Requirements for an Electronic Calendar Tool

Version 1.0
January 2008

Contents

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 calendar information in a variety of useful ways. Administrative users run a Calendar Tool Administration program to manage Calendar Tool databases and perform other administrative 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 addressed by the Calendar Tool project are the following:

a. to provide a reliable and easy to use software tool that manages an individual user's calendar
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. Specific problem areas are the following:

a. scheduling multi-user meetings
b. dealing effectively with overlapping events
c. providing a wide range of calendar viewing options

The Calendar Tool project addresses the requirements a particular customer. This customer needs a Calendaring tool with a collection of features not available in any single existing tool. This aspect of the problem is one of combining a set of existing features in a way that these features have yet to be combined in another product.

In addition, the development of functioning software, the Calendar Tool focuses on two additional 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 categories:

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

End users are those who use the Calendar Tool for its intended purpose. Registered end users are identified in the user database of one or more Calendar Tool central host computers. Group leaders are designated registered users who may perform group scheduling operations which affect other users' calendars. System administrators perform overall system administration functions, including user and group registration. Offline and unregistered users are those who use the Calendar Tool for an individual personal calendar, without being connected to a Calendar Tool central host.

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 interested 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 looses 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,
- Apple iCal,
- Netscape Calendar,
- Now Software Now-Up-to-Date,

- Claris Organizer (no web site available)

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.

A detailed review of each related tool follows in these subsections:

- Outlook review
- iCal review
- Netscape Calendar review
- Now-Up-to-Date review
- Claris Organizer review

Following these reviews, a summary of the features is provided in a feature comparison matrix.

1.5.1. Microsoft Outlook

... Details omitted in paper version of the example.

1.5.2. Apple iCal

... Details omitted in paper version of the example.

1.5.3. Netscape Calendar

... Details omitted in paper version of the example.

1.5.4. Now Software Now-Up-to-Date

... Details omitted in paper version of the example.

1.5.5. Claris Organizer

Claris Organizer has been in use for a long time on Apple machines. It has a good set of basic features for calendaring. Given its age, there a number of significant missing features, in particular there is no support for any kind of calendar sharing amoung multiple users. This means there is no support scheduling meetings for user groups.

Despite its age, Organizer provides fundamental calendaring functionality as well as any of the newer tools. Its features for color-coded categories and filter-based search are in fact exemplary.

Good Features:

- covers the basics fine
- provides good printing functionality
- allows overlapping items and does a pretty good job of displaying them
- provides pretty good functionality for recurring items, but monthly recurring items cannot be specified that well; for example a recurring schedule on several specific days of the month cannot be done easily
- provides good functionality for item categories, allowing naming and color coding
- provides good functionality for filtering, allowing user-defined views that can include or not include particular properties of scheduled items; for example a custom view can be defined to show only items of a particular category with titles containing a certain text pattern

- overall it’s nice and simple, without a lot of superfluous features

Bad Features:

- allows only one view of one calendar to be open at a time; this is a significant problem
- has only one level of undo/redo
- the only kind of lists are for tasks
- provides a calendar "navigator" in the form of a small monthly view that is not very useful
- provides a "special date" type of scheduled item that is not useful given that is not significantly different than the "event" type of item
- provides a "contact list" feature that is not integrated with any external address book tool, making the feature not very useful
- provides a "notes" feature that is not well integrated with the calendar; specifically, notes can have a date stamp, but are not specifically associated with scheduled items

Missing Features

- no import of other calendar formats, including no import of plain text
- no place to put the location of a scheduled item; it must be squeezed into the title
- no details or notes field for scheduled items
- no yearly view
- no multi-user features at all, including no meeting items on calendar, and necessarily no user or group administration functionality

1.5.6. Feature Comparison Matrix

... Details omitted in paper version of the example.

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. When users are connected to a Calendar Tool central host computer, they may share their personal calendar with other users, and view other users' calendars who choose to share them. Regular users may view, but not modify, the Calendar Tool databases on the central host. The databases contain information for registered Calendar Tool users, groups of users, and locations in which meetings can be scheduled.

Administrative users run a separate application called the Calendar Tool Administration program. Through this application, administrators can edit the databases and perform other administrative functions.

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

2.1. User Interface Overview

There are two separate user interfaces for the Calendar Tool system, provided by two separate application programs. The primary interface is for regular users to browse and edit their calendars. This 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 perform other administrative functions. This interface is provided by the program referred to as "Calendar Tool Administration".

2.1.1. Regular User Interface

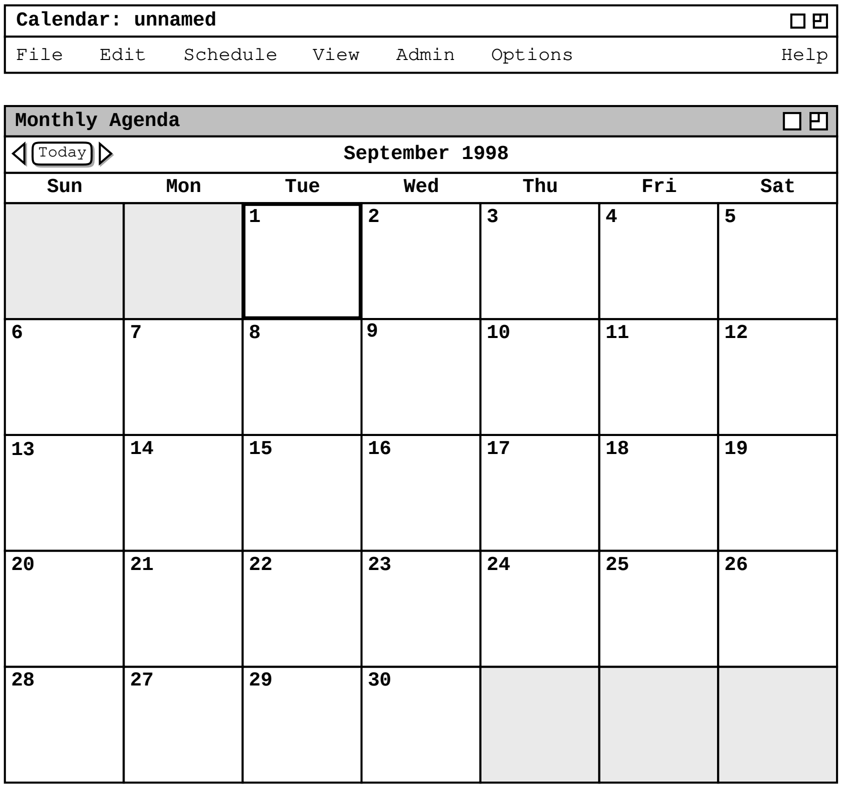
When a user initially invokes the Calendar Tool, the screen appears as shown in Figure 1. The initial display consists of two windows: a command menubar and a monthly calendar view. The name of the current calendar is displayed in the banner of the menubar, the initial default being a new unnamed calendar. 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.

The '`Load Settings`' and '`Save Settings`' commands allow the user to load and save various settings used in the Calendar Tool. There are calendar-specific settings that are individualized for each calendar, and session–wide settings that are common to all calendars.

'`Page Setup`' allows the user to set printing parameters for particular operating environments. '`File Print`' prints the contents of one or more active display windows, per the page setup parameters. '`File Exit`' exits the Calendar Tool, offering to save any modified calendars if necessary.



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 regular user.

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

The `Edit` menu contains commands for manipulating calendar data during editing. ‘`Edit Undo`’ undoes the most recently completed undoable command. ‘`Edit Redo`’ redoes the most recently undone command. There is only a single level of undo/redo, i.e., undo cannot be executed multiple times in succession to undo a series of commands.

‘`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.

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 Today`' changes the current display to contain today's date. '`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 publicly-visible 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.

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.

'`View Calendars`' lists the names of all currently open calendars. The user selects from the list to make a calendar current.

The `Admin` menu contains commands to connect to and access Calendar Tool central host computers. '`Admin Connect`' allows the user to establish central host connections. The three commands below `Connect` provide access to central–host databases -- users, groups, and locations; regular users have read-only access to the databases. The *User* database contains a record for each officially registered user of the Calendar Tool on a host. 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.

The user invokes the '`Admin Password`' command to change the password on the currently-connected central host computer. '`Contact Admin`' allows the user to communicate, via electronic mail, with a Calendar Tool administrator for any known central host.

The '`Options`' menu has commands for setting various system options. Calendar-specific options are organized in five major areas: '`Times and Dates`', '`Fonts`', '`Scheduling`', '`Viewing`', and '`Administrative`'. There are also '`Global`' options that apply to an entire Calendar Tool session, across all calendars. The '`Restore Defaults`' command resets all option settings to their built-in values as of when the Calendar Tool was initially installed.

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.

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 and a window for the Calendar Tool user database. The menubar banner contains the name of the Calendar Tool central host computer on which the administration program is running. The database dialog lists the names and IDs of registered Calendar Tool users. The user database has no pre-defined entries. There are commands to search for a user, add a new user, view the details of an existing user record, and cancel the user database dialog.

Figure 4 shows an expansion of the command menus for the Calendar Tool Administration program. The '`File`'

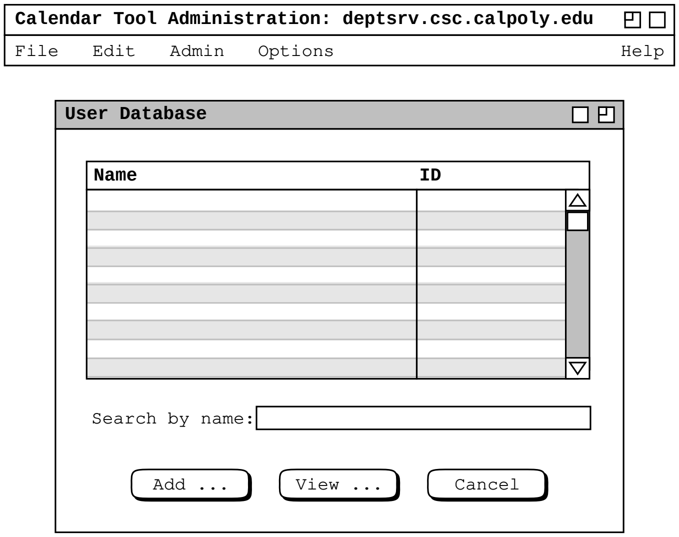


Figure 3: Initial screen for system administrator.

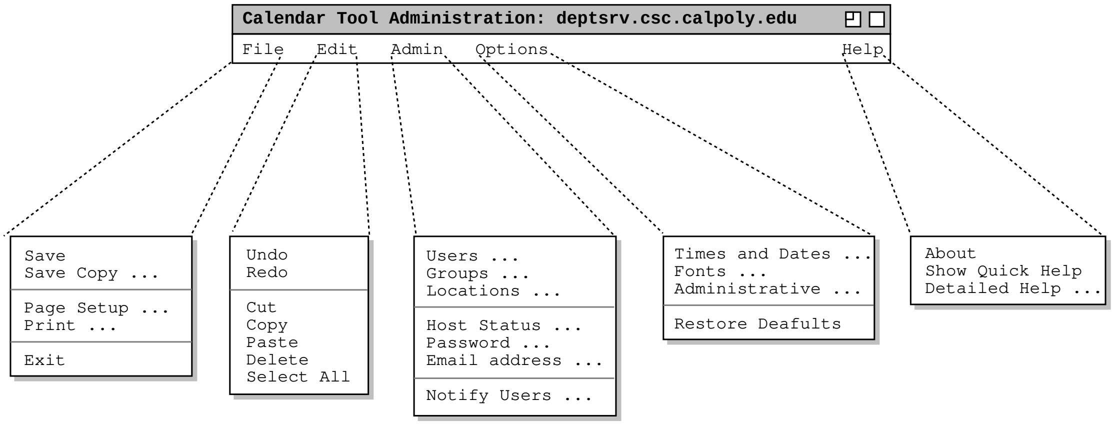


Figure 4: Expanded command menus for the administrative interface.

menu has a reduced set of commands, since databases and other administrative data are stored in specific files in a fixed location. The '`File Save`' command saves all data changes performed by the administrative user. '`File Save Copy`' saves a backup copy of administrative data in a selected file directory.

'`Page Setup`' allows the administrator to set printing parameters for particular operating environments. '`File Print`' prints the contents of one or more active display windows, per the page setup parameters. '`File Exit`' exits Calendar Tool Administration, offering to save if necessary.

The '`Edit Menu`' in the administrative interface is a proper subset of that in the regular user interface. The edit commands that are present have the same functionality as in the regular interface, adapted as appropriate for administrative data.

The '`Admin`' menu provides access to the three databases. Administrative users can add, change, and delete database information. The '`Host Status`' command allows the administrator to view and edit information about the Calendar Tool central host computer. The '`Password`' command is used to set the administrative access password. '`Email Address`' is used to set the email address through which regular users contact an administrator.

'`Notify Users ...`' provides the means to send a message to some or all Calendar Tool users.

The '`Options`' menu has a subset of the commands available in the regular-user options menu. The '`Times and Dates`' and '`Fonts`' options are the same as for the regular user. The '`Administrative`' options are particular to the administrative user. '`Restore Defaults`' serves the same purpose as in the regular-user interface.

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:

a. It is presented in a form independent of any particular computer operating system or platform.
b. The layout style of the interface is generally in separate independent windows, rather than one or a few windows with tiled or stacked information.
c. 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.
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

In the online version of this document, click here to view a hierarchical map of user interface screens for regular users. Click on any thumbnail in the screen map to view the full-size version of the screen, in the context of its initial appearance in the requirements.

Due to its large size, the screen map is not available in the printed version of this document.

2.2. Basic Appointment Scheduling

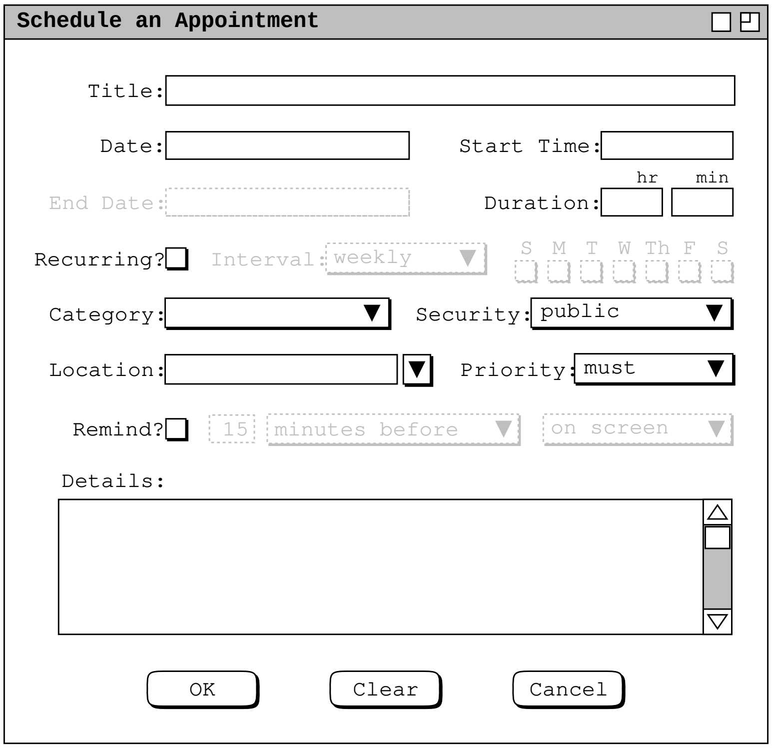
This introductory scenario shows an individual user scheduling two simple appointments. Appointment scheduling is one of the most commonly performed operations with the Calendar Tool. In this scenario, the user is assumed to have just been registered in the Calendar Tool. Hence the user has a brand new 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 5. 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. ‘`Security`’ indicates how much of the appointment 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



Interval: weekly
Security: public
Priority: must
Remind? ☐ 15 minutes before on screen

Figure 5: Appointment scheduling dialog.

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

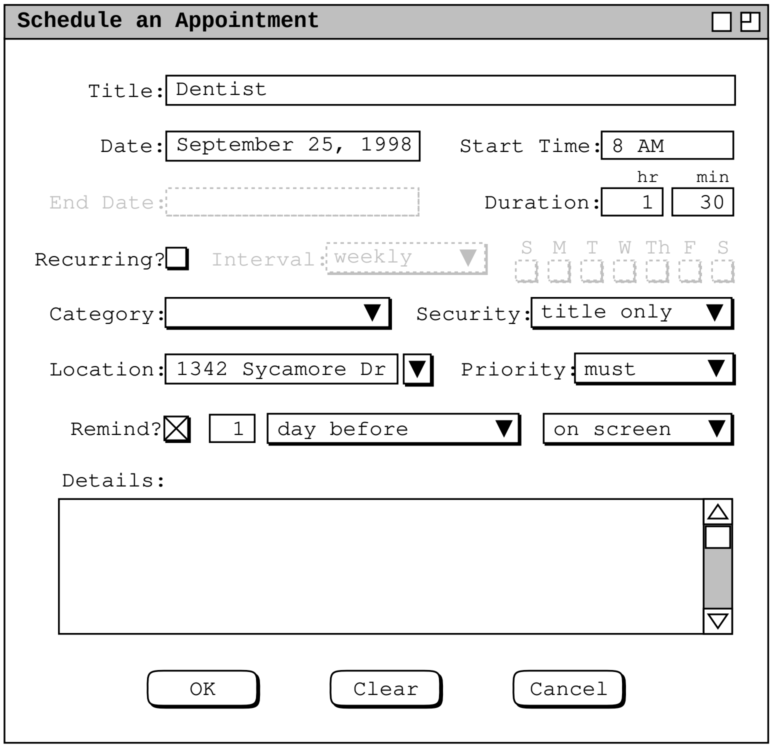
respectively.

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 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 6 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.



Title: Dentist

Date: September 25, 1998

Start Time: 8 AM

End Date:

Duration: hr 1 min 30

Recurring? ☐

Interval: weekly

Category:

Security: title only

Location: 1342 Sycamore Dr

Priority: must

Remind? ☒ 1 day before on screen

Details:

Figure 6: Scheduling a dentist appointment.

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 7. 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 8. The user then selects the '`Add`' button. In response, the system displays the dialog in Figure 9. To create the new '`personal`' category, the user types into the '`Category Name:`' text box, and then selects

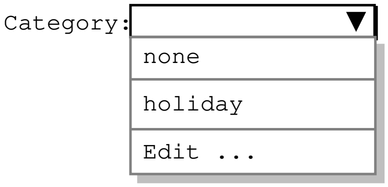


Figure 7: Initial categories menu.

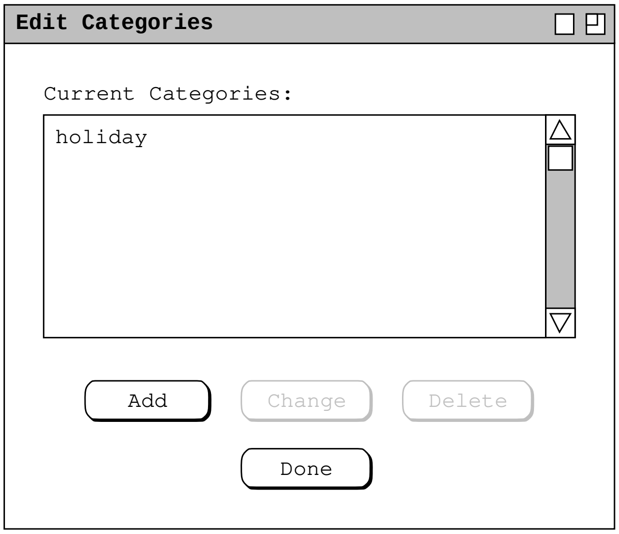


Figure 8: Edit categories dialog.

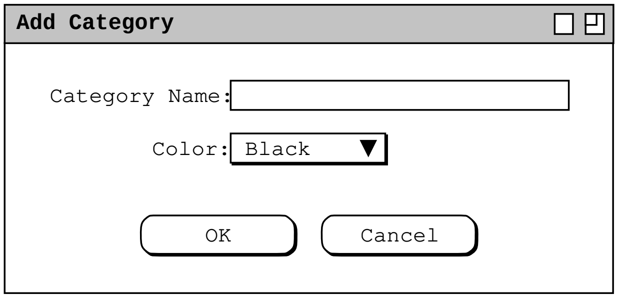
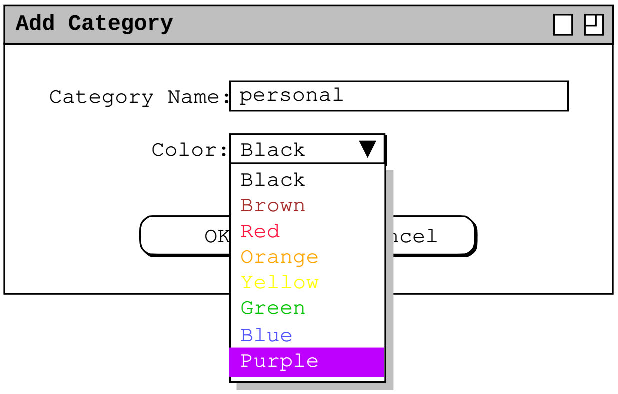


Figure 9: Add category dialog.

'`Purple`' from the color selection menu, as shown in Figure 10. 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 (Figure 8). At this point, the system adds the purple-colored text for the newly created category, as shown in Figure 11.

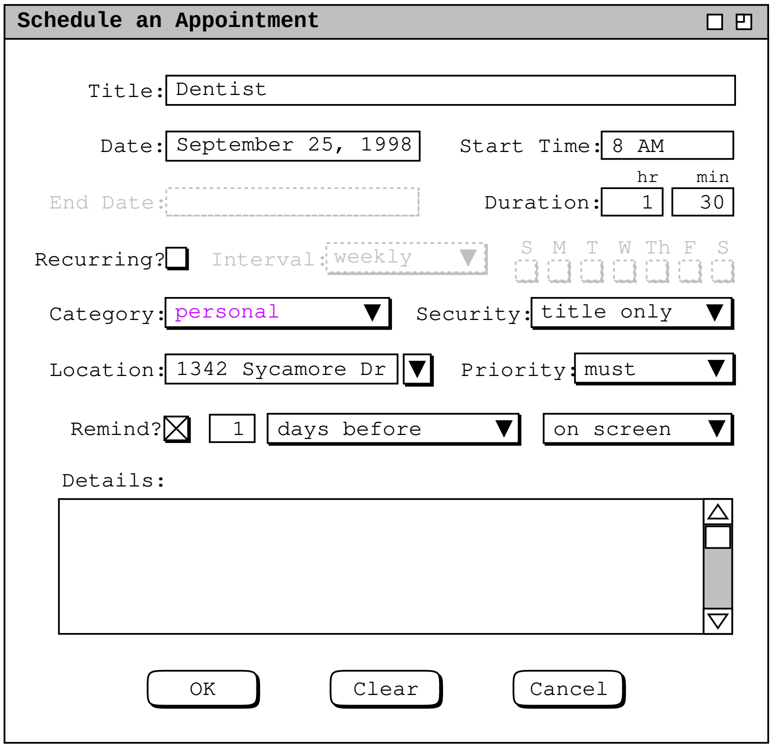
To confirm the scheduling of the dentist appointment, the user presses the '`OK`' button in the main schedule-appointment dialog. At this point, the dentist appointment is added to the user's personal calendar.


Add Category

Category Name: personal

Color: Black

Figure 10: Creating a new category.


Schedule an Appointment

Title: Dentist

Date: September 25, 1998

Start Time: 8 AM

End Date:

Duration: hr 1 min 30

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:

Figure 11: Dentist appointment with category filled in.

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 12. 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.

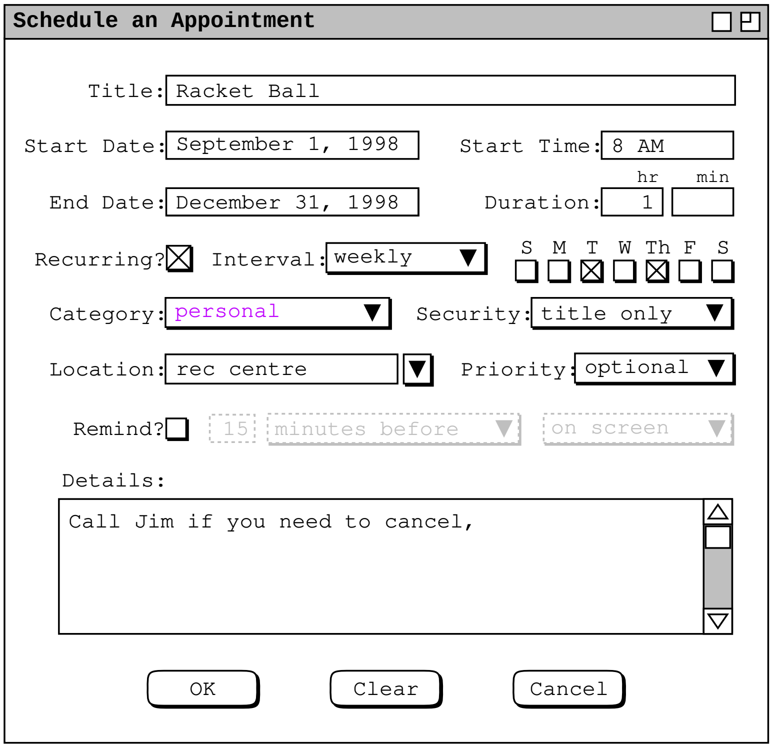
The user has set the priority for the new appointment to '`optional`' instead of '`must`'. The operative effect of selecting '`optional`' priority relates to scheduling meetings, which is discussed further in Section 2.4.1. Aside from meeting scheduling, the meaning of '`must`' versus '`optional`' is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

When the user confirms a recurring appointment with the '`OK`' button, the system adds as single new item to the calendar. The addition of a single recurring appointment *effectively* schedules multiple appointment instances, based on the '`recurring`' settings. For example in the case of the racket ball appointment in Figure 12, the system effectively schedules 36 appointment instances on the Tuesday and Thursday dates between September 1 and December 31.

The idea of an *effectively scheduled* instance is that a single recurring item is stored in the calendar, but multiple instances appear in any calendar view. The instances appear on all dates that meet the values entered by the user in the '`Recurring`' settings. Section 2.5.3.5 discusses further details of recurring items and their effectively-scheduled instances.

The information entered for a recurring appointment initially applies to all instances of the appointment. Subsequently, the user can change some or all of the information in 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 sets all data-entry fields to their default states. Details of default settings are covered in Section 2.7.3.1 To



Schedule an Appointment

Title: Racket Ball

Start Date: September 1, 1998

Start Time: 8 AM

End Date: December 31, 1998

Duration: hr 1 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,

Figure 12: Scheduling a second appointment.

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 .

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 2.8.4

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:

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

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

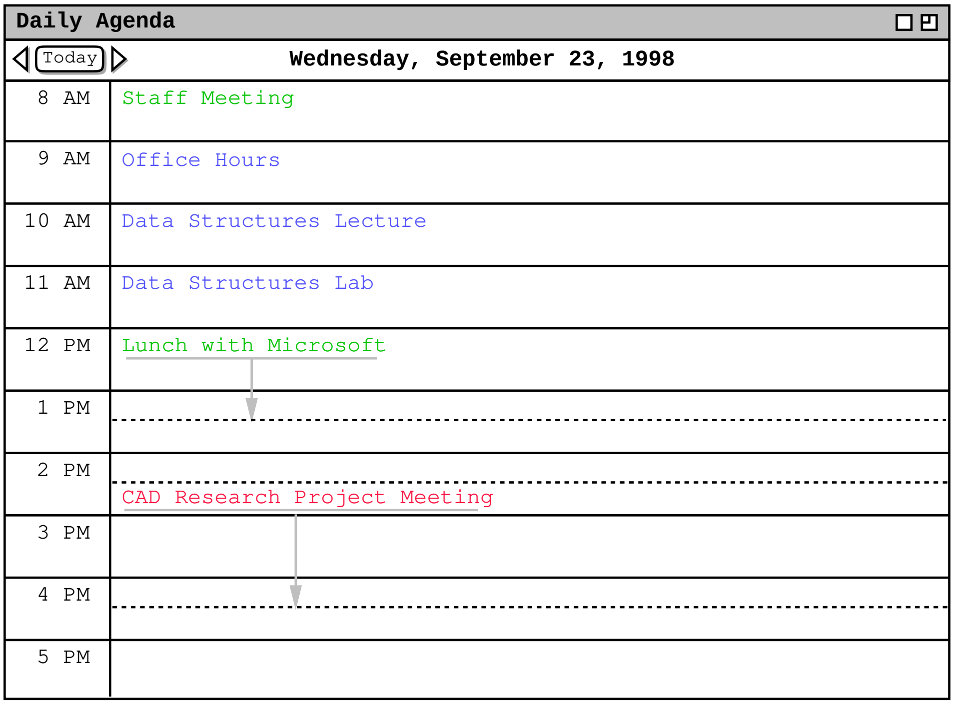
2.3.1.1. Day View

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 13. 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 13, the user is viewing the details of Wednesday, September 23, which is today's date in this scenario. The items shown in Figure 13 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.

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:

a. 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 13 starts on the hour, and is thus displayed at the top of the 8AM to 9AM time slot.



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	

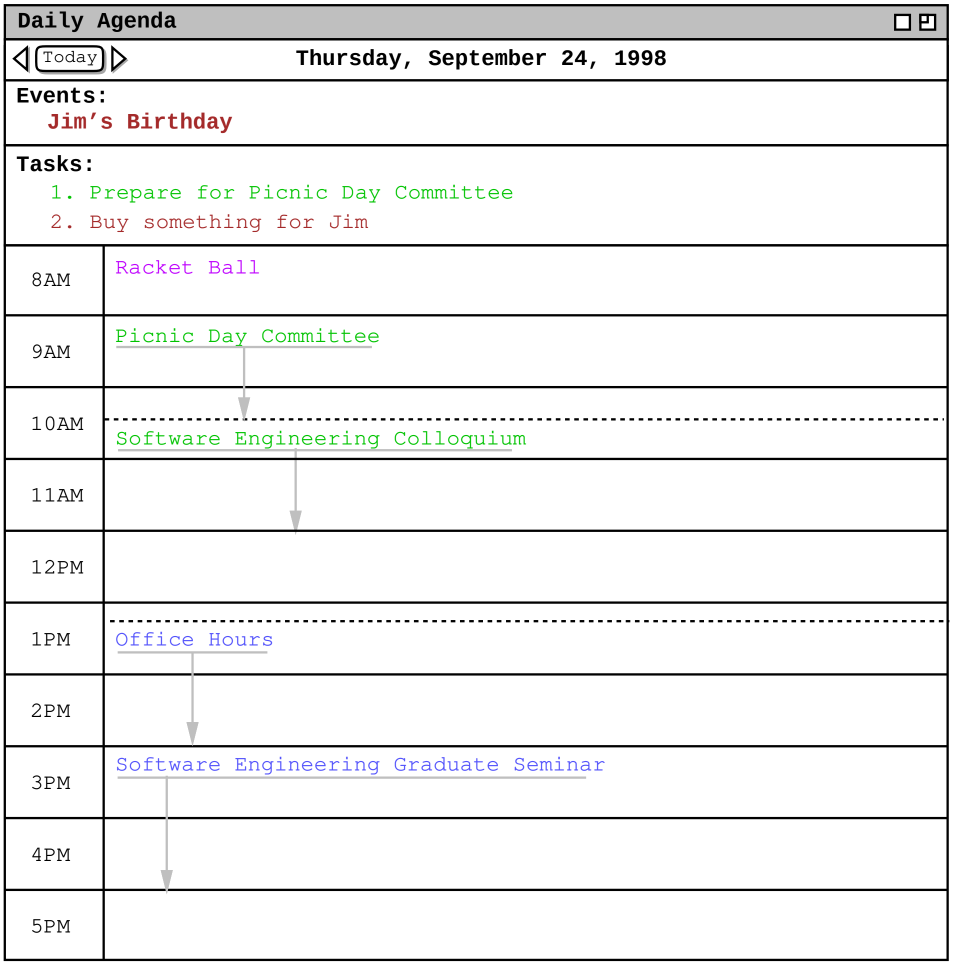
Figure 13: Day view.

b. 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.

c. 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.

d. 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 13, the display changes to that shown in Figure 14. Figure 14 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:



Daily Agenda

Thursday, September 24, 1998

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 14: Result of pressing the next day arrow.

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

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 13 shows the case where no events or tasks are scheduled for a day, whereas Figure 14 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 .

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

As an example of alternate option settings, Figure 15 shows a day view of the same information shown in Figure 13, 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 15 shows a fully expanded window, with all 24 hours of the day visible. If the user shrinks the window

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 15: Day display with non-default option settings.

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 . Specific details on the effect of all the display options are presented in Section 2.7.4.

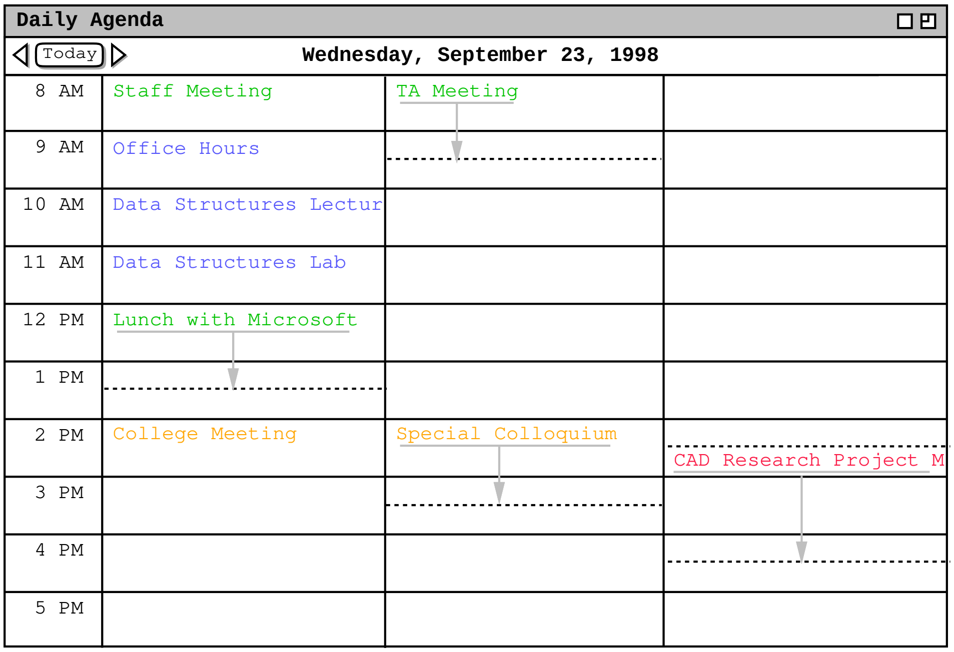
To illustrate the display of overlapping items, Figure 16 shows an updated version of the daily schedule shown in Figures 13 and 15. Here the user has added three new items, with times that overlap with previously scheduled items. The new items are a "TA Meeting" from 8 - 9:30 AM, a "College Meeting" 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 16. The precise layout rules for such cases are covered in Section .

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 .

Figure 17 shows the same information as Figure 16, 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

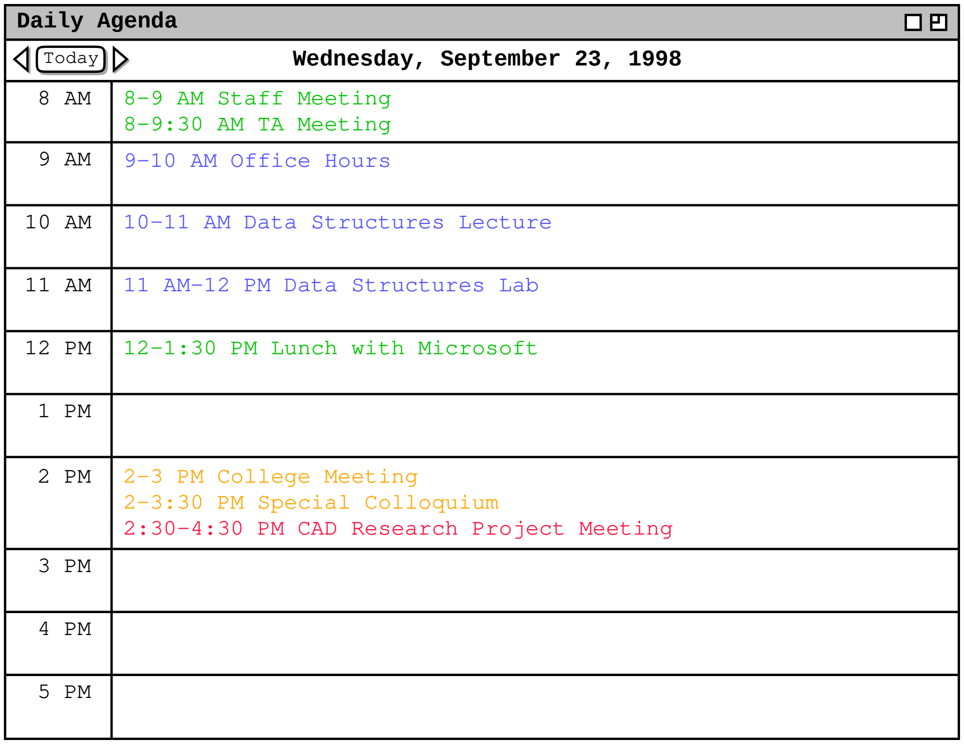


Daily Agenda

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 16: Day view with overlapping times.



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 17: Day view with overlapping times, displayed vertically.

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.

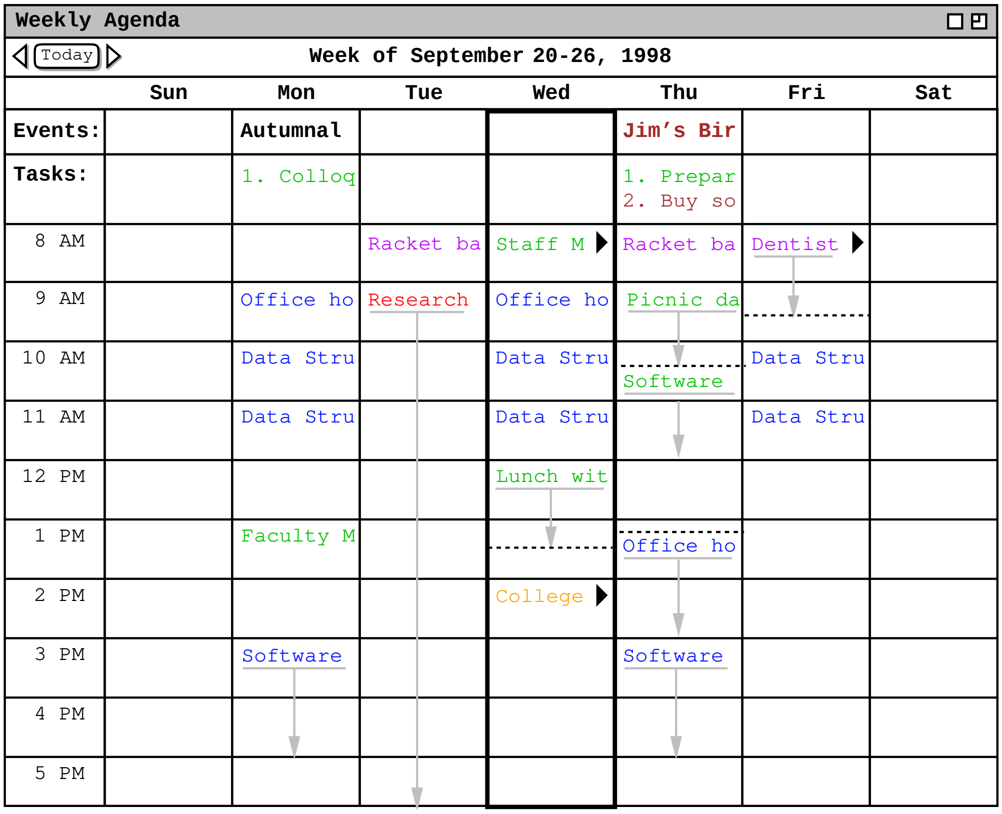
Of note in Figure 17 is the automatic resizing of the 2PM time slot from two to three rows, even though the current setting of the *Increment Height* option is 2. Whenever the number of items in a time increment is greater than the current increment height, the system automatically increases the height 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 18.

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.

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,



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	Software	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 18: Table-style week view.

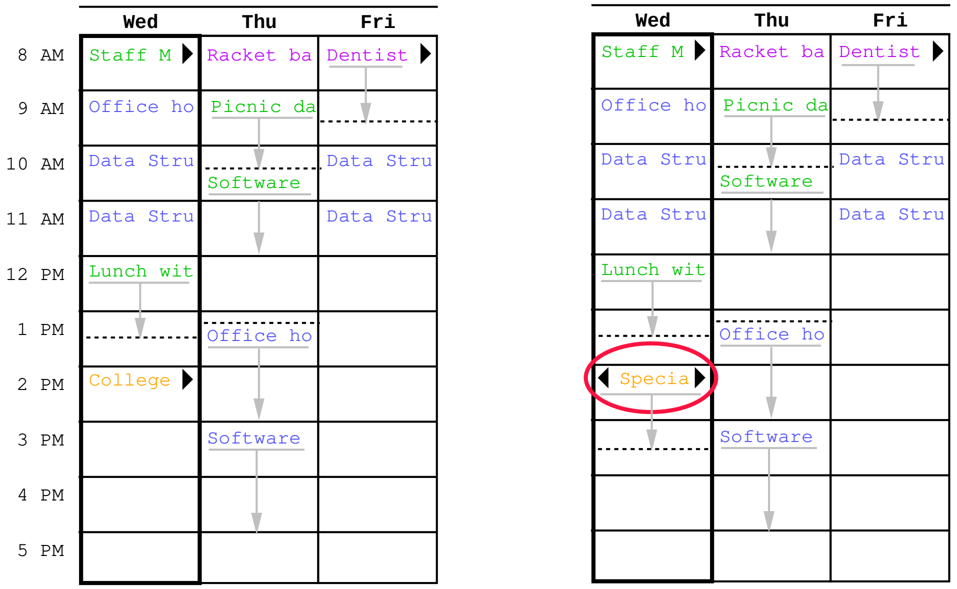
and a '`Today`' button to return to the week containing today's date. As with the day view, if there are no events or tasks scheduled for the entire week, then the event and/or task display areas are absent.

A notable aspect of the weekly table view is the display format for days that contain overlapping items. In the example shown in Figure 18, 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 19 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 19a, 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.

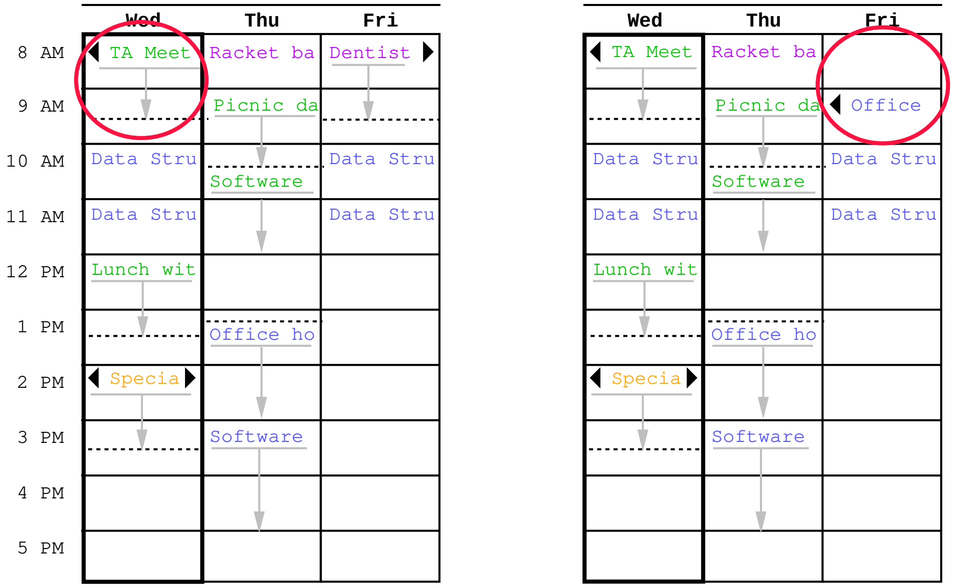
When the user clicks on the right-pointing arrow at Wednesday 2 PM, the list of 2 PM overlapping items scrolls to the left. The result is shown in Figure 19b, with the changed item circled. Figure 19c shows the result of the user clicking on the right-pointing arrow at Wednesday 8 AM. The display change is again circled. Of note in Figure 19c 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 19d 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 all in 19d, even though it would fit partially in the display. Figure 19d also illustrates that the left-pointing arrow always appears at the top of the item that is displayed. Hence, the left-pointing arrow in 19d appears next to the 9 AM office hour appointment, not in the empty 8AM slot where the Dentist appointment previously appeared.



a. Overlapping items to the right.

b. Pressed right arrow at Wed 2 PM.



c. Pressed right arrow at Wed 8 AM.

d. Pressed right arrow at Fri 8 AM.

Figure 19: Clarification of overlap in a weekly table view.

When necessary, a left-pointing arrow appears to allow scrolling back. Examples of left-pointing arrows are shown in Figures 19b, 19c, and 19d. The 2 PM item in Figures 19b-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 19c and the Friday 9 AM item in Figure 19d.

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 20. The Figure shows the same week as in the table-style view in Figure 18. 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.

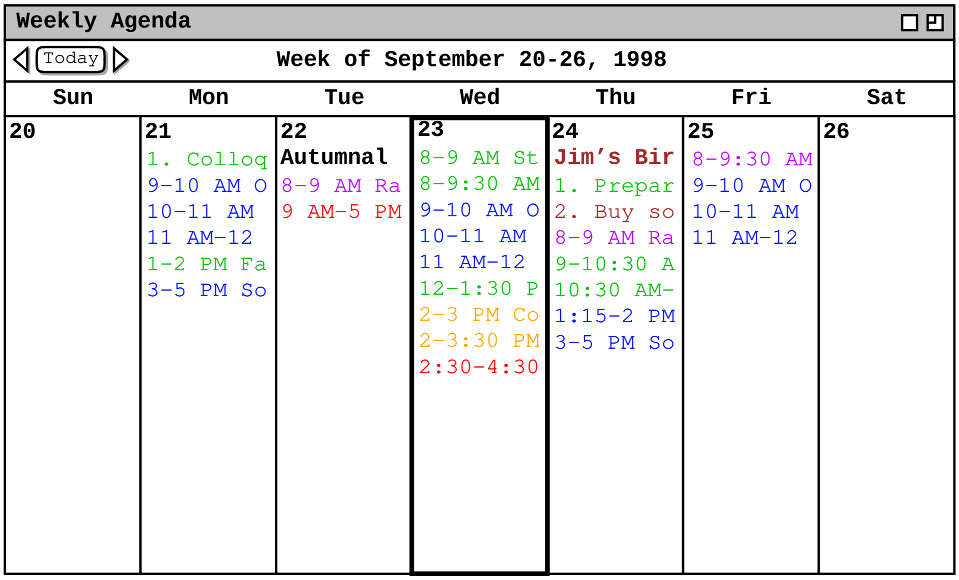
For each day of the week, the items for that day are listed consecutively in the column for that day. The list starts with bold-faced event titles, if any, sorted alphabetically by title. Following the events are the scheduled tasks, if any. Tasks are numbered starting from 1, sorted first by due time, 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*.

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.



Week of September 20-26, 1998

Sun	Mon	Tue	Wed	Thu	Fri	Sat
20	21	22	23	24	25	26
	1. Colloq	**Autumnal**	8-9 AM St	**Jim's Bir**	8-9:30 AM	
	9-10 AM O	8-9 AM Ra	8-9:30 AM	1. Prepar	9-10 AM O	
	10-11 AM	9 AM-5 PM	9-10 AM O	2. Buy so	10-11 AM	
	11 AM-12		10-11 AM	8-9 AM Ra	11 AM-12	
	1-2 PM Fa		11 AM-12	9-10:30 A		
	3-5 PM So		12-1:30 P	10:30 AM-		
			2-3 PM Co	1:15-2 PM		
			2-3:30 PM	3-5 PM So		
			2:30-4:30			

Figure 20: List-style week view.

2.3.1.3. Month View

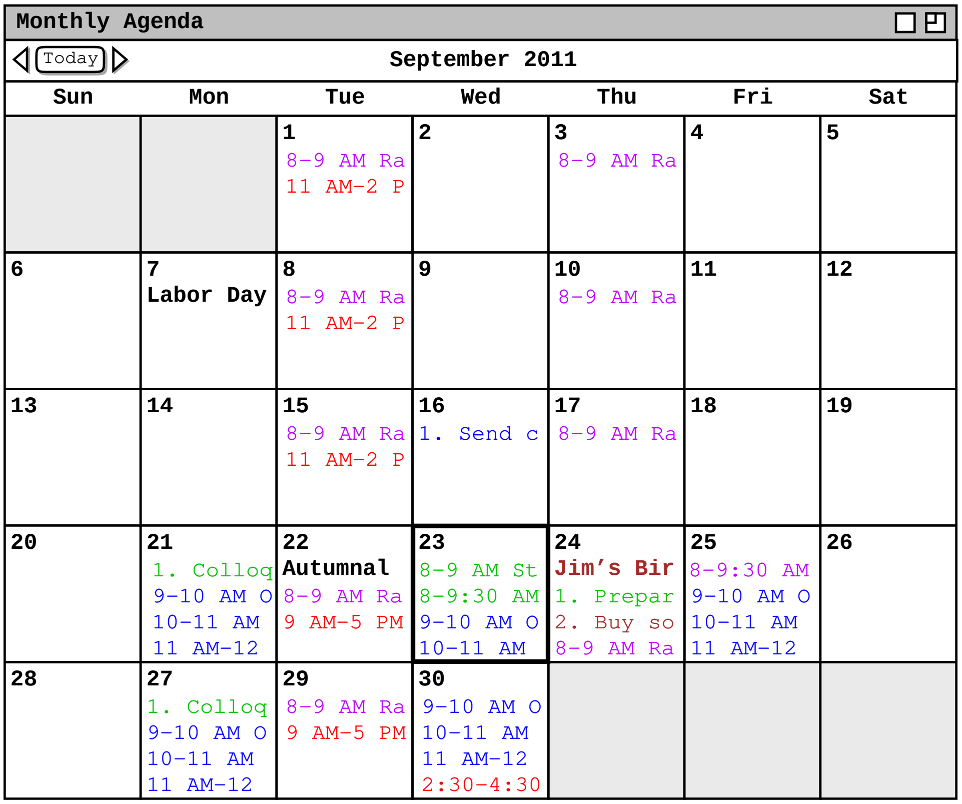
An example of the month view was shown originally in the initial screen configuration (Figure 1), since the month view is the standard default. It is chosen explicitly with the ‘`Month`’ command from the ‘`View`’ menu. Figure 21 shows the result of the user making this selection for the month of September 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 week view. That is, events are listed first, followed by tasks, followed by co-mingled appointments and meetings. Display details and sorting orders are as described for the week view, following Figure 20.

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 21), the scrollbar appears as shown in Figure 22. 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.

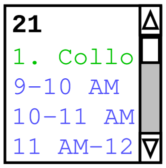
A subset of weekly list-style display options are applicable to the month view. These options are:

a. *Show/Hide Title*.



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 21: Month view.



21
1. Collo
9-10 AM
10-11 AM
11 AM-12

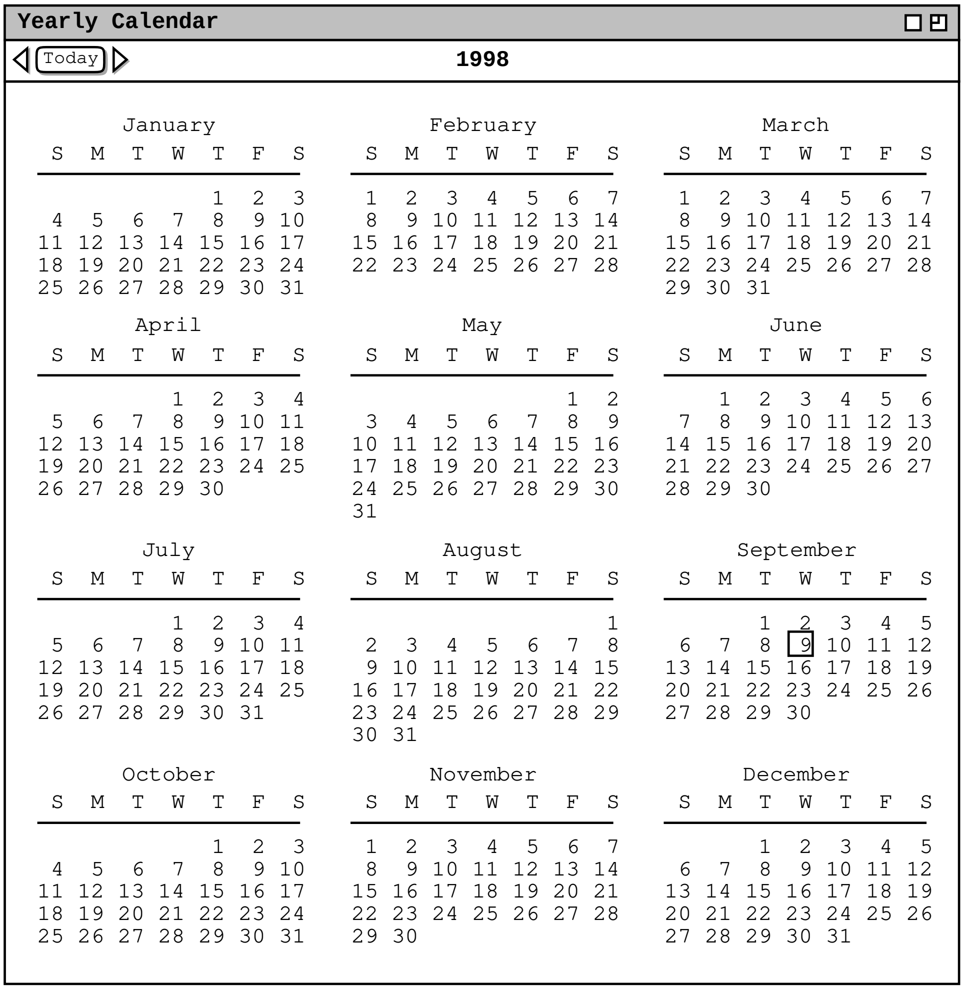
Figure 22: Transient scrolling in a month view.

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 23. The current date is highlighted with a rectangular border. Immediately below the window banner the year number, with arrow buttons to go to the previous and next



1998

January

S	M	T	W	T	F	S
				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

February

S	M	T	W	T	F	S
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

March

S	M	T	W	T	F	S
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				

April

S	M	T	W	T	F	S
			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		

May

S	M	T	W	T	F	S
					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						

June

S	M	T	W	T	F	S
	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				

July

S	M	T	W	T	F	S
			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	

August

S	M	T	W	T	F	S
						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					

September

S	M	T	W	T	F	S
		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			

October

S	M	T	W	T	F	S
				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

November

S	M	T	W	T	F	S
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					

December

S	M	T	W	T	F	S
		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		

Figure 23: Year view.

years, and a ‘`Today`’ button.

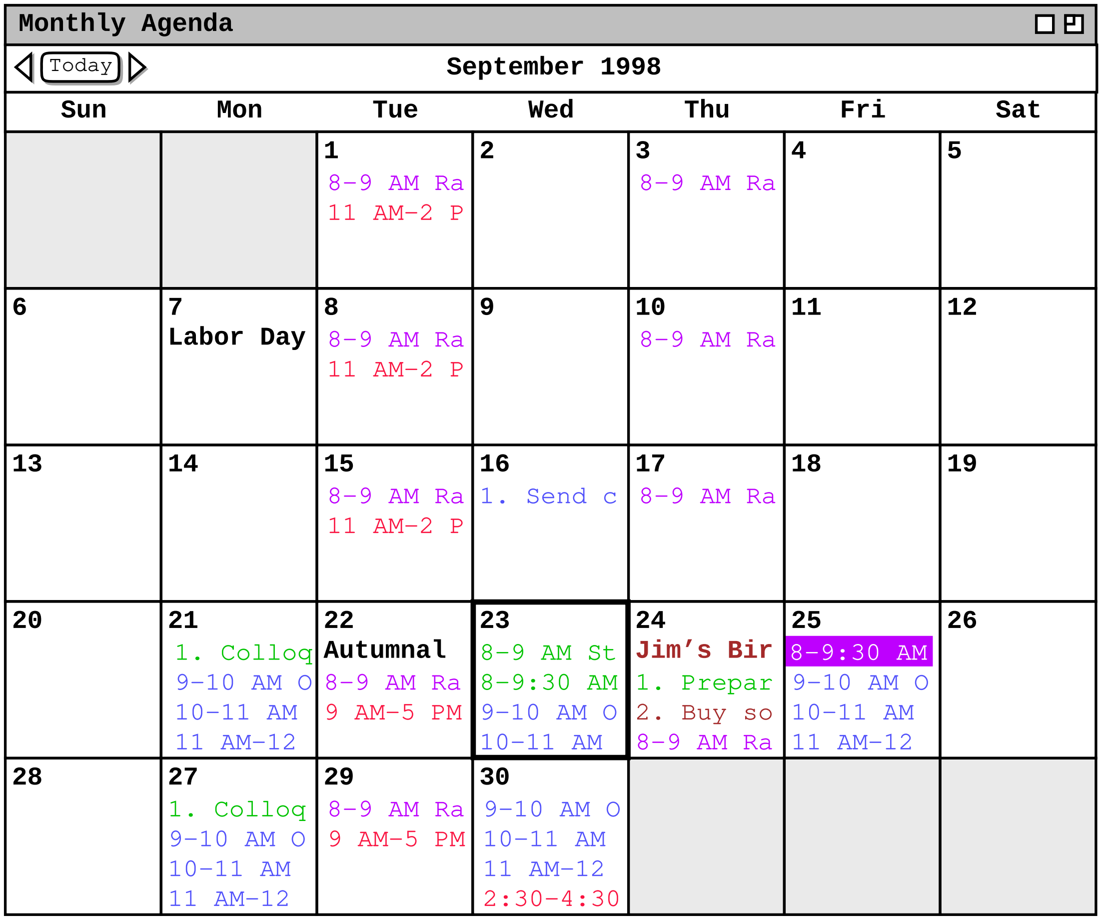
Unlike the smaller-grain views, the yearly view shows no information about scheduled items. The user may select any date as the target of another view command to show item details. The only display option that applies to the yearly view is *Default Display Width* and *Default Display Height*.

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 24 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 25. 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.

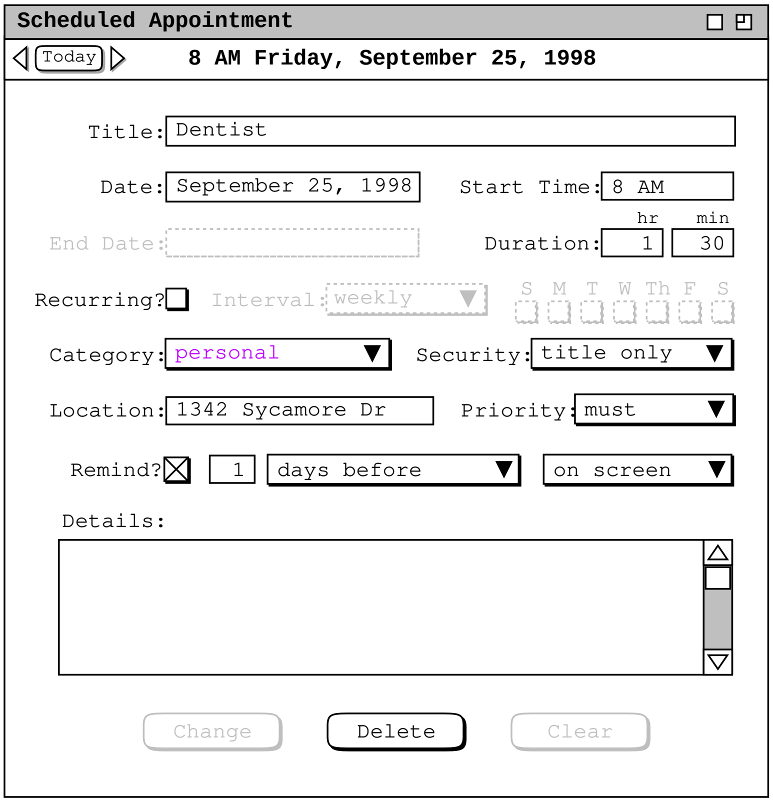
The information displayed for an item is editable by the user. This editability distinguishes the item-level view from the larger-grain daily, weekly, and monthly views. That is, none of the larger-grain displays allows the user to edit any of the displayed item information.



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 24: Item selected in monthly view.


Scheduled Appointment

8 AM Friday, September 25, 1998

Title: Dentist

Date: September 25, 1998

Start Time: 8 AM

End Date:

Duration: hr 1 min 30

Recurring? ☐

Interval: weekly

Category: personal

Security: title only

Location: 1342 Sycamore Dr

Priority: must

Remind? ☒ 1 days before on screen

Details:

Figure 25: Details of selected scheduled item.

The appointment item display in Figure 25 has the same data fields as the appointment scheduling dialog shown in Figure 5. The item display differs from the scheduling dialog as follows:

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 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 13 to Figure 14.

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 13 , 18 , 21 , 23 , and 25 . 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 25), 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 20. 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 '`Prevous`' 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 26. The user enters the desired date and presses the '`OK`' button to proceed. For example, Figure 27 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 27, the system updates the display as shown in Figure 28.

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 27, the system updates the display as shown in Figure 29.

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.

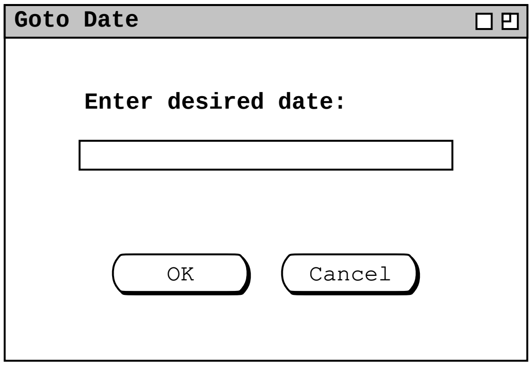
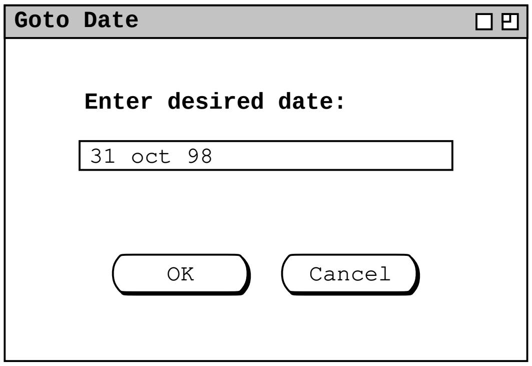
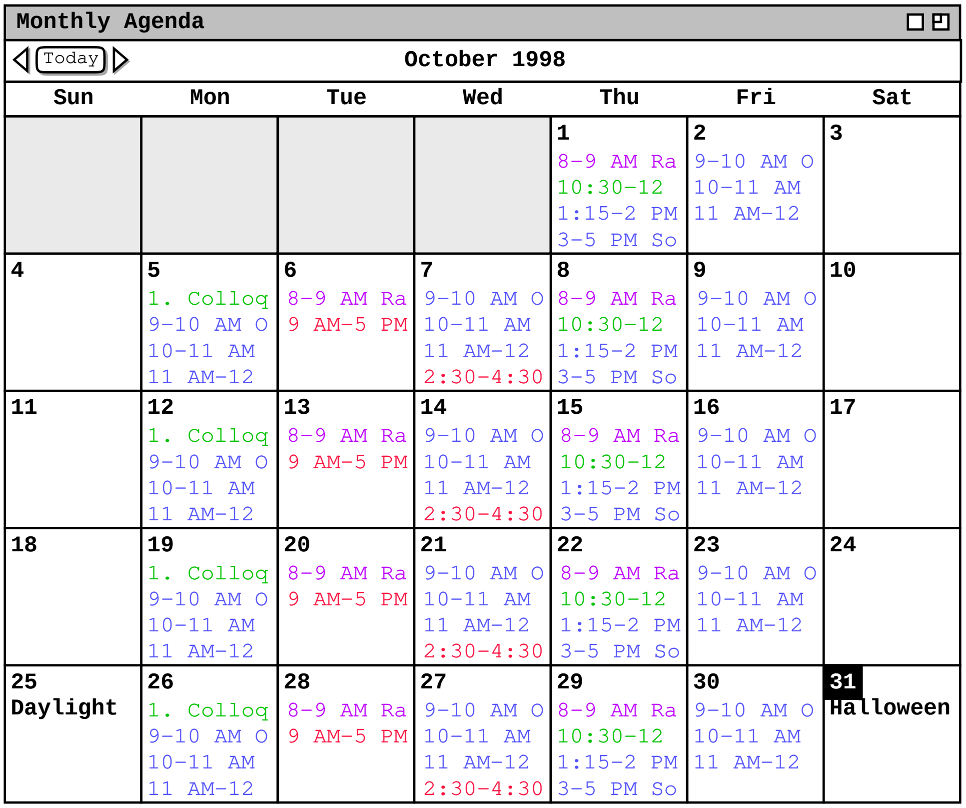


Figure 26: Goto date dialog.



31 oct 98

Figure 27: Goto date 31 Oct 98.

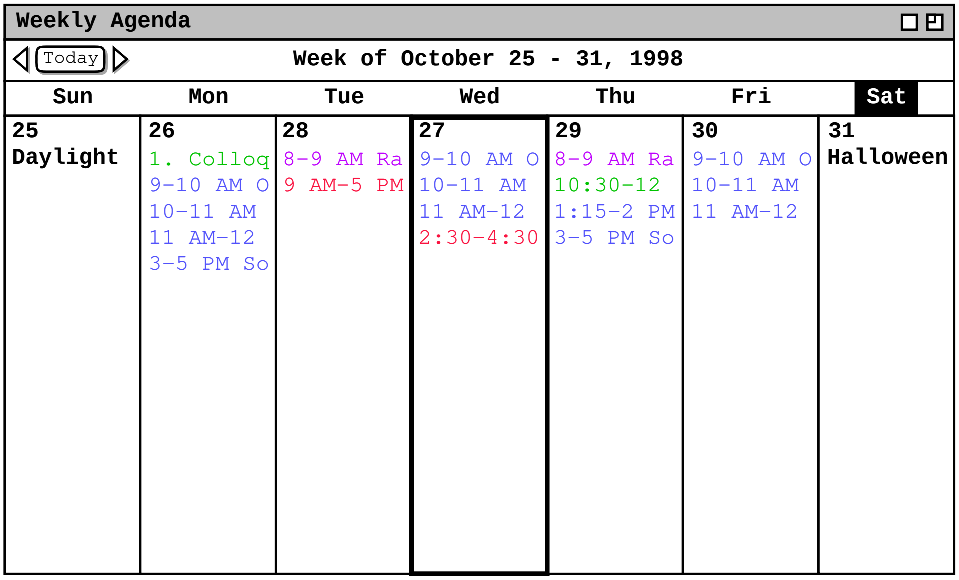


Monthly Agenda

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 28: Goto date 31 Oct 98 at the Month Level.



Weekly Agenda

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 29: Goto date 31 Oct 98 at the Week Level.

e. If there is no calendar view window on the screen, the system creates a new window at the standard start-up 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 . If the user enters an improperly formated 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 .

2.3.2.4. 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 .

2.3.3. 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 30.

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.

Figure 30: View lists submenu.

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 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 .

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 31 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 20 items. These default settings can be changed using the options commands described in Section 2.7.4.1.

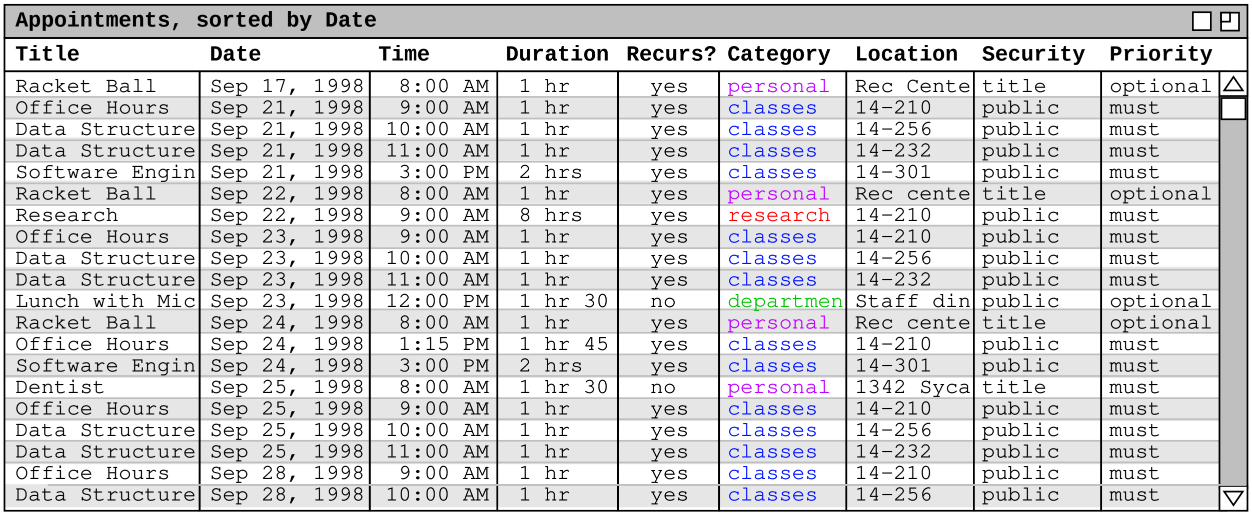
For the ongoing scenario, the current date is September 23. This means that the three-week period covers the dates September 16 through October 6, inclusive. Since there are more than 20 appointments scheduled in this period, the list window can be scrolled or resized to view the additional items.

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. 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.

b. A list item has only a yes/no indication of whether the appointment recurs, not the full recurring information.

c. A list item contains no remind information or details.



Appointments, sorted by Date

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 31: Basic Appointments List.

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 31 the system updates the display to that shown in Figure 32, where the list items are sorted alphabetically by title. The precise sorting rules for each of columns are as follows:

a. 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.

b. `Date` is sorted in ascending chronological order.

c. `Time` is sorted in ascending time-of-day order.

d. `Duration` is sorted is in ascending order from shortest to longest duration.

e. `Recurs` is sorted with all ‘`yes`’ values before all ‘`no`’ values.

f. `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.

g. `Location` is sorted in the same manner as `Category`.

h. `Security` is sorted in the order `public`, `title`, `protected`, `private`.

