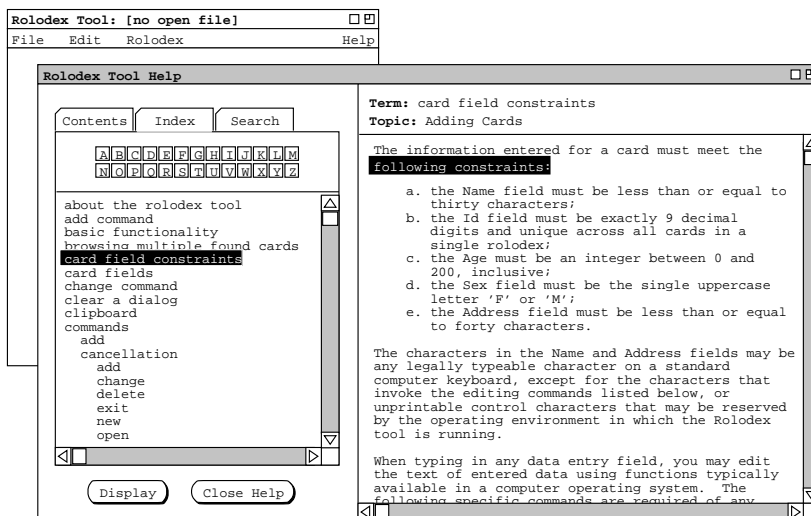


Figure 280: Detailed help for a selected term.

2.10.3.3. Detailed Help Search

When the user selects the Search tab in the help control panel, the system updates the help window as shown in Figure 281.

The user can search for a string that appears anywhere in the the detailed help information. To perform the search, the user types the desired string in the text box below the "Enter search string:" label. For example, Figure 282 shows the result of the user entering the search string "save" and then pressing Display. After a search, the system displays a list of zero or more topics in which the search string is found. In this example, the string "save" is found under the three topics "User Interface Overview", "Adding Cards" and "Saving Files". When a match is found in at least one topic, the system highlights the first topic in the list. In the information pane, the system displays the contents of the highlighted topic, with the first occurrence of the search string

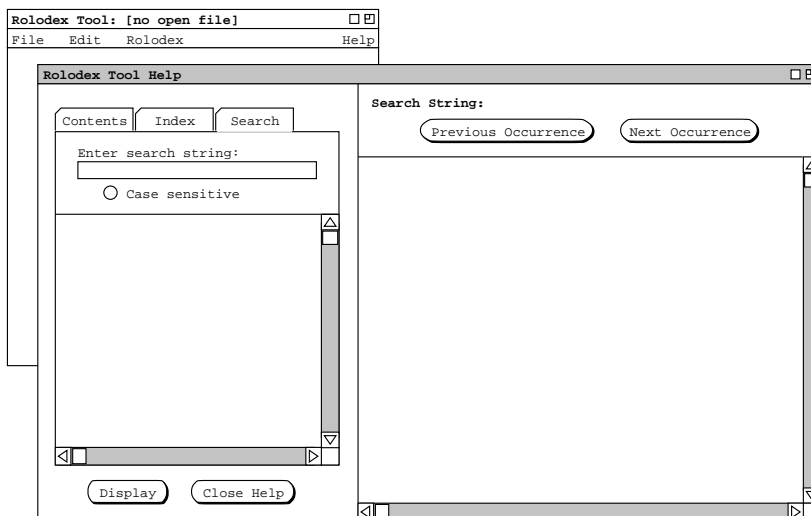


Figure 281: Detailed help search.

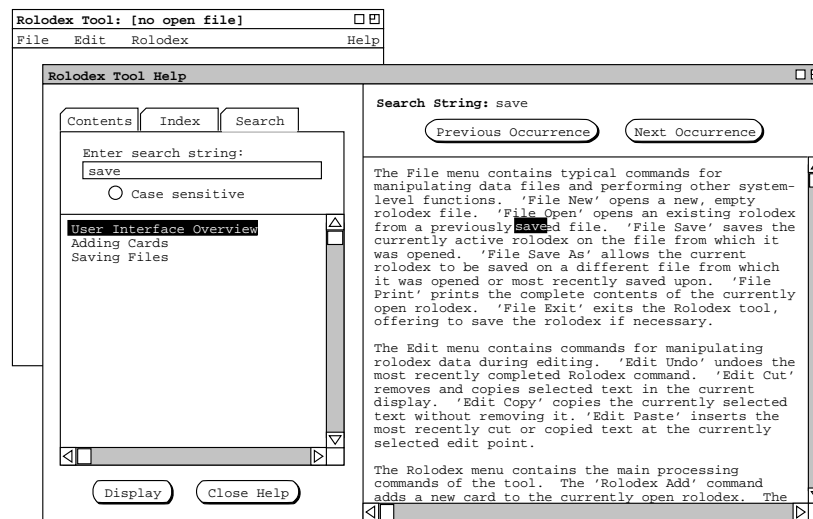


Figure 282: Detailed help for a selected search string.

highlighted. As necessary, the information pane is scrolled to the text location where the search string appears. The scrolling is performed so that the first line of the paragraph in which the string appears is at the top of the display window.

The extent of a search is the text of all help topics, where the available topics are those listed under the `Contents` tab. The order of the search is from beginning to end through each topic, with topics searched in top-to-bottom order of the `Contents` tab list. If a string occurrence is found in two or more topics, the list of topics is ordered in the relative order of the `Contents` tab. For example, the order of the three topics shown in Figure 282 is the relative order of those topics in the `Contents` tab list.

As with the display of index terms, the information pane for a successful search contains the information for one topic, where the available topics are those listed under the `Contents` tab. In contrast to an index term, a search string may occur in more than one location and in more than one topic. If the string does appear in more than one location, the user employs the `Previous Occurrence` and `Next Occurrence` buttons to view each successive occurrence. As each occurrence is displayed, the string is highlighted and the screen is scrolled so that the first line of the paragraph in which the string appears is at the top of the information display window.

The effect of the previous and next buttons moves across topic boundaries. When the user moves to the last occurrence of a search string in one topic, pressing `Next Occurrence` moves to the first occurrence of the string in the next topic, if any. Similarly, when located at the first occurrence in a topic, pressing `Previous Occurrence` moves to the last occurrence in the preceding topic, if any. Pressing `Previous Occurrence` has no effect when the current occurrence is the first in the first topic. Similarly, pressing `Next Occurrence` has no effect at the last occurrence in the last topic.

As the cross-topic moves take place, the system changes the highlighting appropriately under the `Search` tab. That is, the topic highlighted under the `Search` tab pane is always the topic in which the currently highlighted search occurrence appears in the information pane.

The search string may consist of any typeable characters, including spaces and punctuation. If the radio button labeled "Case sensitive" is on, then the search is performed on an exact match basis. That is, an occurrence is considered a match if it is textually identical to the search string, based on all characters in the string. If the `Case sensitive` button is off, then the search is performed on an exact match basis, except alphabetic characters are matched independent of case. Off is the initial value for the `Case sensitive` button.

2.10.3.4. Help Presentation Details

As the user selects different tabs in the control pane, the information pane always changes accordingly. Specifically:

- a. when the `Contents` tab is selected, the information pane contains the "**Topic**" heading and displays information for the most recently selected topic from the `contents` pane;
- b. when the `Index` tab is selected, the information pane contains the "**Term**" heading and displays information for the most recently selected term from the `Index` pane;
- c. when the `Search` tab is selected, the information pane contains the "**Search String**" heading and displays information for the most recently entered search string.

In all cases, the very first time the user selects a tab, the corresponding information pane is empty except for the heading title.

When the user switches between tabs in the help control panel, the state of the control pane is the same as it was when the tab was most recently selected. Specifically, the most recently selected item (if any) is still highlighted, and the scrolled position of the display is as it was when the tab was most recently selected. For example, if the user selects the `Search` tab in the context of Figure 280 and then reselects the `Index` tab, the resulting display again looks exactly the same as Figure 280.

2.11. Data Entry Details

This section covers all data-entry and command usage details that may not have been fully elucidated in the preceding scenarios.

2.11.1. Command Enabling and Disabling

Cover exactly which and when commands are enabled and disabled.

2.11.2. Pressing OK (4sep02)

After a grep scan, it appears that nowhere do we say that scheduling dialogs "remain" on the screen after the user presses "OK". I suppose it's possible that some other word that "remain" was used, but the important point is that it looks like we most recently intended to go with all data-entry dialogs going away after 'OK'. In the back of my mind, I've been thinking that data-entry dialogs stay up after 'OK', because at some point in the past that was the case. Anyway, I like the (seemingly latest) idea of dialogs going away with 'OK', but we need to be absolutely clear that this is the case consistently throughout. Also, it's appropriate either to have subsection here title "Precise Details of OK", or update the extant subsection "Precise Details of Clear and Cancel" to "Precise Details of OK, Clear and Cancel".

2.11.3. Comma-Separated Lists

May have any number of spaces around commas. In multi-line text input fields, may have any number of newline characters around commas.

2.11.4. Typed Input

Typed input is confirmed by one of three user actions

- a. clicking the mouse on any other enabled element in a dialog
- b. typing the tab key, which makes the lexically next enabled element active
- c. typing the return (enter) key, which has the effect of pressing the 'OK' or 'Confirm' button

Once typed input is confirmed by the user, it is validated per the specific data-entry rules in the remainder of this section. Validation may involve further completion of the input, as in completing a date that is input by the user in abbreviated form.

2.11.5. Date Formats

Include min/max values for year, which within reason are 0 and 3000.

When a user enters a time or date in a format that is different than the current setting of the time or date display options, the system automatically converts the time or date to options-specified format when the user confirms the data entry. For example, suppose the current setting of date and time format is

ddmmmyy

and the user enters data as shown in the appointment scheduling dialog in Figure 283. While the dialog remains on the screen, the date formats remain as entered by the user. Once the user presses 'OK', subsequent displays of the information will show the dates and times in whatever the current date and time format options specify. For example, if after confirming the appointment in Figure 283 the user subsequently views that scheduled item via 'View Item', the displayed item appears as shown in Figure 284, where the dates and times are formatted per the current settings of the date and time format options.

Following input confirmation, the parsing rules for date input are the following:

- a. If the input is empty, the completed date is that defined as the default value for the input text field in which the date was entered, or today's date if no default is specified.
- b. If the input is a single integer and that integer is between 1 and the number of days in the current month, the completed date is the input day in the current month and year.
- c. If the input consists of two integers, the integers are interpreted as a day number and month number, in the order defined by the current option setting for 'Date Input Format' described in section Section 2.7.1. If the month number is between 1 and 12 and the day number is between 1 and the number of days in that month, then the completed date is the input month and input day in the current year.
- d. If the input consists of three integers, the integers are interpreted as the day, month, and year, in the order defined by the option setting for 'Date Input Format'. If the month is between 1 and 12, the day between 1 and the number of days in that month, and the year between 0 and 9999, then the completed date is that month, day, and year.
- e. If the input consists of a legal month name and an integer in either order, and the day number is between 1 and the number of days in that month, then the completed date is the input month and input day in the current year.
- f. If the input consists of a legal month name and two integers, the inputs are interpreted as a month, day, and year, with the relative order of the day and year numbers defined by the option setting for 'Date Input Format'.
- g. Any other input is illegal.

A legal month name is three or more of the leading letters of a month, up to the full spelling of the month, case insensitive. Inputs are separated by any number of spaces and optionally one input separator character, as defined by the current option setting for 'Date Input Format'. If the input consists of a month name and one or two integers, the components of the date do not need to be separated by a separator character. If the input consists of a month name and two integers, and the overall order of the three components is not identical to the order defined by the option setting for 'Date Input Format', the order of the two non-blank separators, if they appear, must be as

Enter start and end dates in a couple of different formats, neither of which is the same as current date format option setting. Also, enter a time in 24-hour format, which is also different than current time format option setting.

Figure 283: Dates entered in different format than set in times and dates options.

Show all dates and times per current option settings.

Figure 284: Dates entered shown per current option setting.

defined in the option setting for 'Date Input Format'.

When a date input is illegal, the system responds by highlighting the erroneous text and leaving the input field active with the typing cursor at the beginning of the text. This means that however the user confirmed the input, the original input field remains active, not the field to which the user moved by typing tab or clicking the mouse. If the user confirms an erroneous input by typing return, the 'OK' or 'Confirm' operation is not performed.

In the 'Times and Dates' options tab,

2.11.6. Item Uniqueness Requirements

For appts and meetings, go get it from wherever. I think we need to do a refinement of what we have so far, in that we have to unbundle recurring instances in order to make the check, or at least "conceptually" unbundle them. There's a modelling issue lurking here, viz., whether to model recurring items physically as single or multiple items. If we go with the single-item model, the logic like the last clause of the ScheduleAppt precondition, q.v., needs to be updated to deal properly with recurring items by somehow conceptually "generating" instances of recurring appts.

For events, I think it's gotta be name only, or name plus non-overlapping date. The deal is we need to be able to distinguish between a 3-day event versus 3 separate 1-day events, at least to some extent. It seems if we allow non-unique event names, then there's no way to distinguish multi-day from multiple consecutive 1-day events. I guess the same exact thing goes for an appointment.

A subtlety here is that changing the instance date of a single recurring item to the same date as another instance violates the uniqueness rule. It's nice that the uniqueness rule covers this case in a general way, but I think it is worth mentioning. Way better redundant than sorry.

2.11.7. Scheduling Dialogs and Item-Level Displays

Ed. Note: I think it would be quite good to summarize these requirements in tabular form, with one table for each class of dialog, with one table row for each data-entry field in the dialog.

The following data entry formats apply to any scheduling dialog or item-level display where the indicated field appears:

- The Title and Location fields are free-form, one-line strings. If the text is longer than will fit in the provided data-entry area, then it will automatically scroll as the user types and can be repositioned using the normal text movement arrow keys or such other forms of horizontal scrolling as may be available on a particular implementation platform.
- The value of the Location field can be selected from the pull-down list of available locations, as shown in Figure 285.
- The (Start) Date and End Date are type-in fields, with checking for legal date syntax. The following are legal forms of date entry: ...

2.11.7.1. Meeting Scheduling Dialogs

Ref to two appropriate figures for meeting request and confirmation dialogs.

The precise data-entry rules for the fields that appear in both the scheduling request and confirmation dialogs are as follows:

- o Title can be entered in either the request or confirmation dialog; it must be entered before confirmation.
- o Duration must be entered in the request dialog, where the system-supplied default is 1 hour; it cannot be edited in the confirmation dialog.

Figure 285: List of available locations.

- o `Recurring` is optional in the request dialog, where the system-supplied default is non-recurring; it cannot be edited in the confirmation dialog.
- o `Category` can be entered in either the request or confirmation dialog; the default is no category.
- o `Security` can be entered in either the request or confirmation dialog; the default is public.
- o `Location` can be entered in the request dialog; the default is empty, which is an allowable value. Location may be edited in the confirmation dialog; if edited there, the system validates that the newly entered location is available at the meeting date. If it is not available, the system signals the condition with an error dialog as discussed in Section 2.12. If the location of the confirmed meeting is empty, it is the presumed responsibility of the scheduler to inform attendees of the meeting location via some means external to the Calendar Tool system.
- o `Remind` is optional in the request dialog, where the system-supplied setting is not to remind. It can be edited in the confirmation dialog and is optional there as well.
- o `Attendees` must be entered in the request dialog and cannot be edited in the confirmation dialog.
- o `Details` can be entered in either the request or confirmation dialog; it is optional in both. There is no system-supplied default.
- o `Minutes` can be entered in either the request or confirmation dialog; it is optional in both. There is no system-supplied default.

The precise data-entry rules for fields that appear only in the scheduling request dialog are as follows:

- o `Earliest Possible Date` must be entered; the system-supplied default is today's date.
- o `Latest Possible Date` must be entered; the system-supplied default is two weeks from today's date
- o `Earliest Start Time` must be entered; the system-supplied default is the value of the normal beginning-of-day time, the default for which is 8 AM.
- o `Latest Start Time` must be entered; the system-supplied default is the value of the normal end-of-day time, the default for which is 5 PM, minus the default value for the duration, which is 1 hour; this means the default latest start time is 4 PM.
- o `Earliest End Date` must be entered for a recurring meeting; there is no system-supplied default.
- o `Latest End Date` must be entered for a recurring meeting; there is no system-supplied default.

The precise data-entry rules for fields that appear only in the confirmation dialog are as follows:

- o `Start Date` and `Start Time` can edited only to one of the value pairs that appears in the possibles list.
- o `End Date` can edited only to an earlier date that is not before the start date.
- o `Scheduled By` cannot be edited.

2.11.7.2. Defaults

If the user specifies a recurring event without checking any box in the additional recurring information, the event will be set by default to recur weekly, on the day of the week that is specified for the start date of the event. *[Sketch: some other defaults are made clear by default display of selection box.]*

Also explain what the standard defaults are for the settings of all dialog fields and note that these default settings can be changed in the by editing preferences.

2.11.8. Required Fields

The discussion in the next paragraph needs to be more precise in specifying for which types of item which fields are required. In working on the custom lists section, I've been thinking about adding a table that there with an added third column indicating if the field value is required or optional. If this table is added here, it should probably be ref'd from the custom lists section instead of appearing directly there. The only question is that this complete table (with all three columns) is more that is need either in the custom lists section or here, but seems like a good thing that should appear somewhere. Perhaps it belongs in another place. Anyway, we'll figure it out eventually, if we live that long.

When scheduling or changing an item, the following fields are required to be non-empty and properly formatted: Title, (Start) Date, Start Time, and Duration. If one or more of these fields is missing when the user presses the 'OK' or 'Change' button, the system responds with the error dialog shown in Figure 286.

2.11.9. Selecting Viewing Targets

20aug01 note: need to update this section to include target selection in list displays as well as calendar displays.

The first five commands in the 'View' menu apply to a user-selected target in the calendar display, or to a default target. To select a viewing target, the user clicks on a desired location somewhere in the current display window. The "current display window" is defined precisely as the display window most recently generated by the system or the window most recently made current by the user having clicked on it or selecting it in the 'View Windows' menu. The following are the specific details for how a viewing target is selected in the current display window:

For the 'View Item' command, an item is selected as follows:

- a. In a day, week, or month view, the user selects a scheduled item by clicking anywhere on the time range or title of the item.
- b. An item cannot be selected in a year view, since no times or titles are displayed there.
- c. If no item is explicitly selected by the user, the default is the first item, if any, scheduled on today's date. If there is no such item, then the 'View Item' command has no effect. (The exact system response to commands with no effect is described in Section 2.17.31 on error conditions.)

For the 'View Day' command, a day is selected as follows:

- a. In a table- or list-style week view, the user selects a day by clicking on any of the day names that label the table columns, or on any item in a desired day.
- b. In a list-style week view, a month view, or a year view, the user selects a day by clicking on the numeric date of any day, or on any item in a desired day.
- c. If the current display window is a day view, the 'View Day' command has no effect.
- d. If no day is explicitly selected by the user, the default is today's date.

For the 'View Week' command, a week is selected as follows:

- a. In a day view, the selected week is the one that contains the currently displayed day.
- b. In a month or year view, the user selects a week by clicking on the numeric date of any day within the desired week, or on any item in a day within the desired week.
- c. If the current display window is a table-style week view, the 'View Week Table' command has no effect; if the current display window is a list-style week view, the 'View Week Lists' command has no effect.
- d. If no week is explicitly selected by the user, the default is the week containing today's date.

For the 'View Month' command, a month is selected as follows:

- a. In a day or week view, the selected month is the one that contains the currently displayed day or week.
- b. In a year view, the user selects a month by clicking on the name of any month or on the date of any day within the month.
- c. If the current display window is a month view, the 'View Month' command has no effect.
- d. If no month is explicitly selected by the user, the default is the month containing today's date.

For the 'View Year' command, a year is selected as follows:

- a. In a day, week, or month view, the selected year is the one that contains the currently displayed day, week, or

-- To appear. --

Figure 286: Required fields error dialog.

month.

- b. If the current display window is a year view, the ‘View Year’ command has no effect.
- c. If no year is explicitly selected by the user, the default is the year containing today’s date.

In all of the cases where a name or numeric date is selected, the system confirms the selection by highlighting the name or date in reverse video. For example, Figure 287 shows how the system highlights the date of September 1 when the user clicks on that date number in the monthly display. When a view command is completed by the display of the appropriate window, the highlighted selection is changed back to normal type font.

2.11.10. Deletion of Scheduled Items

2.11.11. Overlapping Items

The system provides a three-way option to control the scheduling of items with overlapping times. The option settings are the following:

- a. disallow overlapping times entirely
- b. allow overlapping times, with warning
- c. allow overlapping times, no warning

When the first of these settings is selected, the system prevents the user from scheduling two or more items with any time overlap. With the second setting, the system allows overlapping items but warns when overlapping items are scheduled. When the third option is selected, overlapping items are allowed to be scheduled without warning.

2.11.12. Category Editing Details

Explain/illustrate the Delete and Change category commands in the edit-categories dialog of Figure 9.

Explain what happens when a user selects the ‘Edit ...’ item in the ‘Add Category’ dialog. Here’s a sketch:

- a. A typical dialog naming/add/delete dialog comes up.
- b. There’s also some form of generic color chooser, e.g., RGB.
- c. Deletion is prohibited if there exists any scheduled item with a category that is colored with the color to be deleted.

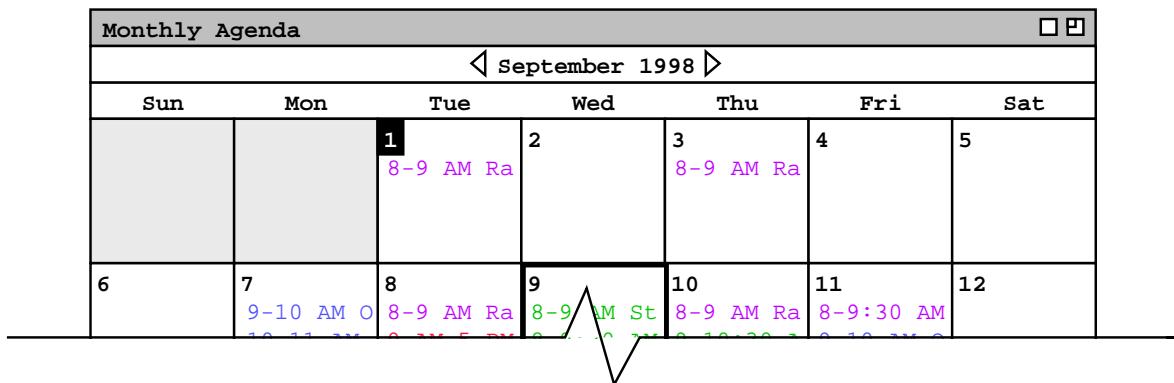


Figure 287: Reverse video selection highlighting.

2.11.13. String and Pattern Matching Rules

Don't forget to explain emacs-style reg expr search for the name lists that appear in the admin dialogs.

2.11.14. Regular Expression Filtering Patterns

2.11.15. Boolean Expression Filtering Patterns

2.11.16. System-Defined Events

While the solstice and time-change events may not strictly be considered "holidays", they are none-the-less included in the 'holiday' category.

2.11.17. Dialog State Details

22aug02 *NOTE*: Given the clarification default options, things may be easier here than the 1st sentence of the ne paragraph suggests.

This is the section where we define all of the horrendously boring details of what initial and historical states the various dialogs assume. Hopefully we can batch bunches of them together under a single heading. Anyway, I think it's pretty clear that such details are necessary, and I think this is the best place to put them. The motivation for this section having come into existence was slogging through the details of the custom list dialog states, and thinking how boring it was to have all of those details piled up there.

2.11.17.1. Custom List Dialog

Dialog state rules:

- a. List Name text box: Initially, contains highlighted text "New List". There after, dialog remembers whatever name that was in the text box when the dialog window was most recently closed; in the case when the last defined list is deleted, the dialog reverts to the initial "New List" state.
- b. Field check boxes: per Table 5
- c. Start and end dates: initially the default, thereafter most-recently entered
- d. Command buttons:
 - o Add enabled when any currently undefined list name appears in the list-name text box, including the system-supplied "New List" and an empty value.
 - i. Change, Delete, and Apply enabled when any currently defined list name appears in list-name text box; Apply is also enabled when the user has done some editing to a not-yet-defined list that is yet to be confirmed with Add.

The following is a way-too-complex version of things, which might be useful at some point as a class/book example, but maybe is best just nuked.

Dialog state is simply that the

- o Initially, 'List Name' field contains "New List", highlighted.
- o When Add is the most recently invoked dialog command, the most recently added list name appears in the List Name box, unhighlighted.
- o When Change is the most recently invoked dialog command, the most recently changed list name appears in the List Name box, unhighlighted.
- o When Apply is the most recently invoked dialog command, the most recently applied list name appears in the List Name box, unhighlighted.
- o When Delete is the most recently invoked dialog command, the most recently added, changed or list name appears in the List Name box, unhighlighted.
- o When Cancel is the most recently invoked dialog command of the first four, the most recently added list name appears in the List Name box, unhighlighted.
- o *Aborted here when noticed that things were getting way too complex.*

2.11.18. Combo-box menu text truncation and other behaviours

A long text string is scrollable in the text box and truncated in the menu, based on how wide the menu otherwise is in the dialog.

This is originally from Section 2.3.3.6. and needs to be generalized here for all combo boxes:

When 'specific date' is initially selected, the system clears the text box and the user types a specific date value into the text field. When 'specific date' is subsequently selected, the system restores and highlights the most recently typed value. If any of the other menu entries is selected, typing is disabled in the text field.

2.11.19. Default Names

Sketch: Form of default names is "New XXX", where XXX is the subject, e.g., "List", "Filter". Initially, we were going to say that "New XXX" form cannot be used to save a filter, but is only displayed as the high-lighted name when the user selects the 'New' item in some name menu. However, instead of this, I think we should say the following. "New XXX" is the initial default name. If the user uses it, so be it, because "New XXX" will not appear again if and until all extant filters are deleted. Sweet.

2.11.20. Name Completion

2.11.21. Meeting Minutes

2.11.22. Merging Scheduler Changes

Need to define precisely what "change" and "most recent" mean in terms of data field editing. It seems like we'll need to model this using time stamps for each meeting field. Here's an interesting scenario that we need to make sure is well understood and covered by our rules:

- a. scheduler schedules
- b. user changes non-critical field F
- c. scheduler changes F, elects not to notify
- d. scheduler changes some critical field, therefore notifies

Question: does scheduler change to F show up in notification? I believe the answer has to be yes, and I think this answer is readily discernable from the rule, though the formal spec will help out, since it will specify things in terms of differences with scheduler and user copies, not in terms of how many executions of Change may have taken place, thereby ignoring the fact that the scheduler changed something once without notification and then subsequently notified. Anyway, what I'm envisioning for the formal model is that it does things in terms of diffs at the time the confirmation takes place, independent of the ups and downs of intermediate changes that may have happened. I.e., we're acting like CVS acts when it computes the 'M' flag. Viz., it does it on diffs at the time, not file date stamps.

2.11.23. Picky Details about Change/Delete Button Enabling

Things work like Emacs, not as hairy-assed as they could be by continuous diffing. Viz., as soon as an edit is made, 'Change' gets enabled and 'Delete' disabled. If the edit is reversed by further editing, the system does not notice that there are effectively no changes, and hence leaves the buttons in the changes-made state. If the user proceeds with a confirmed change at this point, the system goes through the motions, but the change has no effect on the data. This just like adding a char, deleting it, then saving in Emacs -- the same data get saved back on the file.

I believe what this means from the formal perspective is that we don't want the precondition change that says the two args must be different. If we do have this, then we either need to enforce it with diff-style change button enable/disable, or issue an error/warning when the change is confirmed. The latter seems pretty useless, given this seems to be a "no harm done" case if we let the nugatory change just happen.

2.11.24. Precise Details of Clear and Cancel

laug02 -- OK, I'm thinking that the addition of the "cleared" state, as alluded to below to be distinct from the "defaults" state, is probably overkill. I'm ready to go back in the appts-scheduling section to "... to their default states." from the recently updated "... to there cleared states", and deal with the meaning of 'Clear' in the options dialog as a separate issue. Given that there are no option defaults, except in the standard Options file, and having 'Clear' in the options dialog clear to the 'Options file doesn't really make sense, given that the user can load that file if desired.

In addition, the idea that 'Clear' can have the same conceptual meaning in all dialogs is I think wrong, given that it needs to mean "clear to defaults" in a first-time input (aka, create) dialog, versus "clear to state preceding any unsaved changes" in a second-and-beyond-time edit (aka, edit) dialog. And to deal with things consistently in this regard, we can consider the options dialog to be an edit not create dialog.

And to be fully precise, we should here, or perhaps elsewhere, specifically enumerate which dialogs are create and which are edit. You go, boy.

Finally for today, it's just occurred to me (per the rename of this section from "... the Clear Command" to "... Clear and Cancel"), that we need to include here a precise general definition for the meaning of 'Cancel', viz., that unless stated explicitly to the contrary, 'Cancel' always implies clear, and maybe we don't even need the "unless ... to the contrary" part.

OLD and I think invalid stuff starts here.

Need to define "cleared" state for each form of widget, noting (as is done just below) that "cleared" is not equal to "default".

The 'Clear' button behaves uniformly the same in all dialogs that contain it, by clearing all text-entry fields and setting other fields to their unset (not default) states. Note that even when the 'defaults' selection is made for 'Initial Values', and some fields have non-empty defaults, those non-empty values are not set when the clear button is pressed.

But wait -- this may not be the right way for 'Clear' in the options dialog. We need to think about this, and make sure that whatever we say about 'Clear' in the options section is consistent with what we say here.

For simplicity, there's no explicit 'Defaults' button in any data-entry dialog except for the the tabbed 'Options' dialog. This means there's no direct means to restore default settings to a dialog. This can be accomplished explicitly as follows: (1) set 'X.Defaults.Initial Values' to 'as specified in default option settings'; (2) close and re-open the dialog. The 'Edit.Undo' command can also be used to undo the effect of 'Clear', but that may not lead directly to the restoration of default values.

2.11.25. Add User and User Record Dialogs

I'm thinking that validation failure of the user's local computer information should probably not result in a fatal error, since an admin may adding or editing a user record when the user's host computer is down. So, the cal tool admin system will perform the validation, report the results, but not prevent the

We might have the cal tool be out of the business of validating local computer user information altogether. What we can say is that it's up to the admin to use some external means to perform validation. However, this seems harsh.

Some fodder originally from admin section:

If communication cannot be established with the selected host, the first line is changed to

Communication cannot be established with the selected host computer.

If communication can be established, but the host is not running a Calendar Tool server, the message is

The selected host computer is not currently running a Calendar Tool server.

If the selected calendar file does not exist, is not readable, or not a valid Calendar Tool file, the the second line of the dialog is changed, respectively, to one of the following:

The selected Calendar Tool file <file name> is not readable.
 The selected Calendar Tool file <file name> does not exist.
 The selected Calendar Tool file <file name> is not valid.

2.11.26. Non-Modal, Calendar-Specific Dialogs

OK, here's a semi-epiphany -- get rid of the Save button altogether, but retain Save-As. The rationale is to have all non-scheduled-item data handled uniformly. Viz., we have the following calendar-specific non-scheduled-item data:

- a. categories
- b. custom lists
- c. custom filters
- d. options
- e. host(s)

(Note that initialization is not considered cal-specific data, since we can operationally in an init file test for the name of the calendar and thereby do things on a calendar-specific basis.) In each and every one of the editing dialogs for these data types we'll have the following buttons, that have the same meaning in each (except we may not have apply in the category-editing dialog, given that categories affect scheduled item content, not just system behavior (may need to think about this point a bit more wrt options settings (i.e., do option settings affect scheduled item data content ever?))):

- a. Apply
- b. Save As ...
- c. Load ...

The uniform behavior is as follows. Apply means apply any user-entered (including loaded) changes to the current calendar (*never* all calendars). Save-As means save the data to a file (never as part of a calendar file). Load means load from a type-specific data file or from part of a calendar file. Changes to any and all calendar-specific data are saved to the specific calendar file to which they belong using the File->Save command, thereby obviating the need for a separate Save command in the calendar-specific data editing dialogs. As a brain-assisting convenience, the offer-to-save dialog may mention the type of the data that are unsaved, as in

There are unsaved {category, custom list, custom filter, option} changes ...

Extracted from gui-details section:

If more than one calendar is open, only the current calendar is affected by 'Apply'. If no calendar is open, then the 'Apply' button is disabled.

Extracted from gui-details section:

Make sure the effect of window closing is 100% defined. Hopefully close == Cancel works most or all of the time.

Unless otherwise noted in the scenarios, all dialogs are non-modal. *It would be prudent here to list exactly the (hopefully few) cases where modal behavior is enforced.* We should also discuss what happens when would-be modal dialogs are closed, viz., and not necessarily limited to, reminders and meeting notifications. Another interesting case is when the same user executes two or more 'Schedule Meeting' operations while an original such operation is in the synchro stage. It seems clear that this should mean that the user is put in the queue behind himself (!).

I'm thinking that --maybe-- modal dialogs should never appear in the windows list, hence we don't need to mention window modality when we discuss the 'View Windows Close' command, as we don't currently do. We do need to address the issue that any "pending" activities are canceled with a window is closed.

Somewhere we need to address what it means for the system to interrupt a modal dialog, particularly with the proceed-to-schedule notification. In this case, I'm pretty sure that it should mean the modal dialog is canceled, which means we have to be careful to put a cancel button in all modal dialogs.

Extracted from window-viewing section:

I think the following paragraph is way too confusing and potentially error prone. Better is to have all calendar-specific data-entry dialogs apply to the calendar that is current when the dialog is originally opened. Now, this may well mean that when more than one calendar is open, all calendar-specific windows are unconditionally (i.e., not optionally not) labeled with the name of the calendar to which they apply. Also, we may want to allow more than one data-entry dialog for each type of command, for each calendar. The possibilities in this regard are the following:

- a. *one dialog total for each (calendar-specific) data-entry command, and the dialog applies to one of these:*
 - i. *the current calendar*
 - ii. *the calendar that was current when the dialog command was originally executed*
- b. *one dialog for each different calendar for each (calendar-specific) data-entry command, the dialog therefore applying to its particular calendar*
- c. *multiple dialogs for each and every execution of each (calendar-specific) data-entry command, the dialog therefore applying to the calendar that is current when its command is executed*

Here's an evaluation of each of these possibilities:

- a. *one dialog total for each (calendar-specific) data-entry command:*

- *pros: avoids window proliferatino, of which there is potential for much already*
- *cons: see individual points that follow immediately*

(Li

- b. *the current calendar: requires that dialog contents change when cur calendar changes, including the notion of a "pending" operation for a particular calendar so that dialog contents can be intelligently updated between calendars*
- c. *the calendar that was current when the dialog command was originally executed: makes it potentially confusing as to which calendar dialog applies to and potentially inconvenient to change scheduling focus between calendars*

0. *one dialog for each different calendar*

- *pros: avoids cons of one-total approach*
- *cons: window proliferation and still some potential confusion if user doesn't pay (close) attention to changing window banners*

0. *multiple dialogs for each and every execution*

- *pros: consistent with multi-window mode and an overall "the more windows the merrier" philosophy*
- *cons: even more potential window proliferation and multi-window confusion*

So, the choice is b. This leads to a bit of a rethink of the current window business and a rewrite of the options section introduction regarding calendar-specificity.

OLDER STUFF:

All data-entry dialog windows are non-modal, calendar-specific, and unique. "Non-modal" means ... "Calendar-specific" means that when the operation of a data-entry dialog is confirmed with 'OK', the operation applies to the current calendar. "Unique" means that exactly zero or one data-entry dialog of each for data-entry command. *[Enum all data-entry commands explicitly.]* For example, suppose a user has three calendars open, named "A" and "B", and "C". With calendar A current, the user executes the 'Schedule Appointment' command, enters some values in the appointment scheduling dialog, but does not press 'OK'. The user then makes calendar B active, executes a view command for it, and then re-executes the 'Schedule Appointment' command. At this point, the system does not display a new appointment scheduling dialog, but moves the previously-displayed dialog to the front of all windows. That dialog contains the values most-recently entered by the user. If the user were to press 'OK' in the appointment scheduling dialog at this point, the appointment would be scheduled on calendar B, even

though the values were entered in the dialog when calendar A was active. In this scenario, assume that the user proceeds to make calendar C current and executes one or more viewing commands for calendar C. The user then makes the appointment scheduling dialog current by selecting its window name in the 'View Windows' menu. Since there is at most one appointment scheduling dialog active at any time, selecting its name in the 'View Windows' menu has exactly the same effect as re-executing the 'Schedule Appointments' command, viz., the appointments scheduling dialog is made active. The user now proceeds to enter additional information in the appointments scheduling dialog and presses 'OK'. If the entered information is error free, the system schedules the appointment on calendar C.

NOTE that the windowing mode is calendar-specific, meaning that each open calendar must have at least one open view window.

More details on window closing. Since scheduling dialogs are non-modal, the user could do this: Open a calendar, execute a schedule command, close all calendar view windows, then confirm the scheduling command. But hey, this ain't a problem, given that confirming (with 'OK') does not close the scheduling, so it'll just stay open as the only window. It looks like we're OK here, given that all of the editing dialogs stay up until the user presses cancel (which closes) or closes explicitly. And it's at that point (i.e., close/cancel) that the following dialog comes up (before the close takes place):

**This is the last open window for the calendar "name".
Closing this window will close the calendar.**

As a convenience, we may want to merge the offer-to-save stuff into this dialog if it applies, rather than having the offer-to-save dialog come up next. Or, we may just let the offer-to-save come up next. Working out a scenario will clarify this issue. Here's a quick crack at same:

- close-calendar-on-last-window-close dialog appears
- user presses 'Yes' (or 'OK') to proceed with calendar close
- if there are no unsaved changes, system closes window and calendar, with appropriate updates to windows and calendars lists
- if there are unsaved changes, system leaves original window up but removes close-calendar-on-last-window-close dialog, then displaying the offer-to save dialog
- if user presses 'Cancel' in offer-to-save dialog (or subsequent file-saving dialog), original window and calendar remain open stay up
- if user proceeds to completion of offer-to-save, system closes original window and calendar

In this window-viewing section, refer to file section for further details of offer-to-save behavior.

NOTE: Fodder from options section follows.

NOTE: Rewrite all of the introduction to get rid of the long-winded calendar-specific biznis, moving it instead to the end, in the same that it's presented at or near the end of the categories, custom lists, and custom filter sections. Ditto for the revised Save-As and Load semantics. The deal there is that when the user switches calendars, the current options remains unchanged, but it does not belong to the current calendar.

... If more than one calendar is open, only the current calendar is affected by 'Apply'. If no calendar is open, then the 'Apply' button is disabled.

NOTE: The following paragraph needs to be updated per the most recent banner policy, which is controlled by options.

Since options are calendar-specific, the banner of the options dialog indicates which options settings are currently active, and therefore displayed in the options dialog. In Figure XXX, for example, the options are those for the CommunityCalendar file. When the user changes the current calendar while the options dialog remains on the screen, the banner and contents of the dialog are changed to reflect the options for the newly current calendar. When no calendar file is open, the options dialog banner is "Options for new calendars", meaning that the option dialog is operating on the settings in the standard 'Options' file.

Each and every calendar file has a separate copy of option settings. This allows users to define different options for different calendars, as desired. When more than one calendar is open, the active option settings are those of the current calendar. If there is no open calendar, then the active options are those defined in the standard options file named "Options". This file is stored in the user's Calendar Tool directory, as described in Section 2.15. The contents of the Options file are used as the default for all new calendars. Whenever a new calendar is created, the contents of the Options file are copied into the new calendar as the initial settings for all options.

2.11.27. Units of Measure

Put this here, or probably someplace else, but the bottom line is the "pixel" must be defined as "the atomic unit of screen-size measure for a particular display". This is the measure used by the underlying operating environment to describe screen measurements, as in the screen size is 1024 x 768 "pixels".

2.11.28. Options Data

Hopefully in most cases, details have been covered in other dialogs to which options apply. I believe the following are specific to options:

- a. In scheduling meetings: the left and right screen positions are in pixels and must be between 0 and the maximum dimensions of the screen.

2.12. Error Conditions

Review all of the errors referred to in the data entry details section. Dole them out to appropriate subsections herein.

Just recently, I've thought about the following distinction between an error condition versus non-error, particularly in the context of scheduling (and which is also applicable to searching probably):

"It's an error if it's a precondition violation."

Ah, but now that I've written it down, it's exactly right and what we've always been saying. This means that it's not an error for the scheduling op not to find any available times, nor similarly for a search to find applicable search targets. This in turn is (further) justification for searching and scheduling not to be formally spec'd with non-success as preconditions (which I think we already pretty much know). From the requirements perspective, specifically with regard to the guideline that we leave error conditions to later, it's consistent with this guideline to include the "No times found" message" in the scheduling section, but leave other scheduling errors, such as "You're unauthorized to send to these users: ..." to the error section of the requirements.

2.12.1. Commands with No Effect

OK, here's a no-effect issue that must be dealt with. Viz., when the user confirms a change that is really no change at all, the system should not send notifications. Changes that this applies are scheduler meeting changes and administrator database changes. OK -- I just fixed this by adding the "re-edit" conditions to the introduction of the change-delete section.

28jun01 Note: I'm thinking we should not go the Rolodex route with the "no effect" business, but rather do things thusly:

- a. *say that nothing was greyed out in the initial menu expansions for clarity, and since it's a generic display independent of any particular system state.*
- b. *have an explicit discussion here, or in data-entry-details, or even in gui-details, that describes precisely the conditions under which main menu items are enabled and disabled*
- c. *have a similar explicit (summary) discussion of the dialog state conditions in terms of when GUI elements are enabled and disabled.*

The paragraphs that follow are pre the preceding remark and therefore mostly obsolete if we go this way.

The following notes were removed from the structure-viewing section. Deal with them:

- *[Ed. Note: determine if we should discuss here that the system beeps and/or discuss in general elsewhere how*

the system signals an uneffectual command, including allowing an option for such signaling. See just below for further discussion of this.]

- *[Ed. Note: Think about adding a dialog, probably an option-controlled dialog, to each of the above "no effect" cases that says something to the effect of: "No X selected, please select one.", where "X" is one of item, day, week, month, or year. The emacs name of the controlling option would be something like display-dialog-on-no-effect-command.]*

NOTE: "Dealing" with them at this point I'm pretty sure means using disabled GUI elements, rather than some other way of specifying no-effect commands. However, we do need to nail this down completely, and a minimum discuss it clearly, if not here in the gui section.

24jul02 Update: Let's forget error-control options, and just have the system make up it's farging mind about how to do things properly for error handling. This means disabling the widgets for commands that would have no effect or are disabled because their preconds aren't met.

2.12.2. Searching for Unknown User

2.12.3. Scheduling Errors

Potpourri:

- the start date and time must be after the time at which the scheduler operation is confirmed
- the time to send a reminder must be after the time at which the schedule operation is confirmed
- if the 'disallow' option is set for overlaps, then add this to the error conditions, i.e., scheduling an overlapping item is an error.

2.12.3.1. Unknown Meeting Attendees

If the list of attendees contains unregistered attendees not so marked, the system will output the error dialog shown in Figure 288.

2.12.3.2. Meeting Scheduler Validation

As noted in the preceding scenarios, only a group leader may schedule a meeting for a group. When one or more group names appear in the attendees list, the scheduler must be a leader of all groups in order for the scheduling operation to proceed. The determination of group leader status is based on the definitions in the group database (Section 2.6.3).

Scheduler validation is performed when the 'List Times ...' command is executed. If the scheduler is not a leader of all groups in the attendees list, the system displays an error message, the precise format of which is described in Section 2.12.3.

```
The following attendees are not registered users:
```

```
...
```

```
Check 'OK' to mark each of these names as 'Not Registered'.
```

```
OK    Check All  Cancel
```

Figure 288: Unregistered attendees error dialog.

2.12.3.3. Non-Unique Items

The system disallows multiple scheduled items with exactly the same values for date, start time, duration, and title. Error happens if user tries to violate this. Conditions where it can arise are:

- a. schedule appt op; in this case, penciled-in meetings are considered "in"
- b. schedule meeting op for the scheduler (handled during notification for attendees), including the case where the scheduler has a private item that causes violation to occur
- c. changing an appt or meeting

Possibly useful verbiage axed from meeting-scheduling section:

When a single user schedules an item on the user's own calendar, the system displays an error message if the uniqueness condition is violated. When a scheduler schedules a meeting for other user's, the system does not check the condition for all attendees. Rather, the condition is checked at the time the meeting is accepted by the attending users.

2.12.3.4. Changing a Meeting to an Unavailable Location

2.12.4. Errors from an External WWW Browser

2.12.5. Administrative Errors

Review all the errors referred to in the admin section.

To cover:

- invalid user IDs in group member dialog of Figure 212,
- conflicting location booking in book location dialog of Figures 222 and

2.12.5.1. Non-Admin Attempting to Login as Admin

The system should be able to detect this through the login ID.

2.12.6. Failure of Central Host Computer

OK, we might just want to skip all the complicated auto notification stuff and have the central host server run pretty much like an apache server, in that detection of it being down is strictly from the server as opposed to server side. I don't really think it can work completely this way, since the difference here compared to apache is that the server is maintaining data about remote servers in a way that apache is not, i.e., the user ed. We'll have to think about this to see if any apache-inspired simplification is possible or appropriate.

Failure is defined as the central host computer stopping operation or becoming disconnected from the network. In the case of stopped operation, the backup central host immediately becomes the active central host, if it is possible to perform such immediate action in the underlying operating environment. The switch from backup to active host entails the following automatic actions:

- a. on the backup host and in the Calendar Tools of all registered users, the ID of the central host is changed to the ID of the backup host
- b. all users are notified of the host change

If there is no backup host or it is not operational, the following actions are performed automatically, if possible:

- a. in the Calendar Tools of all registered users, the status of the central host computer is set to 'down'
- b. all users are notified of the failure

The format of the central host failure notification is shown in Figure 289. If the failure is handled by immediate transfer of the backup host to active host, then the failure notification is not sent, but rather a central host change notification is sent, as shown in Figure 229.

The Calendar Tool central host computer is down or unavailable.

You can continue to use the Calendar Tool on your local computer, but you will not have access other to user calendars or any of the administrative commands.

Figure 289: Central host failure notification.

If the automatic switch from active to backup host is not possible, then the administrator must perform the reassignment of active host manually on the backup host, as described in Section .

In the case of network failure or when it is not possible to send failure notification automatically from the central host, individual user computers must detect central host failure when executing any command that requires access to the central host. These commands are:

- File Save, Save As, or Save All
- the 'List Times' command during meeting scheduling
- View Other User or Group
- any Admin command

The first time any of these commands is executed and the user has not yet been notified of central host failure, the user is then notified and the status of the central host is set to 'down'. The definition of "not yet notified" is that the status of the central host is detected to be down, but the status value as known to the Calendar Tool is still 'up'.

If no backup host computer is in operation, it is the administrator's responsibility to maintain backup copies of Calendar Tool administrative data files that can be used in the event that the active central host stops operating or becomes disconnected from the network.

2.12.6.1. Loss or Corruption of Central Host Data

Sketch: Individual user hosts can run the Calendar Tool just fine, but they won't have access to any of the databases or other user's calendars. There's not data consistency problem with any extant scheduled meetings that refer to other users, groups, or locations, since all such references are purely symbolic. (Explain what "symbolic reference" means and add any additional general explanation that's necessary.)

2.12.7. Failure of Local User Computer or Invalidation of User Information

Among other things, we should probably think about error logging when communication local computers fails. Or perhaps the whole idea of the central host as the "up to" machine obviates the need to worry about local machine failure when running the non-admin cal tool. Worrying about local machine failure when running cal tool admin may in fact have to be considered.

2.12.8. Missing or Corrupted Options File

If there no valid readable Options file, the Calendar Tool creates one when it starts up. If in cannot, because the user's cal tool dir is non-existent or unwriteable, it issues an error.

2.12.9. File Errors

2.12.10. Zero-Instance Recurring Items

Deal with this, including when this happens via change.

2.12.11. Exceeding Item Limits

When a user confirms a scheduling op such that her item limit is exceeded, the system issues the error message in Figure 290. The deal is that the user will keep seeing this (presumably annoying) message until she (a) asks the

You're at your item limit.
Delete one or more items in order to schedule more.

Figure 290: Item limit exceeded.

admin to up her item limit, (b) deletes some items from the calendar that is associated with the central host on which the item limit is set, which limit the user can see by viewing her user record on that host.

And here's some rationale fodder: Thought about having the system do an auto delete of the oldest scheduled item in the calendar when the item limit is

When a scheduler confirms a meeting and adding that meeting to the user's central host calendar causes that calendar to exceed its limit, the scheduler is notified that the meeting did not get scheduled for that user. Since this is presumably a rare occurrence, we may not need to have the option to delete the oldest other item. So what presumably happens is that the scheduler contacts an admin and/or the user(s) whose item limit(s) was(were) exceeded to deal with the situation.

2.13. Graphical User Interface Details

2.13.1. Screen Coordinates

2.13.2. Enabled and Disabled GUI States

All menu items in the initial overviews of Figures 2 and 4 are shown enabled for clarity in the figure. The precise enabled/disabled state of menu items is summarized in Table 13.

2.13.3. Remembering Previously-Entered Values

Where the user has no explicit control over this, e.g., in the admin dialogs, the dialogs hold the information most recently entered by the use. This history extends during the execution of the Calendar Tool, but not between executions.

2.13.4. 28jun01 Note

Maybe OK to relax dialog history requirements if a particular GUI environment does things consistently differently. Give some examples, of e.g. Java versus how things are defined here.

2.13.5. View Command Display Details

2.13.5.1. Day View Scrolling

The three components of a day view are separately scrolled. *[Show an example of this, e.g., with lots of events, tasks, and some out-of-normal-range items. Also reiterate that hiding non-normal times never hides scheduled items, with scrolling invoked as necessary if there are some odd-ball things scheduled at non-normal hours.*

Menu Item	When Enabled and Disabled
File:	

Table 13: Summary of enabled/disabled states for all menu items..

2.13.5.2. Window Sizes

Need to deal with the (new) fact that there are two font sizes -- item and other. Also need to explain that the rules for scheduled-item display sizes are based on some kind of "whatever fits" rule. Finally, as noted below, need to say that font change does not fully resize a window, including even the funky case where labels don't fit any more (maybe). The tricky deal is with things like month and year views, where the font-change rules are not necessarily crystal clear. Unfortunately, we probably have to go through most of the different display types and specify reasonable font-change rules for all.

The default initial size of the view displays is based on the size of the currently selected display font. In specifying display size, the following standard units of typographic measure are used:

Unit	Meaning
------	---------

p	one printer's point = 1/72 inch
S	user-selected font size in points; default font is 12-point courier
V	vertical height of text line, including vertical spacing = 1.2 x S
em	width of a representative "m" character = 6 x S

Based on these units, Table 14 defines the default size for each of the view-level display windows.

View Level	Dimensions
------------	------------

item	60em wide by 34V high
day	time column is W+4em wide, where W is the width in characters of widest displayed time; title column is 64em wide; total height is TIem x Rem, where TI is the current setting of the <i>time increment</i> option and R is the total number of displayed rows based on the current settings of the <i>normal time range</i> and <i>show/hide 24 hours</i> options
week	each day column is 10em wide; total width is N columns, where N is the number of days based on the current setting of the <i>normal day range</i> option; total height is 24V
month	each day cell is 10em square; total width is 7 cells; total height in cells is based on the number of days in a month
year	72em wide; total height is 14V + W1V + W1V + W1V + W1V, where each W _i is number of weeks in the month with the most weeks for each of the four quarters of the year.

In all displays, border lines are 1p and the spelled-out date title at the top is 1V high.

The user may change the default font family and size in the preference setting dialog.

In general, all views scale evenly when the display window is resized, but font size does not change. There is no maximum window size, other than that dictated by the physical limits of a particular display screen. The minimum window size is generally based on the size of the currently selected text font. Specific details of resizing windows follow.

Display features for the day view:

- The time increment setting does not affect the allowable start and end times for scheduling any item, it only affects the appearance of the display.
- Horizontal lines on the hour are shown in black across the full width of the day display; lines at increments smaller than an hour are shown in light grey across the width of the title column, but not the time label column. See Figure ??.
- When dashed lines are hidden and the increment is smaller than one hour, item titles are positioned vertically so that their tops start at the same height as the increment label nearest to but not after the start time of the item.
- Dashed lines and down-arrows go off when vertical height is too small -- define "too small" here.
- Resizing behaves as follows:

- o No row in the display will ever shrink smaller than the current setting of the increment size.
 - o Hence, if a day display window is shrunk such that there is not enough room to show all rows of a all days, the display will start clipping rows from the bottom of the display, adding a scrollbar on the right of the full display.
 - o If a day display window is grown such that there is more than enough room to show all rows of all days, then each row is increased equally in size, and the scrollbar is removed
 - o When a particular hour has more scheduled items than the current setting of the increment height, and there is not enough vertical height in that hour row per the preceding resizing rules, then a mini vertical scrollbar is added to that hour of the display. Mini scrollbars are only shown when the user clicks within an affected hour. Also, mini scrollbars are only shown for an hour display or greater, not for increments smaller than an hour. The mini scrollbars work, independently from the larger vertical scrollbar for the full day display.
- The font size is not automatically shrunk when window is shrunk.

The precise default format of an item listing is as follows:

- a. The item begins with the starting and ending times, separated by a dash.
- b. If the starting and ending times are both in the AM or PM, then the numeric time values are separated by a dash with no surrounding spaces, and the pair is followed by one space and then "AM" or "PM" designation.
- c. If the starting time is in the

Talk about how font size changes affect resizing, keeping it minimal. In particular, the height and width of the scheduled-item guts of view windows should not change, but the internal height of things like the title date need to change. Include talk about how text is filled horizontally as much as possible in the guts of displays.

2.13.5.3. Details of Horizontal Overlap Display

The example to address here is when in the context of Figure 18 the user adds appointments at the following times: 3-4PM, 2:45-3:15PM, 3:15-3:45PM.

Here's the rule that I think works, which may be easier to explain if we think some more about it. For any day, create sets of items with any overlap whatsoever. Sort those items by the normal criteria, vis a vis earliest start time, shortest duration, alphabetically earliest title. Within this list, move all items over as far as possible to the left, as long as the no two items in the same column overlap and the sorting criteria continue to be met. It appears what we're doing is maximally filling each column, i.e., not leaving any space blank in a column if a none overlapping item will fit there. If two or more items will fit in a give space, then the one chosen is the one earliest in the sorting order, with the unchosen one relegated somewhere to the right, in the (recursively) correct position. "Recursively correct" means in the leftmost column where it will fit such that all criteria are met.

2.13.6. Window Positioning for Next/Previous Commands

At the day through year levels, the window position stays fixed at the top. At the item level, the window position stays fixed at the bottom. OK, the spelling out in the last sentence made me realize that its pretty bogus and non-orthogonal. So, we'll change the item-level display to have the time and data at the top.

2.13.7. Apple Macintosh Menu Conventions

If Macintosh implementors want to adhere to the non-fbating menubar convention, then the current filename display will be missing. This is acceptable, since the user can determine current file using the 'View Calendars' command.

2.14. Multi-User Operating Environment

Figure 291 shows a typical configuration for a Calendar Tool multi-user operating environment.

Figure 291: Typical Multi-User Environment.

So, if it's still not 100% clear below, here's how things work:

- a. The central host *writes to NO local files*; rather, it communicates via messaging to the local client, which in turn consults the settings to figure out what files to write to
- b. Accordingly, there need be no, and therefore is no specific named file that is known to the central host as the community calendar.
- c. Rather, there is the following convention, which will be acted upon by the user via (the newly updated) 'Central Host ...' dialog: -- OK, go see the latest version of the dialog to see what the convention is.

13aug02 Update: Contrary to the 19jun02 Update just below, I'm feeling much warmer to this section at this point. The deal is that I don't think this stuff is spelled out explicitly and clearly anywhere else. And I don't really think it belongs in the admin section, since not all of the material in this section is directly related to admin commands, one of the most important topics being that the host-associated calendar is copied to the central host when it's saved. I suppose we could cover everything in the admin, file, and possibly other sections, however I think it's a good idea to have it all here, in order for all of the multi-user aspects of the system to be laid out clearly.

19jun02 Update: I've always been a tad (to more than a tad) uneasy about the existence of this section. It may be that the new Section on initial system config may be able to cover all of the topics we had in mind for this section. That failing, we may well still want to migrate everything we had in mind for this section into the admin section.

The functional requirements presented in the preceding sections *Here's where the art of not over specifying comes in.*

Draw a nice little picture that shows where the data are actually stored. Viz., in the central repository has a copy of all user calendars, with non-private items and not-yet-accepted (i.e., penciled-in) meetings. Each user has her own copy of the calendar containing all items, including the private ones, but no not-yet-accepted meetings. Hence, when the view commands operate, they are showing information from potentially both the local copy and the central repository. Viz., all local items plus penciled items for each user looking at her own stuff. When viewing another user, everything comes from the central repository. This all means that when the user is disconnected from a central host, penciled-in meetings do not appear. Further, each host has its own, quite possibly different set of not-yet-accepted meetings for a particular user.

Mention the requirement for distributed clock synchronization for comparing the times of scheduler and attendee meeting field changes.

In terms of physical computer configuration, there is no requirement for user's local computers to be physically different machines. In fact, an entire Calendar Tool user community can all be located on the same physical computer, including all users as well as the central host. In such a case, the terms "local" user computer and "remote" central host are conceptual, not physical. The important physical distinction that always exists is that of user calendars and the central host repository. That is, each user always has a copy of all her or his calendar files in a Calendar Tool directory owned by the user. Also, the Calendar Tool data files are always stored in a directory owned by the Calendar Tool administrators. The specific definition of file "ownership" is platform-specific, but abstractly it means the following: the owner(s) of a file (including a directory) have the right to control all access to the file, where access consists of its name, visibility, readability, writeability, and existence.

2.14.1. Invoking the Calendar Tool from the Operating Environment

There are two separate executable programs named `CalendarTool` and `CalendarToolAdmin`. When a user invokes `CalendarTool`, the system determines if the invoking user is a registered user of the Calendar Tool system. This check is performed by searching for a user record in the Calendar Tool database containing the invoking user's computer ID on the user's host computer. If the system confirms that the user is registered, then the system displays the regular user interface, otherwise it displays the unregistered user interface.