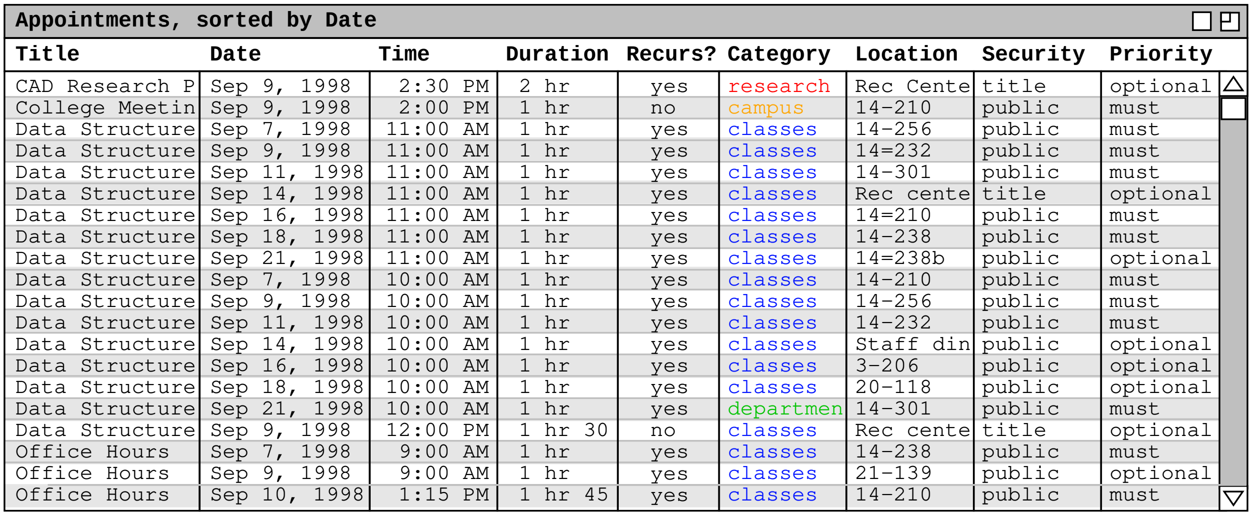
i. `Priority` is sorted with `must` values before `optional` values.

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 32 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.



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 32: Appointments list sorted by title.

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

Table 1: Sorting suborders for appointment and meeting lists.

2.3.3.2. Meeting Lists

To view a list of scheduled meetings, the user selects the '`View Lists Meetings`' menu command. Figure 33 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.

The default sorting order for meeting lists is the same as for appointments. The '`Sched By`' column is sorted lexically in the same manner as '`Title`'. When '`Sched By`' is used as the primary sorting key, the secondary through quintinary sorting keys are, respectively: Date, Time, Duration, and Title.

Three radio buttons at the bottom of the display control the visibility of meetings based on whether a meeting is scheduled or penciled-in. As described in Section 2.4.1, a scheduled meeting is one that the user has scheduled herself, or accepted from another user. A penciled-in meeting is one scheduled by another user but not yet accepted by a receiving user. Specific details of accepting externally-scheduled meetings are covered in Section 2.4.1.5.

2.3.3.3. Task Lists

When the user selects the '`Tasks`' command from the '`View Lists`' menu, the system displays a window of the form shown in Figure 34. The list reflects the fact that there are only six tasks scheduled in the default three-week list period.

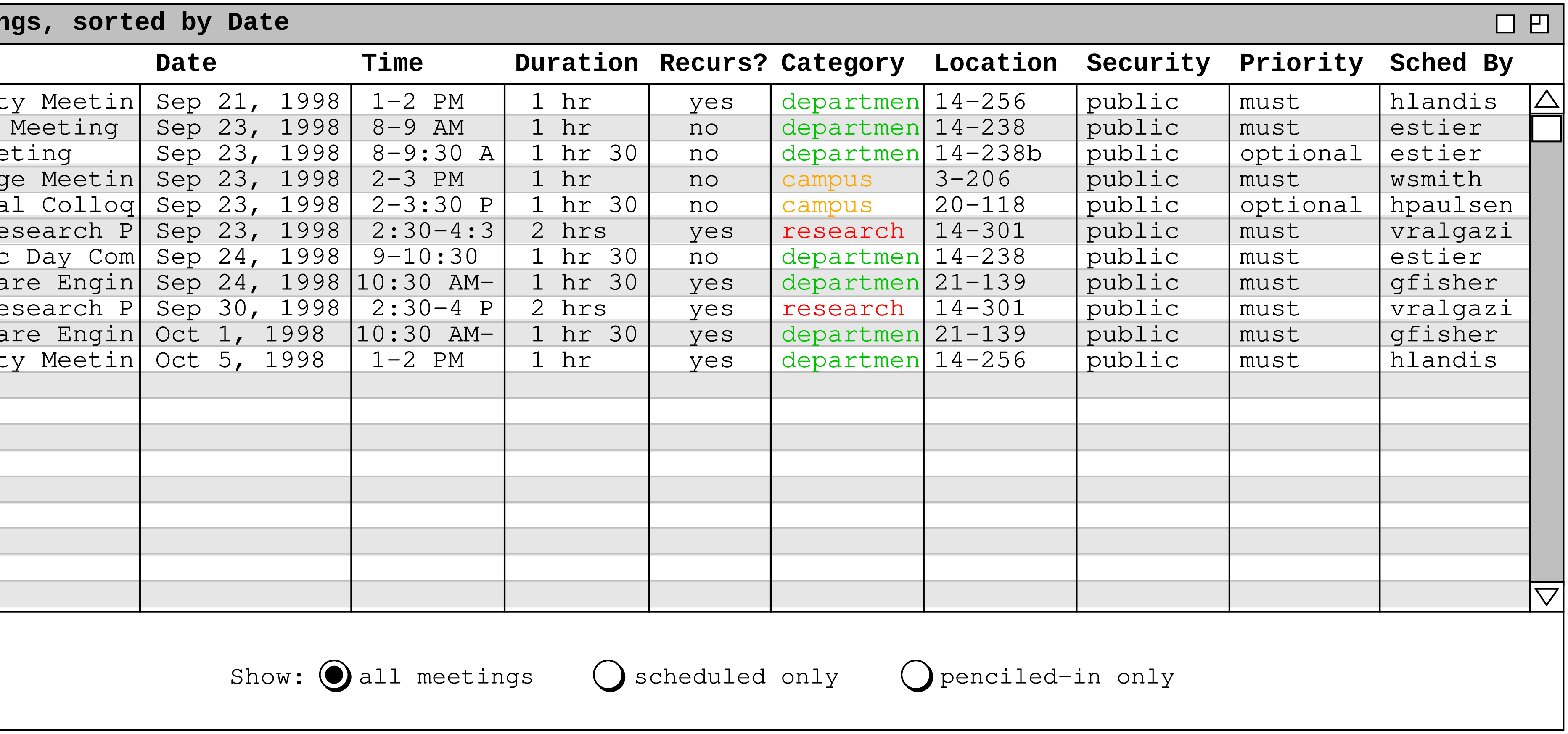
By default, the task list is sorted primarily by due date, secondarily by due time, thirdly by title. Tasks without a user-specified due time are considered to have a time after those with user-specified due times.

As for the appointments list, the task list is an abbreviation of the full information for scheduled tasks. Specifically,

a. There is a separate list item for each instance of a recurring task.
b. A list item has only a yes/no indication of whether the task recurs, not the full recurring information.
c. A task list has a completion date column, but no separate checkbox to indicate task completion; a blank completion date in the list indicates the task is not yet completed.
d. A list item contains no remind information or details.

Also as with appointments lists, the user can change the default order of the list by clicking on any of the column labels that appear in the display. For list columns that are common to both appointments and tasks, the precise sorting rules for tasks lists are the same as those given above for appointment lists. Sorting rules for the other task list columns are as follows:

a. `Due Date` is sorted in ascending chronological order.
b. `Due Time` is sorted in ascending time-of-day order for those tasks with a user-specified due time; tasks

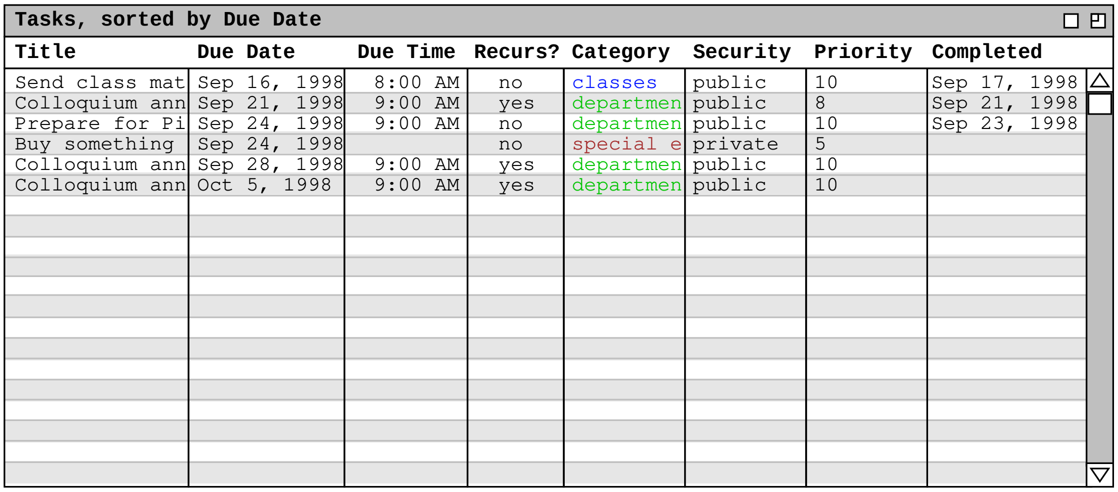


ngs, sorted by Date

	Date	Time	Duration	Recurs?	Category	Location	Security	Priority	Sched By
ty Meetin	Sep 21, 1998	1-2 PM	1 hr	yes	departmen	14-256	public	must	hlandis
Meeting	Sep 23, 1998	8-9 AM	1 hr	no	departmen	14-238	public	must	estier
eting	Sep 23, 1998	8-9:30 A	1 hr 30	no	departmen	14-238b	public	optional	estier
ge Meetin	Sep 23, 1998	2-3 PM	1 hr	no	campus	3-206	public	must	wsmith
al Colloq	Sep 23, 1998	2-3:30 P	1 hr 30	no	campus	20-118	public	optional	hpaulsen
esearch P	Sep 23, 1998	2:30-4:3	2 hrs	yes	research	14-301	public	must	vralgazi
c Day Com	Sep 24, 1998	9-10:30	1 hr 30	no	departmen	14-238	public	must	estier
are Engin	Sep 24, 1998	10:30 AM-	1 hr 30	yes	departmen	21-139	public	must	gfisher
esearch P	Sep 30, 1998	2:30-4 P	2 hrs	yes	research	14-301	public	must	vralgazi
are Engin	Oct 1, 1998	10:30 AM-	1 hr 30	yes	departmen	21-139	public	must	gfisher
ty Meetin	Oct 5, 1998	1-2 PM	1 hr	yes	departmen	14-256	public	must	hlandis

Show: ● all meetings ○ scheduled only ○ penciled-in only

Figure 33: Meetings List.



Tasks, sorted by Due Date

Title	Due Date	Due Time	Recurs?	Category	Security	Priority	Completed
Send class mat	Sep 16, 1998	8:00 AM	no	classes	public	10	Sep 17, 1998
Colloquium ann	Sep 21, 1998	9:00 AM	yes	departmen	public	8	Sep 21, 1998
Prepare for Pi	Sep 24, 1998	9:00 AM	no	departmen	public	10	Sep 23, 1998
Buy something	Sep 24, 1998		no	special e	private	5	
Colloquium ann	Sep 28, 1998	9:00 AM	yes	departmen	public	10	
Colloquium ann	Oct 5, 1998	9:00 AM	yes	departmen	public	10	

Figure 34: Tasks list.

without user-specified due time appear after those with user-specified due times.

c. `Priority` is sorted in descending numeric order, i.e., higher priority tasks appear earlier in the list.

d. `Completed` is sorted in ascending chronological order, with all uncompleted items appearing after all completed items.

The subordering for each of the primary sorting keys in a task list is given in Table 2. These are the same subordering rules as for appointments, without the `Duration` field.

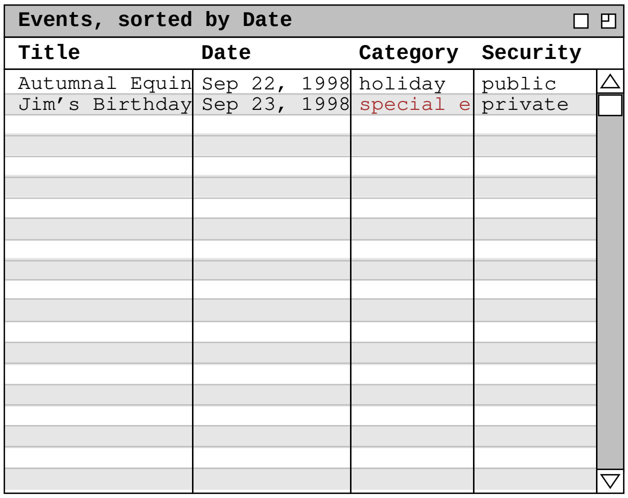
2.3.3.4. Event Lists

Figure 35 shows the result of the user selecting the '`View Lists Events`' command. Only two events are scheduled for the three-week listing period. The September 24 event was scheduled by the user. The September 22 event is built-in to the Calendar tool and hence automatically scheduled. The complete list of system-defined events is given in Section . By default, the events list is sorted primarily by date, secondarily by title.

Events do not recur, but multi-day events can be defined with a start and end date. There is a separate list item for each day of a multi-day event. For example, there are three separate list items for an event with start date September

Primary	Secondary	Tertiary	Quaternary
Title	Date	Time	
Due Date	Time	Title	
Due Time	Date	Title	
Recurs	Date	Time	Title
Category	Date	Time	Title
Security	Date	Time	Title
Priority	Date	Time	Title
Completed	Date	Time	Title

Table 2: Sorting suborders for task lists.



Title	Date	Category	Security
Autumnal Equin	Sep 22, 1998	holiday	public
Jim's Birthday	Sep 23, 1998	special e	private

Figure 35: Events list.

21 and end date September 23.

The events-list columns are a subset of the appointments list. For those columns that are present in the events list, the precise sorting rules are the same as given above for appointment lists.

The subordering for each of the primary sorting keys is given in Table 3.

2.3.3.5. All Items Lists

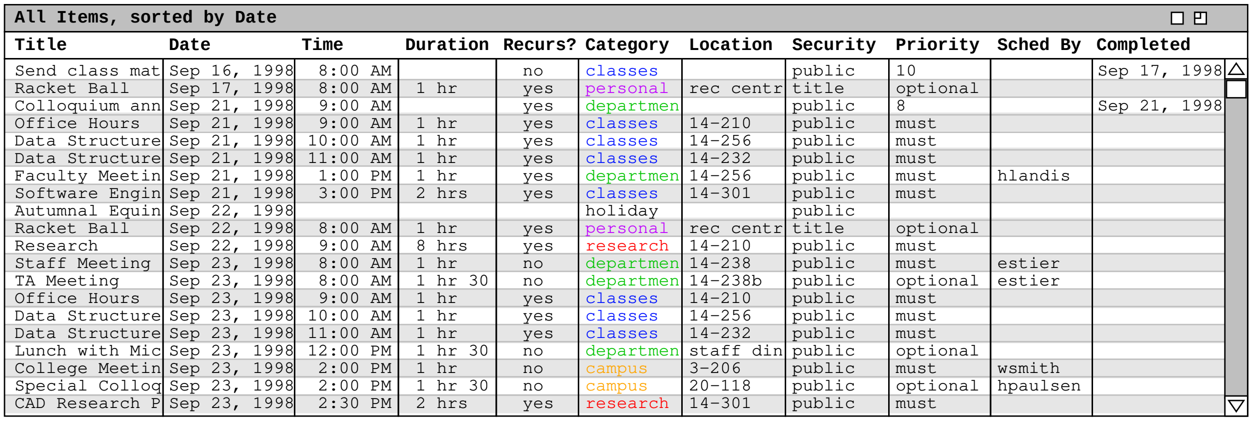
The final listing form contains all four types of scheduled item together in one list. To select this form, the user executes the '`View Lists All Items`' command, in response to which the system displays a window of the form shown in Figure 36.

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.

Primary	Secondary	Tertiary
Title	Date	
Date	Title	
Category	Date	Title
Security	Date	Title

Table 3: Sorting suborders for event lists.



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 36: 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.

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:

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 37. 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 38. 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.

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 field boxes must be checked.

The field check boxes are selectively disabled depending on which check boxes are on under ‘`Show these items`’. When all four item check boxes are on, the ‘`Due Time`’ check box is disabled. This is the case since in an all-items list, the ‘`Time`’ field covers the due time for tasks, as well as the start time for appointments and meetings. Further explanation of selective check box disabling is given below, following Figure 40.

The value for ‘`Maximum number of items` specifies how many items may be in the list. The default is 30. If there are more than the maximum number of items in any given list, based on the other settings in the dialog, then the list is truncated after sorting. For example, if the maximum setting is 30 and there 50 items in a list, then the list is sorted by its primary key, and the last 20 items are removed from the list before it is displayed.

The value for the ‘`Initial display height`’ specifies how many elements are visible in the list display window when it is initially displayed. If there are more than this many elements, the display has a vertical scrollbar

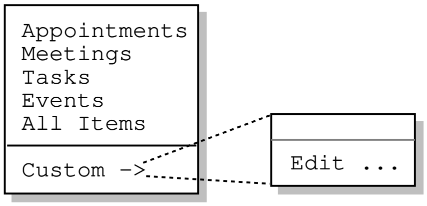
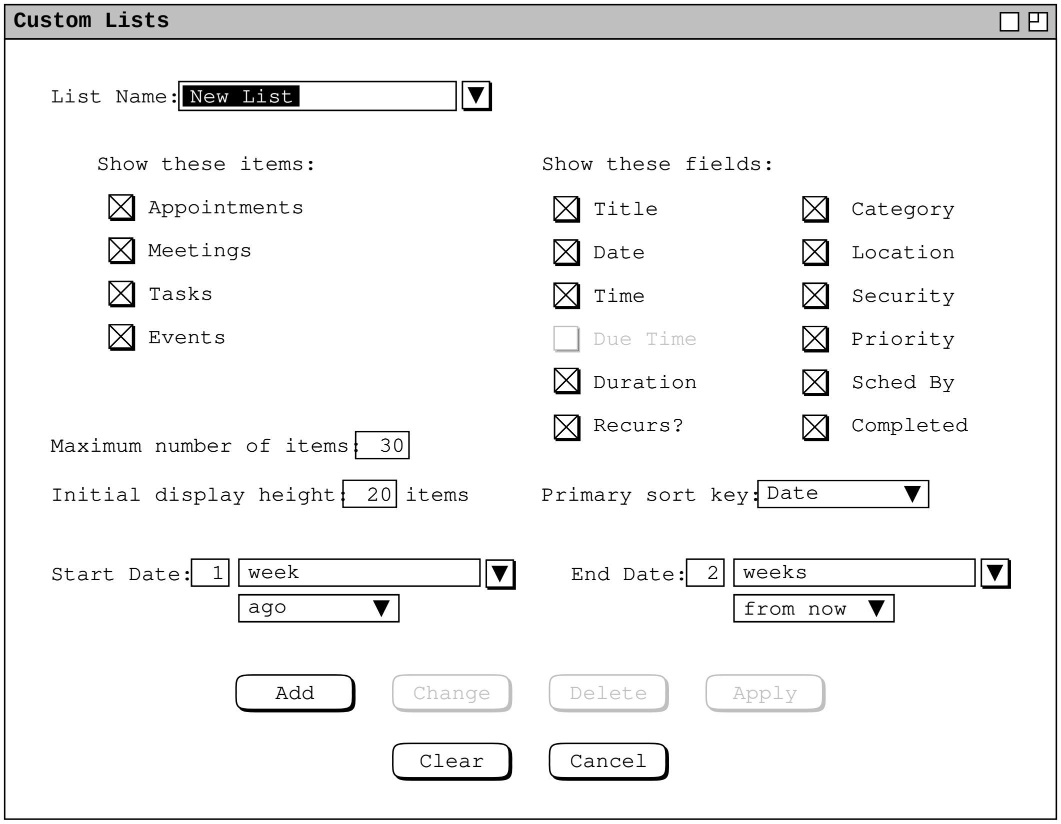


Figure 37: Initial custom lists menu.



Custom Lists

List Name: New List ▼

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: 30

Initial display height: 20 items

Primary sort key: Date ▼

Start Date: 1 week ▼ ago ▼

End Date: 2 weeks ▼ from now ▼

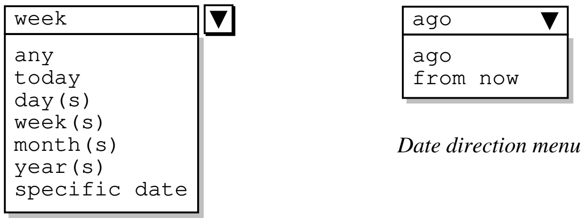
Figure 38: Custom list dialog.

added. The display can be resized as desired to make more or less items visible.

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 39 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 theses 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.

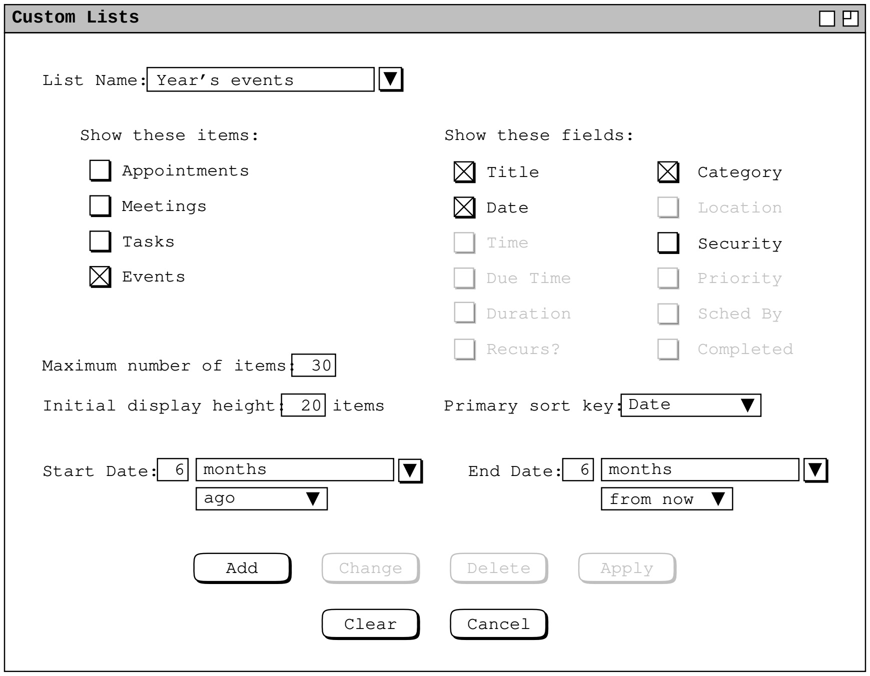
Figure 40 shows the result of the user having defined a new list named "Year's events". The list applies only to events, leaves out the security column, and extends over the date range of six months before and after today. The figure illustrates how the check boxes for some fields are selectively disabled 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`



Date range menu

Date direction menu

Figure 39: Custom list date menus.


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: 30

Initial display height: 20 items

Primary sort key: Date

Start Date: 6 months ago

End Date: 6 months from now

Figure 40: 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.

`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 example, Figure 41 shows the state of the ‘`Primary sort key`’ menu in the context of Figure 40 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 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.

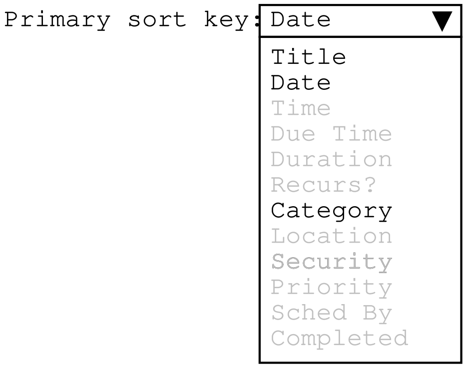


Figure 41: Primary sort key menu in Year’s events custom list.

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 new list to be defined.

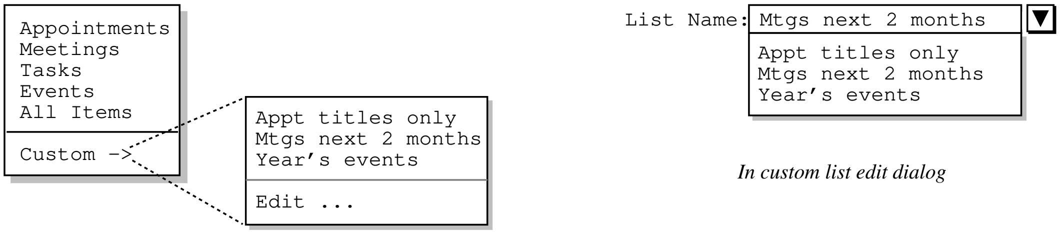
The names of available lists appear in two contexts:

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 42. 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 43 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.

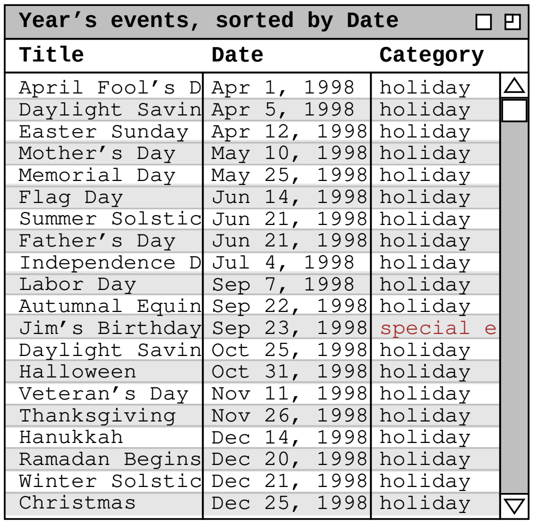
In the context of the scenarios thus far, "Jim's Birthday" is the only user-scheduled event. The other events in the ‘`Year's Events`’ list are all pre-defined holidays, as discussed further in Section .



In custom list edit dialog

In main menubar

Figure 42: 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

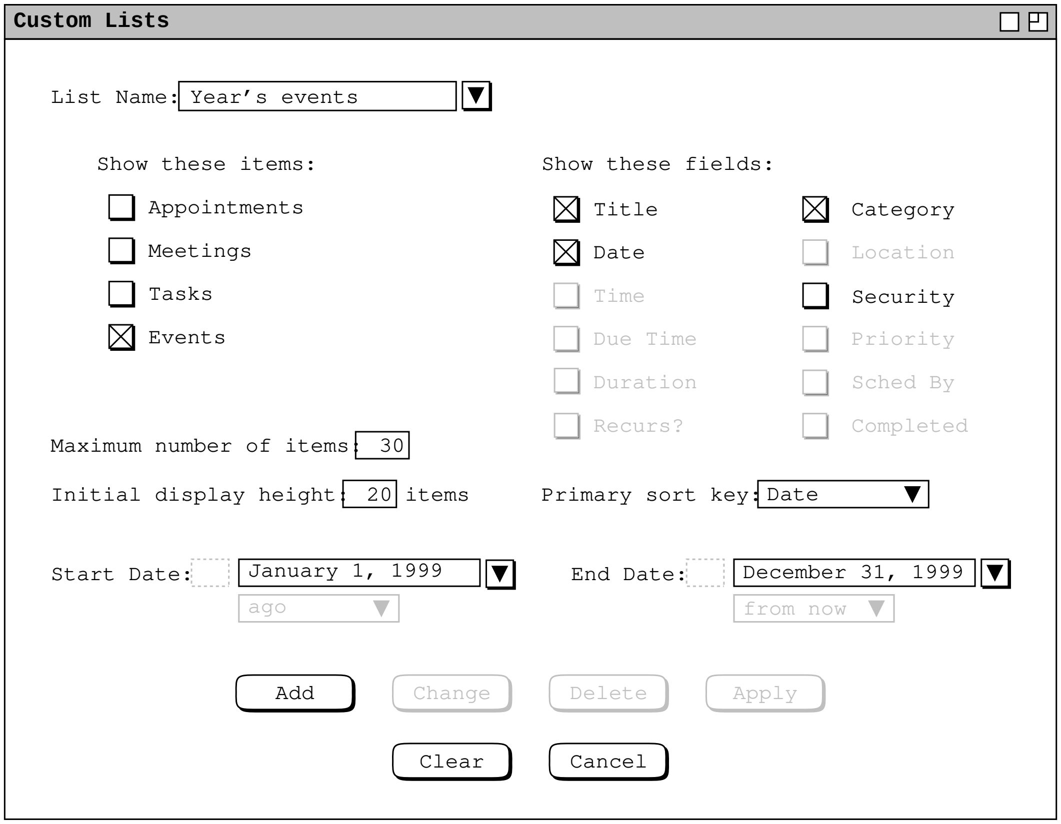
Figure 43: Year‘s events list display.

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 44 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 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.

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:



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: 30

Initial display height: 20 items

Primary sort key: Date

Start Date: January 1, 1999
ago

End Date: December 31, 1999
from now

Figure 44: Changed years-events list.

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 45. 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 21. If the user selects ‘`Filter Hide Appointments`’ in the context of that view, the system updates the display as shown in Figure 46. If the user then selects ‘`Hide Meetings`’, the system updates the display to that shown in Figure 47. 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 48. If the user selects ‘`Show Appointments`’ and ‘`Show Meetings`’, the month display is restored to the original state shown in Figure 21.

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 49 and 50 show the results of hiding both events and tasks in the context of the daily and weekly views shown originally in Figures 14 and 18.

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.

The four show/hide filtering 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. At the item level of viewing, filtering applies to the entire item-level display window, not to individual fields within the window. Specific details of filter application at the item level are covered in Section 2.3.4.3.

Show/hide filtering does not apply to list views, since lists are already specialized for each kind of item. The custom filters described in the next section do apply to lists, as well as to calendar views. Operational details of show/hide and custom filtering used in conjunction are covered in Section 2.3.4.3.

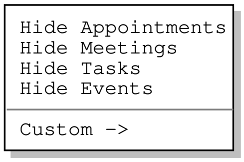
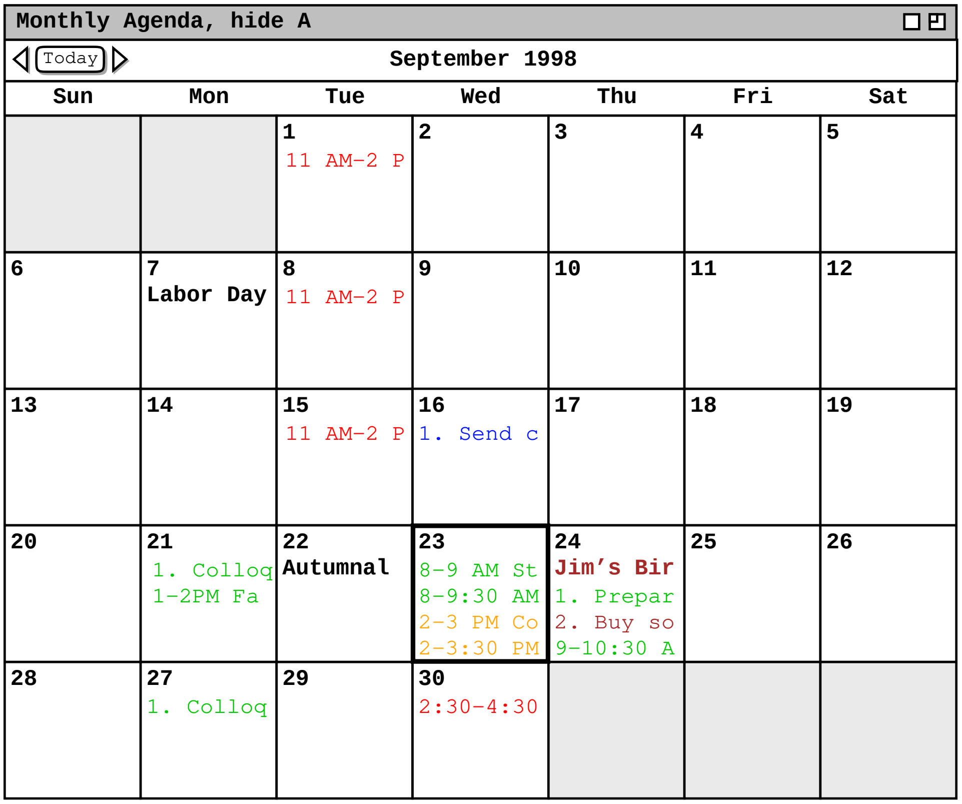


Figure 45: Filter submenu.

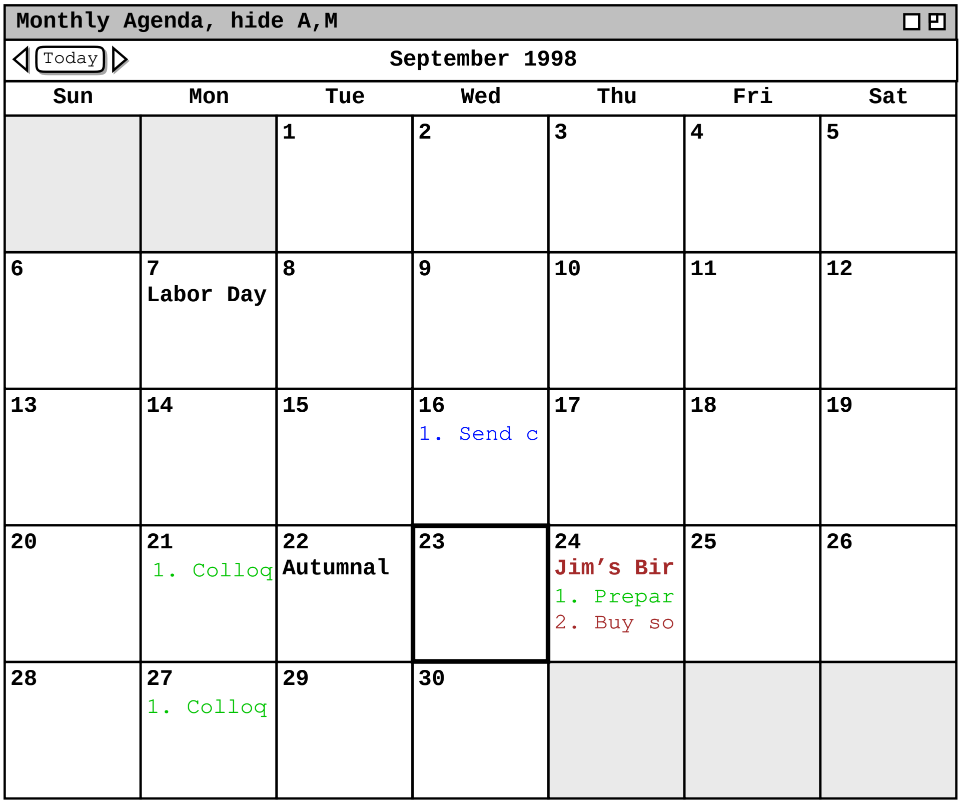


Monthly Agenda, hide A

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			

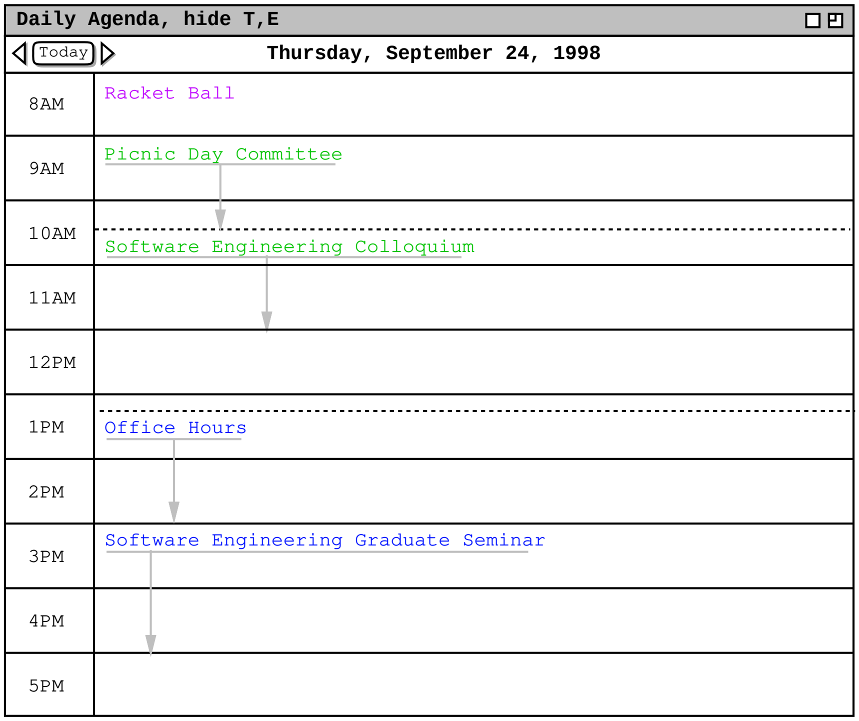
Figure 46: Month view with appointments filtered out.



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 47: Month view with meetings filtered out.

Figure 48: Filter menu in ‘Show Appointments’, ‘Show Meetings’ state.

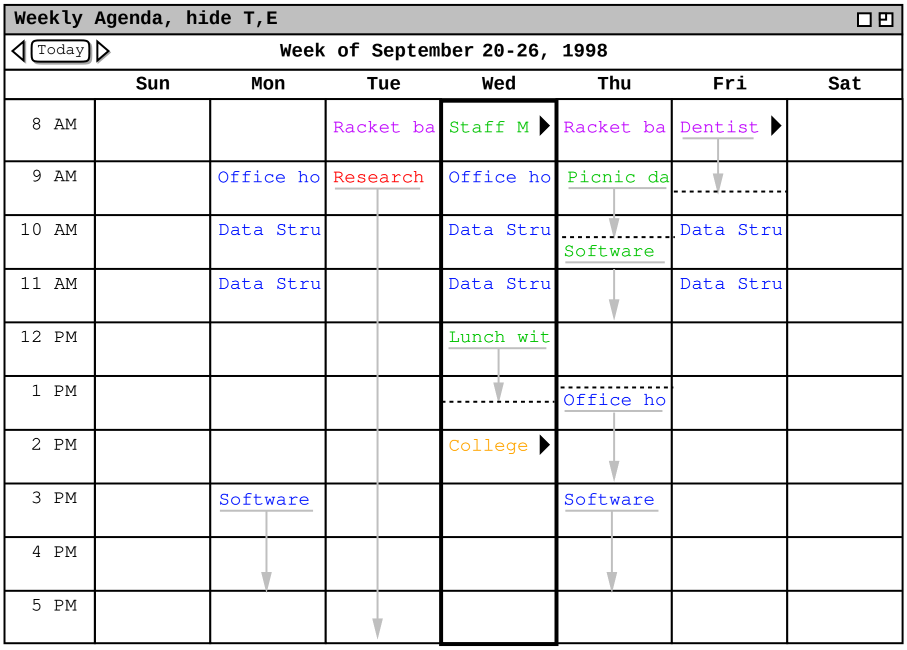


Daily Agenda, hide T,E

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 49: Events and tasks filtered out of a day view.



Weekly Agenda, hide T,E

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

2.3.4.1. Custom Filter Definition

The custom filtering feature of the Calendar Tool allows the user to define finer-grain filters than simply all items of a particular type. For example, the user can choose to filter out all but a particular category of item that is scheduled between some particular dates.

By default, there are no custom filters defined. Hence, when the user initially selects the '`View Filters Custom`' menu item, the system displays the submenu as shown in Figure 51. The blank space above the '`Edit ...`' command is where the names of custom filters appear once they are defined by the user.

To create a new custom filter, the user selects the '`Edit ...`' command, whereupon the system displays the filter definition dialog shown in Figure 52.

The '`Filter Name`' text box contains the name of the filter being defined. The initial default name is "`New Filter`". The user enters a chosen filter name by typing in the text box. The name for each defined filter must be unique among all other filters.

The editing fields below the filter name allow the user to define detailed filtering 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 filter commands. The criteria apply uniformly to all calendar and list displays.

The fields in the filter definition constitute the union of components from all four types of scheduled item. Given this, each field is not necessarily applicable to all types of scheduled item. Table 6 defines specifically which fields apply to which item types. (This table is a superset of Table 5 defined for custom lists.)

There are a number of pulldown menus associated with the fields of the filter definition. Figure 53 shows an the expansion of these menus. Descriptions of each filtering field follow. Section defines the precise pattern matching rules that apply to filtering

2.3.4.1.1. Title

The '`Title`' field in the filter definition defines 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. Specifically:

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.
b. When '`contains`' is selected, items are visible when they have a title that contains the pattern string.
c. When '`does not contain`' is selected, items are visible when they have a title that does not contain the pattern string.
d. When '`starts with`' is selected, items are visible when they have a title that starts with the pattern string.
e. When '`ends with`' is selected, items are visible when they have a title that ends with the pattern string.
f. When '`equals`' is selected, items are visible when they have a title that exactly equals the pattern string.
g. When '`does not equal`' is selected, items are visible when they have a title that does not equal the pattern string.

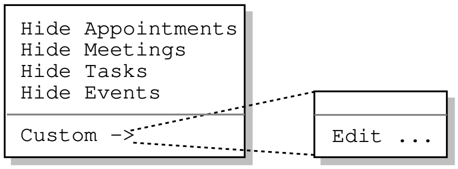
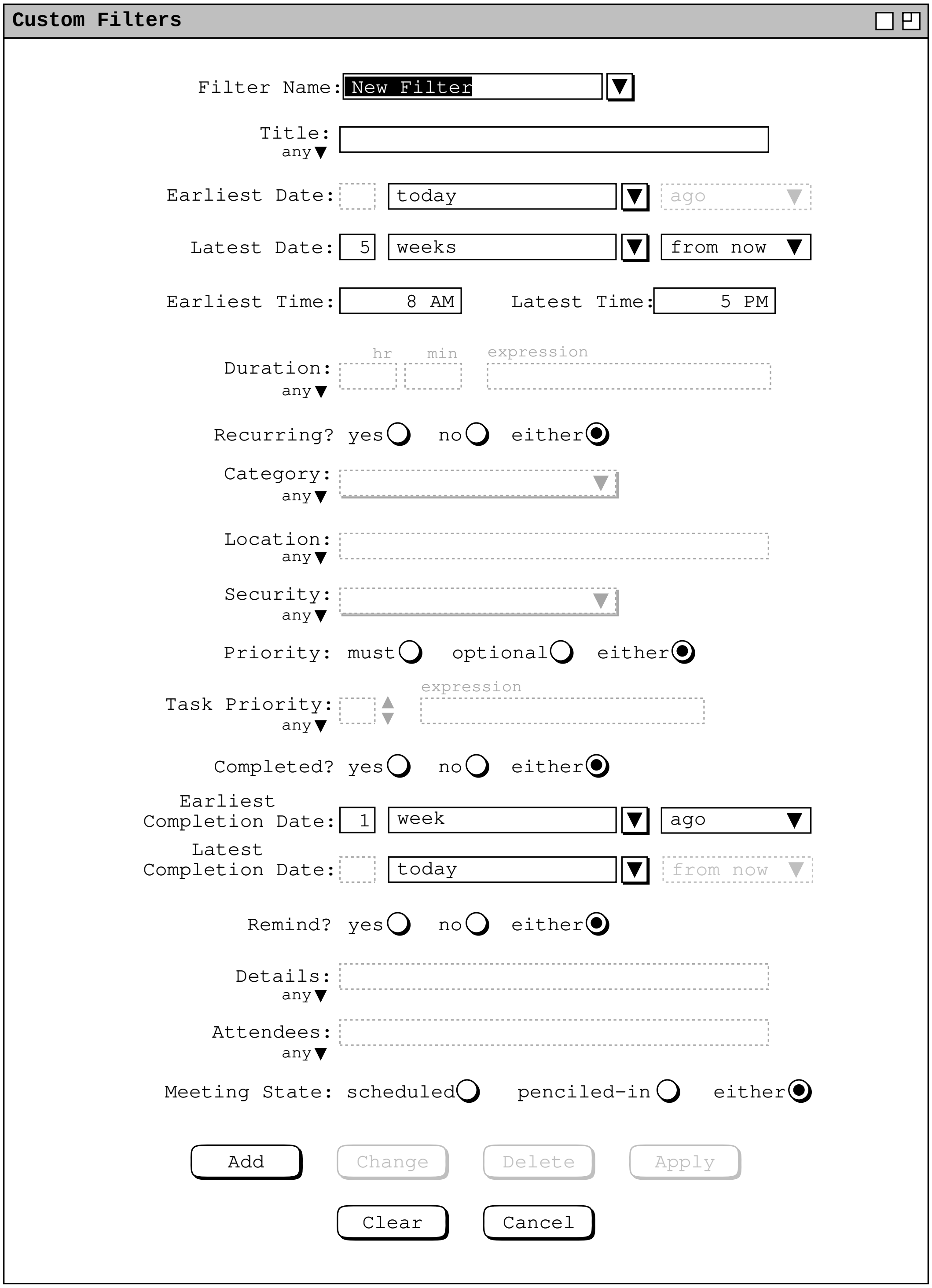


Figure 51: Initial custom filters menu.



Custom Filters

Filter Name: New Filter

Title: any

Earliest Date: today ago

Latest Date: 5 weeks from now

Earliest Time: 8 AM Latest Time: 5 PM

Duration: any hr min expression

Recurring? yes ○ no ○ either ●

Category: any

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

Meeting State: scheduled ○ penciled-in ○ either ●

Figure 52: 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
Meeting State	meetings

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

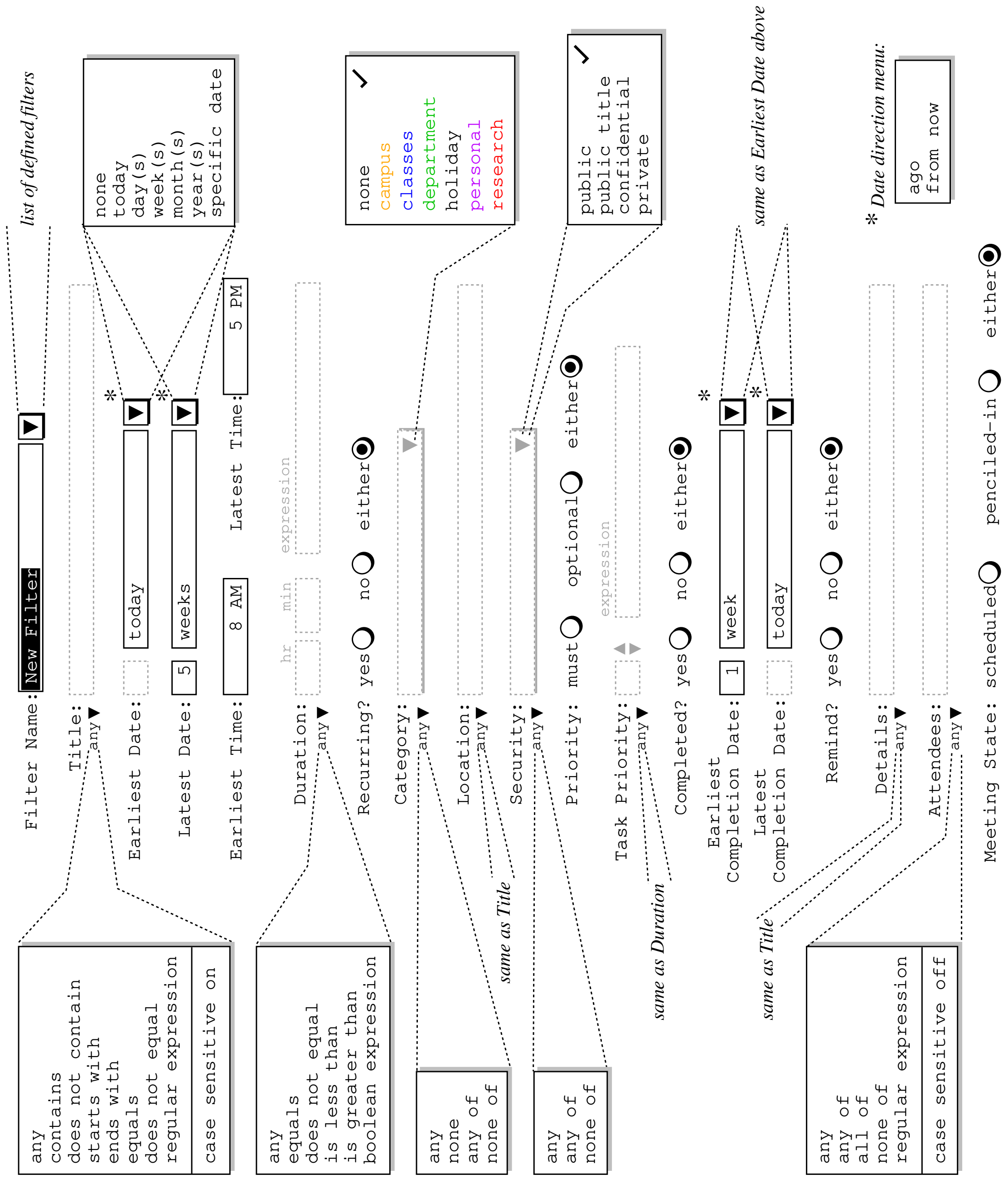


Figure 53: Expansion of filter definition menus.

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 .

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 .

2.3.4.1.5. 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 the selection in the pull-down 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 .

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.

2.3.4.1.15. Meeting State

The '`Meeting State`' field defines the visibility of meetings based on whether or not they have been accepted by the user. As described in Section 2.4.1, a scheduled meeting is one that the user has scheduled herself, or accepted from another user. A penciled-in meeting is one scheduled by another user but not yet accepted by a receiving user. If the '`scheduled`' radio is selected, then only scheduled meetings are visible, i.e., not-yet-accepted meetings are not visible. If '`penciled-in`' is selected, only meetings not yet accepted by the user are visible. If '`either`' is selcted, all meetings, both scheduled and not-yet-accepted, are visible.

2.3.4.2. Using Custom Filters

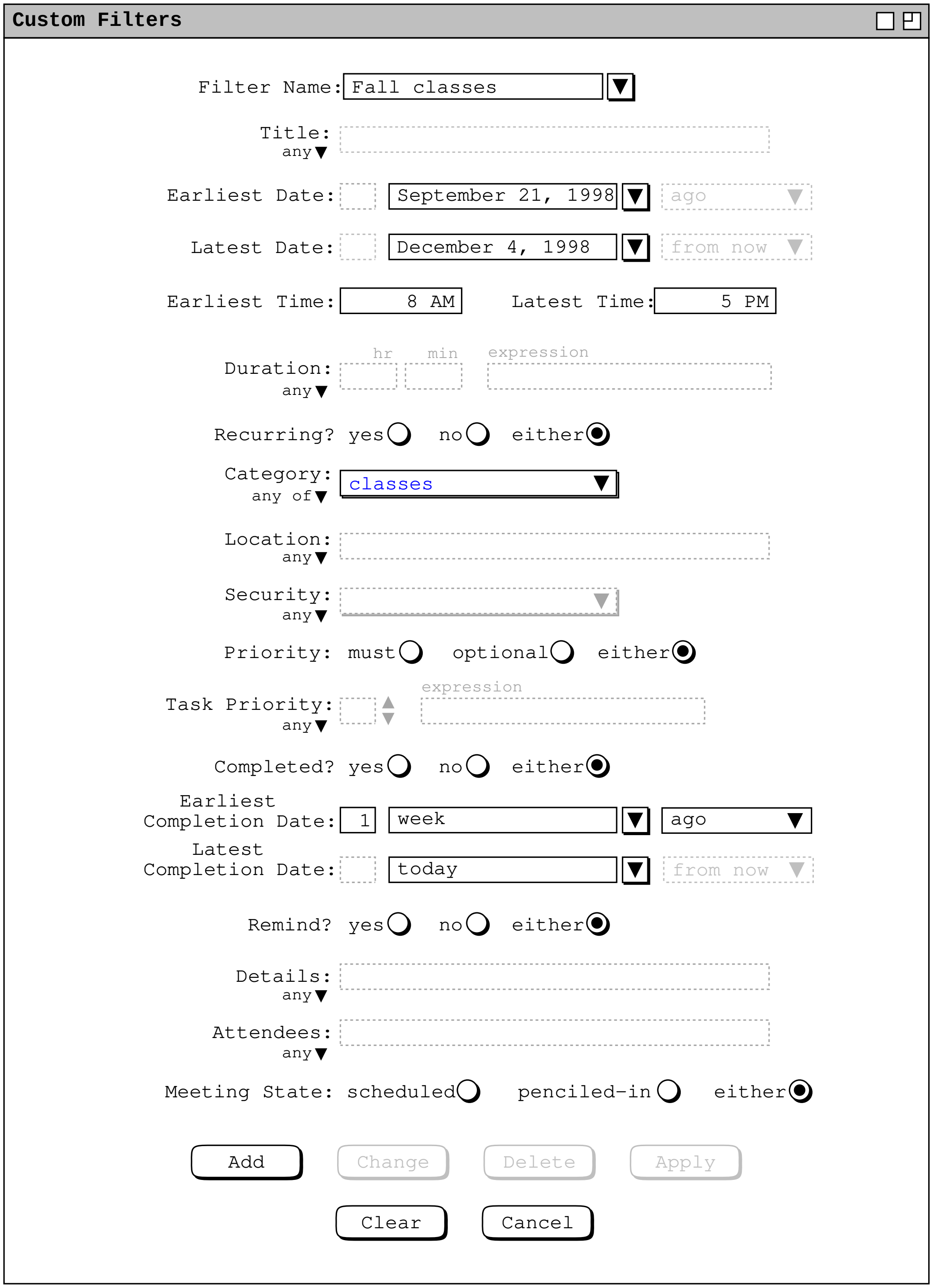
Figure 54 shows the result of the user having defined a filter named "Fall classes" with the following criteria:

o earliest date is September 21

o latest date is December 4

o 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



Custom Filters

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: any hr min expression

Recurring? yes ○ no ○ either ●

Category: any of 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

Meeting State: scheduled ○ penciled-in ○ either ●

Figure 54: Fall classes custom filter.

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 names of available filters appear in two contexts:

a. the ‘`View Filters Custom`’ submenu, accessible from the main command menubar;
b. the ‘`Filter Name`’ menu, in the custom filters dialog.

These contexts are illustrated in Figure 55. 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 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 21. Figure 56 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:

> **`, hide`** *hide-list* **`, filter`** *custom-filter-name*

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

To clarify that a filter is active, its name is checked in the custom filter menu. For example, Figure 57 shows the state of the filter menus as established by the user at this point in the scenarios.

When a filter is selected from the list, it is applied automatically to all active display windows. For example, consider the appointments list shown originally in Figure 31. Activating the "`Fall classes`" filter when this display is visible causes the system to update its display to that shown Figure 58. This filtered list contains only appointments of the `classes` category, per the definition of the ‘`Fall classes`’ filter.

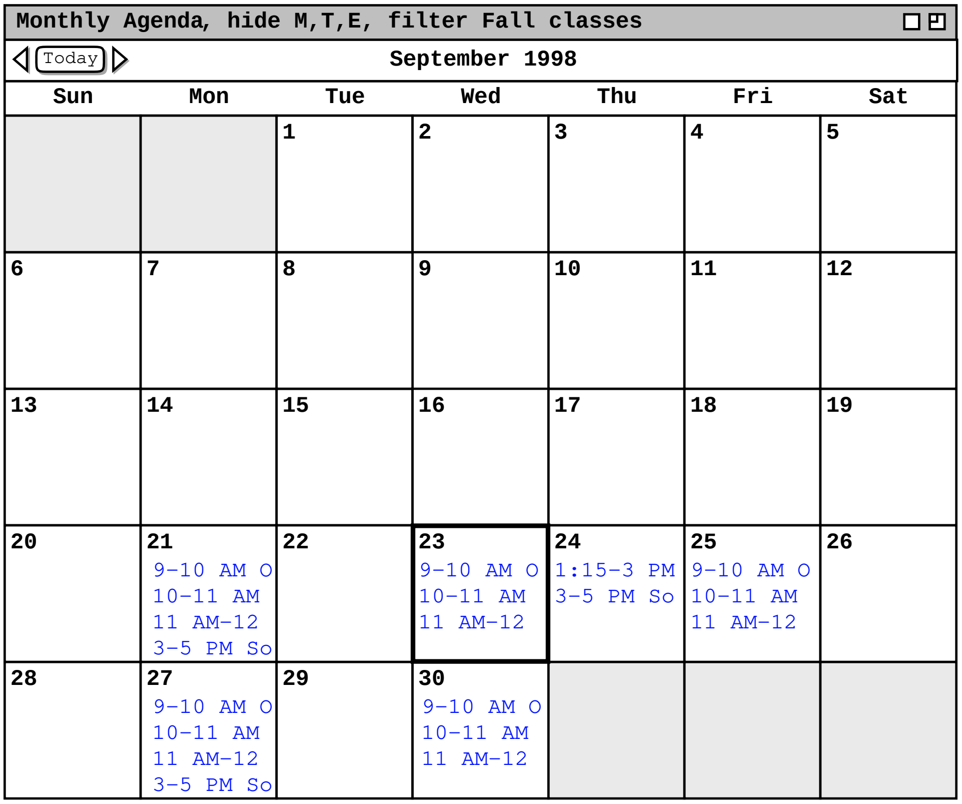
Since the show/hide filtering commands do not apply to lists, the format of the banner string is different for a list display. Viz., the filtering banner for a list display contains only a custom filter name, with no hide-list.

The user can change or delete defined filters in the custom filter dialog. To change an existing filter, the user selects the desired filter 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 59 shows the user having edited the previously defined ‘`Fall classes`’ filter. The edits involve changing the ‘`Title`’ filtering field 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 ‘`case sensitive off`’ menu selection is also made. To confirm these changes, the user presses the ‘`Change`’ button at the bottom of the dialog.

In custom filter edit dialog

In main menubar

Figure 55: Menu contexts in which filter names appear.



Monthly Agenda, hide M,T,E, filter Fall classes

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 56: Month view with Fall classes filter applied.

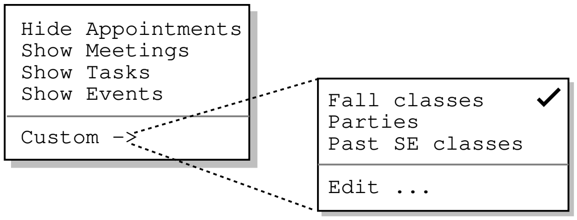
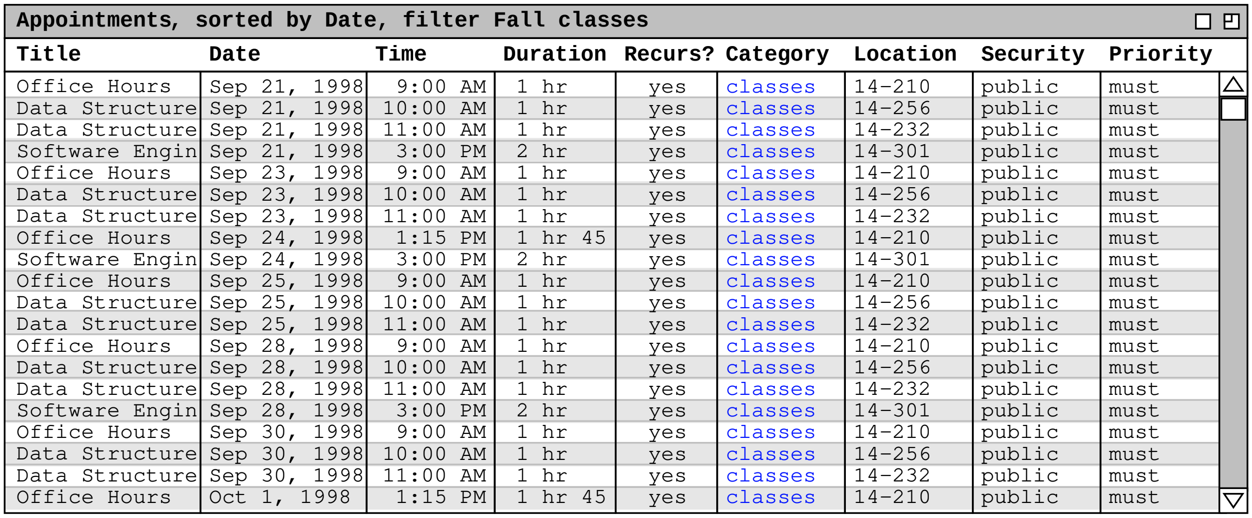


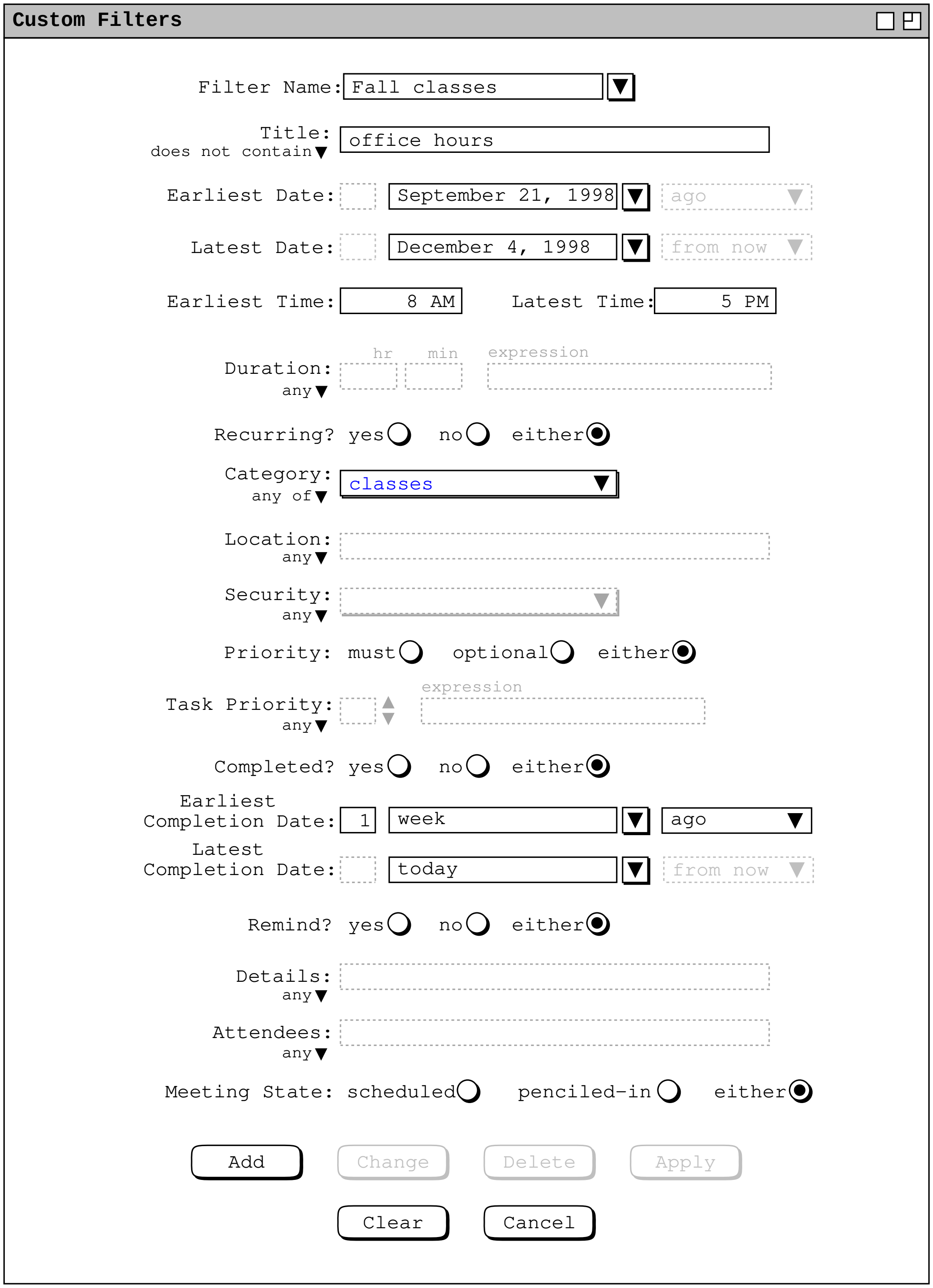
Figure 57: State of the filter menus with filtering applied.



Appointments, sorted by Date, filter Fall classes

Title	Date	Time	Duration	Recurs?	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 58: Appointments list with Fall classes filter applied.



Custom Filters

Filter Name: Fall classes

Title: does not contain office hours

Earliest Date: September 21, 1998

Latest Date: December 4, 1998

Earliest Time: 8 AM Latest Time: 5 PM

Duration: any

Recurring? yes ○ no ○ either ●

Category: any of classes

Location: any

Security: any

Priority: must ○ optional ○ either ●

Task Priority: any

Completed? yes ○ no ○ either ●

Earliest Completion Date: 1 week ago

Latest Completion Date: today

Remind? yes ○ no ○ either ●

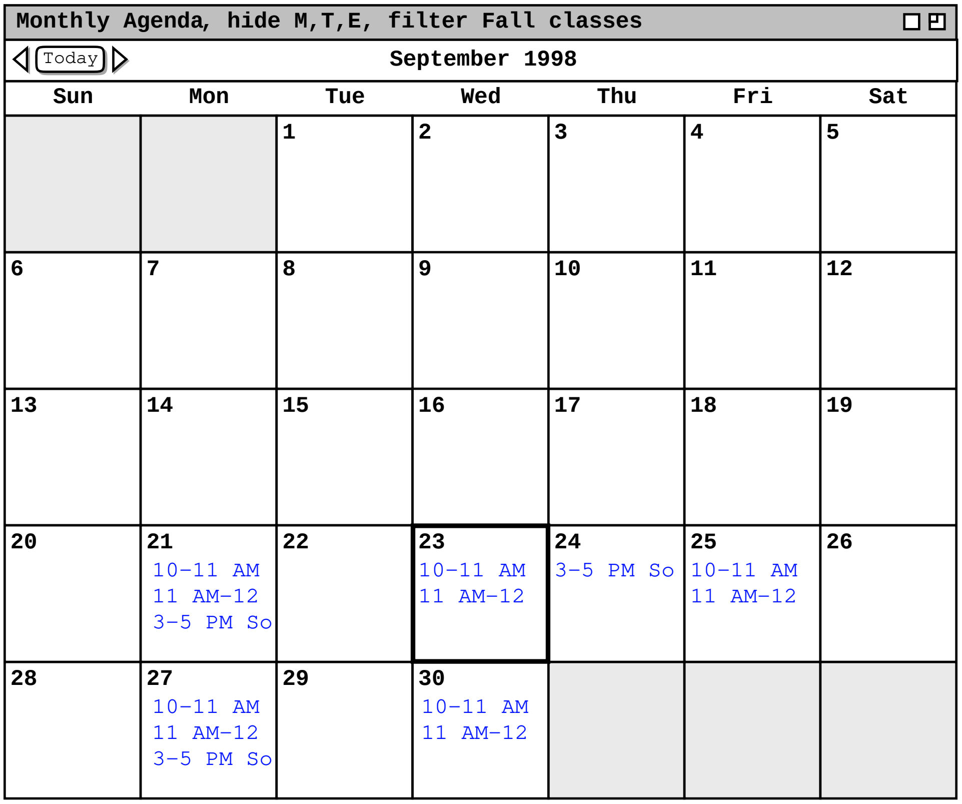
Details: any

Attendees: any

Meeting State: scheduled ○ penciled-in ○ either ●

Figure 59: Changed years-events filter.

If the user applies this changed version of the ‘`Fall classes`’ filter in the context of Figure 56, the system updates the display as shown in Figure 60, where the appointments with "Office hours" in the title have been filtered out.



Monthly Agenda, hide M,T,E, filter Fall classes

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 60: 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`’.

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 eariler, 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, it takes effect immediately in all applicable display windows. The displays to which filtering applies are those generated by the top nine commands in the ‘`View`’ menu, as shown in Figure 61. 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 49 and 50. 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 57 above. In this filtering state, the ‘`View`

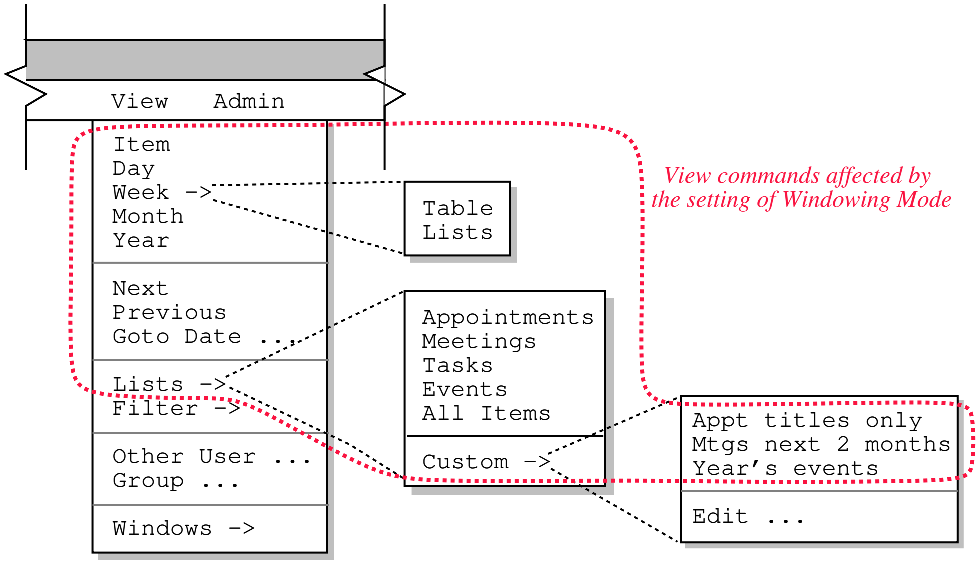


Figure 61: Commands affected by filter settings.

`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 controled by an option setting, as described in Section 2.7.4.2.

2.3.5. Viewing Other Users and Group Calendars

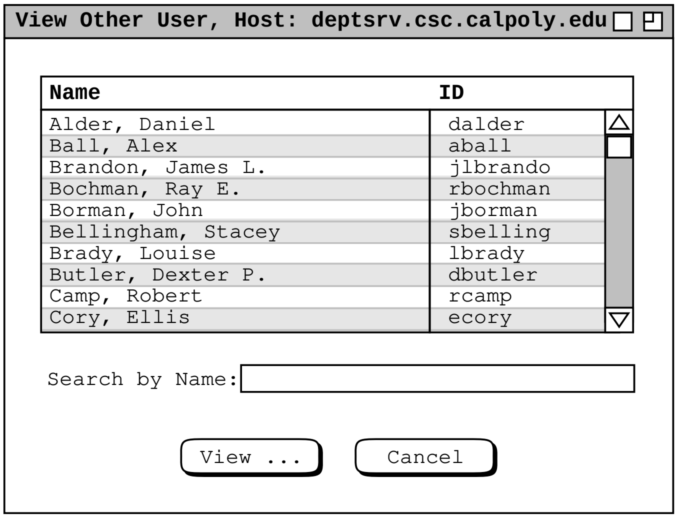
The Calendar Tool provides cross-user access to the calendars it manages. It also provides access to the group calendars that contain the meetings scheduled for user groups. In order to view the calendar of another user or group, the current user must be connected to a Calendar Tool central host computer on which the other user or group calendar resides. Details of central host connection are covered in Section 2.6.6.1. When the user is not connected to any central host, the '`Other User`' and '`Group`' items are disabled in the '`View`' menu. In the scenarios of this section of the requirments, it is assumed that the user is connected to the Calendar Tool central host identified as "`deptsrv.csc.calpoly.edu`".