Explain that the system establishes the user's calendar tool ID by matching it against computer user ID. This means, in effect, that in order for a cal tool user to participate in the network functionality of the system, the user must have an account in an operating environment with a user name.

As explained in ..., admin users must login with a password when invoking CalendarToolAdmin.

2.14.2. The Central Calendar Repository

If the item limit value is set such that it is smaller than the current number of items on the user's calendar in the central repository, the system deletes enough items to reach the limit value. Item deletion is performed in the following order, until the limit is reached:

- a. items before today's date are deleted, in order from the earliest to latest scheduled time and date
- b. if further deletion is required, items are deleted in order from that latest scheduled time and

... excised from the admin section: That is, the act itself of invoking Calendar Tool Administration defines the invoked-upon computer to be a central host. The administrator

Any changes that the administrator makes are stored in the Calendar Tool database files for the invoked-upon computer only. There is no means to remotely configure one central host from a different host. If there are to be multiple central hosts in a given operational setting, then each must be configured independently. To avoid highly repetitious data entry tasks, administrators may copy Calendar Tool database files from one host to another.

2.14.3. Messaging

Points:

- Central system stores all notifications (including host-change messages) until user computers are ready to receive them.
- As soon as user machines are ready to receive them, central machine sends them.
- Implementors can figure how to do this in any combination of distributed implementations.

2.14.4. Coordination

The central system coordinates the synchronization of meeting schedulers.

OK, we should skip the "hot backup" stuff in the next paragraphs by simply saying that whenever a connected user does file save, the system sends public calendar data to the central host. When this happens, we need to make sure that the not-yet-accepted items on the host are not overwritten. So, we need some kind of rule that says that not-yet-accepted items are separate from the rest of the calendar, at least (and hopefully solely) for the purposes of updating local user changes with the central host. The deal is that whether or not not-yet-accepted meetings are physically separate on the central host, they are conceptually separate in that they stay as is on the central host until they are explicitly accepted or declined by the user. At the risk of over specification (but see the preceding sentences and the last sentence of this paragraph), we should be clear about precisely what portions of a user calendar are kept on the central host. It seems that categories need to be there, given the rules for viewing other users's calendars. However, it seems pretty clear that custom lists, custom filters, and options do not belong on the central host. A detailed walk-through of all commands is in order to be absolutely sure about this. (Actually, we're not at over-specification risk here, since it is appropriate for data security and privacy reasons for the user to be aware of exactly what data are copied to the central host.)

OK, we need to be very clear about exactly what gets copied and when. Here's some additional information:

- a. On file open or save (including offer-to-save): system attempts connection to host designated for the calendar and if connection is established public information gets copied to designated host (if any), without changing any not-yet-accepted items on the host.
- b. On file close: system disconnects designated host (if any).
- c. On connect: system attempts connection to selected host and if connection is established public information gets copied to designated file (must be one), without changing any not-yet-accepted items on the host.

- d. On disconnect: closes designated file.
- e. On receipt of meeting notification from a particular host: system attempts (presumably establishes, unless in rare case host goes immediately down) connection to originating host and if connection is established system does open of designated calendar (including, therefore, the copy-to-host specified above for open)

What this means, among probably other things, is that a host cannot be connected without its designated calendar being open. Hope this is cool.

NOT (pre preceding simplification): The central host copy of a user's calendar is kept in a hot backup state whenever the user is connected. This means that effects all confirmed scheduling operations are transmitted immediately to the central host, the only acceptable delay being that of network latency in a particular operating environment.

2.14.5. Implementation Considerations

An important characteristic of the central Caltool host is that it is assumed to stay running all of the time, whereas individual users may turn their computers off or disconnect them from the network. If users aren't allowed to do this, then a distributed implementation is allowed. Here's the deal with this -- whenever a Caltool user's computer is running and connected to the network, other users can see the latest public items and receive notifications immediately from that user. Whether they

OK, maybe we do need to be a bit more restrictive in the model than the "weak view" paragraphs below indicate. Try this. From each user's perspective, there's her own computer with her own personal stuff and the Caltool host computer, which is the "gateway" to everyone else's stuff. Now, the implementation freedom comes in in how the gateway computer chooses to store stuff. It can either store central copies or ship out distributed copies. If it does the latter, things might get tricky cause it's not allowed to ever let anyone's copy of the "central" database get out of date or inconsistent. As far as the central calendar is concerned, it could in fact be highly distributed. There does need to be a place wher

The following paragraphs discuss a weak view of the implementation:

The idea of the "central" calendar need not be implemented on a single central server, it need only appear that way to users. Viz., from a single user's perspective, the user owns a local file that is her personal calendar. Everyone else's calendar is "out there", accessible when asked for, assuming the computer holding the desired information is running.

The central host computer is where the database information is stored. It may or may not be where a central calendar is stored.

Need to allow the case where central calendar contains references to local calendars, as long as proper semantics are maintained.

2.14.6. Clock Synchronization Issues

These requirements assume that the system clocks for all registered users are synchronized to a common time and date. All implementations of Calendar Tool may consider this assumption to be true.

2.14.7. Files

Next paragraph is WRONG given nixing of ComCal as a distinguished file. User community calendar data are *copied* in two places -- on the user's local host and the Caltool's central host. The specific local file is that named "CommunityCalendar" in the Calendar Tool directory. Whenever the user saves this file (*not for each change*), the system sends a copy of the file to the central host to which the user is currently connected, if any.

OLD and WRONG: The central host copy is updated *every time* the user makes a change, by sending a message to the central host to have the change made.

Admin data are stored in just one place -- the central host, and *not copied* to users' local hosts. When the user views admin data, she's viewing a the host data directly, cached locally as necessary. However, admin data are never stored in users' local filesystem.

2.14.8. Administrative Purging of Excess Calendar Data

For large installations and/or for central hosts with limited storage resources, Calendar Tool administrators can purge as much as they want. As a courtesy, they may want to notify users, but they're not required to do so. See the 'Admin Purge' command.

2.14.9. Security Considerations

There are no "heroic" ones, such as ones that would try to second guess how a hacker might mess things up by fiddling the system with data, or related malicious behavior.

2.14.10. Local/Remote Calendar Synchronization

When a user is disconnected from a central host computer, the local and remote copies of the calendar can get out of sync in two ways -- (1) user adding items locally and (2) meetings being scheduled remotely. To avoid complicated remote/local file comparisons, synchronization is accomplished as follows: whenever a connect is (re)established, the complete contents of the local calendar is copied to the remote host. All of the not-yet-accepted meetings on the remote host are penciled-in on the local calendar.

2.15. System Installation and Operating Environment Interface

Cover in as OS-independent way as possible how to install the system, and how it's invoked from the OS.

2.15.1. Regular User Installation

Want to keep prompts to a minimum. I think these are they, er, this is it:

- a. "Enter the name of the directory where the Calendar Tool will store its standard files. _____"

Instead of a prompt, conclude the installation with a tip of one of the following forms, as appropriate:

You are a confirmed registered user on the
Calendar Tool central host computer *host name*.

You can use the 'Admin->Central Host' command
to change your central host or disconnect from the host
to run in stand-alone mode.

-- or --

This installation of the Calendar Tool is configured
to run as a stand-alone program that is not initially
connected to a Calendar Tool central host computer.

If you are a registered user on a Calendar Tool central host,
you can use the 'Admin->Central Host' command to connect to the host.

If you are not a registered and want to become one,
please contact your local Calendar Tool administrator.

The first form of tip happens by virtue of a calendar tool admin having built a custom config with the default name of a central host set to a host on which the user running the install is registered. The second form happens otherwise.

2.15.2. Implementor or Administrator Configuration of the Calendar Tool

When the app is built by the implementors or configured by an administrator, it contains the following hard-wired information:

- a set of default option values

2.15.3. User Installation of the Application

When installed, the app is additionally configured with the following info:

- the user's calendar tool directory

2.15.4. Administrator Installation of the Server

2.15.5. Location of Calendar Tool Files

Some directory needs to be designated as a user's caltool directory, and this directory must be that specified in a register's user's central host user database record. The default name for this directory in the standard Calendar Tool distribution is .CalendarTool. System installers and administrators must take care that the following conditions do not cause confusion or problems for users:

- a. the installation script, by some funky OS convention, puts the user's cal tool dir is some place line c:53340e user's cal tool database record says the user's cal tool dir is someplace else
- b. a user is registered on more than one calendar tool central host, with different cal tool directories, or other different user-record information on those hosts

As necessary, expand and elaborate upon these conditions, including explanation of exactly where they're problems or can lead to problems.

2.15.6. Invoking the Calendar Once Installed

Points to include:

- Init file only invoked one time at start-up, not at each file open
- Prev rule means no init file happens when other user's calendars are open.

2.16. Future Enhancements

This concluding section of the functional requirements presents an overview of possible future enhancements to the Calendar Tool system. These are enhancements that may be incorporated in a version of the system beyond the initial release that meets the functional requirements defined of the preceding sections.

Sketch of what to include:

- a. fancy printing and HTML generation
- b. more sophisticated undo/redo, including multiple levels of undo/redo
- c. more advanced scheduling that takes location attributes into account, such as room size and available equipment
- d. integration of a scripting language into the textual command functionality
- e. import/export of calendar data from and to other tools, such as Microsoft Outlook
- f. user-defined hot key mappings and other UI customization features

2.17. Work in Progress

2.17.1. Bug

In the possible meeting times list, the year should be displayed if the dates span a year. DONE.

2.17.2. The Term "Client"

Consider changing the term "local host" to "client" or "local client".

2.17.3. Window “Banner” versus “Title”

Think seriously about replacing the term "[window] banner" with "[window] title", or vice versa, or at minimum making certain that the two different terms are used consistently. The potential difference in meaning is that "banner" refers to the physical part of the top of a display window and "title" refers to the string that appears within the "banner". The reason "banner" might be better is that "Title" is already used for scheduled items, and it's potentially confusing to use the term "title" for windows as well. Anyway, think about it an MUYFM.

Also make sure that the terms "current" (for calendars (?files?) and windows) and "active" (for windows (?calendars?)) are used 100% correctly and consistently.

2.17.4. Filtering Todo Items

Go here and here.

2.17.5. Options Section Wording Fodder that Will Probably Be Nuked

Calendar Tool options ?exist? at three levels: local, standard Options file, and hard-wired into the Calendar Tool itself as it was configured for distribution.

2.17.6. Possible Fix Needed

In a paragraph of Section 2.4.1.1 just above Figure 89 there are references to sections where it's claimed that viewing of meeting minutes is covered, but a quick scan of those sections seems to reveal that meeting minutes viewing may not be all that well covered there (or anywhere, perhaps). So, check this out, to make sure that the correct sections are in fact referenced, and that there's a decent explanation of meeting minutes viewing there, or somewhere else that can be referenced.

2.17.7. Fix Links in Admin Section

Once the admin section is done, read it carefully to make sure all of the links point at the best possible places. In particular, there's currently a link to 2.14 (Multi-User Op Environment) in 2.6.2 (User DB). Depending on how much of the host computer discussion goes in 2.6 versus 2.14, the link to 2.14 may or may not be best.

2.17.8. Filtering Update Needed

To be fully consistent, the new 'Scheduled By' and Scheduled By' on fields should be listed in the custom filter dialog.

2.17.9. Tree Viewer for Groups and Other Admin-Related Discussion

Immediately below list of groups in the group DB dialog, add a small-text button labeled "Tree View ...". This brings up tree-style viewer with group and user IDs as tree node labels. There are View and Cancel buttons at the bottom of the tree viewer. The View button zooms in on a group or user record.

In the book, if not here, discuss how the collection ops for the user DB are spread over two dialogs. Viz., Add and Find are in the DB-level dialog; Change and Delete are in the record-level dialog. The rationale is that for Change and Delete, the user should be looking at the record in order to do either of these operations. In particular for delete, we want the user to see the entire details of the group record before deleting.

2.17.10. External Operating Environment Issues

I think we need to add a new 2.X-level section to deal with the interface to the outside operating environment. The section needs to address, if not among other points:

- launching the tool, for each of its launch points
- initially configuring the tool, as necessary
- details of multi-user operation

At present, we have the "Multi-User Operating Environment" section that addresses the last of these points, but not really the first two. Also, we rather weakly address launching in the File section.

From a 205 example perspective, this proposed new "Launch and Config" or "Dealing with the Underlying Operating Environment" section really needs to be there. For the Cal tool, the current "Multi-User Operating Environment" rather muddies the waters, in particular since it's not applicable to all tools. Be that as it may, we do need the "Operating Environment" section in the Cal tool requirements. A sweet way to integrate the current "Multi-User" section in this context might be to make it a subsection of "Operating Environment", so in fact the three bullets above comprise the three subsections. I think this has potential and should get to it.

2.17.11. Custom Filter Menu Options

I think we need to add a None element to the custom filter menu, since the reselection of the name is clumsy, particularly if we implement with Java-style radio button menu items.

2.17.12. Misc

Make sure list views of fac mtg show it at biweekly, not weekly intervals.

Get rid of "audible beep only" in any figure, if any, where it appears.

Say that the initial default file name for a save is the path defined by the user's home dir concatenated with the default Cal Tool file name set by the admin.

Make sure "Staff Meeting" versus "Computer Science Staff Meeting" titles are in the right places.

Move the scroll handle down in Figure 109.

Add a Close item to the windows menu and describe it. To wit, Close removes any non-modal window from the screen as well as from the windows list. Any pending action in the removed window is canceled. The definition of modal versus non-modal display window is covered in section

2.17.13. Make Sure Button and Menu Item States are Accurate

I'm pretty sure it's already in here, but don't forget to go through all dialogs and make sure the state of each button is correct. In particular, I notice today that state of the 'Clear' button in Figure 91 should be disabled, given the newly clarified meaning of 'Clear' in a dialog context such as this. Viz., 'Clear' is only enabled after at least one edit has been made, since it does not clear out all data, but only edits made since the initial display. We may want to make sure that this not entirely uniform behavior of clear makes good sense. I think it does, since even the most work-doing cases, 'Clear' does not clear everything to empty, but to some default values, or perhaps back to the most recently confirmed values, or to some well-defined "cleared" values. In the case of things like confirmation dialogs, the "default" values are those that were computed rather than seemingly more static option values. However, I think the principle can be considered effectively the same in both case, which means we're OK here rationally.

2.17.14. Quick Note

Consider adding the list of example items as an appendix, the rationale being that it was helpful during development and may therefore be helpful for the requirements reader.

2.17.15. Hmm, Centralized Server Architecture Rears Its Ugly Head

I'm inclined at this point to think that we need to add a new level-2 functional subsection that addresses the issues discussed in the next couple paragraphs. Call it something like "Inter-User Data Access". The requirements relate to all Cal Tool commands that require access to other user's calendar's and system-wide administrative information, which are: 'View Other User', 'Schedule Meeting', all admin commands.

Well, we need to deal with the issue of where not-yet-accepted meetings go. I think we want to prohibit any write access to any other users calendar. This means that not-yet-accepted items need either to go in the scheduler's place

or the central place, or perhaps some combo of the two. Perhaps we should craft requirements that allow any form of implementation. One problem with being too vague about whether there's a central server is that when a user fires up the tool, he should be told if the central server is down so he can know that there may be pending meetings that he can't see. This is getting very complicated.

So here's some more on this. In order for things to work, everyone needs to have access to everyone else's files and/or to a centralized repository. So, we need to assume some form of network where the files stay accessible at all times. When the user logs on, if there is a server problem for one or more users, then the logging on user will get a message to that effect. This way, we may well be able to get away with not specifying exactly how many central servers there are. Here's what we do say: (1) users *own* their own calendar files and no one else can write on them, including the central Cal Tool server; (2) user files are stored in such a way that they remain accessible, even if their local computer is turned off.

If we want to do a more explicit design, here's one that I think works:

- a. Every user owns a local file that's her calendar.
- b. There's a central Cal Tool site that has a *hot copy* of all publicly-accessible portions of the local file.
- c. The central site contains the admin databases.

It feels (at the moment) like the second of these requirements means there needs to be new objects defined that are the readable parts of title-only and confidential items. This really is getting complicated. Maybe what we need to do is leave things abstract and say that the central database has full data, but only provides ops that access parts of it. Hmm, this is interesting as well as complicated.

OK, here's a bit more refinement (next morning) on the (hopefully abstract) model of inter-user data access. The 'File Save' op saves both a local copy of the full calendar and a central copy. Need to think more about this, obviously.

2.17.16. Nuked These Things and Think They Can Stay Nuked

We may want to discuss the issue that if the scheduler fills in the scheduling criteria in such a way that there can be only one possible meeting time, that the system displays it in some other form than a list. Or if it's in a list, we'll just mention that the system does not distinguish this particular case from others, but just proceeds to show a list of length one. I think this is reasonable, given that there may be some who can't attend, so a length of list one seems fine. Nuff said here.

Include an "trailer" message that can be added to the "you're double booked" message that the cal tool sends to all double-booked attendees.

2.17.17. Scripting Language Derived from UI Command

I think the best place for this is in an appendix. It will be ref'd initially from the appropriate subsection of the file section.

2.17.18. A Couple More Todo Items

Need to have a discussion somewhere about invoking the tool from the command line. See where it's done in the Rolodex doc.

Consider adding a command that will save the current screen configuration. Alternately, this could be done with a .ini script file. I think I like the latter better, because the former requires some way to save the config and invoke it from the command line or icon, and we get into a bunch of OS-specific stuff. But wait -- maybe they're really the same idea. Viz., a config file is internally rep'd as a script, and can be invoked from an icon. So, we probably do as a convenience want a 'Save Config' command added to the 'File' menu.

-- OK, the save config thing is done. We'll see if we can live with it.

2.17.19. Prev/Next Arrow Issue

According to the rule that says the upper left corner of a window stays put, the left/right arrows in the full date bannerlette wont be aligned under each other when the size of the banner string changes, e.g., when the length of the month name changes. This is pretty clearly why claris put these buttons on the far left, and next to each other for convenient access. We really need to do the same. This might in fact be a cool thing to put in an SCO.

OK, here's a cool way to do it -- leave them on either side of the date label, but make the date label a fixed size, wide-enough to hold the widest possible entry, at all levels. That way the buttons stay put. Wait, that won't work if we keep the rule that the upper left corner of the window stays put,

2.17.20. Make Sure that Disabled (Greyed) State of All Buttons is Correct

The deal is we should use the concept of disabled (greyed out) means precondition is not (fully) met. That's a cool formal basis, I think, but we need to make sure all the screen shots are correct with respect to it. I think in particular that the custom list dialog screen shots at the end of Section 2.3.3.6 may not be exactly correct wrt when Change and Delete are enabled and disabled.

2.17.21. Consider Where To Put Dialog State Details

The issue was raised to prominence when finishing up the list-viewing section. The current thinking is that as a matter of style, we'll factor out such lower-level details to the data-entry-details section. Here are the pros and cons of doing so:

- Pros:
 - o relieves clutter from main scenario sections, in keeping with error-message-at-the-end philosophy
 - o clutter relief makes for better reading
- Cons:
 - o defers things unnecessarily; it might be better just to embed things in the scenario sections, or put the details at the end, either way not waiting until a later section
 - o the data-entry details section may become unnecessarily complex, effectively becoming a clone of earlier scenario sections
 - o the data-entry details section may be confusing in its back references, such as to detail tables in earlier scenarios; e.g., the ref to [list-viewing.html#Table Fields](#) applicable to each type of scheduled item in the description of custom list dialog details

OK, here's some bolstering for the Pro side -- in exploring the idea of view modeling in some new text writings, I had a bit of a sick feeling that having discovered something new about view modeling during the course of writing the custom list dialog requirements, I may have to go to the appt-scheduling section and add some refinements about initial dialog states. That was quite a sickening thought indeed, given that we're trying desperately at this point to get some closure on things. What lessened, if not eliminated altogether, this sickening feeling was that we don't need (or most likely want) to put any further dialog state discussion in the appt-scheduling section, but rather wait to do this in data-entry-details. And I do reasonably honestly believe that I'm saying this with keeping the requirements "best", not because I'm being lazy about going back to update the appt-scheduling section. I.e., it really would crud up the appt-scheduling section, I believe, if we went back and put in a bunch of details about initial dialog states.

Now what we might say about structuring a requirements document is that the very early scenarios are clean and simple, without, e.g., dialog state requirements discussed. In subsequent sections we could add in such discussion incrementally. E.g., we could put the custom-lists section, and maybe (probably also) in data-entry details. HOWEVER, the (potential) problems with this incremental approach are the following:

- a. If we rely on the initial scenarios to present concrete information, even though they're not covering all possible cases initially, then we eventually need to cover dialog state details somewhere. Since we likely will not have some section like "More Appt Scheduling Details" (witness that we don't currently in the Calendar

- requirements, nor do I think we need one), there's no place else for such details to go.
- b. So, continuing with the previous point, it doesn't seem to make good structural sense to have dialog details presented like this:
 - i. early scenario(s) free of dialog state details
 - ii. later scenarios incrementally add some or all dialog state details
 - iii. data-entry-details section does dialog details missing from early scenario(s) and some or no details for later scenarios that had some or all dialog details there.

2.17.22. Maybe for custom list section

[Also do some another kind of example that we'll apply to all items, and do some sorting -- some example that I might actually find useful.]

2.17.23. Consider adding Emacs to list of related systems in section 1.5

Do it.

2.17.24. Complete Clear Statement for Item Uniqueness Criteria

At present, I believe the only place that it's stated explicitly is in (Ah "structural-viewing.html#item uniqueness" this paragraph of Section 2.3.1.2 This is hard way to obscure a place to have such a significant piece of specification. Hence, I suggest that we state it quite clearly in the data-entry details section, if not elsewhere, such as in some appropriate intro spot.

Related to this, here's a bit of verbiage that was removed from the list-viewing section when sorting orders were fully clarified, but it may come in handy at some point when we're spec'ing item uniqueness, and how that applies to sorting:

The date/time key consists of both a date and time for appointments and meetings. Lists of the other three types of scheduled items (meetings, tasks, and events) are very similar to appointment lists. Section ??? covers the scheduling as well as the listing formats for the three additional types of items.

2.17.25. Check Consistency between Task Scheduling and Task List Sections

Since the task listing section is being completed before the task scheduling and viewing section, once the latter is fully done we need to go back to the former to make sure that we got all of the details right.

2.17.26. Default Task Priority

DONE 27jul02, given all-encompassing style of default settings.

Needs to be in the options. Also, it's a good place, if one does not show up elsewhere, to illustrate what a non-typable combo-box menu looks like when its middle item is selected. OK, it looks like we'll use the Swing style of combo boxes, where the menu never comes up centered, or otherwise relative to what's selected, but rather always below the initial box item.

2.17.27. Color

The current Figure 11 shows an 'Edit ...' item at the end of the category colors menu. We have to show an example of this or get rid of it. While it's a cool example to have, it's (a) not part of Claris, (b) of potentially limited actual utility, and (c) a bit much given that we're struggling just to get the basics done. Whatever, we'll probably leave it.

2.17.28. Beware (potentially) of automatic rescheduling

Our long-held notion of automatic rescheduling of optional items may have an impact on the list order of overlapping items, if we decide to allow the idea of preempting instead of rescheduling. In general, here seem to be at least some of the user-selectable alternatives for how/if auto resched is performed:

- move rescheduled item to the next available time slot
- move rescheduled item to the next available time slot, based on the normal hours setting
- preempt optional item(s) by moving them to the optional end of the overlaps list
- alert user when items are rescheduled or preempted (this is an on/off option by itself)

2.17.29. General fixes and updates to do

12dec00 (DONE). *Add precise explanation of the next/previous order of scheduled items, based on the aforementioned sorting order of start-time, duration, title. There's a potential problem with the total ordering of all items, given that events and tasks don't have durations. This needs to be fixed up and perhaps explained earlier in the daily and/or weekly sections. It may not have to go earlier, since in the daily and weekly views, events and tasks are displayed in separate panes, whereas with Prev/Next nav, all four item types are in one list (effectively). Anyway, it does need to be ironed out.*

3oct00. In gui-details, add discussion for details of proportional placing of items in day display. Specifically, describe what happens when an item starts at say 55 minutes after the hour, in which case proportional spacing requires that the either the item is placed as far down as possible or the box is enlarged vertically. I think I like the latter.

1oct00. Fix Figure 7 by changing the 'End Date' label to greyed out, as it appears in the immediately preceding Figure 6. Also fix the 'outside appt' string in Figure 9 to be consistent with its capitalization used in the text, viz. 'Outside Appt'. (Done, since this category has been nuked entirely.)

2jun00. Add a Text menu item between Find . . . and Command . . . on the Edit menu. Selecting the Text item brings up a simple text editor on the currently active calendar. An alternative to opening the current calendar as text would be to have an ellipses (Text . . .) menu item which brings up a file chooser to select which calendar to open as text. Yet another would be to make "plain text" an option of the Open command. I lean towards the first, since opening a calendar as plain text is likely to be a pretty unusual option, that's mostly being considered since it provides a useful MVP design and implementation example. Also, from a strictly "right thinking" end users perspective, since it is an infrequently-used command, it goes better at the end of the Edit menu than as a regularly seen item in the very frequently used File->Open command.

Make sure there's a +/- zooming control in the Help contents viewer, and the index viewer as well if the index is greater than one level deep.

Make sure there's a "Trouble Shooting" section at the end of the users manual.

Add a screen road map section.

See www.sun.com/calendaring for another related system. In particular, a page under this URL mentions the following about calendaring standards, into which we should look further:

Support for IETF Calendar Standards -- iCalendar

Sun Calendar Server adheres to IETF calendar standards (iCalendar). Its standards-based features makes it easier for users to share calendaring information across the network. And as standards continue to evolve and emerge, Sun plans to continue to provide support for these new and emerging IETF standards:

- iTIP (iCalendar Transport-Independent Interoperability Protocol)
- iMIP (iCalendar Message-Based Interoperability Protocol)
- iRIP (iCalendar Real-Time Interoperability Protocol)

2.17.30. General Constraints

By specifying a duration rather than an end time, we have to deal with events that span multiple days, and in particular how to display them in the various views. This could be a pain, which might mean that we want to go to start-time/end-time format rather than star-time/duration format. Think about this.

Having thought at some length about the end-time versus duration tradeoff, I think we've pretty much settled on duration. A discussion of this issue is in the rationale section. The current thinking for how to display items that span a day (i.e., go past midnight) is to put a small down-pointing and up-pointing arrows all the way to the left of the item titles in the display. The down-pointing arrow goes with the starting title and the up-pointing arrow goes with the ending title. I think this will work nicely and it is consistent with the small right- and left-pointing in overlapping item displays.

2.17.31. Commands with No Effect

27jul02 Note: I'm just about ready to rework this so there's no such thing as a command with no effect, but rather any time there's a command that would have no effect, it's activation element in the GUI is disabled.

This needs to go somewhere. When the user performs an command that has no effect, such as 'View Item' when there is no current item to view, the systems responds by default with an audible beep. The user has the option to change this alert to one of the following alternatives: (a) Visible bell (platform-specific screen blink); (b) context-specific alert dialog; (c) no action. For alt b, we need to list fully all of the context-specific alert dialog messages.

2.17.32. Trimming

I think it's time to get rid of the multi-week, and multi-month view ideas. These were going to allow the user to select two or more (but not a lot) of weeks or months to display in a single window. I don't really think they're necessary, given we can have as many individual windows open at a time as we want. Plus it helps with simplicity. This is a good item to put in the rationale.

2.17.33. Uniform Effect of the 'Cancel' Command

This must be explained somewhere and clearly we must make sure that the effect of 'Cancel' truly is uniform.

2.17.34. Explanatory Error and Confirmation Dialogs

We should use the term "explanatory" error dialog throughout, and then in one section show what the generic such dialog looks like and enumerate a specific error message for each place where the term "explanatory error dialog" was used. Putting the list in chronological order of term use is probably best.

2.17.35. Help

2.17.35.1. Possible Options

27jul02 note: For simplicity, screw these.

Consider adding an option `scroll-contents-on-index-display`, which if on automatically scrolls the Contents pane to the topic in which the most recently selected index term appears. Note that this rule affects the requirement for historical context of the help information, in that "last selected by user" must be changed to "last selected by user or automatically displayed by the system".

Similar to the previous, consider adding option `scroll-contents-on-search-display`.

We might also consider auto-scrolling the Index pane as the Search selections are made. However, this sounds potentially confusing, given how exactly we'd match a search string to an index topic. Hence, at this point what sounds good is just auto-scrolling the Contents pane on Index and/or Search display.

NOTE WELL that these options may be more confusing and trouble than they're worth. A hint at this is that no existing help system, including Mac, does it this way. BUT, given that they're options that are off by default, they're probably OK, if not good.

2.17.35.2. Paragraph (from Rolodex) about Quick Help Activation Area

Consider moving this paragraph to the gui-details section. In the Rolodex, it's currently in the help section. We need to consider carefully the criteria from when something goes in a main scenario versus when it gets relegated to

the gui-details section.

3. Non-Functional Requirements

Here are dumps of fodder from the old biggie burger and rolodex.

Rolodex Fodder

The non-functional requirements for the Rolodex Tool are organized into three major categories:

- system-related,
- process-related, and
- personnel-related

In general, the non-functional requirements for the Rolodex Tool are limited, given the simplicity of tool's functionality and the fact that it is public domain non-commercial software.

3.1. System

System-related non-functional requirements cover performance, operational environment, and general system characteristics.

3.1.1. Performance

Given the limited functionality of the Rolodex Tool and its limited user community, performance is not a significant issue. In general, the speed of all operations must be comparable to similar tools, such as those cited in Section 1.5 on related systems. "Comparable to" means that the Rolodex Tool operations must execute at a speed that is in the same order of magnitude as similar tools under similar computer load conditions.

There are no specific requirements for the allowable size of a rolodex in terms of the number of cards. A Rolodex on the order of thousands of cards is the likely maximum size during normal use. However, there is no specific reason that the size of the Rolodex be limited.

3.1.2. Operational Environment

There are no specific requirements for the hardware or software platform on which the Rolodex Tool must operate. As discussed in Section 2.13, specific elements of the graphical user interface may differ per the conventions of a specific operating environment.

There are no specific requirements for interoperability with external software, such as other database programs.

3.1.3. General Characteristics

The functionality of the Rolodex tool is sufficiently straightforward that the tool must be 100% reliable, accurate, and correct. "Reliable" means that the system must not fail due to a software fault that is traceable to one of its functional software components. "Accurate" means that the system must not lose or corrupt data due to a software fault that is traceable to one of its functional software components. "Correct" means that in the absence of external hardware or external software failures, the system performs all operations as defined in the Rolodex Tool formal specification.

The system must be robust in the face of hardware power failures in terms of data loss. "Robust" means that when computer power fails or is interrupted, the Rolodex Tool performs a Save operation on the currently open rolodex file, if the workspace is modified.

There are no specific requirements for security, privacy, safety, portability, modifiability, or extensibility.

3.2. Process

The system is to be developed using the software process employed by Gene Fisher in his software engineering classes at Cal Poly university. The process is currently described in lecture notes for the graduate course in software engineering.

3.3. Personnel

The development is to be conducted by a single individual (Gene Fisher) who is assumed to have the necessary knowledge and skills to complete the project. There are no specific requirements for the end users of the Rolodex Tool. Given its simple functionality, it is intended to be usable by end users with only basic computer literacy skills.

Biggie Burger Fodder
Named "Goals, Constraints, and Non-Functional Requirements" there

This section contains the information indicated in the section heading. Examples for the Biggie Burger system follow.

Given below are quantitative and qualitative constraints that must be met by the Biggie Burger system. Also given are constraints on the procurement and development of the system. The following terminology is used:

- Owners: the owners of the Biggie Burger restaurant for whom the system is being specified
- Customers: the food-ordering restaurant customers who will use the food ordering subsystem.
- Managers: the restaurant management staff, including the owners, who will use the inventory control subsystem.
- Developers: the team or individual who will procure and/or develop the system to meet the specifications set forth in this document.
- Analysts: the authors of this specification document.

3.4. System Performance Constraints

- The system shall respond to any single customer ordering operation within five seconds. "Respond" means that the appropriate operation output will be displayed to the user within the five second limit.
- When the AllDone selection is made and confirmed by the customer, the receipt shall be produced within ten seconds and the customer order shall be transmitted to the cook within five seconds.
- The overall performance of the inventory control system shall be no worse than the average performance of comparable systems currently available in the market place. Average performance data are based on the article "Comparing Restaurant Inventory Control Systems" appearing in the November 1990 issue of Restaurant Owners Digest.

3.5. Qualitative system characteristics

- The overriding characteristic of the customer ordering interface is simplicity of operation. If any interface changes are made (see below), this goal of simplicity must be met.
- Simplicity is also a desired characteristic for the inventory control subsystem.

3.6. Development Goals and Constraints

The owners seek to have the system fully operational within six months from the date of issuance of a contract to the developers. All on-site development work must be done after normal restaurant operation hours.

The concrete user interface defined in the final section of this document is subject to change by developers if they deem it necessary. Any such interface changes must be fully negotiated with the owners and analysts.

The developers shall install the system in the following stages, to allow a graceful phase-in of the new system:

- a. The owners shall evaluate the entire system prior to any installation in the restaurant.
- b. Five restaurant tables will be equipped with the new food ordering system in the first stage of operation.
- c. After evaluation of the first stage, including any necessary system upgrades, the remaining tables shall be equipped.
- d. The inventory management subsystem will be installed after the initial five-table test phase.
- e. After complete evaluation, the system shall be deemed fully operational.
- f. During all stages preceding full operation, the extant food ordering and inventory control system shall remain in place, and can be brought into service immediately in case of failure in the new system.

4. Developer Overview

Here is fodder from Bigger Burger era 440 final report overview handout.

The Biggie Burger system is composed of two subsystems: customer food ordering and inventory management. The customer ordering subsystem communicates with the food-ordering customers seated at the table. This subsystem handles customer requests, and sends the food order to the kitchen when ordering is completed. The inventory control subsystem handles manager updates to the restaurant databases.

The major operation of the customer ordering subsystem is OrderFood. It takes a restaurant menu and customer selections as input, and produces a customer order. The order is accumulated incrementally, by suboperations OrderBurgers, OrderSides, and OrderDrinks.

The major objects handled by OrderFood are the FoodMenu, CustomerOrder, CustomerReceipt and CookOrder. The FoodMenu and Customer order are displayed on the ordering screen. The CustomerReceipt is produced on paper when the customer indicates that the ordering is done. The CookOrder is sent electronically to the kitchen where it is displayed on the cook's screen. The cook order is also sent electronically to the inventory control system, wherein the inventory database is automatically debited to reflect the ordered items.

The principal ordering constraint is that the customer-selected food item is currently available (i.e, there is stock on hand to make up the food item). The Menu will be updated as necessary during normal business to reflect immediate changes in inventory. E.g., if all chili has been consumed, then the Chili Burger item will be deleted from the menu, where deletion is specified by marking the item as "temporarily out of stock" on the menu.

The inventory control subsystem has three major objects and corresponding operations: the FoodMenu, the FoodStuffsInventory, and the FoodItemBreakDownList. The FoodMenu is the top-level menu seen by the customer from which order selections are made. It is composed of sections for Burgers, Side Orders, and Drinks. Manager-level Operations are provided to add, delete, and modify menu sections.

FoodStuffsInventory is the inventory of raw food items, such as burger patties, buns etc. This inventory contains a description of all stock on hand. Operations are provided to add, delete, and modify inventory items. Inventory report generation operations are also provided. These operations can produce daily, weekly, monthly, and yearly inventory reports.

The FoodItemsBreakDownList contains the complete breakdown for each food item on the menu. Each breakdown describes exactly what raw food stuffs are need to prepare a menu item. Operations are provided to add, delete, and modify food item breakdowns.

5. Formal Specification

The modules of the formal specification are generally organized following the functional decomposition presented in the command menu bar and outlined in the developer overview. The modules are the following:

- scheduling
- calendar database
- viewing
- administration
- options
- file handling
- data editing

Since the file handling and editing are less germane to the specific functionality of the Calendar Tool, these two modules are presented after the others. In addition, there is a calendar database module that defines the user-level functionality of the calendar storage repository.

5.1. Scheduling (schedule.rsl)

```
(****
*
* Module Schedule defines the objects and operations related to scheduling
* appointments, meetings, tasks, and events in the Calendar system. It also
* defines the objects and operations for defining calendar item categories.
*
*)

module Schedule;

  from StandardLibrary import Object, AddWithRetentionAndNoJunk;

  from CalendarDB import CalendarDB, UserWorkspace, UserCalendar,
    RequiresSaving;
  from Admin import UserId;
  export ScheduledItem, Title, StartTime, Duration, Category, Categories,
    Location, Security, Priority, Date, Hours, Minutes, DateNumber, Time,
    Hour, Minute, AmOrPm, DayName, MonthName;

  object ScheduledItem is
    components: title:Title and start_or_due_date:StartOrDueDate and
      end_date:EndDate and category:Category and recurring_id:RecurringId;
    description: (*
      A ScheduledItem is the generic definition for the types of items stored
      in a calendar. The Title component is a brief description of what the
      item is for. The StartOrDueDate and EndDate components indicate when
      the item is scheduled. The Category component is used to organize
      items into related color-coded categories. RecurringId is used to
      identify the common instances of a recurring item.
      <p>
      There are four specializations of ScheduledItem. They are <a href=
      #Appointment> Appointment </a>, <a href = #Meeting> Meeting </a>, <a
      href = #Event> Event </a>, and <a href = #Task> Task </a>, q.q.v.
      *);
  end ScheduledItem;

  object Appointment inherits from ScheduledItem
    components: start_time:StartTime and duration:Duration and
      recurring:RecurringInfo and location:Location and security:Security and
      priority:Priority and remind_info:RemindInfo and details:Details;
    description: (*
```

An Appointment adds a number of components to a generic ScheduledItem. The StartTime and Duration indicate when the appointment starts and how long it lasts. The RecurringInfo defines if and how an appointment recurs. The Location is where it is held. The Security indicates who can see that the appointment is scheduled. Priority is how important the appointment is. RemindInfo indicates if and how the user is reminded of the appointment. Details are free form text describing any specific appointment details.

```

*);
end Appointment;

object Meeting inherits from Appointment
  components: Attendees and MeetingMinutesLocation and ScheduledBy and
    ScheduledOn and FieldEditDates;
  description: (*
    A Meeting adds five components to an Appointment. The Attendees
    component reflects the fact that a meeting is scheduled for more than
    one person, whereas an appointment is for a single user. The
    MeetingMinutesLocation is URL for the minutes of a meeting, once it has
    been held. The ScheduledBy component specifies which user scheduled
    the meeting. ScheduledOn specifies the date and time on which it is
    scheduled. Taken together, the ScheduledBy and ScheduledOn values
    provide a unique identification for a scheduled meeting, for use by the
    meeting change and delete operations, q.q.v. The FieldEditDates
    component defines the most recent change date for each meeting field.
    This information used by the MergeMeetingChanges auxiliary function.
  *);
end Meeting;

object Task inherits from ScheduledItem
  components: DueTime and RecurringInfo and SimpleSecurity and
    TaskPriority and RemindInfo and Details and CompletedFlag and
    CompletionDate;
  description: (*
    Like an Appointment, a Task adds a number of components to a generic
    ScheduledItem. A task differs from an appointment as follows: (1)
    Appointments have Duration and Location, Tasks do not. (2) For
    appointments, the priority is either 'Must' or 'Optional'; for tasks,
    priority is a positive integer indicating the relative priority of a
    task compared to other tasks. (3) Tasks have a CompletedFlag and
    CompletionDate components; appointments do not.
  *);
end Task;

object Event inherits from ScheduledItem
  components: SimpleSecurity;
  description: (*
    An Event is the simplest type of ScheduledItem. The only component it
    adds to a ScheduledItem is SimpleSecurity.
  *);
end Event;

object Title is string
  description: (*
    A Title is a free-form text string describing a scheduled item.
  *);
end Title;

object StartOrDueDate is Date
  description: (*
    StartOrDueDate is a multi-purpose component of ScheduledItem. Its

```

purpose depends on whether an item is a Task and whether it is recurs. For non-recurring appointments, meetings and events, StartOrDueDate is used as the single date on which the item is scheduled. If the item is recurring or a multi-day event, StartOrDueDate is the first date on which it occurs. For a non-recurring Task, StartOrDueDate is the single date the task is due. If the task is recurring, StartOrDueDate is the first date it is due.

```

*);
end StartOrDueDate;

object EndDate is Date
  description: (*
    In recurring appointments, meetings, and tasks, the end date defines
    the last date on which the item will recur. In events, the end date
    defines the last date of a multi-day event. When the value of end date
    is empty, the StartOrDueDate component is interpreted as the single
    date on which the item occurs.
  *);
end EndDate;

object Date is day:DayName and number:DateNumber and month:MonthName and
  year:Year
  description: (*
    A Date is the basic unit of calendar time keeping, consisting of a day
    of the week, numeric date, month, and year.
  *);
end Date;

object DayName is Sunday or Monday or Tuesday or Wednesday or Thursday or
  Friday or Saturday
  description: (*
    The seven standard names of the days of the week.
  *);
end DayName;

object Sunday
  description: (* One of the seven days of the week. *);
end;

object Monday
  description: (* One of the seven days of the week. *);
end;

object Tuesday
  description: (* One of the seven days of the week. *);
end;

object Wednesday
  description: (* One of the seven days of the week. *);
end;

object Thursday
  description: (* One of the seven days of the week. *);
end;

object Friday
  description: (* One of the seven days of the week. *);
end;

object Saturday
  description: (* One of the seven days of the week. *);
end;

```



```
end;

object MonthName is jan:January or feb:February or mar:March or apr:April or
    may:May or jun:June or jul:July or aug:August or sep:September or
    oct:October or nov:November or dec:December
    description: (*
        One of the twelve months of the year.
    *);
end MonthName;

object January
    description: (* One of the twelve months of the year. *);
end;

object February
    description: (* One of the twelve months of the year. *);
end;

object March
    description: (* One of the twelve months of the year. *);
end;

object April
    description: (* One of the twelve months of the year. *);
end;

object May
    description: (* One of the twelve months of the year. *);
end;

object June
    description: (* One of the twelve months of the year. *);
end;

object July
    description: (* One of the twelve months of the year. *);
end;

object August
    description: (* One of the twelve months of the year. *);
end;

object September
    description: (* One of the twelve months of the year. *);
end;

object October
    description: (* One of the twelve months of the year. *);
end;

object November
    description: (* One of the twelve months of the year. *);
end;

object December
    description: (* One of the twelve months of the year. *);
end;

object Year is integer
    description: (*
        The four-digit year number. (Yes, this Calendar Tool has a Y10K
```

```
        problem.)
    *);
end Year;

object StartTime is Time
  description: (*
    StartTime is the starting time of a scheduled appointment or meeting.
  *);
end StartTime;

object DueTime is Time
  description: (*
    DueTime is the time a task is due.
  *);
end DueTime;

object Duration is
  components: hours:Hours and minutes:Minutes;
  description: (*
    Duration is the time length of a scheduled item, in hours and minutes.
    The minimum duration value is 1 minute. The maximum is 999 hours.
  *);
end Duration;

object Hours is integer
  description: (*
    Hours component of a scheduled item duration, between 0 and 999.
  *);
end Hours;

object Minutes is integer
  description: (*
    Minutes component of a scheduled item duration, between 0 and 60*999.
  *);
end Minutes;

object Time is hour:Hour and minute:Minute and am_or_pm:AmOrPm
  description: (*
    A Time consists of an hour, minute, and AM or PM indicator. A time
    value is expressed using a 12-hour or 24-hour clock style. The clock
    style is set as an option by the user. If the clock style is 24-hour,
    the AmOrPm indicator is nil.
  *);
end Time;

object Hour is integer
  description: (*
    The hour component of a time value, between 1 and 12 or 0 and 24 based
    on the clock style in use.
  *);
end Hour;

object Minute is integer
  description: (*
    The minute component of a time value, between 0 and 59.
  *);
end Minute;

object AmOrPm is AM or PM
  description: (*
    Standard suffix used in 12-hour time value.
  *);
end AmOrPm;
```

```

    *);
end AmOrPm;

object AM
  description: (* The 12-hour time suffix indicating a morning time. *);
end;

object PM
  description: (* The 12-hour time suffix indicating an afternoon time. *);
end;

(**
 * Something's gotta get fixed in this vicinity given that recurring
 * information is different for Meetings versus MeetingRequests.
 *)
object RecurringInfo is
  components: IsRecurring and Interval and IntervalDetails;
  description: (*
    RecurringInfo has components to specify the nature of a recurring item.
    IsRecurring is an on/off flag that indicates whether an item recurs.
    Interval is one of Weekly, Biweekly, Monthly, or Yearly. The
    IntervalDetails component defines the precise means to define
    recurrence for the different interval levels.
  *);
end RecurringInfo;

object IsRecurring is boolean
  description: (*
    True of a scheduled item is recurring.
  *);
end IsRecurring;

obj Interval is
  components: Weekly or Biweekly or Monthly or Yearly;
  description: (*
    Interval specifies the granularity at which recurring items are
    defined. The Weekly and Biweekly settings allow the user to specify
    recurrence on one or more days of the week. The Monthly setting allows
    the user to specify recurrence on one or more days in one or more weeks
    of each month. The Yearly setting allows the user to specify
    recurrence on one or more specific dates in the year.
  *);
end Interval;

object Weekly
  description: (* Indicates a scheduled item recurs weekly. *);
end;

object Biweekly
  description: (* Indicates a scheduled item recurs bi-weekly. *);
end;

object Monthly
  description: (* Indicates a scheduled item recurs monthly. *);
end;

object Yearly
  description: (* Indicates a scheduled item recurs yearly. *);
end;

object IntervalDetails is

```

```

components: WeeklyDetails or MonthlyDetails;
description: (*
    IntervalDetails are either weekly or monthly.
*);
end IntervalDetails;

object WeeklyDetails is
components: OnSun and OnMon and OnTue and OnWed and OnThu and OnFri and
    OnSat;
description: (*
    WeeklyDetails has an on/off setting for each day of the week on which
    an item recurs. These details are also used for the BiWeekly setting
    of the recurrence interval.
*);
end WeeklyDetails;

object MonthlyDetails is
components: MonthlyDayDetails or MonthlyDateDetails;
description: (*
    MonthlyDetails can be specified on a day-of-the-week basis or on
    specific date(s) basis. The two components contain the specific
    details for these two types of settings.
*);
end MonthlyDetails;

object MonthlyDayDetails is
components: FirstWeekDetails and SecondWeekDetails and ThirdWeekDetails and
    FourthWeekDetails and FifthWeekDetails and LastWeekDetails;
description: (*
    MonthlyDayDetails contains a weekly details component for each possible
    week of a month. The First- through ThirdWeekDetails are distinct for
    all possible months. Depending on the configuration of a particular
    month in a particular year, there is potential conflict in specifying
    recurrence in the fourth, fifth, or last weeks. The conflicts are
    resolved as follows:


<p>


    For months with 4 weeks only, the settings in FifthWeekDetails do not
    apply, and the settings in LastWeekDetails, if present, override any
    settings in FourthWeekDetails. For months with 5 weeks only, the
    settings in LastWeekDetails, if present, override any settings in
    FifthWeekDetails. (For months with 6 weeks, the LastWeekDetails
    component applies to the 6th week, and there are no conflicts.)
*);
end MonthlyDayDetails;

object MonthlyDateDetails is
components: DateNumber*;
description: (*
    MonthlyDateDetails is a list of zero or more specific dates in a month
    on which an item recurs.
*);
end MonthlyDateDetails;

object DateNumber is integer
description: (*
    The numeric value of the date in a month, between 1 and 31.
*);
end DateNumber;

object OnSun is boolean
description: (* Indicates that an item recurs on Sunday. *);

```

```
end;

object OnMon is boolean
  description: (* Indicates that an item recurs on Monday. *);
end;

object OnTue is boolean
  description: (* Indicates that an item recurs on Tuesday. *);
end;

object OnWed is boolean
  description: (* Indicates that an item recurs on Wednesday. *);
end;

object OnThu is boolean
  description: (* Indicates that an item recurs on Thursday. *);
end;

object OnFri is boolean
  description: (* Indicates that an item recurs on Friday. *);
end;

object OnSat is boolean
  description: (* Indicates that an item recurs on Saturday. *);
end;

object FirstWeekDetails is WeeklyDetails
  description: (*
    Scheduling details for the first week of a month.
  *);
end;

object SecondWeekDetails is WeeklyDetails
  description: (*
    Scheduling details for the second week of a month.
  *);
end;

object ThirdWeekDetails is WeeklyDetails
  description: (*
    Scheduling details for the third week of a month.
  *);
end;

object FourthWeekDetails is WeeklyDetails
  description: (*
    Scheduling details for the fourth week of a month.
  *);
end;

object FifthWeekDetails is WeeklyDetails
  description: (*
    Scheduling details for the fifth week of a month.
  *);
end;

object LastWeekDetails is WeeklyDetails
  description: (*
    Scheduling details for the last week of a month.
  *);
```

```

end;

object RecurringId is integer
  description: (*
    RecurringId is the unique identifier shared by all instances of a
    recurring item or multi-day event. When a recurring item is scheduled,
    a collection of separate instances is created for each recurrence.
    When an instance is subsequently changed, the system may need to access
    all other instances in the collection. The RecurringId is used for
    this purpose. Viz., all instances in the same recurring collection
    have the same value for RecurringId. Recurring Id is used in the same
    way for the multiple days of a multi-day event.

    <p>
    Using an integer representation for RecurringId is reasonable, since
    the exact representation is not directly visible to the user. From the
    user's perspective, the requirements say that "the system keeps track
    of instance collections as a whole, for the purposes of change and
    delete." In this sense, a simple integer tag is a perfectly fine
    representation.
  *);
end RecurringId;

object Category inherits from Object is
  components: name:Name and color:StandardColor;
  description: (*
    A Category has a Name and StandardColor, which serve to distinguish it
    from other categories. Colored-coded categories serve as visual cues
    to the user when viewing lists of scheduled items in some form.
    Categories can also be used in filtered viewing.
  *);
end Category;

object StandardColor is
  components: Black or Brown or Red or Orange or Yellow or Green or Blue or
    Purple;
  description: (*
    A StandardColor is one of a fixed set of possibilities.
  *);
end StandardColor;

object Black
  description: (* One of the built-in category colors. *);
end;

object Brown
  description: (* One of the built-in category colors. *);
end;

object Red
  description: (* One of the built-in category colors. *);
end;

object Orange
  description: (* One of the built-in category colors. *);
end;

object Yellow
  description: (* One of the built-in category colors. *);
end;

object Green

```

```
    description: (* One of the built-in category colors. *);
end;

object Blue
  description: (* One of the built-in category colors. *);
end;

object Purple
  description: (* One of the built-in category colors. *);
end;

object Location is string
  description: (*
    Location is a free-form string indicating in what physical location an
    item is scheduled.
  *);
end Location;

object Security is
  components: Public or TitleOnly or Confidential or Private;
  description: (*
    Security is one of four possible levels, from Public to Private. The
    levels specify the degree of visibility an appointment or meeting has
    to other users.
  *);
end Security;

object Public
  description: (*
    Public security means other users can see the scheduled item and all of
    its component information.
  *);
end Public;

object TitleOnly
  description: (*
    TitleOnly security means other users can see title, time, and date
    components only. TitleOnly security applies only to appointments and
    meetings.
  *);
end TitleOnly;

object Confidential
  description: (*
    Confidential security means other users can only see that a user is
    unavailable for the time period of a scheduled item; no other
    information about the scheduled item is visible. Confidential security
    applies only to appointments and meetings.
  *);
end Confidential;

object Private
  description: (*
    Private security means other users see no information at all about a
    scheduled item, not even that the item is scheduled. Note that private
    security hides a scheduled item from the ScheduleMeeting operation,
    q.v., so that a meeting may be scheduled at the same time as a private
    appointment. It is up to the user to handle this situation by
    accepting or refusing the scheduled meeting.
  *);
end Private;
```

```

object SimpleSecurity is
  components: Public or Private;
  description: (*
    Simple security has only two levels -- public or private. It applies
    to tasks and events. Cf. Security.
  *);
end SimpleSecurity;

object Priority is
  components: Must or Optional;
  description: (*
    Priority indicates whether an appointment is a must or if it is
    optional. This information is used to indicate the general importance
    of an appointment to the user. The operational use of Priority is in
    the ScheduleMeeting operation, where the meeting scheduler can elect to
    consider optional appointments as allowable times for a meeting.
  *);
end Priority;

object Must
  description: (*
    Indicates a scheduled item is 'Must' priority.
  *);
end;

object Optional
  description: (*
    Indicates a scheduled item is 'Optional' priority.
  *);
end;

object TaskPriority is integer
  description: (*
    A TaskPriority is a positive integer that defines the priority of one
    task relative to others. The priority value range is 0 to 10, with a
    higher value indicating a higher priority.
  *);
end TaskPriority;

axiom forall (tp: TaskPriority) tp >= 0;

object RemindInfo is
  components: IsReminded and RemindWhen and RemindWhere;
  description: (*
    RemindInfo defines if, when, and where a scheduled item reminder is
    sent to the user.
  *);
end RemindInfo;

object IsReminded is boolean
  description: (*
    True of a reminder is to be sent for a scheduled item.
  *);
end IsReminded;

object RemindWhen is
  components: integer and ReminderTimeUnits;
  description: (*
    RemindWhen defines how soon before a scheduled item the reminder is to
    be sent. The time units are minutes, hours, or days.
  *);

```



```
end RemindInfo;

object ReminderTimeUnits is
  components: MinutesBefore or HoursBefore or DaysBefore;
  description: (*
    Reminders can come minutes, hours, or days before the start time of a
    scheduled item. Each of these units can be fractional, for maximum
    flexibility.
  *);
end ReminderTimeUnits;

object MinutesBefore is real
  description: (*
    Number of minutes before a scheduled item to issue the reminder.
  *);
end MinutesBefore;

object HoursBefore is real
  description: (*
    Number of hours before a scheduled item to issue the reminder.
  *);
end HoursBefore;

object DaysBefore is real
  description: (*
    Number of days before a scheduled item to issue the reminder.
  *);
end DaysBefore;

object RemindWhere is OnScreen or BeepOnly or Email
  description: (*
    RemindInfo defines one of three ways that the user is alerted when a
    scheduled event is to occur. OnScreen means the user is reminded with
    a pop-up alert on her computer screen. BeepOnly means the user is
    reminded with a simple audible tone on the computer. Email means the
    user is sent an electronic mail message reminder.
  *);
end RemindInfo;

object OnScreen
  description: (*
    Indicates that reminders are displayed in an on-screen pop-up window.
  *);
end OnScreen;

object BeepOnly
  description: (*
    Indicates that reminders are signaled with an audible beep only, with
    no pop-up window.
  *);
end BeepOnly;

object Email
  description: (*
    Indicates that reminders are send as email messages.
  *);
end Email;

object Details is string
  description: (*
    Details is a plain text string describing any details about a scheduled
```

```

        item the user may care to add.
    *);
end Details;

object Attendees
  components: Name*;
  description: (*
    Attendees is a list of names of those who attend a meeting.
  *);
end Attendees;

object Name is string
  description: (*
    The name of a category or a meeting attendee.
  *);
end;

object MeetingMinutesLocation is string
  description: (*
    MeetingMinutes is a plain text string that records the file or URL
    location of the minutes taken for a meeting. The syntax of the
    location string is presumed to be compatible with the program that is
    used to view meeting minutes, as specified in the system options. This
    syntax has no specific meaning to the Calendar Tool.
  *);
end MeetingMinutes;

object ScheduledBy is UserId
  description: (*
    ScheduledBy is the id of user who scheduled the meeting, i.e., the user
    who executed the ScheduleMeeting operation that led to the meeting being
    scheduled.
  *);
end ScheduledBy;

object ScheduledOn
  components: Date and Time;
  description: (*
    ScheduledOn is the date and time on which a meeting is scheduled.
  *);
end ScheduledOn;

object FieldEditDates is
  components: FieldEditDate*;
  description: (*
    List of dates for meeting fields that have been edited since the last
    time the meeting was saved.
  *);
end FieldEditDates;

object FieldEditDate is
  components: FieldName and Date;
  description: (*
    A FieldEditDate is the name of a meeting field and the date it was last
    edited. This information is used to merge the changes made
    independently by the scheduler and attendee of a meeting. The policy
    is to merge changes, such that the merge value for each field is the
    value most recently edited by either the scheduler or attendee.
  *);
end FieldEditDate;

```

```

object FieldName is string
  description: (*
    The string name of a meeting field.
  *);
end;

object CompletedFlag is boolean
  description: (*
    CompletedFlag is true if a Task has been completed, false if not. The
    system does not enforce any specific constraints on the setting of a
    task's CompletedFlag. That is, the user may set or clear it at will.
    Hence the meaning of the CompletedFlag is subject to user
    interpretation, particularly for recurring tasks.
  *);
end CompletionDate;

object CompletionDate is
  components: Date;
  description: (*
    CompletionDate is date on which as task is completed. The system does
    not enforce any specific constraints on the setting of a task's
    CompletionDate (other than it being a legal Date value). The meaning
    of the CompletionDate value is up to user interpretation, particularly
    for recurring tasks.
  *);
end CompletionDate;

operation ScheduleAppointment is
  inputs: cdb:CalendarDB, appt:Appointment;
  outputs: cdb':CalendarDB;
  description: (*
    ScheduleAppointment adds the given Appointment to the current Calendar
    in the UserWorkSpace of the given CalendarDB. In the case of a
    recurring appointment, a separate appointment instance is added per
    the recurring information.

    <p>
    The addition is only made if an appointment of the same time, duration,
    and title is not already scheduled. In the case of a recurring
    appointment, this means that none of recurring instances to be added
    can have the same time, duration, and title as an already scheduled
    appointment.

    <p>
    If the addition is made, the RequiresSaving flag of the calendar is set
    to true.
  *);
precondition:
  (*
  * The StartOrDueDate field is not empty and a valid date value.
  *)
  (appt.start_or_due_date != nil) and DateIsValid(appt.start_or_due_date)

  and

  (*
  * If non-empty, the EndDate field is a valid date value.
  *)
  (appt.end_date != nil) or DateIsValid(appt.end_date)

  and

  (*