2.3.5.1. Viewing Other Users' 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



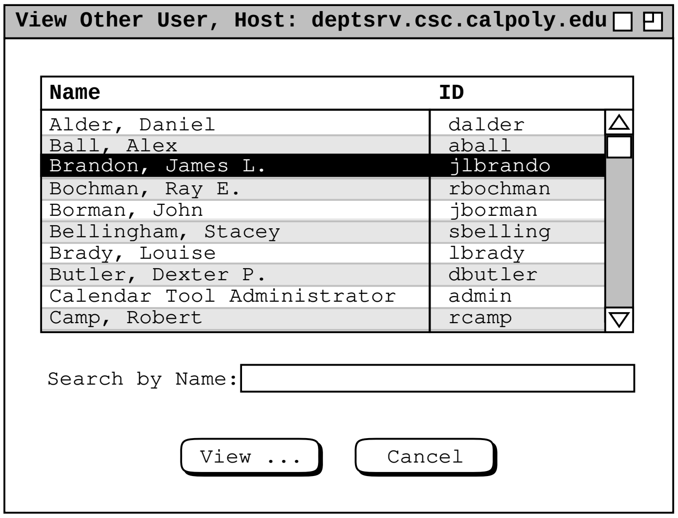
Name	ID
Alder, Daniel	dalder
Ball, Alex	aball
Brandon, James L.	jlbrando
Bochman, Ray E.	rbochman
Borman, John	jborman
Bellingham, Stacey	sbelling
Brady, Louise	lbrady
Butler, Dexter P.	dbutler
Camp, Robert	rcamp
Cory, Ellis	ecory

Figure 62: View other user dialog.

"`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 .

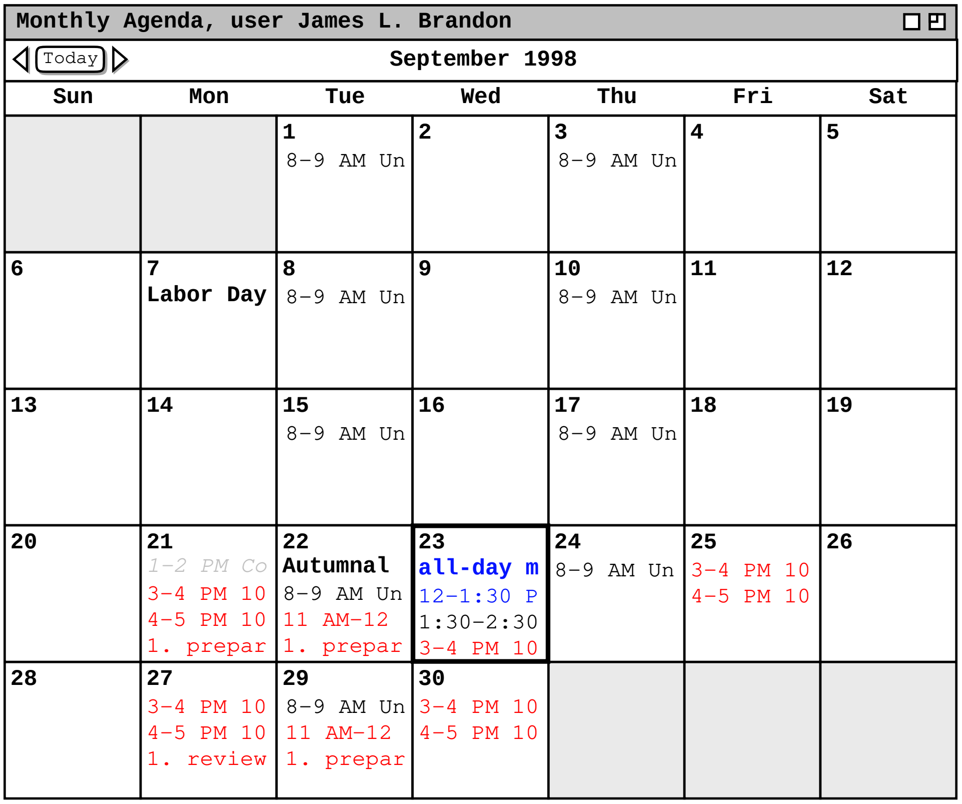
Figure 63 shows the user having selected user `James L. Brandon` for viewing. When the user presses the '`View ...`' button, the system responds with a new display window as shown in Figure 64. Other users' calendars may be viewed with all of the commands available in the `View` menu. For example, Figure 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.

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



Name	ID
Alder, Daniel	dalder
Ball, Alex	aball
Brandon, James L.	jlbrando
Bochman, Ray E.	rbochman
Borman, John	jborman
Bellingham, Stacey	sbelling
Brady, Louise	lbrady
Butler, Dexter P.	dbutler
Calendar Tool Administrator	admin
Camp, Robert	rcamp

Figure 63: Selecting user for viewing.

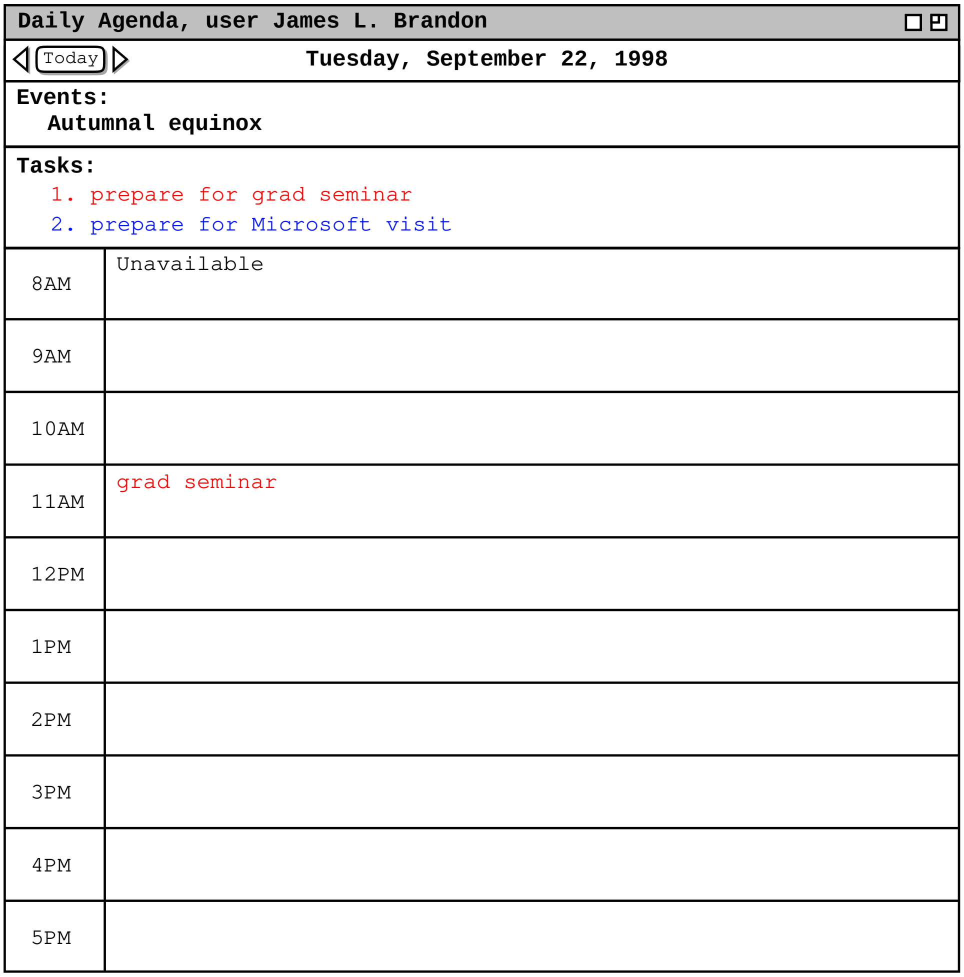


Monthly Agenda, user James L. Brandon

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

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.

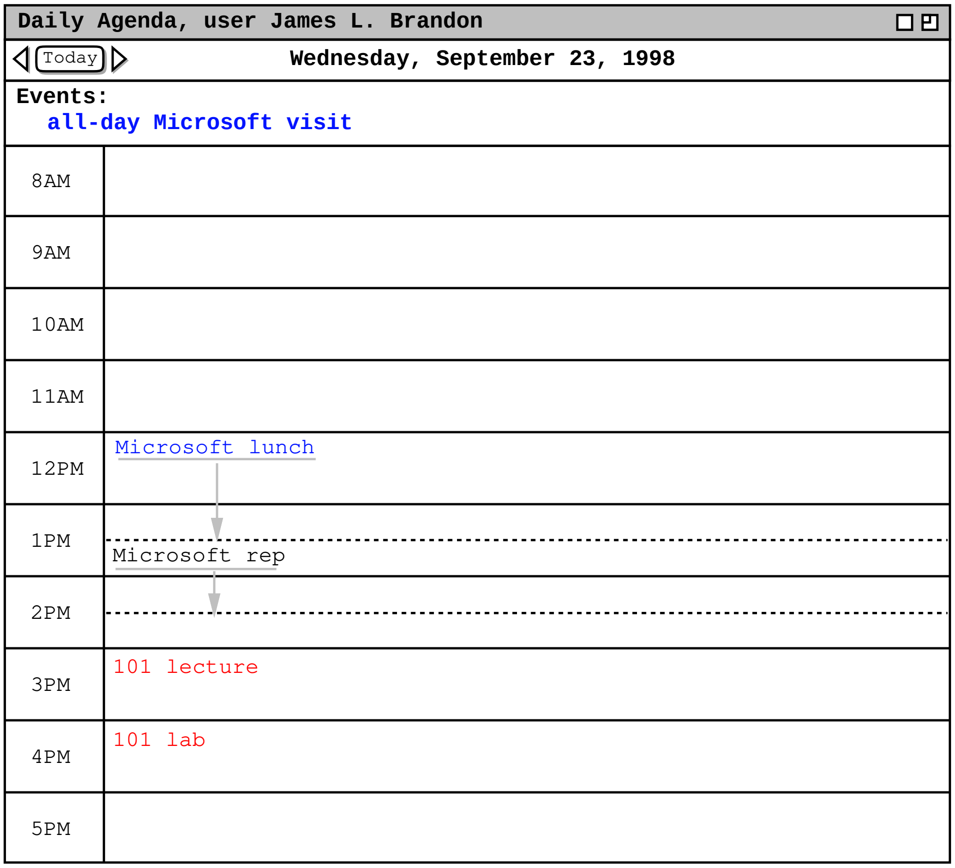


Figure 66: Another day view in the James L. Brandon calendar.

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).

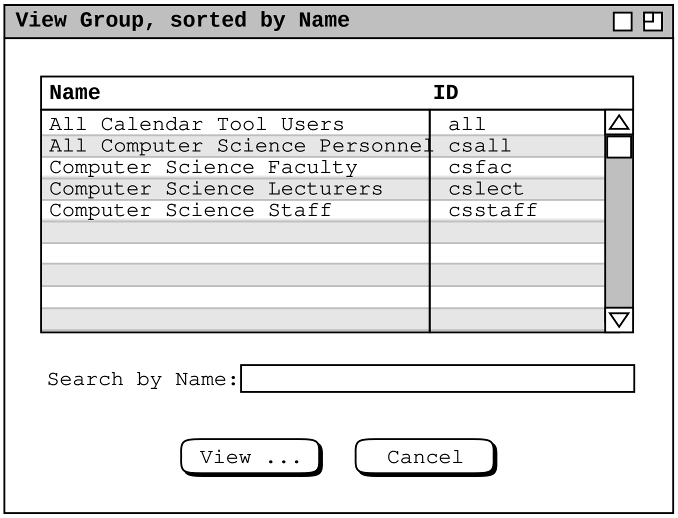
The figures show how the banner of the display window is altered to indicate that the calendar of another user is displayed. The format of the banner is the following:

`, 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



Name	ID
All Calendar Tool Users	all
All Computer Science Personnel	csall
Computer Science Faculty	csfac
Computer Science Lecturers	cslect
Computer Science Staff	csstaff

Figure 67: View group dialog.

212 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.

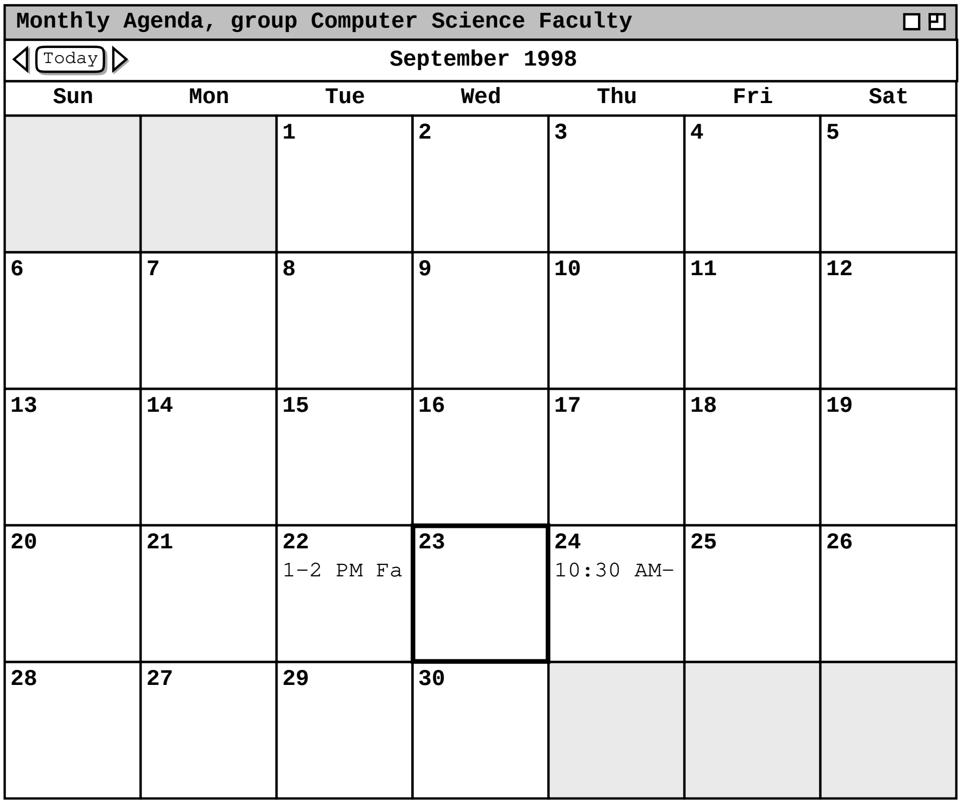
Figure 68 shows how the banner of the display window is altered to indicate that the calendar of a group is displayed. The format of the banner is the following:

`, 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 Multiple Windows and Multiple Calendars

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

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.

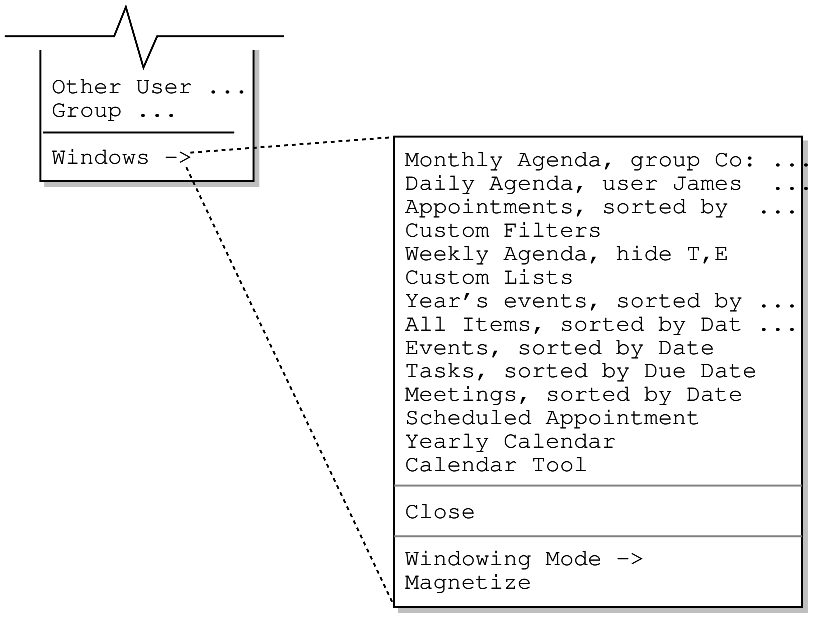


Figure 69: Windows submenu with chronological list of active windows.

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. In editing windows without a ‘`Cancel`’ button, such as item-level displays, closing the window cancels any unconfirmed edits, such that the next time that item-level display appears, any previously unconfirmed edits have been lost. 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 uniconfies 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 2.8.7.

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 2.8.3.

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

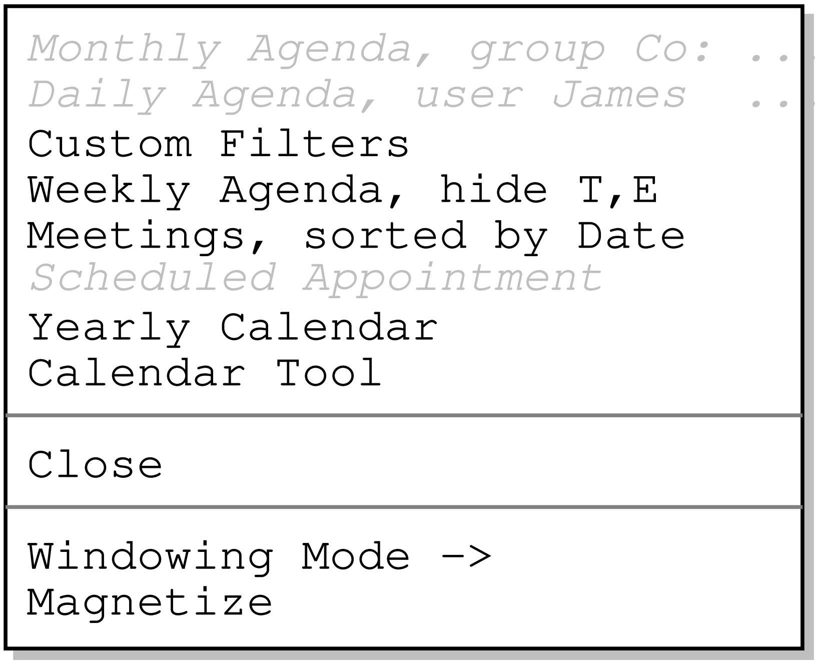


Figure 70: Iconified names in the windows menu.

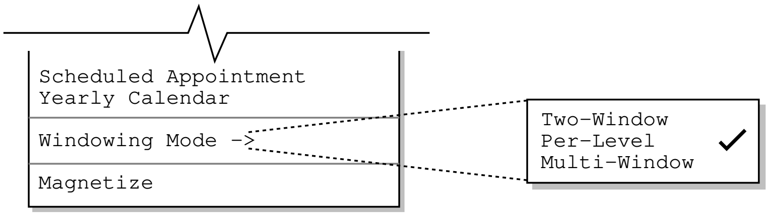


Figure 71: Windowing mode submenu.

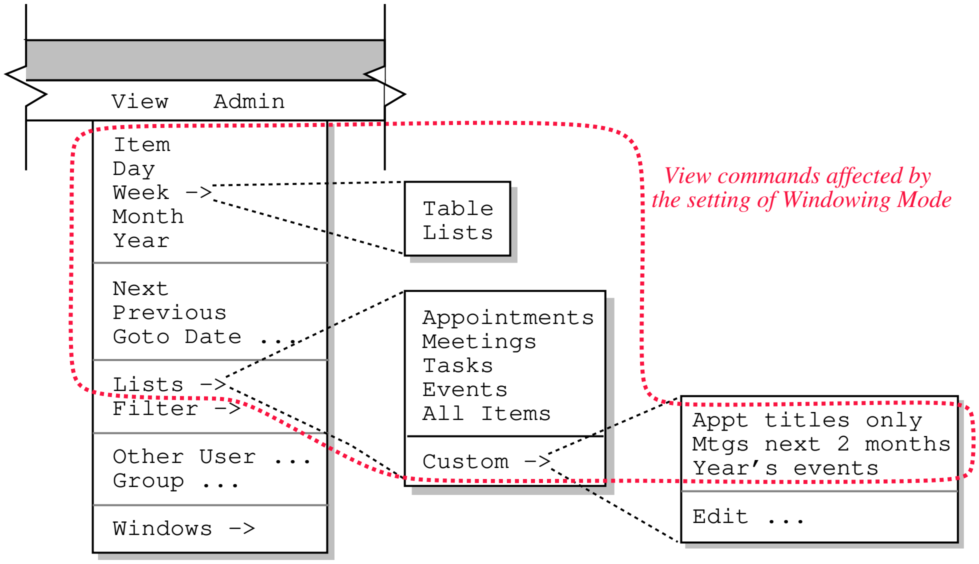


Figure 72: Commands affected by the setting of Windowing Mode.

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.

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

Window Title	**Command that Generates Window**
"Scheduled *item-type*"	`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`
custom list name	`View Lists Custom` *custom list name*

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

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

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

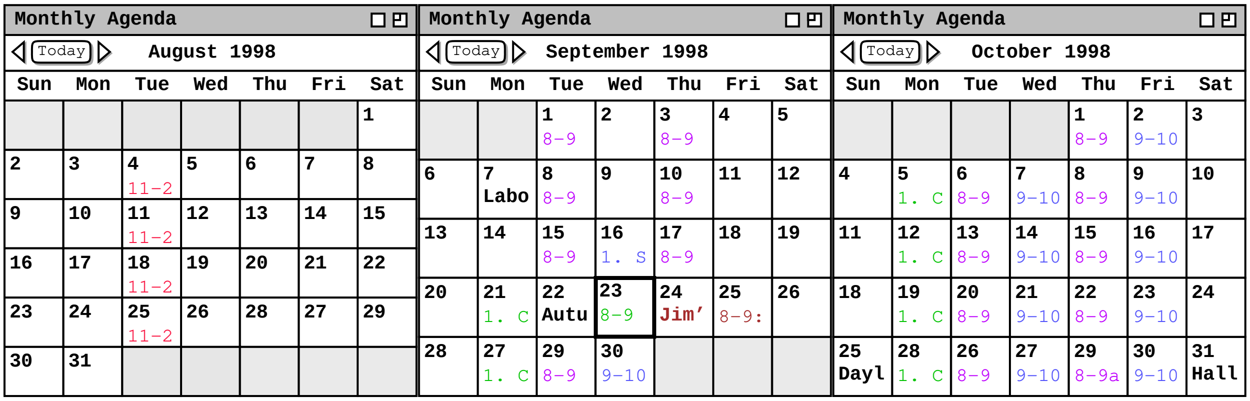
The windowing mode is calendar-specific. That is, each calendar has its own windowing mode setting. When more than one calendar is open, the windowing mode shown in the '`Windowing Mode`' menu is that for the current calendar and applies only to the current calendar. In per-level and two-window modes, 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



August 1998

Sun	Mon	Tue	Wed	Thu	Fri	Sat
						1
2	3	4 11-2	5	6	7	8
9	10	11 11-2	12	13	14	15
16	17	18 11-2	19	20	21	22
23	24	25 11-2	26	28	27	29
30	31					

September 1998

Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1 8-9	2	3 8-9	4	5
6	7 Labo	8 8-9	9	10 8-9	11	12
13	14	15 8-9	16 1. S	17 8-9	18	19
20	21 1. C	22 Autu	23 8-9	24 Jim'	25 8-9:	26
28	27 1. C	29 8-9	30 9-10			

October 1998

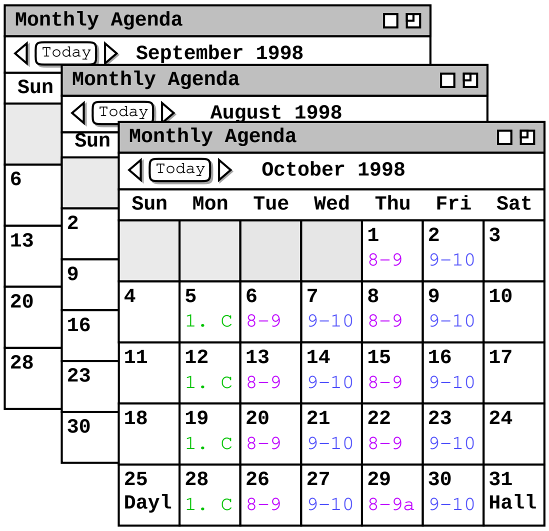
Sun	Mon	Tue	Wed	Thu	Fri	Sat
				1 8-9	2 9-10	3
4	5 1. C	6 8-9	7 9-10	8 8-9	9 9-10	10
11	12 1. C	13 8-9	14 9-10	15 8-9	16 9-10	17
18	19 1. C	20 8-9	21 9-10	22 8-9	23 9-10	24
25 Dayl	28 1. C	26 8-9	27 9-10	29 8-9a	30 9-10	31 Hall

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

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.

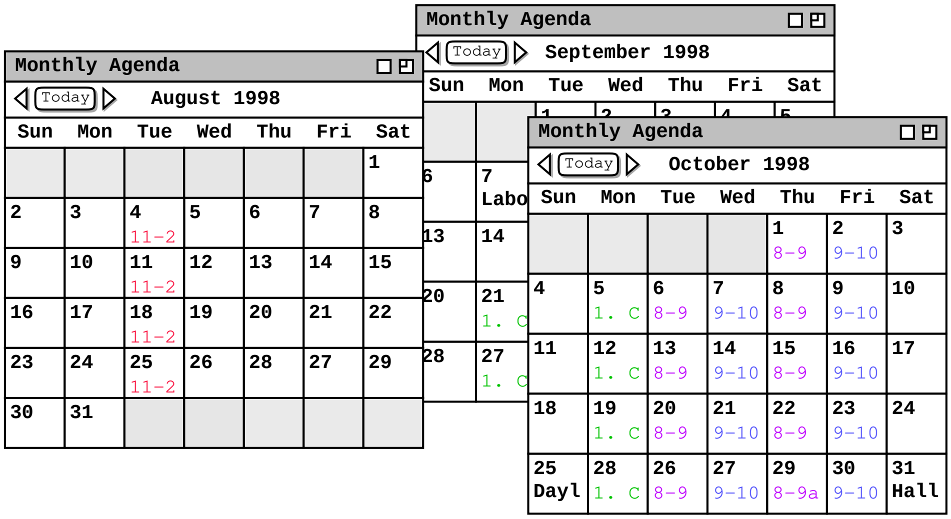
When two or more windows are magnetically connected, moving any one of the windows moves all connected windows along with it, so that the relative positions of the windows is maintained. A magnetized window can be moved independently by holding down the shift key while moving, which temporarily demagnetizes the window during the move. A window can be 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



Sun	Mon	Tue	Wed	Thu	Fri	Sat
				1 8-9	2 9-10	3
4	5 1. C	6 8-9	7 9-10	8 8-9	9 9-10	10
11	12 1. C	13 8-9	14 9-10	15 8-9	16 9-10	17
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

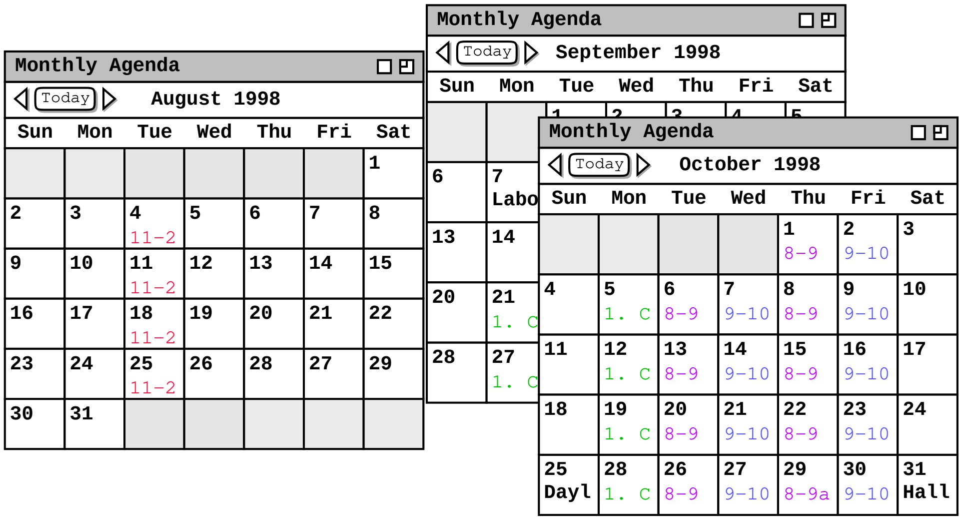
Figure 74: Initial unaligned positions of three magnetized windows.



Sun	Mon	Tue	Wed	Thu	Fri	Sat
						1
2	3	4 11-2	5	6	7	8
9	10	11 11-2	12	13	14	15
16	17	18 11-2	19	20	21	22
23	24	25 11-2	26	28	27	29
30	31					

Sun	Mon	Tue	Wed	Thu	Fri	Sat
				1 8-9	2 9-10	3
4	5 1. C	6 8-9	7 9-10	8 8-9	9 9-10	10
11	12 1. C	13 8-9	14 9-10	15 8-9	16 9-10	17
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 75: User moves right side of August window near left side of September window.

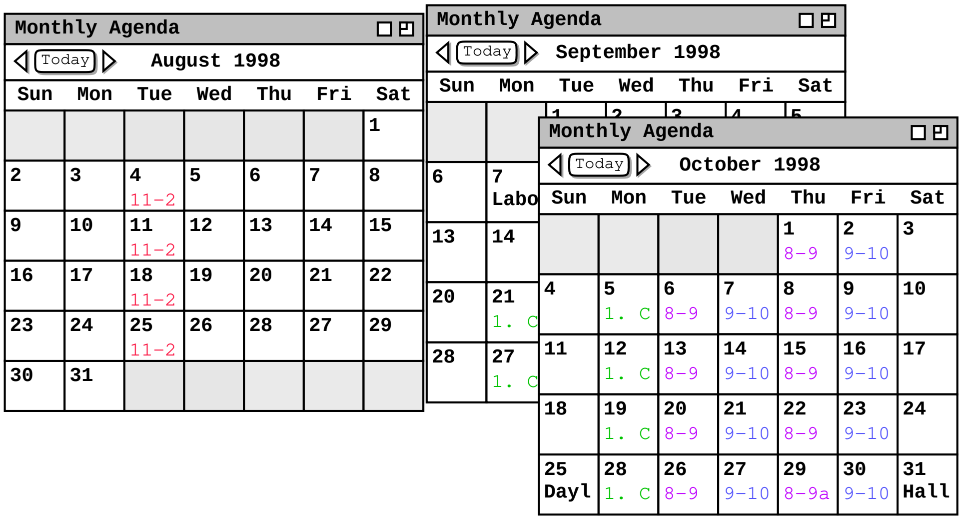


Sun	Mon	Tue	Wed	Thu	Fri	Sat
						1
2	3	4 11-2	5	6	7	8
9	10	11 11-2	12	13	14	15
16	17	18 11-2	19	20	21	22
23	24	25 11-2	26	28	27	29
30	31					

Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5
6	7 Labo					
13	14					
20	21 1. C					
28	27 1. C					

Sun	Mon	Tue	Wed	Thu	Fri	Sat
				1 8-9	2 9-10	3
4	5 1. C	6 8-9	7 9-10	8 8-9	9 9-10	10
11	12 1. C	13 8-9	14 9-10	15 8-9	16 9-10	17
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 76: System snaps windows into left-to-right alignment.

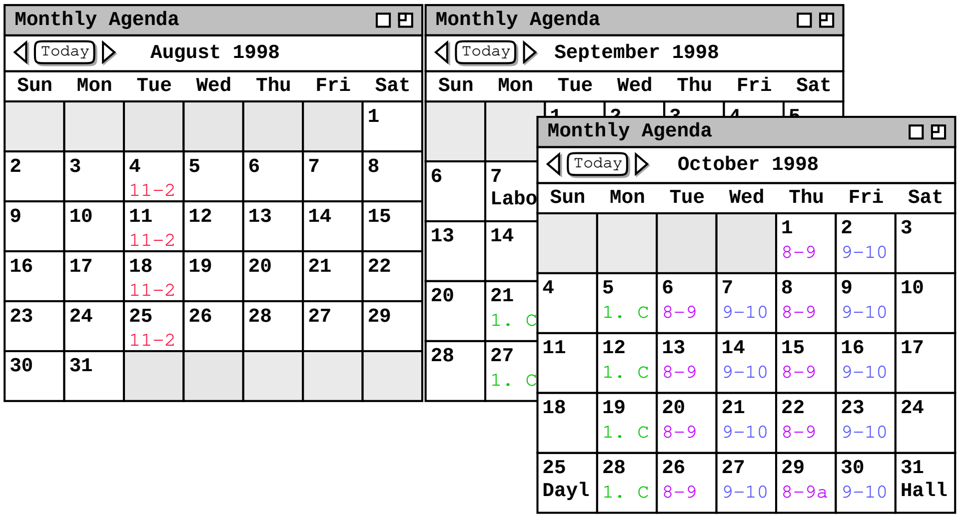


Sun	Mon	Tue	Wed	Thu	Fri	Sat
						1
2	3	4 11-2	5	6	7	8
9	10	11 11-2	12	13	14	15
16	17	18 11-2	19	20	21	22
23	24	25 11-2	26	28	27	29
30	31					

Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5
6	7 Labo					
13	14					
20	21 1. C					
28	27 1. C					

Sun	Mon	Tue	Wed	Thu	Fri	Sat
				1 8-9	2 9-10	3
4	5 1. C	6 8-9	7 9-10	8 8-9	9 9-10	10
11	12 1. C	13 8-9	14 9-10	15 8-9	16 9-10	17
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 77: Holding shift key, user moves top of August window near top of September window.



Monthly Agenda

◁ Today ▷ August 1998

Sun	Mon	Tue	Wed	Thu	Fri	Sat
						1
2	3	4 11-2	5	6	7	8
9	10	11 11-2	12	13	14	15
16	17	18 11-2	19	20	21	22
23	24	25 11-2	26	28	27	29
30	31					

Monthly Agenda

◁ Today ▷ September 1998

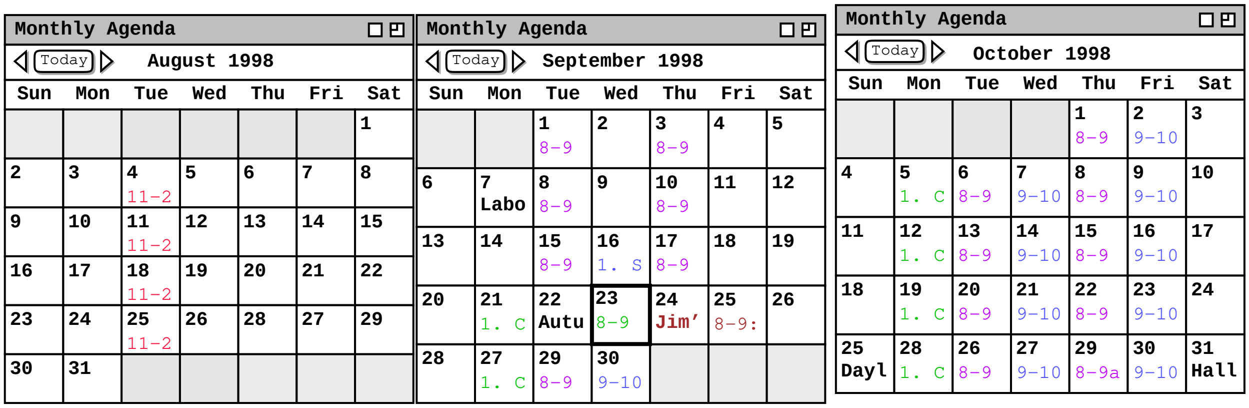
Sun	Mon
6	7 Labo
13	14
20	21 1. C
28	27 1. C

Monthly Agenda

◁ Today ▷ October 1998

Sun	Mon	Tue	Wed	Thu	Fri	Sat
				1 8-9	2 9-10	3
4	5 1. C	6 8-9	7 9-10	8 8-9	9 9-10	10
11	12 1. C	13 8-9	14 9-10	15 8-9	16 9-10	17
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

◁ Today ▷ August 1998

Sun	Mon	Tue	Wed	Thu	Fri	Sat
						1
2	3	4 11-2	5	6	7	8
9	10	11 11-2	12	13	14	15
16	17	18 11-2	19	20	21	22
23	24	25 11-2	26	28	27	29
30	31					

Monthly Agenda

◁ Today ▷ September 1998

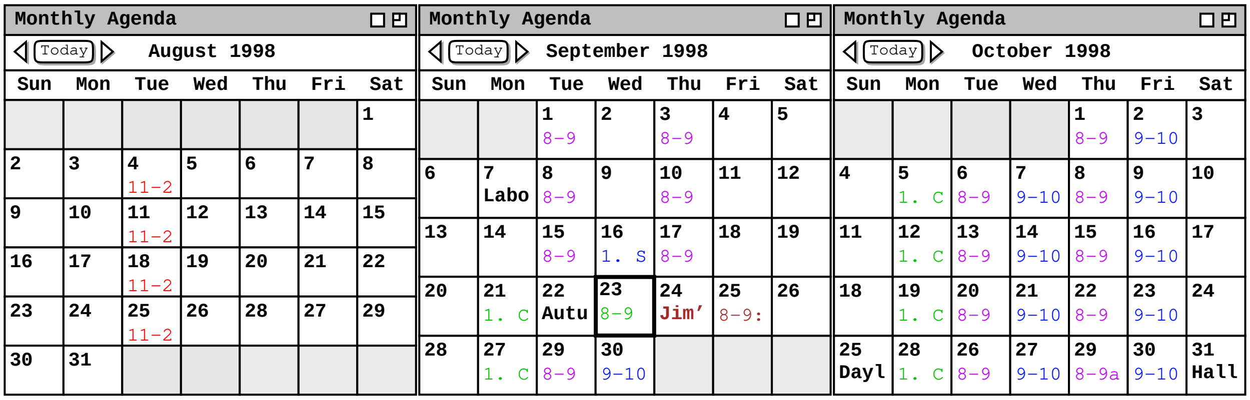
Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1 8-9	2	3 8-9	4	5
6	7 Labo	8 8-9	9	10 8-9	11	12
13	14	15 8-9	16 1. S	17 8-9	18	19
20	21 1. C	22 Autu	23 8-9	24 Jim'	25 8-9:	26
28	27 1. C	29 8-9	30 9-10			

Monthly Agenda

◁ Today ▷ October 1998

Sun	Mon	Tue	Wed	Thu	Fri	Sat
				1 8-9	2 9-10	3
4	5 1. C	6 8-9	7 9-10	8 8-9	9 9-10	10
11	12 1. C	13 8-9	14 9-10	15 8-9	16 9-10	17
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 79: Without shift, user moves top and right of August/September pair near top and left of October.



Monthly Agenda

◁ Today ▷ August 1998

Sun	Mon	Tue	Wed	Thu	Fri	Sat
						1
2	3	4 11-2	5	6	7	8
9	10	11 11-2	12	13	14	15
16	17	18 11-2	19	20	21	22
23	24	25 11-2	26	28	27	29
30	31					

Monthly Agenda

◁ Today ▷ September 1998

Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1 8-9	2	3 8-9	4	5
6	7 Labo	8 8-9	9	10 8-9	11	12
13	14	15 8-9	16 1. S	17 8-9	18	19
20	21 1. C	22 Autu	23 8-9	24 Jim'	25 8-9:	26
28	27 1. C	29 8-9	30 9-10			

Monthly Agenda

◁ Today ▷ October 1998

Sun	Mon	Tue	Wed	Thu	Fri	Sat
				1 8-9	2 9-10	3
4	5 1. C	6 8-9	7 9-10	8 8-9	9 9-10	10
11	12 1. C	13 8-9	14 9-10	15 8-9	16 9-10	17
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 80: System snaps windows into top and left-to-right alignment.

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 show in Figure 78.

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 . 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

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.

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.

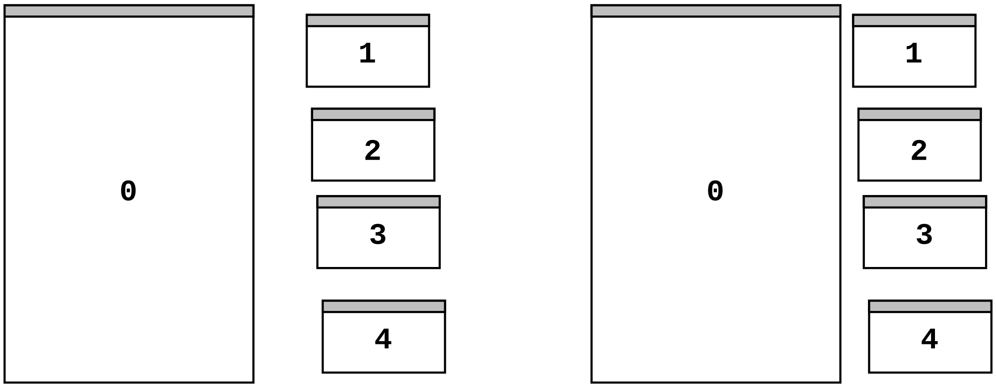
If the user moves a magnetized window in close proximity to the aligned sides of two or more other windows, all windows become magnetically connected. If the user moves a magnetized window in close proximity to the *un*aligned sides of two or more windows, the system establishes the magnetic connection 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 show 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 show 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 transfered 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 transfered 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.

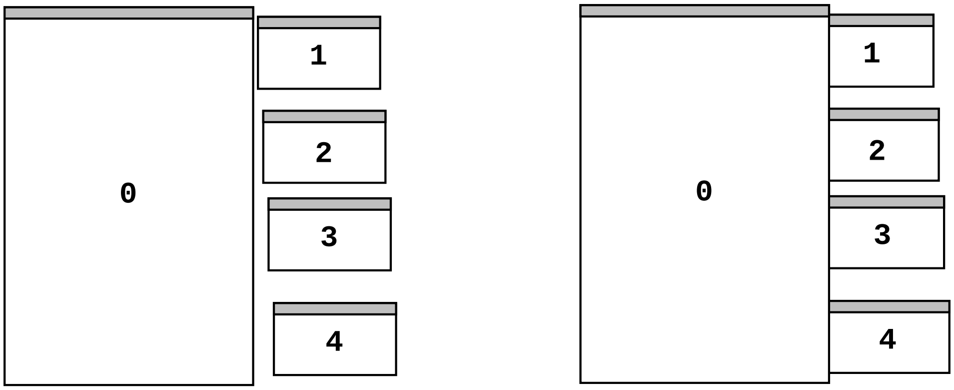
2.3.6.4. The Calendars List

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. The definition of the term "current calendar" used throughout the requirements is "the



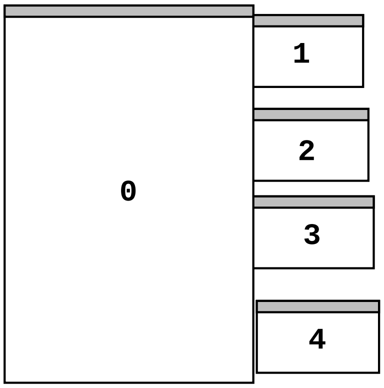
a. Windows 1- 4 are slightly out of alignment.

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.

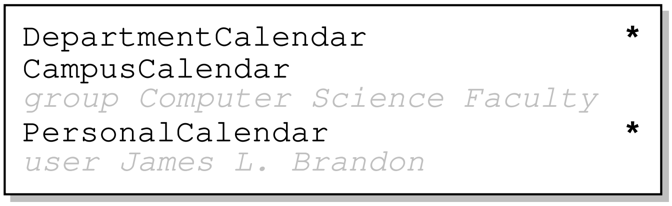


Figure 82: Calendars submenu.

calendar listed at the top of the calendar list". In this example, the user has opened five different calendars, with `DepartmentCalendar` current.

If a calendar has unsaved changes, its name in the calendars list is marked with a boldface asterisk character. The precise definition of unsaved changes is given in Section 2.8.4. 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, the same root name with different extensions, or if 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.

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 currently defined categories to be those of the selected calendar (see Section 2.5.5 for explanation of calendar-specific categories)

e. 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)

f. 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)

g. updates the windowing mode setting to that of the selected calendar (see Section 2.3.6.2.4)

h. updates the current host to be that of the current calendar, if any (see Section 2.6.6.1 for explanation of calendar-to-host association)

i. updates the banners of all windows that contain the name of the current calendar (see Sections 2.3.6.5 and 2.7.4.2 for explanation of window banner contents and viewing options related to showing the current calendar in window banners)

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 covers details of command enabling and disabling.

The association between a calendar and its windows is based on 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. Zero or more initial view windows are 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 created or opened by these commands becomes the current calendar; all windows displayed as a result of these commands are associated with the created or opened 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 indirectly calendar-specific. These commands apply to a central host computer that must be associated with a particular calendar. When an administrative command window is current, the current calendar is the one associated with the host to which the administrative commands apply. Section 2.6.6.2 fully explains the association between the current calendar and a remote host, if any.

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. The calendar name may also appear in other window banners, depending on the setting of the window-viewing options described in Section 2.7.4.2.

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 2.6.6.1. 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. Section 2.6 covers details of the Calendar Tool Administration program and central host computers.

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:

o meetings

o tasks
o 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:

o a list of the meeting attendees
o the ID of the user of who schedules the meeting
o the date and time scheduling takes place
o 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 . The following paragraph summarizes the environment features that are relevant to the meeting-scheduling scenarios presented in this section.

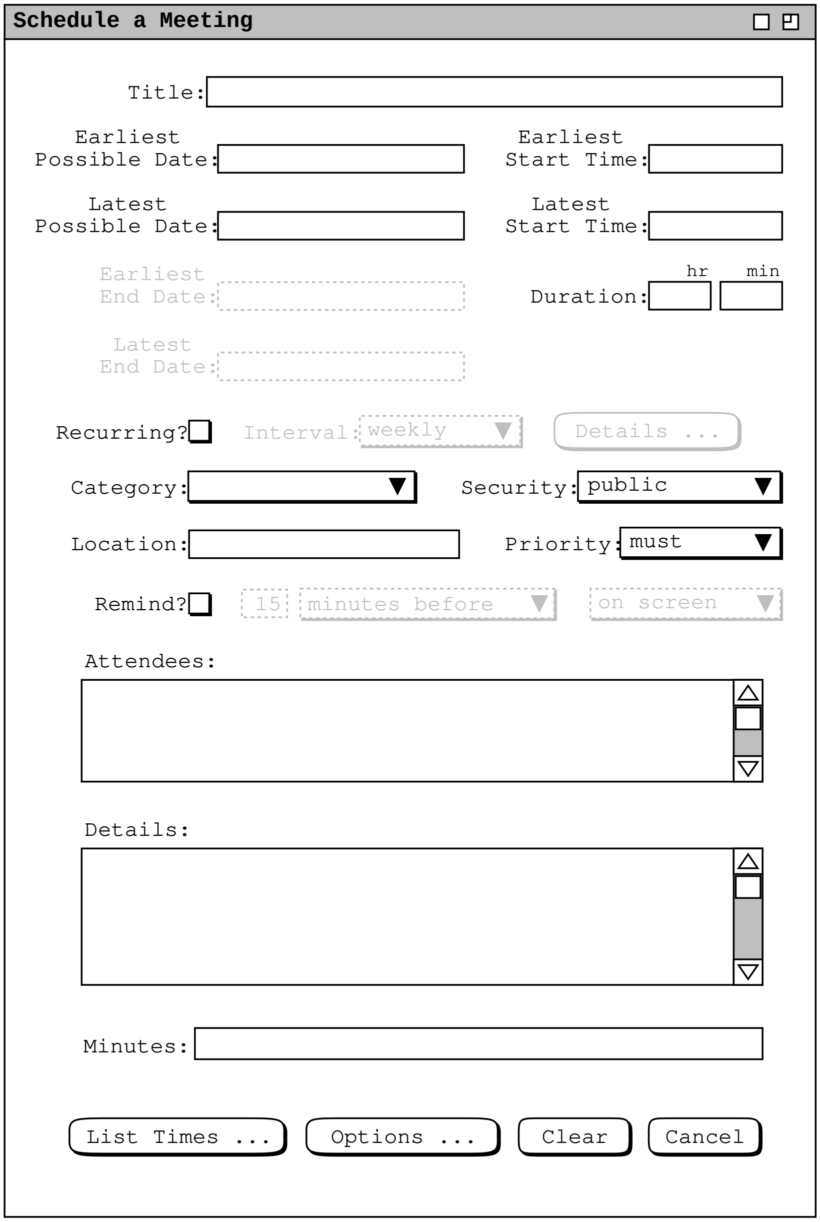
The Calendar Tool multi-user operating environment consists of a central host computer and user client computers. Calendar Tool users, including meeting schedulers, run the Calendar Tool on their local client computers and connect to a common central host computer via some form of computer network. User calendars are stored in files on their local computers. In addition, the central host has a copy of 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, i.e., the meeting is "penciled in". For attending users who are actively connected to the central host, the meeting is also penciled in on their local calendars. Once a meeting is penciled in, a meeting notification is sent to each attending user. The user can respond to the notification by accepting the meeting or declining to accept it. Accepting means the meeting is changed from a penciled-in meeting to a regularly-scheduled one. Declining means the penciled-in meeting is removed from the calendar. If a user is not connected to the central host when a meeting is scheduled, the meeting is penciled in on the central host, and will subsequently be penciled in on the local calendar the next time the user connects. Notifications sent when a user is not connected are queued on the central host for delivery when the connection is next established.

In the scenarios that follow, meeting schedulers and meeting attendees are all assumed to be actively connected to the same central host computer. Details of establishing a central host connection are presented in Section 2.6.6.1. Since meeting scheduling is an inherently multi-user operation, it can only be performed when the scheduler is connected to a central host computer. When not connected to a central host, meeting scheduling is disabled.

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



Interval: weekly

Security: public

Priority: must

15 minutes before on screen

Figure 83: Meeting scheduling request dialog.

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 5:

- `Title`
- `Duration`
- `Category`
- `Security`

o `Location`
o `Priority`
o `Remind`, and its accompanying fields
o `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 2.5.2.2.1 Details of how users view meeting minutes are covered in Section 2.5.1.2.

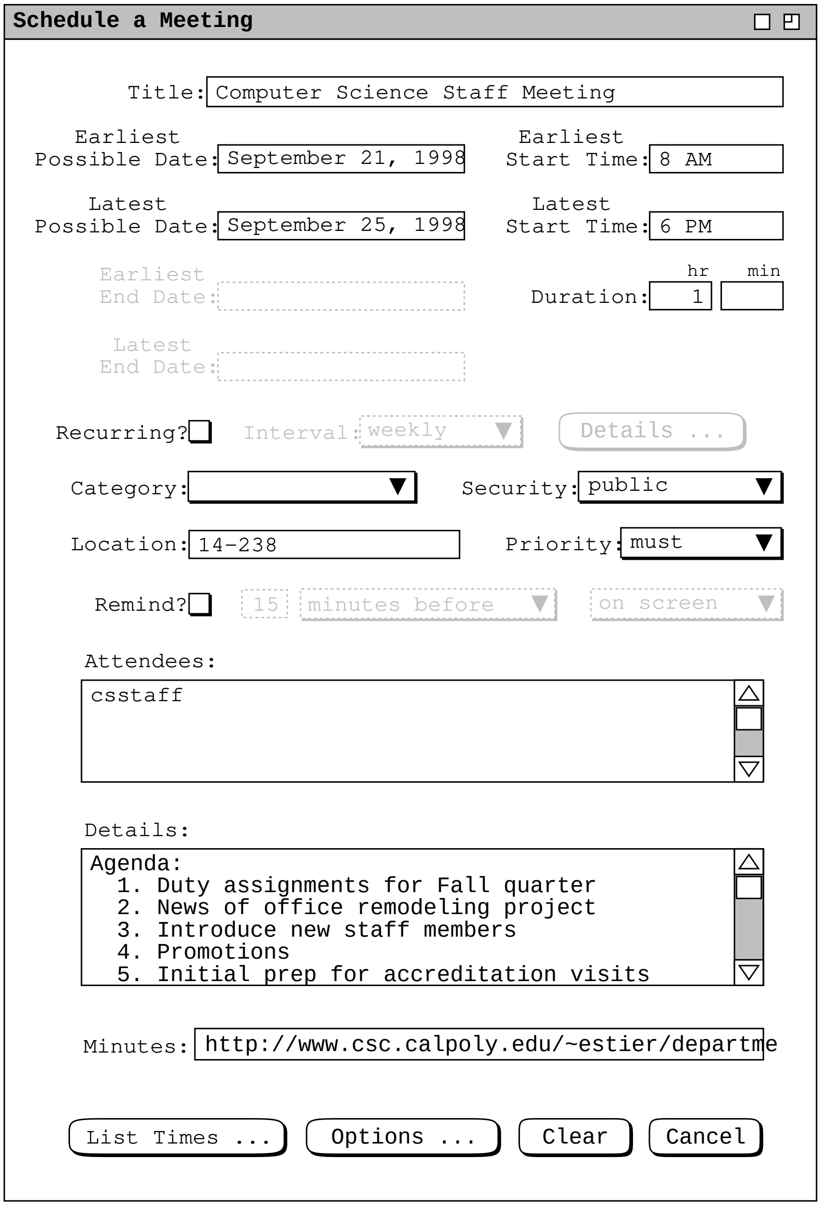
Figure 84 shows the scheduler having entered information for a non-recurring meeting of the computer science staff. The earliest and latest possible dates are the Monday and Friday of the week of September 20. The earliest and latest start times are 8 AM to 6 PM. The category is blank and there is no 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 possibles list are between the earliest and latest possible dates entered in the scheduling request dialog.
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 20, which value can be changed as an option setting. When there are more possible times than maximum number setting, the list is truncated based on the sorted order. For example, if the maximum setting is 20 and there are 35 possible times, the list contains the 20 earliest times, with the 15 times later than that truncated from the list.

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.



Title: Computer Science Staff Meeting

Earliest Possible Date: September 21, 1998

Earliest Start Time: 8 AM

Latest Possible Date: September 25, 1998

Latest Start Time: 6 PM

Earliest End Date:

Latest End Date:

Duration: hr 1 min

Recurring? ☐ Interval: weekly

Category:

Security: public

Location: 14-238

Priority: must

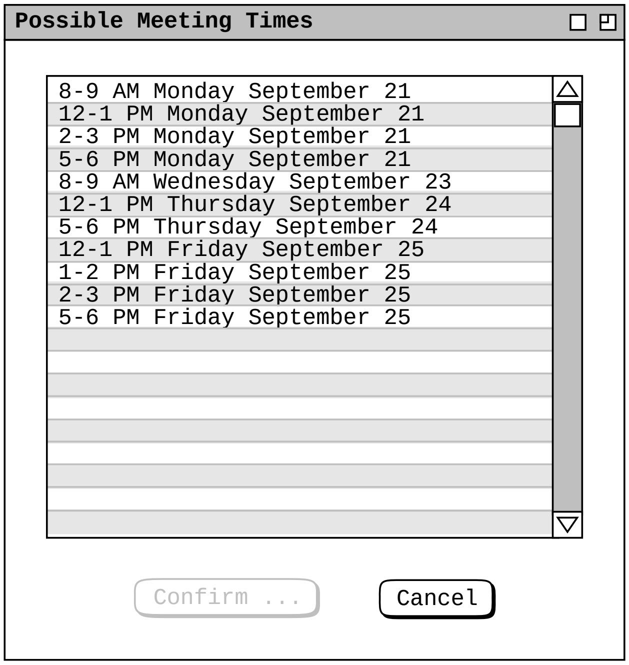
Remind? ☐ 15 minutes before on screen

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

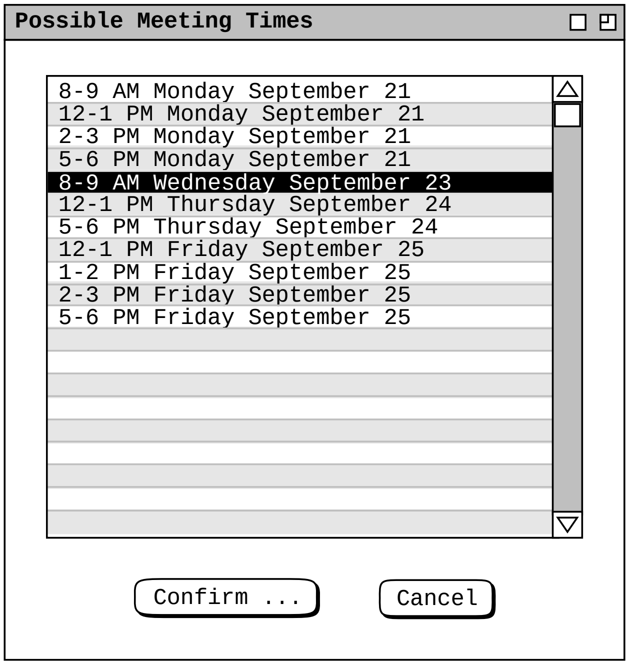
Figure 84: Scheduling a staff meeting.



8-9 AM Monday September 21
12-1 PM Monday September 21
2-3 PM Monday September 21
5-6 PM Monday September 21
8-9 AM Wednesday September 23
12-1 PM Thursday September 24
5-6 PM Thursday September 24
12-1 PM Friday September 25
1-2 PM Friday September 25
2-3 PM Friday September 25
5-6 PM Friday September 25

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

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.



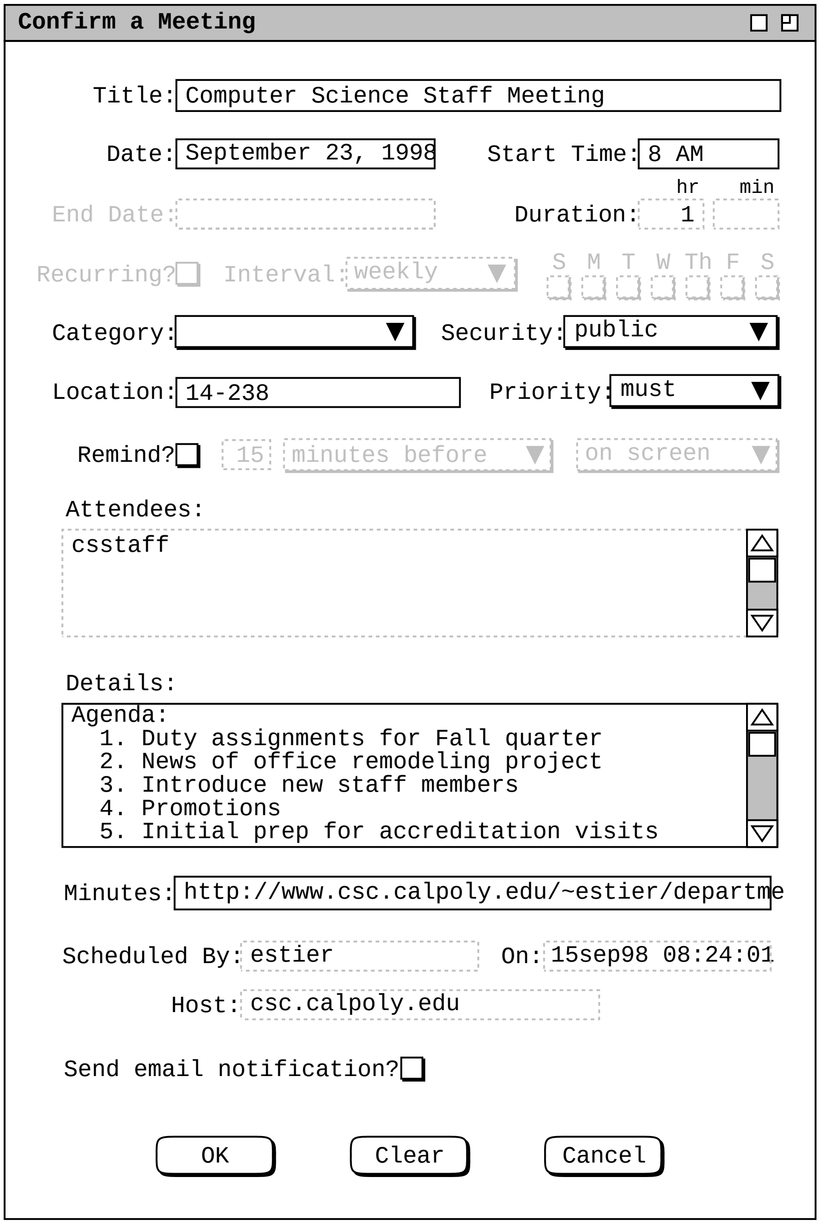
8-9 AM Monday September 21
12-1 PM Monday September 21
2-3 PM Monday September 21
5-6 PM Monday September 21
8-9 AM Wednesday September 23
12-1 PM Thursday September 24
5-6 PM Thursday September 24
12-1 PM Friday September 25
1-2 PM Friday September 25
2-3 PM Friday September 25
5-6 PM Friday September 25

Figure 86: Selection in the possible meeting times list.

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 formated as

ddmmmyy hh:mm:ss



Title: Computer Science Staff Meeting

Date: September 23, 1998 Start Time: 8 AM

End Date: Duration: hr 1 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? ☐

Figure 87: Staff meeting confirmation dialog.

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 discusses 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. 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 .

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.

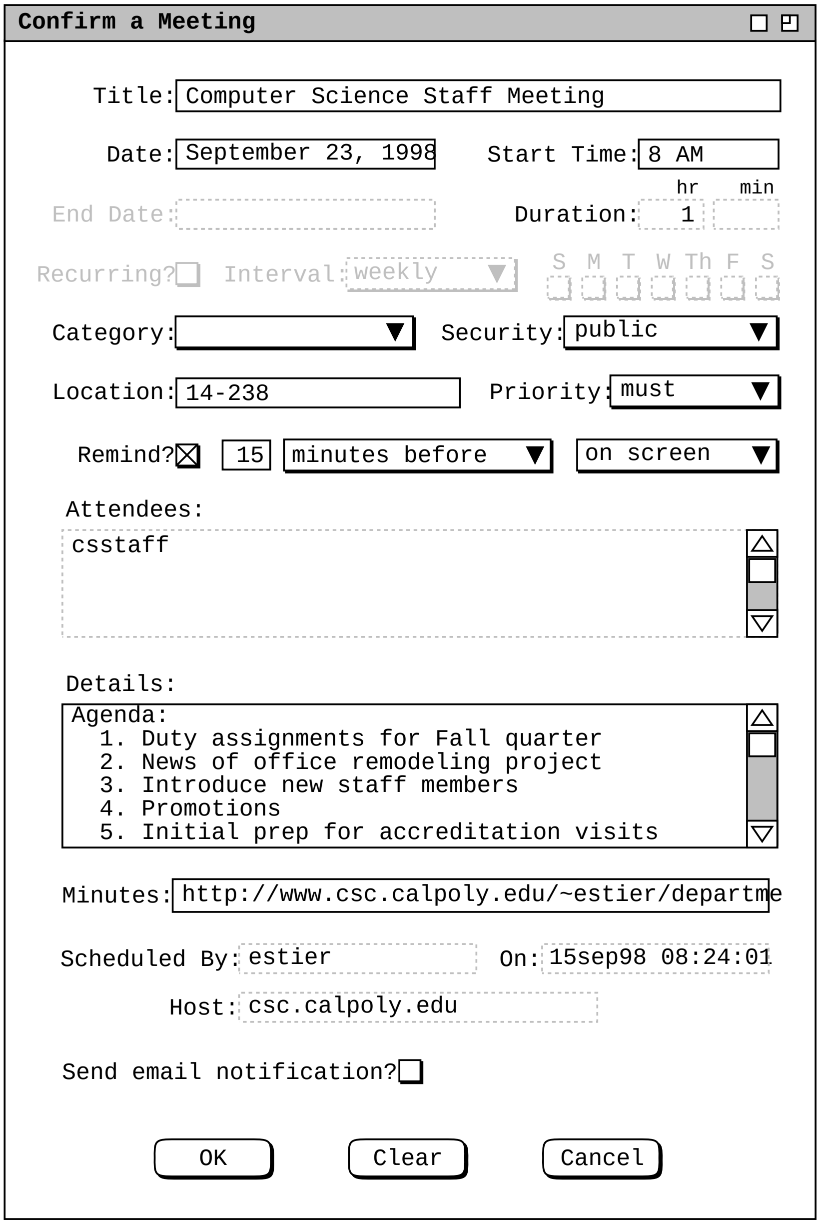
The precise effects of scheduling a meeting are the following:

a. The system adds the meeting to each attendee's calendar on the central host computer and marks each meeting as "not-yet-accepted"; i.e., the meeting is "penciled in".
b. For all attendees who are actively connected to the central host, the system also pencils in the meeting on the user's local calendar; for attendees not connected, the local penciling in occurs when a connection is next established.
c. The system notifies each attendee except the scheduler, as discussed in Section 2.4.1.5.
d. 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.
e. 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.
f. 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 .

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. adds a not-yet-accepted meeting to the local calendars of all members of `csstaff` who are currently connected to the central host, except the scheduler; attendees who are not connected will receive the not-yet-accepted meeting when a connection is next established
c. notifies all members of `csstaff`, except the scheduler;



Confirm a Meeting

Title: Computer Science Staff Meeting

Date: September 23, 1998

Start Time: 8 AM

End Date:

Duration: hr 1 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? ☐

Figure 88: Meeting confirmation dialog with some edits.

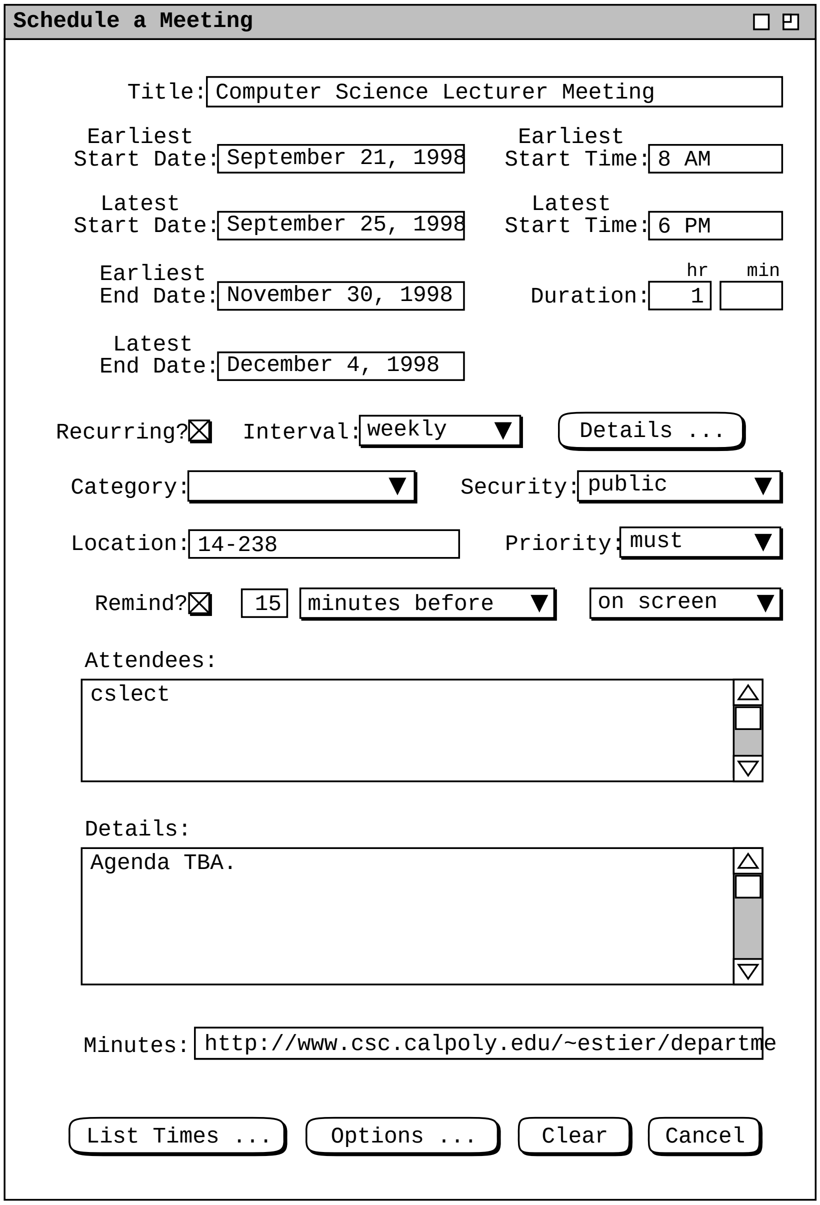
d. adds the meeting to the scheduler's central host and local calendars;

e. adds the meeting to the `csstaff` group calendar;

f. 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 considers all days of the week in the normal day range for



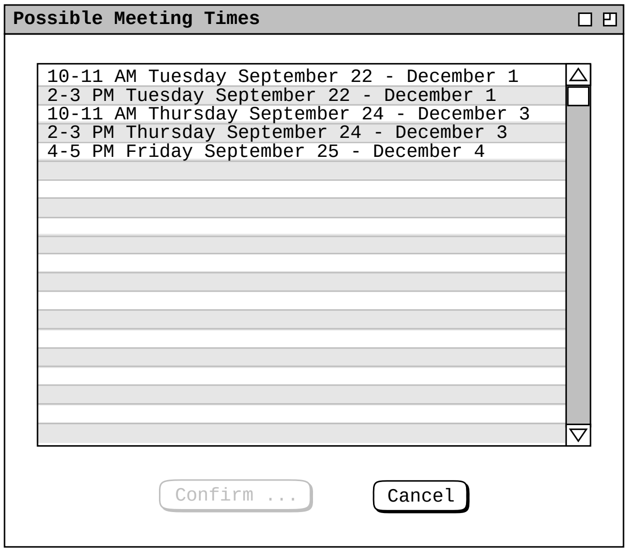
Title: Computer Science Lecturer Meeting
Earliest Start Date: September 21, 1998
Earliest Start Time: 8 AM
Latest Start Date: September 25, 1998
Latest Start Time: 6 PM
Earliest End Date: November 30, 1998
Duration: hr 1 min
Latest End Date: December 4, 1998
Recurring? ☒ Interval: weekly
Category:
Security: public
Location: 14-238
Priority: must
Remind? ☒ 15 minutes before on screen
Attendees: cslect
Details: Agenda TBA.
Minutes: http://www.csc.calpoly.edu/~estier/departme

Figure 89: Scheduling another meeting.

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.