```

```

    * The duration is between 1 minute and 999 hours, inclusive.
    *)
    DurationInRange(appt.duration)

    and

    (*
    * No appointment or meeting instance of the same StartTime, Duration,
    * and Title is in the current workspace calendar of the given
    * CalendarDB. The current calendar is
    *
    *     cdb.workspace.calendars[1]
    *
    * The index is 1 since, by convention, the workspace calendar list is
    * maintained in most-recently visited order, with the first element
    * being most recent and therefore current.
    *)
    not (exists (item in cdb.workspace.calendars[1].items)
        (item.start_or_due_date = appt.start_or_due_date) and
        (item.<Appointment.duration = appt.duration) and
        (item.title = appt.title))
    ;
postcondition:
    (*
    * A scheduled item is in the output calendar if and only if it is the
    * new appointment to be added or it is in the input calendar. As
    * noted in the precond, the current calendar is
    *
    *     DB.workspace.calendars[1]
    *
    * where here in the postcond, DB is either cdb or cdb'.
    *)
    -- NOTE: This must be updated to reflect separate instances being added
    -- for recurring appointments.
    (forall (item:ScheduledItem)
        (item in cdb'.workspace.calendars[1].items) iff
        ((item.<Appointment = appt) or
        (item in cdb'.workspace.calendars[1].items)))

    and

    (cdb'.workspace.calendars[1].requires_saving = true);

end ScheduleAppointment;

operation ScheduleMeeting is
    inputs: CalendarDB, MeetingRequest;
    outputs: PossibleMeetingTimes;
    description: (*
        ScheduleMeeting uses the given MeetingRequest to determine possible
        times that the requested meeting might be held, within the existing set
        of scheduled items in the CalendarDB. The PossibleMeetingTimes output
        is a list of zero or more possible times and dates that the meeting can
        be held.
    *);
    precondition: (* User must be group leader *);
    postcondition: ;
end ScheduleMeeting;

(**
* Something's gotta get fixed in this vicinity given that recurring

```

```
* information is different for Meetings versus MeetingRequests. In order
* for a where clause sub of RecurringInfo to work, we gotta do some work to
* distinguish recurring info for meetings versus meeting requests.
*)
object MeetingRequest inherits from Meeting
  where: StartDate = EarliestStartDate, EndDate=LatestEndDate,
         StartTime=EarliestStartTime;
  components: LatestStartDate and LatestEndDate and LatestStartTime;
  description: (*
    A meeting request has all the components of a meeting plus three
    additional components to specify the latest dates and time at which the
    meeting can be scheduled. A meeting request is used to specify a range
    of possible meeting times, to allow scheduling alternatives to be
    considered. The description of the ScheduleMeeting operation has
    further details on how meeting requests are handled.
  *);
end MeetingRequest;

object EarliestStartDate is Date
  description: (*
    In a meeting request, the earliest possible start date for the meeting
    to be scheduled.
  *);
end;

object LatestStartDate is Date
  description: (*
    In a meeting request, the latest possible start date for the meeting to
    be scheduled.
  *);
end;

object EarliestEndDate is Date
  description: (*
    In a meeting request, the earliest possible end date for a recurring
    meeting to be scheduled.
  *);
end;

object LatestEndDate is Date
  description: (*
    In a meeting request, the latest possible end date for a recurring
    meeting to be scheduled.
  *);
end;

object EarliestStartTime is Time
  description: (*
    In a meeting request, the earliest possible start time for the meeting
    to be scheduled.
  *);
end;

object LatestStartTime is Time
  description: (*
    In a meeting request, the latest possible start time for the meeting to
    be scheduled.
  *);
end;

object PossibleMeetingTimes is (StartTime and Date)*;
```

```

operation ConfirmMeeting is
  inputs: CalendarDB, MeetingRequest, PossibleMeetingTimes, SelectedTime;
  outputs: CalendarDB;
  description: (*
    ConfirmMeeting takes a CalendarDB, MeetingRequest, list of
    PossibleMeetingTimes, and a Selected time from the list. It outputs a
    new CalendarDB with the given request scheduled at the selected time.
    Further details of output constraints are forthcoming.
  *);
  precondition: ;
  postcondition: ;
end ConfirmMeeting;

object SelectedTime is integer;

operation ScheduleTask is
  inputs: CalendarDB, Task;
  outputs: ;
  description: (*
    ScheduleTask adds the given Task to the given CalendarDB, if a task of
    the same time, duration, and title is not already scheduled.
  *);
  precondition: ;
  postcondition: ;
end ScheduleTask;

operation ScheduleEvent is
  inputs: cdb:CalendarDB, e:Event;
  outputs: cdb':CalendarDB;
  description: (*
    ScheduleEvent adds the given Event to the given CalendarDB, if an event
    of the same start date and title is not already scheduled.
  *);
  precondition:
    (*
      * The Title field is at least one character long.
    *)
    (#(e.title) >= 1)

    and

    (*
      * The StartOrDueDate field is not empty and a valid date value.
    *)
    (e.start_or_due_date != nil) and DateIsValid(e.start_or_due_date)

    and

    (*
      * If non-empty, the EndDate field is a valid date value.
    *)
    (e.end_date != nil) or DateIsValid(e.end_date)

    and

    (*
      * The current workspace is not nil.
    *)
    (cdb.workspace.calendars != nil)

    and

```

```

(*)
* No event of same StartDate and Title is in the current workspace
* calendar of the given CalendarDB. The current calendar is
*
*      cdb.workspace.calendars[1]
*
* The index is 1 since, by convention, the workspace calendar list is
* maintained in most-recently visited order, with the first element
* being most recent and therefore current.
*)
not (exists (item in cdb.workspace.calendars[1].items)
      (item.start_or_due_date = e.start_or_due_date) and
      (item.title = e.title))
;
postcondition:
(*)
* A scheduled item is in the output calendar if and only if it is the
* new event to be added or it is in the input calendar. As noted in
* the precond, the current calendar is
*
*      DB.workspace.calendars[1]
*
* where here in the postcond, DB is either cdb or cdb'.
*)
(forall (item:ScheduledItem)
      (item in cdb'.workspace.calendars[1].items) iff
      ((item.<Event = e) or
      (item in cdb'.workspace.calendars[1].items)))

      and

      (cdb'.workspace.calendars[1].requires_saving = true);
end ScheduleEvent;

operation ChangeItem is
  inputs: cdb:CalendarDB, si_old:ScheduledItem, si_new:ScheduledItem;
  outputs: cdb':CalendarDB;
  description: (*
    ChangeItem changes the given old item to the given new item in the
    given CalendarDB.
  *);
  precondition: ;
  postcondition: ;
end ChangeItem;

operation DeleteItem is
  inputs: cdb:CalendarDB, si:ScheduledItem;
  outputs: cdb':CalendarDB;
  description: (*
    DeleteItem deletes the given item from the given CalendarDB.
  *);
  precondition: ;
  postcondition: ;
end DeleteItem;

(**
* Auxiliary functions.
*)
function DateIsValid(date:Date)->boolean = (

```

```

(*)
* The date number is between one and the proper month-specific upper
* bound.
*)
(date.number >= 1) and DateNumUpperBoundIsValid(
    date.number, date.month, date.year)

    and

(*)
* The year is between 1 and 9999.
*)
(date.year >= 1) and (date.year <= 9999);

)
description: (*
    Auxiliary function that specifies date validity.
*);
end;

function DateNumUpperBoundIsValid(num:DateNumber, month:MonthName,
    year:Year)->boolean = (

    (*)
    * 30 days has September, and all that ... .
    *)

    (*)
    * Upper bound for jan, mar, may, jul,aug, oct, and dec is 31 days.
    *)
    if (month?jan or month?mar or month?may or month?jul or month?aug or
        month?oct or month?dec)
    then num <= 31
    else

        (*)
        * Upper bound for apr, jun, sep, and nov is 30 days.
        *)
        if (month?apr or month?jun or month?sep or month?nov)
        then num <= 30
        else (* Month is feb *)

            (*)
            * Upper bound for feb is 29 if leap year, 28 if not.
            *)
            if IsLeapYear(year)
            then num <= 29
            else num <= 28;

        )
    description: (*
        Auxiliary function that specifies valid upper bound for day number
        component of a date.
    *);
end;

function IsLeapYear(year:Year)->boolean = (

    (*)
    * The year is divisible by 4 but not by 400.
    *)
    (year mod 4 = 0) and (year mod 400 != 0);

```



```

)
description: (*
    Auxiliary function that specifies leap year definition.
*);
end;

function DurationInRange(duration:Duration)->boolean = (

    (*
    * Both the hours and minutes fields must be non-negative.
    *)
    (duration.hours > 0) and (duration.minutes > 0)

        and

    (*
    * The maximum duration is 999 hours.
    *)
    (duration.hours + duration.minutes/60 <= 999);

    (*
    * Note that there is no specific upper bound on the minutes field.
    *)

)
description: (*
    Auxiliary function that specifies range criteria for the duration of a
    scheduled item.
*);
end;

(**
* Category-related objects and operations.
*)
object Categories is
    components: Category*;
    description: (*

        *)
end CategoryList;

operation AddCategory is
    inputs: cs:Categories, c:Category;
    outputs: cs':Categories;
    description: (*
        AddCategory adds the given Category to the given CategoryList. No
        category of the same name can be in the list already. Category names
        are compared on a character-by-character, case sensitive basis.
    *);

    precondition:
        not (exists (c' in cs) (c'.name = c.name));

    postcondition:
        AddWithRetentionAndNoJunk(c, cs, cs');

end AddCategory;

operation ChangeCategory is
    inputs: cs:Categories, c_old:Category, c_new:Category;

```

```

    outputs: cs':Categories;
    description: (*

    *);
    precondition: ;
    postcondition: ;
end ChangeCategory;

operation DeleteCategory is
    inputs: cs:Categories, c:Category;
    outputs: cs':Categories;
    description: (*

    *);
    precondition: ;
    postcondition: ;
end ChangeCategory;

end Schedule;

```

5.2. Calendar Database (caldb.rsl)

```

(****
*
* Module CalendarDB defines the major database objects of the calendar system
* as well as the objects that define the underlying calendar structure. The
* core object of the calendar system is the UserCalendar, which is modeled as
* a collection of scheduled items. The calendar structure of years, months,
* weeks, and days is not directly modeled in the UserCalendar. These
* structures are computed views that are derived from the underlying scheduled
* item collection.
*
*)

module CalendarDB;

    from Schedule import ScheduledItem, Date, Categories;
    from View import CustomLists, Filtering, WindowingMode;
    from Admin import UserDB, GroupDB, LocationDB, UserId;
    from File import FileSpace, File;
    from Edit import Clipboard, Selection, SelectionContext;
    from Options import Options;
    export CalendarDB, UserWorkspace, UserCalendar, RequiresSaving,
        PreviousState;

    object CalendarDB is
        components: user_calendars:UserCalendars and user_db:UserDB and
            group_db:GroupDB and loc_db:LocationDB and
            global_options:Options and workspace:UserWorkSpace;
        description: (*
            The CalendarDB is the top-level object for the Calendar Tool. The
            first component is the list of UserCalendars for all registered users.
            The next three components are the registered UserDB, the user GroupDB,
            and the LocationDB. The Options component is the set of calendar
            options common by default for all users, both registered and
            non-registered. The UserWorkSpace component is active calendars upon
            which the current user is working.
            *);
    end;

    object UserWorkSpace is

```

```

components: uid:UserId and calendars:UserCalendar* and
           previous_state:PreviousState and clipboard:Clipboard and
           selection:Selection and context:SelectionContext;
description: (*
    The UserWorkSpace contains the active calendars upon which the user is
    working. The first component is the UserId of the current user, which
    is used as necessary by operations that input the workspace to
    determine who the user is. The UserCalendar* component is the list of
    active calendars; the list is maintained in the order visited by the
    user, with the first element being the most recently visited, and hence
    current, and the last element being the earliest visited. The
    previous_state component is used to support one level of command undo.
    The Clipboard is used with the Edit cut, copy, and paste operations.
*);
end;

object PreviousState is
    components: ScheduledItem*;
    description: (*
        PreviousState is the snapshot of scheduled items before the most
        recently performed scheduling operation used by EditUndo. The
        Schedule, Change, and Delete operations save the previous state to
        support Undo.
    *);
end PreviousState;

object UserCalendars is
    components: UserCalendarFileName*;
    description: (*
        UserCalendars is the list of public calendars for all registered users.
        The list consists of calendar file names, not calendars themselves. In
        this way, the list of user calendars is modeled as a distributed
        collection of user calendar files, not as a centralized database of
        calendars. This allows users to retain autonomous control of their own
        individual calendars.
    *);
end UserCalendars;

object UserCalendarFileName is string
    description: (*
        A UserCalendarFileName is the unique path to a user calendar file. The
        specific syntax of such a path is implementation-platform-specific. In
        general, a file path is a sequence of file directory names, ending in a
        file name.
    *);
end UserCalendarFileName;

object UserCalendar
    components: items:ScheduledItem* and categories:Categories and
               lists:CustomLists and filtering:Filtering and options:Options and
               windowing_mode:WindowingMode and uid:UserId and file:File and
               selected_date:Date and requires_saving:RequiresSaving;
    description: (*
        A UserCalendar has a list of scheduled items and the calendar-specific
        information for categories, custom lists, filters, options, and
        windowing mode. Calendar bookkeeping components are the ID of the
        user who owns the calendar, the file it's stored on, the currently
        selected date, and a flag indicating if the calendar requires saving.
    *);
end UserCalendar;

```

```

object UserName is string
  description: (* The name of a registered user. *);
end Username;

object RequiresSaving is boolean
  description: (*
    True if a calendar requires saving, which is the case if one or more
    successful edit operations has been performed on the calendar since the
    most recent save. The calendar edit operations are Schedule, Change,
    and Delete.
  *);
end RequiresSaving;

end CalendarDB;

```

5.3. Viewing (view.rsl)

```

(****
*
* Module View defines the objects and operations related to the different
* calendar views available to the user. The structural viewing levels are
* item, day, week, month, and year. There are operations to go to the
* previous and next views at any level, as well as an operation to go to a
* specific date. Lists of scheduled items can be viewed in a variety of ways.
* A general view filter operation can be applied to both structural and list
* views. Operations are available to view other users' calendars and to view
* a list of active viewing windows.
*
* NOTE: this is work in progress. A good deal of objects are yet to be
* defined.
*)
module View;

  from CalendarDB import CalendarDB, UserCalendar, UserWorkspace;
  from Schedule import ScheduledItem, StartTime, Duration, Title, Category,
    Location, Security, Priority, Date, Hours, Minutes, DateNumber, Time,
    Hour, Minute, AmOrPm, DayName, MonthName;
  export CustomLists, Filtering, WindowingMode;

  object DailyAgenda is
    components: Date and EventTitle* and TimeSlotDescriptor* and
      TaskDescriptor*;
    description: (*
      A DailyAgenda contains scheduled item information for a calendar day.
      The Date indicates the day of the agenda. The EventTitle list contains
      the titles of zero or more events scheduled to occur on this day. The
      TimeSlotDescriptor list contains time-tagged descriptors for scheduled
      appointments and meetings. The TaskDescriptor list is a numbered list
      of tasks scheduled to be due on this day.
    *);
  end DailyAgenda;

  object EventTitle is Title
    description: (*
      The title of an event as shown in the list of events for a day.
    *);
  end;

  object TimeSlotDescriptor is
    components: TimeSlotName and BriefItemDescriptor* and Overlaps;

```

```

description: (*
    A time slot descriptor represents one slot (physically, a row) in a
    daily agenda. The TimeSlotName component is the start time for the
    slot. The list of BriefItemDescriptors contains the items that begin
    within the slot, where "within" is defined as the start time plus the
    current time increment. The overlaps component is a list of items with
    start times that overlap with an item in the BriefItemDescriptor list.
*);
end TimeSlotDescriptor;

object TimeSlotName is Time
    description: (*
        A TimeSlotName consists of a numeric TimeValue and an AmOrPm indicator.
    *);
end TimeSlotName;

object BriefItemDescriptor is
    components: Title and StartTime and Duration and Category;
    description: (*
        A brief item descriptor contains a subset of the information for a full
        scheduled item. The information is a Title, StartTime, Duration, and
        Category.
    *);
end BriefItemDescriptor;

object Overlaps is
    components: BriefItemDescriptor*;
    description: (*
        Overlaps contain zero or more BriefItemDescriptors that overlap with
        with the master item in a given time slot. An overlapping item is one
        with a StartTime within the same time slot as other items. The
        "master" item in a time slot is the item that is first in a sorted
        order based on start time, duration, and alphabetic title as the
        primary, secondary, and tertiary sort keys, respectively.
    *);
end Overlaps;

object TaskDescriptor is
    components: TaskNumber and TaskTitle;
    description: (*
        A TaskDescriptor is one item in a number list of tasks.
    *);
end TaskDescriptor;

object TaskNumber is integer
    description: (*
        The ordinal number of a task relative to other tasks.
    *);
end;

object TaskTitle is Title
    description: (*
        The free-form title of a task.
    *);
end;

object DailyFormatOptions is
    components: NormalTimeRangeOption and TimeIncrementOption and
        IncrementHeightOption and ShowHide24HoursOption and
        ShowHideExactTimeOption and ShowHideDashedLinesOption and
        ShowHideExtensionArrowsOption and ProportionalSpacingOnOffOption and

```

```

        DisplayOverlapsOption and ShowHideEventsOption and
        ShowHideTasksOption and DefaultHeightAndWidthOption;
description: (*
    DailyFormatOptions control the details of how daily agenda information
    is formatted.
*);
end DailyFormatOptions;

object NormalTimeRangeOption is StartTime and EndTime
description: (*
    NormalTimeRangeOption defines the starting and ending times that
    normally appear in the left column of the daily agenda. The default
    range of times is 8AM to 5PM.
*);
end;

object TimeIncrementOption is Hours and Minutes
description: (*
    TimeIncrementOption defines the time increment of the left column
    labels in the daily agenda. The smallest allowable time increment is
    five minutes, the largest is twelve hours. When the time increment is
    less than one hour, the time labels are shown in hour and minute
    format; when the increment is one hour or greater, the labels are shown
    as hours with no minutes. The default increment is one hour.
*);
end;

object IncrementHeightOption is integer
description: (*
    IncrementHeightOption option defines the default minimum number of text
    lines shown in each time increment of a daily agenda. If there are
    more items starting in an increment than fit in the minimum number of
    lines, the system automatically increases the height for that increment
    to a size sufficient to display all items fully. The default value is
    two lines per increment.
*);
end;

object ShowHide24HoursOption is Show or Hide
description: (*
    ShowHide24HoursOption toggles between showing a full 24 hours of
    display versus the normal time range only in daily agenda. If the
    24-hour time range is shown, non-normal times are shown with a light
    grey background, to distinguish them from normal times. The default
    value for this option is to hide non-normal times.
*);
end;

object ShowHideExactTimeOption is Show or Hide
description: (*
    ShowHideExactTimeOption toggles between showing or not showing the
    starting and ending times for each scheduled item immediately to the
    left of the title in a daily agenda. This option is useful when items
    do not typically start or end exactly on the hour. The default value
    for this option is to hide the exact time.
*);
end;

object ShowHideDashedLinesOption is Show or Hide
description: (*
    ShowHideDashedLinesOption toggles between showing and not showing the

```

```
        dashed lines that appear above and below items that do not start or end
        on the hour in a daily agenda. The default is to show dashed lines.
    *);
end;

object ShowHideExtensionArrowsOption is Show or Hide
description: (*
    ShowHideExtensionArrowsOption toggles between showing and not showing
    the down-pointing arrows for items with end times that extend beyond
    the hour of the start time in a daily agenda. The default is to show
    extension arrows.
    *);
end;

object ProportionalSpacingOnOffOption is On or Off
description: (*
    ProportionalSpacingOnOffOption toggles between placing items
    proportionally down from the top of the hour based on starting time in
    a daily agenda. When the option is off, the first (only) item in an
    hour is placed at the top of that hour and any other items starting in
    the hour are placed immediately below it. The default is proportional
    spacing on.
    *);
end;

object DisplayOverlapsOption is Horizontal or Vertical
description: (*
    DisplayOverlapsOption controls how items with overlapping times are
    displayed in a daily agenda. When displayed horizontally, overlapping
    items appear side-by-side in separate columns of the day view. When
    displayed vertically, overlapping items appear on separate lines in a
    single column, sorted by start time, duration, and title. The default
    is horizontal display.
    *);
end;

object ShowHideEventsOption is Show or Hide
description: (*
    ShowHideEventsOption toggles between showing or not showing the list of
    events in a daily agenda. The default is to show events if any are
    scheduled.
    *);
end;

object ShowHideTasksOption is Show or Hide
description: (*
    ShowHideTasksOption toggles between showing or not showing the list of
    tasks in a daily agenda. The default is to show tasks if any are
    scheduled.
    *);
end;

object DefaultHeightAndWidthOption is Height and Width
description: (*
    DefaultHeightAndWidthOption controls the width and height of the full
    daily agenda. If one or both of the default dimensions is too small to
    accommodate all information in the display, the information is
    truncated appropriately and scrollbars are added to the display window.
    *);
end;
```

```
object EndTime is TimeSlotName
  description: (*
    Ending time in the normal time range option.
  *);
end;

object Show
  description: (*
    Option indicator for a show/hide option, indicating that the option
    state is "show".
  *);
end;

object Hide
  description: (*
    Option indicator for a show/hide option, indicating that the option
    state is "hide".
  *);
end;

object On
  description: (*
    Option indicator for an on/off option, indicating that the option state
    is "on".
  *);
end;

object Off
  description: (*
    Option indicator for an on/off option, indicating that the option state
    is "off".
  *);
end;

object Horizontal
  description: (*
    Option indicator for displaying overlaps horizontally.
  *);
end;

object Vertical
  description: (*
    Option indicator for displaying overlaps vertically.
  *);
end;

object Height is integer
  description: (*
    Height value for the default height and width option.
  *);
end;

object Width is integer
  description: (*
    Width value for the default height and width option.
  *);
end;

object WeeklyAgendaTable is
  components: ;
  description: (*
```



```

    *);
end WeeklyAgendaTable;

object WeeklyAgendaList is
  components: ;
  description: (*

    *);
end WeeklyAgendaList;

object MonthlyAgenda is
  components: FullMonthName and FirstDay and NumberOfDays and SmallDayView*;
  description: (*
    A MonthlyAgenda contains a small daily view for each day of the month,
    organized in the fashion typical in paper calendars.
  *);
end MonthlyAgenda;

object FullMonthName is MonthName and Year;
object Year is integer;
object FirstDay is DayName;
object NumberOfDays is integer;
object SmallDayView is
  components: DateNumber and BriefItemDescriptor*;
  description: (*
    A SmallDayView has the number of the date and a list of zero or more
    brief item descriptors.
  *);
end SmallDayView;

object YearlyCalendar is
  components: ;
  description: (*

    *);
end YearlyCalendar;

operation ViewItem(CalendarDB)->ScheduledItem;
operation ViewDay(cdb:CalendarDB)->agenda:DailyAgenda
  postcondition:
    (*
      * The output is the agenda is for the currently selected date, if any,
      * today's date otherwise.
    *)
  ;
  description: (*
    ViewDay produces a DailyAgenda for the currently selected date in the
    given calendar DB, or for today's date if no other date is currently
    selected.
  *);
end ViewDay;

operation ViewWeekTable(CalendarDB)->WeeklyAgendaTable;
operation ViewWeekList(CalendarDB)->WeeklyAgendaList;
operation ViewMonth(CalendarDB)->MonthlyAgenda
  description: (*
    The output is taken both from the user's own calendar as well as the
    system version of the user's calendar when the latter contains
    penciled-in meetings.
  *);
end;

```

```

operation ViewYear(CalendarDB)->YearlyCalendar;

operation GotoNext(CalendarDB)->ViewLevel;
operation GotoPrevious(CalendarDB)->ViewLevel;
operation GotoDate(CalendarDB)->ViewLevel;

operation ViewAppointmentsList(CalendarDB)->AppointmentsList;
operation ViewMeetingsList(CalendarDB)->MeetingsList;
operation ViewTasksList(CalendarDB)->TasksList;
operation ViewEventsList(CalendarDB)->EventsList;
operation ViewAllItemsList(CalendarDB)->AllItemsList;

operation ViewOtherUser(CalendarDB)->UserCalendar;

operation ViewWindows(CalendarDB)->WindowNameList;

object ViewLevel is ScheduledItem or DailyAgenda or WeeklyAgendaTable or
    WeeklyAgendaList or MonthlyAgenda or YearlyCalendar;

object AppointmentsList is AppointmentListItem*
  description: (*
    An AppointmentsList is the collection of items that appear in the list
    view of appointments.
  *);
end;

object MeetingsList is MeetingsListItem*
  description: (*
    A MeetingsList is the collection of items that appear in the list
    view of meetings.
  *);
end;

object TasksList is TasksListItem*
  description: (*
    A TasksList is the collection of items that appear in the list
    view of tasks.
  *);
end;

object EventsList is EventsListItem*
  description: (*
    An EventsList is the collection of items that appear in the list
    view of events.
  *);
end;

object AllItemsList is ListItem*
  description: (*
    An AllItemsList is the collection of items that appear in the list
    view of all scheduled items.
  *);
end;

object AppointmentListItem is
  components: Title and Date and Time and Duration and Recurs and
    Category and Location and Security and Priority;
  description: (*
    An AppointmentListItem is the somewhat abbreviated description of an
    appointment that appears in appointments and meetings list views. The
    precise relation between a full Appointment compared to an

```

AppointmentListItem is as follows:

```

<ul><li>
  The titles are the same.
  <li>
    The list item has only a start date, with no end date. This is
    because there is a separate list item for every instance of a
    recurring appointment.
    <li>
      The start time and duration are the same.
      <li>
        There is only a boolean-valued Recurs component in the list item,
        instead of the full recurring structure of the Appointment. The
        rationale is that the list would be too cluttered if it contained
        all of the recurring information of the full appointment.
        <li>
          The category, location, security, and priority are the same.
          <li>
            The list item contains no remind information or details. The
            rationale is again not to clutter the list.
          </ul>
        *);
end AppointmentListItem;

object Recurs is boolean
  description: (*
    Indicates in an AppointmentListItem whether the appointment recurs.
  *);
end Recurs;

object WindowNameList is WindowName*;
object WindowName is string;

operation SelectDate(cdb:CalendarDB, date:Date)->(cdb':CalendarDB)

  postcondition:
    (*
      * The selected date in the current calendar is set to the given date.
    *)
    cdb'.workspace.calendars[1].selected_date = date;

  description: (*
    Select a date in the current calendar. The most typical reason for
    date selection is as the argument to a view command.
  *);

end SelectDate;

(*
* A major bit of aux function spec will be opaque functions that define the
* externally defined day/date computations.
*)

object CustomLists is ... ;

object Filtering is ... ;

object WindowingMode is 'TwoWindow' or 'PerLevel' or 'MultiWindow';

end View;
```

5.4. Administration (admin.rsl)

```
(****
*
* Module Admin defines the objects and operations related to maintaining the
* user, group, location, and global options databases of the Calendar Tool.
*
*)
module Admin;

    from Schedule import Title;
    export UserDB, GroupDB, LocationDB, UserId;

    object UserDB is
        components: UserRecord*;
        description: (*
            UserDB is the repository of registered user information. It is a
            collection of UserRecords.
        *);
    end UserDB;

    object UserRecord is
        components: name:Name and id:UserId and
            phone:PhoneNumber and email:EmailAddress and host:HostComputer and
            uid:ComputerUserID and home_dir:HomeDirectory and
            item_limit:ItemLimit and privileges:AdminPrivileges;
        description: (*
            A UserRecord is the information stored about a registered user of the
            Calendar Tool. The Name component is the user's real-world name. The
            nickname is an individualizable short-hand name for the user. The
            UserId is the unique identifier by which the user is known to the
            Calendar Tool. The EmailAddress is the electronic mail address used by
            the Calendar Tool to contact the user when necessary. The PhoneNumber
            is for information purposes; it is not used by the Calendar Tool for
            contacting the user.
        *);
    end UserRecord;

    object Name is string
        description: (* The name of a registered user. *);
    end;

    object UserId is string
        description: (* The unique id of a registered user. *);
    end;

    object EmailAddress is string
        description: (* The electronic mail address of a registered user. *);
    end;

    object PhoneNumber is area:Area and number:Number
        description: (*
            A phone number consists of a three-digit area code and seven-digit
            number.
        *);
    end;

    object Area is integer
        description: (*
            The three-digit area code component of a user phone number.
        *);
```

```
end Area;

object Number is integer
  description: (*
    The seven-digit number component of a user phone number.
  *);
end Number;

object HostComputer is string
  description: (*
    The unique identifier of the host computer on which the user runs the
    Calendar Tool. The format of the identifier is platform-dependent,
    based on the type of network to which host computers are attached. IP
    address is a typical format.
  *);
end HostComputer;

object ComputerUserID is string
  description: (*
    The user's identification on the host computer.
  *);
end ComputerUserID;

object HomeDirectory is string
  description: (*
    The file directory (i.e., folder) assigned as the user's home directory
    on the host computer.
  *);
end HomeDirectory;

object ItemLimit is integer
  description: (*
    The maximum number of scheduled items that can be stored for the user
    in the Calendar Tool central repository. The limit value is optional.
    If it is absent, there is no limit. The item limit can be zero, in
    which case the user has an empty calendar in the central repository.
  *);
end ItemLimit;

object AdminPrivileges is boolean
  description: (*
    Indicates if the user has administrative privileges.
  *);
end AdminPrivileges;

operation AddUser is
  inputs: udb:UserDB, ur:UserRecord;
  outputs: udb':UserDB;
  description: (*
    Add the given UserRecord to the given UserDB. The UserId of the given
    user record must not be the same as a user record already in the
    UserDB. The UserId component is required and must be eight characters
    or less. The email address is required. The phone number is optional;
    if given, the area code and number must be 3 and 7 digits respectively.
  *);

precondition:
  (*
  * There is no user record in the input UserDB with the same id as the
  * record to be added.
  *)
```

```

(not (exists (ur' in udb) ur'.id = ur.id))

    and

(*
 * The id of the given user record is not empty and 8 characters or
 * less.
 *)
(ur.id != nil) and (#(ur.id) <= 8)

    and

(*
 * The email address is not empty.
 *)
(ur.email != nil)

    and

(*
 * If the phone area code and number are present, they must be 3 digits
 * and 7 digits respectively.
 *)
(if (ur.phone.area != nil) then (#(ur.phone.area) = 3)) and
(if (ur.phone.number != nil) then (#(ur.phone.number) = 7));

postcondition:
(*
 * A user record is in the output db if and only if it is the new
 * record to be added or it is in the input db.
 *)
forall (ur':UserRecord)
    (ur' in udb') iff ((ur' = ur) or (ur' in udb));

end AddUser;

operation FindUser is
inputs: udb:UserDB, id:UserId;
outputs: ur':UserRecord;
description: (*
    Find a user by unique id.
 *);

precondition: ;

postcondition:
(*
 * If there is a record with the given id in the input db, then the
 * output record is equal to that record, otherwise the output record
 * is empty.
 *)
(exists (ur in udb) (ur.id = id) and (ur' = ur))
    or
(not (exists (ur in udb) (ur.id = id)) and (ur' = nil));

end FindUser;

operation FindUser is
inputs: udb:UserDB, n:Name;
outputs: url:UserRecord*;
description: (*

```

```

        Find a user or users by real-world name.  If more than one is found,
        the output list is sorted by id.
    *);

precondition: ;

postcondition:
    (*
     * The output list consists of all records of the given name in the
     * input db.
     *)
    (forall (ur' in url) (ur' in udb) and (ur'.name = n))

        and

    (*
     * The output list is sorted alphabetically by id.
     *)
    (forall (i:integer | (i >= 1) and (i < #url))
        url[i].id < url[i+1].id);

end FindUser;

operation FindUser is
    inputs: udb:UserDB, id:UserId, n:Name;
    outputs: ur':UserRecord;
    description: (*
        Find a user by both name and id.  This overload of FindUser is
        presumably used infrequently.  Its utility is to confirm that a
        particular user name and id are paired as assumed.
    *);

precondition: ;

postcondition:
    (*
     * If there is a record with the given name and id in the input db,
     * then the output record is equal to that record, otherwise the output
     * record is empty.
     *)
    (exists (ur in udb) (ur.name = n) and (ur.id = id) and (ur' = ur))
        or
    (not (exists (ur in udb) (ur.name = n) and (ur.id = id)) and
        (ur' = nil));

end FindUser;

operation ChangeUser is
    inputs: udb:UserDB, gdb:GroupDB, old_ur:UserRecord, new_ur:UserRecord;
    outputs: udb':UserDB, gdb':GroupDB;
    description: (*
        Change the given old UserRecord to the given new record.  The old and
        new records must not be the same.  The old record must already be in
        the input db.  The new record must meet the same conditions as for the
        input to the AddUser operation.  Typically the user runs the FindUser
        operation prior to Change to locate an existing record to be changed.
        <p>
        If the user record id is changed, then change all occurrences of the
        old id in the group db to the new id.
    *);

```

```

precondition:
  (*
   * The old and new user records are not the same.
   *)
  (old_ur != new_ur)

  and

  (*
   * The old record is in the given db.
   *)
  (old_ur in udb)

  and

  (*
   * There is no user record in the input UserDB with the same id as the
   * new record to be added.
   *)
  (not (exists (new_ur' in udb) new_ur'.id = new_ur.id))

  and

  (*
   * The id of the new record is not empty and 8 characters or less.
   *)
  (new_ur.id != nil) and (length(new_ur.id) <= 8)

  and

  (*
   * The email address is not empty.
   *)
  (new_ur.email != nil)

  and

  (*
   * If the phone area code and number are present, they must be 3 digits
   * and 7 digits respectively.
   *)
  (if (new_ur.phone.area != nil) then (length(new_ur.phone.area) = 3)) and
  (if (new_ur.phone.number != nil) then (length(new_ur.phone.number) = 7));

postcondition:
  (*
   * A user record is in the output db if and only if it is the new
   * record to be added or it is in the input db, and it is not the old
   * record.
   *)
  forall (ur':UserRecord)
    (ur' in udb') iff ((ur' = new_ur) or (ur' in udb)) and
    (ur' != old_ur)

  and

  (*
   * If new id is different than old id, then all occurrences of old id
   * in the GroupDB are replaced by new id.
   *)
  if (old_ur.id != new_ur.id)

```



```

    then
        ... (* Logic left as exercise for the reader. *) ;
end ChangeUser;

operation DeleteUser is
    inputs: udb:UserDB, gdb:GroupDB, ur:UserRecord;
    outputs: udb':UserDB, gdb':GroupDB, lgw:LeaderlessGroupsWarning;
    description: (*
        Delete the given user record from the given UserDB.  The given record
        must already be in the input db.  Typically the user runs the FindUser
        operation prior to Delete to locate an existing record to delete.
        <p>
        In addition, delete the user from all groups of which the user is a
        member.  If the deleted user is the only leader of a one more groups,
        output a warning indicating that those groups have become leaderless.
    *);

precondition:
    (*
    * The given UserRecord is in the given UserDB.
    *)
    ur in udb;

postcondition:
    (*
    * A user record is in the output db if and only if it is not the
    * existing record to be deleted and it is in the input db.
    *)
    (forall (ur':UserRecord)
        (ur' in udb') iff ((ur' != ur) and (ur' in udb)))

        and

    (*
    * The id of the deleted user is not in the leader or member lists of
    * any group in the output GroupDB.  (NOTE: This clause is not as
    * strong as a complete "no junk, no confusion" spec.  Why not?  Should
    * it be?)
    *)
    (forall (gr in gdb')
        (not (ur.id in gr.leaders)) and (not (ur.id in gr.members)))

        and

    (*
    * The LeaderlessGroupsWarning list contains the ids of all groups
    * whose only leader was the user who has just been deleted.
    *)
    (forall (gr in gdb)
        forall (id:UserId)
            (id in lgw) iff ((#(gr.leaders) = 1) and
                (gr.leaders[1] = ur.id)));

end DeleteUser;

object LeaderlessGroupsWarning is
    components: Name*;
    description: (*
        LeaderlessGroupsWarning is an secondary output of the Change and
        DeleteUser operations, indicating the names of zero or more groups that

```

```

        have become leaderless as the result of a user having been deleted.
    *);
end LeaderlessGroupsWarning;

object GroupDB is
  components: GroupRecord*;
  description: (*
    UserDB is the repository of user group information.
  *);
end GroupDB;

object GroupRecord is
  components: name:Name and id:GroupId and leaders:Leaders and
    members:Members;
  description: (*
    A GroupRecord is the information stored about a user group. The Name
    component is a unique group name of any length. Leaders is a list of
    zero or more users designated as group leader. Members is the list of
    group members, including the leaders. Both lists consist of user id's.
    Normally a group is required to have at least one leader. The only
    case that a group becomes leaderless is when its leader is deleted as a
    registered user.
  *);
end GroupRecord;

object GroupId is string
  description: (* The unique id of a registered group. *);
end GroupId;

object Leaders is UserId*
  description: (*
    List of user ids for the zero or more leaders of a group.
  *);
end;

object Members is UserId*
  description: (*
    List of user ids for the zero or more members of a group.
  *);
end;

operation AddGroup is
  inputs: gdb:GroupDB, udb:UserDB, gr:GroupRecord;
  outputs: gdb':GroupDB;
  description: (*
    Add the given GroupRecord to the given GroupDB. The name of the given
    group must not be the same as a group already in the GroupDB. All
    group members must be registered users. The leader(s) of the group
    must be members of it.
  *);

precondition:
  (*
    * All group members are registered users.
  *)
  (forall (id in gr.members) exists (ur in udb) ur.id = id)

  and

  (*
    * All group leaders are members of the group.
  *)

```

```

    *)
    (forall (id in gr.leaders) id in gr.members);

postcondition:
  (*
   * A group record is in the output db if and only if it is the new
   * record to be added or it is in the input db.
   *)
  forall (gr':GroupRecord)
    (gr' in gdb') iff ((gr' = gr) or (gr' in gdb));

end AddGroup;

operation FindGroup is
  inputs: gdb:GroupDB, n:Name;
  outputs: gr':GroupRecord;
  description: (*
    Find a group by unique name.
  *);

precondition: ;

postcondition:
  (*
   * If there is a record with the given name in the input db, then the
   * output record is equal to that record, otherwise the output record
   * is empty.
   *)
  (exists (gr in gdb) (gr.name = n) and (gr' = gr))
  or
  (not (exists (gr in gdb) (gr.name = n)) and (gr' = nil));

end FindGroup;

operation ChangeGroup is
  inputs: gdb:GroupDB, udb:UserDB, old_gr:GroupRecord, new_gr:GroupRecord;
  outputs: gdb':GroupDB;
  description: (*
    Change the given old GroupRecord to the given new record. The old and
    new records must not be the same. The old record must already be in
    the input db. The new record must meet the same conditions as for the
    input to the AddGroup operation. Typically the user runs the FindGroup
    operation prior to Change to locate an existing record to be changed.
  *);

precondition:
  (*
   * The old and new group records are not the same.
   *)
  (old_gr != new_gr)

  and

  (*
   * All group members are registered users.
   *)
  (forall (id in new_gr.members) exists (ur in udb) ur.id = id)

  and

  (*

```

```

    * All group leaders are members of the group.
    *)
    (forall (id in new_gr.leaders) id in new_gr.members);

postcondition:
  (*
   * A group record is in the output db if and only if it is the new
   * record to be added or it is in the input db, and it is not the old
   * record.
   *)
  forall (gr':GroupRecord)
    (gr' in gdb') iff (((gr' = new_gr) or (gr' in gdb)) and
      (gr' != old_gr));

end ChangeGroup;

operation DeleteGroup is
  inputs: gdb:GroupDB, gr:GroupRecord;
  outputs: gdb':GroupDB;
  description: (*
    Delete the given group record from the given GroupDB. The given record
    must already be in the input db. Typically the user runs the FindGroup
    operation prior to Delete to locate an existing record to delete.
  *);

precondition:
  (*
   * The given GroupRecord is in the given GroupDB.
   *)
  gr in gdb;

postcondition:
  (*
   * A group record is in the output db if and only if it is not the
   * existing record to be deleted and it is in the input db.
   *)
  (forall (gr':GroupRecord)
    (gr' in gdb') iff ((gr' != gr) and (gr' in gdb)));

end DeleteGroup;

object LocationDB is
  components: LocationRecord*;
  description: (*
    The LocationDB contains the location records that provide information
    about the locations at which items are scheduled.
  *);
end LocationDB;

object LocationRecord is
  components: Name and Number and Uses and Remarks;
  description: (*
    A LocationRecord has a Name and Number which serve to identify where
    the location is. Both fields are free-form strings and the Calendar
    Tool enforces no constraints on their values. The Bookings component
    is a list of the titles of the items that are scheduled in the
    location. The Remarks component is a free-form text that can be used
    to describe any other pertinent information about the room.
  *);
end LocationRecord;

```

```

obj Bookings is ... ;
obj Remarks is ... ;

(*
 * Global options TBD.
 *)

end Admin;

```

5.5. Options (options.rsl)

```

module Options;

  export Options;

  object Options is
    components: TimesAndDates and Fonts and Scheduling and Viewing and
      Administrative;
    description: (*

      *);
  end Options;

  object TimesAndDates is
    components: TimeDisplayFormat and DateDisplayFormat and DateSample;
    description: (*

      *);
  end TimesAndDates;

  object TimeDisplayFormat is
    components: 'TwelveHour' or 'TwentyFourHour';
    description: (*

      *);
  end TimeDisplayFormat;

  object DateDisplayFormat is
    components: prefix:Prefix and p1:Position and s1:Separator and
      p2:Position and s2:Separator and p3:Position and s3:Separator and
      p4:Position and s4:Separator;
    description: (*

      *);
  end DateDisplayFormat;

  obj Prefix is string;
  obj Separator is string;
  object Position is
    components: 'DayOfWeekFull' or 'DayOfWeekThreeChar' or
      'DayOfWeekOneOrTwoChar' or 'DateNumber' or 'MonthFullName' or
      'MonthShortName' or 'MonthNumber' or 'YearFourDigits' or
      'YearTwoDigits';
    description: (*

      *);
  end Position;

  obj DateSample is string;

  object Fonts is
    components: UserEnteredData and HeadersAndLabels;

```

```

    description: (*
        *);
end Fonts;

object UserEnteredData is FontOption
    description: (*
        *);
end UserEnteredData;

object HeadersAndLabels is FontOption
    description: (*
        *);
end HeadersAndLabels;

object FontOption is
    components: FontName and Style and Size and DisplayType;
    description: (*
        *);
end FontOption;

obj FontName is string;
obj Style is string;
obj Size is integer;
obj DisplayType is 'Item' or 'Day' or 'Week' or 'Month' or 'Year' or
    'Lists' or 'Dialogs' or 'AllDisplays';

obj Scheduling;
obj Viewing;
obj Administrative;

end Options;

```

5.6. File Handling (file.rsl)

```

(****
*
* Module File defines the objects and operations related to file processing
* in the calendar system.
*
*)

module File;

    from CalendarDB import CalendarDB, UserCalendar, UserWorkSpace,
        RequiresSaving;
    from Admin import UserId;
    from Options import Options;
    export FileSpace, File;

    object FileSpace is File*
        description: (*
            A FileSpace is an abstract model of a file space in the operating
            environment in which the CalendarTool is run. The FileSpace is simply
            a collection of zero or more Files, with no other properties modeled
            here.
            *);
    end;
end;

```

```

object File is
  components: name:FileName and permissions:FilePermissions and
    file_type:FileType and data:FileData;
  description: (*
    A File is an abstraction of a file stored in the file space. It has a
    name, permissions, type, and data. These are the components sufficient
    to specify the behavior of Calendar Tool file operations.
  *);
end File;

object FileName is string
  description: (*
    The name of a file. The string representation here is an abstraction
    of file names used in specific operating environments. Implementations
    may obey any syntactic or semantic constraints imposed by a particular
    environment.
  *);
end;

object FilePermissions is is_readable:IsReadable and is_writable:IsWritable
  description: (*
    FilePermissions indicate whether a file is readable and/or writable.
  *);
end;

object IsReadable is boolean
  description: (*
    Flag indicating whether a file is readable, which is required to be
    true by the FileOpen operation.
  *);
end;

object IsWritable is boolean
  description: (*
    Flag indicating whether a file is writable, which is required to be
    true by the FileSave operation.
  *);
end;

object FileType is calendar_type:CalendarType or other_type:OtherType
  description: (*
    The type of file data is either CalendarType data (which we care about)
    or any other type of data that can be stored in a file, which we don't
    care about.
  *);
end FileType;

object CalendarType
  description: (*
    File data typing tag indicating that a file contains calendar data
    created by the Calendar Tool
  *);
end CalendarType;

object OtherType
  description: (*
    File data typing tag indicating that a file contains data other than
    calendar data created by the Calendar Tool
  *);
end OtherType;

```

```

object FileData is UserCalendar
  description: (*
    The abstract representation of calendar-type FileData is a UserCalendar
    object. Calendar Tool implementors may use any concrete file data
    representation that accurately holds all UserCalendar components.
  *);
end FileData;

operation FileNew is
  inputs: uws:UserWorkSpace;
  outputs: uws':UserWorkSpace;

  description: (*
    Add a new empty calendar to the workspace and make it current.
  *);

  precondition: ;

  postcondition:
    (*
      * The output workspace has a new empty calendar and that calendar is
      * current. The user id of the new calendar is that of the workspace,
      * the file is empty, the options are the given global options input,
      * and the calendar does not require saving. The calendars in
      * positions 1-last in the the input workspace are in positions
      * 2-last+1 in the output workspace.
    *)
    (exists (uc:UserCalendar)
      (uc = uws'.calendars[1]) and
      (uc.uid = uws.uid) and
      (uc.file = nil) and
      (not uc.requires_saving) and
      (uc.items = nil) and
      (#(uws'.calendars) = #(uws.calendars) + 1) and
      (forall (i:integer | (i >= 2) and (i <= #(uws.calendars)))
        uws'.calendars[i+1] = uws.calendars[i+1]
      )
    );
end FileNew;

operation FileOpen is
  inputs: fs:FileSpace, fn:FileName, uws:UserWorkSpace;
  outputs: fs':FileSpace, uws':UserWorkSpace;

  description: (*
    Open an existing calendar file of the given name and put the data from
    that file in the workspace.
  *);

  precondition:
    (*
      * A file of the given name exists in the given file space, the file
      * is readable, and the file's data are of type calendar.
    *)
    exists (file in fs)
      (file.name = fn) and
      file.permissions.is_readable and
      file.file_type?calendar_type;

```



```

postcondition:
  (*
    * The output workspace has a new calendar containing the file data of
    * the input file, and that calendar is current. The user id of the
    * new calendar is that of the workspace, the options are the given
    * global options input, and the calendar does not require saving. The
    * calendars in positions 1-last in the the input workspace are in
    * positions 2-last+1 in the output workspace.
    *)
  (exists (uc:UserCalendar)
    (uc = uws'.calendars[1]) and
    (uc.uid = uws.uid) and
    (exists (file in fs)
      (file.name = fn) and
      (uc.file = file) and
      (uc.items = file.data.items)
    ) and
    (not uc.requires_saving) and
    (#(uws'.calendars) = #(uws.calendars) +1) and
    (forall (i:integer | (i >= 2) and (i <= #(uws.calendars)))
      uws'.calendars[i+1] = uws.calendars[i+1]
    )
  );

end FileOpen;

operation FileClose is
  inputs: fs:FileSpace, uws:UserWorkSpace;
  outputs: fs':FileSpace, uws':UserWorkSpace;

  description: (*
    Close the current calendar if it does not require saving.
    *);

  precondition:
    (*
      * The calendar does not require saving.
      *)
    not (uws.calendars[1].requires_saving);

  postcondition:
    (*
      * The current calendar is deleted from the workspace. The remaining
      * calendars, if any, are shifted in position in the list one position
      * earlier.
      *)
    (not (uws.calendars[1] in uws'.calendars)) and
    (#(uws'.calendars) = #(uws.calendars) - 1) and
    (forall (i:integer | (i >= 1) and (i < #(uws.calendars)))
      uws'.calendars[i] = uws.calendars[i+1];
    );

end FileClose;

operation FileSave is
  inputs: fs:FileSpace, uws:UserWorkSpace;
  outputs: fs':FileSpace, uws':UserWorkSpace;

  description: (*
    If the calendar in the given workspace requires saving, save it in the

```

```

    given file space.
*);

precondition:
  (*
  * The given workspace requires saving. Also, there is a writable
  * file of the current workspace filename in the given FileSpace. Note
  * that the only way the current file could be unwritable is through an
  * external change to the file space since the file was opened by the
  * Calendar Tool. Note further that this precondition disallows the
  * case where the current calendar file has been externally deleted
  * since it was opened by the calendar tool. That is, the file must
  * both exist and be writable at the time the save is attempted.
  *)
  (uws.calendars[1].requires_saving)

    and

  (exists (file in fs)
    (file.name = uws.calendars[1].file.name) and
    (file.permissions.is_writable));

postcondition:
  (*
  * There is a calendar-type file in the resulting FileSpace containing
  * the current workspace calendar as its file data. In the resulting
  * workspace, the requires saving indicator and is false.
  *)
  (exists (file in fs')
    (file.name = uws'.calendars[1].file.name) and
    (file.permissions.is_writable) and
    (file.file_type?calendar_type) and
    (not uws'.calendars[1].requires_saving)
  );
end FileSave;

end File;

```

5.7. Data Editing (edit.rsl)

```

(****
*
* Module Edit defines the objects and operations related to generic editing
* in the calendar system.
*
*)
module Edit;

  from CalendarDB import UserWorkSpace, UserCalendar, PreviousState;
  export Clipboard, Selection, SelectionContext;

  object Clipboard is
    components: string;
    description: (*
      The clipboard holds a selection of cut or copied text.
    *);
  end Clipboard;

  object Selection is
    components: start_position:integer and end_position:integer and
      context:string;

```

```

description: (*
    The workspace text selection is the defined as the starting and ending
    character positions in current workspace text context.
*);
end Selection;

object SelectionContext is
    components: string;
    description: (*
        SelectionContext is the text context in which the user makes a
        selection.
    *);
end SelectionContext;

operation EditUndo is
    inputs: uws:UserWorkSpace;
    outputs: uws':UserWorkSpace;

    description: (*
        Undo the most recent scheduling operation. Only one level of undo/redo
        is specified here. Successive invocations of undo toggle between the
        current and previous states of the calendar. Note that the calendar
        editing operations support undo by saving a snapshot of the input
        calendar in the previous_state component of the workspace.
    *);

    precondition:
        (*
            * The previous saved calendar state is not nil.
        *)
        (uws.previous_state != nil);

    postcondition:
        (*
            * The state of the current calendar is the previous state, and vice
            * versa.
        *)
        (uws'.previous_state = uws.calendars[1].items) and
        (uws.calendars[1].items = uws.previous_state);

end EditUndo;

operation EditCut is
    inputs: uws:UserWorkSpace;
    outputs: uws':UserWorkSpace;

    description: (*
        The currently selected text segment is copied into the clipboard and
        removed from its context. The workspace selection in set to empty.
    *);

    precondition:
        (*
            * The selection is not empty.
        *)
        uws.selection != nil;

    postcondition:
        (*
            * The clipboard of the output workspace equals the selection. The
            * selection context of the output workspace has the selection removed.
        *)

```

```

    * The selection of the output workspace is nil.
    *)
    (uws'.clipboard = uws.context[
      uws.selection.start_position:uws.selection.end_position])

      and

    (uws'.context = uws.context[1:uws.selection.start_position-1] +
      uws.context[uws.selection.start_position+1:
        #(uws.context)])

      and

    (uws'.selection = nil);

end EditCut;

operation EditCopy is
  inputs: uws:UserWorkSpace;
  outputs: uws':UserWorkSpace;

  description: (*
    The currently selected text segment is copied into the clipboard;
  *);

  precondition:
    (*
    * The selection is not empty.
    *)
    uws.selection != nil;

  postcondition:
    (*
    * The clipboard of the output workspace equals the selection. The
    * context and selection of the output workspace are unchanged.
    *)
    (uws'.clipboard = uws.context[
      uws.selection.start_position:uws.selection.end_position])

      and

    (uws'.context = uws.context)

      and

    (uws'.selection = uws.selection);

end EditCopy;

operation EditPaste is
  inputs: uws:UserWorkSpace;
  outputs: uws':UserWorkSpace;

  description: (*
    Paste the contents of the clipboard into the currently selected
    start position, replacing any selected text from start to end position.
  *);

  precondition:
    (*
    * The clipboard is not empty.

```

```

    *)
    uws.clipboard != nil;

postcondition:
  (*
   * The context in the output workspace is the string consisting of
   * everything up to the selection, followed by the clipboard, followed
   * by everything after the selection. The selection of the output
   * workspace is nil and the clipboard is unchanged.
   *)
  (uws'.context = uws.context[1:uws.selection.start_position-1] +
    uws.clipboard +
    uws.context[uws.selection.start_position+1:
      #(uws.context)])

    and

  (uws'.selection = nil)

    and

  (uws'.clipboard = uws.clipboard);

end EditPaste;

operation EditDelete is
  inputs: uws:UserWorkSpace;
  outputs: uws':UserWorkSpace;

  description: (*
    The currently selected text is removed from the context. The workspace
    selection is set to empty.
  *);

  precondition:
    (*
     * The selection is not empty.
     *)
    uws.selection != nil;

  postcondition:
    (*
     * The selection context of the output workspace has the selection
     * removed. The selection of the output workspace is nil.
     *)
    (uws'.context = uws.context[1:uws.selection.start_position-1] +
      uws.context[uws.selection.start_position+1:
        #(uws.context)])

      and

    (uws'.selection = nil);

end EditDelete;

end Edit;
```

6. Rationale

6.1. Nuked from Admin; Presumably to Go In Some Form in Installation

sh 3 "Initial System Configuration"

12aug02 Update: Think about moving the last paragraph above here. Or perhaps it belongs in the installation section, with only a ref to it here.

29jul02 Update: I think this section stays, in addition to the high-level installation section. An important point (if not the point) of a cal tool admin doing a config is to coordinate the cal tool app's hard-wired locale of a user's cal tool dir with the convention for it to use in users' cal tool admin user db records.

3jul02 Note: This section may go in favor of the higher-level installation section that's now been added. Still need to work this out fully.

Talk about what things look like straight out of the box to the admin and what needs to be set up. Constrain this to the essentially negligible set up required for non-admin users.

Need to be able to set default for calendar-host mapping, for convenience of the user. This way, when it comes up out of the shrink wrap, it'll be config

6.2. Fodder from Window Viewing Section, Ca. Aug 02

NOTE prime: OK, it looks like we're going (back) to the rule that a calendar cannot be windowless. This avoids the problem of what window should be current when a windowless calendar is current. This decision obsoletes all of the discussion notes that follow that refer to a solution that assumes windowless calendars are legal. Further, this decision lets us leave the chronological current-on-top layout of the windows submenu as is, as well as leaving the related scenario narrative pretty much, if not entirely in tact. (The bottom line for the "But farg" that follow is that we'll have both windows and cal lists chrono sorted.) But farg, we need to reconcile the chrono-sorted windows list with the alpha-sorted calendar list. I'm inclined at this point to go with alpha sorting for both, given that things seem a bit twitchy with all the movement in the chrono-sorted list. But farg again, the alpha sorting of the windows lists does not give any chrono information. So, the two kinds of sorting (chrono for windows, alpha for cal) may be OK, given that chrono for cal is a bit funky to determine, since it could be "most recently open" or "most recently active based on window selection". But farg one more time, I think we can define chrono order for cal just fine, it being "most recently visited (i.e., current)" based on window order. I.e., the chrono order of the cal can be defined as a suborder of the cal in the windows list.

NOTE: The new deal is that when a calendar has no active window, the menubar window should be current when the user selects that calendar. This means that we need to adjust our thinking about "current" versus "has focus" in terms of cal tool windows. The (mini) problem is that the command window gets focus a lot, whenever the user clicks on it to run a command, however when this happens, we don't really want to move it to the top of the windows list. Also, even when the menu window is current, there's still the notion of a current calendar, the open window for which is current. Hmm, maybe we need to separate "current window" from "current calendar" even more clearly than we've done so far. Another way to help deal with the issue is to list windows alphabetically in the windows list instead of chronologically, with the current window checked in lieu of it being at the top of the list. Well, the problem with this is that if we consider the menu to be a window, then technically it should always be checked, since it's always physically current whenever the user runs the View->Windows command. Farg.

OK, here's the conundrum we're dealing with: We don't want to make the menubar window current unless there's a windowless open calendar. OK, how about we go back to the rule that windowless calendars are disallowed, or more precisely, view-windowless calendars are disallowed. (It really doesn't have to be view-windowless, since the user could open a view, open a schedule dialog, then close the view, leaving the scheduling dialog as the (sole) window open on the calendar. This should be just fine, as long as there's at least one cal-specific window open.)

6.3. Meeting Notifications and Calendar-Host Association

Much thought given to whether notification options should be global (i.e., for the entire tool) versus calendar-specific. Also, some thought was given to having an option to auto-accept scheduled meetings. The first idea was nixed

because we do not want to open the global/local option can of worms at all. The second idea was nixed because it can effectively be achieved by leaving the 'Not-yet-accepted Meetings' option as 'show' and setting the 'Accept Notification' option to 'never'. In this way, the user is not bothered with seeing the notifications, which is presumably the point of the auto-accept option. Without the auto-accept option, meetings that would appear in normal type on calendars instead appear penciled in. But this is fine, because it means that if the user bothers to look at the details of the meeting, all she has to do to get it to normal type is press accept (instead of cancel or close) in the viewing dialog.

Much thought was also given to the cal-to-host association rules, the possibilities being:

- a. ... enumerate 'em

Describe things, when you feel like it.

6.4. Nuked from Options, Related to Overly-Complicated Home Dir Specs

The 'Home directory leading path' setting defines the default leading file name path prepended onto the 'Calendar Tool Directory'. If the path is non-empty, it is prepended iff the value specified for a particular user is not an absolute or user-relative path, where absolute paths start with "/" or "<drive-letter>:" and user-relative paths start with "~".