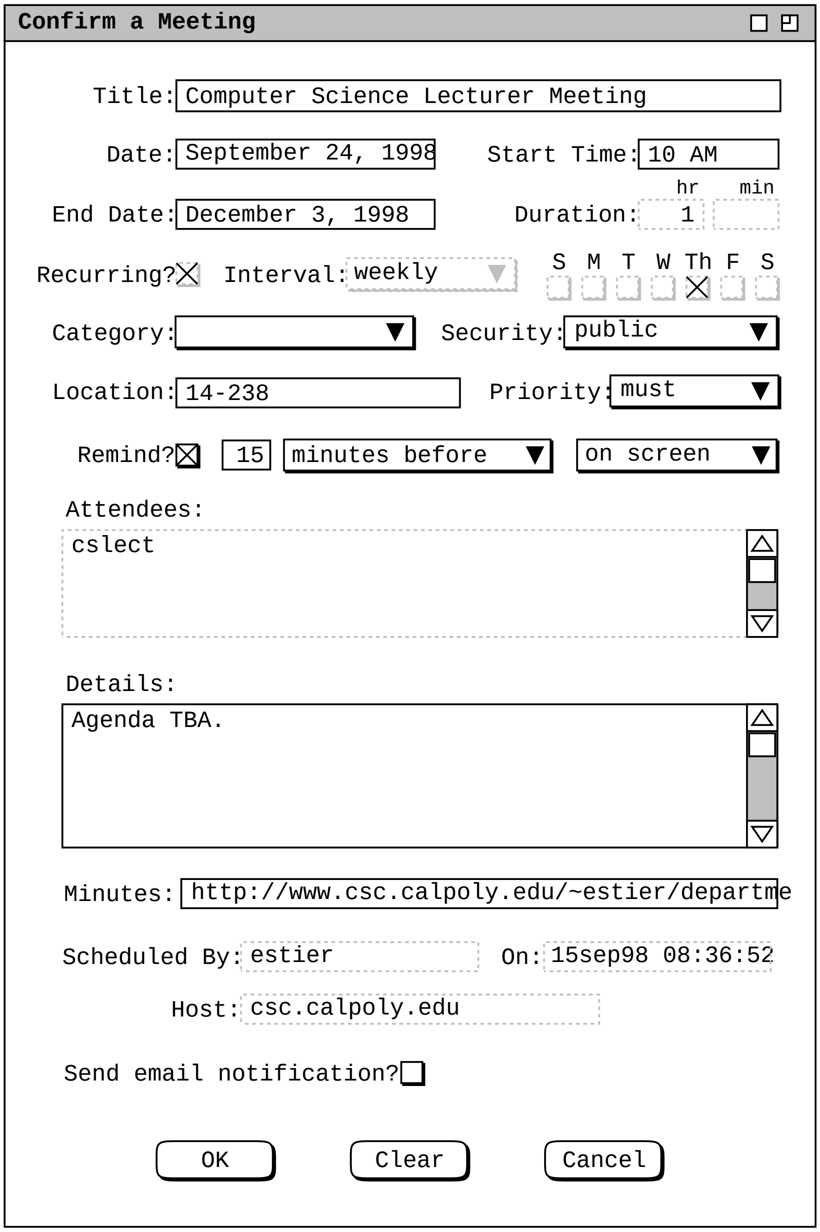
When the scheduler presses the '`List Times ...`' button for the lecturer meeting, the system responds with the list of possible meeting times shown in Figure 90. For a weekly recurring meeting, each item in the list consists of a time range, day(s) of the week, start date, and end date. The scheduler 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



10-11 AM Tuesday September 22 - December 1
2-3 PM Tuesday September 22 - December 1
10-11 AM Thursday September 24 - December 3
2-3 PM Thursday September 24 - December 3
4-5 PM Friday September 25 - December 4

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



Confirm a Meeting

Title: Computer Science Lecturer Meeting

Date: September 24, 1998   Start Time: 10 AM

End Date: December 3, 1998   Duration: hr 1 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:

cslect

Details:

Agenda TBA.

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

Scheduled By: estier   On: 15sep98 08:36:52

Host: csc.calpoly.edu

Send email notification? ☐

Figure 91: Lecturer meeting confirmation dialog.

include the scheduler;

b. adds a not-yet-accepted meeting to the local calendars of all connected members of `cslect`, which does not include the scheduler;

c. notifies all members of `cslect`, which does not include the scheduler;

d. adds the meeting to the scheduler's calendar, even though the scheduler is not an attendee;

e. adds the meeting to the `cslect` group calendar;

f. 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 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, canceling the scheduling operation is a three-step process: first the confirmation is canceled, then the possibles 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 canceling scheduled meetings are discussed in Section .

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.

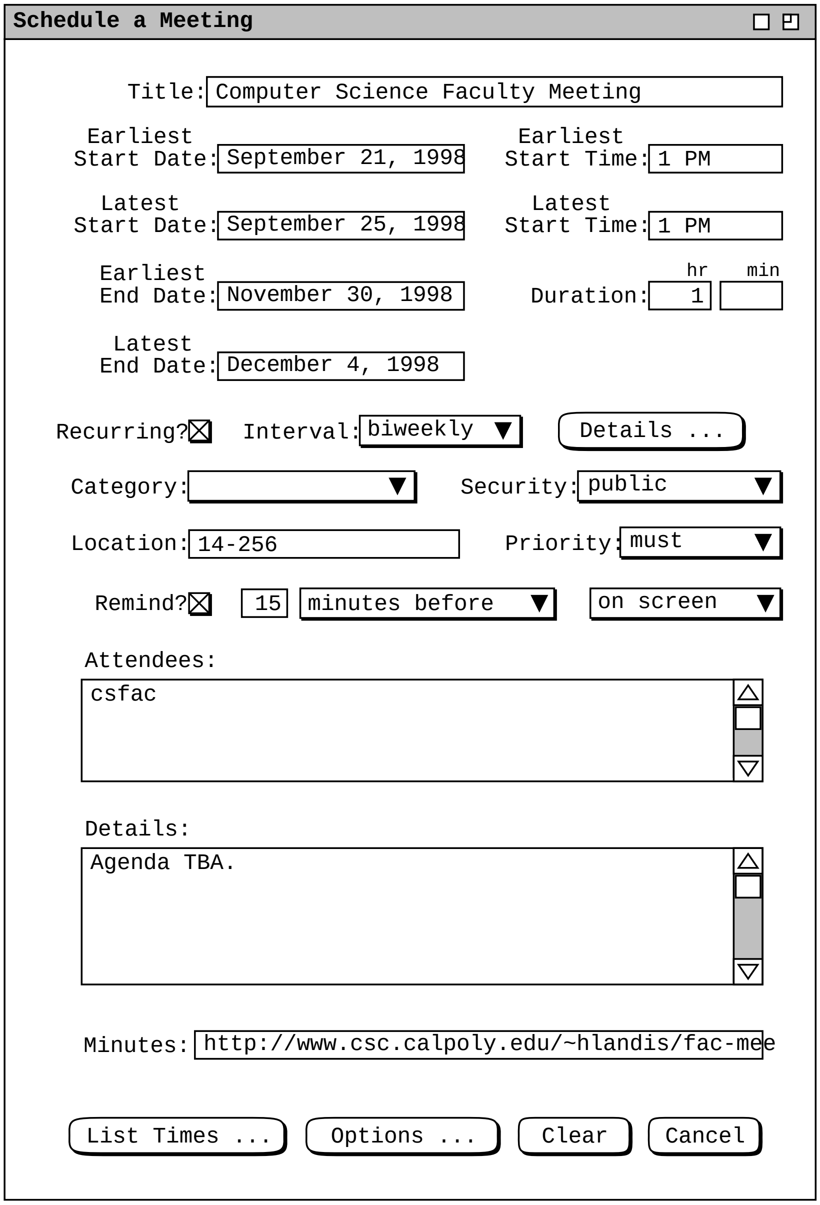
To begin, the scheduler selects '`Schedule Meeting`' in the command menu and fills in the dialog as shown in Figure 92. In this case, the scheduler wants to schedule a recurring meeting on a specific day and time. The single specific time is chosen by entering the same 1 PM value for both the earliest 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.

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.

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



Title: Computer Science Faculty Meeting

Earliest Start Date: September 21, 1998

Earliest Start Time: 1 PM

Latest Start Date: September 25, 1998

Latest Start Time: 1 PM

Earliest End Date: November 30, 1998

Duration: hr 1 min

Latest End Date: December 4, 1998

Recurring? ☒ Interval: biweekly

Category:

Security: public

Location: 14-256

Priority: must

Remind? ☒ 15 minutes before on screen

Attendees:

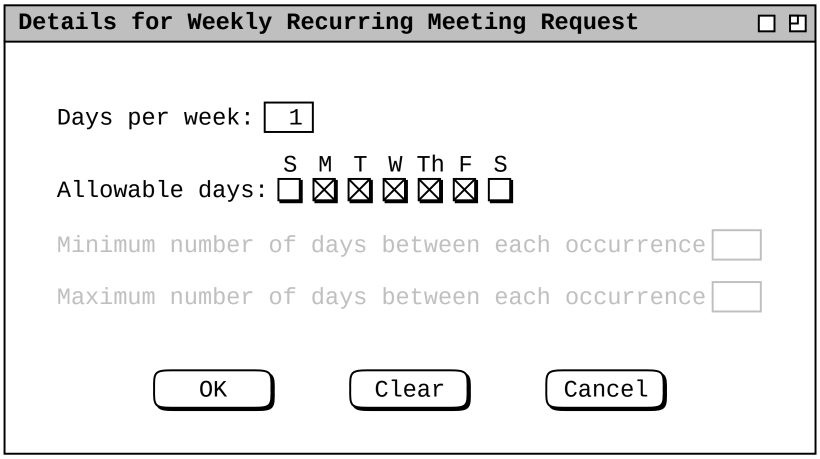
csfac

Details:

Agenda TBA.

Minutes: http://www.csc.calpoly.edu/~hlandis/fac-mee

Figure 92: Scheduling a Faculty Meeting.



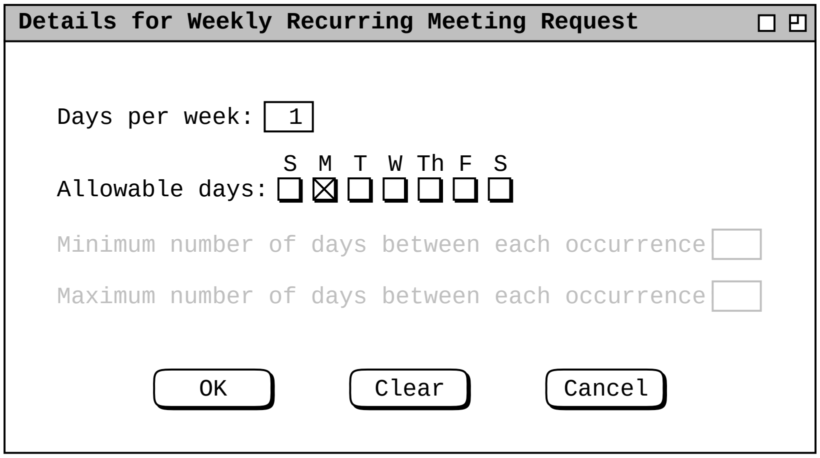
Days per week: 1

Allowable days: S ☐ M ☒ T ☒ W ☒ Th ☒ F ☒ S ☐

Minimum number of days between each occurrence

Maximum number of days between each occurrence

Figure 93: Details of a weekly recurring schedule.



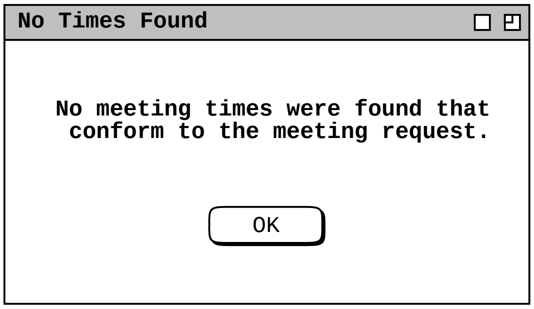
Days per week: 1

Allowable days: S ☐ M ☒ T ☐ W ☐ Th ☐ F ☐ S ☐

Minimum number of days between each occurrence

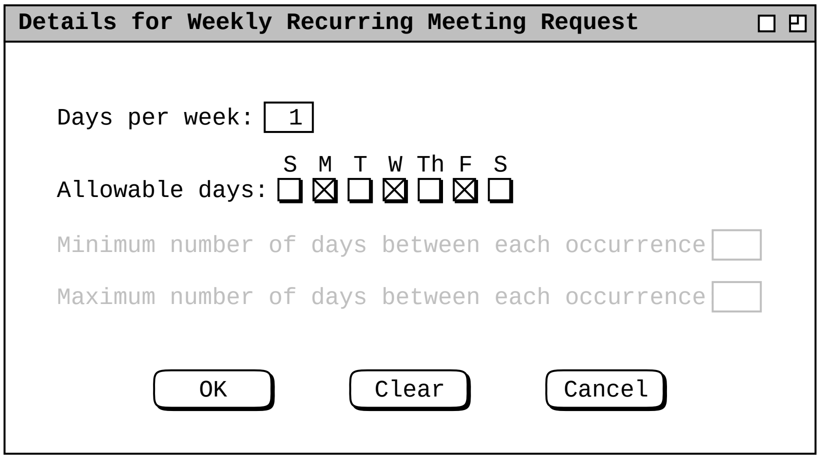
Maximum number of days between each occurrence

Figure 94: Weekly recurring details filled in.



No meeting times were found that conform to the meeting request.

Figure 95: No times found dialog.



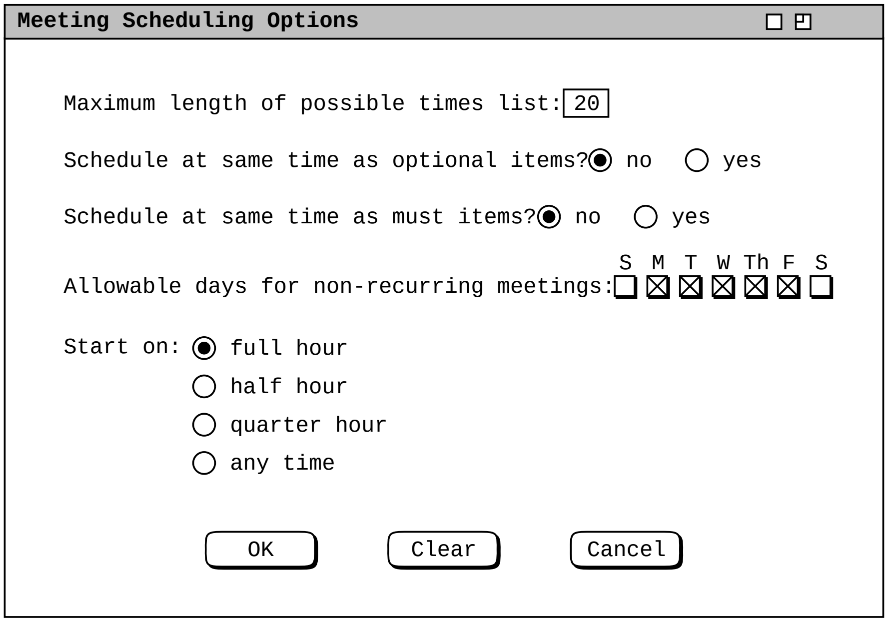
Days per week: 1

Allowable days: S ☐ M ☒ T ☐ W ☒ Th ☐ F ☒ S ☐

Minimum number of days between each occurrence

Maximum number of days between each occurrence

Figure 96: Weekly recurring details modified.



Maximum length of possible times list: 20

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 97: Meeting scheduling options.

with ‘`optional`’ or ‘`must`’ priority on attendee calendars. If the default value of ‘`no`’ is selected for both options, the system does 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 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.

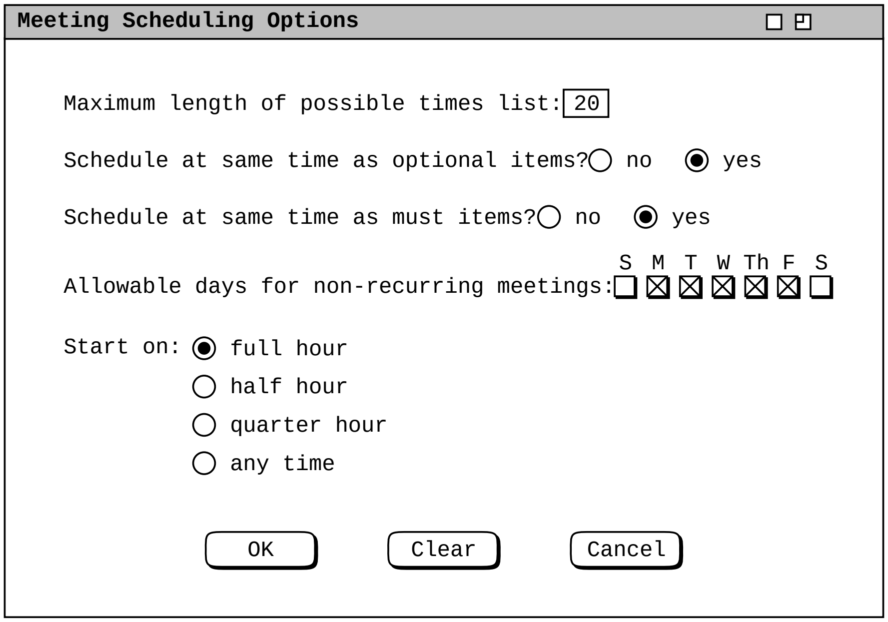
The final scheduling option indicates the time increment on which meetings can start. The default is for meetings to start on the full hour. The other selections are on the half hour, quarter hour, or at any time. For the '`any time`' selection, the smallest possible increment is five minutes.

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

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.



Meeting Scheduling Options

Maximum length of possible times list: 20

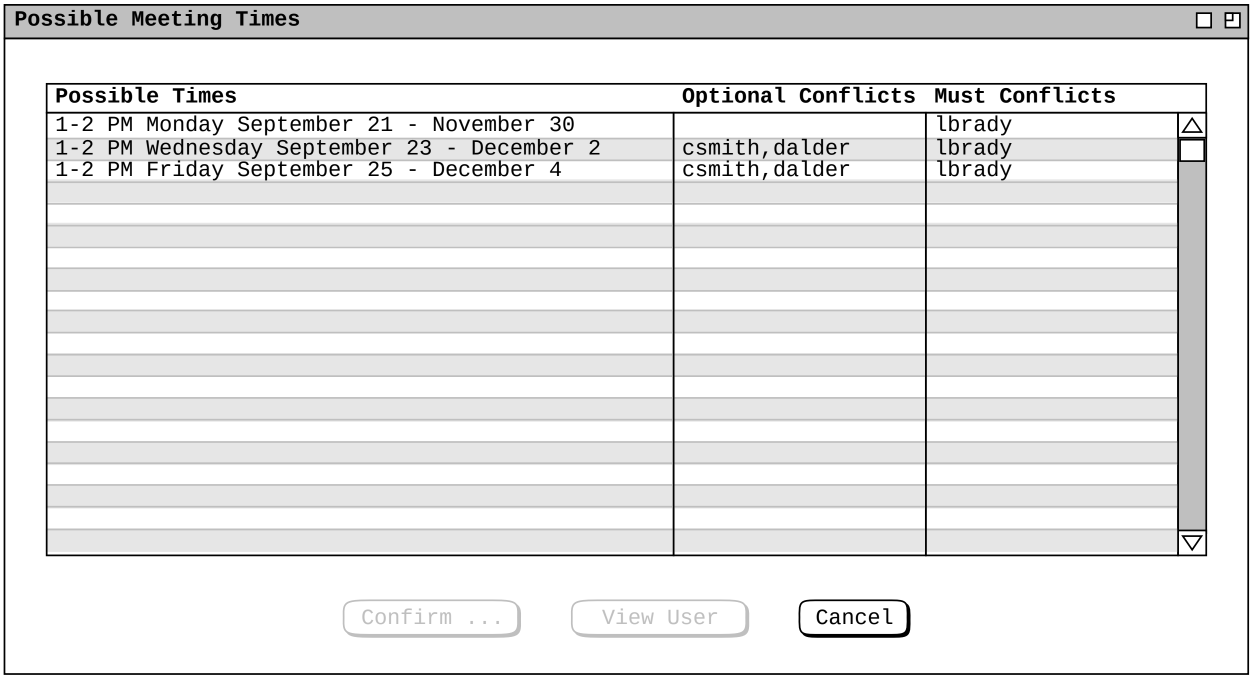
Schedule at same time as optional items? ○ no ● yes

Schedule at same time as must items? ○ no ● yes

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

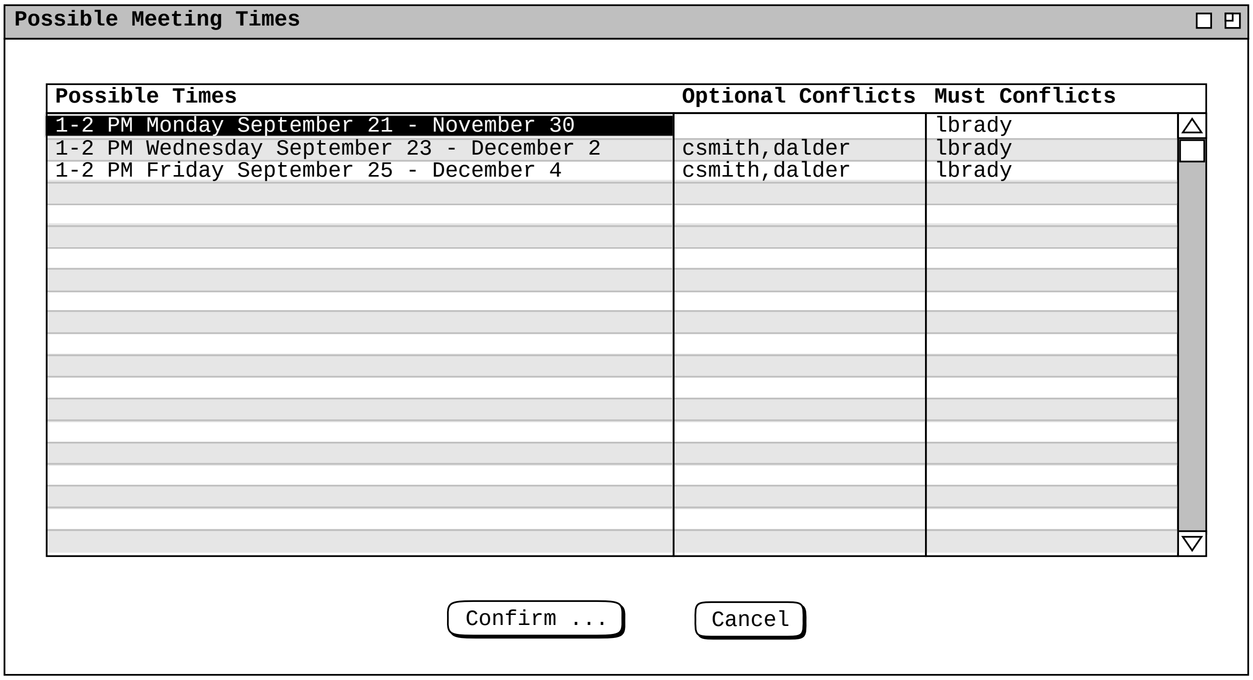
Start on: ● full hour
○ half hour
○ quarter hour
○ any time

Figure 98: Modified meeting scheduling options.



Possible Times	Optional Conflicts	Must Conflicts
1-2 PM Monday September 21 - November 30		lbrady
1-2 PM Wednesday September 23 - December 2	csmith,dalder	lbrady
1-2 PM Friday September 25 - December 4	csmith,dalder	lbrady

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



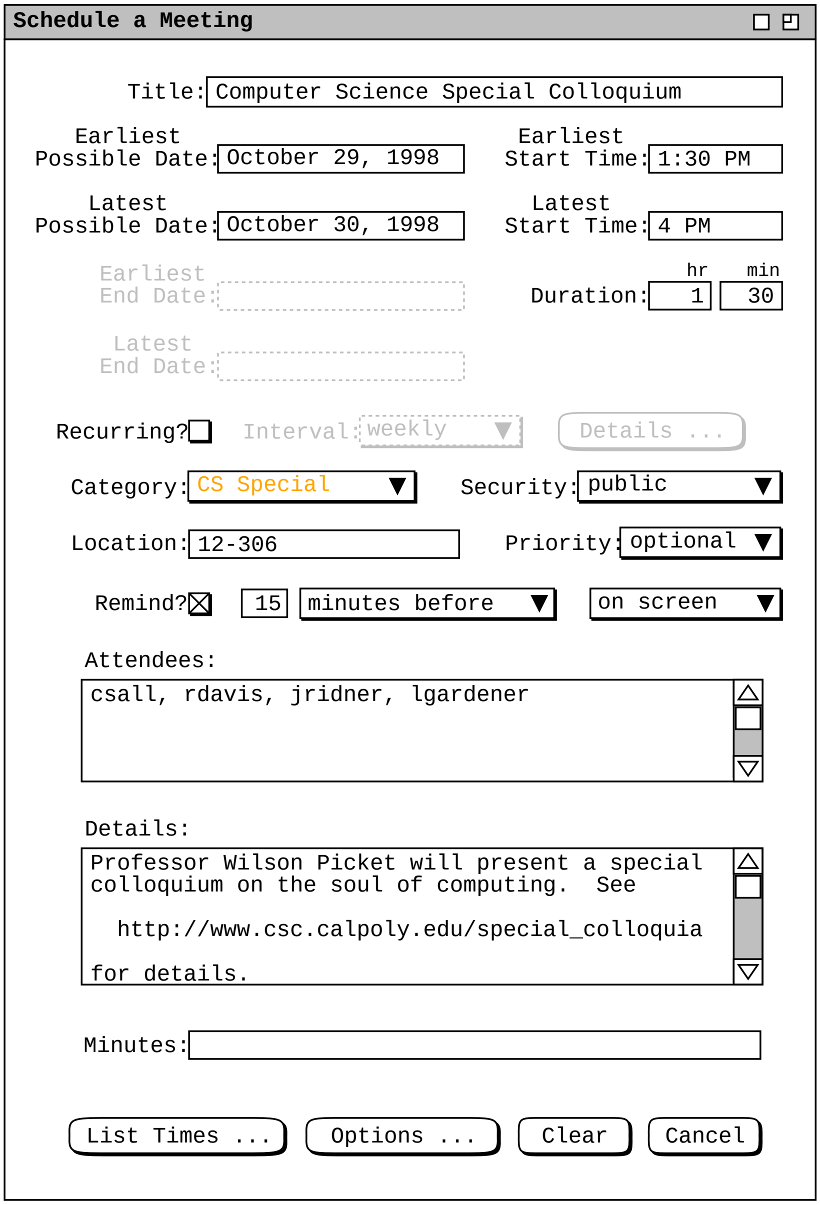
Possible Times	Optional Conflicts	Must Conflicts
1-2 PM Monday September 21 - November 30		lbrady
1-2 PM Wednesday September 23 - December 2	csmith,dalder	lbrady
1-2 PM Friday September 25 - December 4	csmith,dalder	lbrady

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`'.



Schedule a Meeting

Title: Computer Science Special Colloquium

Earliest Possible Date: October 29, 1998

Earliest Start Time: 1:30 PM

Latest Possible Date: October 30, 1998

Latest Start Time: 4 PM

Earliest End Date:

Duration: hr 1 min 30

Latest End Date:

Recurring? ☐ Interval: weekly

Category: CS Special

Security: public

Location: 12-306

Priority: optional

Remind? ☒ 15 minutes before on screen

Attendees:
csall, rdavis, jridner, lgardener

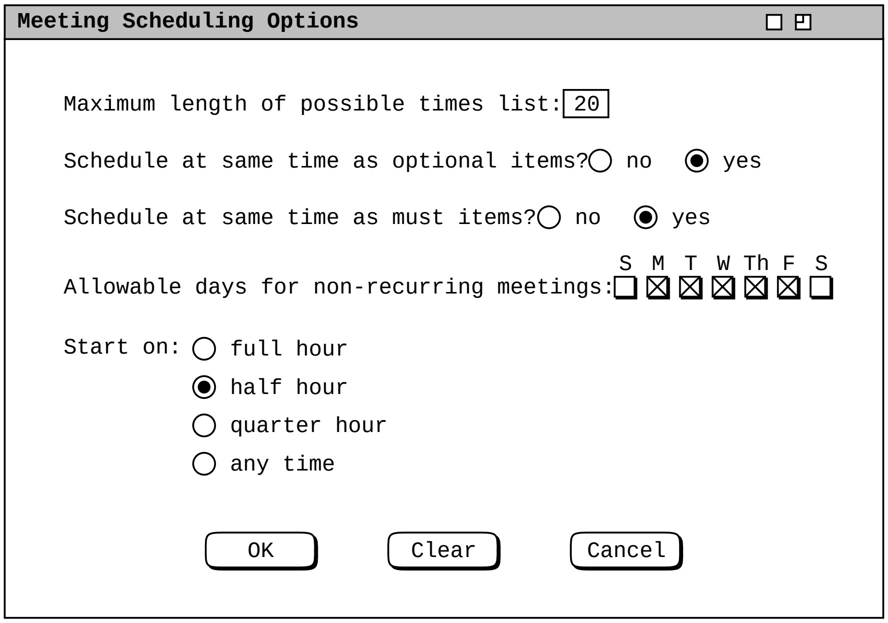
Details:
Professor Wilson Picket will present a special colloquium on the soul of computing. See

http://www.csc.calpoly.edu/special_colloquia

for details.

Minutes:

Figure 101: Scheduling a special colloquium.



Maximum length of possible times list: 20

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.

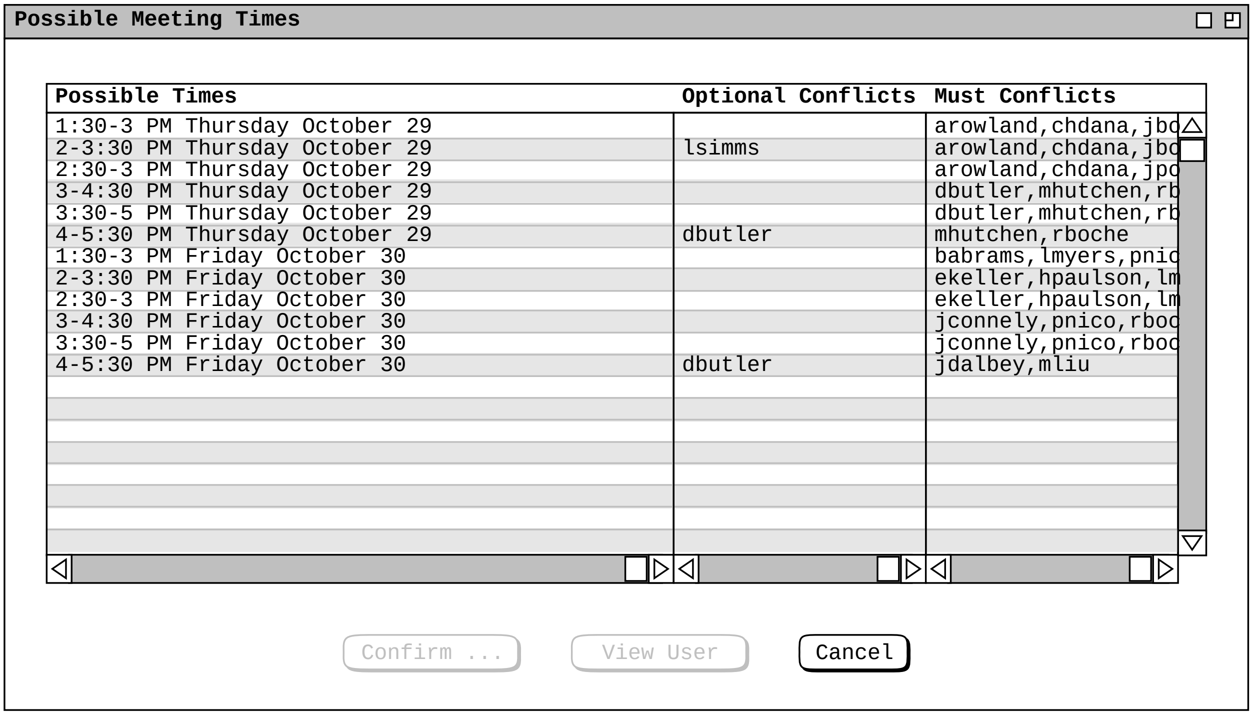
As defined in Section 2.6.3, `csall` is a Calendar Tool supergroup, comprised of the subgroups `csfaculty`, `cslect`, and `csstaff`. These groups are in turn comprised of individual Calendar Tool users. When a supergroup is listed among the attendees of a meeting, individual attendees are all members 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 meeting to the sub(...sub)group calendars.

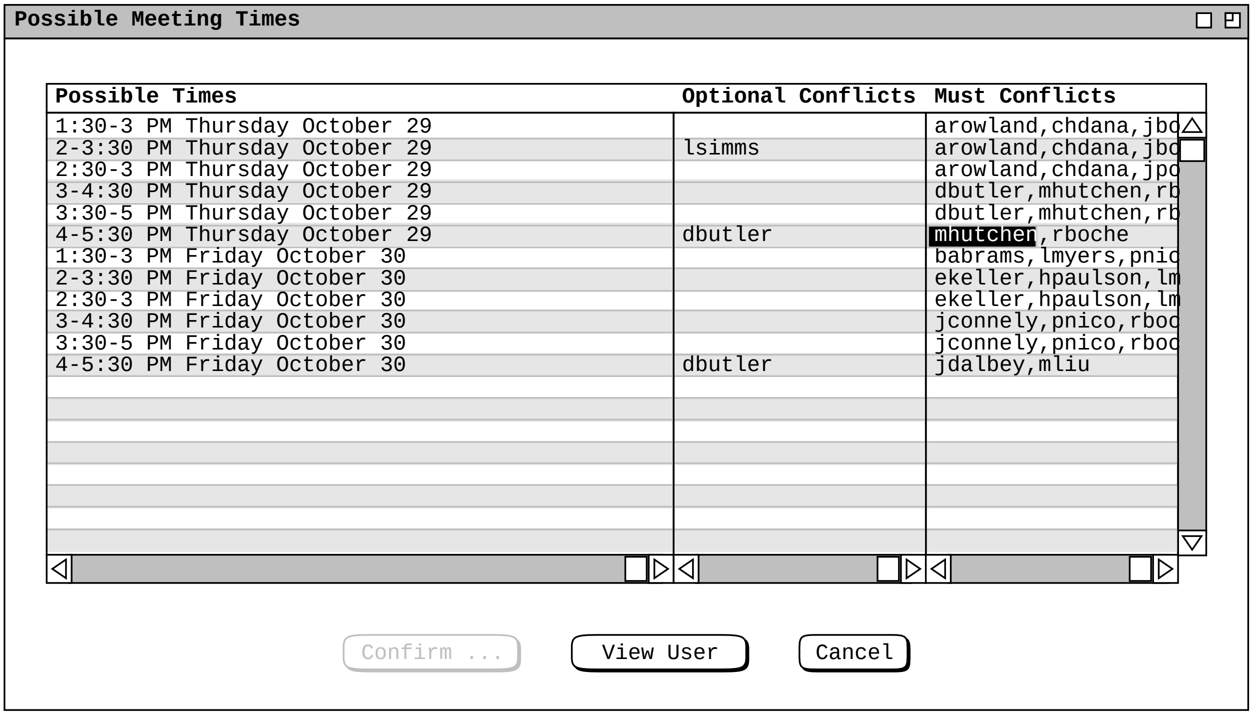
2.4.1.4. A Non-Leader Scheduling a Meeting for Selected Individuals

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.



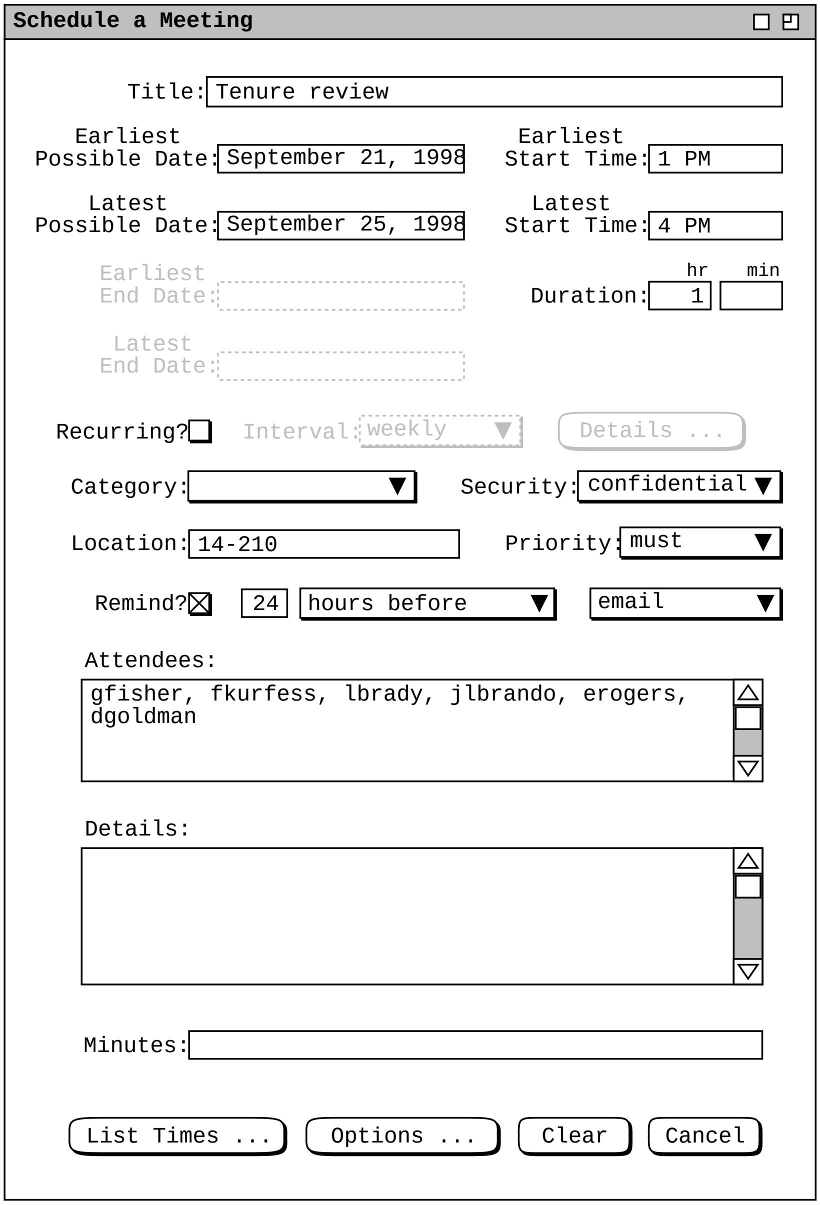
Possible Times	Optional Conflicts	Must Conflicts
1:30-3 PM Thursday October 29		arowland,chdana,jbo
2-3:30 PM Thursday October 29	lsimms	arowland,chdana,jbo
2:30-3 PM Thursday October 29		arowland,chdana,jpo
3-4:30 PM Thursday October 29		dbutler,mhutchen,rb
3:30-5 PM Thursday October 29		dbutler,mhutchen,rb
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,pnico,rboc
3:30-5 PM Friday October 30		jconnely,pnico,rboc
4-5:30 PM Friday October 30	dbutler	jdalbey,mliu

Figure 103: Possible times for the special colloquium.



Possible Times	Optional Conflicts	Must Conflicts
1:30-3 PM Thursday October 29		arowland,chdana,jbo
2-3:30 PM Thursday October 29	lsimms	arowland,chdana,jbo
2:30-3 PM Thursday October 29		arowland,chdana,jpo
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,pnico,rboc
3:30-5 PM Friday October 30		jconnely,pnico,rboc
4-5:30 PM Friday October 30	dbutler	jdalbey,mliu

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



Schedule a Meeting

Title: Tenure review

Earliest Possible Date: September 21, 1998

Earliest Start Time: 1 PM

Latest Possible Date: September 25, 1998

Latest Start Time: 4 PM

Earliest End Date:

Latest End Date:

Duration: hr 1 min

Recurring? ☐ Interval: weekly

Category:

Security: confidential

Location: 14-210

Priority: must

Remind? ☒ 24 hours before email

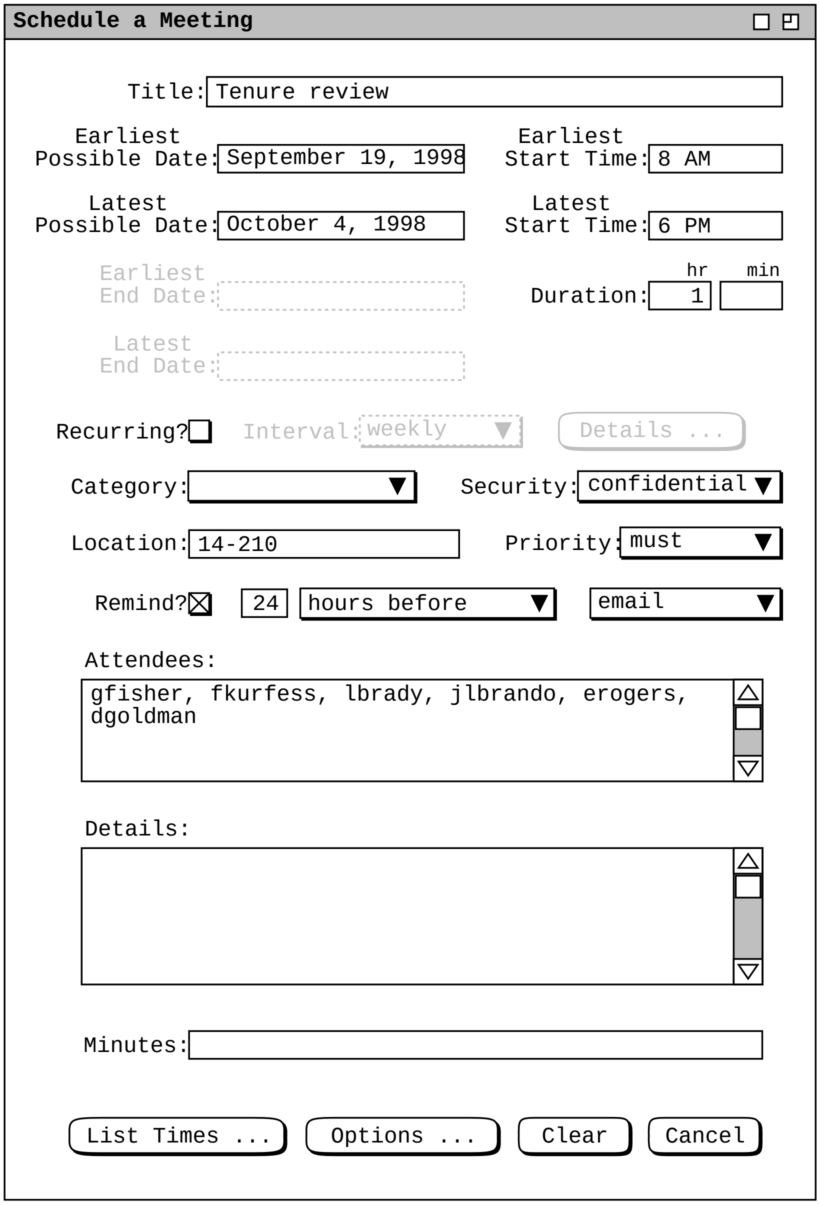
Attendees: gfisher, fkurfess, lbrady, jlbrando, erogers, dgoldman

Details:

Minutes:

Figure 105: Scheduling a tenure review meeting.

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.



Schedule a Meeting

Title: Tenure review

Earliest Possible Date: September 19, 1998

Earliest Start Time: 8 AM

Latest Possible Date: October 4, 1998

Latest Start Time: 6 PM

Earliest End Date:

Latest End Date:

Duration: hr 1 min

Recurring? ☐ Interval: weekly

Category:

Security: confidential

Location: 14-210

Priority: must

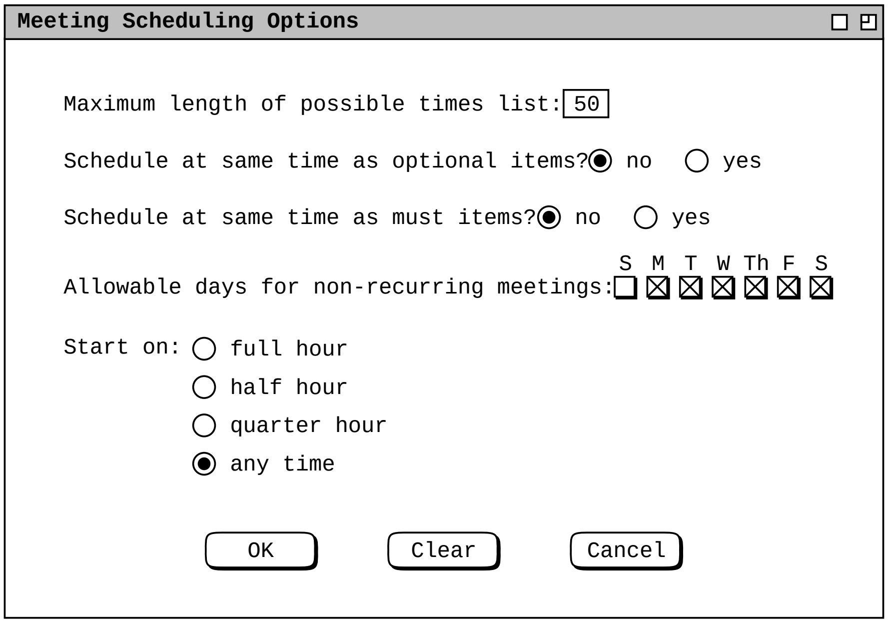
Remind? ☒ 24 hours before email

Attendees:
gfisher, fkurfess, lbrady, jlbrando, erogers, dgoldman

Details:

Minutes:

Figure 106: Modified tenure review meeting request.



Maximum length of possible times list: 50

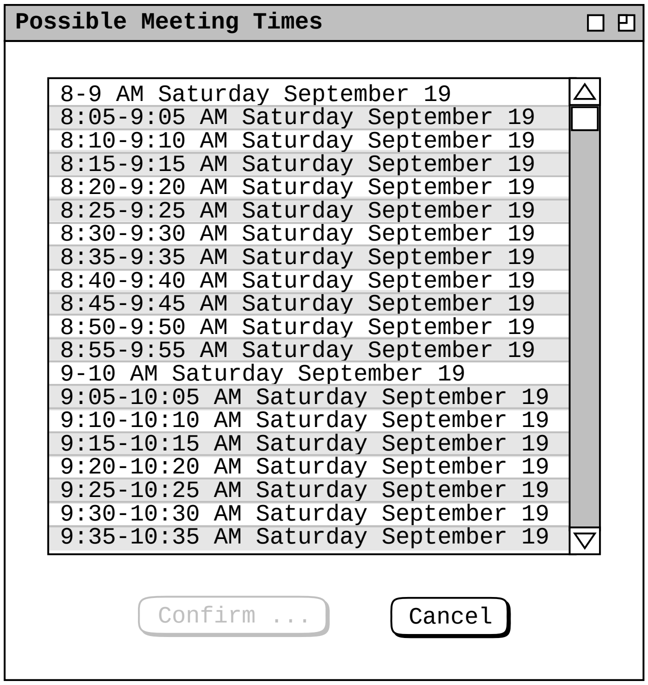
Schedule at same time as optional items? ● no ○ yes

Schedule at same time as must items? ● no ○ yes

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

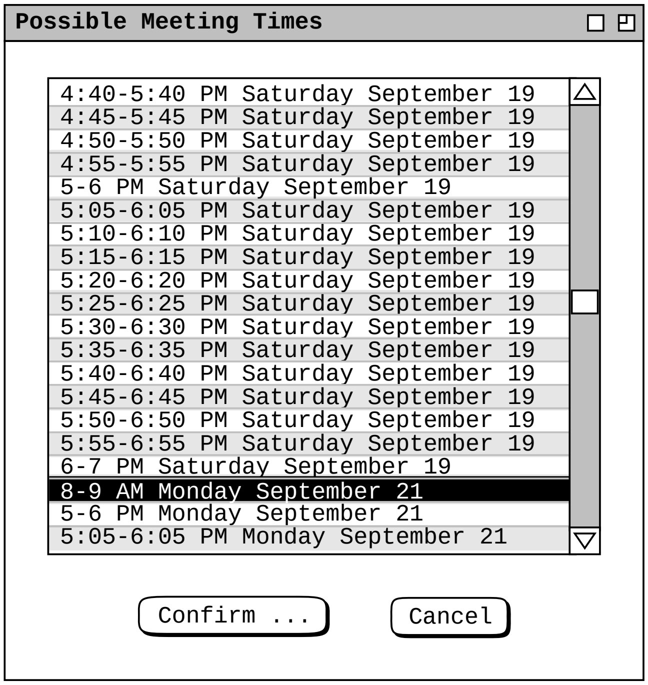
Start on:
○ full hour
○ half hour
○ quarter hour
● any time

Figure 107: Tenure review meeting options.



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.



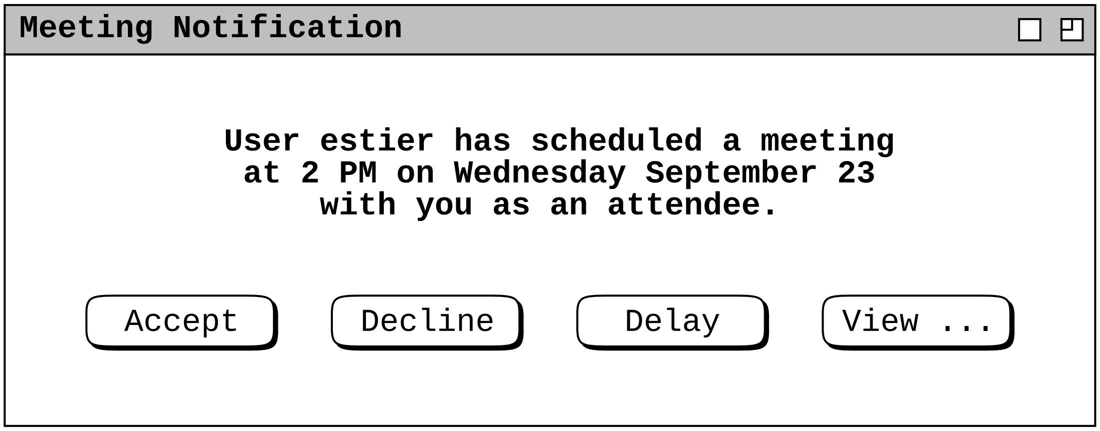
4:40-5:40 PM Saturday September 19
4:45-5:45 PM Saturday September 19
4:50-5:50 PM Saturday September 19
4:55-5:55 PM Saturday September 19
5-6 PM Saturday September 19
5:05-6:05 PM Saturday September 19
5:10-6:10 PM Saturday September 19
5:15-6:15 PM Saturday September 19
5:20-6:20 PM Saturday September 19
5:25-6:25 PM Saturday September 19
5:30-6:30 PM Saturday September 19
5:35-6:35 PM Saturday September 19
5:40-6:40 PM Saturday September 19
5:45-6:45 PM Saturday September 19
5:50-6:50 PM Saturday September 19
5:55-6:55 PM Saturday September 19
6-7 PM Saturday September 19
8-9 AM Monday September 21
5-6 PM Monday September 21
5:05-6:05 PM Monday September 21

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

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.

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 a notification dialog for the non-recurring staff meeting as scheduled in



User estier has scheduled a meeting
at 2 PM on Wednesday September 23
with you as an attendee.

Figure 110: Meeting notification dialog.

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.

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 2.5.2.2.5

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`', '`Decline`', and '`Delay`' 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. If the user chooses '`Delay`', the system removes the notification dialog from the screen and the meeting remains in a not-yet-accepted until the user decides at a later time. If the user has edited any data in the notification, pressing '`Delay`' clears those edits. Edits to a not-yet-accepted meeting are only retained if the user accepts the meeting.

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 local calendar.

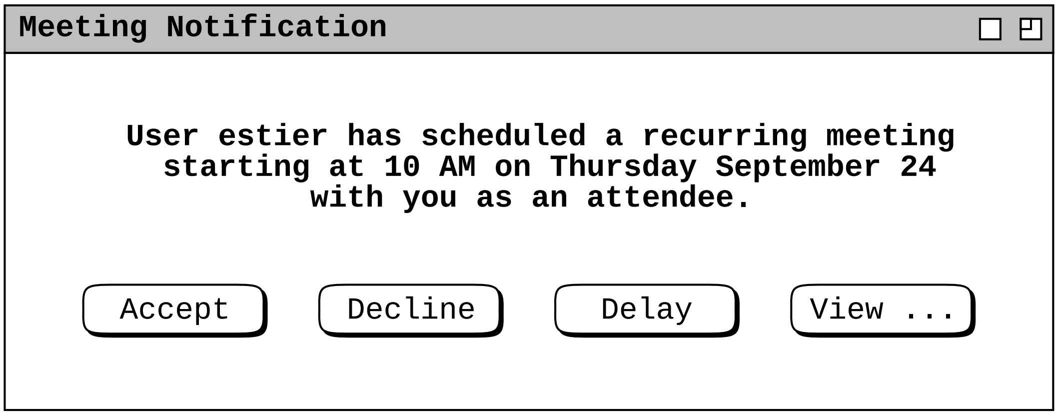
b. If the user is currently connected to the originating central, the system removes the not-yet-accepted marking from the meeting in the user's central host calendar; if not connected, the system will remove the not-yet-accepted mark on the central host the next time the user connects.

c. The system removes the notification dialog from the screen.

The precise effects of declining a meeting are:

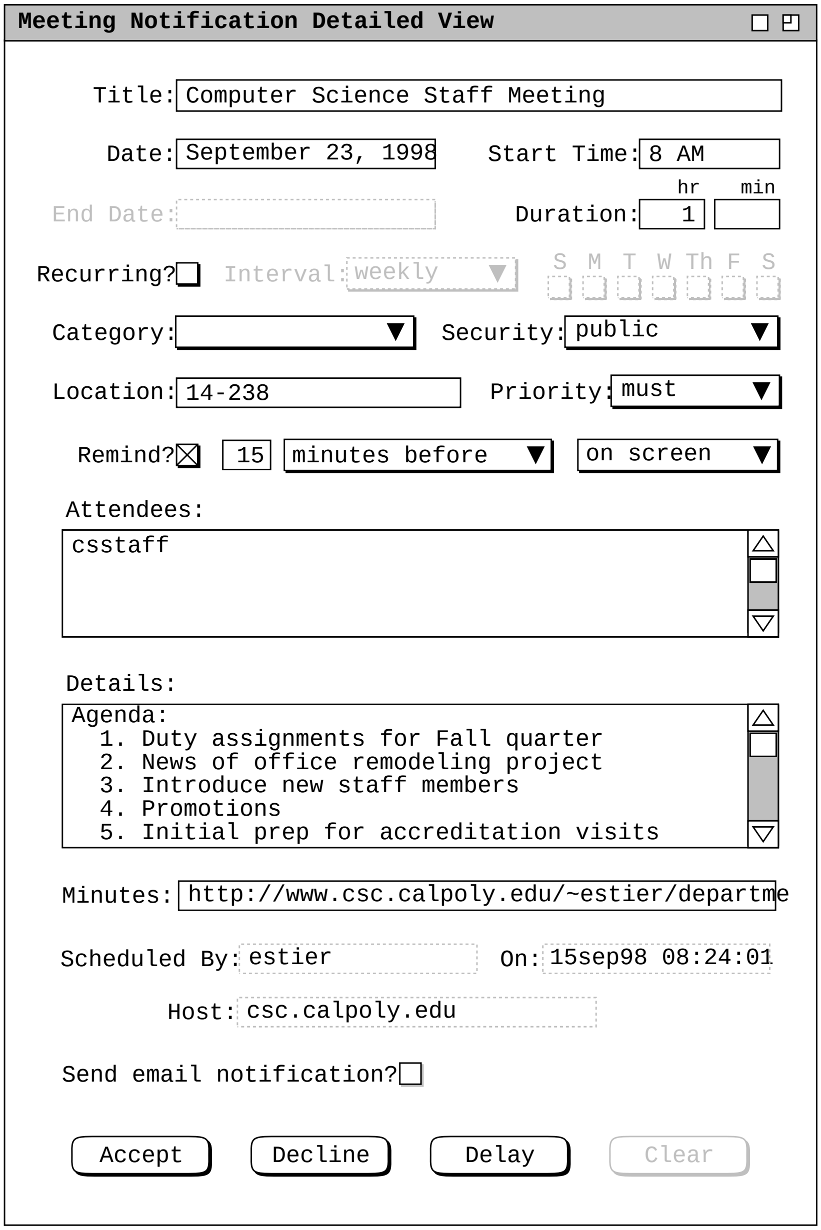
a. The system deletes the not-yet-accepted meeting from the user's local calendar.

b. If the user is currently connected to the originating central, the system removes the not-yet-accepted meeting in the user's central host calendar; if not connected, the system will remove the meeting on the central host



User estier has scheduled a recurring meeting starting at 10 AM on Thursday September 24 with you as an attendee.

Figure 111: Recurring meeting notification dialog.



Meeting Notification Detailed View

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? ☐

Figure 112: Viewing details of a meeting notification.

the next time the user connects.

c. The system removes the notification dialog from the screen.

The precise effects of delaying the accept/decline decision are the following:

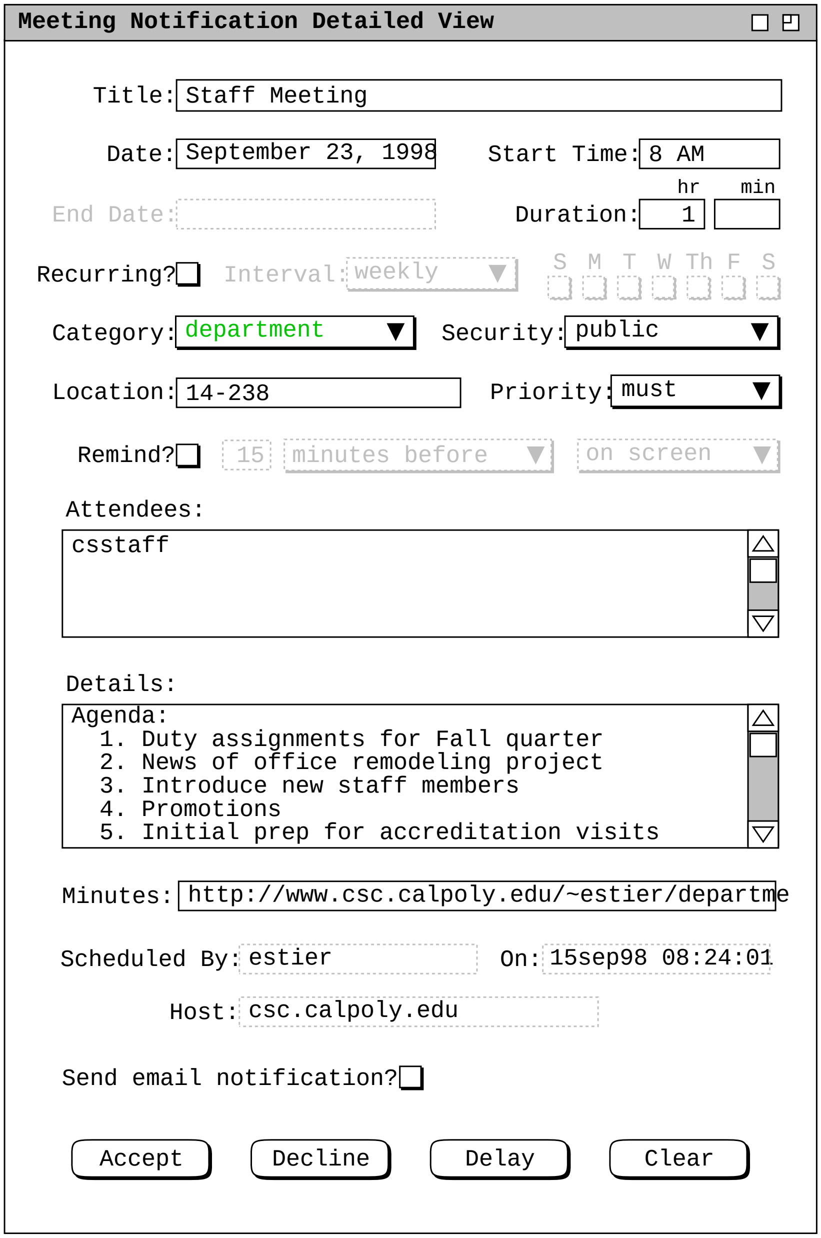
a. The meeting remains marked as not-yet-accepted on both the local and central host calendars.

b. The system removes the notification dialog from the screen.

The system displays a meeting notification dialog at most once for each meeting. Once the notification dialog is removed from the screen, it will not appear again. It is possible for a notification dialog never to appear, based on the notification option settings described in Section 2.4.1.5.

If the user closes the notification dialog without selecting '`Accept`', '`Decline`', or '`Delay`', the effect is as if '`Delay`' were selected. The 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



Title: Staff Meeting

Date: September 23, 1998
Start Time: 8 AM

End Date:
Duration: hr 1 min

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

Category: department
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? ☐

Figure 113: Notification details dialog edited.

displayed, the user can press ‘`Clear`’. As illustrated in Figures 112 and 113 the ‘`Clear`’ button is disabled until the user performs some editing.

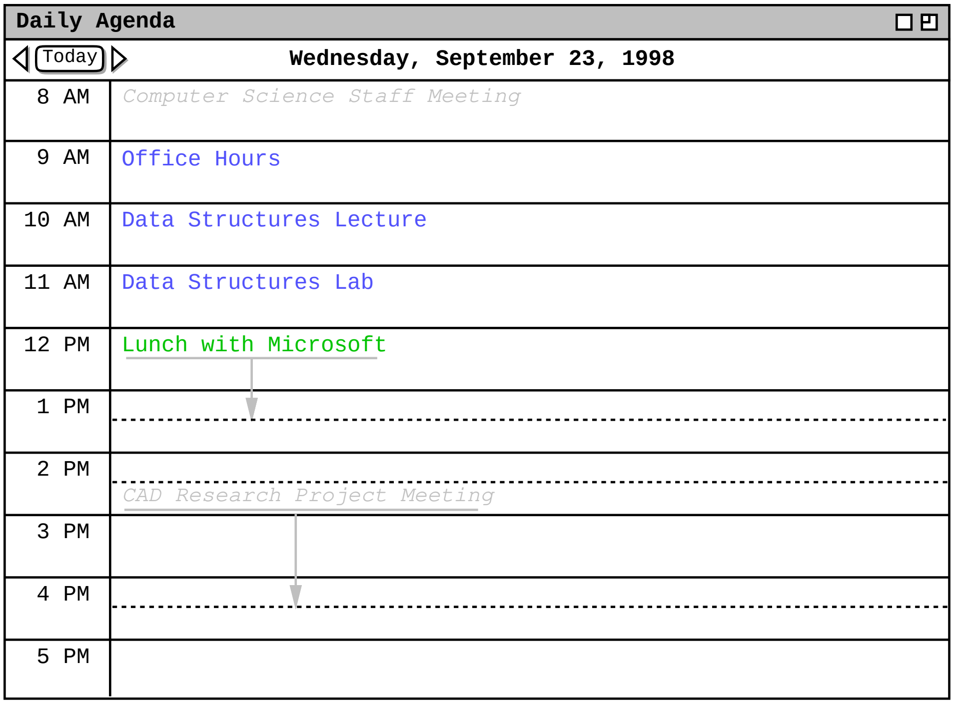
When the detailed notification dialog is displayed, the initial smaller notification dialog remains active. The user can press ‘`Accept`’, ‘`Decline`’, or ‘`Delay`’ in either dialog. When the user does so, the active dialog is removed from the screen, and the other dialog, if displayed, is also removed.

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

and 2.5.2.2.5.

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.

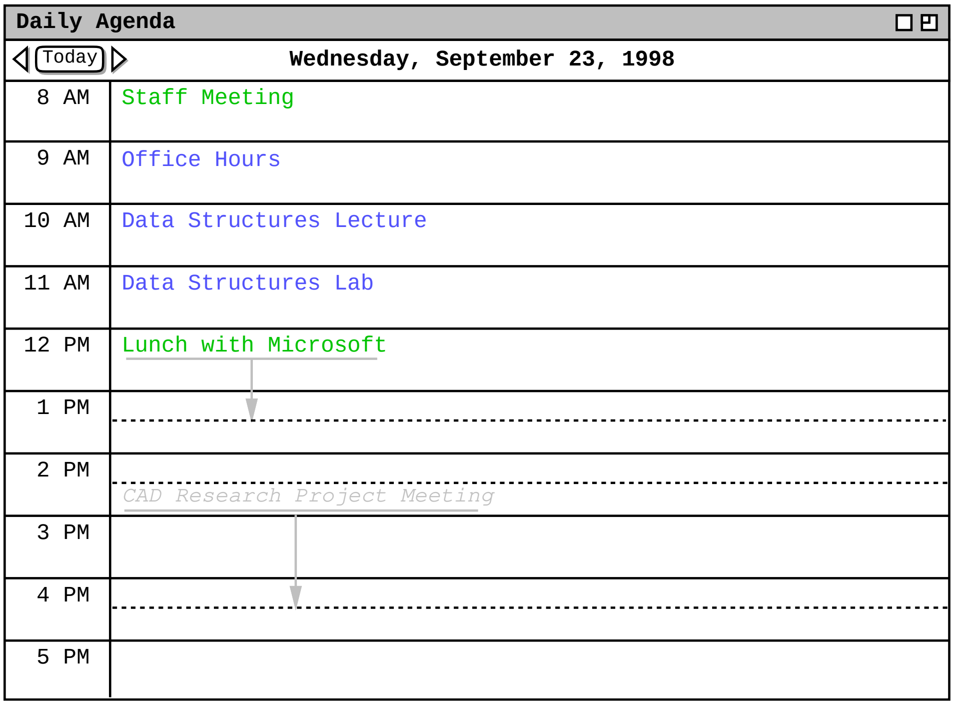
In item-level calendar views, not-yet-accepted meetings are indicated by different command buttons at the bottom of the display. Details of this are presented in Section 2.5.1.2. accurate any more.



Wednesday, September 23, 1998

8 AM	*Computer Science 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 114: Two meetings displayed in penciled-in form.



Daily Agenda

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 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 not-yet-accepted meetings and notification dialogs are controlled by two factors:

a. whether or not the user is connected to the Calendar Tool central host computer on which the meeting is scheduled

b. the settings of meeting-scheduling options that control the acceptance of penciling in and meeting notifications

The necessary enabling condition for local receipt of penciled-in meetings and their notifications is an active connection to the Calendar Tool central host computer on which the meeting is scheduled. Details of establishing a central host connections are covered in Section 2.6.6.1.

The option settings for when to accept penciled-in meetings, that is meetings scheduled by an outside party, are the following:

o '`whenever connected to the originating host`' -- Meetings are penciled in the local calendar any time the user is connected to the Calendar Tool central host from which a meeting originates; if the user is connected to more than one host, penciled-in meetings are accepted from all of them.

o '`at initial connection to originating host`' -- Penciled-in meetings are copied from the central host when the user first establishes a host connection, but not there after; if the user wants to receive additional meetings, she must disconnect and reconnect to the host, or use the '`Get now`' button, as explained in Section 2.7.3.3 on meeting-scheduling options.

o '`never`' -- Penciled-in meetings are never accepted, on either the central host calendar or the local calendar.

When either of the first two settings is selected, meetings are penciled in to the central host calendar immediately upon confirmation from the scheduler; meetings are then penciled in the local calendar as the setting indicates. When the '`never`' setting is selected, meetings scheduled by other users are automatically declined, without ever appearing on central host or local calendars. The ‘`never`’ setting may not be advisable in all circumstances, in the sense that the user may miss important meetings. However, the use of this setting is left to the user’s discretion.

The option setting for when to show notification dialogs has one of the following values:

- ‘`at the same times pencil-in is accepted`’ -- Notifications appear at the same times penciled-in items are copied to the local calendar, i.e., whenever connected to the host or at initial host connection.
- ‘`never`’ -- Notifications never appear, whether or not pencil-in is accepted.

The dependence of the notification option on the pencil-in option disallows the case where notifications are accepted but pencil-in is not. The converse, however, is possible. I.e., notification acceptance can be ‘`never`’ when pencil-in acceptance is something other than ‘`never`’. 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

The acceptance or non-acceptance of penciled-in meetings is a separate issue from whether they appear on a calendar. Using the filtering functions, the user can control the visibility of penciled-in items, as described in Section 2.3.3.2 and Section 2.3.4.1.15.

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.

The system displays the notification dialog for any given meeting at most one time. The notification dialog appears as soon as the connection state of the user’s local 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 notification dialog is allowed to appear, the system never displays it.

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. If the meeting notification option is sent to ‘`never`’, email notifications are not sent.