6.5. Decision about User Control Over Notifications

24jul02 bottom line -- There are two basic categories of notification: meeting and admin. The user has full control over meeting notifications, none over admin notifications.

6.6. Some Uncommonly To-The-Point Fodder from the Options Section

Provide a dialog that looks like this:

```

OUT:

    Accept Calendar Tool Meeting Notifications from the leaders of these groups:
    * all
    * specific group list, with a radio button beside each, scrollable in
      case there are a large number of groups defined
    * none
:TUO (see "NO" paragraph below)

    Accept Email Meeting Notifications from the leaders of these groups:
    * all
    * specific group list, with a radio button beside each, scrollable in
      case there are a large number of groups defined
    * none

    Accept Email Meeting Notifications from new groups you're added to:
    * yes
    * no

    Accept Meeting Notifications from these individuals:
    * all
    * specific user list
    * none

```

The idea is that each user can control who she accepts being bugged by for meeting scheduling. But hmm, am having a bit of second thoughts about being able to reject requests from group leaders, since it doesn't make necessarily good sense; it may well be better just to have a user request to be removed from the group. I do still however very much the ability to be able to accept meeting requests from individuals, as the next paragraph attests.

But OK, here's a case where rejecting requests from some group leaders makes perfectly good sense -- I'm going on vacation and I don't want my calendar and possibly email cluttered up with meeting requests.

So, here's a nice happy medium -- the leader of a group for whom meeting requests are turned off is notified when the turn off happens. This may be getting overly complicated, but hey so's the whole darn system.

NO, screw the happy medium -- users CANNOT decline meeting notification from the leaders of groups they're members of, only email notifications. If they're gonna participate in the Cal Tool community, they're on the hook for receiving meeting notifications.

NO, NO, users can turn off notifications, as imprudent as it may be. See latest version of meeting scheduling section.

The ability to accept meeting request from individuals effectively allow Cal Sys users the ability to establish unofficial groups, by establishing mutual acceptance of meeting requests.

Some question/details pursuant to the above dialog idea:

- a. *Should a user be able to accept meeting requests from the leader of a group she's not a member of? Well, this doesn't really make much sense, because if she's not a member, then she won't normally be sent a meeting request for that group, so it looks like the answer is "no". This means that a user is only asked to accept from member-of groups. [But this is basically bogus, given latest requirements that non-leaders can schedule meetings, and users have full control over penciling-in, in that nothing happens on a user's own local calendar until the user accepts the meeting.]*
- b. *The radio-buttoned of groups list should probably be in tree form, since it will allow the user to select sub-groups by clicking on a parent group.*
- c. *The default ..., fell off.*

6.7. Nuked from Options Section

When the user presses 'Save' the system first performs 'Apply' (if it is enabled), then saves all option settings in the current calendar. If no calendar is open, options are saved in the standard 'Options' file. As with the 'Apply' button, 'Save' is initially disabled, becoming enabled when the user changes the options settings in one or more tabs, makes current a calendar with unsaved changes, or loads a new options file. While execution of 'Save' performs 'Apply', the converse is not true. Therefore, the settings in the options dialog exist in one of three states:

- a. unapplied and unsaved
- b. applied but unsaved
- c. applied and saved

The enabled/disabled states of the 'Apply' and 'Save' buttons reflect which of these states exists for the current calendar file. Since the user may change the current calendar while the options dialog remains active, the applied/saved state of options is specific to each calendar.

Saving options for the current calendar does not save any unsaved scheduled items in the calendar. Scheduled items are saved using the 'File Save' command, described in Section 2.8.2.

6.8. Hopefully Final Nukes for Options

```
Admin
  Default shared user calendar file name: [default = "Calendar"]
  Home directory leading path:
```

6.9. Nice Simplification to User Records

In thinking through clearly how the IPC will work, it must be based on the fact that the host does not have direct write access to the local machines. Something like message passing must happen, along the following lines:

- a. When local cal tool (and/or local cal tool server) starts up, it sends hostname/uid message to central host and establishes two-way comm.
- b. When local host has cal updates, it sends message to central host with the data.
- c. When central host has notifications and pencil ins, it sends message to local host with the data.
- d. Implementors can figure out how to ensure that messages sent from the central host are properly processed

when the cal tool user app is not running. Such messages consist of meeting and other notifications sent by cal tool admins. A local cal tool server is presumably in order, but the user does not have any control of it, i.e., the local user is (must be) unaware of any local server that is running.

Given these observations, the new simplification we've made is to eliminate the 'Calendar Tool Directory' field from the user record. This is appropriate because since the central host cannot write directly on a local machine, it need not know anything about where local files are stored. Rather, it sends a data message to the cal tool app or server, and the server is the one who knows about where to put the data, if appropriate to do so based on user option settings.

6.10. Overly Complicated Font Options Removed ...

```

For User-Entered Data:
  Font Name: pulldown of font names
  Size: pulldown of environment-specific ints
  Style: one of Plain, Bold, Italic, Bold Italic
  In these displays: one of      For these items: one of
    Item                          Appointment
    Day                            Meeting
    Week                           Task
    Month                          Event
    Year                            All items
  Lists
  Dialogs
  All displays
For Permanent Text:
  -- same as type-in text --

```

...

The user can set font options for two forms of text: scheduled items and all other text. The font settings under 'Scheduled Items' apply to the text of scheduled items as it appears in day, week, month, and list views. The settings under 'All Other Text' apply to all other text appearing in Calendar Tool display windows, including user-entered text as well as permanent system-generated text.

The 'Font' and 'Size' data-entry fields are used to enter basic font information. There are two forms of display text for which font selections can be made: typed-in text and permanent text. The settings for typed-in text apply to all text entered by the user in the text-entry fields of dialogs. Permanent text settings apply to all uneditable text that appears in displays, except for the text in window banners. Display-banner text is considered to be under the control of the underlying operating environment, and is therefore not settable from within the Calendar Tool.

The font options tab provides no control of font style or color. The normal, boldface, or italic style of text is controlled entirely by the system. The user can change the text color of scheduled items using the category editing features described in Section 2.5.5. No other control of text coloring is available.

The text of warning and error messages is always shown in bold typeface, independent of the 'Style' option selected for permanent text. The settings for font name and size do apply to error and warning message text.

The '**Scheduled Items**' settings apply to all text entered by the user, *blah, blah, blah*. The '**Headers and Labels**' settings apply to all of the header and labeling text, such as current date, dates, etc, *blah, blah, blah*

... Describe that font, style, and size entries are platform-dependent, but give some typical values. The 'Font Name' and 'Size' menus are platform-specific lists of available fonts and their sizes. The 'Style' shown in Figure 292.

The 'Display Type' pulldown menus specify the types of displays to which the selected fonts apply. The possible selections are the same for both menus and are shown in Figure 292. The default setting for both menus is 'All Levels', which means that all display windows appear in the same font. The user can choose different fonts for each of the display types in the menus. To do so, the user selects the display type from the menu, and enters the desired font information. For example, Figure 293 shows the user having entered values for the monthly-level of calendar display. For user-entered data, 9-point Times-Roman regular font is selected. For headers and labels,



Figure 292: Font options menu.

To appear.

Figure 293: Font option values selected for month-level displays.

10-point Times-Roman bold font is selected. Figure 294 shows the result of the new font selections for the monthly view shown originally in Figure 22.

Sketch of the remainder:

- Describe that the 'Restore Defaults' puts everything back to the system-wide default settings. If the user wants to see what the system default was but not actually go back to it for all displays, she does 'Restore Defaults' and then 'Cancel'.
- Show another example of the effect of font changes on the user-db admin dialog, to clarify the what exactly is user-entered versus header-and-label information.
- Define precisely, probably in tabular form, what the different forms and types of text are, for all possible displays and dialogs.

6.11. Some Serious Rationale for Options

There's no question we went around quite a bit about how best to do options. We considered making them global for all calendars, calendar-specific but still applicable to multiple calendars at the same time, and the current solely calendar-specific style. All of this is a bit much, given that changing options is not likely to be all that frequent an activity, but we do want to get it right.

There have also been plenty of go-rounds for specific details. Noteworthy considerations are these:

- a. the button arrangement, which corresponded to the different global versus calendar-specific explorations
- b. Others that maybe I'll think of at some point.

Methodologywise, the options requirements have been developed thusly:

- a. Do preceding scenarios, describing applicable options as they are developed and refined.
- b. When all options are introduced, scan all sections for them and organize them into the options hierarchy.
- c. Do the detailed scenarios for the options dialog.
- d. Refine in earlier sections as necessary.

To appear.

Figure 294: Monthly view with user-selected fonts.

6.12. More Potential Rationale Fodder Cleansed from Rapidly Maturing Options Section

1jul02 Note. If it's not the case already, and I do not think it is, I believe there should be an option setting for each and every data-entry field in each and every dialog. If this is done entirely uniformly, it seems that we may be able to have the same data-entry dialogs used for default setting as for actual data entry. This could be pretty cool if we could pull it off. -- Done. --

4jan01 thot about 14dec00 thot: Hmm, with movement of prefs to options menu, I'm not sure about the tabbing dialog any more. We may want to put a prefs menu back, with 'Edit Prefs' being tool prefs versus the top-level 'Options' menu being data prefs. I'm not sure the distinction is worth making, so I guess we have to think about this some more. 24jul02 update -- screw this. --

14dec00 thot (and I think it's quite a good one): organize preferences into a tabbing dialog with exactly one tab for each menu. -- Done. --

Among the other options, there should be some for little things. Here's the beginnings of a list:

- (yes/no) remind on File Save As -- turn on/off the reminder about only the default calendar name being used for group scheduling purpose. 6jul02 note: forget this -- the user has to be a big girl; but perhaps we'll have some special annotation in the menubar banner to indicate when the community calendar is active.

Once upon a time, there was going to be a 'Defaults ...' button in the options dialog to restore "default" option values. This has been replaced now with 'Load ...' button, the dialog for which is given just below. The deal is we now have what are best called "standard" options, in the Options file, rather the kind of "inherited from" default options we had been envisioning earlier.

Once upon a time, we said that: "The three radio buttons at the bottom left of the options dialog indicate which option settings are currently active". These buttons are now nugatory, given the current scheme of things, viz., the only "active" options are those for the current calendar. To make other options "active", they have to be loaded, and in the current scheme of things, they become active (and unsaved) for the current calendar only. As rationalized above, to make a set of options active for multiple calendars, they have to be loaded separately for each calendar (which can be done with a script, in the (presumably quite rare) case that we want to do it for a bunch of calendars).

6.13. Here's an Important Bit that Finally Got Handled

This bit is not really in and of itself a part of rationale, but the idea that it's an important piece of information that lingered as only a note for a long time is interesting rationalewise, or someotherwise at any rate. Anyway, here is a note from the options 'Times and Dates' section that finally got officially dealt with. Finally dealing with this note led to an addition to the previously-thought-done structural viewing section.

- a. Be sure to say that none of the settings of these options will ever leave anything out. I.e., the breadth of a display, hour and daywise is maximum of the normal range setting and the range determined by actually scheduled items.

6.14. The Real Deal for Options

The following are the notes that elucidate the real deal for options. These notes were refined into the current requirements in the Options section. Only the last item of these notes needs to survive as rationale, since all of the other items have been incorporated nearly verbatim in the requirements, whereas the last item appear nowhere since it's pretty much strictly rationale. OK, try this for how to deal with "global" versus calendar-specific options.

- Each and every calendar has its own options, period. When a calendar is created, the default options are copied into it. When a calendar's options are saved (see details below for how), they are stored within the calendar's data file. I.e., there is a single file containing calendar data plus options for that calendar.
- The standard (default) options are stored in the file `~/CalendarTool/Options`.
- If the user changes and applies options while the calendar tool is running, the changes apply to the current calendar only, as unsaved options for that calendar.
- If user executes 'File Save' for a calendar with unsaved options, the file-save dialog contains the message and radio buttons:

```
"There are unsaved options for this calendar."
(x) Save these changed options for this calendar
( ) Don't save changed options, i.e., leave the most
    recently saved options for this calendar as is
```

- If the user exits or closes a calendar with unsaved options, then in addition to whatever offer-to-save file messages there may be, there are also offer-to-save options messages. Viz., there is one such message for each active calendar for which options have been applied but not saved. The text of these offer-to-save messages should most likely be integrated into the file offer-to-save dialogs, so two dialogs don't appear if both the file data and options need to be saved for a particular calendar.
- There's no (non-scripting) way to change options for all existing calendar files, or even for all currently open calendars, as there would be if there was a dynamic options selection feature in the GUI, i.e., that allowed local options to be turned off in order to dynamically enable the "global" default options in the Options file. The rationale here is that there are not likely to be a lot of calendar files period, let alone a lot of them that all need to have their options changed at once. In another system, it might be worthwhile to have a GUI way to select dynamically whether to use global or local options, but not in the Cal Tool given the preceding observation about the small number of files.
- See the save and load dialogs below for how the user performs these options operations.

And the following section was just nuked from the options section, per the explanatory 29jul02 remark that precedes the rest of the notes:

.sh 3 "The Format of Options Files"

29jul02 Update. For doability, we need to abandon the idea of option files being parsable command files, or even user-accessible text files, at least for now. The reasons are several:

- a. With the latest scheme of options semantics, the single-options-file idea is basically dead. This is because options are now embedded in every calendar file, and as such, not that amenable to formatting as command streams or even in textual format, unless we want to impose parsing requirements, which at present we do not.
- b. Putting the standard Options file in command-stream format might still work, but if we do so its format will be inconsistent with the calendar-embedded options, so we'll skip this too.
- c. We're getting pretty darn close to dropping the whole command language thing anyway, which if it happens would make the whole options-as-commands thing moot.

Here's where things started, before the 29jul02 bail out:

This should be, a la emacs, executable commands. Here's a plausible/possible example:

```
caltool.options.timesAndDates.set...
file.open("PersonalCalendar");
Schedule.ScheduleAppointment(
    new Appointment(
    );
```

Hmm, context is everything here. Having some trouble figuring out how to establish some, whether by arg or "set" methods. Also having trouble figuring out the scripting API vis a vis the current model API; supposedly they're the same, but it looks like java packaging and static versus non-static method syntax could be a serious pain to deal with in an options script file.

24jul02 -- Try this (29jul02 -- italics after each point indicate whether the point has been incorporated (OK) or nixed (NIXED) from the requirements):

- Calendar data created with any 'Schedule' or 'View' command are stored in individual calendar files. (OK)
- Calendar-specific option data are also stored in the calendar files. (OK)
- All-calendar option data are stored in a separate "Options" file (or some such name) in the cal tool user dir (folder), where this dir is as spec'd in the user's record. This is where we might use the idea of putting all-calendar options in the init file (as "set" commands), or have separate options and init files. I'm inclined towards the emacsesque options-in-init-file-as-commands form. (OK somewhat, but refined to latest semantics of standard Options file and

set command stuff NIXED.)

- Note that even though no user record data are stored in local files, the installed calendar tool itself must have some knowledge of the location of the cal tool folder; we could do this netscape/mozilla style and put this in a fixed (known) place under the user's, or figure it the you-know-what out so that the installed cal tool always knows where it needs to go to find its options file. (OK)
- We need to figure out if we want to have the standard format of a calendar file be readable text, or generate a separate plain text file as a save option. Given the presumed Java implementation, it's probably the former for starters, since we'll only have to write a generator. Eventually, I'd really like the text form to be *the* form, with a full-blown parser, so we can edit cal files in emacs and just read them in to the cal tool. (We've figured it out, having NIXED the parseable file format idea totally for this version of things, and therefore most likely forever (!).)

6.15. Thinking about Options and Defaults

Nuked the following idea in favor of a button at the bottom of the options dialog. I think we made the right decision.

NOTE: Add a separated 'Save ...' item at the bottom of the 'Options' menu. It launches a dialog that asks something like the following:

```
* Save for this calendar only
* Save for all calendars

Options name:  [ .calendar_options ]    ( Browse ... )

( OK )      ( Clear )      ( Cancel )
```

Note also that this item probably only applies to non-admin users, since for the admin, the options settings are saved as part of the complete set of admin DBs, meaning admin-level options are covered by 'File Save'.

The following was a related bit of murky thinking about defaults-related processing in the appt-scheduling section. It's (obviously) been refined. Note that the reference to "the preceding two refs" were to places I vaguely understood as where the details of global default setting would go.

To clear all information entered in an appointment dialog, the user presses the 'Clear' button. In response, the system clears all typing areas and restores all other data-entry fields to their default states. Details of default settings are covered in Sections 2.6.7 and To cancel a scheduling command entirely, the user presses the 'Cancel' button. In response, the system removes the dialog from the screen without performing any scheduling action.

[NOTE: The preceding two refs need to be fixed, and we need to figure out exactly what scheduling options include, in particular if they allow default values to be entered for each and every field. Perhaps the easiest and most general solution is simply to have a 'Set Defaults ...' button in each schedule options subtab. This button brings up a standard dialog for each type of scheduled item, with "DEFAULTS" somewhere prominently in the banner or title. I think this is pretty cool, actually, because it will illustrate a nice general and orthogonal way to allow the user to set defaults.]

6.16. ScheduledItem Inheritance Structure

We need to rationalize how we let the pedagogical goal influence the model structure of scheduled event a bit, given that we wanted to specify (and subsequently implement) a "good looking" inheritance structure. This includes a discussion of the Event item, which we've gone back and forth with in terms of exactly what fields it has. As of 9mar02, we in fact have the situation that the implementation is wrong vis a vis the specs, since the implementation has location in an Event, whereas the spec has SimpleSecurity. This component change to Event apparently happened on 16jul01, according to the CVS log entry of that date for version 1.17 of schedule.rsl. The only explanation of what was evidently some deeper thought on the subject was "Removed Location and added SimpleSecurity to Event (this being a significant update)". I guess the thinking was that location doesn't necessarily make sense for an event, particularly for things like holidays, as well as for the example events we have in the scenarios, like "Jim's birthday". Anyway, this biz should be discussed here. Yes, we just did, and so just a bit of "grooming" of this discussion needs to happen before we release this tome.

6.17. A Feature Removal (!)

The following started out to be an Advanced Option, but as the bulleted items indicate, it's not needed.

A useful but low priority option would be to find all events that have reminders turned on. This would be useful to me because I may have accidentally turned on a reminder when I didn't want to, or forgot to turn on a reminder for something I should have. Anyway, this is a low priority requirement.

Ah, but we can do this pretty easily as follows:

- filter out all but events
- sort event list by reminder, which will put all events with reminders on first in the list

6.18. More on Admin Decisions

Removed the 'List Admins' command in favor of having an 'admins' group that has the IDs of all users with admin privileges.

The following was decided to be too cumbersome:

The 'Members' and 'Leaders' lists are editable by direct typing. To add, delete, or change a user from either list, the administrator performs the appropriate typing.

The following was enacted as stated, and this paragraph is a bit of rationale, perhaps:

what happens to all of the meetings for a group when the group is deleted; I think the group name just stays there; hopefully we haven't provided a way for the user to double click on an attendee group to see its information; if we have, that has to give a "the group no longer exists" error message when executed on a deleted group; otherwise, we just have the somewhat funky state where a group name is in a meeting, but if the user goes to look up that group it won't be there; this will have to be the only indication to the looking user that the group has been deleted

This stuff got nixed:

[Include a field that allows the existence of the group and/or its membership to be kept private. Also include confirmation protocol that allows the intended group leader to accept or decline becoming group leader.]

When the administrator assigns one or more users to be a group leader, the assignment does not take effect until the assignee(s) is(are) notified and confirm the assignment. Specifically, when the uGroup Add command is confirmed by the administrator, a confirmation request is made to each of the assigned group leaders. The form of the confirmation request depends upon the current setting of an assigned leader's reminders options. Specifically, the confirmation request is sent immediately as an on-screen pop-up window if the assigned leader has 'on screen' selected as the default value for receiving reminders. For any other setting default reminders setting, the system sends the leader confirmation request via email.

To avoid the potential difficulties of interfacing the Calendar Tool with a variety of different email systems, *blah, blah, blah ...*

6.19. Fodder Nuked from Admin Section

.sh 4 "Group Membership Requests"

An individual user should be able to:

- a. see what group she's a member of
- b. request (demand?, force?) removal from the group
- c. request (demand?, force?) membership in a new group

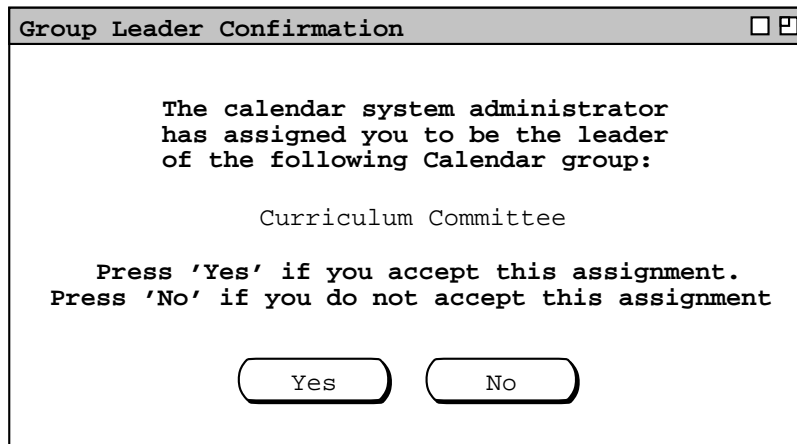


Figure 295: New group leader confirmation.

6.20. Detailed Explanation of Recurring Date Change Restrictions

Recall that all instances in the same collection must have the same start and end dates. To see why this is necessary, consider the case where some instances in a collection, call them "Part A", have an later start date than other instances, call them "Part B". If this were allowed to happen, then the calendar would have to appear inconsistent from the perspective of either Part A or Part B. If the later end date of Part A is considered the correct, then there are later items in the calendar than any of the Part B instances indicate. Hence, when locating at a Part B instance, the end date is inconsistent with the items that are actually in the calendar. Conversely, if the earlier end date Part B is considered correct, then the Part A instances are inconsistent with the calendar, since they indicate that there should be items in the calendar that are not there.

6.21. Changing and Deleting Meetings

Discuss all of the issues related to scheduler and non-scheduler change and delete. There's a boatload of fodder below that needs to be dragged into here.

6.22. From the end of the change and delete section

```
.sh 4 "Changing the Status of a Recurring Item"
```

At least some of this section is wrong. It should be nuked entirely.

When the user changes the status of an item from non-recurring to recurring, or vice versa, the change has a more global effect on the calendar than does a change to any of the other data fields.

When the user changes an item from non-recurring to recurring, what was a single item becomes an instance of a recurring item. When the change is confirmed, calendar and list views may have multiple new items added to their displays.

When a recurring item is changed to non-recurring, the effect varies depending on the scope of the change selected in the change confirmation dialog. When just one instance is changed from recurring to non-recurring, that instance becomes a single non-recurring item, dissociated from all other instances with which it was previously associated. All other instances except the changed one remain as associated instances of the same recurring item. Any subsequent changes made to the changed instance have no effect on the formerly associated instances. Changes made to any formerly associated instance have no effect on the changed instance, but do have an affect on the remaining associated instances.

When all or all future instances are changed from recurring to non-recurring, all of the affected instances become single items, dissociated from all other previously associated instances. Subsequent changes to any dissociated

instance affect that instance only. In the case of a change to all future instances, subsequent changes to a still-associated past instance affect all past instances, but none of the dissociated future instances.

.sh 4 "Changing and Deleting Past Items"

There may be something to say here -- not sure right at the moment.

6.23. From the task scheduling section

The following issues were all dealt with as appropriate.

..., subject to the following restrictions:

- a. for a non-recurring task, the completion date must be on or after the due date
- b. for a recurring task with 'this instance' selected, the completion date must be on or after the instance date
- c. for a recurring task with 'all instances' selected, the completion date must be on or after the instance date

Refer to all fields that are in common with appointments, which should be all be completed and completion-date. Do two or three scenarios for these fields and we should be done.

The following was moved from the task item viewing section, since it appears pretty clearly to belong here instead.

More notes based on task viewing work:

- tasks can be completed early or late, without the kind of annoying "carry forward" behavior of Claris
- the default completion date entered for a task will be the current date by default, but it's fully editable without interference from Cal Tool (or perhaps we'll allow interference or prevention via an option, but I'm inclined not to, not least for reasons of simplicity)
- tasks can be scheduled and "back dated" without interference from the tool (again, interference of some kind may be a settable option, but it definitely won't be the default behavior)

6.24. Possible fodder for schedule meeting

Of note in Figure 91 is the "[2]" suffix in the 'On' date field. The suffix indicates that this is the second meeting scheduled by estier on the same date. The suffix number is increased for each additional same-date meeting made by the same scheduler. The suffix ensures that the 'Scheduled By' and 'On' fields provide unique identification for the purposes of changing or deleting a meeting, as discussed in Section .

6.25. Possible fodder for meeting change/delete section rationale

[in the sched delete confirmation dialog] The user may edit any data fields except 'Scheduled By' or 'On'. Performing edits is only meaningful if the user plans to decline the deletion. The user may choose to do so if she wants to retain a copy of a cancelled meeting.

The following was improved by the policy of disabling scheduler-changed data fields in non-scheduler item-level displays with not-yet-accepted scheduler changes:

While it may be imprudent to do so, the user may continue to change his own copy of a scheduler-changed meeting, prior to accepting or declining the scheduler's changes. A particularly imprudent sequence of actions would be

- a. changing a critical data field, without first viewing the scheduler's changes in the notification dialog
- b. pressing 'Accept' in the item-level display, again without viewing the notification dialog

Since the system retains the most recently changed data values, this action sequence could override scheduler changes that the user never sees. *And blah, blah, blah about a bit of the rationale behind this.*

The following was way too simple an explanation for what it was aimed at:

Explain that for the accepting user, accepting the notified changes has exactly the same effect as if the user had made the changes herself to her own calendar.

The following turned out to be wrong, since we show scheduler changes and deletions in penciled-in form on users calendars:

Explain how attendee display screens are updated. Viz., changes don't happen until acceptance of change/delete notification.

The following turned out to be too complicated, since we went the most-recent change route for merging scheduler changes:

Deal with the issue of what happens when the user has changed one or more data fields when the notification arrives. It probably needs to be some kind of diff3 deal.

The next six paragraphs were unrefined issues appearing in the non-scheduler change and delete section. The issues have all been resolved, one way or another. A good rationale discussion will/would discuss which ideas were refined and which rejected.

Any critical meeting change made by the scheduler must be reported to all attendees. As discussed in the next two sections, individual users may decline to receive notifications or change their own copies of meeting announcements. The policy upheld by the Calendar Tool is that the scheduler's version of a meeting is considered to be the definitively correct version. Attendees are always notified of scheduler changes, which they may deal with as they see fit.

Change restrictions are imposed on a meeting scheduler because the scheduler is considered the responsible party for the meeting and it has to make sense there, ... or something like that.

An individual user can change any fields of a group meeting (except 'Scheduled By' and 'On'), since the user owns the item. The user can even do stupid things like change or remove attendees, or other changes that may not make good sense. It's up to the user. Any such changes are reflected only in the individual user's copy of the item. The official record of the meeting is the one that appears in the user calendar of the user who scheduled the meeting.

Or maybe it's a bit more sensible to allow only some of the fields to be changed, viz.: category and reminder. Changing any of the other fields is arguably non-sensible.

OK, here's the deal. If for some reason, the scheduler cannot or does not want to use the Calendar Tool to notify attendees of a meeting change, then it does in fact make sense to allow individuals to change all fields. The following warning dialog probably makes good sense to make things all nice and happy:

Changing any of the following fields makes your version of this meeting inconsistent with the scheduler's version:

**(Start) Date, End Date, Start Time,
Duration, Recurring, Location, Minutes**

The only fields that cannot be changed are 'Scheduled By' and 'On', since these is the permanent record of the original scheduler and schedule date, which serve as the unique identifier of the item on all users' calendars.

Or maybe we don't want the scheduler to have to be the attendees' nanny. I.e., we don't care if an attendee deletes a meeting. Attendees are assumed to be grown ups that can come to a meeting or not. If an attendee wants to inform the scheduler that s/he can't make it, then the attendee can do it through some means of communication outside of the Calendar Tool. I'm leaning towards this, just to keep things from getting out of hand notificationwise. I.e., not having attendee cancellation notifications go to the scheduler will make things less cluttered.

Do these examples:

...

- a. Delete the tenure review meeting; show how the details field can be used to send a message to the attendees, even though it will be deleted upon acceptance of the notification -- THIS CANT HAPPEN because you cant edit then delete. ... all other examples were dealt with properly

The following is some old stuff that needs to be weeded through.

Basically, the same rules should apply as for individual meetings (see Section 2.5.2), but with the wider-ranging consequences of having to notify all participants.

The dialog for changing or deleting a recurring works a la Claris. When a leader changes or deletes a group meeting that she scheduled, a dialog of the form shown in Figure 296 appears. Or, the dialog may provide no means to forgo attendee notification.

We (?may/probably?) also want to allow for someone other than the original scheduler to cancel a meeting, which seems most sensibly restricted to leaders of the at least one of the groups for which the meeting was scheduled. While this may weaken the cancellation security a bit, it's probably the easiest and most sensible way to do it, without getting involved with some big complicated deal like what percentage of attendees you must have control over in order to cancel a meeting. The issues we need to deal with in this regard are the following:

- a. Should people other than the scheduler be able to cancel a meeting at all? Pros are convenience, cons are security and complication of who to allow.
- b. If we allow non-schedulers to

At this point, I'm inclined to say the only people who can cancel a meeting are the original scheduler or a super user. Allowing the super user to do it provides the necessary flexibility, and keeps from having to figure out who the heck besides the scheduler should be able to delete meetings.