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 multiuser aspects of scheduled meetings give rise to operational details that must be carefully considered.

```
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.

2.4.1.6.1. 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.

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.

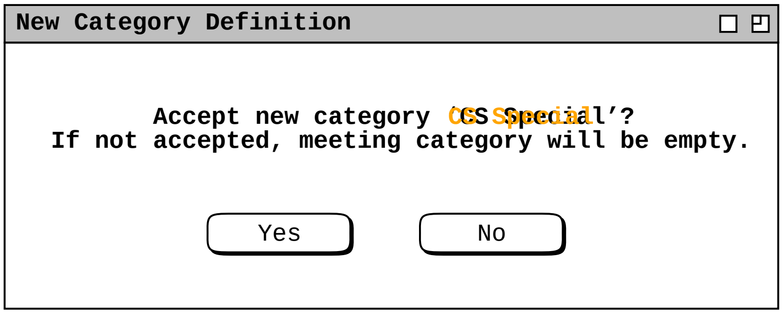
2.4.1.6.3. The Status of Not-Yet-Accepted Meetings

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 filter setting for penciled-in items has no affect on the status of the items for scheduling purposes. Specifically, if a user chooses to filter-out penciled-in items, their times are still considered unavailable for meeting scheduling. Sections 2.3.3.2 and 2.3.4.1.15 describe the ways in which the user can filter out penciled-in meetings.

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.



Accept new category 'CS Special'?
If not accepted, meeting category will be empty.

Figure 117: Prompting dialog for new category definition.

2.4.1.6.5. The Role of the Location Database

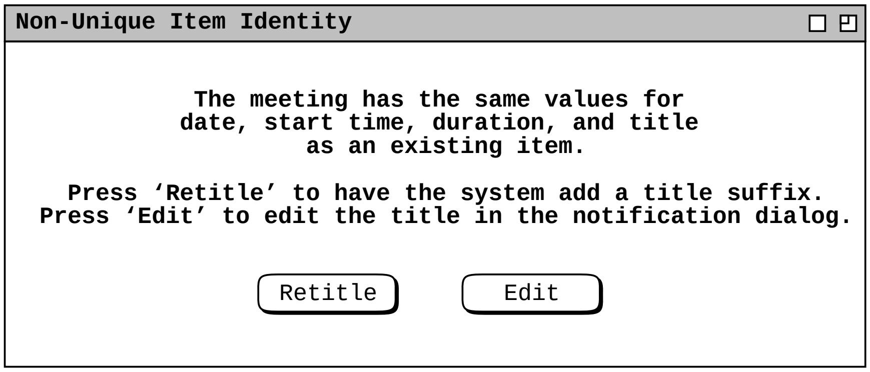
The role of the location database is strictly limited in scheduling a meeting. Specifically, when the scheduler enters a '`Location`' value that is in the location database, the system excludes from the possibles list any times during which that location is unavailable.

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 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 , 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 meetings scheduled 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, and when the same calendar is associated with more than one central host computer.

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



The meeting has the same values for
date, start time, duration, and title
as an existing item.

Press 'Retitle' to have the system add a title suffix.
Press 'Edit' to edit the title in the notification dialog.

Figure 118: Accepted meeting with non-unique identity.

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, but such editing is presumably atypical.

For either '`Retitle`' or '`Edit`' selection, the system removes the non-unique warning dialog from the screen. In the case of 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 redisplays 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'.

2.4.1.6.7. Notifications for Multiple Calendars from Multiple Central Hosts

As described in Section 2.6.6.1, the user can be connected to more than one central host with more than one calendar. The specific rules for calendar/host association are the following:

a. a calendar may be associated with zero or more hosts, at most one of which can be connected at any given time;
b. any single host must be associated with exactly one calendar.

These rules mean that multiple notifications may arrive when the user is actively connected to more than one host. Since the host-to-calendar association must be unique, the 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. makes the calendar associated with host X current, per the calendar currency rules specified in Section 2.3.6.4.
b. 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 Meeting Notifications

Depending on the state of an attendee's connection to a central host 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. In the presumably rare case of two meetings being confirmed at exactly the same time from two different central hosts, the items are ordered in ascending lexical-sort order of host name.

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 , 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. If the user fails to respond to one or more notifications prior to exiting the Calendar Tool, all un-responded notifications are discarded. Section discusses non-modal dialogs in further detail.

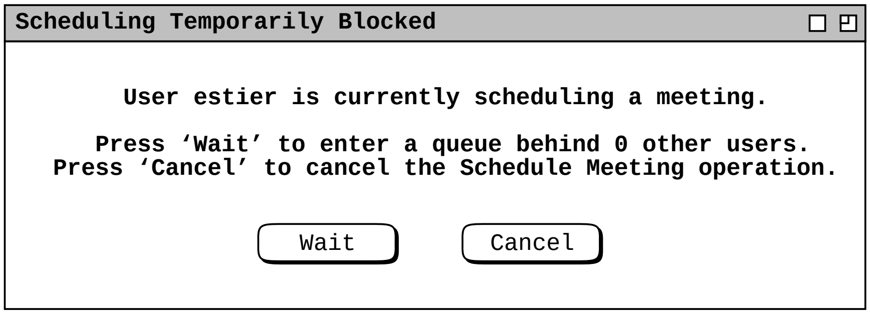
By default, the system removes pending notifications from users' queues after the time for the notified meeting has past. The user can change this default behavior with an option setting.

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 ‘`whenever connected`’, ‘`at initial connection`’, and ‘`never`’ respectively. Notifications for the ‘`whenever connected `’ calendar are queued when the connection to the host is inactive; notifications are displayed as soon as the connection is established and during the time it remains established. For the ‘`at initial connection`’ calendar, notifications are queued up to and immediately after the establishment of the host connection; all pending notifications are displayed immediately after the connection is established. Notifications for the ‘`never`’ calendar are never queued and never displayed.

2.4.1.6.9. Concurrent Scheduling

For a given Calendar Tool central host, 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 on a particular host 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 the 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 Temporarily Blocked

User estier is currently scheduling a meeting.

Press ‘Wait’ to enter a queue behind 0 other users.
Press ‘Cancel’ to cancel the Schedule Meeting operation.

Figure 119: Scheduling temporarily blocked.

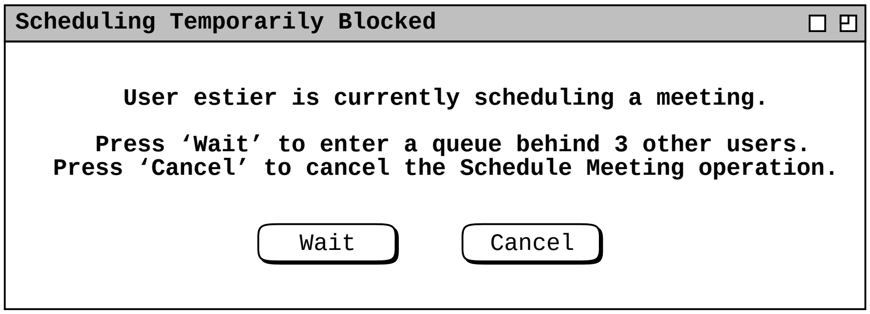
scheduling request dialog.

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 is 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 scheduling 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 ‘`List Times`’ operation may proceed.

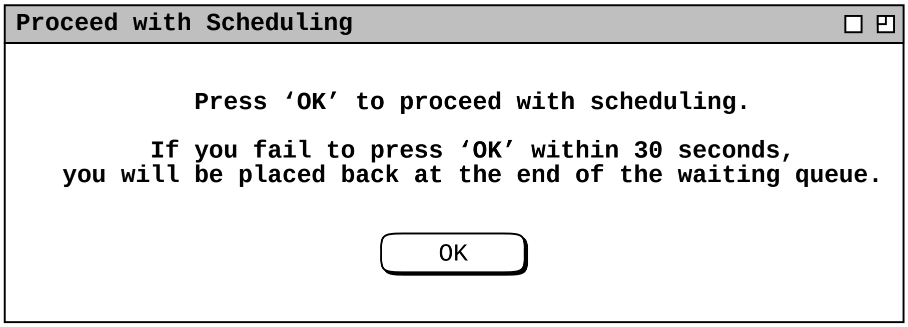
If there are two or more users waiting in the queue, the system displays a dialog of the form shown in Figure 121, accompanied with an audible alert. As the dialog explains, the user has 30 seconds to respond before being put back at the end of the waiting queue. If the user does respond within 30 seconds,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



User estier is currently scheduling a meeting.

Press ‘Wait’ to enter a queue behind 3 other users.
Press ‘Cancel’ to cancel the Schedule Meeting operation.

Figure 120: Fourth blocked user.



Press ‘OK’ to proceed with scheduling.

If you fail to press ‘OK’ within 30 seconds,
you will be placed back at the end of the waiting queue.

Figure 121: Notification to proceed with scheduling.

‘`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.

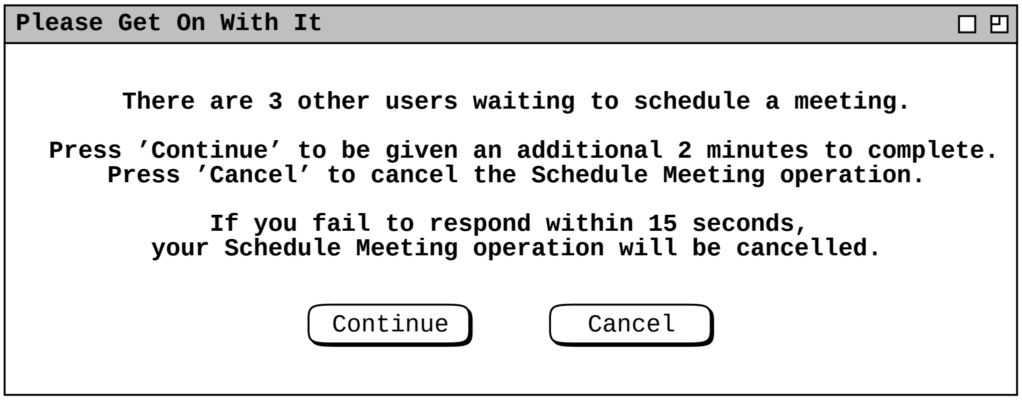
There is no limit to the number of times that a scheduler may respond with ‘`Continue`’ in the time warning dialog. In theory, this allows a user to block indefinitely all other users from scheduling. Such a circumstance is presumably unlikely and must be remedied by means outside of the Calendar Tool system.

The preceding rules for concurrent scheduling apply on a host-specific basis. That is, two meetings cannot be scheduled at the same time on any paricular host. However, concurrent scheduling is not prohibited among different hosts. Further, it is possible for the same user to be connected simultaneously to two different hosts, as described at the end of Section 2.6.6.1.

Given these facts, the host-specific rules for concurrent scheduling are not sufficient to guarantee that the ‘`Scheduled By`’ and ‘`On`’ fields are sufficient as a unique identification for all scheduled meetings. While it is not possible for two meetings to be scheduled at exactly the same time on any single host, it is possible for this to happen on two different hosts. Further, if the same user ID is used to gain simultaneous access to two different hosts, it is possible for two different meetings to have the same values for the ‘`Scheduled By`’ and ‘`On`’ fields. In such cases, the value of the ‘`Host`’ field is used to guarantee unique identification of two different meetings.

The circumstances under which the ‘`Host`’ field is necessary to guarantee meeting uniqueness are presumably quite rare. Given the concurrent scheduling rules described here in Section 2.4.1.6.9, plus the uniqueness rules described in Section 2.4.1.6.6, the ‘`Host`’ field is only necessary to guarantee unique identification of different meetings when all of the following conditions are true:

o the meetings were scheduled by the same user ID at the same time on two different hosts



There are 3 other users waiting to schedule a meeting.

Press ’Continue’ to be given an additional 2 minutes to complete.
Press ’Cancel’ to cancel the Schedule Meeting operation.

If you fail to respond within 15 seconds,
your Schedule Meeting operation will be cancelled.

Figure 122: Scheduling time warning.

- the meetings must have the same values for date, start time, duration, and title
- a local calendar in which the meetings reside must have been connected to the two different hosts on which the meetings were scheduled
- the local user cannot have accepted either of the meetings, i.e., they must both be in the penciled-in state.

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 5:

  - `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.

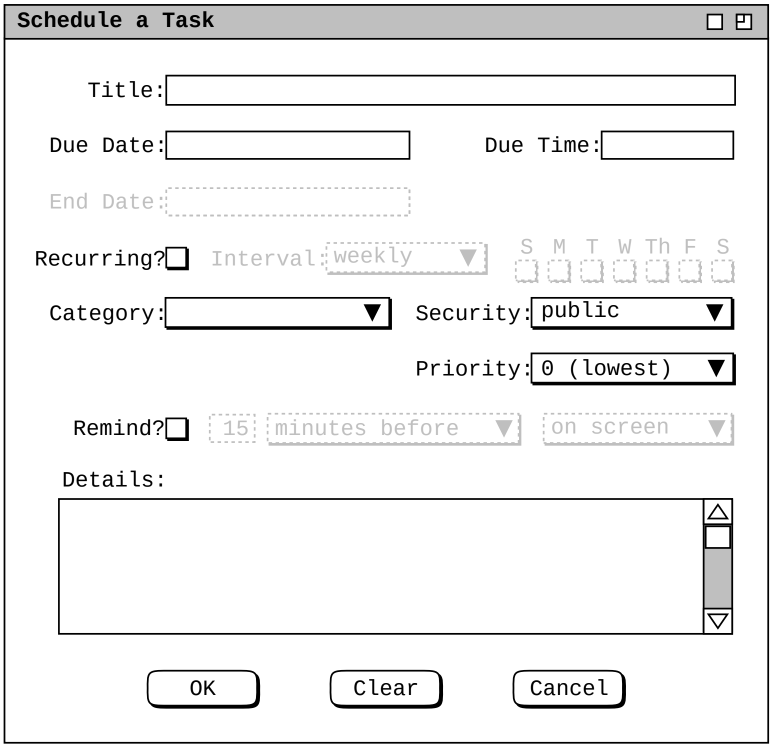

Interval: weekly
Security: public
Priority: 0 (lowest)
Remind? ☐ 15 minutes before on screen

Figure 123: Task scheduling dialog.

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 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`’.

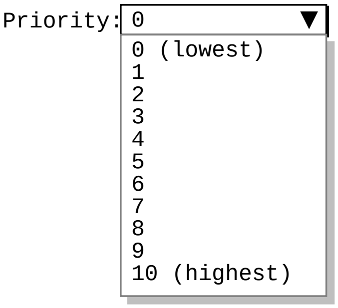
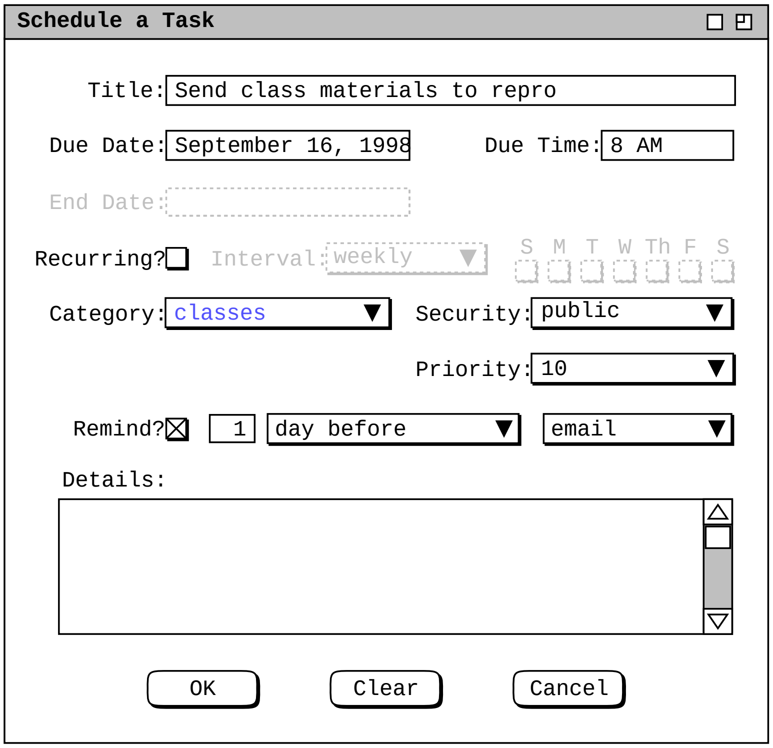

Priority: 0

Figure 124: Task priority menu.


Schedule a Task

Title: Send class materials to repro

Due Date: September 16, 1998

Due Time: 8 AM

End Date:

Recurring? ☐ Interval: weekly

S ☐ M ☐ T ☐ W ☐ Th ☐ F ☐ S ☐

Category: classes

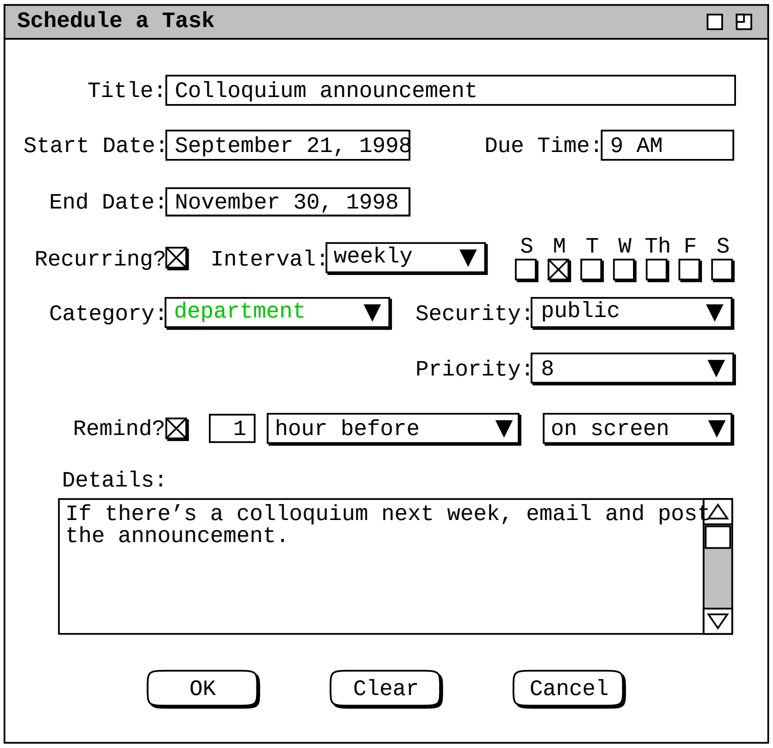
Security: public

Priority: 10

Remind? ☒ 1 day before email

Details:

Figure 125: Scheduling a non-recurring task.



Title: Colloquium announcement

Start 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

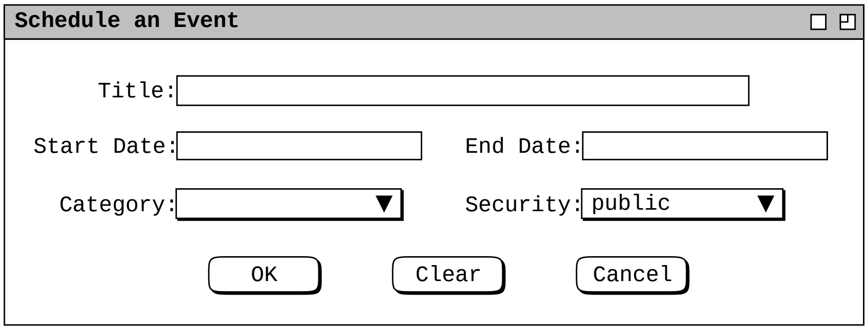
Remind? ☒ 1 hour before on screen

Details:
If there’s a colloquium next week, email and post the announcement.

Figure 126: Scheduling a recurring task.

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 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`’.



Security: public

Figure 127: Event scheduling dialog.

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`’.

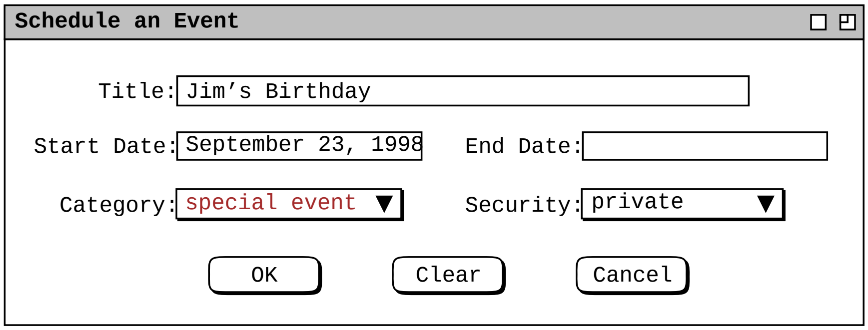
Figures 130 and 131 show how a multi-day event appears in calendar and list views. The figures show how each day of a multi-day event is displayed as a separate item in both calendar and list views. The list view contains the "`Year’s events`" custom list, shown originally in Figure 43. Here the "`Ski vacation`" event has been added and the list is scrolled down to show all its occurrences.

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 .

2.5. Finer Points of Viewing and Scheduling

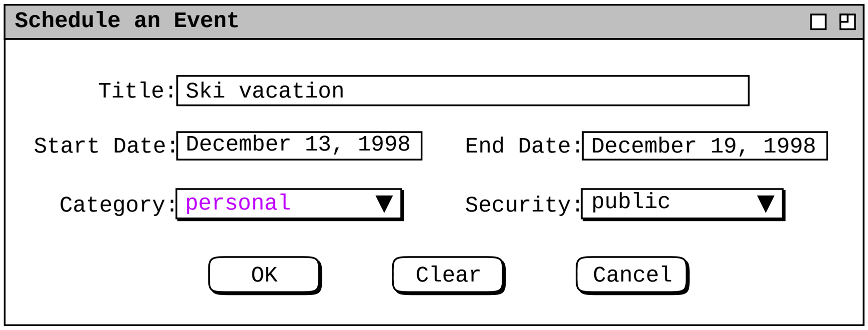
This section of functional requirements covers the remaining details of viewing and scheduling calendar items. The topics covered are:

o viewing at the individual item level
o changing and deleting scheduled items
o details of recurring items
o scheduling and receiving reminders
o editing categories



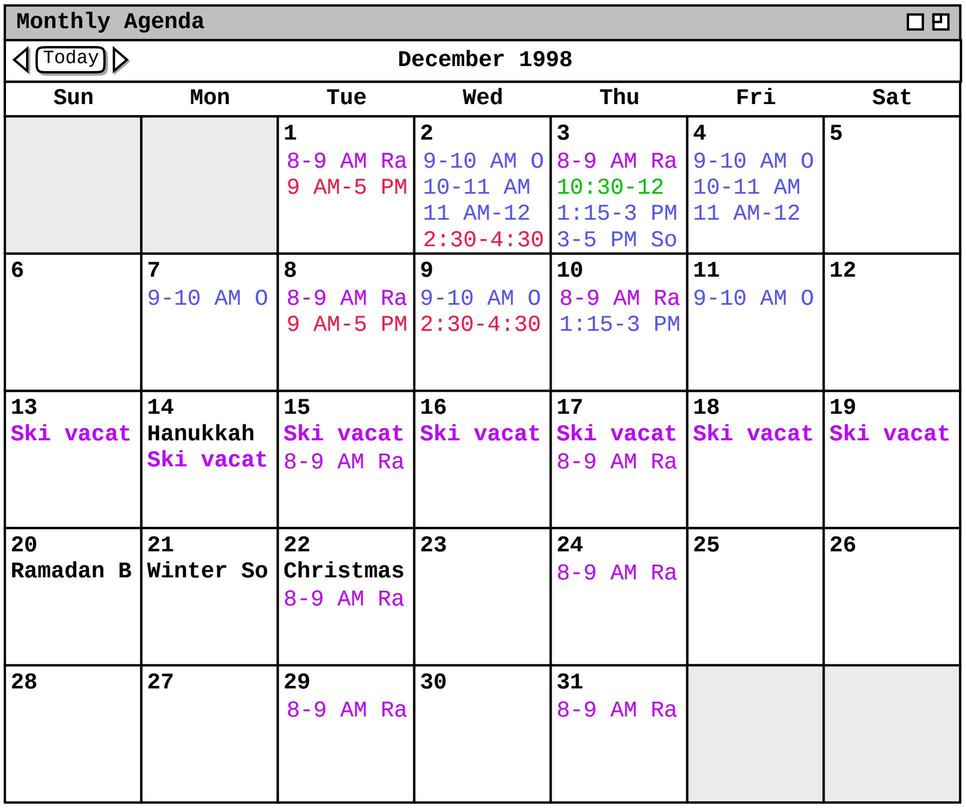
Title: Jim’s Birthday
Start Date: September 23, 1998
End Date:
Category: special event
Security: private

Figure 128: Jim‘s birthday event.



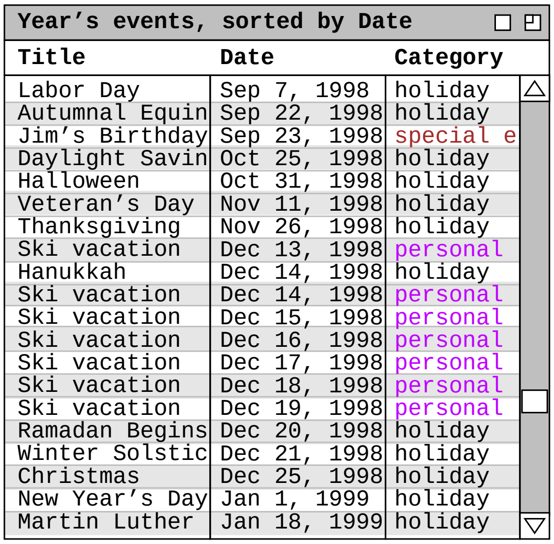
Title: Ski vacation
Start Date: December 13, 1998
End Date: December 19, 1998
Category: personal
Security: public

Figure 129: Multi-day vacation event.



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.



Title	Date	Category
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
Ski vacation	Dec 13, 1998	personal
Hanukkah	Dec 14, 1998	holiday
Ski vacation	Dec 14, 1998	personal
Ski vacation	Dec 15, 1998	personal
Ski vacation	Dec 16, 1998	personal
Ski vacation	Dec 17, 1998	personal
Ski vacation	Dec 18, 1998	personal
Ski vacation	Dec 19, 1998	personal
Ramadan Begins	Dec 20, 1998	holiday
Winter Solstic	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.

o 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 24 and 25 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 .

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

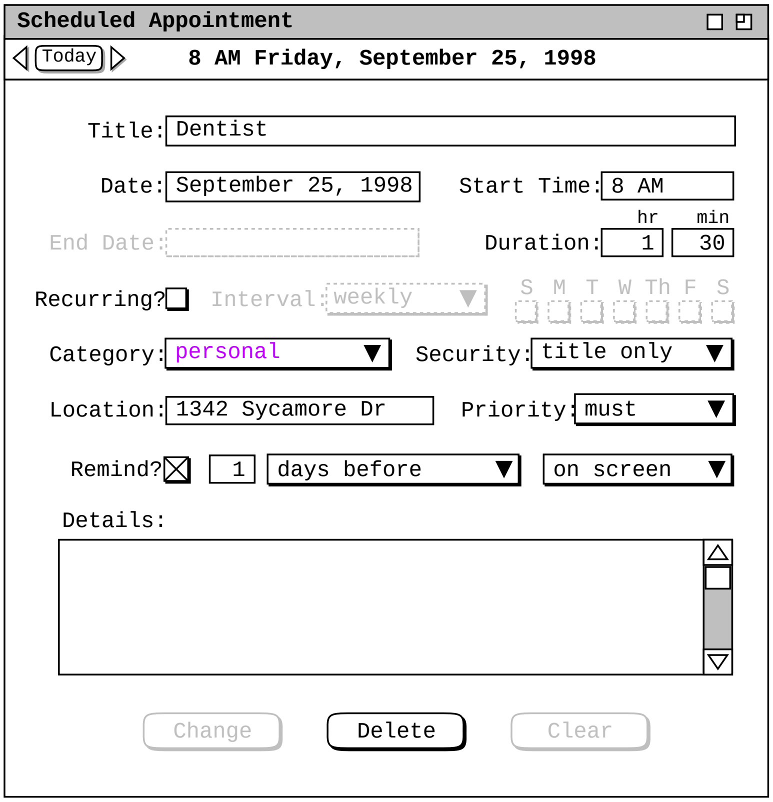
2.5.1.1. Viewing Appointment Items

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 5. 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



8 AM Friday, September 25, 1998

Title: Dentist

Date: September 25, 1998

Start Time: 8 AM

End Date:

Duration: hr 1 min 30

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:

Figure 132: Item-level appointment view.

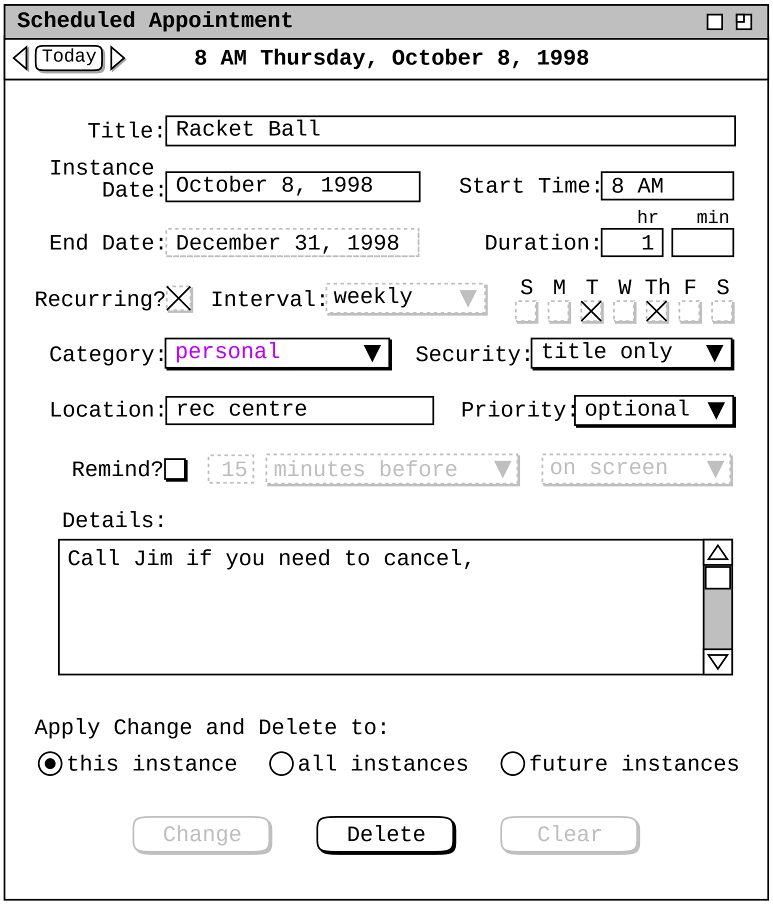
'`OK`', '`Clear`', and '`Cancel`'.

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 12.

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.



Scheduled Appointment

8 AM Thursday, October 8, 1998

Title: Racket Ball

Instance Date: October 8, 1998

Start Time: 8 AM

End Date: December 31, 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:

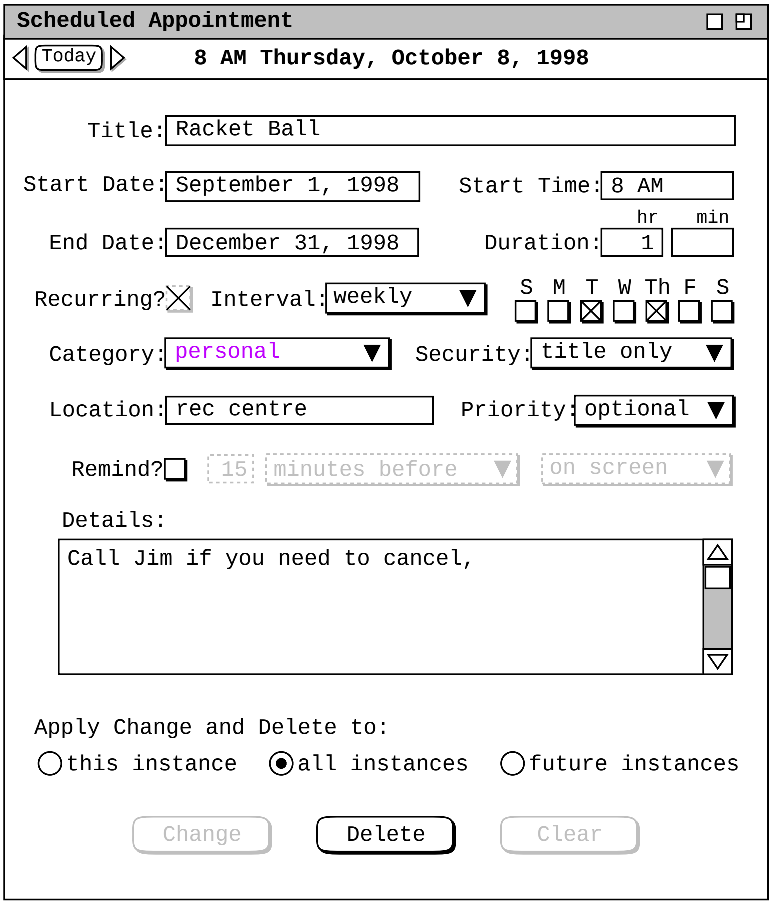
● this instance ○ all instances ○ future instances

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

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 12). The value in the start date field is that entered when the appointment was scheduled, which is September 1 in this case. The ‘`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.



8 AM Thursday, October 8, 1998

Title: Racket Ball

Start Date: September 1, 1998

Start Time: 8 AM

End Date: December 31, 1998

Duration: hr 1 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 ▼

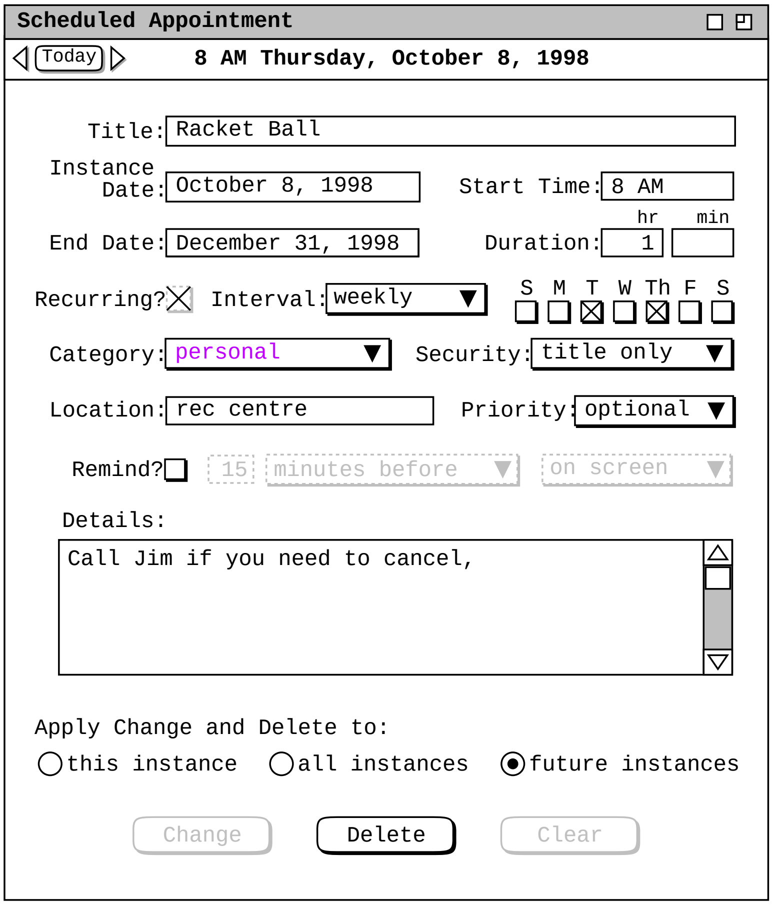
Details:

Call Jim if you need to cancel,

Apply Change and Delete to:

○ this instance ● all instances ○ future instances

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



Scheduled Appointment

8 AM Thursday, October 8, 1998

Title: Racket Ball

Instance Date: October 8, 1998

Start Time: 8 AM

End Date: December 31, 1998

Duration: hr 1 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

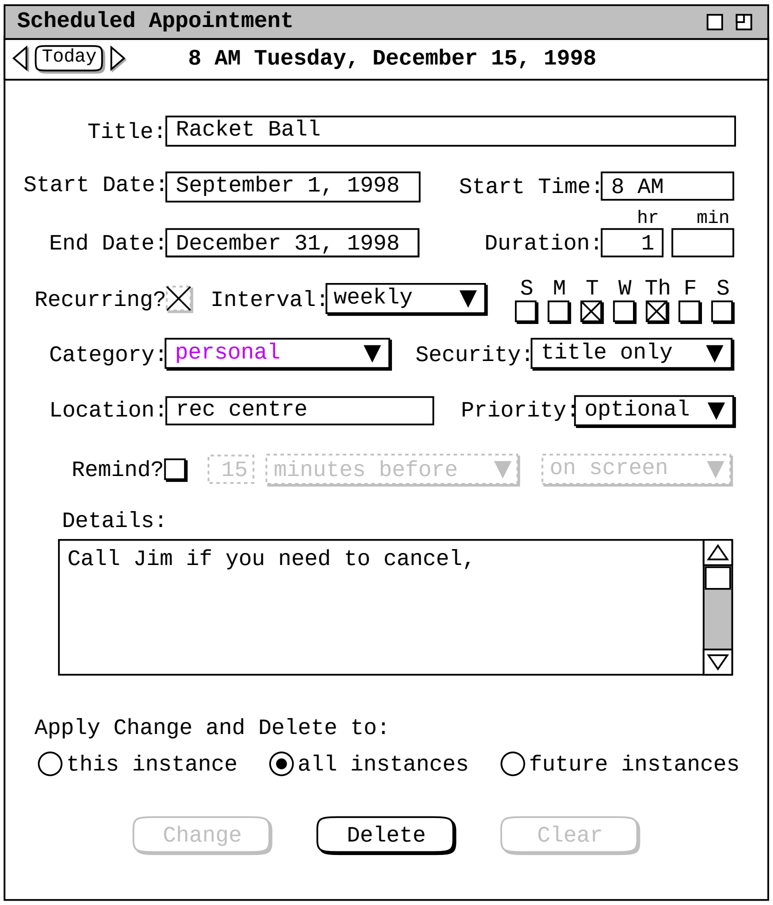
Details:

Call Jim if you need to cancel,

Apply Change and Delete to:

○ this instance ○ all instances ● future instances

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



8 AM Tuesday, December 15, 1998

Title: Racket Ball

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

End Date: December 31, 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

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

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.

2.5.1.2. Viewing Meeting Items

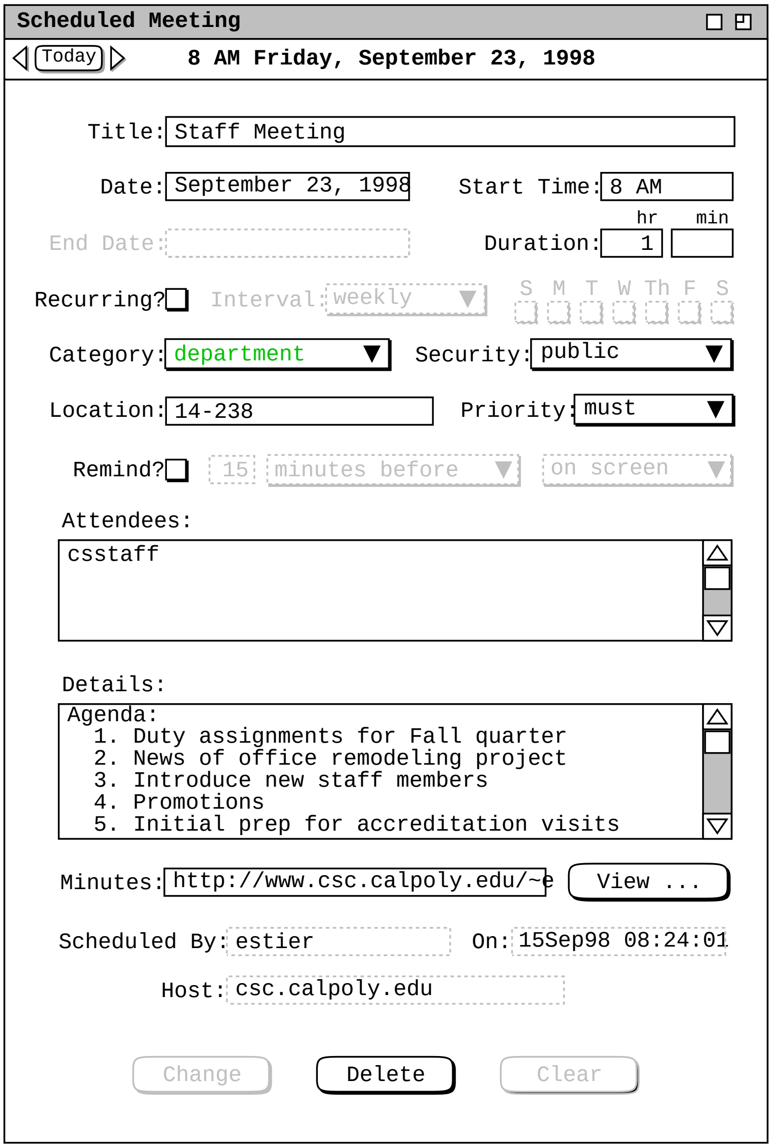
Figure 137 shows the result of the user viewing the item for 8 AM September 23. This is a view of the staff meeting scheduled in the scenario of Section 2.4.1.1. In particular, it is the view of the meeting after it has been accepted by the user in the notification dialog of Figure 113. The meeting view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

a. There is a time and date summary in the viewing display, just below the window banner.
b. There is a ‘`View`’ button to the right the ‘`Minutes`’ field in the viewing display.
c. There is no checkbox for sending an email notification in the viewing display.
d. 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

and 2.5.2.2.5.

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



8 AM Friday, September 23, 1998

Title: Staff Meeting

Date: September 23, 1998

Start Time: 8 AM

End Date:

Duration: hr 1 min

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

Category: department

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

Scheduled By: estier On: 15Sep98 08:24:01

Host: csc.calpoly.edu

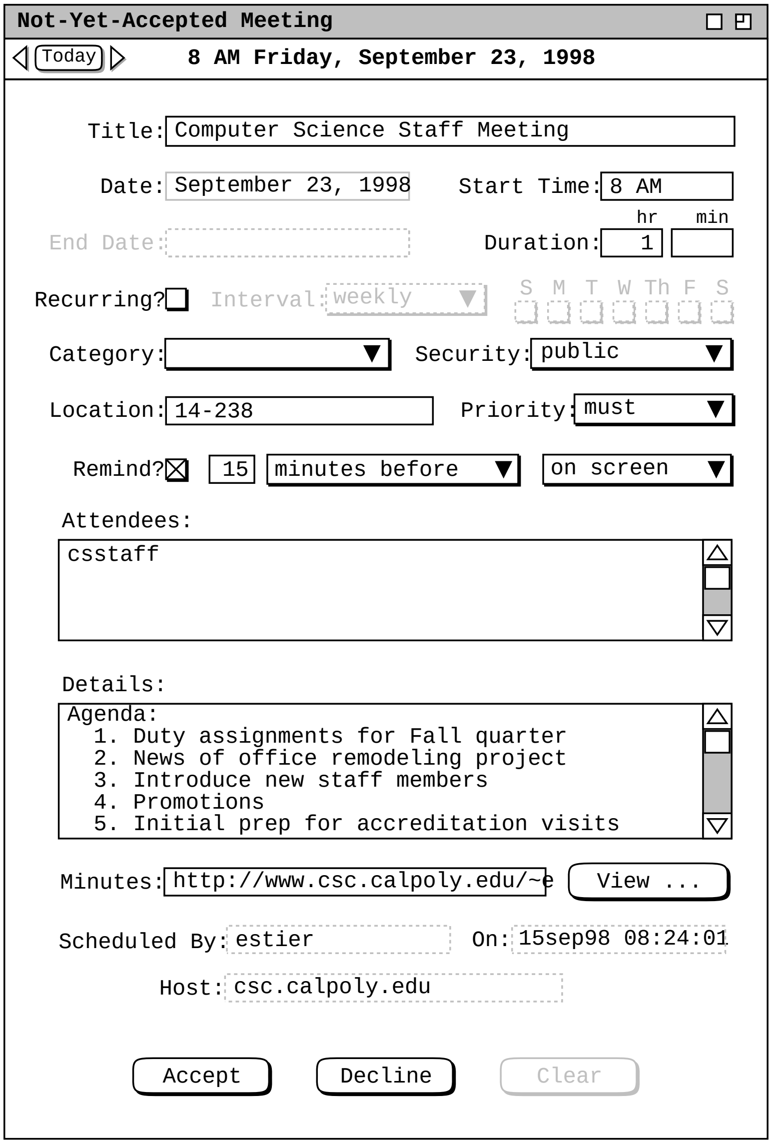
Figure 137: Item-level meeting view.

meeting minutes and the ‘`View`’ button is disabled. When the user presses an enabled ‘`View`’ button, the system displays the minutes using an external web 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 .

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 web browser may signal an error, as discussed in Section . 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



Not-Yet-Accepted Meeting

8 AM Friday, September 23, 1998

Title: Computer Science Staff Meeting

Date: September 23, 1998

Start Time: 8 AM

End Date:

Duration: hr 1 min

Recurring? ☐

Interval: weekly

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/~e

Scheduled By: estier

On: 15sep98 08:24:01

Host: csc.calpoly.edu

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

made in the dialog after it is initially displayed. There is no '`Delay`' button in the item-level view of a meeting, as there is in the meeting notifcation dialog. To effect delay of the accept/decline decision, the user can simply close the item-level view window.

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.

Unless the user hides penciled-in meetings with a filter, 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 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:

a. there is a time and date summary in the item display, just below the window banner;
b. there are two additional data fields in the item display: the '`Completed?`' checkbox and the '`Completion Date`' text field.
c. the command buttons at the bottom of the item display are '`Change`', '`Delete`, and `Clear`' instead of '`OK`', '`Clear`', and `Cancel`';
d. 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 2.5.2.3. The change/delete radio buttons have exactly the same functionality as in the appointment view, including the same effects on the display format.

2.5.1.4. Viewing Event Items

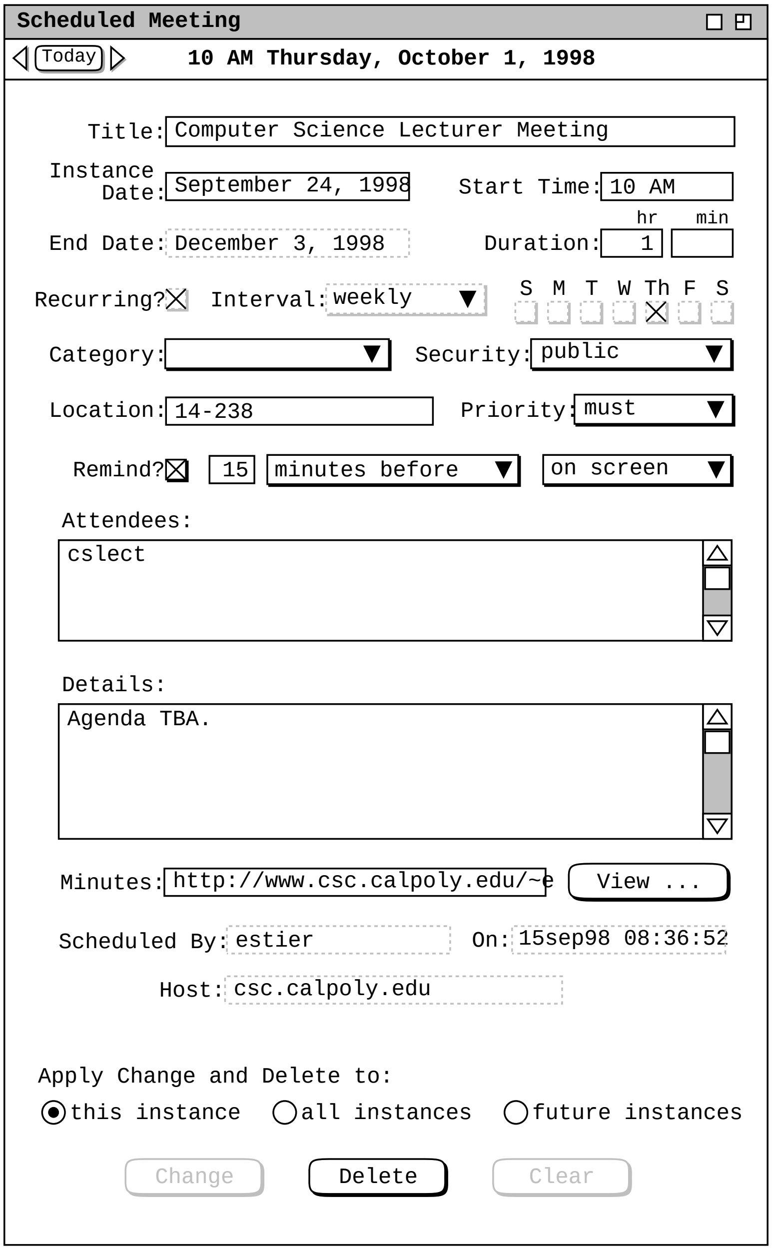
Figure 141 shows the result of the user executing '`View Item`' for the event scheduled in Figure 128. The item-level view of an event contains mostly the same information as the event scheduling dialog (Section 2.4.3), with the following differences:

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 2.5.2.4.

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 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.



10 AM Thursday, October 1, 1998

Title: Computer Science Lecturer Meeting

Instance Date: September 24, 1998

Start Time: 10 AM

End Date: December 3, 1998

Duration: hr 1 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:
cslect

Details:
Agenda TBA.

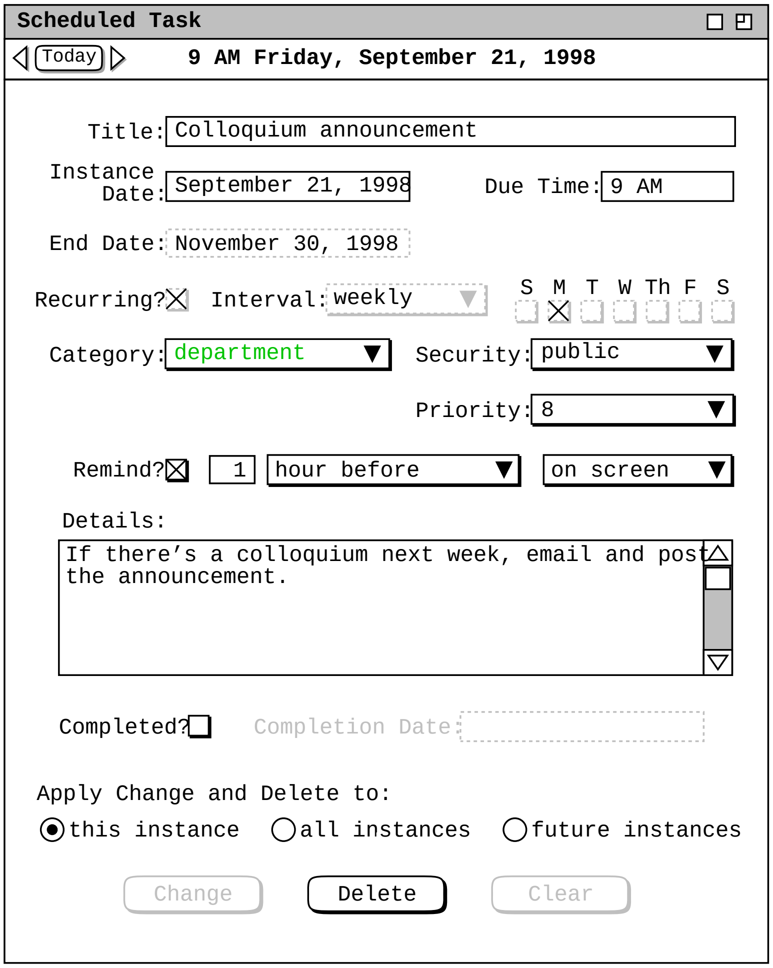
Minutes: http://www.csc.calpoly.edu/~e

Scheduled By: estier On: 15sep98 08:36:52

Host: csc.calpoly.edu

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

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



Scheduled Task

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:

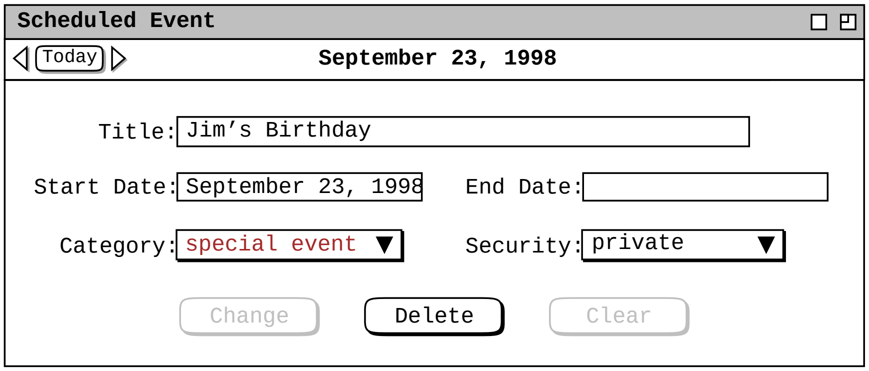
If there’s a colloquium next week, email and post the announcement.

Completed? ☐ Completion Date:

Apply Change and Delete to:

● this instance ○ all instances ○ future instances

Figure 140: Item-level task view.



Scheduled Event

September 23, 1998

Title: Jim’s Birthday

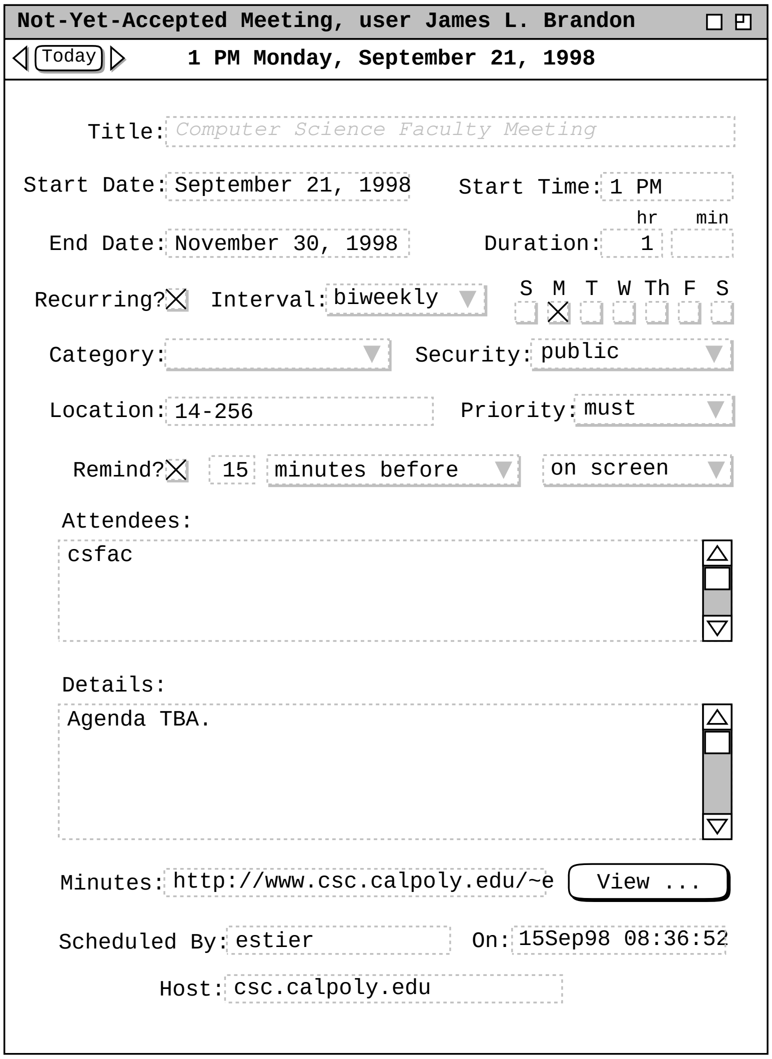
Start Date: September 23, 1998

End Date:

Category: special event

Security: private

Figure 141: Item-level event view.


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

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: hr 1 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
Scheduled By: estier
On: 15Sep98 08:36:52
Host: csc.calpoly.edu

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

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

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 fields.

all edited data fields back to their original values. Executing the ‘`Change`’ command changes the item in the affected calendars and all affected displays. Executing the ‘`Clear`’ command clears all edits, restoring the data fields to their original values when the item-level view was initially displayed, or most 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 fields 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, there is a separate copy of a meeting for the scheduler and for each attendee. 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 scenarios presented in Section 2.5.2.2 address these details.

2.5.2.1. Changing and Deleting Appointments

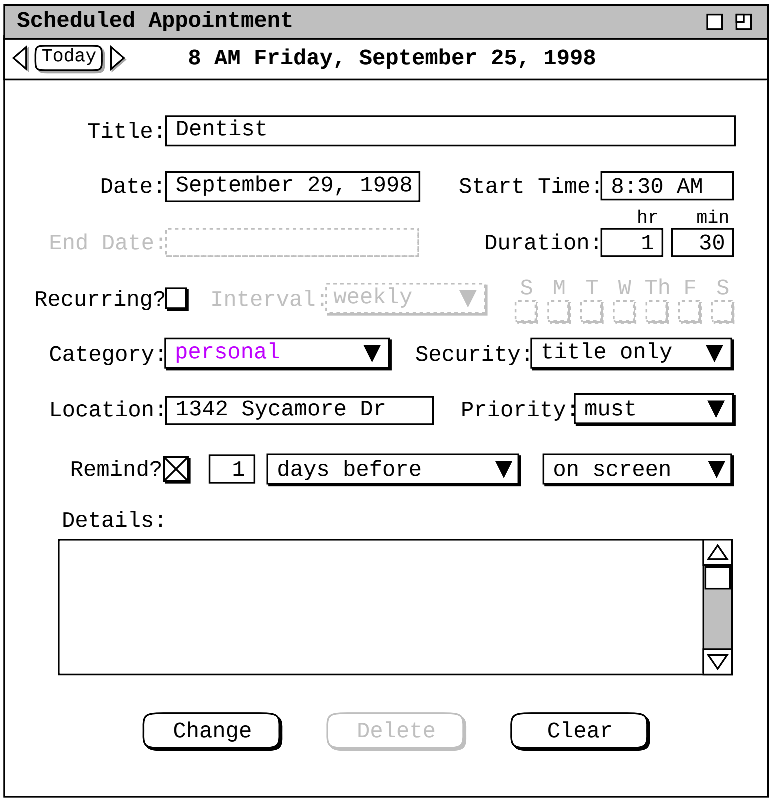
To change a scheduled appointment, the user edits one or more data fields in the item-level display and then presses the ‘`Change`’ button. For example, Figure 145 shows the user having edited the date and time fields of the dentist appointment scheduled in Figure 6. The change to the date is from September 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).

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 21 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, with a few restrictions. The same data-entry rules apply when changing an appointment as when initially scheduling it. These rules are covered in detail in Section . 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;

b. changes made to any of the following data fields in a recurring appointment must apply to all instances of


Scheduled Appointment

8 AM Friday, September 25, 1998

Title: Dentist

Date: September 29, 1998

Start Time: 8:30 AM

End Date:

Duration: hr 1 min 30

Recurring? ☐

Interval: weekly

S M T W Th F S ☐ ☐ ☐ ☐ ☐ ☐ ☐

Category: personal

Security: title only

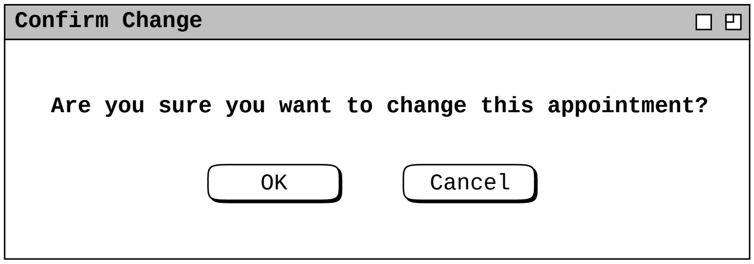
Location: 1342 Sycamore Dr

Priority: must

Remind? ☒ 1 days before on screen

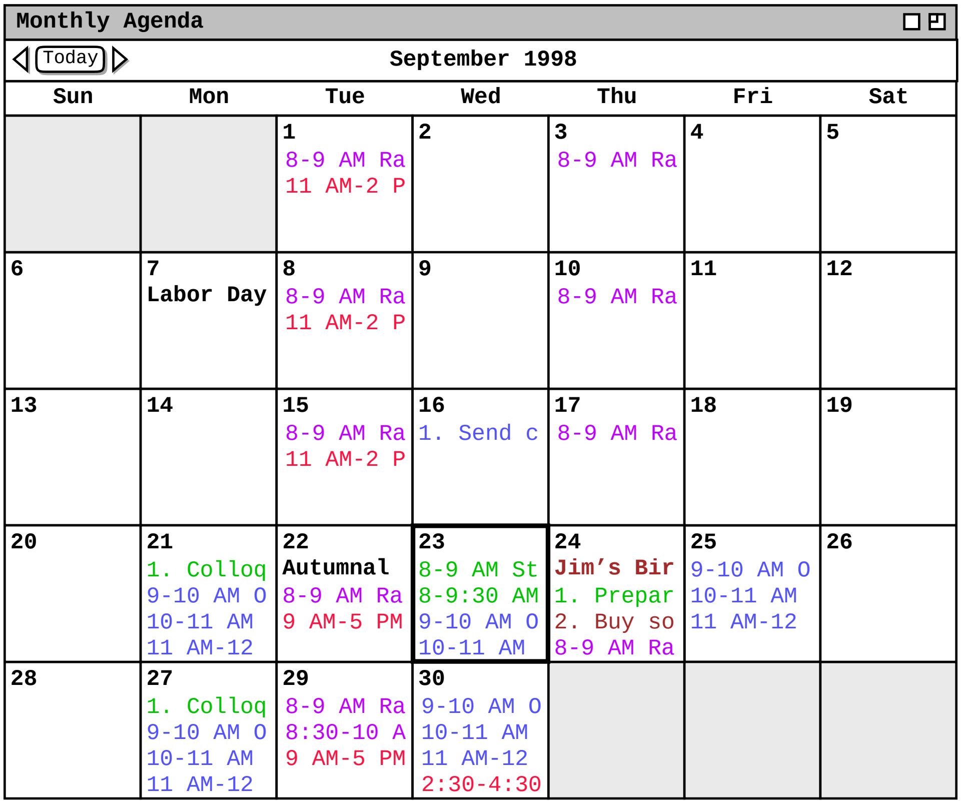
Details:

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


Confirm Change

Are you sure you want to change this appointment?

Figure 146: Change confirmation dialog.



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.

the appointment:

o Start Date (as opposed to Instance Date)

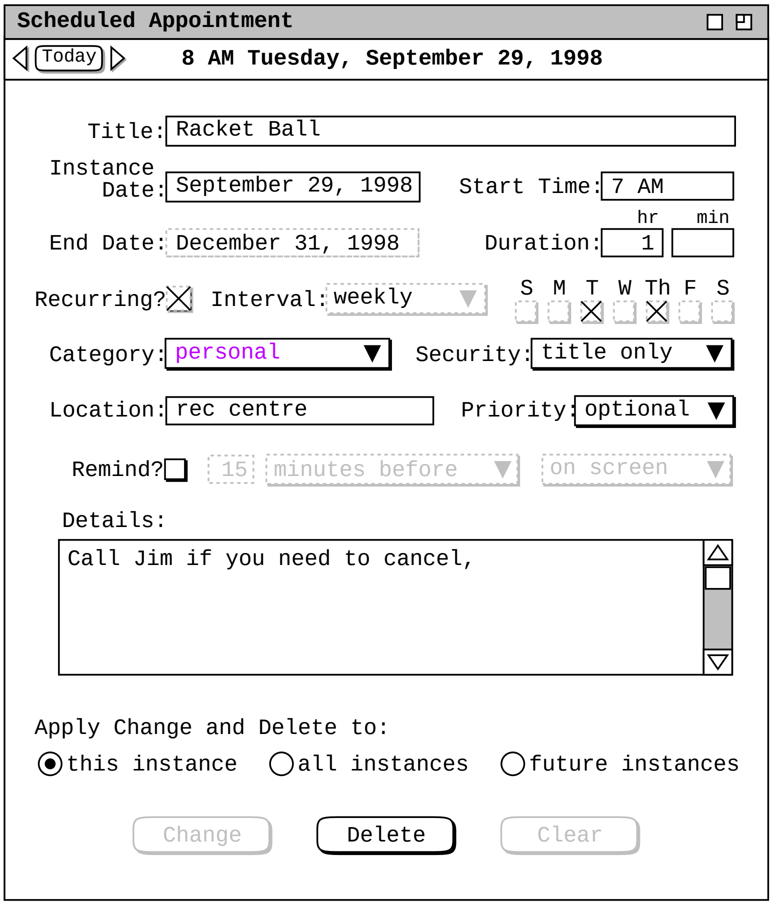
o End Date

o Recurring Interval and the associated recurring details

Figure 148 shows the user having edited an instance of a recurring appointment. The user has selected the ‘`this instance`’ radio button under ‘`Apply Change and Delete to:`’. The system reflects this selection by changing the ‘`Start Date`’ label to ‘`Instance Date`’. Also, the ‘`End Date`’ entry field and all of the ‘`Recurring?`’ entry fields are disabled. These display changes enforce the restrictions noted just above that changes to ‘`Start Date`’, ‘`End Date`’ and ‘`Recurring`’ information must apply to all instances, and therefore cannot be changed when only a single instance is being edited.

In figure Figure 148, 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 selected ‘`all instances` under ‘`Apply Change and Delete to:`’. The system reflects this selection by enabling the ‘`Start Date`’, ‘`End Date`’, and `Recurring`’ edit fields. The `Recurring?` checkbox remains disabled, since a recurring appointment can never be changed to non-recurring, but it’s other recurring information can be changed.



8 AM Tuesday, September 29, 1998

Title: Racket Ball

Instance Date: September 29, 1998

Start Time: 7 AM

End Date: December 31, 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

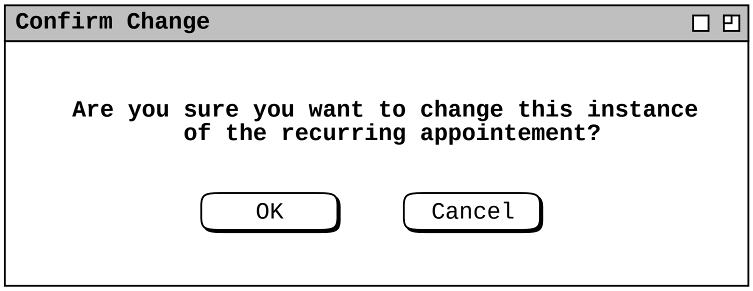
Remind? ☐ 15 minutes before on screen

Details: Call Jim if you need to cancel,

Apply Change and Delete to:

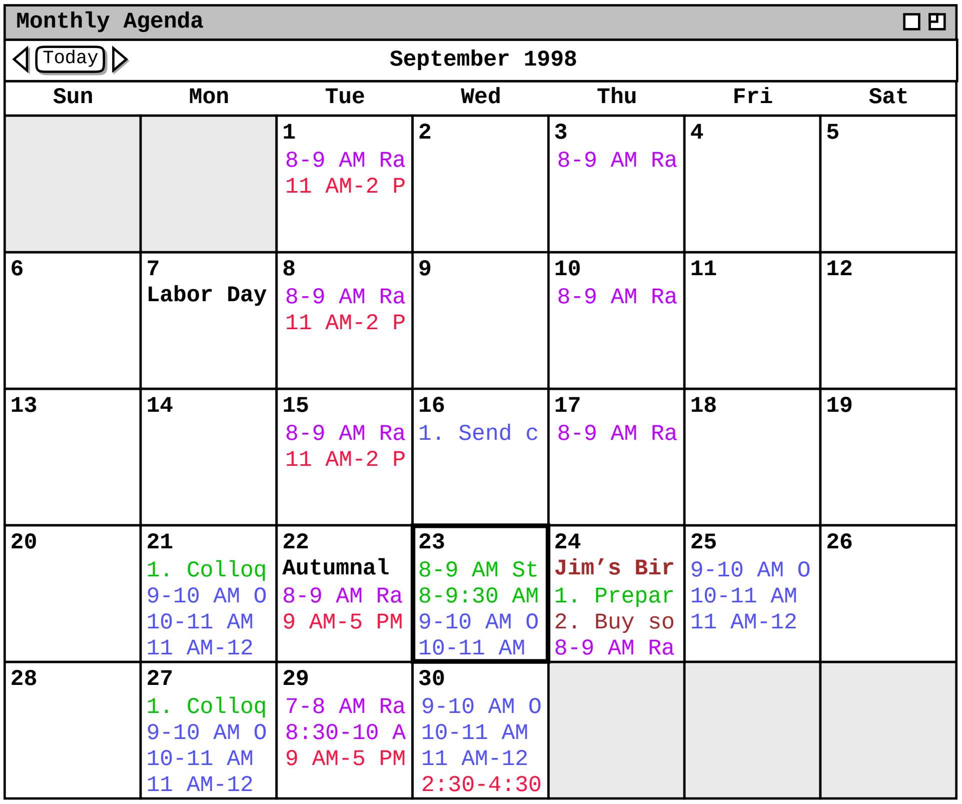
● this instance ○ all instances ○ future instances

Figure 148: Edits made to a recurring appointment.



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

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

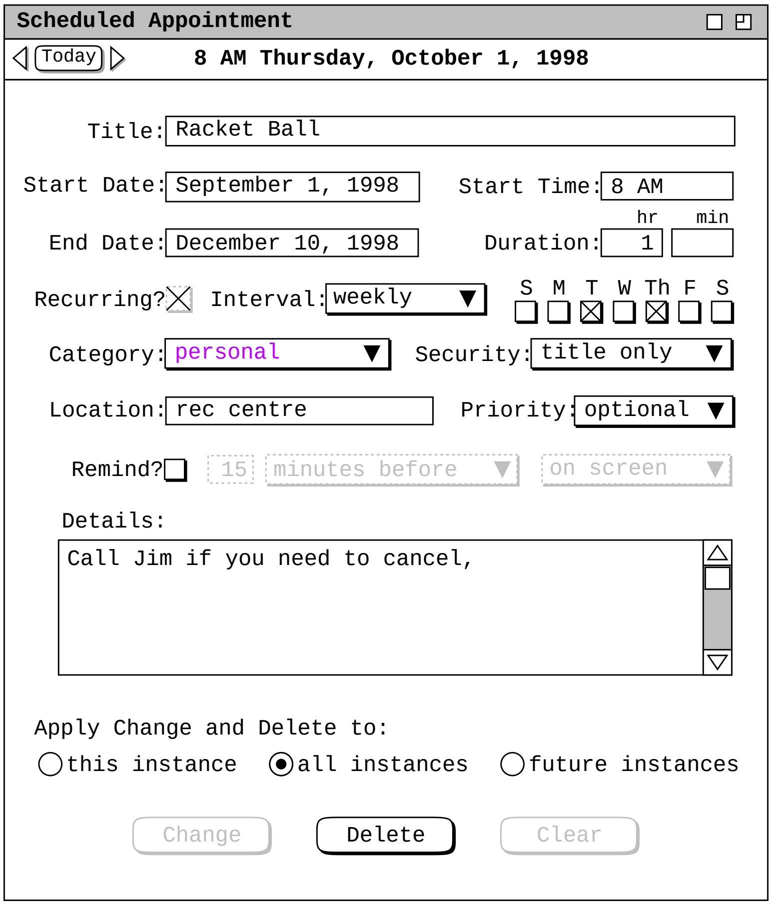


Monthly Agenda

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

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

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

Figure 151: Additional edits made to a recurring appointment.

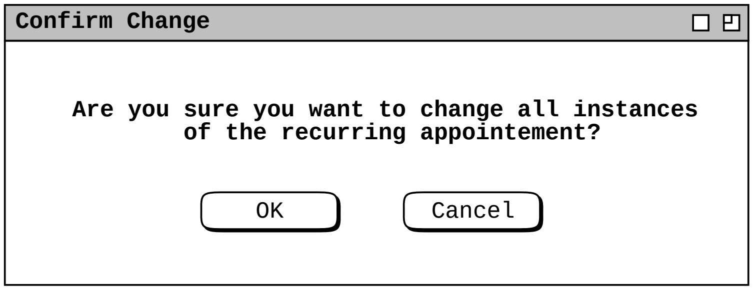
In Figure 151, the user has changed the end date from December 31 to December 10. Of note is that the user moved to an instance other than the edited September 29. If the user had remained at September 29 to make the all-instance change to the end date, then the September 29 start-time change would be 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 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.

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.

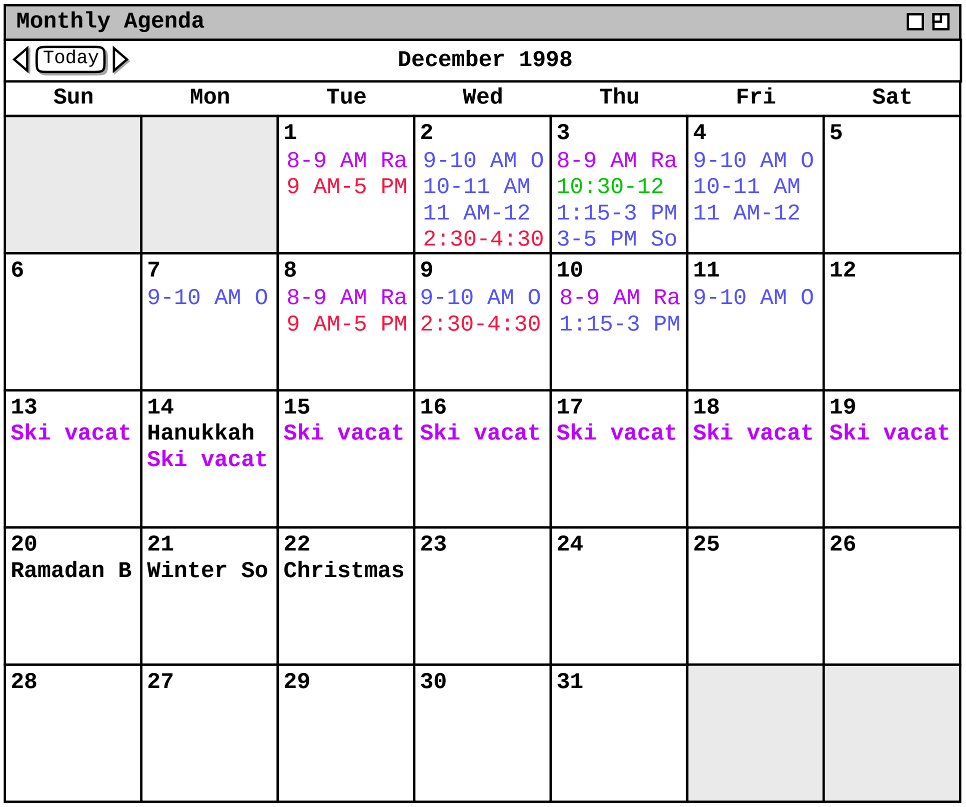
To delete a non-recurring appointment, the user presses '`Delete`' in the item-level display. In response, the system displays the confirmation dialog of Figure 155. The user presses '`OK`' to proceed with the delete, in response to which the system deletes the appointment from the user's calendar and 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.

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`',



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

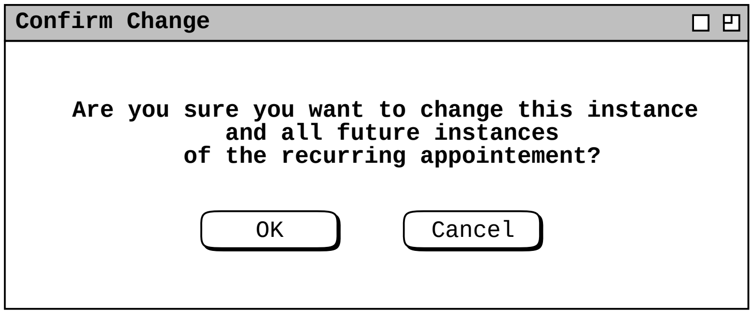
Figure 152: Confirmation dialog for changing all instances of a recurring appointment.



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.

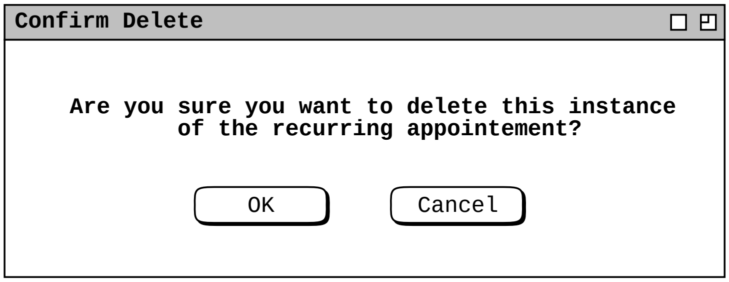


Are you sure you want to change this instance
and all future instances
of the recurring appointement?

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

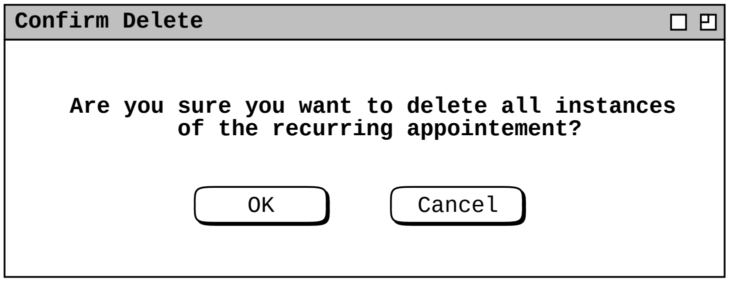
Are you sure you want to delete this appointment?

Figure 155: Delete confirmation dialog.



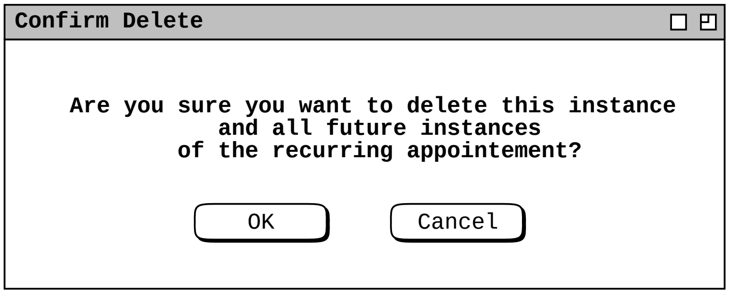
Are you sure you want to delete this instance
of the recurring appointement?

Delete confirmation for one instance of a recurring appointment.



Are you sure you want to delete all instances
of the recurring appointement?

Delete confirmation for all instances of a recurring appointment.



Are you sure you want to delete this instance
and all future instances
of the recurring appointement?

Delete confirmation for all future instances of a recurring appointment.

Figure 156: Delete confirmation dialogs for a recurring appointment.

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.

When the user deletes the last or only instance of a recurring appointment, the appointment is completely removed from the calendar. When the user deletes one or more instances, but others remain, then the recurring item remains and all undeleted instances continue to appear on the calendar. Section 2.5.3.5 discusses in further detail the issues of recurring items and instances.

2.5.2.2. Changing and Deleting Meetings

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

Individual users can also change their own local copies of scheduled meetings. When both schedulers and non-schedulers make changes to the same meeting, their changes must be reconciled. What it means to have separate copies of the "same meeting" is that values of the ‘`Scheduled By`’, ‘`On`’, and ‘`Host`’ fields are the same in each of the copies. The set of 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 precise relationship between the scheduler and non-scheduler copies of a meeting can be characterized as follows. The meeting copies are "synchronized" when they contain exactly the same values for all data fields. The synchronized state exists when an attendee accepts a meeting without making any changes, and the scheduler makes no changes subsequent to confirming the scheduling operation. Synchronization also occurs when a non-scheduler accepts scheduler-initiated changes, as discussed in Section 2.5.2.2.1.

When the scheduler and/or non-scheduler change their copy of a meeting, it becomes "unsynchronized". The only time an unsynchronized meeting is brought to the attention of an attendee is when the scheduler changes the meeting and then notifies the attendees. At this point, the user must be able to identify which data fields have been changed and by whom. To assist in this identification, the system color codes the changes when it displays the change notification dialog. Table 9 describes this coding. In order for change notifications to be sent and received, the scheduler and attendees must be connected to the same Calendar Tool central host computer, and it must be the host from which the changed meeting was originally scheduled. The scenarios of Sections 2.5.2.2.1 through 2.5.2.2.5, assume that the scheduler and affected non-schedulers are so connected, and that the changes they make do not conflict. Sections 2.5.2.2.6 and 2.5.2.2.7 address the cases where scheduler and non-scheduler make changes while disconnected from the host and then subsequently reconnect. Section 2.5.2.2.8 covers conflicting scheduler and non-scheduler changes.

2.5.2.2.1. Scheduler Changing Meetings

To change an existing meeting, the meeting’s scheduler edits its item-level display, in the same manner as appointment editing is performed (see Section 2.5.2.1.). If the scheduler wants all attendees of the meeting to be notified of the changes, the scheduler must be connected to the same central host computer to which she was connected when the meeting was originally scheduled. If the scheduler is not connected to this host at the time a change is confirmed, then the changes are made only to the scheduler’s local copy of the meeting and no change notification is sent (see Section 2.5.2.2.6).

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 10 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

Changed By Scheduler	**Changed By Non-Scheduler**	**Color**
no	no	black
yes	no	blue
no	yes	green
yes	yes	red (a conflict)

Table 9: Color Coding of Meeting Changes in Change Notification Display.

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	any, with warning if conflicts detected	yes
Details	any	no
Minutes	any	no
Scheduled By	none	
On	none	
Host	none	

Table 10: Summary of allowable changes to meeting data fields by the scheduler.

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 2.5.2.2.3

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:

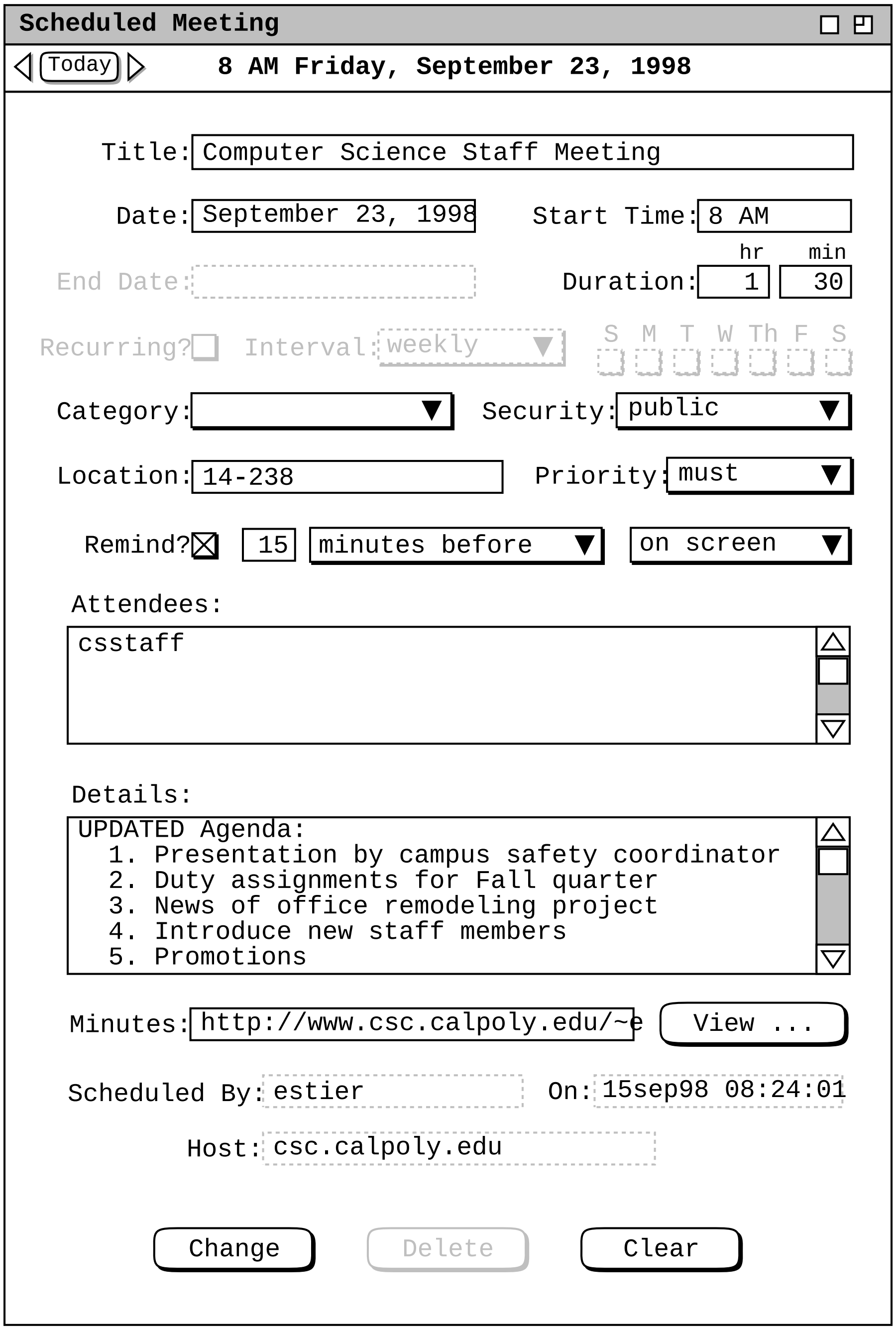
o (Start) Date

o End Date

o Start Time

o Duration

o Recurring



8 AM Friday, September 23, 1998

Title: Computer Science Staff Meeting

Date: September 23, 1998 Start Time: 8 AM

End Date:

Duration: hr 1 min 30

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

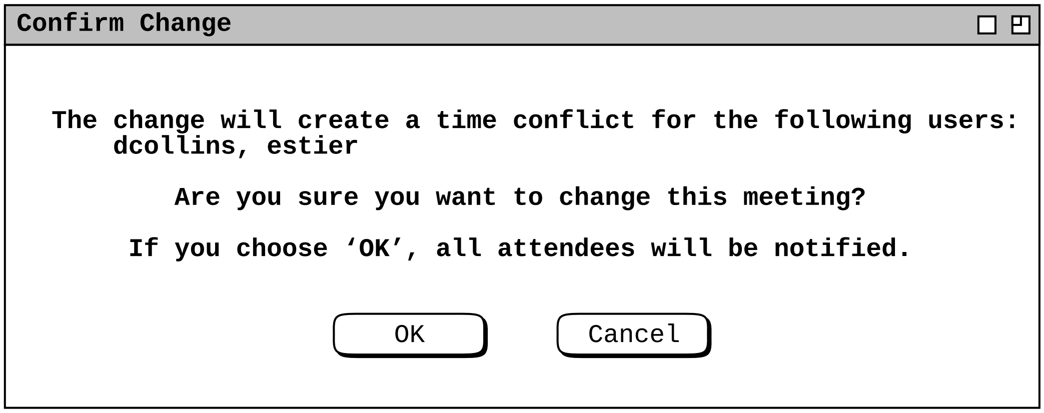
Details:
UPDATED Agenda:
  1. Presentation by campus safety coordinator
  2. Duty assignments for Fall quarter
  3. News of office remodeling project
  4. Introduce new staff members
  5. Promotions

Minutes: http://www.csc.calpoly.edu/~e

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

Host: csc.calpoly.edu

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



The change will create a time conflict for the following users:
dcollins, estier

Are you sure you want to change this meeting?

If you choose ‘OK’, all attendees will be notified.

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

o Attendees

In the case where the `Attendees` list is changed, the users checked for conflict are those in the *changed* list, i.e., all original attendees not deleted from the list, plus any newly-added attendees.

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 example, 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.

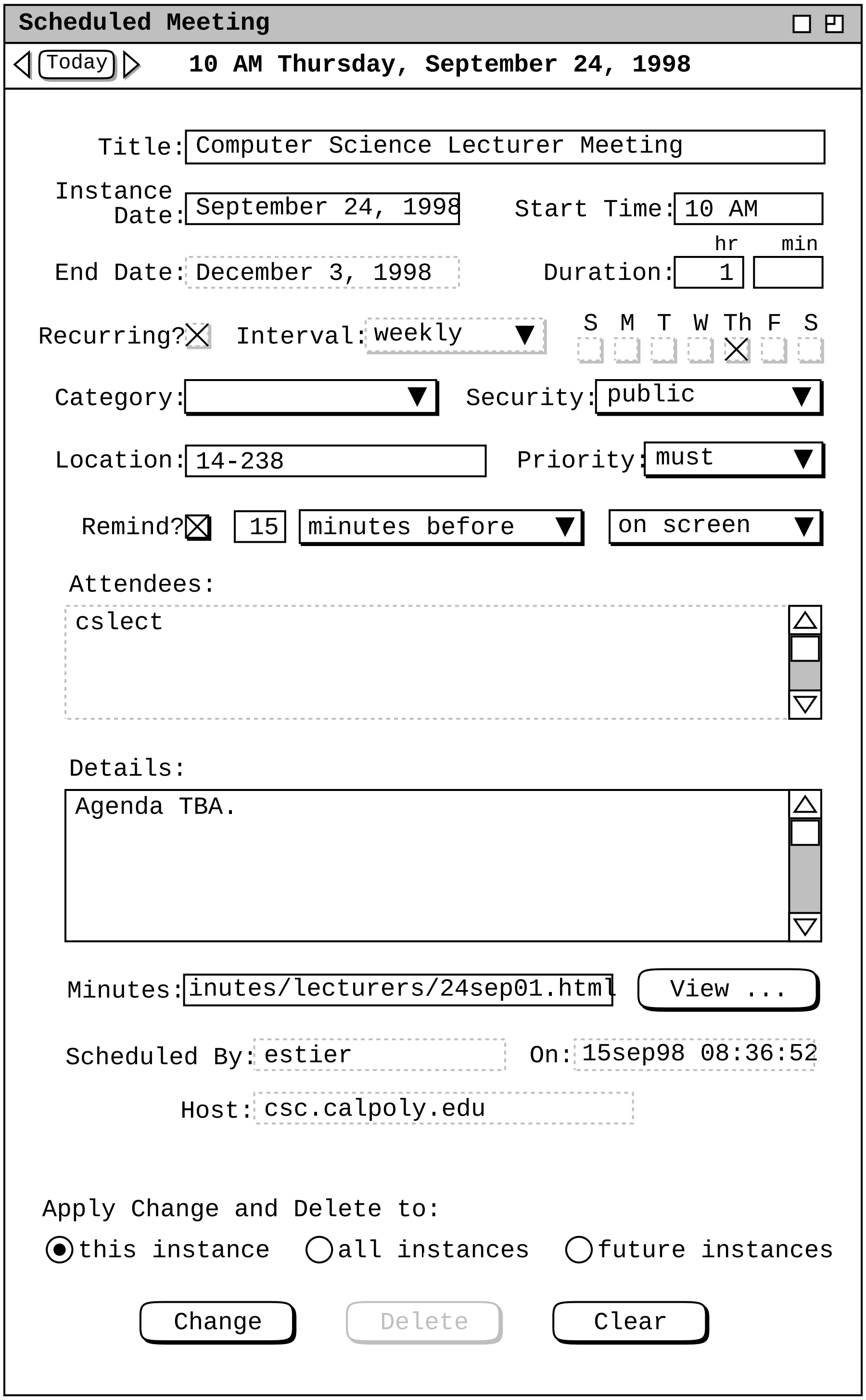
Figure 161 shows another change to the lecture meeting, this time to future instances, at and beyond the October 1 instance. The figure reflects the scheduler having navigated to the October 1 instance, i.e., the scheduler did not change the instance date. The changes are to the ‘`Location`’ and ‘`Details` 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 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:

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

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



Scheduled Meeting

10 AM Thursday, September 24, 1998

Title: Computer Science Lecturer Meeting

Instance Date: September 24, 1998

Start Time: 10 AM

End Date: December 3, 1998

Duration: hr 1 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:
cslect

Details:
Agenda TBA.

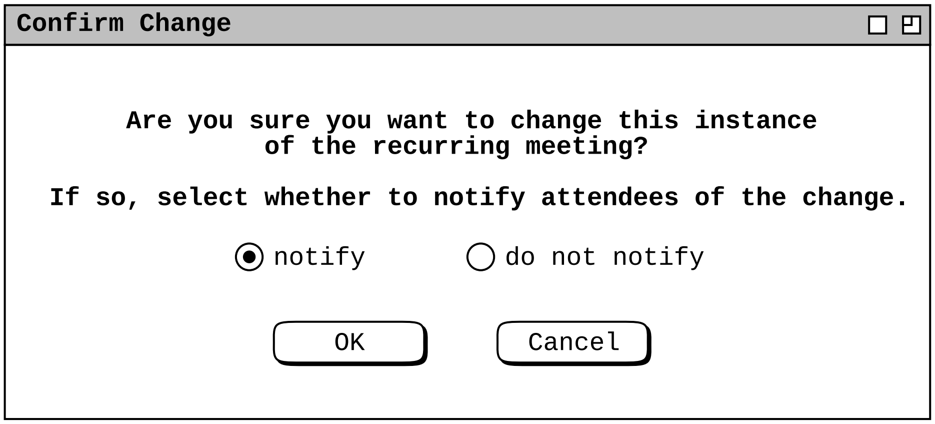
Minutes: inutes/lecturers/24sep01.html

Scheduled By: estier On: 15sep98 08:36:52

Host: csc.calpoly.edu

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

Figure 159: Scheduler edits a recurring meeting.

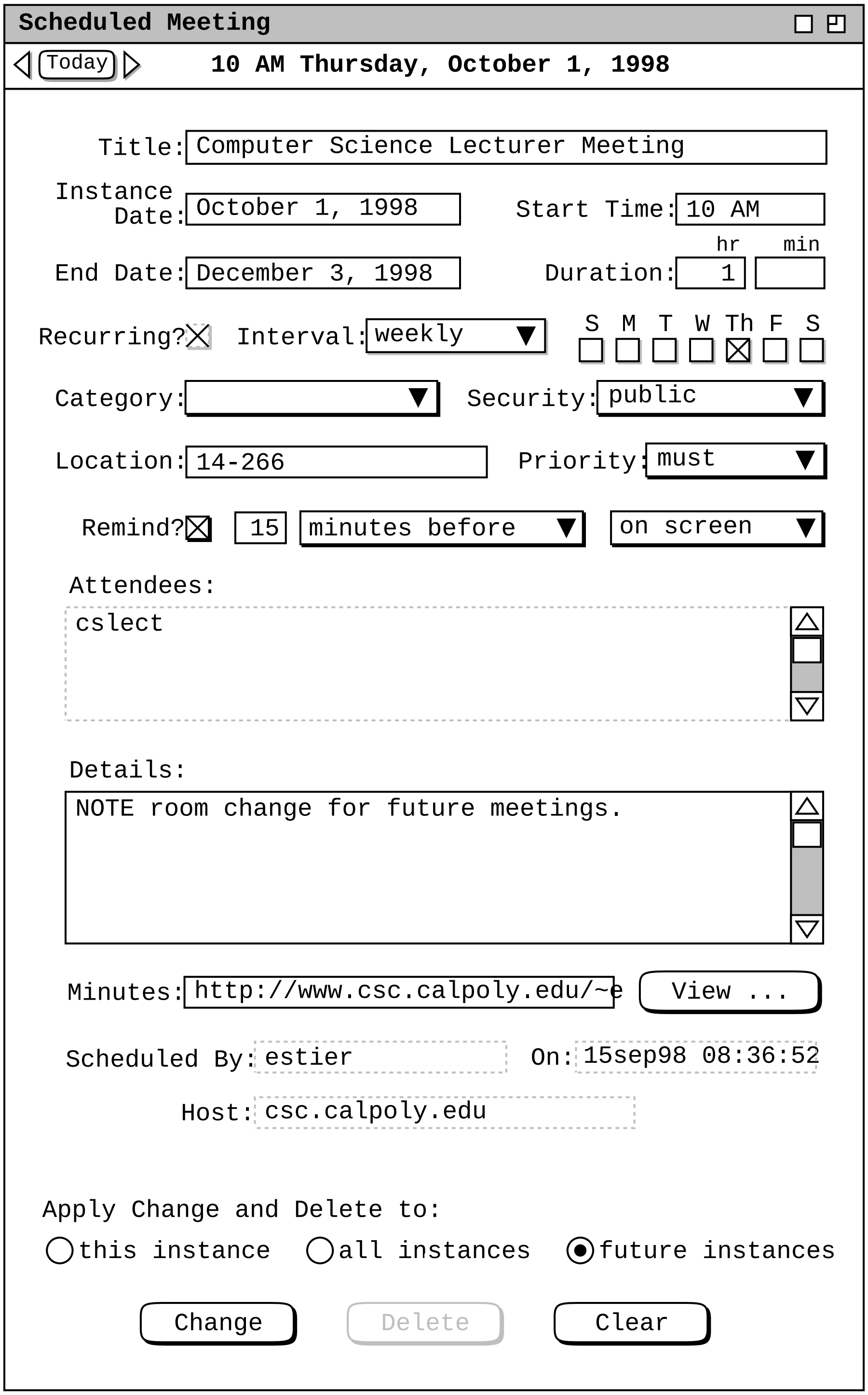


Are you sure you want to change this instance
of the recurring meeting?

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

● notify ○ do not notify

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



Scheduled Meeting

10 AM Thursday, October 1, 1998

Title: Computer Science Lecturer Meeting

Instance Date: October 1, 1998

Start Time: 10 AM

End Date: December 3, 1998

Duration: 1 hr, min

Recurring? ☒ Interval: weekly

S ☐ M ☐ T ☐ W ☐ Th ☒ F ☐ S ☐

Category:

Security: public

Location: 14-266

Priority: must

Remind? ☒ 15 minutes before on screen

Attendees: cslect

Details: NOTE room change for future meetings.

Minutes: http://www.csc.calpoly.edu/~e

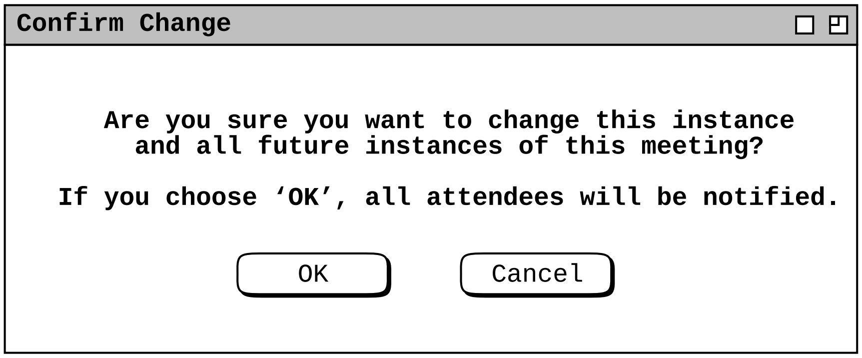
Scheduled By: estier On: 15sep98 08:36:52

Host: csc.calpoly.edu

Apply Change and Delete to:

○ this instance ○ all instances ● future instances

Figure 161: Location change to future lecturer meetings.



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

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

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

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

o The confirmation message appears below the conflict message, or at the top of the dialog if there is 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?**"

o 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.

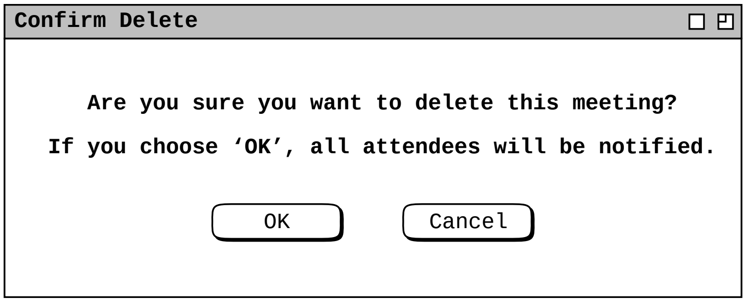
o At the bottom of the dialog are '`OK`' and '`Cancel`' buttons.

2.5.2.2.2. Scheduler Deleting a Meeting

To delete a non-recurring meeting, the scheduler presses '`Delete`' in the item display. In response, the system displays a dialog of the form shown in Figure 163. Scheduler deletion of a meeting always requires attendees 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 2.5.2.2.3. 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.

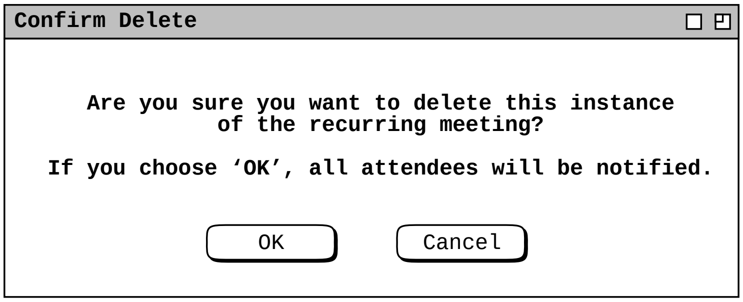


Confirm Delete

Are you sure you want to delete this meeting?

If you choose ‘OK’, all attendees will be notified.

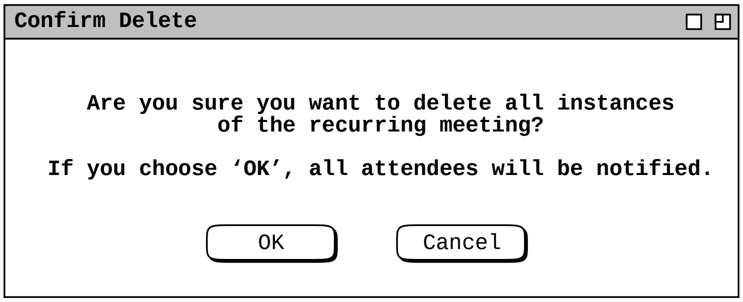
Figure 163: Confirmation dialog for scheduler-deleted meeting.



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

If you choose ‘OK’, all attendees will be notified.

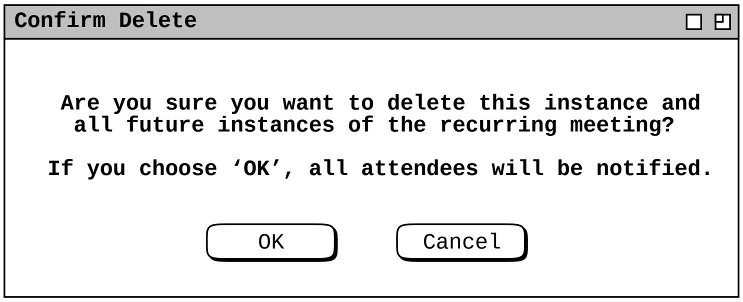
Delete confirmation for one instance of a recurring appointment.



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

If you choose ‘OK’, all attendees will be notified.

Delete confirmation for all instances of a recurring appointment.



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

If you choose ‘OK’, all attendees will be notified.

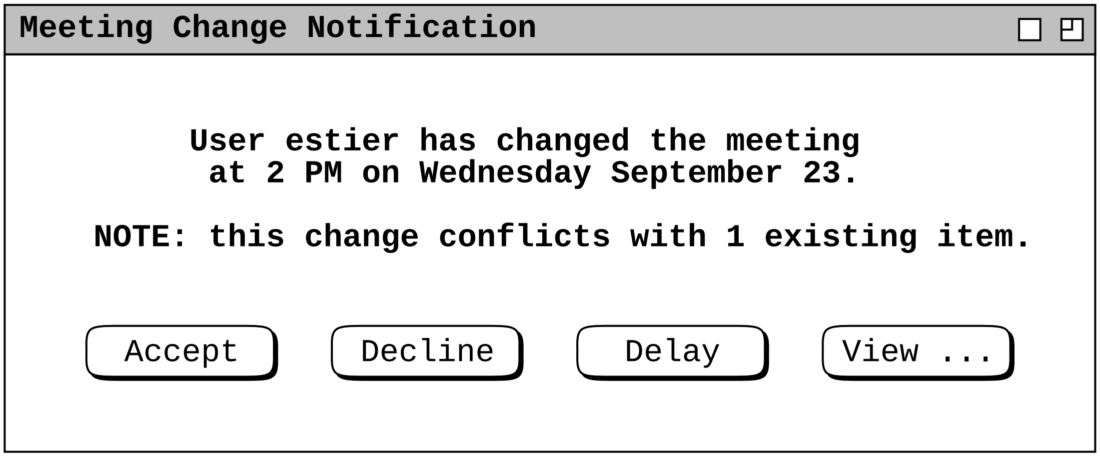
Delete confirmation for all future instances of a recurring appointment.

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

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 canceled 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. To bring the changed field(s) to the attention of notified users, the system color highlights the title and border of the changed field(s), as described in the next section.

2.5.2.2.3. 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. The notification is sent in an on-screen change notification 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.



User estier has changed the meeting
at 2 PM on Wednesday September 23.

NOTE: this change conflicts with 1 existing item.

Figure 165: Meeting change notification.

If the attendees list in a meeting is changed, the change notification is sent to those users appearing in the attendees list prior to the change who have not been deleted by the change. Users deleted from or newly added to the attendees list are sent a different form of notification, as described below.

When a scheduler change causes a time conflict with one or more existing items in a 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 singular if $N = 1$, plural otherwise.

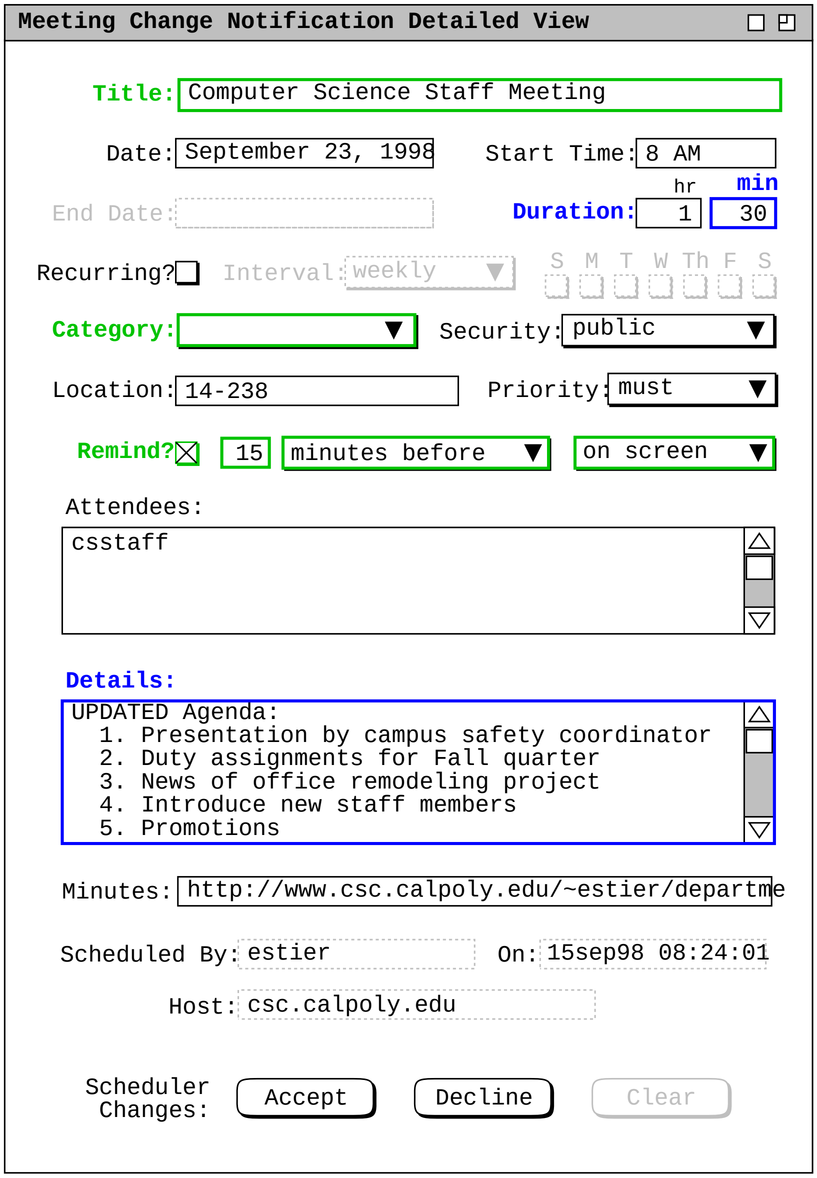
The user can view the complete details of the change by pressing the '`View ...`' button in the initial change notification dialog. For example, when the user presses '`View ...`' in Figure 165, the system responds by displaying the notification details dialog in Figure 166. The data fields changed by the scheduler are highlighted with widened blue borders and field titles in bold blue type. The data fields changed by the attending user are highlighted with widened green borders and field titles in bold green type. Unchanged data fields are shown with normal black borders and titles. Note that for multi-component data fields, such as `Duration`, the data field title is highlighted as are any changed components. Unchanged components are not highlighted, hence the '`hr`' component is shown in normal black type.

In Figure 166, the scheduler changes in blue are those made in a preceding scenario (see Figure 157). The non-scheduler changes in green were made before the user accepted the meeting (see Figure 113.). The case of conflicting scheduler and non-scheduler changes is not shown in Figure 157; it is covered later in Section 2.5.2.2.8.

All data values shown in Figure 166 are those in the scheduler's copy of the meeting. The user can compare these values with hers for the same meeting by viewing the same meeting using the '`View Item`' menu command, as discussed in Section 2.5.2.2.5

There are '`Accept`' and '`Decline`' buttons at the bottom of the detailed notification dialog. To the left of the buttons is a label explaining that the buttons apply to the 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 merged. The prudence of declining scheduler changes to a meeting may questionable, but left to the user's discretion.



Title: Computer Science Staff Meeting

Date: September 23, 1998

Start Time: 8 AM

End Date:

Duration: 1 hr 30 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:

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

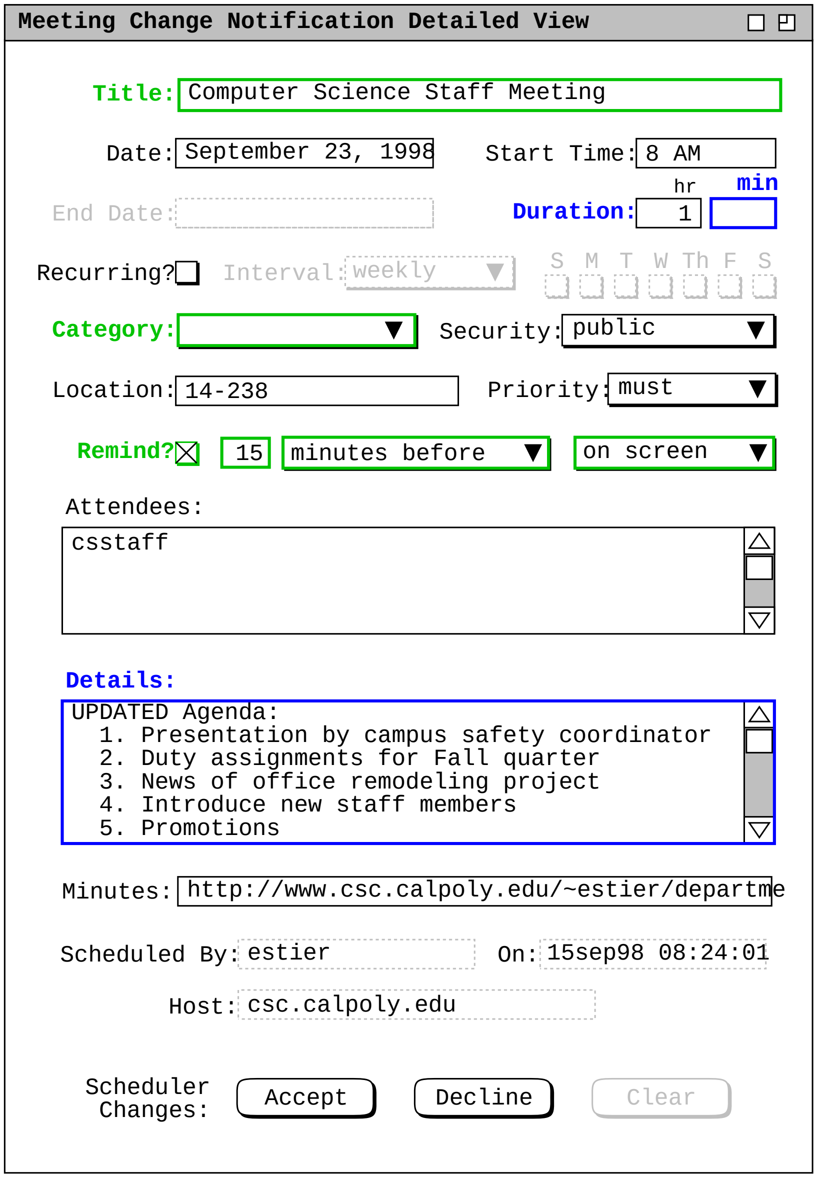
Scheduled By: estier On: 15sep98 08:24:01

Host: csc.calpoly.edu

Figure 166: Viewing details of a meeting change notification.

To delay the accept/decline decision, the user can close the notification dialog window, using the '`View Windows Close`' command or equivalent window shortcut. 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 2.5.2.2.5.



Meeting Change Notification Detailed View

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:

```
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
```

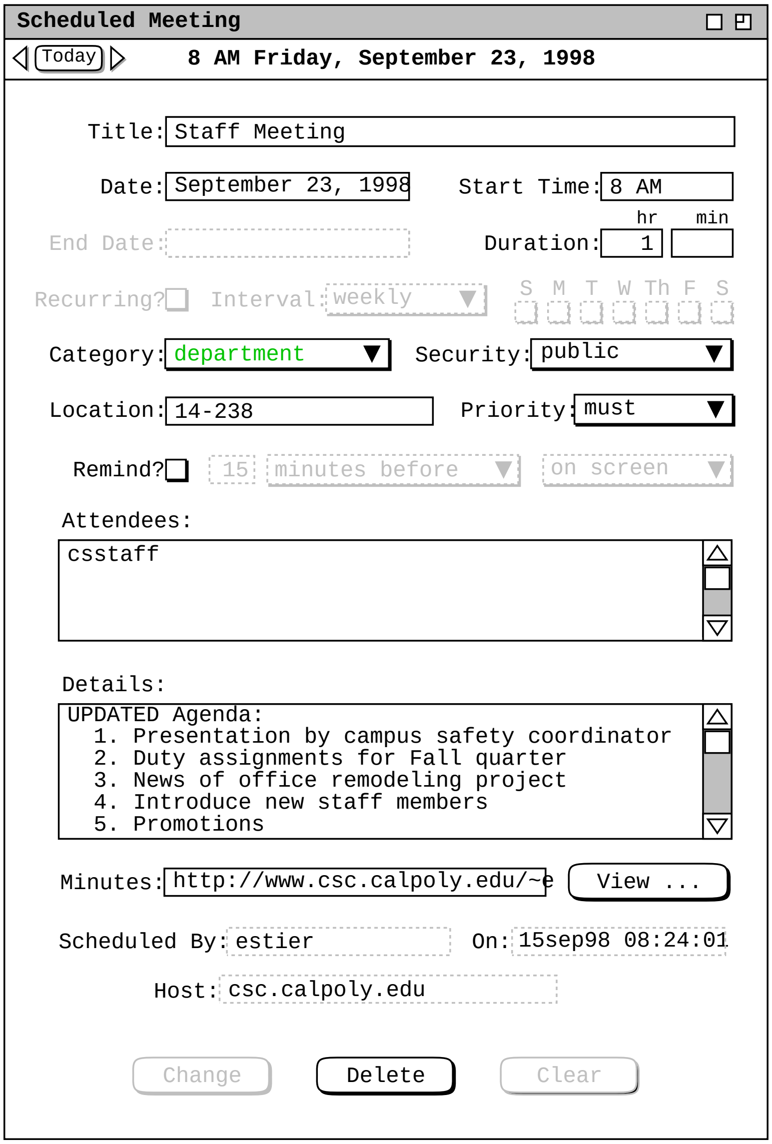
Minutes: http://www.csc.calpoly.edu/~estier/departme

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

Host: csc.calpoly.edu

Figure 167: Change notification details dialog edited.

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 where 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 unchanged by either the scheduler or user from their original scheduled values, and hence are retained as they are. Any user edits made in the notification dialog are reflected in the ultimately accepted changes. As a result, when the user executes '`View Item`' after pressing '`Accept`' in Figure 167, the system displays the item-level view shown in Figure 168. The item view has a full date description at the top, with '`Change`' and '`Delete`' command buttons



8 AM Friday, September 23, 1998

Title: Staff Meeting

Date: September 23, 1998

Start Time: 8 AM

End Date:

Duration: 1 hr, min

Recurring? ☐ Interval: weekly

Category: department

Security: public

Location: 14-238

Priority: must

Remind? ☐ 15 minutes before on screen

Attendees:

csstaff

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: http://www.csc.calpoly.edu/~e

Scheduled By: estier

On: 15sep98 08:24:01

Host: csc.calpoly.edu

Figure 168: Scheduler changes merged into meeting.

instead of '`Accept`' and '`Decline`'.

The policy to merge only the most recent changes may not be optimal in all circumstances. For example, the user may wish to retain a previous value, as was the case with the meeting '`Duration`' in the preceding example. 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.

There are two local copies of a 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 dialog reflects changes made by the scheduler. The scheduler-changed copy of the meeting is stored in the notification dialog itself, and is not yet a scheduled item in the user's calendar. The meeting in the user calendar *refers to* the scheduler-changed copy, via the '`View ...`' button, as shown in Figure 171. In terms of physical storage, the notification copy of the meeting is stored on the user's local computer when the notification is received from the central host computer. In this way, the user may subsequently disconnect from the central host and still have a record of the scheduler changes to that point. Further discussion of local and central data storage are covered in Section .

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*"

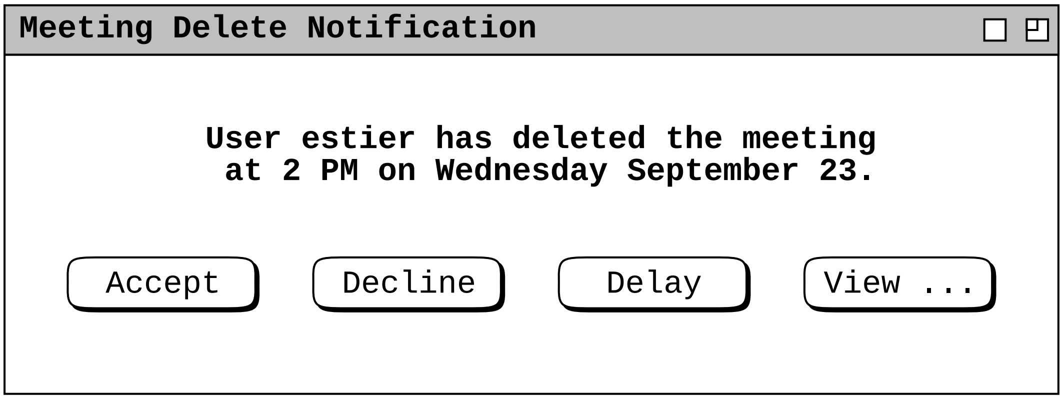
"**User** *user ID* **has changed all future instances**
of the recurring meeting starting on *date*"

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 canceled meeting.

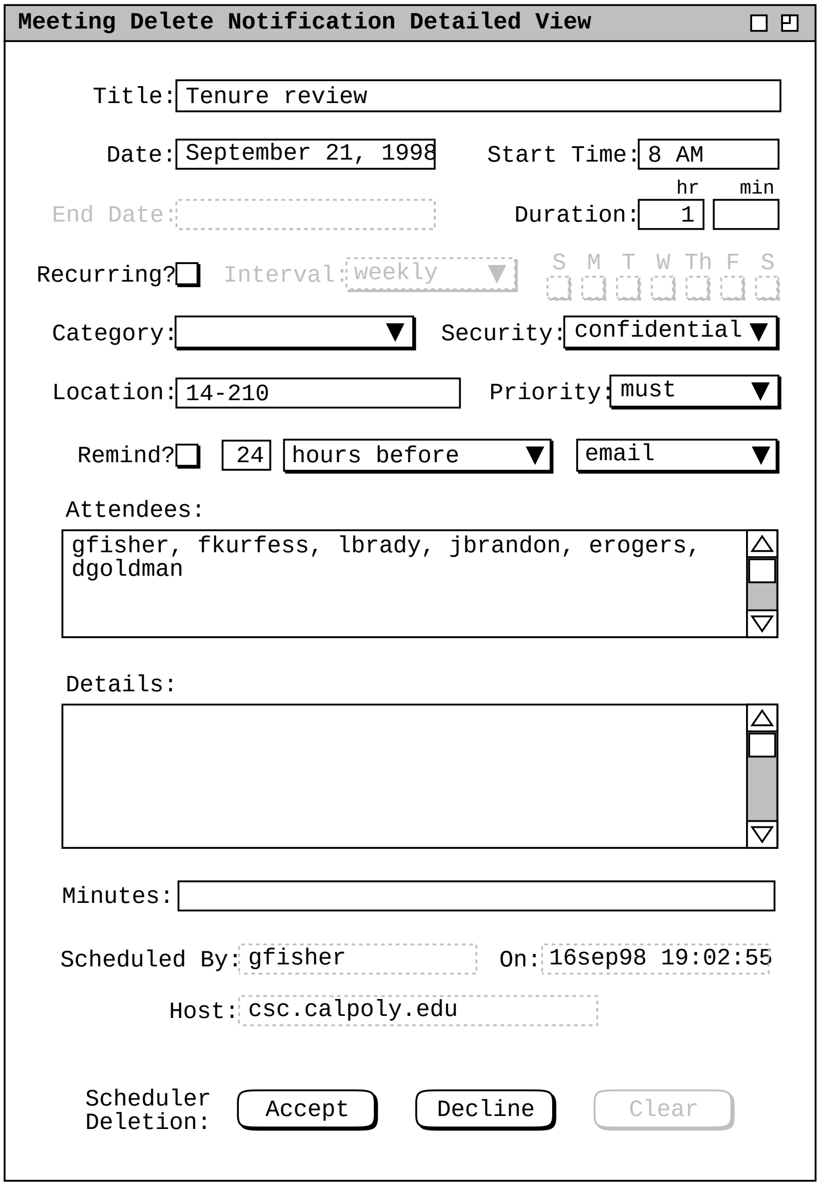
If the user presses ‘`Accept`’, the system deletes the meeting from the user’s calendar, updates all affected displays, and removes the delete notification dialog(s) from the screen. If the user presses ‘`Decline`’, no deletion is performed. If the user has edited one or more data fields prior to pressing ‘`Decline`’, the system displays a change confirmation of the form shown in Figure 172. If the user confirms the change, the system removes the notification



Meeting Delete Notification

User estier has deleted the meeting
at 2 PM on Wednesday September 23.

Figure 169: Delete notification dialog.



Title: Tenure review

Date: September 21, 1998

Start Time: 8 AM

End Date:

Duration: 1 hr, min

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

Category:

Security: confidential

Location: 14-210

Priority: must

Remind? ☐ 24 hours before email

Attendees: gfisher, fkurfess, lbrady, jbrandon, erogers, dgoldman

Details:

Minutes:

Scheduled By: gfisher On: 16sep98 19:02:55

Host: csc.calpoly.edu

Figure 170: Viewing details of a meeting delete 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 2.5.2.2.5

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. When an attending user is connected to the originating central host, and options settings permit, 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 2.5.2.2.5. Further details of dynamic view updating are covered in Section 2.5.6.

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 user makes the accept/decline decision, the system automatically executes decline on the user's behalf. As a result, the scheduler's changes are ignored, and the meeting is changed from penciled-in form back to a regular scheduled item. If the start date and time of the meeting pass before the notification dialog is allowed to appear, the system never displays it. The further details of notification handling discussed in Section 2.4.1.5 apply to change/delete notifications in the same way as they apply to originally scheduled meetings.

In terms of physical storage, the scheduler-changed copy of the meeting is effectively part of the change notification itself, not in the user's local calendar. This effective behavior is necessary so that the user can compare, if desired, the complete content of the extant local copy of the meeting with the scheduler-changed copy, as explained in Section 2.5.2.2.5.

In the case where a meeting change involves addition or deletion in the '`Attendees`' list, the notification for added or deleted attendees is different from the change notification described above. Specifically, each newly added attendee receives a regular meeting notification, as described in Section 2.4.1.5. The only difference in the added-attendee notification is that the third line of text is changed to "**`with you as an added attendee`**" instead of "**`with you as an attendee`**" (see Figure 110). The notification is sent to newly-added attendees in lieu of the change notification shown in Figure 165. The process of accepting or declining an added-attendee notification is precisely the same as described in Section 2.4.1.5.

Each removed attendee receives a meeting delete notification, as described above near Figure 169. The only difference in the deleted-attendee notification is that the first line of text is changed to "**`User ... has deleted you as an attendee of the meeting`**" instead of "**`User ... has deleted the meeting`**". This notification is sent to removed attendees in lieu of the change notification sent to retained attendees.

2.5.2.2.4. 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 .

2.5.2.2.5. 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.

The only meeting fields that a non-scheduler can never edit are '`Scheduled By`', '`On`', and '`Host`. This is because, as noted above, the set of values for these three fields constitutes the unique identification of a meeting on all users' calendars. All other fields are editable, subject to the following conditions:

a. the general data-entry rules in Section 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 in an item-level display until the user accepts or declines the scheduler changes.

Of note is that the scheduler editing restrictions in Table 10 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.

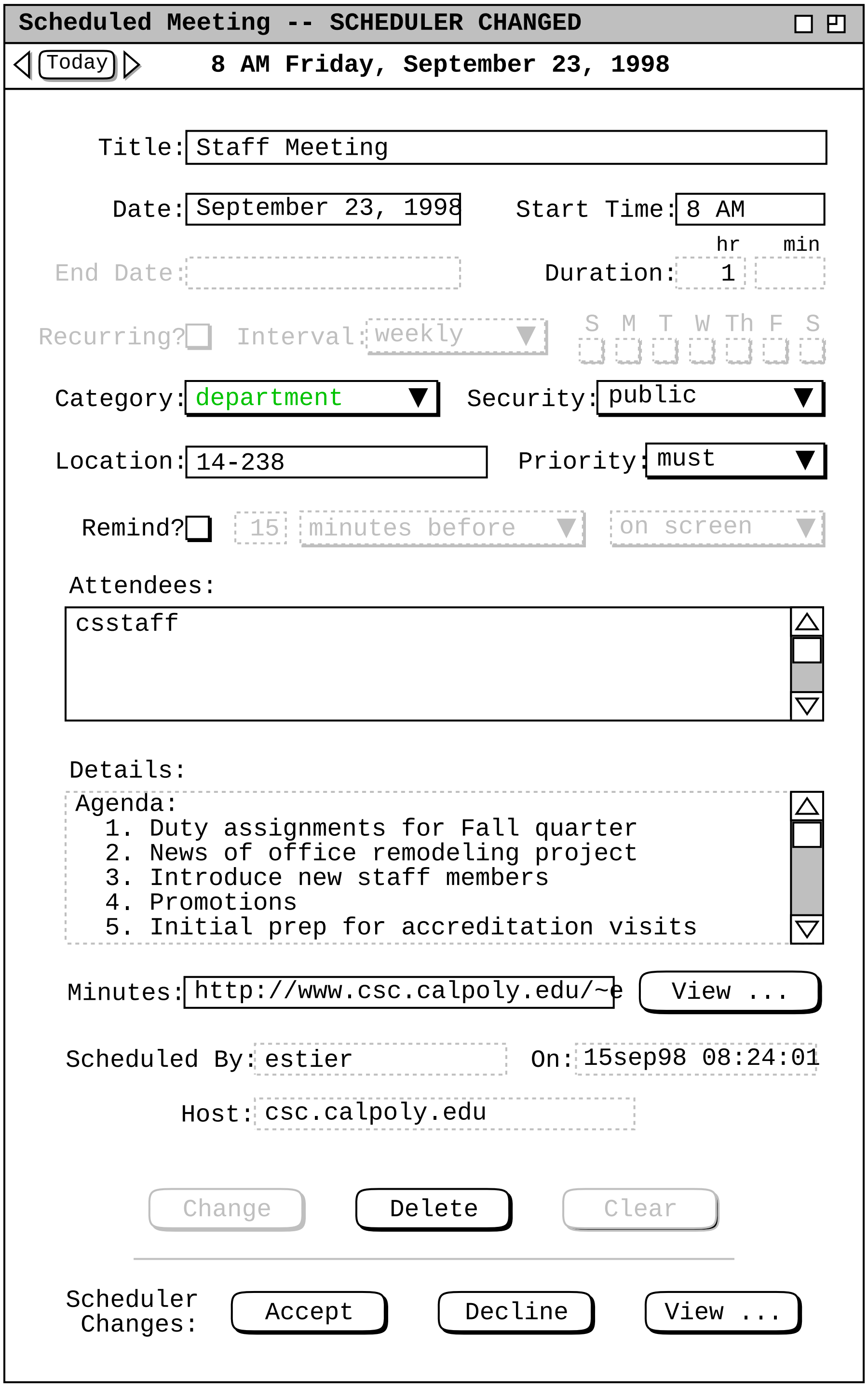
The display in Figure 171 contains the same information as the regular item-level view in Figure 137, with the addition of accept/decline command buttons at the bottom. For added clarity, the banner of the display is augmented with the "**SCHEDULER CHANGED**" text. The additional command buttons have 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 with blue highlight 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.

As explained in a preceding scenario, the scheduler-changed copy of the meeting is not officially a scheduled item in the user's calendar. As such, it does not appear separately in calendar views, only the user's copy so appears. The only access to the scheduler-changed copy is through the '`View`' command in the item-level view of the user's copy (Figure Figure 171) or in the meeting change notification (Figure 165).

Once any scheduler-generated changes to a meeting are dealt with, the use may make any desired changes to a meeting. 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.



Title: Staff Meeting

Date: September 23, 1998 Start Time: 8 AM

End Date: Duration: 1 hr min

Recurring? ☐ Interval: weekly

Category: department Security: public

Location: 14-238 Priority: must

Remind? ☐ 15 minutes before on screen

Attendees:
csstaff

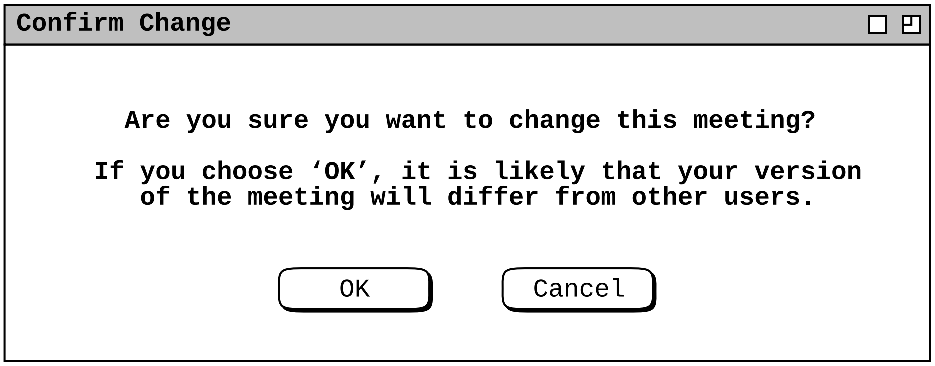
Details:
Agenda:
1. Duty assignments for Fall quarter
2. News of office remodeling project
3. Introduce new staff members
4. Promotions
5. Initial prep for accreditation visits

Minutes: http://www.csc.calpoly.edu/~e

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

Host: csc.calpoly.edu

Figure 171: Item-level view of meeting with not-yet-accepted scheduler changes.



Are you sure you want to change this meeting?

If you choose ‘OK’, it is likely that your version of the meeting will differ from other users.

Figure 172: Non-scheduler change confirmation dialog.

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.

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.

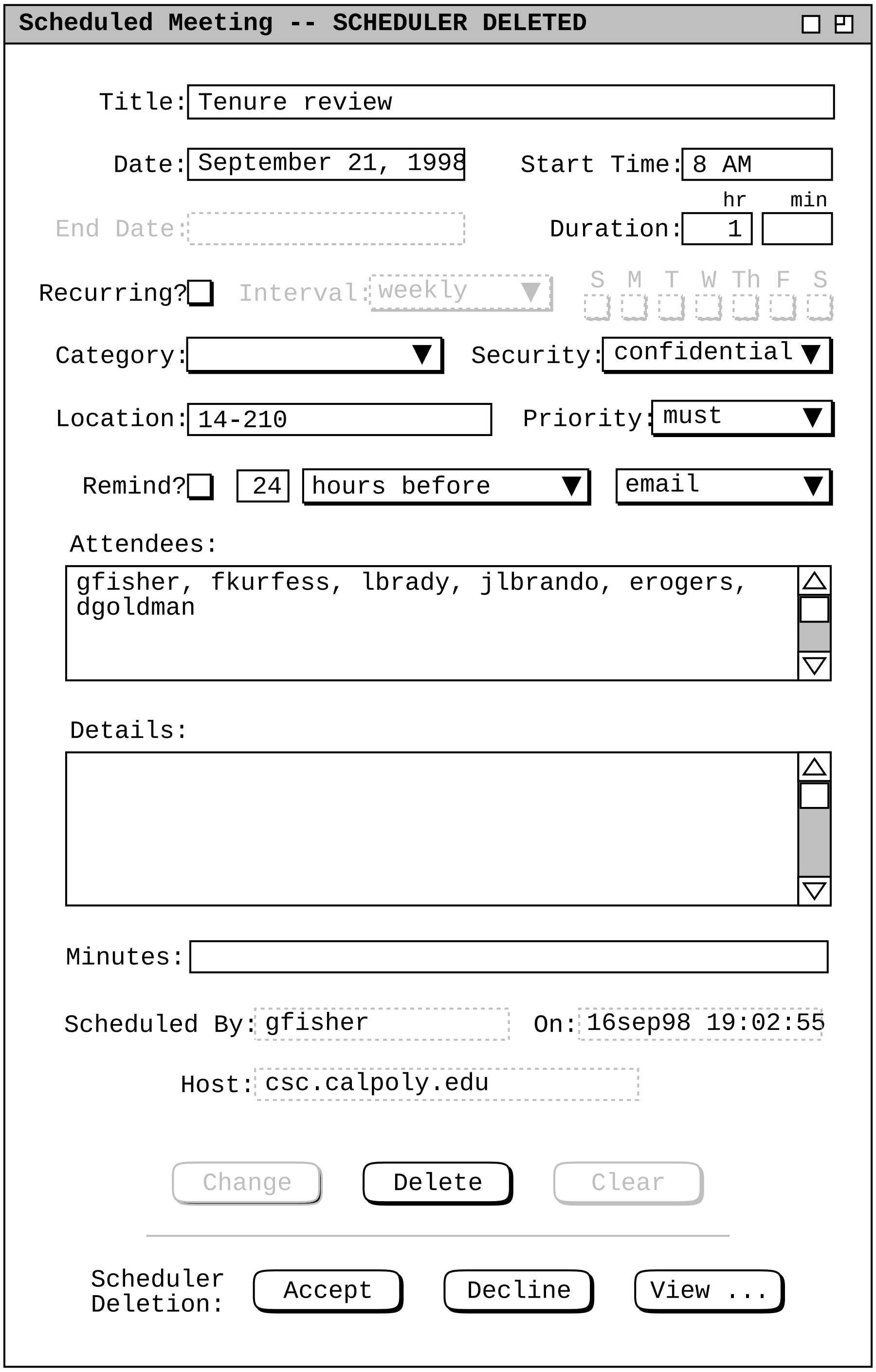
The two groups of command buttons in Figures 171 and 173 correspond to the two different copies of a meeting with not-yet-accepted scheduler changes or deletion. The change/delete buttons operate on the pre-accept copy. The accept/decline buttons operate on the notification copy with the scheduler's 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".

A non-scheduler deleting a meeting has no effect on any other users' copies of the meeting. In particular, a user who deletes a meeting is not removed from the attendees list in any one else's copy of the meeting.

After an attending user deletes a meeting, that user does not receive any subsequent change or delete notifications from the scheduler of the meeting. Operationally, this can mean that the notifications are never sent to the local computer, or are deleted upon receipt by the local computer. The *effective* behavior is that a user never sees change/delete notifications for a locally deleted meeting, i.e., a meeting deleted by an attending user but not by the scheduler.

It is possible for a meeting in the user's local calendar to be in the not-yet-accepted state when the scheduler makes (further) changes, or deletes the meeting. This is the case where the user has not yet accepted the original scheduling of the meeting, but the scheduler has made a later change or deleted the meeting. Operationally, there is no difference in such cases compared to that described above (around Figures 171 and 173), except that the banner of the item-level display window reflects the not-yet-accepted state of the original meeting. Specifically, when the scheduler changes a not-yet-accepted meeting, the item-level banner is "**Not−Yet−Accepted Meeting -- SCHEDULER CHANGED**" (cf. Figure 171). The banner for a scheduler-



Scheduled Meeting -- SCHEDULER DELETED

Title: Tenure review

Date: September 21, 1998
Start Time: 8 AM

End Date:
Duration: 1 hr, min

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

Category:
Security: confidential

Location: 14-210
Priority: must

Remind? ☐ 24 hours before email

Attendees: gfisher, fkurfess, lbrady, jlbrando, erogers, dgoldman

Details:

Minutes:

Scheduled By: gfisher
On: 16sep98 19:02:55

Host: csc.calpoly.edu

Figure 173: Item-level view of meeting with not-yet-accepted scheduler deletion.

deleted not-yet-accepted meeting is "**Not−Yet−Accepted Meeting -- SCHEDULER deleted**" (cf. Figure 173.).

2.5.2.2.6. Offline Scheduler Change and Delete

As noted in Section 2.5.2.2.1, the scheduler must be connected to the originating central host in order for meeting attendees to be notified of changes or deletions. If the scheduler is not so connected, she cannot change any critical meeting field or delete the meeting until a connection is re-established. If the scheduler attempts to confirm such a change or delete, the system displays the error message described in Section . The scheduler may change non-critical meeting fields when not connected to the originating host. When the scheduler confirms such a change, the system displays the warning dialog shown in Figure 174. This dialog appears whenever the scheduler is not connected to the originating host, even if connected to some other host. There are appropriate different forms of the dialog text for recurring meetings, as described in Section 2.5.2.2.1.

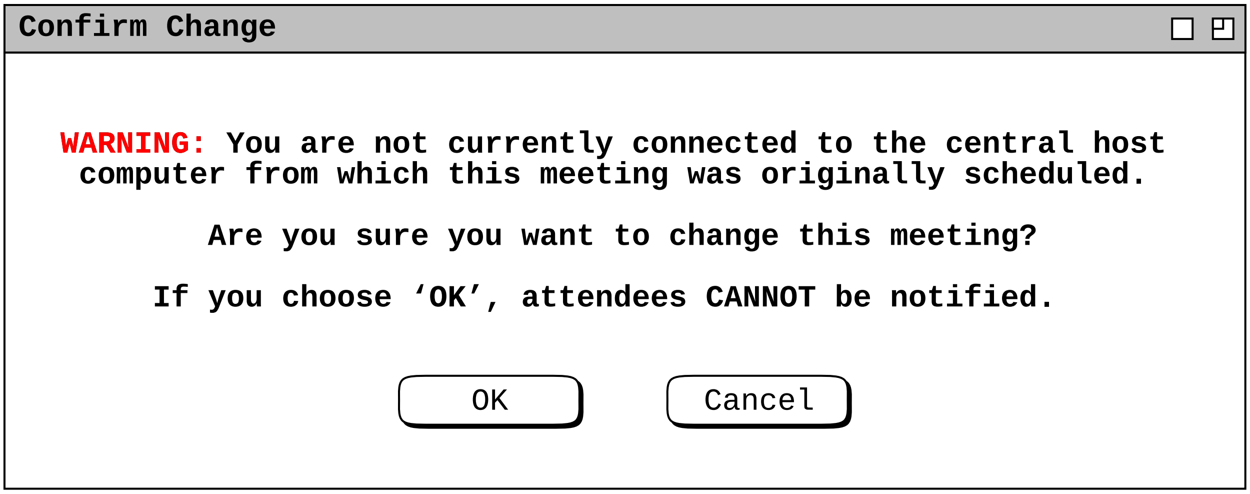
If the scheduler subsequently connects to the originating host, the attendees are *not* then notified. The scheduler must make and confirm additional changes in order to have the opportunity to notify attendees. When a subsequent change is confirmed while connected, it is based on the cumulative editing that has been performed, both online and offline, since the last time the meeting was synchronized.

2.5.2.2.7. Offline Non-Scheduler Changes

A non-scheduling user may make changes to her local copy of a meeting whether or not she is connected to the originating central host. The only operational difference when not connected is that the user may change a field that has also been changed by the scheduler. Such a change is possible when a scheduler change notification has not yet been received due to the lack of connection, or having set the notification option to '`never`'. When a scheduler change notification *has* been received, the system is able to prevent conflicting changes by disabling edits to scheduler-changed meeting fields, as shown in Figure 171. The system can continue to prohibit conflicting changes any time after the change notification has been received, even when subsequent changes are made offline. This is the case because the system retains the change notification on the local computer once received.

2.5.2.2.8. Conflicting Scheduler and Non-Scheduler Changes

A conflicting meeting change occurs when a non-scheduling user changes a meeting field that has also been changed by the scheduler, before the scheduler change notification has been received. As discussed in the preceding scenarios, the Calendar Tool system is designed to avoid conflicting meeting changes insofar as is possible. As long as a non-scheduling user remains up to date with scheduler change notifications, the system prohibits conflicting changes. The user can accomplish this up-to-datedness by retaining or establishing a central host connection any



Confirm Change

WARNING: You are not currently connected to the central host computer from which this meeting was originally scheduled.

Are you sure you want to change this meeting?

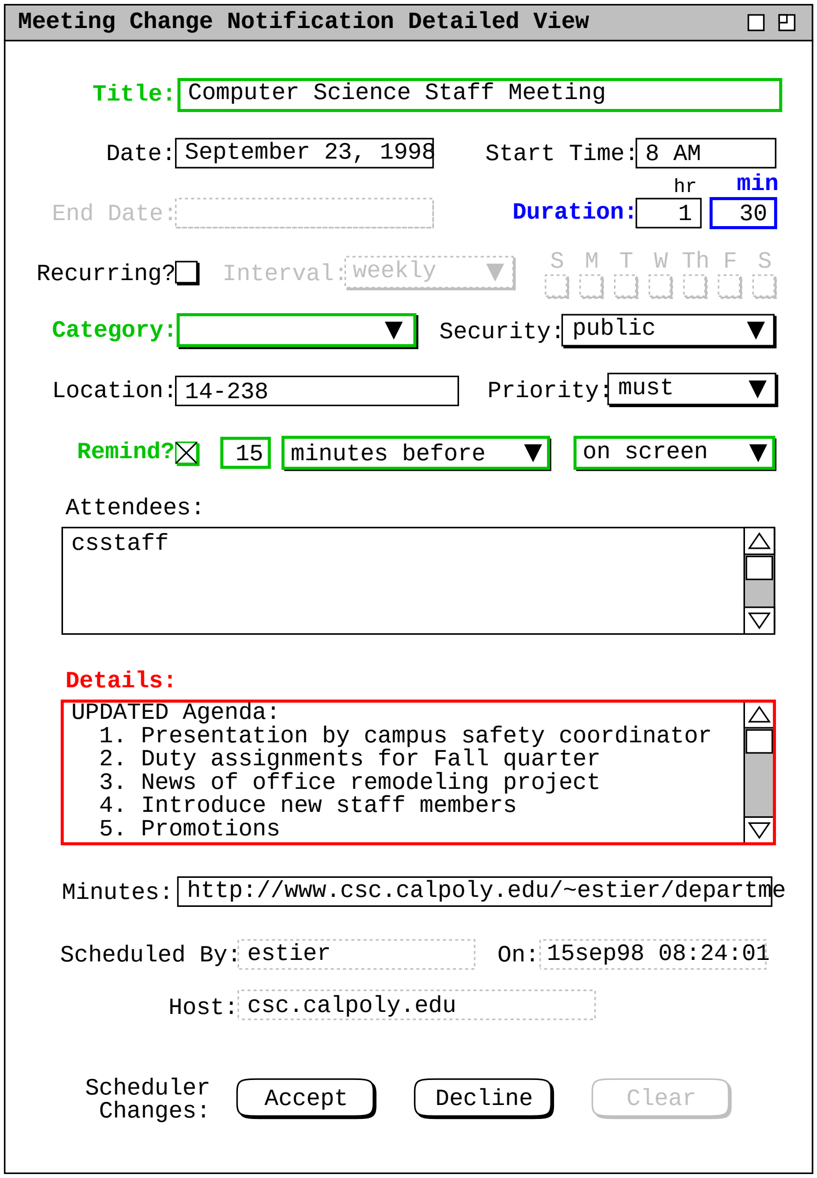
If you choose 'OK', attendees CANNOT be notified.

Figure 174: Warning when changing a meeting while not connected to the originating host.

time prior to editing a scheduled meeting, and by not setting the meeting notification option to '`never`'.

A conflicting change becomes apparent to the non-scheduling user when the user reconnects to the originating host and then receives the scheduler's change notification. In this case, conflicting data fields are shown with red highlights. For example, suppose the user had made a conflicting change to the '`Details`' field of the meeting shown in Figure 166. In such a case, the change notification dialog would appear as shown in Figure 175. Here the red-highlighting indicates the conflict in the '`Details`' field.

In terms of change merging, the system treats conflicting changes the same as non-conflicting changes. Namely, if the user presses '`Accept`', the system merges the scheduler's and user's changes by retaining the most recently entered value for each meeting data field. Given that the merge behavior is the same for both conflicting and non-conflicting changes, the only purpose for distinguishing them is informational. That is, when viewing the change



Title: Computer Science Staff Meeting

Date: September 23, 1998

Start Time: 8 AM

End Date:

Duration: 1 hr 30 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:

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

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

Host: csc.calpoly.edu

Figure 175: Conflicting change.

notification dialog, the use can see precisely which fields have been changed and by whom.

2.5.2.2.9. Pending Change/Delete Notifications

Depending on the state of an attendee's connection to a central host, it is possible for a scheduler to send one or more change/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 be 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 meeting 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 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.

2.5.2.3. 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

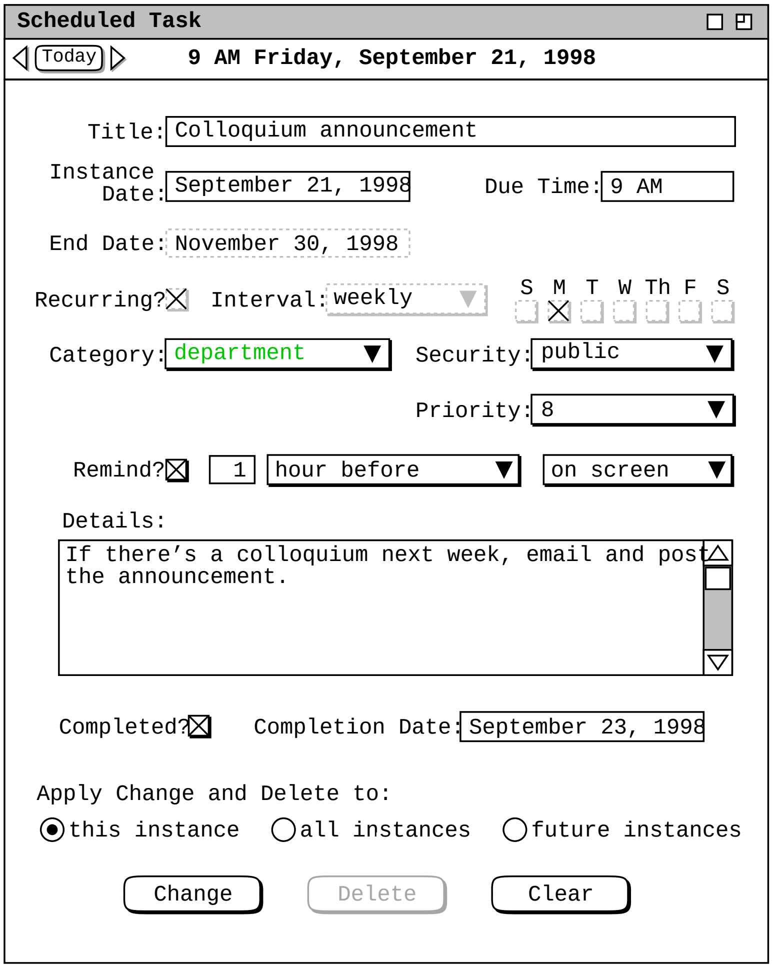
Editing functionality for all other fields is the same as for appointments. Figure 176 shows the user having edited an instance of the recurring task shown being scheduled in Figure 126 and being viewed in Figure 140. Here the '`Completed`' check box has been turned on by the user. When the user turns '`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 176, 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.

2.5.2.4. 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 177 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 178. 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 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.



Scheduled Task

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:

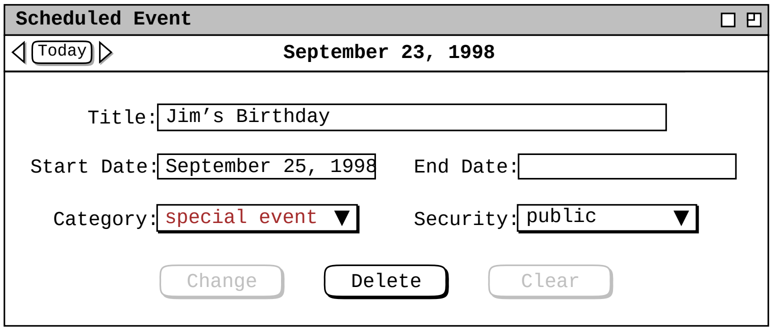
If there’s a colloquium next week, email and post the announcement.

Completed? ☒ Completion Date: September 23, 1998

Apply Change and Delete to:

● this instance ○ all instances ○ future instances

Figure 176: Edits made to a recurring task.



Scheduled Event

September 23, 1998

Title: Jim’s Birthday

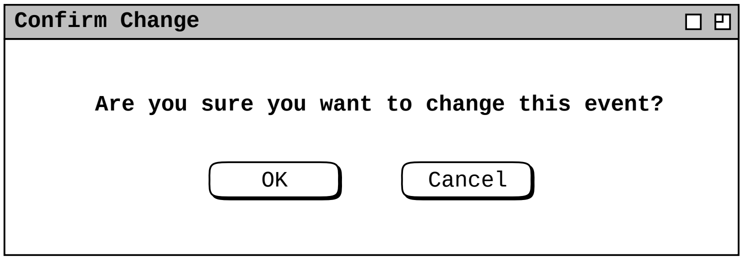
Start Date: September 25, 1998

End Date:

Category: special event

Security: public

Figure 177: Birthday event changed.



Are you sure you want to change this event?

Figure 178: Event change confirmation dialog.

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 There is an option to hide these events if the user does not want to see them.

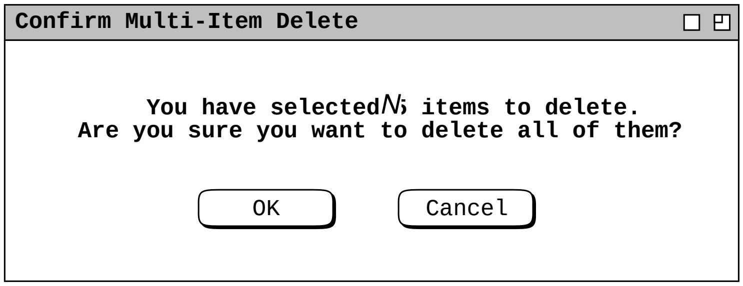
2.5.2.5. Deletion Shortcut from Higher-Level Calendar Views

The user may select calendar items for deletion in any daily, weekly, or monthly view. To do so, the user selects the item(s) to be deleted, and then presses the ‘`delete`’ key. To select more than one item, the user clicks on successive items while holding down the shift key. Complete details of item selection are covered in Section .

When the user presses ‘`delete`’ for a single selected item, the system responds in precisely the same manner as when the user presses the ‘`Delete`’ button in the item-level display for that item, as described in the preceding scenarios. In the case where the user selects a single instance of a recurring item for deletion, the system responds as when the user selects the ‘`this instance`’ radio button in the item-level display and presses the ‘`Delete`’ button.

When the user presses ‘`delete`’ for more than one selected item, the system responds with a confirmation dialog of the form shown in Figure 179. The dialog indicates how many items are to be deleted, with *N* replaced with the number of items currently selected by the user. When the user presses ‘`OK`’, all of the items are deleted, with all associated effects as described in the preceding scenarios.

It is noteworthy that only individual instances of recurring items can be deleted using selection and the ‘`delete`’ key. In order to delete all or all future instances of a recurring item, the user must do so in an item-level display, using the radio buttons under "`Apply Change and Delete to:`".



You have selected *N* items to delete.
Are you sure you want to delete all of them?

Figure 179: Confirmation dialog for deleting multiple items.

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 180 shows the general format. The '`Interval`' menu in Figure 181 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, 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, 2.4.1, and Section 2.4.2.

For any weekly or bi-weekly recurring item, there are constraints on the start and end dates. Namely, both these dates must fall on days that are checked in '`Recurring`' check boxes. For example, if the start date falls on a Tuesday, then Tuesday must be among the checked recurring days. If the user enters values for the start or end dates that violate these constrints, then then system displays an error dialog, as discussed in Section .

2.5.3.2. Monthly Intervals

When the user selects the monthly interval, the system changes the recurring information display as shown in Figure 182. When the user presses '`Details ...`', the system displays the dialog in Figure 183. 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`' alternative allows the user to select one or more days within the different weeks of the month. The '`Date`' alternative

Figure 180: Information for a recurring item.

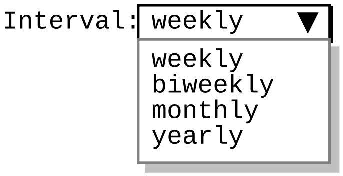
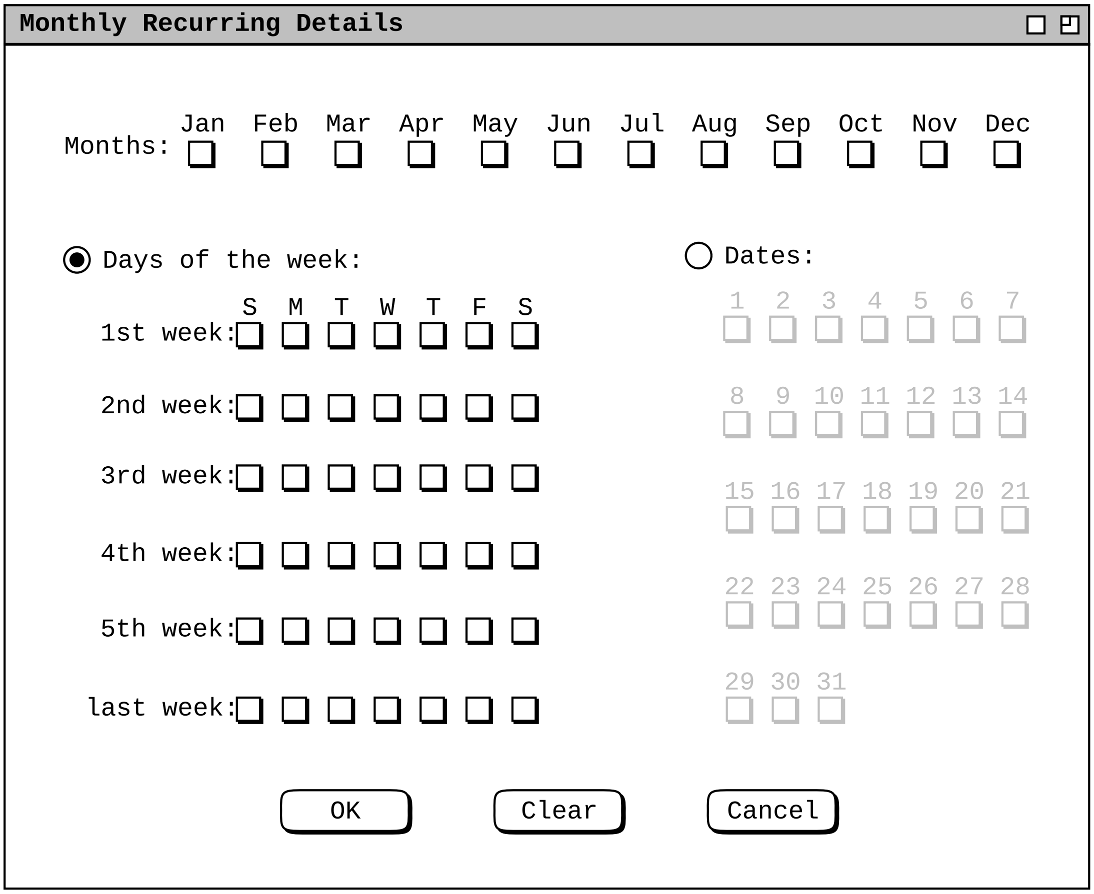


Figure 181: Interval selections.

Recurring? ☒ Interval: monthly

Figure 182: Monthly interval selected.



● Days of the week:

○ Dates:

Figure 183: Monthly interval scheduling dialog.

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.

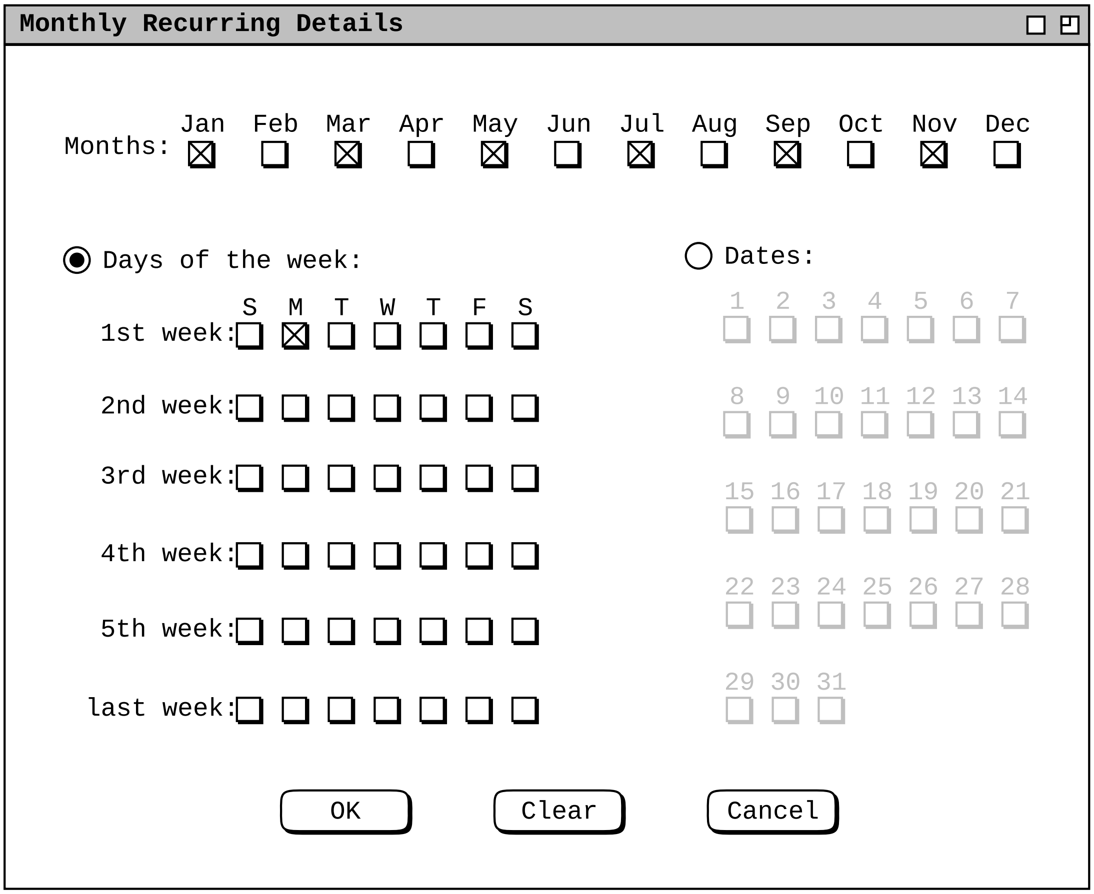
Figure 184 shows settings for an item that recurs on the first Monday of every other month, starting in January. Figure 185 shows an item that recurs on the first and third Friday of every month. An item scheduled in this way recurs at most two times per month. In particular, it does not occur on any fifth Fridays, when they happen. Figure 186 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



Monthly Recurring Details

Months:

Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
☒	☐	☒	☐	☒	☐	☒	☐	☒	☐	☒	☐

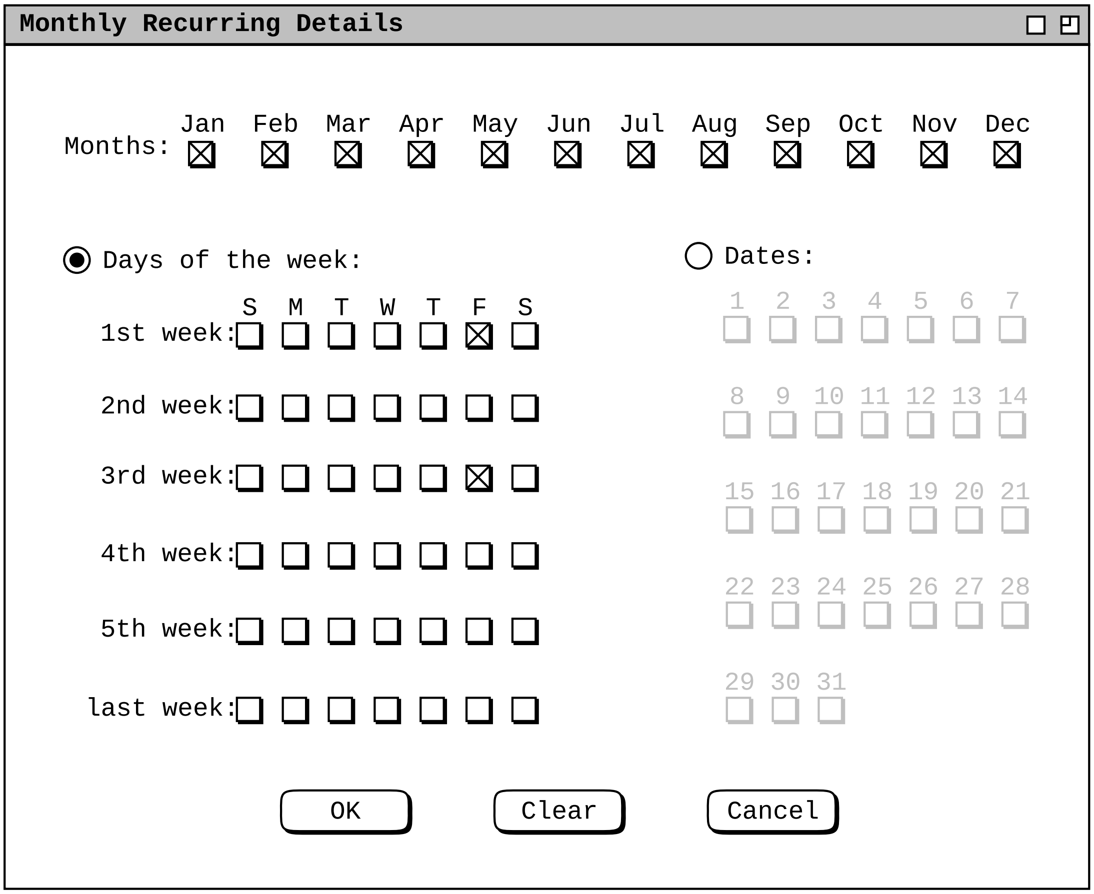
● Days of the week:

	S	M	T	W	T	F	S
1st week:	☐	☒	☐	☐	☐	☐	☐
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				
☐	☐	☐				

Figure 184: Item recurs on the first Monday of every other month.



	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Months:	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒

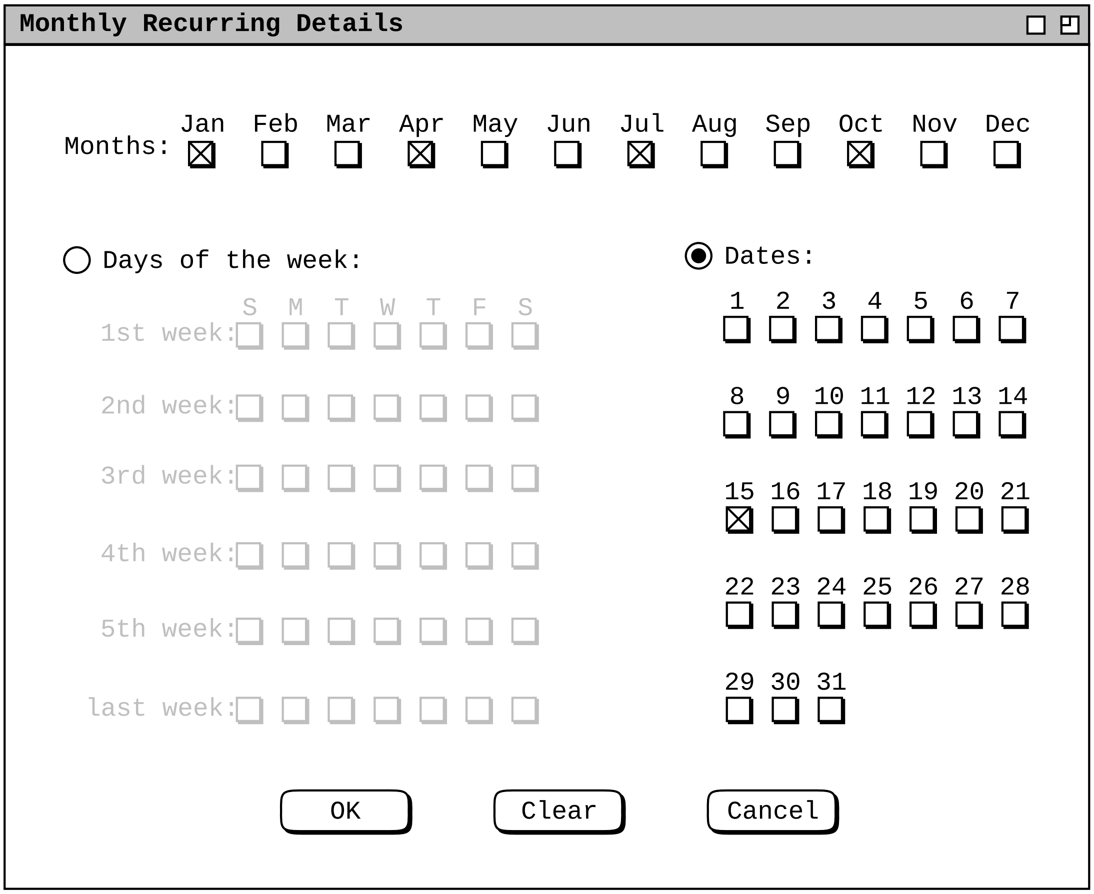
● Days of the week:

	S	M	T	W	T	F	S
1st week:	☐	☐	☐	☐	☐	☒	☐
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 ☐				

Figure 185: 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:

● 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 ☐

Figure 186: Item recurs quarterly on the 15th.

'`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 30-day months are selected, or if '`Feb`' is selected along with one or more 30-day months. The 30-day months are '`Apr`', '`Jun`', '`Sep`', and '`Nov`'. The 29 selection applies to '`Feb`' only on leap years. Since leap year occurrence varies from year to year, The 29 selection is never disabled, even when only '`Feb`' is selected.

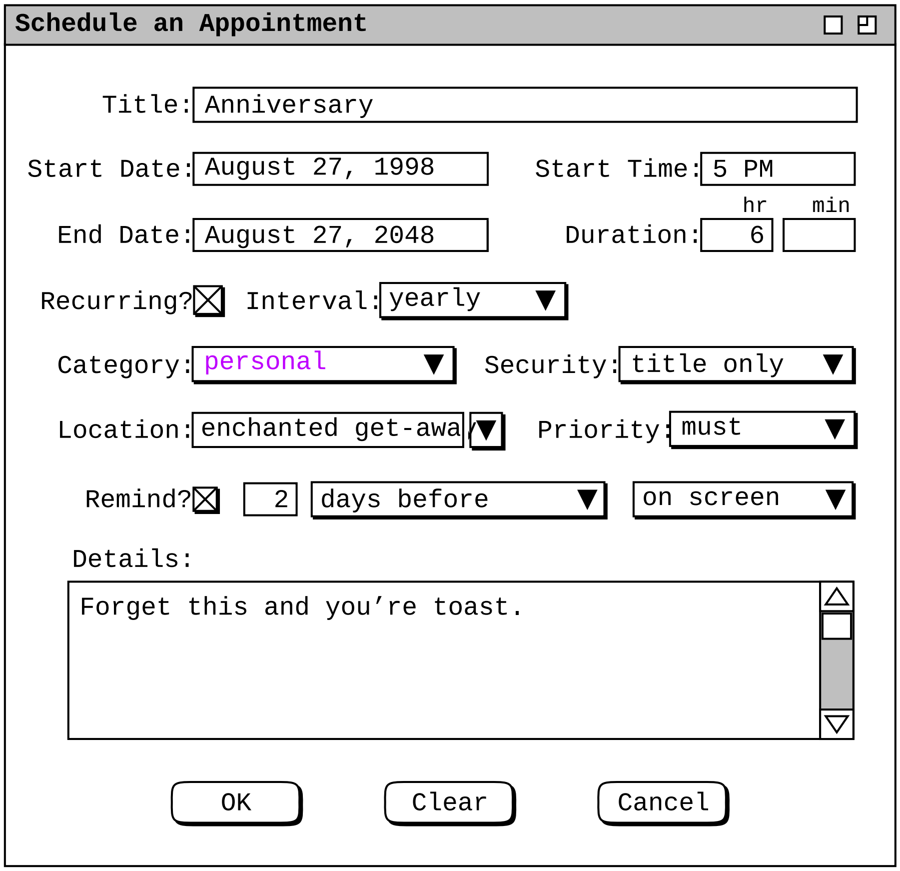
2.5.3.3. Yearly Intervals

The yearly interval setting means that an item recurs once every year on the same date. For example, Figure 187 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 188. 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.



Schedule an Appointment

Title: Anniversary

Start Date: August 27, 1998

Start Time: 5 PM

End Date: August 27, 2048

Duration: 6 hr min

Recurring? ☒ Interval: yearly

Category: personal

Security: title only

Location: enchanted get-awa

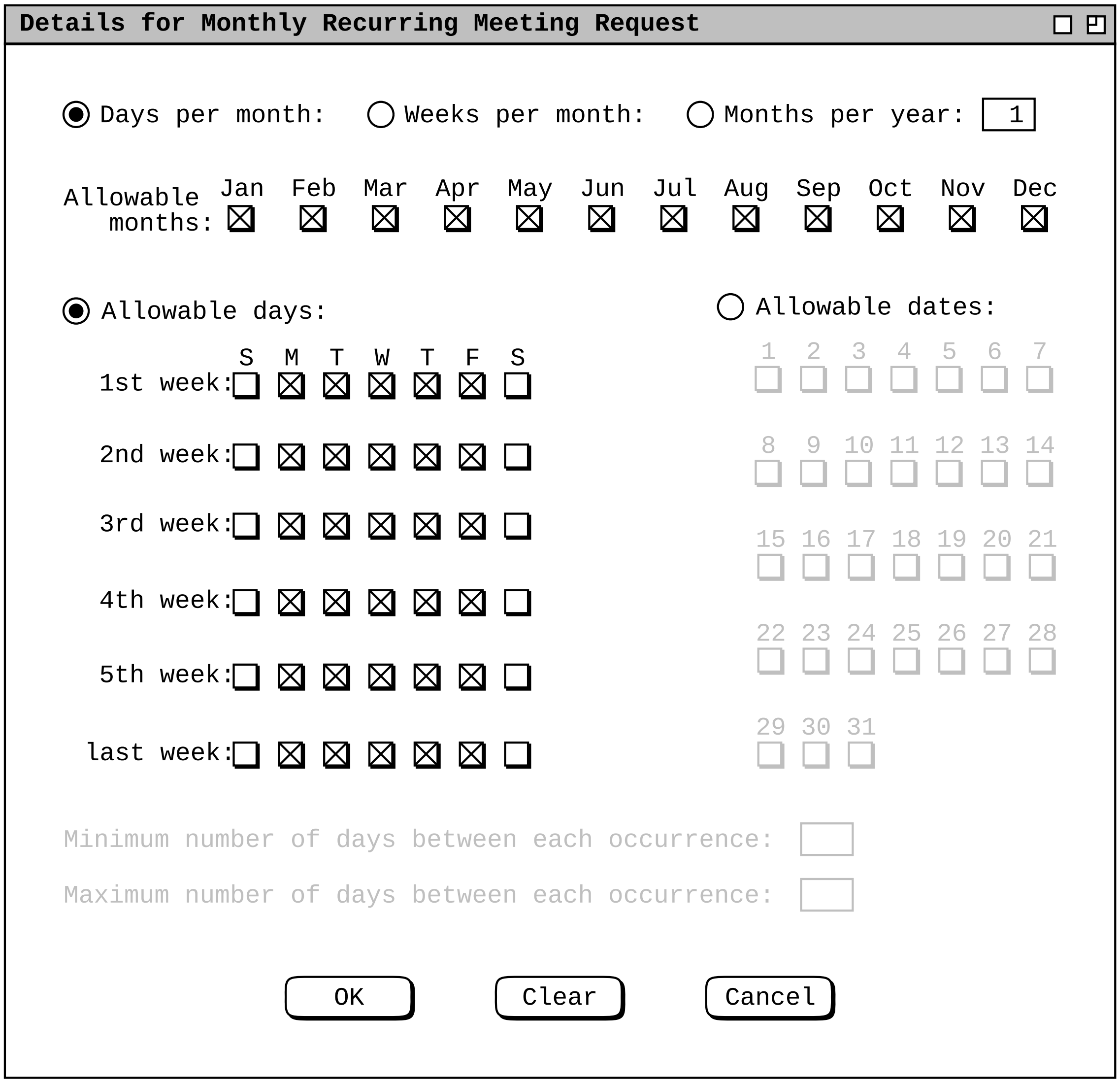
Priority: must

Remind? ☒ 2 days before on screen

Details:

Forget this and you’re toast.

Figure 187: Anniversary appointment.



Details for Monthly Recurring Meeting Request

● Days per month: ○ Weeks per month: ○ Months per year: 1

Allowable months:	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒

● Allowable days:

	S	M	T	W	T	F	S
1st week:	☐	☒	☒	☒	☒	☒	☐
2nd week:	☐	☒	☒	☒	☒	☒	☐
3rd week:	☐	☒	☒	☒	☒	☒	☐
4th week:	☐	☒	☒	☒	☒	☒	☐
5th week:	☐	☒	☒	☒	☒	☒	☐
last week:	☐	☒	☒	☒	☒	☒	☐

○ Allowable 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 ☐				

Minimum number of days between each occurrence:

Maximum number of days between each occurrence:

Figure 188: Monthly recurring details for a meeting request.

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 dialog in Figure 184. 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 188 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 the string `"days"` is replaced with `"weeks"` in both the minimum and maximum labels. If '`Months per year`' is on, the 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 effectively-scheduled instances in the user's calendar. The instances appear on the dates specified by the recurring information. The idea of an *effectively-scheduled* instance is that an item appears on the calendar in the proper place, but does not increase the physical size of the calendar. The issue of actual versus effective calendar size is discussed further in the next section.

For recurring items, the system keeps track of instance collections as a whole, for the purposes of change and delete. Each instance that is generated from a particular recurring item is considered to be 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.

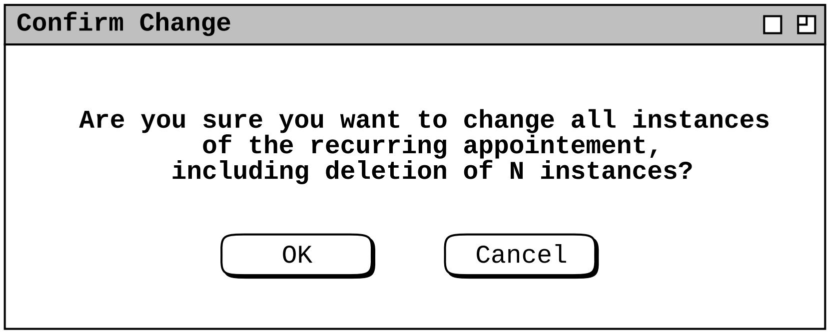
When the user changes the start date, end date, or recurring details for all instances, the effect can be to delete one or more instances. Specifically, the following forms of change result in instance deletion:

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 189, 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 any item-level displays for the changed instance if the change results in its deletion.



Are you sure you want to change all instances of the recurring appointement, including deletion of N instances?

Figure 189: Confirmation dialog for deleting instances by date or recurring details changes.

2.5.3.6. The Effect of Recurring Items on Calendar Size

The idea of *effectively-scheduled instances* warrants additional clarification with regards to how it effects the measure of a calendar's size, in terms of the number of items the calendar contains. Since a recurring item need not have an end-date, the number of effectively-scheduled items in a calendar can be unbounded. Clearly, a calendar of genuinely unbounded size is not feasible in any implementation.

For the purposes of specifying practical bounds on calendar size, there are the following two definitions:

- o *actual size* -- the number of items that are physically stored in the calendar, independent of any of its views
- o *effective size* -- the number of items that are conceptually stored in the calendar, such that the items appear as they should in any and all calendar views displayed on the screen

The meaning of "*appear as they should*" is that all effectively-scheduled instances appear whenever the user views a calendar with dates where recurring instances are scheduled.

The actual size of a calendar is defined by the following specific formula:

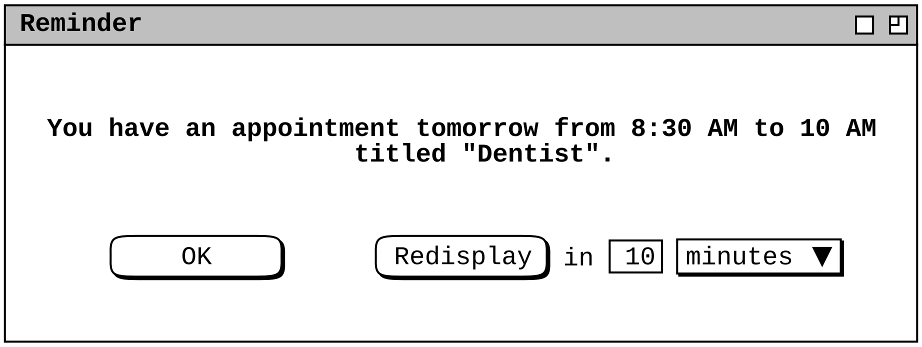
number of non-recurring items +
number of recurring base items +
number of changed recurring instances

The number of non-recurring items is the total number of such items scheduled and not deleted. A recurring *base item* is a single calendar item that represents all of the unchanged instances in the recurring collection. Any time the user changes one item or "all future" items, the Calendar Tool creates a new actual item. In the case of a single-item change, one new changed instance is created. In the case of future instances being changed, the Calendar Tool creates a new recurring base item to represent all future changed recurring instances. In either case, a single change results in an increase of one item in actual size for the calendar in which the user makes the change.

The end user may not be concerned with the exact details of actual versus effective calendar size, other than to be aware that the actual size is clearly bounded. For the implementors, the requirements about actual versus effective size mean that the actual size cannot grow proportional to the number of unchanged recurring items. I.e., when the user adds a single recurring item, the actual size of a calendar increases by one, independent of how many effectively-scheduled instances the recurring item represents.

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 local 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 190. 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



You have an appointment tomorrow from 8:30 AM to 10 AM titled "Dentist".

Figure 190: Dentist appointment on-screen reminder.

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.

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 190, 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 redisplays 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 the 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 191. If the category '`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 192. When the user selects '`Edit ...`' in the categories menu, the system displays the dialog of Figure 193. 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 191 through 193 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 194 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 195. The

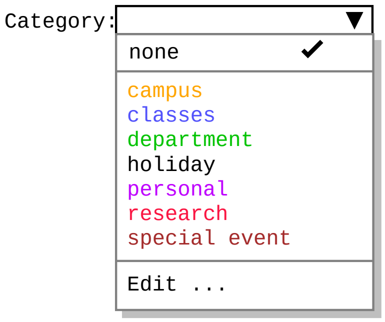


Figure 191: Category display menu.

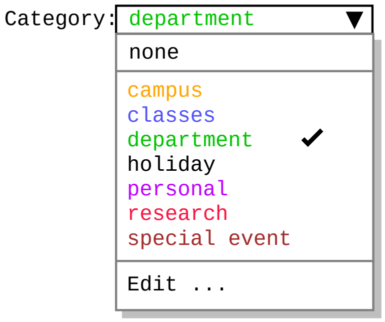


Figure 192: Category menu with non-empty category selected.

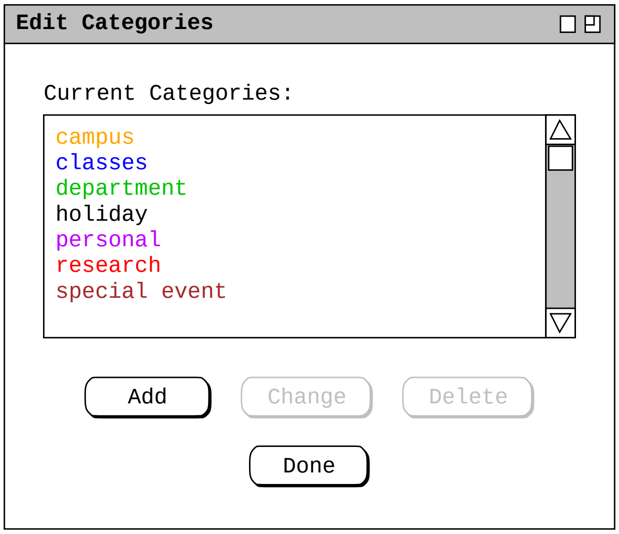


Figure 193: Edit categories dialog with several user-defined categories.

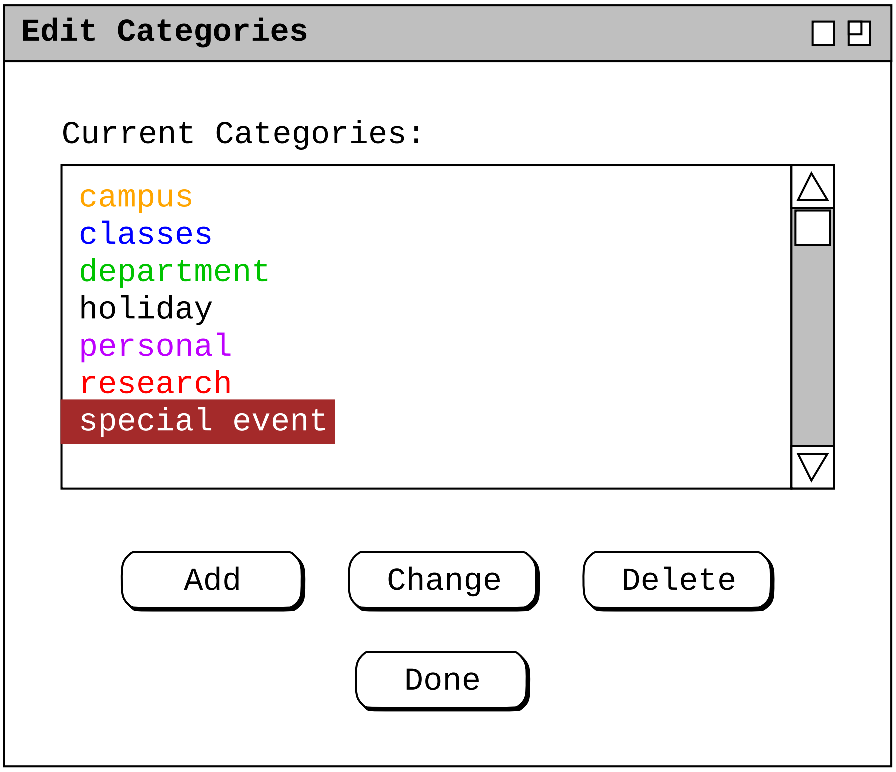
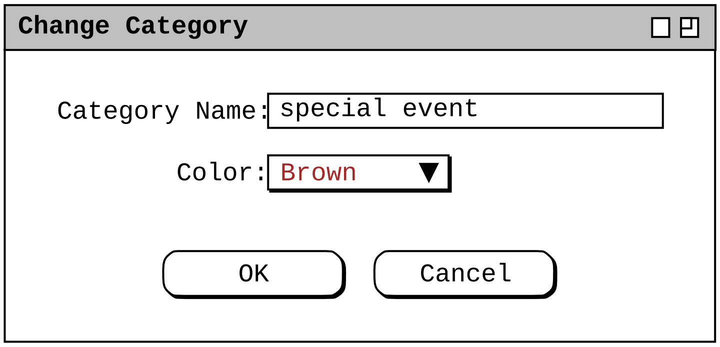


Figure 194: Special event category selected for editing.



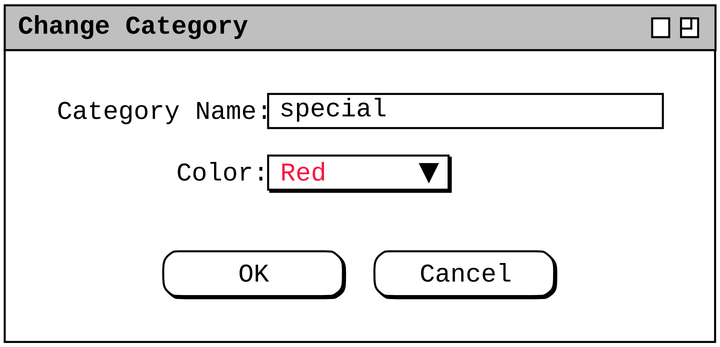
Category Name: special event

Color: Brown

Figure 195: Change category dialog.

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 196, 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 197. 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.

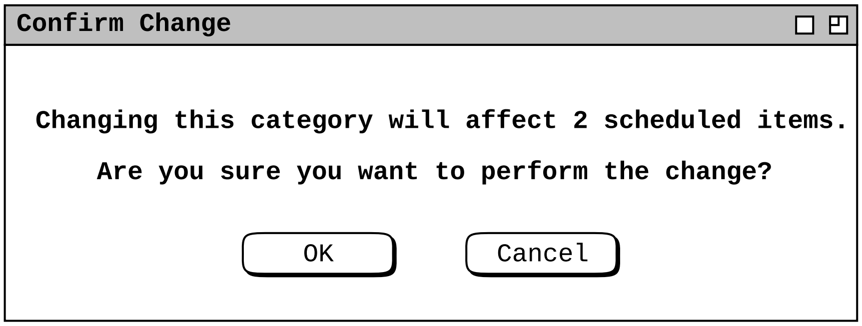
To delete an existing category, the user selects its name and presses '`Delete`'. In response, the system displays a dialog of the form shown in Figure 198. Here the user has chosen to delete a category that is used in 15 scheduled items. As the dialog explains, the affected items will all have their categories set to '`none`' if the delete is



Category Name: special

Color: Red

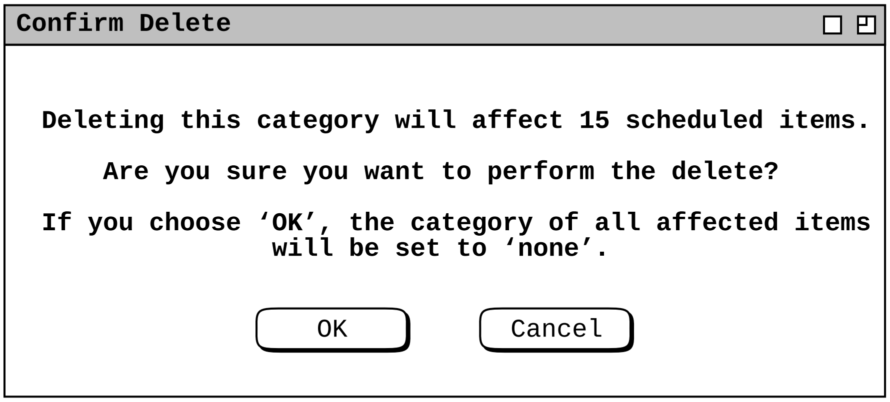
Figure 196: Change category dialog, edited.



Changing this category will affect 2 scheduled items.

Are you sure you want to perform the change?

Figure 197: Category change confirmation dialog.



Deleting this category will affect 15 scheduled items.

Are you sure you want to perform the delete?

If you choose ‘OK’, the category of all affected items will be set to ‘none’.

Figure 198: Category delete confirmation dialog.

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.

If a category to be deleted is not used in any scheduled items, then the delete confirmation dialog in Figure 198 does not appear. Rather, the system proceeds to delete the category immediately, without any confirmation.

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, and who are connected to the central host computer from which the notification originates. If user option settings or host connection state limit when the notification dialog can appear, it appears as soon as the option settings and connection allow or are changed to allow. When a user hides penciled-in items, such items appear immediately after the user changes the appropriate setting to allow them to appear.

Display updates to other user and group calendars do not appear dynamically. For example, if user *A* is viewing a calendar of user *B*, changes made by user *B* to that calendar 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 adminstrative user's display of the Calendar Tool Adminstration program. Also, any administrator-generated notifications appear dynaically on the affected users' displays. Adminstrative notifications are generated for adding, changing, and deleting user records, and by the '`Notify Users`' command.

Aside from administrative notifications, changes to adminstrative data do *not* appear dynamically on affected users' screens. For example, if a user is viewing a record in the user database, changes made by an admistrator 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 adminstrator 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 administrative commands for configuring and connecting to Calendar Tool central host computers, and for performing other managerial functions.

As described in Section 2.1, there are two separate interfaces to administrative commands, provided by two separate application programs. Regular users access administrative commands through the regular Calendar Tool interface, shown in Figures 1 and 2. Users with administrative privilege can run the Calendar Tool Administration program, through the interface shown Figures 3 and 4.

Calendar Tool databases are physically stored on designated central host computers. Administrative users run the Calendar Tool Administration program on a central host. Regular users run the Calendar Tool program on their local client computers and connect to a central host via a computer network. A user's local computer may be physically the same machine as a central host, but logically the function of local computer and central host are distinct.

The sole purpose of the Calendar Administration program is to manage the databases and communication services on a central host. The administration program provides no commands for calendar scheduling or viewing. Any computer on which the Calendar Tool Administration program is installed can function as a central host, as long as that computer has proper network access to the client computers of Calendar Tool users.

The purpose of administrative commands for the regular user is to connect to central hosts, view the Calendar Tool databases, and perform other limited administrative functions. Since the database repository resides on a central host computer, the regular user must establish a central host connection in order to have access to administrative data. Details of establishing host connections are covered in Section 2.6.6.1. Before a host connection is established, only the '`Connect`' and '`Contact Admin`' commands are enabled in the regular–user `Admin` menu. Once a connection is established, the other commands become enabled.

In the scenarios to follow, use of the Calendar Tool Administration program is covered in Sections 2.6.1 through 2.6.5. Access to administrative commands by regular Calendar Tool users is covered in Section 2.6.6. Further details on the Calendar Tool multi–user environment, consisting of local clients and central hosts, are covered in Section .

2.6.1. Gaining Administrative Access

When a user invokes the Calendar Tool Administration program, the system presents the password dialog shown in Figure 199. The user must enter the current administrator password in order to be confirmed as an administrator. If the user does so, 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 200. 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.

At any given time, there is a single legal administrative password for a particular installation of the Calendar Tool Administration program. Given this, the definition of "administrative privileges" is simply the knowledge of the

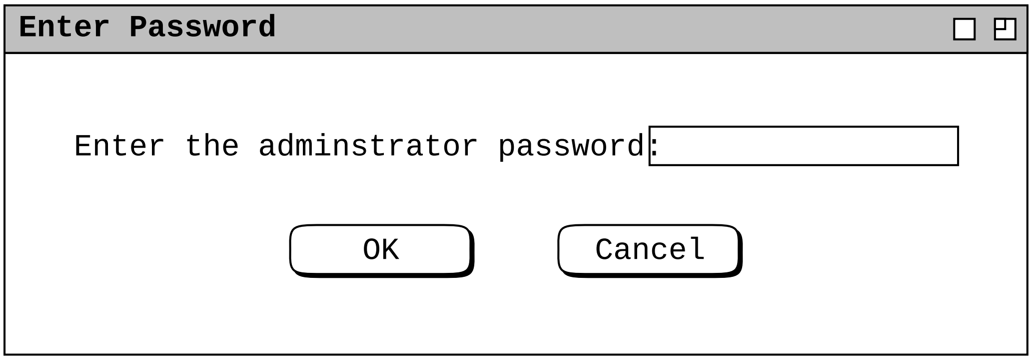
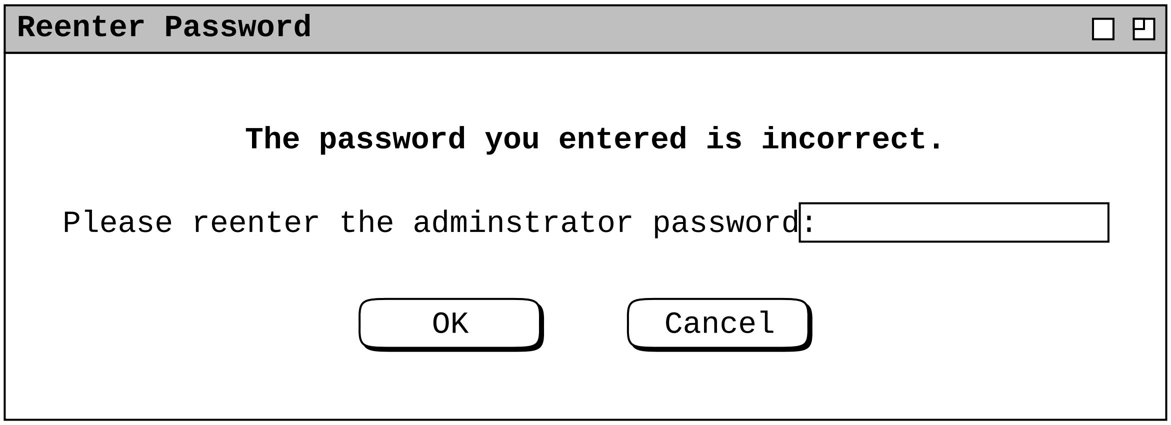


Figure 199: Password entry dialog.



The password you entered is incorrect.

Please reenter the adminstrator password:

Figure 200: Password re-entry dialog.

current administrative password. Section covers installation procedures, including establishing the initial administrative password.

Whenever an administrator invokes the Calendar Tool Administration program, the computer on which the invocation takes place is considered to be the Calendar Tool central host for the purposes of all subsequent administrative commands. The name of the host computer is displayed in the banner of the Calendar Tool Administration menubar, as shown in Figures 3 and 4.

2.6.2. User Database

As shown in Figure 3, the default administrative interface consists of a menubar and a dialog for accessing the Calendar Tool user database. The same dialog is displayed when the administrator selects the '`Users ...`' item in the '`Admin`' menu. Figure 201 shows the state of the user database after a number of user records have been added.CSC 308 Midterm The scrolling area and search field operate in the same manner as described in Section 2.3.5 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 202. 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 201 are formatted as follows:

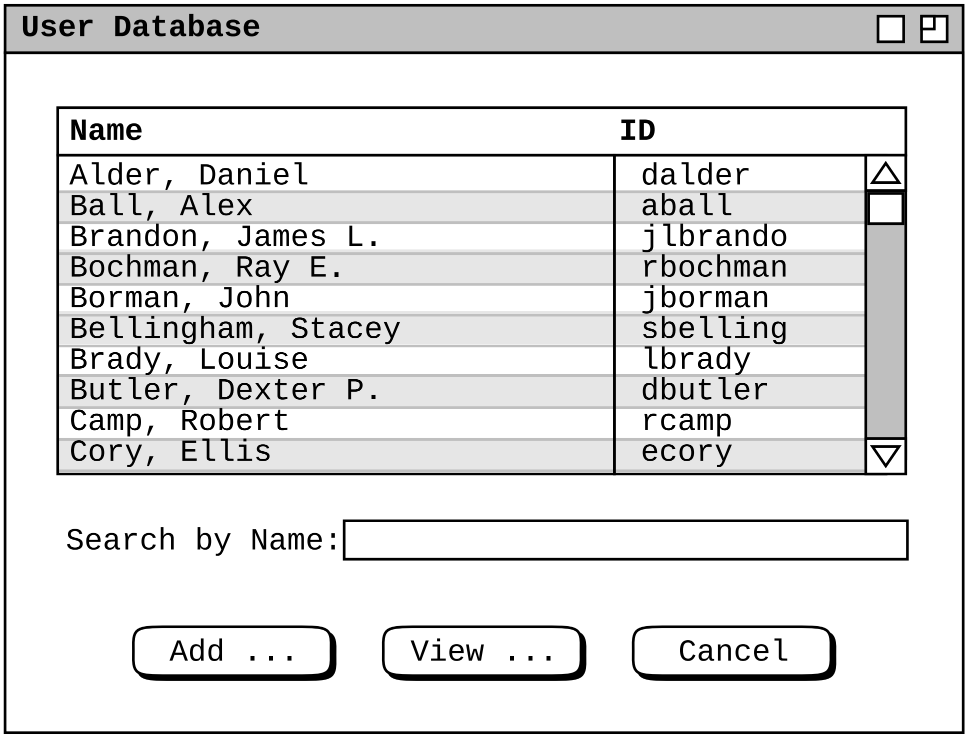
last, *first middle*

The comma and spaces are absent if there are no first and middle names.

The '`Calendar Tool ID`' is the identifier by which a user is known in the Calendar Tool system served by a particular host. The ID must be unique among all Calendar Tool users and groups on that host. That is, no two or more users can have the same ID, and no user can have the same ID as any user group. The '`Password`' is required along with the ID when the user establishes a connection to the central host. The password is encrypted using a suitably secure encryption mechanism.

The '`Email`' address is optional, but highly recommended. It is the electronic mail address to which meeting notifications and other general notifications are sent. The phone number is also optional; it can be provided as an additional means for communicating with a Calendar Tool user, outside of the context of the Calendar Tool itself.

The '`Calendar Size Limit`' specifies the maximum size in megabytes of the user's calendar on the Calendar Tool central host. Its value is a real number. The limit value is optional. If it is absent, there is no limit. The size limit can be zero, in which case the user has an empty, and therefore unusable calendar on the central host. Complete details on central host data storage are covered in Section .



Name	ID
Alder, Daniel	dalder
Ball, Alex	aball
Brandon, James L.	jlbrando
Bochman, Ray E.	rbochman
Borman, John	jborman
Bellingham, Stacey	sbelling
Brady, Louise	lbrady
Butler, Dexter P.	dbutler
Camp, Robert	rcamp
Cory, Ellis	ecory

Figure 201: User database dialog.

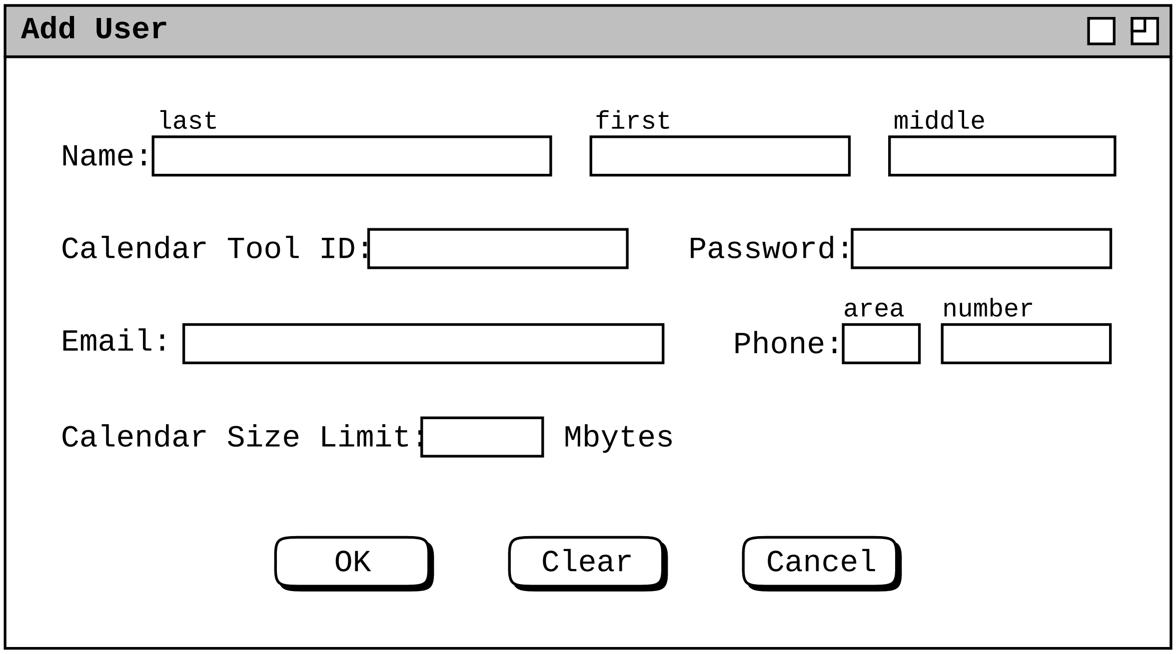


Figure 202: Add user dialog.

Figure 203 shows the administrator having entered values for a new user. The password is echoed as typed in the add–user dialog, in contrast to the user editing dialog, where it is shown only as asterisk characters. When the administrator presses '`OK`' in the add–user dialog, the system performs the following actions:

a. validates that the Calendar Tool ID is unique among all extant users and groups;

b. validates the email address insofar as is possible (see Section for details);

c. adds the new user record to the user database, if the validation checks succeed

d. updates the list of names in the user database dialog

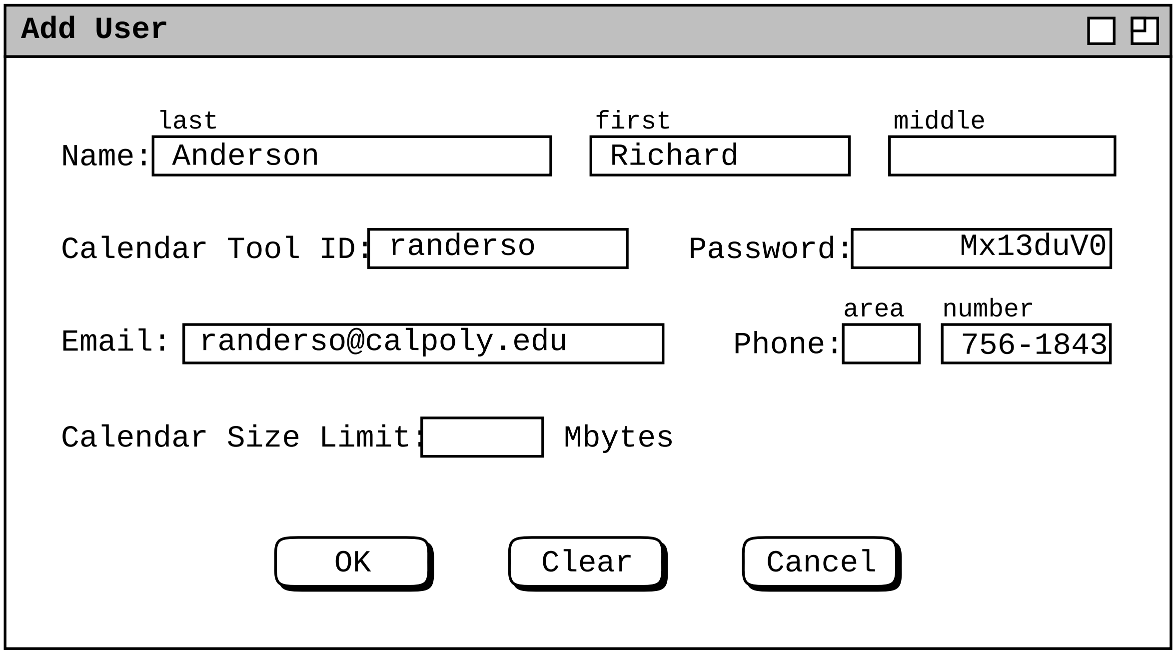
e. adds the new user's ID to the `All Users` group, as described further in Section 2.6.3.

f. notifies the new user of the addition via the electronic mail address in the user record; if an email address is not provided, then the administrator must notify the user via some form of communication outside of the Calendar Tool.

The new–user notification message is of the form shown in Figure 204. 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 205 shows the administrator having selected James L. Brandon for view. Only one user can be selected at a time. When the administrator presses '`View ...`' in Figure 205, the system displays the dialog of Figure 206. 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 data fields present in the add–user dialog. 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 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.



Name: last Anderson first Richard middle

Calendar Tool ID: randerso

Password: Mx13duV0

Email: randerso@calpoly.edu

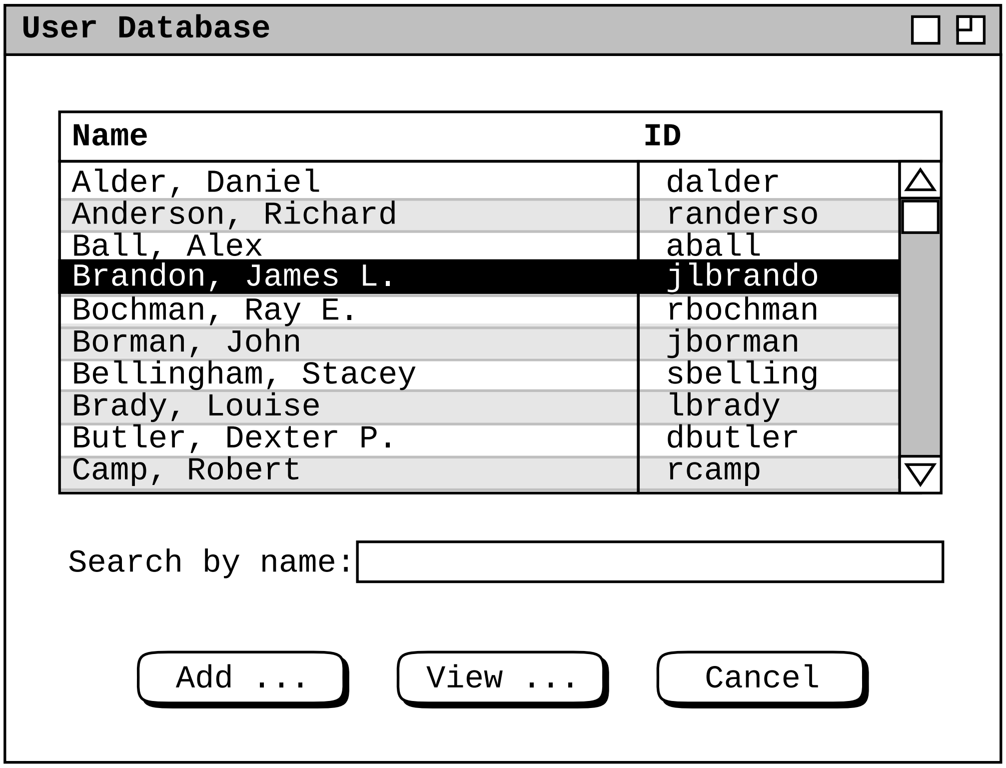
Phone: area number 756-1843

Calendar Size Limit: Mbytes

Figure 203: Adding a new user.

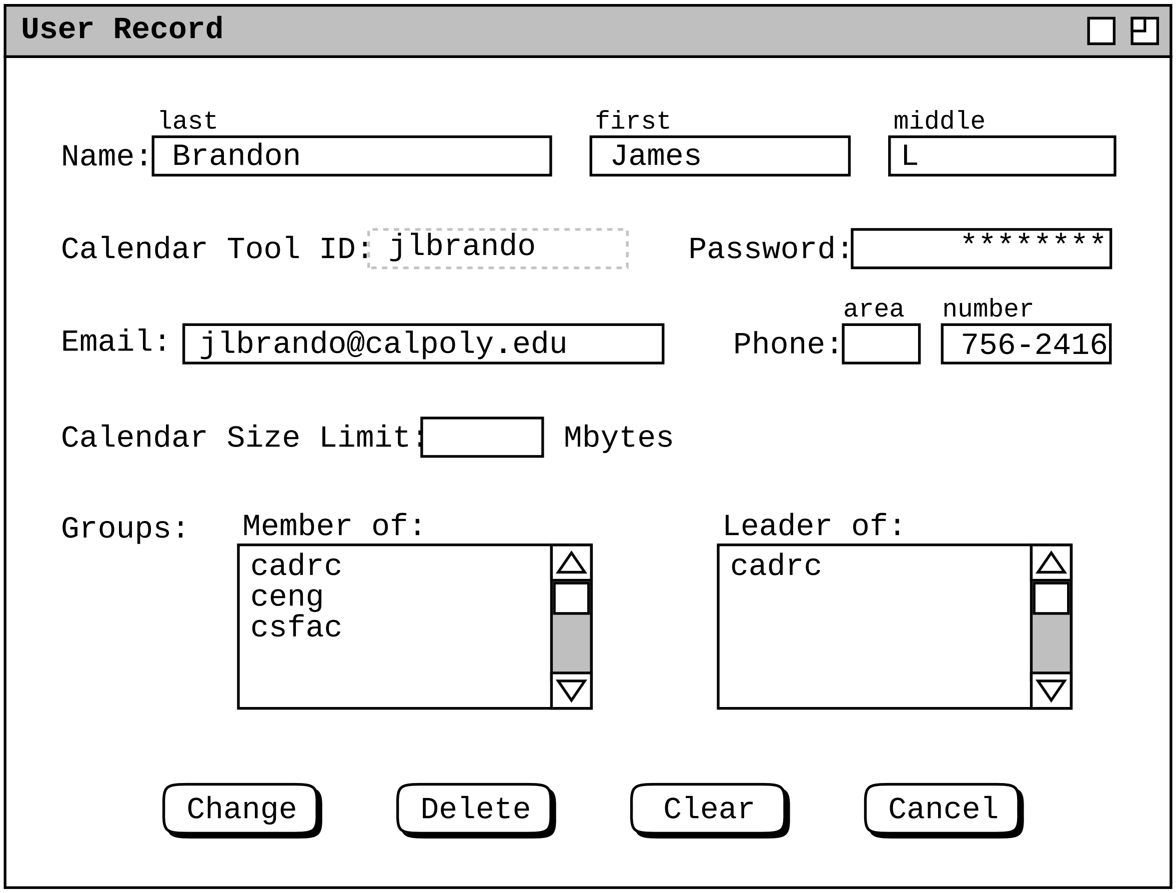
```
You have been added as a new registered user on the Calendar Tool
central host computer deptsrv.csc.calpoly.edu.  Your registration
information is as follows:

    Name: Richard Anderson
    ID: randerso
    Password: Mx13duV0

To view your complete user record, connect to this host using the
Calendar Tool ‘Admin->Connect’ command, then select the ‘Admin->Users’
command, and then select your name or ID in the list of registered
users.  As soon as you establish the initial connection, it is highly
recommended that you change your initial password using the
‘Admin->Password’ command.
```

Figure 204: New user notification message.



Name	ID
Alder, Daniel	dalder
Anderson, Richard	randerso
Ball, Alex	aball
Brandon, James L.	jlbrando
Bochman, Ray E.	rbochman
Borman, John	jborman
Bellingham, Stacey	sbelling
Brady, Louise	lbrady
Butler, Dexter P.	dbutler
Camp, Robert	rcamp

Figure 205: Brandon selected in users list.



Name: last Brandon first James middle L

Calendar Tool ID: jlbrando

Password: ********

Email: jlbrando@calpoly.edu

Phone: area number 756-2416

Calendar Size Limit: Mbytes

Groups: Member of: cadrc
ceng
csfac

Leader of: cadrc

Figure 206: User database record for James L. Brandon.

Figure 207 shows the results of the administrator having performed edits to the user record for James L. Brandon. The edits are to the ’`Email`’, ‘`Phone`’, and ‘`Password`’. When the administrator changes the `Password`, the new password is echoed as the actual characters typed, not as asterisks. After the administrator confirms the change, the password is subsequently displayed as asterisks.

When the administrator presses ‘`Change`’, the system displays the confirmation dialog in Figure 208. The administrator confirms the change with ‘`OK`’ or cancels it with ‘`Cancel`’. Upon confirmation, the system performs the following actions:

a. if the email has changed, validates the address insofar as is possible (see Section)

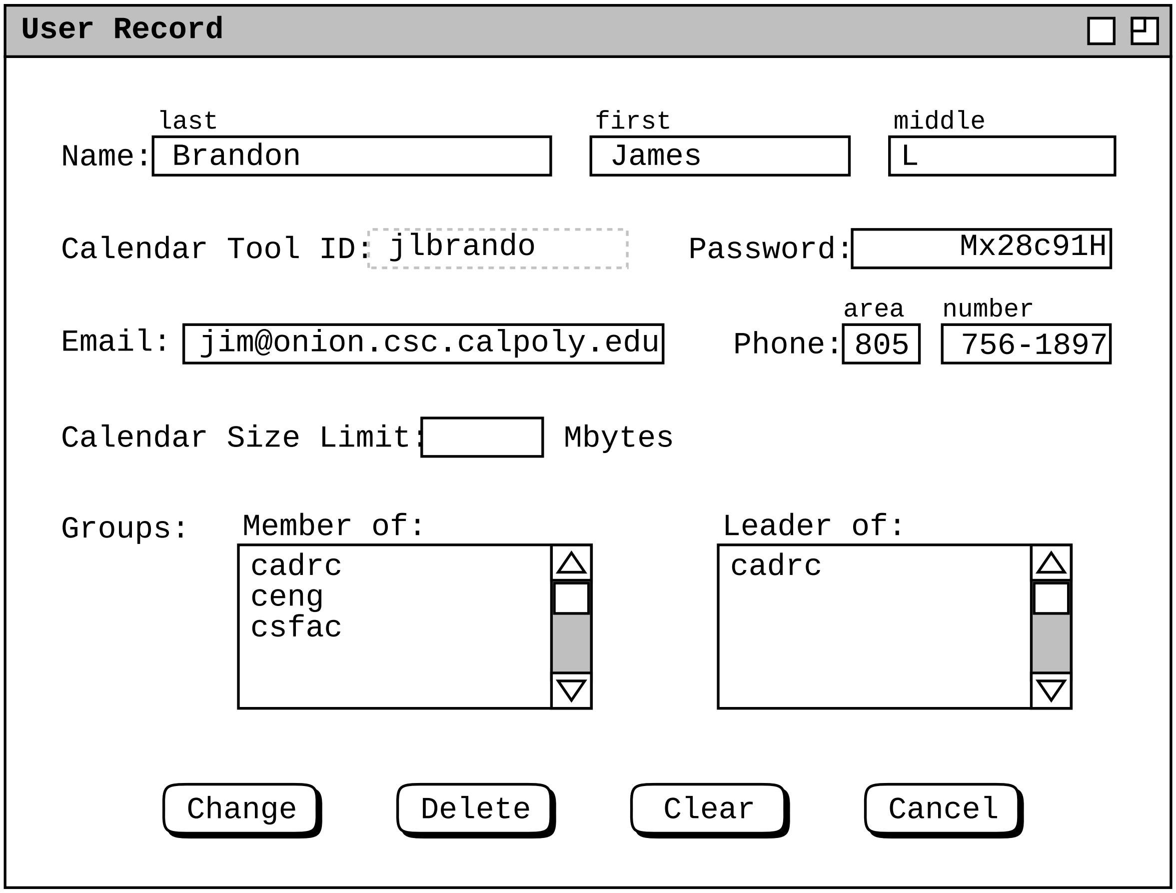
b. updates the user record in the database

c. notifies the affected user of the change via the electronic mail address in the user record; if an email address is not provided, then the administrator must notify the user via some form of communication outside of the Calendar Tool.

d. if the size limit has changed, deletes items from the user’s calendar on the central host repository, if necessary, to satisfy a changed size limit value; Section Section has further details on central repository size limits

e. if the password has been changed, redisplays its value as asterisks characters in the user record dialog

The user notification for the change confirmed in Figure 208 is shown in Figure 209. The notification message lists the specific details of the change.



User Record

Name: last Brandon first James middle L

Calendar Tool ID: jlbrando

Password: Mx28c91H

Email: jim@onion.csc.calpoly.edu

Phone: area 805 number 756-1897

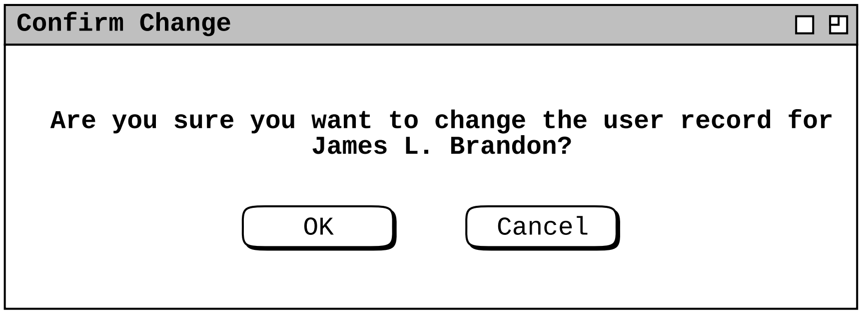
Calendar Size Limit: Mbytes

Groups:

Member of:
cadrc
ceng
csfac

Leader of:
cadrc

Figure 207: User record for James L. Brandon edited.



Confirm Change

Are you sure you want to change the user record for James L. Brandon?

Figure 208: Confirmation dialog for changing a user record.