Somewhere in here, talk about changing minutes locations for recurring meetings. A reasonable scenario is edit each individual meeting record with the specific file in which minutes are held. At the exact moment of this writing, I'm still thinking about whether to allow directories for minutes of recurring meetings. If this subsequently gets figured out, then we'll deal with it here accordingly.

6.26. Possible fodder from appt changing section

As the dialog explains, the user is being asked to select how many of the recurring instances to change. If the user selects 'This one', then only the single instance shown in the item-level display is changed. All other instances remain unchanged. If the user selects 'All', the system changes each and every recurring instance per the edits made in the item display. If the user selects 'Future', the system changes the displayed instance and all future instances. If the user selects 'Cancel', no changes are performed. For any selection, the system removes the confirmation dialog from the screen. For any selection except 'Cancel', the system changes the button state to the initial configuration. For 'Cancel', the button state remains unchanged.

In this scenario, the user selects 'This One' in the confirmation dialog, whereupon the system proceeds with the single-item change and all necessary view updates.

6.27. (Weakly) Possible fodder from meeting item viewing

Depending on the user who is viewing the meeting, different data fields are editable. Specifically, the scheduler of a meeting may edit all data fields except 'Scheduled By'. User's who are not the scheduler may edit some of the data fields. Figure 137 shows the case where a non-scheduler is viewing the meeting, with the non-editable fields disabled (the edit fields have grey borders). Further details on editing scheduled meetings are covered in Section 2.5.2. Selecting a recurring item for viewing is the same as a non-recurring item. Namely, the user selects the desired instance in the current display window then executes 'View Item'. For example, Figure 297 shows the user having selected the October 1 meeting instance in a month-level display.

You are the scheduler of this meeting. Do you want all attendees to be notified of the change?

Figure 296: Changing or deleting a meeting as leader.

6.28. Possible fodder from task item viewing

Make note also of the fact that task titles are integer-enumerated in day and week views, not so enumerated in month views or lists. This is a Clarisism that we may want to reconsider. (Update: nope, we aint' reconsidering it at this late date, having drawn all of the task view pictures this way.)

6.29. Possible fodder in the area of external file viewing

The following was nuked from the meeting scheduling requirements:

The 'Browse . . .' button next to the 'Minutes' text field leads to a file browser, of the form described in Section . The browser is used to select the name of a file or directory that will hold the minutes. If the minutes are to be located in a URL, the scheduler can browse for the location in terms of a file name, and then type the additional URL prefix in the text field. Alternatively, the scheduler can type the entire file name or URL without using the browser. The minutes field is optional and may therefore be left blank.

The following was nuked from the item-level meeting viewing section:

The figure shows the default form of minutes display in plain text form. As an option, the Calendar Tool system administrator can select an external program for viewing minutes, such as a WWW browser. Selection of such an external viewing program is explained in Section 2.7.5.

It is entirely the responsibility of the scheduler to name the files in such a way that users can discern which minutes file applies to a particular meeting occurrence.

6.30. 2aug01 -- Nixing the notification enabling stuff (then not)

But "oh contrair" to what follows. Per latest version of meeting scheduling stuff, users can in fact decline all forms of notification. Rationale needs to be written accordingly.

OK, I think I'm ready to forget about the whole user-enabling of meeting notifications, and simplify it to these rules:

- a. only leaders can schedule meetings for groups
- b. anyone can schedule meetings for individuals

Users have no ability to decline any kind of meeting notification. I think it's reasonable to assume that the nature of the cal tool is one of openness and trust. Therefore, if one chooses to participate in the system, one must be prepared to accept meeting notifications from other individual users and from group leaders on behalf of the entire group.

The deal about only group leaders being able to schedule for a group seems pretty sensible to me at this point. The deal is that if someone wants to be authorized to schedule for a group, it's easy enough to be added to the leader list for that group.

The bottom line is that it's pretty much like email, in that anyone can send to anyone else. There's a bit more in favor of receivers in that only group leaders can define aliases to send to. There's a bit less in favor of receivers in that "penciling in" can be done, which is a bit more "personal" than just dropping something in a mail box. However, it really just looks that way, in that it's not really writing on another user's file. After some usage, we may want to allow users to have an option to disallow penciling in. OK, I think I just talked myself into providing the option to at least toggle penciling in on or off.

The following verbiage was nuked because of this nix:

Meetings scheduled by group leaders have an official status that meetings scheduled by non-leaders do not have. Specifically, the following apply to leader-scheduled meetings:

- a. notification is sent to all attendees
- b. the meeting is added to the group calendars of which the scheduler is a leader
- c. the leader may make changes or cancel the meeting for all attendees
- d. when an individual attendee deletes a leader-scheduled meeting, the leader is notified

Further details of these matters are covered in upcoming scenarios.

Sketch: The scheduler wants to schedule a one-time meeting among some Cal Tool users. There is no user group defined for the attendees and the scheduler does not have notification privileges for any of them. Hence what the Calendar Tool can do is find some meeting times, and the scheduler can notify the attendees externally. So, the scheduler tries a narrowish range of possible times and comes up empty. The scheduler then widens the times, finds some, and makes a choice.

When scheduler is unauthorized to schedule a meeting for one or more attendees, the system displays a dialog of the following form (this should most likely go in error conditions section):

The following attendees will not automatically be sent meeting requests because you are not authorized to schedule a meeting for them: ...

A scheduler is authorized to send (via the Calendar Tool) to user X if either of the following is true:

- a. she is a leader of a group of which user X is a member
- b. she is listed by user X as one of the individual users from whom meeting requests are accepted (see Section).

Explain that the Caltool requires one of these forms of authorization to avoid (in)advertant spewage of mail by unauthorized Cal Tool users. Furthermore, the receiving users will have explicitly authorized the the Cal Tool to send them msgs when they either accept group membership or set their options to allow meeting announcements from selected Cal Tool users. This user acceptance is an important piece of non-spam business.

The meeting scheduling auto-notification rule is the following: a Cal Tool user must take *some* explicit action in order for the Cal Tool to send a meeting notification. That explicit action is one of the following: accepting membership in a group; setting an option that authorizes a particular Cal Tool user to schedule meetings for the granting user. The point is to strike a balance between the following: (1) want to give users control over from whom they accept meeting notifications; (2) want to allow non-leaders to be able to schedule occasional meetings based on users calendars.

Users can elect not to receive email notification but not cal tool notification. The only way not to receive cal tool meeting notification for a group is to be removed from that group. The way not to receive meetings scheduled by non-leaders is not to set the option the default for which is off.

6.31. 30jul01

Here's a bit of overly complicated spec for the possible meetings time list that may be of some historical interest:

The items are sorted in the three separate sections:

- a. possible times that all attendees can make
- b. possible times that overlap with optional items for one or more attendees, but not with any must items
- c. possible times that overlap with must items for one or more attendees, and possibly with optional items for some attendees

Each of of the sections is sorted separately, first by date and second by time, from earliest to latest. The optional overlap section are sorted third by the number of attendees with overlapping items. The must overlaps section is s

6.32. Axed from Meeting Scheduling

The following notes were removed from meeting-scheduling; there may be some useful rationale fodder in them.

6jul01 IMPORTANT NOTE: *An important realization that I'm coming to is that the idea of allowing group leaders to physically "pencil in" is probably a major invitation for security breaches and implementation problems, which means I'm inclined at this point to say that there's always a dialog that pops up during or at the initial lanuch of the tool that asks the user to accept or reject a scheduled meeting. In this way, group leaders, including the maximal leader, can never directly write on any user's calendar.*

Thots:

1. Lose the columns in the Attendees list in Fig 12 and the check boxes at the bottom of Fig 13. In Fig 12, the 'Attendance Required' flag is not needed fro reason described below. The 'Not Registered' flag can be

figured out by the system and a dialog of the form "The following are registered users of the Calendar System, do you want to include them in the attendees list anyway?" Of the check boxes at the bottom of 13, the first goes in the new "Adj Params" dialog described below and the second check box is gone, for reasons described below.

2. Add an "Adjust Parameters" button at the bottom of the dialog.
3. Don't think the "Attendance Required" flag really helps, given the new way we'll display the possibles list.
4. OK, here's the new I think pretty darn cool way to display the possibles list: All times where all can meet, followed by all times all but one can meet, etc. This gives a good set of information and I think obviates the need for attendance required since we can double click to see who cant come. Plus in the scheme of things at a practical level, I dont think it really helps any, since if we dont care about someone, then why bother to include them in the first place, plus, again, we'll now be about to see them in the list of people who cant make it.
5. In current Fig 13, lose choice for screen alert -- it's mandatory. Per recent additions to user options, the user can control when she sees alert -- at startup, exit, during use. Also, recent thinking is that group leaders are the only ones for whom Cal Tool notification will be sent of scheduled meetings. This will prevent (in)advertent scheduling of meetings by peons. As far as scheduling for individuals goes, this is controlled by the users's settings of from who to accept announcements.

6.33. A Bit on Task Scheduling

This was wacked from the task-scheduling section:

17dec00: I think we want to include a due time for tasks. Don't think we need durations. The motivation is for reminding, including being reminded in a smaller granularity than a whole day. The time can be left blank, in which case it defaults to the currently set value for the end of the normal day range setting. (I think the last bit is pretty cool and sensible, actually.) 10jul01 update: As cute as the last bit is, it goes because tasks without a time are considered to have times latter than all those with times, per latest details of list sorting. Keep it simple, dickhead.

6.34. Documentation Section Ordering

Here's the basic story we want to tell:

- a. Show the reader how some basic scheduling happens, using an appt as a representative example item.
- b. Then, without getting into all the details of scheduling just yet, show how the user views calendars.
- c. Then return to finish up the details of scheduling.
- d. Wrap up viewing and scheduling finer points, particularly changing and deleting, which is a combination of viewing and scheduling functionality.
- e. Do the admin and options commands.
- f. In terms of menu commands do file, edit, and help last, since they are the most mundane.
- g. Follow the scenarios with the real nitty gritty details of data entry, error handling, and GUI requirements.

6.35. Relative Importance of Actual Tool versus Pedagogical Exmample Goals

Some frank and 100% truthful discussion goes here.

6.36. Filter Dialog Layout

Pretty darn complex. Thot about trying shortest/longest pair, but decided on current form because (a) it can be shared with task priority; (b) expression is nice and general and with the popular advent of search engine query exprs, the concept of a search expression is not entirely foreign to what might be considered typical users.

At this point, I'm not 100% clear on the relative expressive power of various forms of GUI versus text expr, and I don't think I want to be. E.g., what are the ease-of-use versus power trade offs in a smalles/largest numeric par GUI versus a bool expr text box? I thought about it plenty and am happy with the result.

Specific actual issues in this case are:

- a. Most of it won't be used much by typical users.
- b. Since I'm designing it for me, and I'm happy, we've achieved the goal of customer satisfaction.

6.37. Refinement Example

See (and discuss here or elsewhere).

Start Date: End Date:

6.38. Alternatives for Multi-Window Mode Behavior

In an earlier version of the requirements, the following statement was made in Section 2.3.2.1 regarding the behavior of the next and previous arrow buttons vis a vis the behavior of the 'Next' and 'Previous' menu items:

"... Pressing one of these arrows has the same effect as the corresponding menu command, with one exception. The exception is that the setting of multi-window mode is ignored when the arrow keys are used. That is, pressing an arrow key always changes the display in the window to which it is attached, and never displays a new window."

This statement was removed on the grounds of non-uniformity of behavior. The original rationale for having the non-uniform behavior (i.e., arrows ignore multi-window mode) was solely one of UI ergonomics. Specifically, ergonomically it is nice to leave the mouse in the same place when traversing via the buttons, so the user can just "stand" on the same button and keep moving along. The reason this is not an issue when using the menu command is that the user has to have moved the mouse anyway.

"Mouse-steady" traversal could be achieved in multi-window mode by having the windows stack up directly on top of each other, but this should be rejected as too easy a way to have invisible windows pile up on the screen.

In the end, we figured that it is easy enough for the user simply to turn off multi-window mode to get the ergonomically desirable behavior. In general, the presumption is that multi-window mode will be off far more often than it is on anyway, with users turning it on temporarily to set up a particular form of side-by-side display, and then turning it off. Hence, the disadvantage of non-uniform behavior is greater than an ergonomic inconvenience that can easily be avoided by changing an option setting.

6.39. Precise Behavior of Next and Previous at the Item Level

The exact behavior of 'Next' and 'Previous' can be specified in a number of ways. Two different behaviors were considered in these requirements. The behaviors can be characterized as an *all-item* traversal versus a *type-specific* traversal. In the all-item style, 'Next' and 'Previous' traverse through a list of all scheduled items of all types. For example, consider a schedule consisting of the following three items: a 4PM Monday appointment, a 9AM Tuesday meeting, and a 10AM Tuesday appointment. Consider also that the current display is the 10AM Tuesday appointment. In the all-item style of traversal, pressing 'Previous' changes the display to the 9AM Tuesday meeting. In the type-specific style, pressing 'Previous' changes the display to the 4PM Monday appointment.

As explained in Section 2.3.2.1, the all-items style was chosen. The rationale for doing so based on the following considerations:

- a. the effect of the item-specific style can be exactly achieved using filtering; e.g., to traverse only through appointments, the user filters out the other item types
- b. with multi-window mode off, the all-item style does cause the size of the item-level window to change when traversing to an item of a different type, which could be distracting to the user; however, this was deemed to be an acceptable distraction, given the more important ability to traverse easily among all items, plus the aforementioned fact that item-specific traversal can be achieved via filtering

The would-be rationale for choosing the item-specific style was based on the following considerations:

- a. the all-item style traversal would cause the size of the display window to change when traversing between different types of item; this could be distracting
- b. while the all-item traversal has the advantage of allowing the user conveniently to traverse among different types of item, the same effect can be achieved in multi-window mode as follows:
 - i. the user turns on multi-window mode
 - ii. she selects to view an appointment item
 - iii. she selects to view a meeting item
 - iv. she uses 'Next' and 'Previous' in the two different windows

This reasoning seems to be considerably more confusing and contorted than the reasoning in favor of the all-items style of traversal, which led to the adoption thereof.

6.40. Misc Ideas from Relatively Early On

The time range; times not in the range will not appear at all until the option value is reset; I'm considering a quick command to toggle between full and reduced display, but maybe it's really just as easy to change the option. This is really a minor ergo matter, but worth considering for its perhaps general applicability. To wit, the reason that a separate toggle-full-partial view command is useful in addition to changing an option setting is that it's typically not that easy to change back to a previous option setting, even with general undo/redo. I.e., the user would go to the options, find the option for how many hours to display in the daily view, change that to 'show full', with presumably a quick way to select 'full', then go back to the daily display. But then if the user wants to change back to the restricted range that was the previous option setting, there seems to be no quick way to do this based on general commands such as undo. We might want to postulate some general way to do this, but it seems rather esoteric and a case where we're trying to get a bit too scientific in the UI design for our own good.

We should consider allowing time range to be changed for a single appt, affecting only a single display, as well as for the full calendar, affecting all daily displays.

What fields of the scheduling details are displayed; depending on how carried away we get here, we could lapse into UI building, which I do *NOT* want to get into in this version of the system. OK, let's not get carried away at all; here's the deal. The only options the user has for display is (i) whether or not the actual time is shown along with the title, (ii) whether or not duration arrows are shown, and (iii) whether or not dashed lines are shown. The precise position of the dashed line is up to the system. A funky thing that can happen is for a really short event, the text height of the event title may be physically taller (in terms of time displacement) than the duration of the event. In order to keep this from happening, the system will automatically adjust the height of the daily display based on the granularity of the time division option set by the user. Specifically: (i) the display always shows one hour granularity of labels; (ii) the height of each hour is equal to one text-line -- NO this sucks -- there should either be no setting of time division at all by the end user (i.e., get rid of the option), or the option setting should only be used to prevent a meeting from being scheduled of a particular duration, but in any case, the time division option should not affect the height of the display. Rather, if a user schedules a bunch of really short meetings within one hour, then some little micro scroll bars should show up in the individual hour that is affected and/or the hour should grow to a different size wrt to the other hours in the day and then if necessary the whole day can be scrolled down to see other hours. YES -- I like the last of these ideas. Viz., that the height of an hour within a day will grow if necessary to show all non-overlapping appts that start (and end, except may for the last), within that hour. The default height for an hour should be two lines, based on the height of the current font. One last thing about the duration arrow -- it won't be drawn if there's not enough vertical room based on the height of an arrow head and some reasonable default for the minimum size of the line tail of the arrow.

6.41. Start/End versus Start/Duration, Revisited

Below it says we've gone start/end time, but I'm leaning towards going back to start time/duration, given that it makes more sense overall. The specific reasons now are: (1) we liked it better in the first place, and I think there was a good reason for this that I've forgotten; (2) the late-night appointment thing is infrequent, but important enough; (3) the major conceptual problem of distinguishing multi-day recurring from multi-day single I'm pretty sure will be a non-issue when we think it through.

OK, here's the gist of what we've been worried about recently. There's a consistency problem with tasks vis a vis other items when it comes to the due date, particularly with a recurring task. Viz., if there's a fixed due date for a recurring task, what does this mean the due dates are for the tasks that recur. Though I've not looked into it perhaps to the total depth necessary, it appears that both Claris and dtcm have problems here. It's particularly noteworthy in Claris, where in the Task creation dialog, 'Due Date' changes to sense at all.

A related problem is in the creation of the generic ScheduledItem object. Yes, we know of course that inheritance should not drive requirements, but the other way around. That said, there seems to be a good reason for inheritance in the modeling sense, since we would like to use the notion of a generic scheduled item in sorting, and probably other cases. What's useful about generics in these cases is that we can refer to common fields shared by all of the objects. Anyway, if we go for tasks having due dates instead of (start) dates, the structure of the generic scheduled item starts to break down. What happens is we can change the StartDate component to just "Date", but this then lacks modeling power, since the generic "Date" field is in fact interpreted as "Start Date" or "Due Date" in specializations. An alternative is to name the type of the generic field StartOrDueDate, but feels like it's getting pretty hokey.

The serendipitous solution to this mini-mess appears to be the consistent use of duration in all items. In appointments and meetings, it has the obvious meaning, and the UI provides hours and minutes to enter its value. In tasks, the meaning of duration seems pretty clear as well, plus the definition of "due date" is the sum of start date plus duration. This seems to be quite sensible indeed, and the only small drawback is that the words "due date" don't appear explicitly in the task scheduling dialog. However, we may well use the term "due date" in other UI windows, such as some form of task list. (Though such a list may not happen in version 1, for the sake of simplicity.) Also, the UI for duration of tasks and events can include a days entry box (and maybe nix the minutes box), since tasks and events typically span days, whereas appointments and meetings do not. So, the conclusion here is using duration instead of end time seems to work very well overall, and serendipitously allows the ScheduledItem object to be nicely defined.

6.42. Start/End versus Start/Duration (Older Ideas)

We've gone to start/end, since it appears to be the method used in other calendar systems. The reason is probably because of the underlying conceptual problem with start/duration with respect to scheduled items that span a single day. In one sense, it would be nice not to restrict an item to be within one day, particularly for night people who do regularly do things around midnight. Evidently, since there aren't a lot of people like this, calendaring systems don't care that scheduling a meeting to start at 11PM and last to 1AM the next day is a problem.

However, the deep conceptual problem is distinguishing between a multi-day recurring event from a multi-day *single* event. We need to think about this. And I'm not just falling off here -- I really haven't thought it out fully nor do I want to right now.

6.43. Big Issue about Factoring Options

Well, we've never fully come to terms to what exactly an option is and how it should be represented in a UI. This issue comes a bit more into focus when we consider whether UI access to options should be distributed across command menus and individual dialogs and/or centralized in an "centralized" 'options' menu. There's more to say and think about here.

27jul02 update: cWell wonder of wonders, I do believe we have come to a firm and quite satisfactory decision here.

6.44. Old Remarks from the List Viewing Section

OK, what we need to do is make sense of applying filters to both calendar and list views. I think this can be perfectly sensible, in fact. A potential oddity is filtering a calendar view by date range, since this type of filter seems to make more sense for lists, where it's a way to shorten the lists. However, upon a bit of reflection, a date-filtered calendar view is probably just fine if it means that filtered-out items simply don't show up in whatever calendar views are showing -- which come to think of it is just the normal meaning of filtering (duh). So, for example, if we filter out all but a particular range of dates within a month, in the month view the filtered-out dates are simply displayed as blank. This sounds just fine to me. In the scenario, we'll go through all the other forms of filtering and ensure there

sensibleness in both calendar and list views. Cool.

6.45. Filtering Issues

This is potentially a rather complex area. In the style of dialog chosen, one of the key alternatives is whether the "Apply filtering to" choices are one-of radio buttons or multi-select check boxes. At first, the Emacs user in me wanted to make them check boxes, but this is really overly complex and potentially confusing to the user. Hence, for simplicity sake we changed to one-of radio buttons.

The trade-off here is a pretty straightforward of power versus simplicity. The even more specific pro/con is that with check boxes, it's easy to define the same filtering for two or more types of item, whereas with radio buttons the user must repeat the defs for each type. However, the complexity of how consistently to define what selecting one after two have been selected means is high, and any convenience gain is outweighed by the complexity. Plus if we analyze things in terms of how frequently one is likely to do this, the decision was to keep it simple. The future work section discusses the broader issue of much more powerful filtering capabilities.

Here's something that was item under the heading "sketch of ideas" in the filtering section; it may or may not be useful here:

I don't think the named filter menu is a good style of UI, primarily from a convenience standpoint. The main issue is that it requires the user first to define a filter before using it at all, when naming and storing a filter is likely to be secondary for most users. Rather, the primary thing they want to do is define a quick filter, and then after they've used it, save it if it appears to be particularly useful. So, what we need is a filtering dialog that's very simple at the top-level, and allows more advance form of filtering as a second-level option.

6.46. Back and Forth with the Monthly Recurring Functionality Notes:

1. **UNHOT FLASH:** Let's leave the "Which dates" option at the monthly level. Then yearly will be simply a boolean, with no yearly calendar displayed. Even though yearly could be specified using monthly, we'll leave it as a convenience for the user, i.e., a quick way to do a yearly thing.
2. **HOT FLASH:** Let's nix the "Which dates" option at the monthly level. It's redundant with the yearly level. This leaves us with the monthly level allowing the user to specify recurrence using the "nth week in the month" form, and the yearly level allowing the user to specify recurrence using the "list of dates" form. I think this works out well.
3. **NOT Deprecated:** Days of week versus dates selections are mutually exclusive. In terms of GUI rendering, if anything is selected in the days display, the date display is disabled (physically greyed out). If anything is typed in the date edit box, the days display is disabled.
4. Details of potential conflict between specifying some or all of 4th, 5th, and last weeks need to be clearly specified. To wit, for months with 4 weeks only, the specifications for the 5th week do not apply, and the specification for the last week, if present, override any specification for the 4th week, if present. For months with 5 weeks, the specification for the last week, if present, override any specification for the 5th week.

Per the latest thinking, there are no yearly interval details. The yearly interval is simply an on/off setting indicating that an item recurs once every year on the same date.

6.47. Maximum Date Range

Some interesting variations happen here in different systems. Claris just plain blows it. Emacs specifies that 0 is the min, and has a very large max, but blows it at some point (I've not experimented exactly where) due to int overflow error.

Yahoo calendar does not provide a 'Goto Date' of any form, so it is limited to whatever is predefined. Netscape quits at 1990 and 2037, not exactly sure why. I've yet to investigate what outlook does.

It might be argued that as long as the tool doesn't blow up, it shouldn't really matter how it behaves "on the fringe". However, I would argue that a quality tool should never output "bizarre" results, even when the results don't really matter.

6.48. Lists

28dec00 Ed. Note: I'm leaning towards losing the 'Format ...' item at the bottom. The rationale is that we can allow the user to control which columns are visible in a custom view, but don't need to for the fixed views. The problem with allowing it for the fixed views is that it'll be redundant with what you can do in the custom views, and probably confusing. There may not be a perfect solution here, so we should just get on with it.

6.49. Canonical Modeling Form

Interestingly, it looks as if the canonical simple-print form may (or could have) shed some good light on the canonical modelling form, with respect in particular to how to model recurring instances.

Appendix A. Users Manual

For organizational clarity, the users manual is presented in a separate stand-alone document. The manual contains no additional information from that presented in the functional requirements of Section 2. Rather, the manual presents Rolodex Tool functionality in a different form than the scenarios. The form is that of a software reference manual, which is terser and less tutorial than the scenario-style form of the requirements.

Appendix B. Command Language

This appendix defines the syntax of the command language used in all Calendar Tool script files, including system start-up files such as that saved by the 'File Save Config' command.

Describe the general derivation rules for commands, using related verbiage from earlier efforts along these lines, including DEMO. A new rule (I think) is provide an int-valued arg to functions that select from a dynamic-valued list, such as ViewWindows.

Here's a first crack at things. I'd like to avoid a CFG syntax if possible, but we'll see. CFG notation may ultimately be the easiest thing, but we do need to remember our audience, i.e., they're not coders. Also, this obviously needs to be updated per the most recent changes to the menus.

```

FileNew
FileOpen(file_name)
FileClose
FileCloseAll
FileSave
FileSaveAs(file_name)
FileSaveAll
FileSaveConfig(file_name)
FilePrint(print_spec)
FileExit

EditUndo
EditRedo
EditRepeat(count)
EditCut
EditCopy
EditPaste
EditDelete
EditSelectAll
EditFind(reg_expr)
EditCommand(command_string)

ScheduleAppointment(appt_info)
ScheduleMeeting(meeting_info)->(confirm_info)
ConfirmMeeting(confirm_info, selection)
ScheduleTask(task_info)
ScheduleEvent(task_info)
AddCategory(category_info)
ChangeCategory(category_name, category_info)
DeleteCategory(category_name)

ViewItem
ViewDay
ViewWeekTable
ViewWeekLists
ViewMonth
ViewYear
ViewNext
ViewPrevious
ViewGotoDate
ViewListsAppointments
ViewListsMeetings
ViewListsTasks
ViewListsEvents
ViewListsAllItems
ViewListsCustom(custom_list_name)
ViewListsCustom(menu_position)
ViewListsSelectSortField(list_field_name)
AddCustomList(custom_list_info)

```

```

ChangeCustomList(custom_list_name, custom_list_info)
DeleteCustomList(custom_list_name)
ApplyCustomList(custom_list_name)
ViewFilterHideAppointments
ViewFilterHideMeetings
ViewFilterHideTasks
ViewFilterHideEvents
ViewFilterHideCustom(custom_filter_name)
ViewFilterHideCustom(menu_position)
AddCustomFilter(custom_filter_info)
ChangeCustomFilter(custom_filter_name, custom_filter_info)
DeleteCustomFilter(custom_filter_name)
ApplyCustomFilter(custom_filter_name)
ViewOtherUser(user_identity)
ViewGroup(group_identity)
ViewWindows(window_name)
ViewWindows(menu_position)
WindowsWindowingMode(window_mode)
WindowsMagnetize

AdminUsersAdd(user_info)
AdminUsersFind(user_info)
AdminUsersChange(old_user_info, new_user_info)
AdminUsersDelete(user_info)
AdminGroupsAdd(group_info)
AdminGroupsFind(group_info)
AdminGroupsChange(old_group_info, new_group_info)
AdminGroupsDelete(group_info)
AdminLocationsAdd(location_info)
AdminLocationsFind(location_info)
AdminLocationsChange(old_location_info, new_location_info)
AdminLocationsDelete(location_info)
AdminGlobalOptionsTimesAndDates(time_and_date_options)
AdminGlobalOptionsScheduling(scheduling_options)
AdminGlobalOptionsViewing(viewing_options)
AdminGlobalOptionsAdministrative(administrative_options)

OptionsTimesAndDates(time_and_date_options)
OptionsScheduling(scheduling_options)
OptionsViewing(viewing_options)
OptionsAdministrative(administrative_options)

HelpAbout
HelpShowQuickHelp(command_name)
HelpDetailedHelpContents()->(help_contents)
Help(contents_selection)->(help_topic_info)
HelpDetailedHelpIndex(topic_name)->(help_index)
Help(index_selection)->(help_topic_info)
HelpDetailedHelpSearch(search_string, case_sensitive)->(search_topics)
Help(search_topic_selection)->(help_topic_info)

Additional commands not in menus:
SelectField(window, field_name)
ClearField(window, field_name)
EnterText(window, text_field_name)
SelectValue(window, selection_field_name)

```

Appendix C. Help Content

This appendix specifies the complete content delivered for both quick help and detailed help.

C.1. Quick Help Messages

C.2. Detailed Help Content

Appendix D. Feature Comparison Matrix

Features	Microsoft Outlook	Netscape Calendar	Qualcomm Now-Up-to-Date	FileMaker Pro	On Technology Meeting Maker	Proposed
Feature 1						
Feature 1.1						
Feature 1.1.1						
Feature 1.1.2						
Feature 1.1.3						
Feature 1.2						
Feature 1.3						
Feature 1.3.1						
Feature 1.3.2						
Feature 2						
...						