```
The following information has been changed in your Calendar Tool user
record on central host deptsrv.csc.calpoly.edu:

    Password: from previous value
              to Mx28c91H

    Email: from jlbrando@calpoly.edu
           to   jim@onion.csc.calpoly.edu

    Phone: from 756-2416
           to   805 75601897

If you have questions about these changes, please contact a Calendar
Tool administrator on this host using the ‘Contact Admin’ command.
```

Figure 209: Notification of change to James L. Brandon user record.

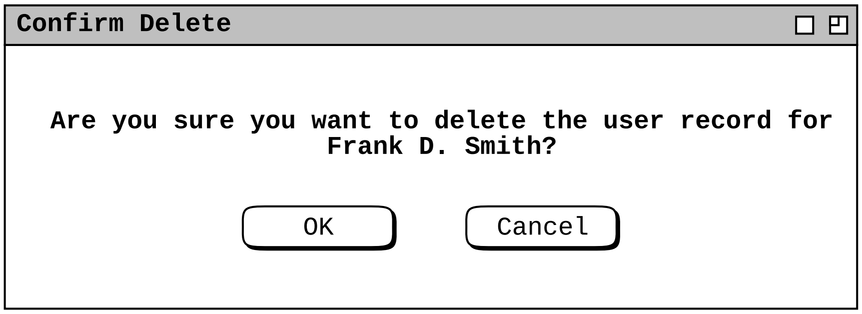
To delete a user, the administrator presses ‘`Delete`’ in the user record display for the user to be deleted. In response, the system displays a confirmation dialog of the form shown in Figure 210. When the administrator confirms the deletion with ‘`OK`’, the system performs the following actions:

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
c. removes the user’s calendar from the central host repository
d. cancels any meeting scheduling operation that the deleted user is executing at the time the deletion is confirmed
e. notifies the user of the deletion via the electronic mail address in the user record; if an email address is not provided, then the administrator must notify the user via some form of communication outside of the Calendar Tool.

The notification message is of the form shown in Figure 211. This is the notification sent to user Frank D. Smith. The notification text explains the delete action.

Deleting a user does not delete the user’s ID from any meeting attendees lists on which it appears. Meeting attendee lists can be manually edited, as described in Section 2.5.2.2.

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.



Are you sure you want to delete the user record for Frank D. Smith?

Figure 210: Confirmation dialog for deleting a user record.

```
You have been deleted as a registered user of the Calendar Tool system
on host deptsrv.csc.calpoly.edu.

You can continue to use the Calendar Tool to manage your own local
calendars, and to manage Calendars on any other central hosts where
you are registered.

If you have questions about this action, please contact a Calendar
Tool administrator.
```

Figure 211: Notification sent to deleted user.

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.

It is worth noting that all the notifications for user database operations are sent via email, not via on–screen dialogs. The reason is that in each case, the user cannot or may not be able to view an on–screen notification, because in order to do so the user must be connected to the central host from which the notification originates. In the case of addition, the user does not yet know the user ID that must be provided to establish a connection to the originating host. In the case of a change, the notification will not be deliverable on–screen if the action is a password change. In the case of delete, the user's ability to establish a connection to the originating host has been terminated, which again makes the user unable to see an on–screen notification originating from that host.

2.6.3. Group Database

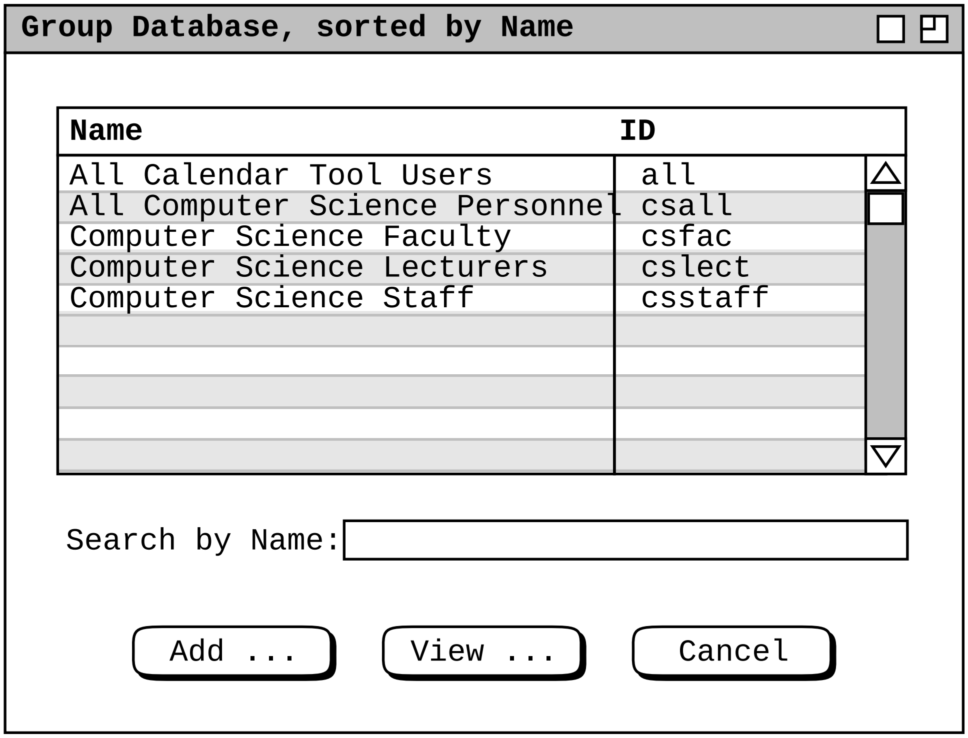
User groups are a convenient means to define a named list of related users. In comparison to an individual user, a group can be considered a virtual entity. No single user owns the group; a group has no email address and no password. As explained in Section 2.4.1, meetings scheduled by group leaders 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 212. This dialog has the same basic format as that for the user database in Figure 201. The '`Search by`' field operates in the same manner as in the user database dialogs. One group is built–in to the Calendar Tool: the `all` group consisting of all registered users.

To add a new group, the administrator presses '`Add ...`', whereupon the system displays the dialog of Figure 213. 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. That is, no two or more groups can have the same ID, and no group can have the same ID as any user. The '`Size Limit`' specifies the maximum size in megabytes of the group calendar in the central host repository.

The '`Members`' and '`Leaders`' lists contain the 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, i.e., administrators may delete dormant and leaderless groups if they so choose. A group leader need not be a member. This allows a leader to schedule a meeting at which the leader is not an attendee.

Figure 214 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 215. 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 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 The administrator can view the list of all



Name	ID
All Calendar Tool Users	all
All Computer Science Personnel	csall
Computer Science Faculty	csfac
Computer Science Lecturers	cslect
Computer Science Staff	csstaff

Search by Name:

Figure 212: Group database dialog.

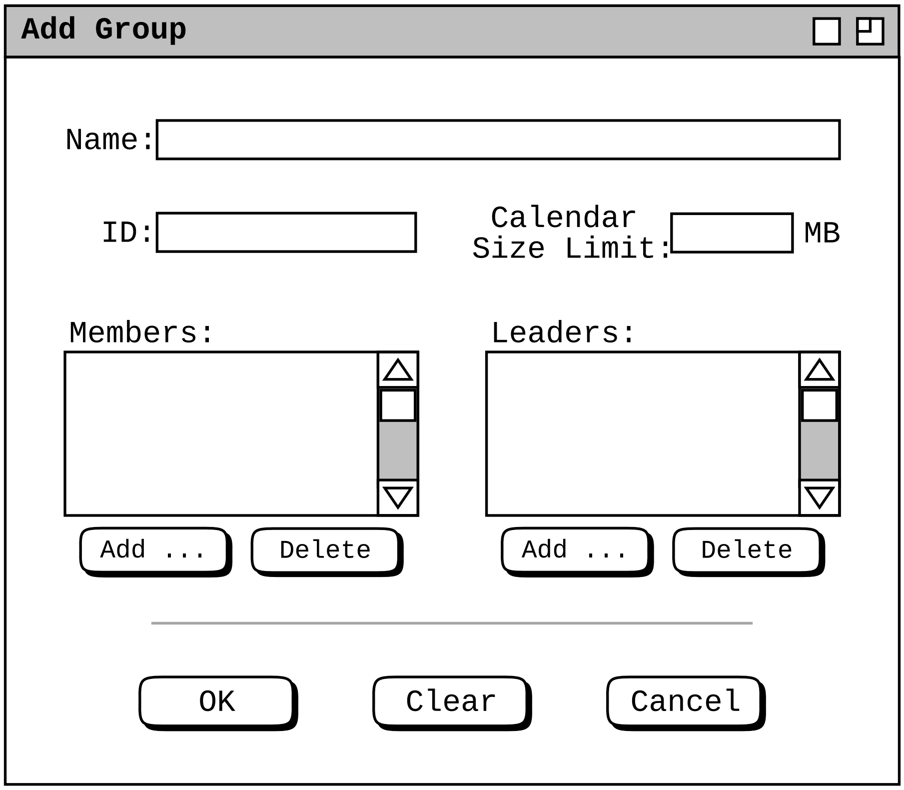
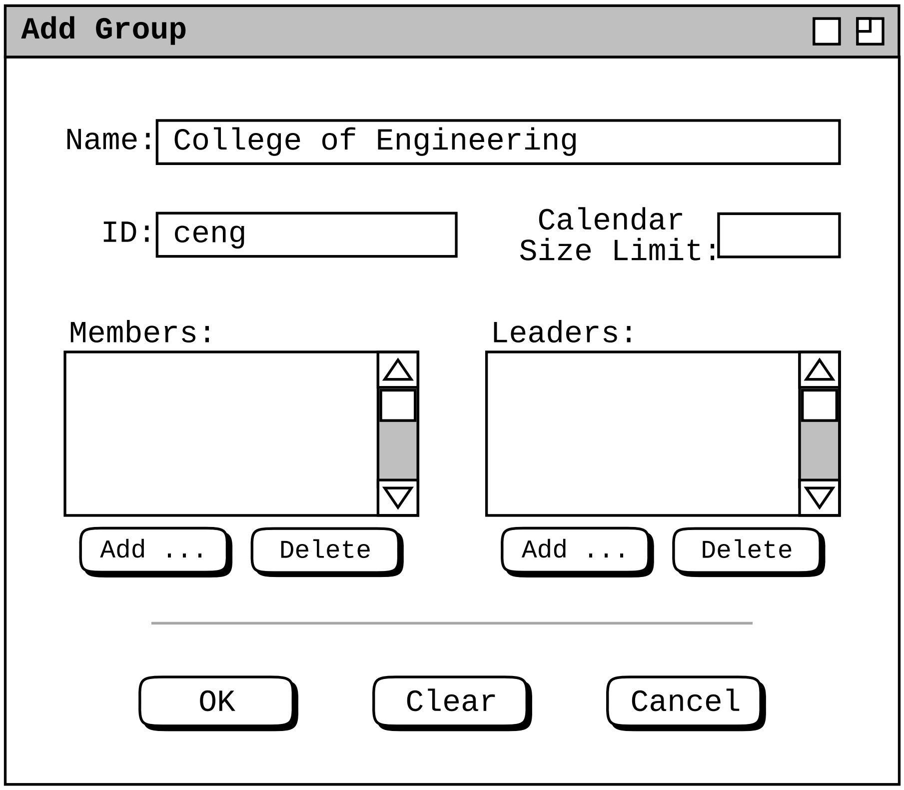


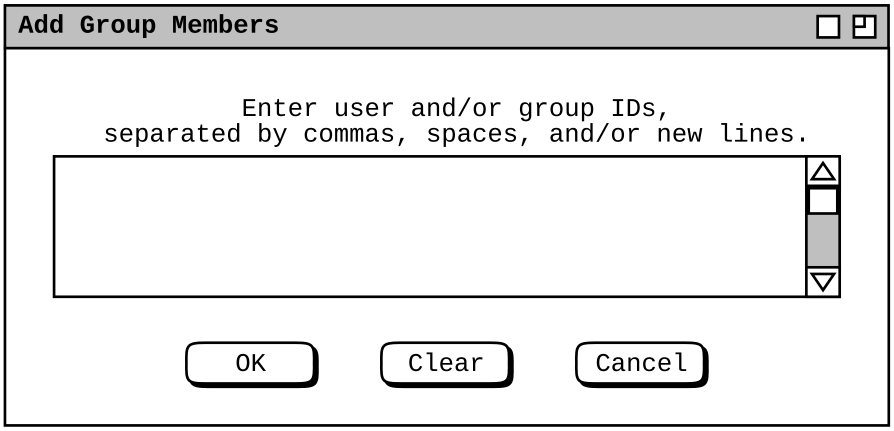
Figure 213: Add group dialog.



Name: College of Engineering

ID: ceng

Figure 214: Adding a new group.



Enter user and/or group IDs,
separated by commas, spaces, and/or new lines.

Figure 215: Add group member dialog.

available user and group IDs in the database dialogs of Figures 201 and 212, respectively. 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.

Duplicate IDs are ignored by the system. The precise definition of group membership is the set of entered IDs, with duplicate IDs considered to be a single member. The rationale for allowing duplicates in the members list is to avoid complexity when entering (super)group members, as described just below.

Figure 216 shows the administrator having entered IDs for the `ceng` group. The first row of IDs are 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 *super group*. There is no limit on the depth of the super group hierarchy. That is, a super group can have one or more super groups 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.

Circularity is disallowed in supergroup definition. That is, a group may not contain itself as a member, either directly or indirectly.

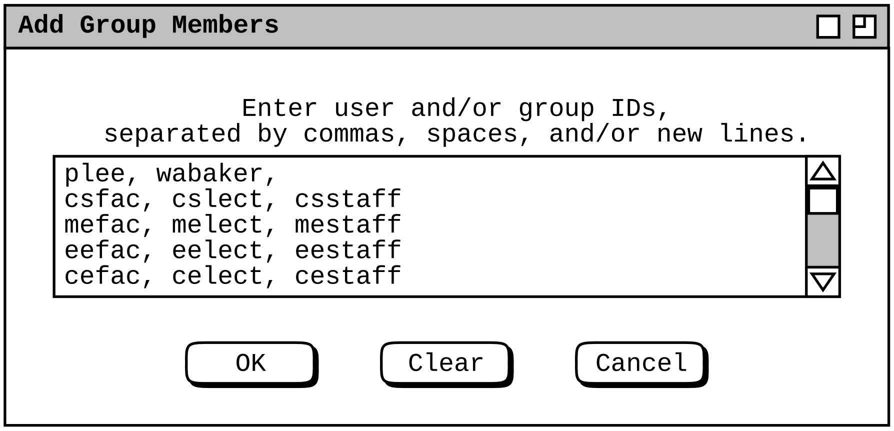
When the user presses '`OK`' in Figure 215, the system validates all entered IDs and checks for any circularity in a supergroup definition. If one or more IDs are invalid or a circularity is detected, the system issues an appropriate error message as described in Section . If all IDs are valid, the system 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 217 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.

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 212 and then presses the '`View ...`' button. For example, Figure 218 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 213.

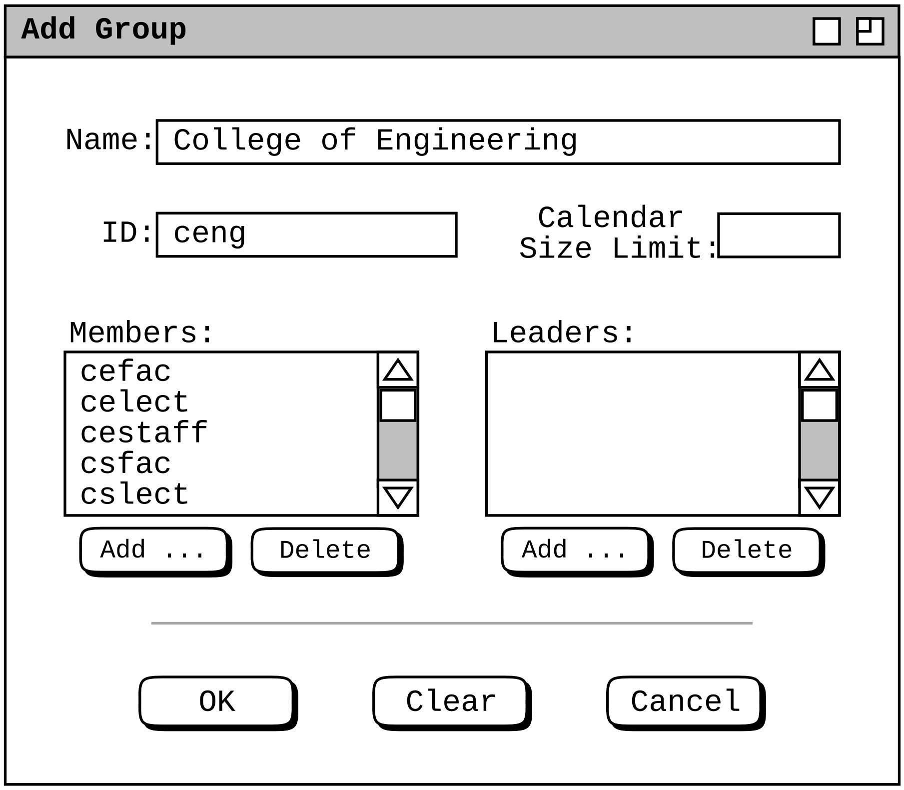
To delete one or more group members, the administrator selects the name(s) in the '`Members`' list and presses the '`Delete`' button immediately below the list. To select multiple names for deletion, the user holds down the `SHIFT` key as each name is selected in the list. In response to executing '`Members: Delete`', the system displays a confirmation dialog of the form shown in Figure 219. 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 confirming the delete, the system removes the names from the members list, as shown in Figure 220. Since the members list is longer than fits in the scrolling area, only the deletion of '`cdaniels`' is immediately visible.



Enter user and/or group IDs,
separated by commas, spaces, and/or new lines.

```
plee, wabaker,
csfac, cslect, csstaff
mefac, melect, mestaff
eefac, eelect, eestaff
cefac, celect, cestaff
```

Figure 216: Add group member dialog filled in.



Add Group

Name: College of Engineering

ID: ceng

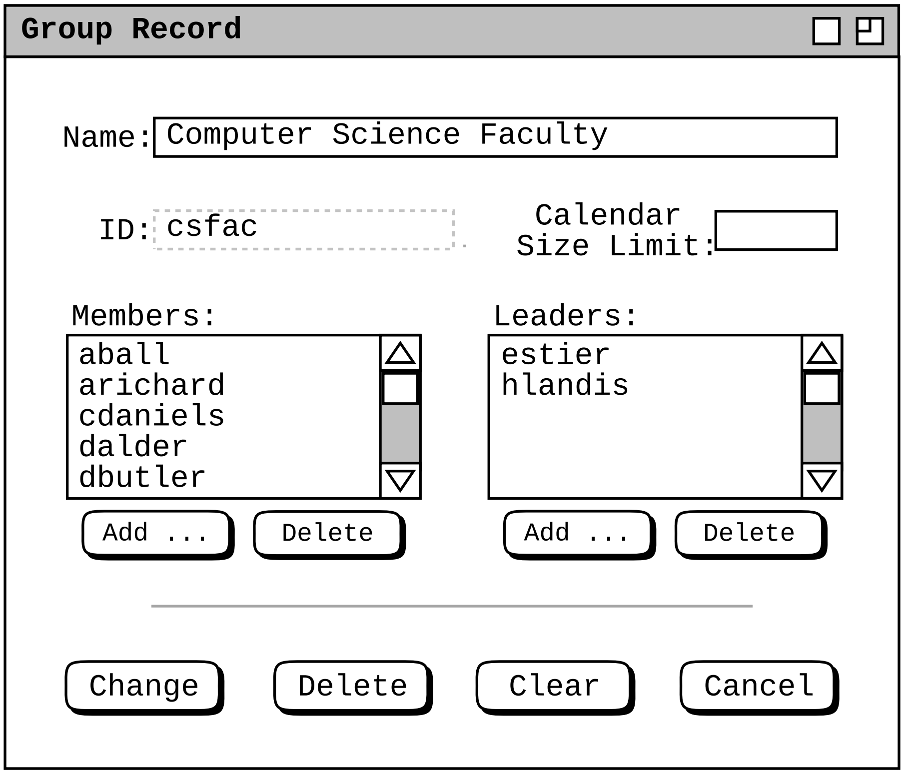
Calendar Size Limit:

Members:

cefac
celect
cestaff
csfac
cslect

Leaders:

Figure 217: Add group dialog with members added.



Group Record

Name: Computer Science Faculty

ID: csfac

Calendar Size Limit:

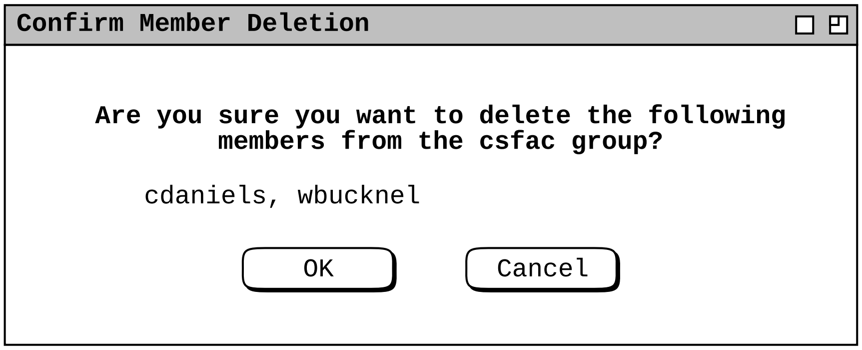
Members:

aball
arichard
cdaniels
dalder
dbutler

Leaders:

estier
hlandis

Figure 218: Group database record for the Computer Science Faculty.

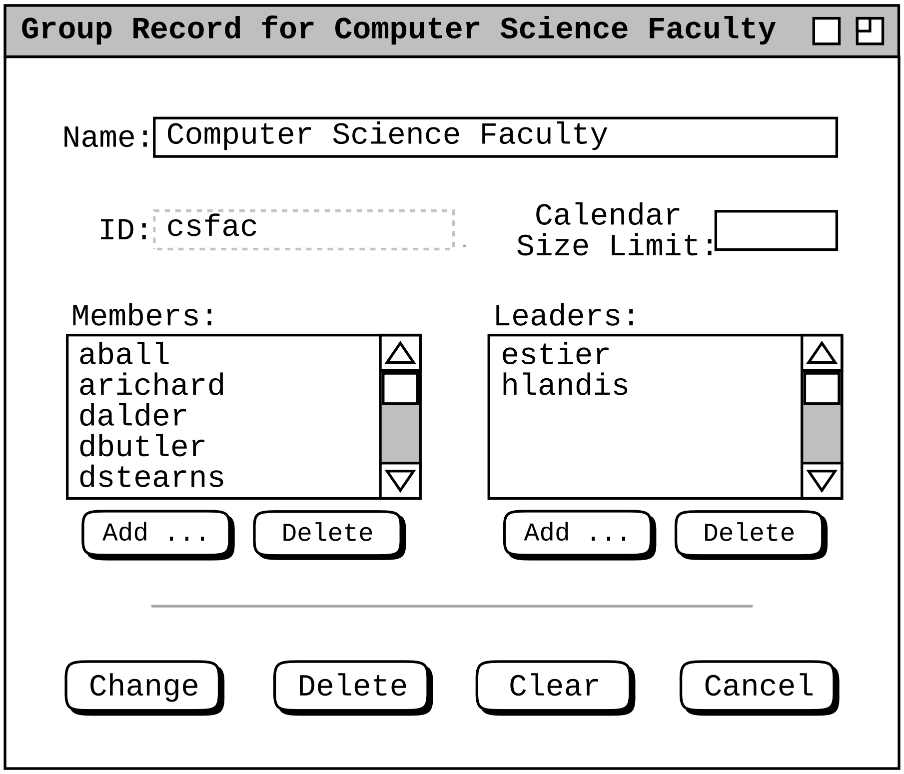


Confirm Member Deletion

Are you sure you want to delete the following members from the csfac group?

cdaniels, wbucknel

Figure 219: Group member delete confirmation.



Name: Computer Science Faculty

ID: csfac

Calendar Size Limit:

Members:
aball
arichard
dalder
dbutler
dstearns

Leaders:
estier
hlandis

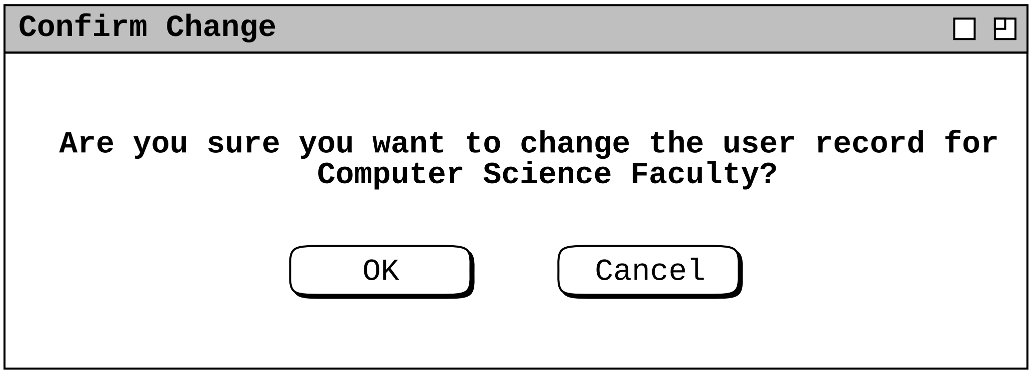
Figure 220: Members deleted in group record dialog.

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 the 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 221. 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



Are you sure you want to change the user record for Computer Science Faculty?

Figure 221: Confirmation dialog for changing a group record.

c. notifies all added or deleted users of the change, via an on–screen dialog

d. notifies all added or deleted leaders of the change, via an on–screen dialog

If the same user is added as both a member and leader, only one notification is sent. The notification takes the following form:

`You have been` *actioned*
`the` *group* **`Calendar Tool group.`**

`If you have questions about this`
`please contact the Calendar Tool administrator at`
central host

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 *central host* string is the name of the Calendar Tool central host computer on which the change or delete operation is performed, i.e., the central host on which the administrator is current running Calendar Tool Administration.

When the administrator executes '`Delete`' for a group record, the system displays the same form of confirmation dialog as shown in Figure 221, 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, i.e., all users who were a member and/or leader of the deleted group

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`**"

It is important to note that 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 specific commands for a user to request addition to or removal from a group. Such requests can be made using the Calendar Tool '`Contact Admin`' command, or by some means outside of the Calendar Tool.

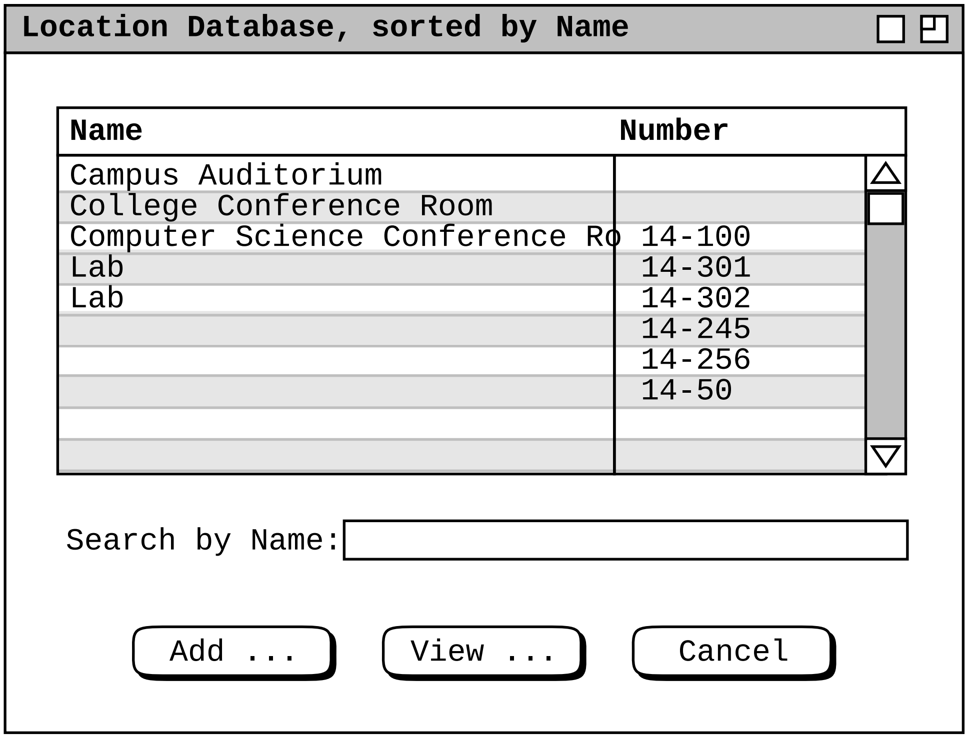
2.6.4. Location Database

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 a dialog of the form shown in Figure 222. The figure shows the state of the location database after eight records have been added. The location database dialog has a similar format to the dialogs for the user and group databases. 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 unique, the secondary sorting order is by number when name is the primary sort key.

Either the name or location may be empty, but not both. When items are sorted by name, all locations with empty names appear at the end of the list. This is illustrated in Figure 222 by the appearance of the empty-named locations at the end of the list. When items are sorted by number, all locations with empty numbers appear at the end of the list. For example, Figure 223 shows the result of the administrator having clicked on the '`Number`' column heading in the dialog of Figure 222.

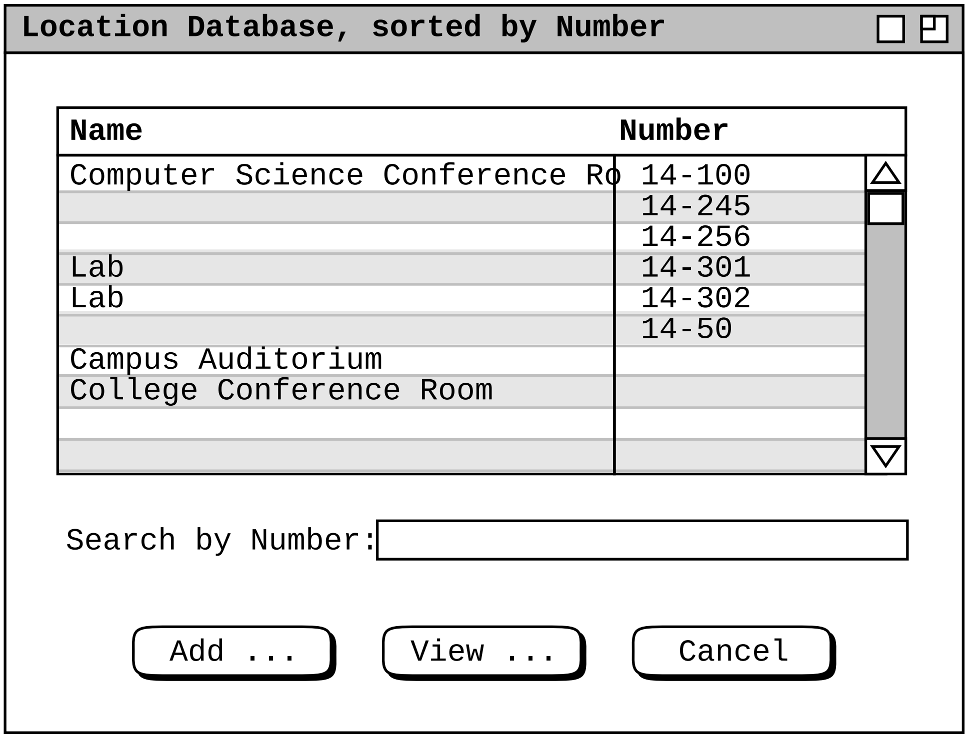
While the '`Number`' field may typically contain numeric components, it is treated as string value for the purposes of sorting. For example, the '`Number`' value "14-50" lexically follows the value "14-256", even though the numeric value `50` precedes the value `256`. This ordering is illustrated in both Figures 222 and 223 by the placement of the location numbered "`14-50`" relatively after other locations with non-empty numbers. Full details of lexical sorting are covered in Section .

The '`Search by`' field in the location database dialog operates in the same manner as in the user and group database dialogs. Since the search function requires typing, it is not possible to search for a location by an empty name or number.



Name	Number
Campus Auditorium	
College Conference Room	
Computer Science Conference Ro	14-100
Lab	14-301
Lab	14-302
	14-245
	14-256
	14-50

Figure 222: Location database dialog.



Location Database, sorted by Number

Name	Number
Computer Science Conference Ro	14-100
	14-245
	14-256
Lab	14-301
Lab	14-302
	14-50
Campus Auditorium	
College Conference Room	

Search by Number:

Figure 223: Location database sorted by number.

To add a new location, the administrator presses ‘`Add ...`’, whereupon the system displays the dialog of Figure 224. The ‘`Name`’ field is a free–form text string. ‘`Number`’ is an additional location identifier, typically in the form of a building/room number, but not restricted to any specific format. At least one of name or number is required to be non-empty. The following are the uniqueness rules for location name and number:

a. two or more locations can have the same name, as long as the number for each is different
b. if the number is non-empty, it must be unique among all locations

These rules allow any or all locations to have empty names or empty numbers, as long as the other component of the name/number pair is unique. And while names may be duplicated, non-empty numbers may not be. Overall, the rules mean that the pair of values for Name and Number constitute a unique identification for all rooms.

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 225 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 226. The ‘`Use`’ field describes the use for which the location is booked; it is optional, i.e., it may be empty. The ‘`Start Time`’ and ‘`Duration`’ specify the time slot of the booking; both of these fields is required, and the entered values must follow the normal data entry rules described in Section . At least one of the seven checkboxes for ‘`Days` must be on.

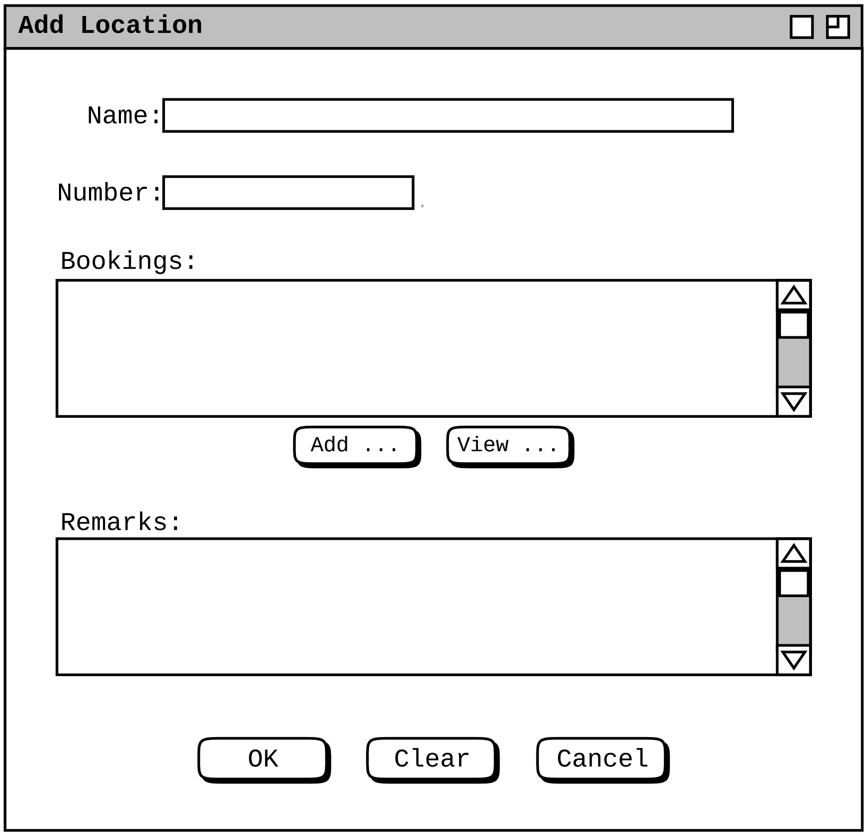
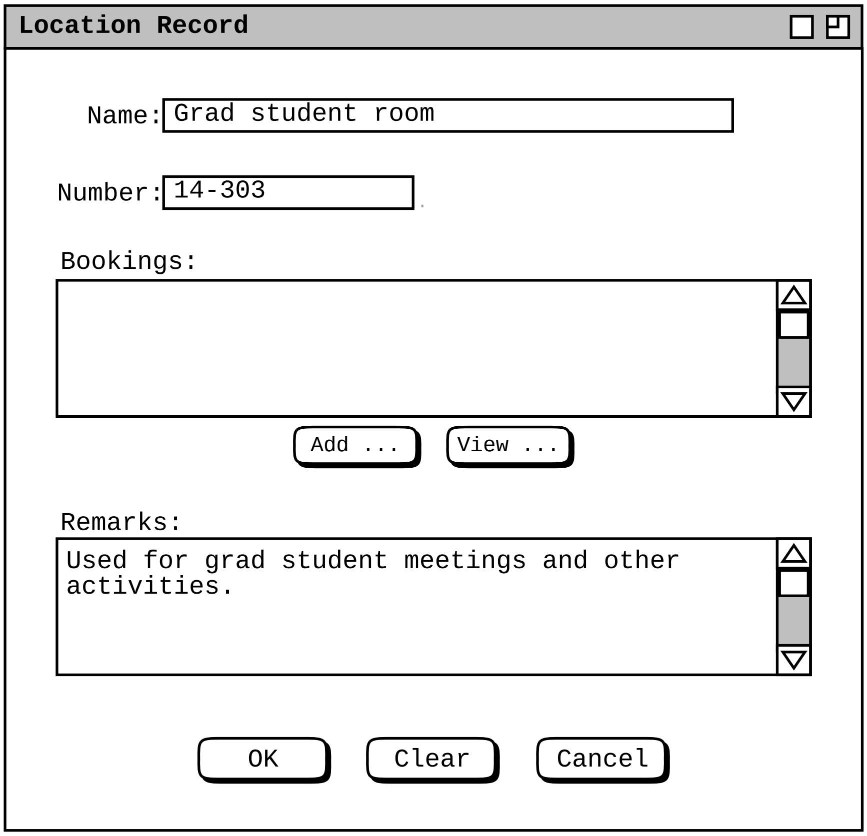


Figure 224: Add location dialog.



Name: Grad student room

Number: 14-303

Remarks: Used for grad student meetings and other activities.

Figure 225: Location database record.

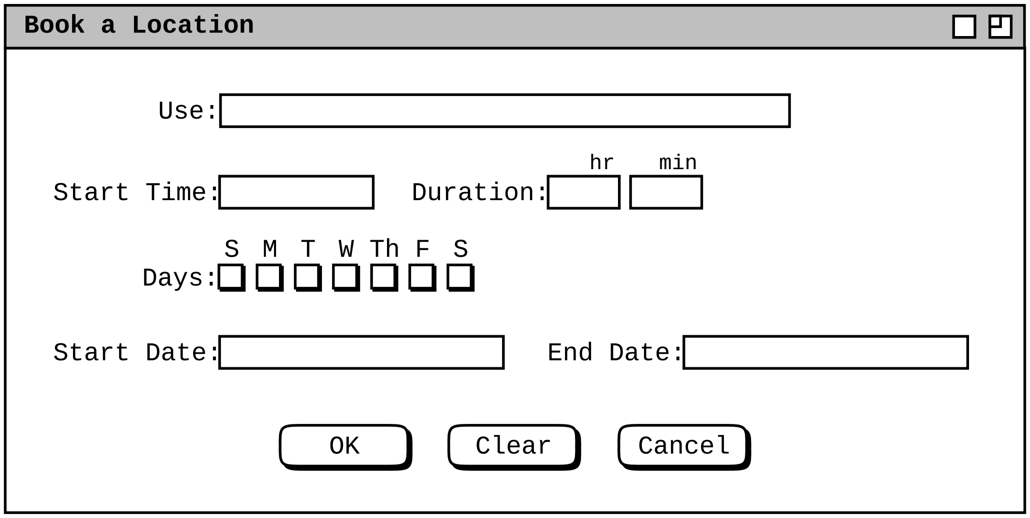


Figure 226: Book location dialog.

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 date values must follow the normal data entry rules described in Section . 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 227 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 . 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.

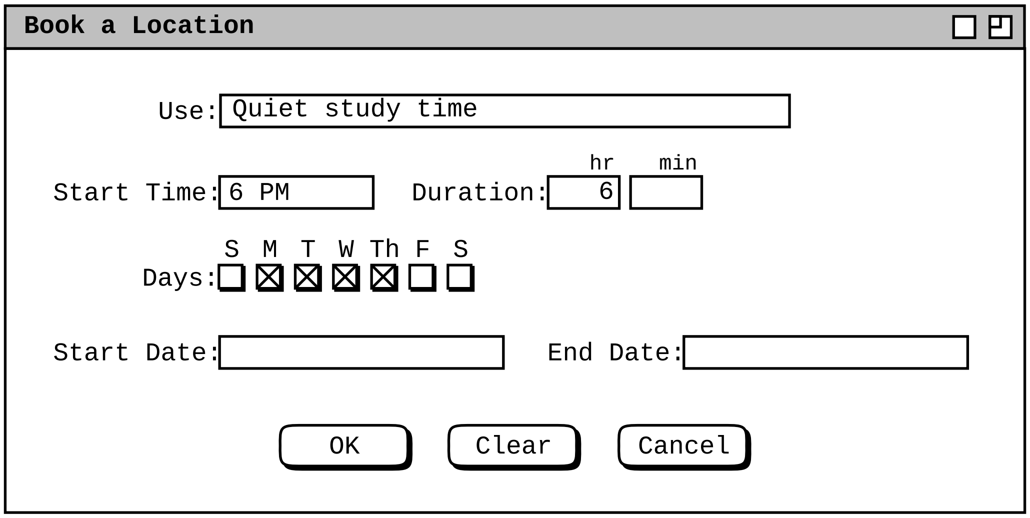
To view an existing location record in the database, the administrator selects a location name or number in Figure 222 and presses ‘`View ...’.` For example, Figure 228 shows the result of the administrator viewing the second listed location in Figure Figure 222. The location record dialog has exactly the same data fields as the add–location dialog shown in Figures 224 and 225. 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 central host copy of the meeting 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.

Figure 229 shows the result of ‘`View ...`’ for the second booking listed in Figure 228, which is a scheduled meeting. None of the data fields is editable in the displayed meeting item. Figure 230 shows the result of ‘`View ...`’ for the third booking listed, which is a non–meeting.

Since an other–usage booking is for a contiguous time period, it takes multiple bookings to define discontinuous usages. For example, the ‘`Morning Classes`’ booking in Figure 228 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`’. The edits must comply with the same data–entry rules described after Figure 226 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.



Use: Quiet study time

Start Time: 6 PM

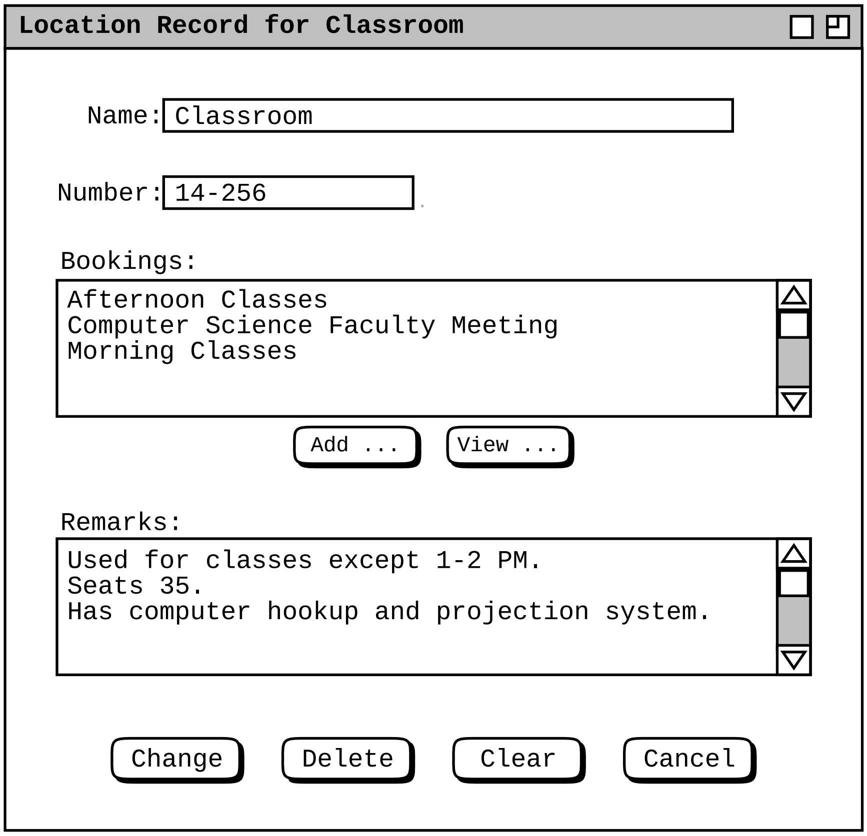
Duration: hr 6 min

Days: S ☐ M ☒ T ☒ W ☒ Th ☒ F ☐ S ☐

Start Date:

End Date:

Figure 227: A location booking for the Grad Student Room.



Name: Classroom

Number: 14-256

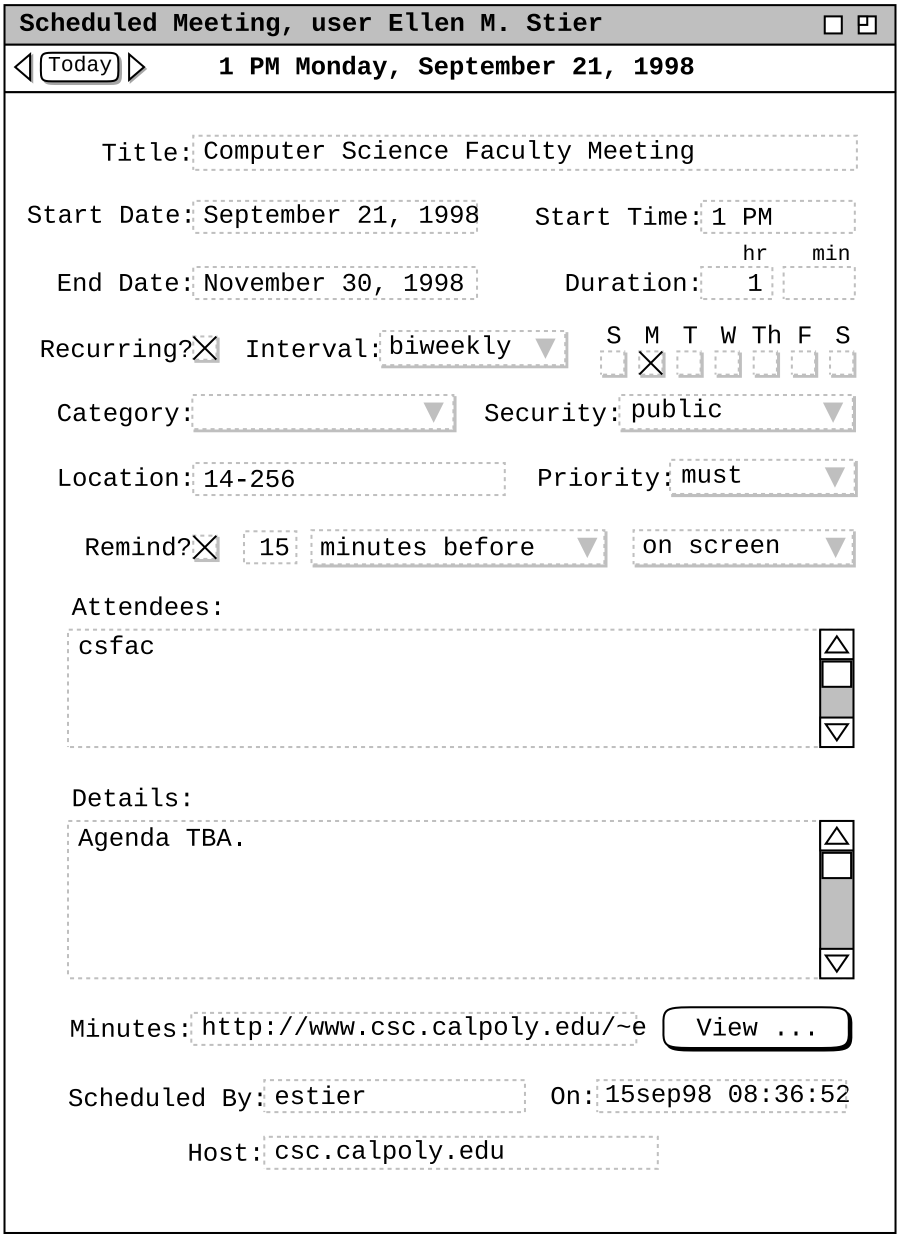
Bookings:

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 228: Viewing a location record.



Scheduled Meeting, user Ellen M. Stier

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: hr 1 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.

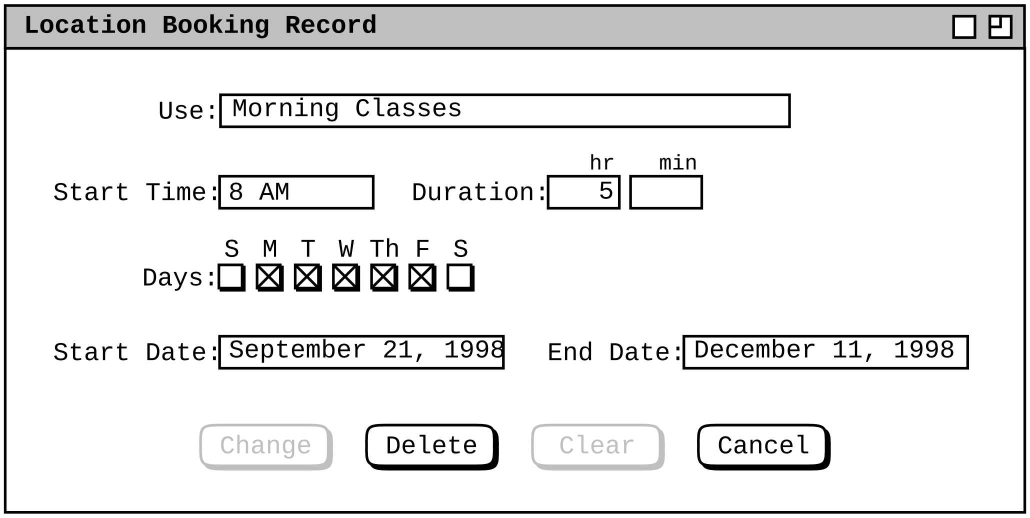
Minutes: http://www.csc.calpoly.edu/~e

Scheduled By: estier

On: 15sep98 08:36:52

Host: csc.calpoly.edu

Figure 229: Viewing a meeting from the location record dialog.



Use: Morning Classes

Start Time: 8 AM Duration: 5 hr

Days: ☐ S ☒ M ☒ T ☒ W ☒ Th ☒ F ☐ S

Start Date: September 21, 1998 End Date: December 11, 1998

Figure 230: Viewing an other-usage booking from the location record dialog.

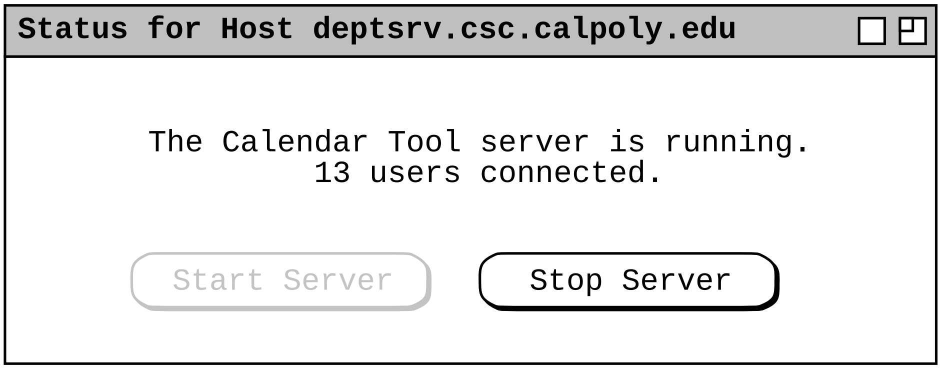
An other–usage booking is not a scheduled item, so it does not appear on any calendar. The effect that a booking has on scheduling is to prohibit a meeting from being scheduled in the location during any of the booked times and dates. An other–usage booking expires after its end date or single start 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 Privileged Admin Commands

In addition to maintaining Calendar Tool databases, the administrator can perform commands to configure the central host computer and perform other managerial functions. Scenarios follow.

2.6.5.1. Administrative Control of the Central Host Server

When the administrator selects the '`Host Status`' command in the privileged `Admin` menu, the system displays a dialog of the form shown in Figure 231.



The Calendar Tool server is running.
13 users connected.

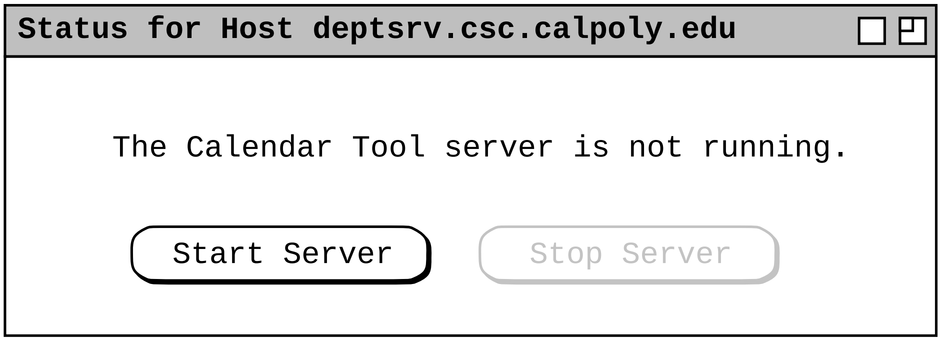
Figure 231: Central host status dialog.

The dialog shows the status of the Calendar Tool server and if running, how many users are connected. The Calendar Tool server is an independently–executing program that handles user communication with the central host. The server program runs on the same host computer where the administrator is running the Calendar Tool Administration program. The host name shown in the banner of the dialog is the unique network identification of that central host. This is the identification employed by users to establish a central host connection. Complete details on the function of the server program are presented in Section .

In the case of Figure 231, the status of the server is running, with 13 Calendar Tool users connected. When the server is not running, the host–status dialog appears as shown in Figure 232. When the server is running, the '`Start Server`' command button is disabled and the '`Stop Server`' button is enabled. The buttons are in opposite states when the server is not running.

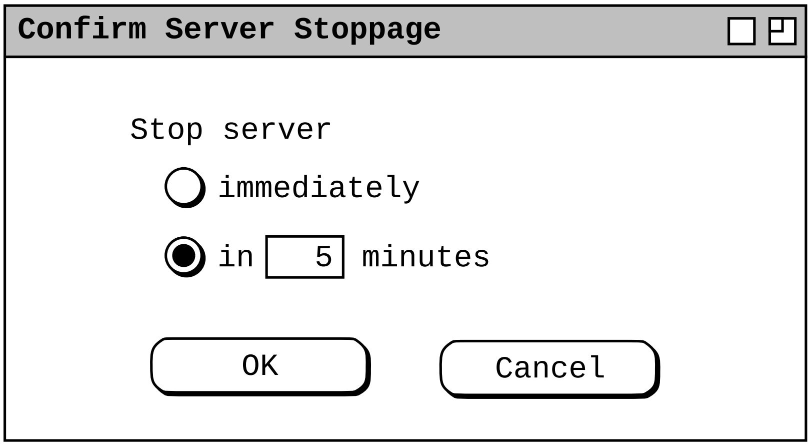
The start and stop buttons control the execution status of the server. When the administrator presses '`Start Server`', the system responds by invoking the server program on the current host and updating the status dialog message from "not running" to "running" with "0 users connected". Also, the enabled/disabled states of the command buttons are swapped, so that the stop button becomes enabled.

When the administrator presses '`Stop Server`', the system responds with the dialog shown in 233. The administrator can choose to stop the server immediately or in a specified number of minutes, between 1 and 999. The default is in 5 minutes. When the administrator presses '`OK`', the system responds by stopping the server



The Calendar Tool server is not running.

Figure 232: Central host status not running.



Stop server

○ immediately

● in 5 minutes

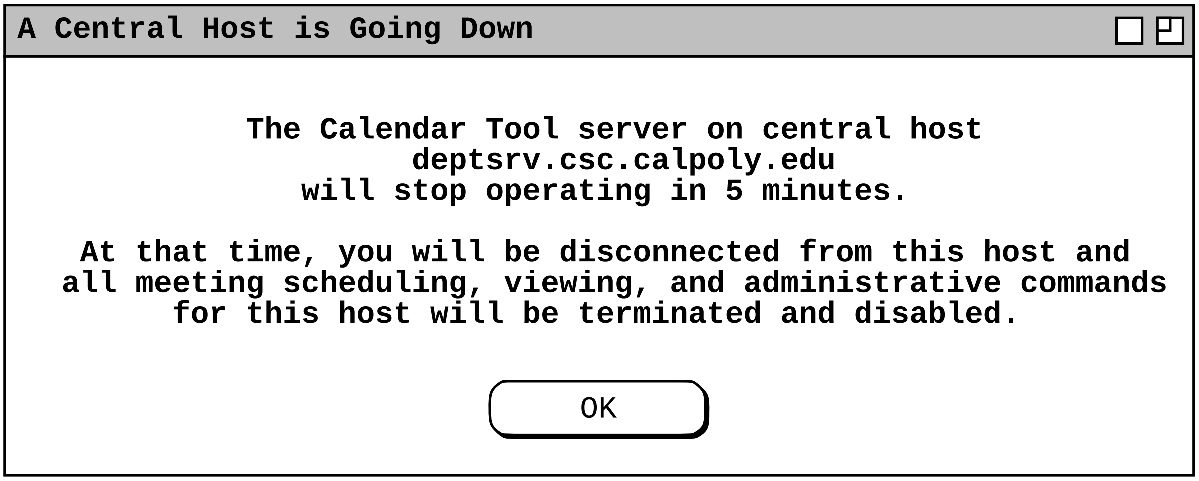
Figure 233: Confirmation for stopping the central host server.

immediately or scheduling the server to be stopped in the specified number of minutes.

When the administrator chooses to stop the server in some number of minutes, the system notifies all connected users with an on−screen notification of the form shown Figure 234. The dialog message explains when the server will stop and the effects of the stoppage.

When the time comes for the server to stop, either immediately or after the specified number of minutes have elapsed, the system notifies all connected users with the dialog shown in Figure 235. In conjunction with the notification, the system performs the following actions on the local computer of all connected users:

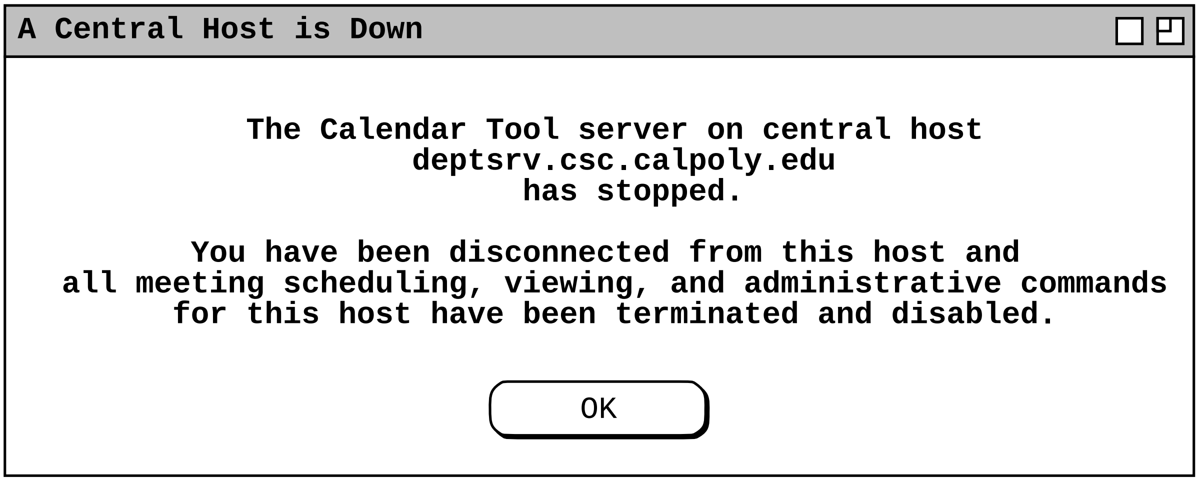
a. updates the display in the central host connection dialog (Section 2.6.6.1) to show the host as not connected;
b. immediately terminates any meeting−scheduling operations for the host and removes all windows associated with meeting scheduling for the host;
c. immediately terminates any administrative commands for the host and removes all windows associated with administrative commands for the host;


A Central Host is Going Down

The Calendar Tool server on central host
deptsrv.csc.calpoly.edu
will stop operating in 5 minutes.

At that time, you will be disconnected from this host and
all meeting scheduling, viewing, and administrative commands
for this host will be terminated and disabled.

Figure 234: Notification that server will stop.


A Central Host is Down

The Calendar Tool server on central host
deptsrv.csc.calpoly.edu
has stopped.

You have been disconnected from this host and
all meeting scheduling, viewing, and administrative commands
for this host have been terminated and disabled.

Figure 235: Notification that server has stopped.

d. removes all display windows for the calendars of other users and groups on the host;

e. if the user is not connected to any other host, disables all items on the '`Admin`' menu except '`Connect`' and '`Contact Admin`', and disables the '`Other User`' and '`Group`' items in the '`View`' menu.

2.6.5.2. Setting the Administrator Password

As noted in the introduction, there is a single administrative password for any given installation of the Calendar Tool Administration program. The password is set when the program is initially installed (Section), and may be changed any time thereafter. To do so, the administrator selects the '`Password`' item in the '`Admin`' menu, whereupon the system displays the dialog shown in Figure 236. To effect the password change, the administrator enters the current password, a new password, and then reenters the new password to confirm. As all password values are typed, they are echoed as asterisk characters, not as the actual characters typed. A password cannot be empty, but there is otherwise no restriction on the password value; it may be one or more characters that are legally typeable on the platform where the administration program is running.

When the administrator presses '`OK`', the system confirms that the current password is correct and that both versions of the new password are identical. If so confirmed, the system changes the password for all subsequent administrator access. If the current password is incorrect or the versions of the new passwords differ, the system displays an error dialog to that effect, upon dismissal of which the user can try again.

It is the responsibility of an administrator who changes the password to notify any other individuals who need to be aware of the change. The Calendar Tool provides no means to perform this notification.

2.6.5.3. Setting the Administrator Email Address

Regular Calendar Tool users can contact an administrator using the '`Contact Admin`' command described in Section 2.6.6.5. The contact is made through an electronic mail address for the Calendar Tool administrator(s) on a particular host. The address is any legal email address which the administrator(s) designate as the contact point for administrative communication. It can be an address specifically created for Calendar Tool administration or the address of some individual administrator.

The email address is set when the Calendar Tool Administration program is initially installed (Section), and may be changed any time thereafter. To do so, the administrator selects the '`Email`' item in the '`Admin`' menu, in response to which the system displays the dialog shown in Figure 237. The current address is the default value when the dialog appears. The administrator can enter any other address and press '`OK`' to confirm the change, or '`Cancel`' to

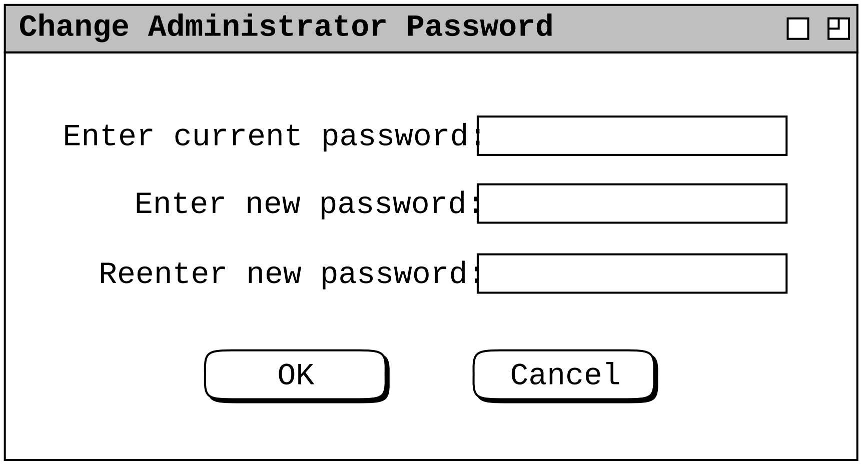


Figure 236: Administrator password change dialog.

Enter admin email address: calsrv@csc.calpoly.edu

Figure 237: Email address dialog.

leave the address unchanged. The Calendar Tool Administration program validates the entered address insofar as is possible, as described in Section . Actual communication using this address is performed entirely through the services of an external email agent, the identity of which is specified as an option (see Section 2.7.5). If the administrator enters an incorrect or invalid address that cannot be identified as such in advance, this situation is made evident to the user through whatever behavior the external mail program exhibits given the incorrect or invalid address.

The creation of the address itself as a valid electronic mail destination is beyond the scope of Calendar Tool Administration. The address must be created by the staff of the underlying operating environment or some other individual who has the capability to create a valid and usable email address.

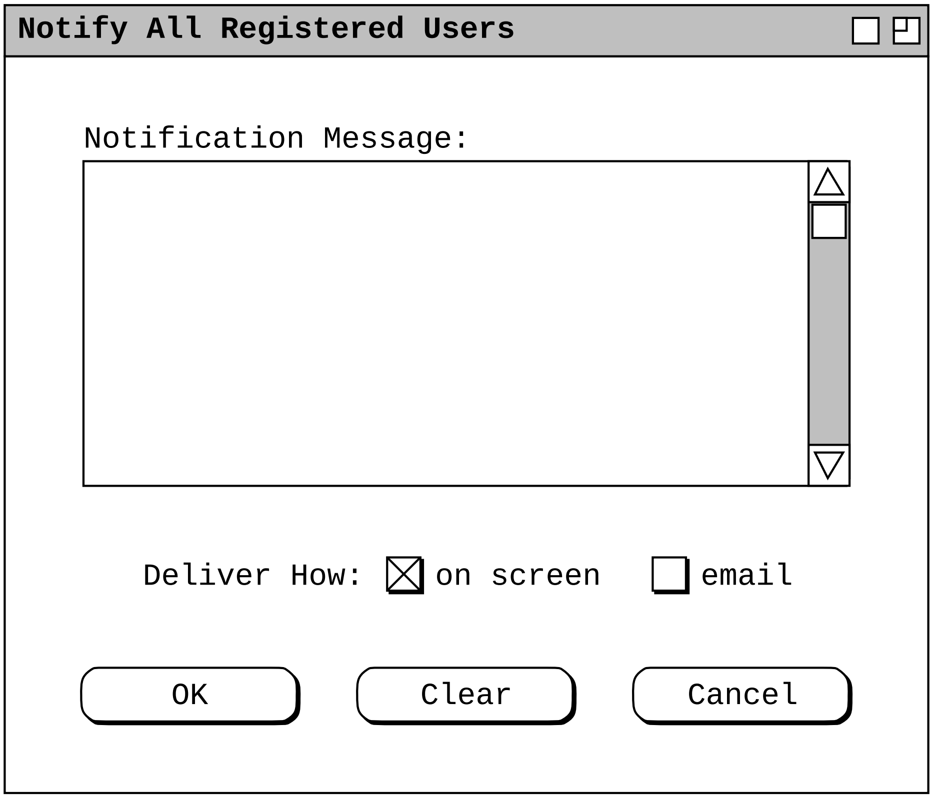
2.6.5.4. Notifying Users

Preceding scenarios have described a variety of notifications sent to Calendar Tool users to inform them of significant administrative actions. The administrator can also send a general−purpose notification to all registered users with the '`Notify Users`' command. When the administrator executes this command, the system displays the dialog in Figure 238. The administrator enters the desired message text and chooses whether to deliver the notification in an on−screen dialog, in an email message, or both. At least one of the two forms of delivery must be selected.

Figure 239 shows the administrator sending a notification that the central host computer will go down for scheduled maintenance.

Unlike meeting notifications (Section 2.4.1.5), there is no user option to disable on−screen administrative notifications. This includes all administrative notifications, whether sent with the '`Notify Users`' command, or as the result of executing any of the other administrative commands that generate the notifications. On−screen administrative notifications appear immediately for all affected users who are connected to the originating host at the time the notification is sent. When a user is not connected, the notification is queued for display the next time the user connects to the originating host, as described in Section 2.6.6.1. Aside from the fact that administrative notifications cannot be refused, the queue of pending administrative notifications is managed in the same way as the meeting notification queue described in Section 2.4.1.6.8. All administrative notifications are kept in a separate queue, the complete contents of which are displayed to a connecting user before any meeting notifications are displayed.

The Calendar Tool Administration program provides no command specifically for communication with individual users. Individual communication occurs when notifications are sent to users as a result of administrative commands that generate the notifications. An administrator can reply via external email to an individual message sent with the '`Contact Admin`' command (Section 2.6.6.5). An administrator can also send an individual email message to any user, by way of the email address listed in a user's database record. However, the Calendar Tool Administration program provides no command to launch an email program for this purpose. The administrator can use the copy and paste facility of the underlying operating environment to copy the address from the database record and paste it into the external email program.

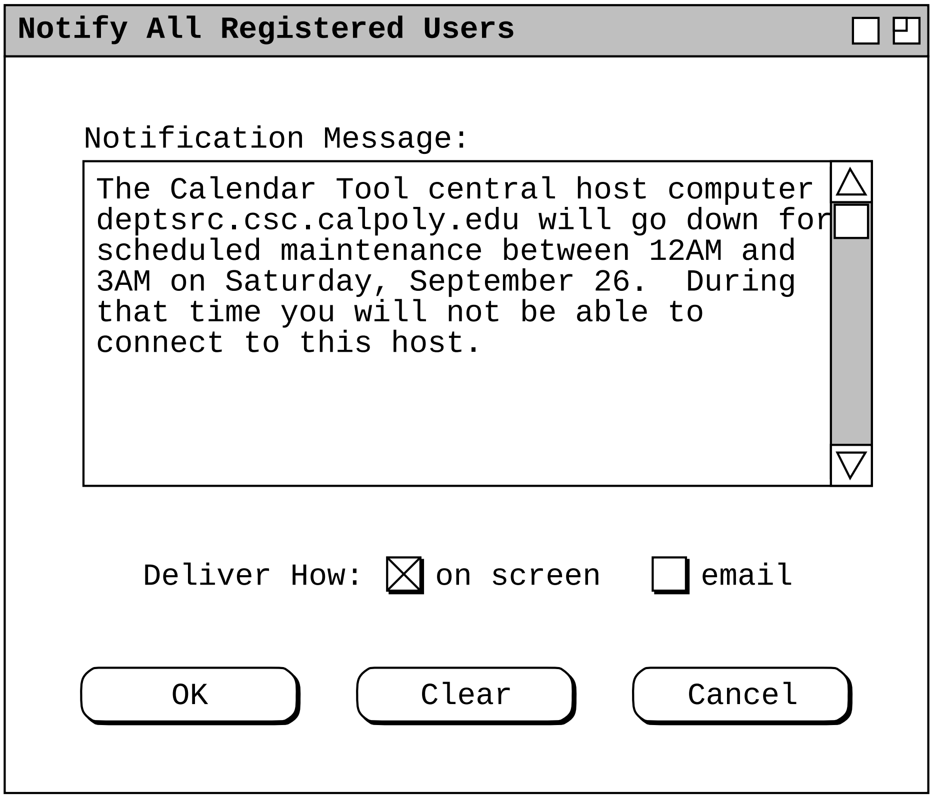


Notify All Registered Users

Notification Message:

Deliver How: ☒ on screen ☐ email

Figure 238: Notify users dialog.



Notification Message:

The Calendar Tool central host computer deptsrc.csc.calpoly.edu will go down for scheduled maintenance between 12AM and 3AM on Saturday, September 26. During that time you will not be able to connect to this host.

Deliver How: ☒ on screen ☐ email

Figure 239: A notification about central host maintenance outage.

2.6.5.5. Setting Calendar Size Limits

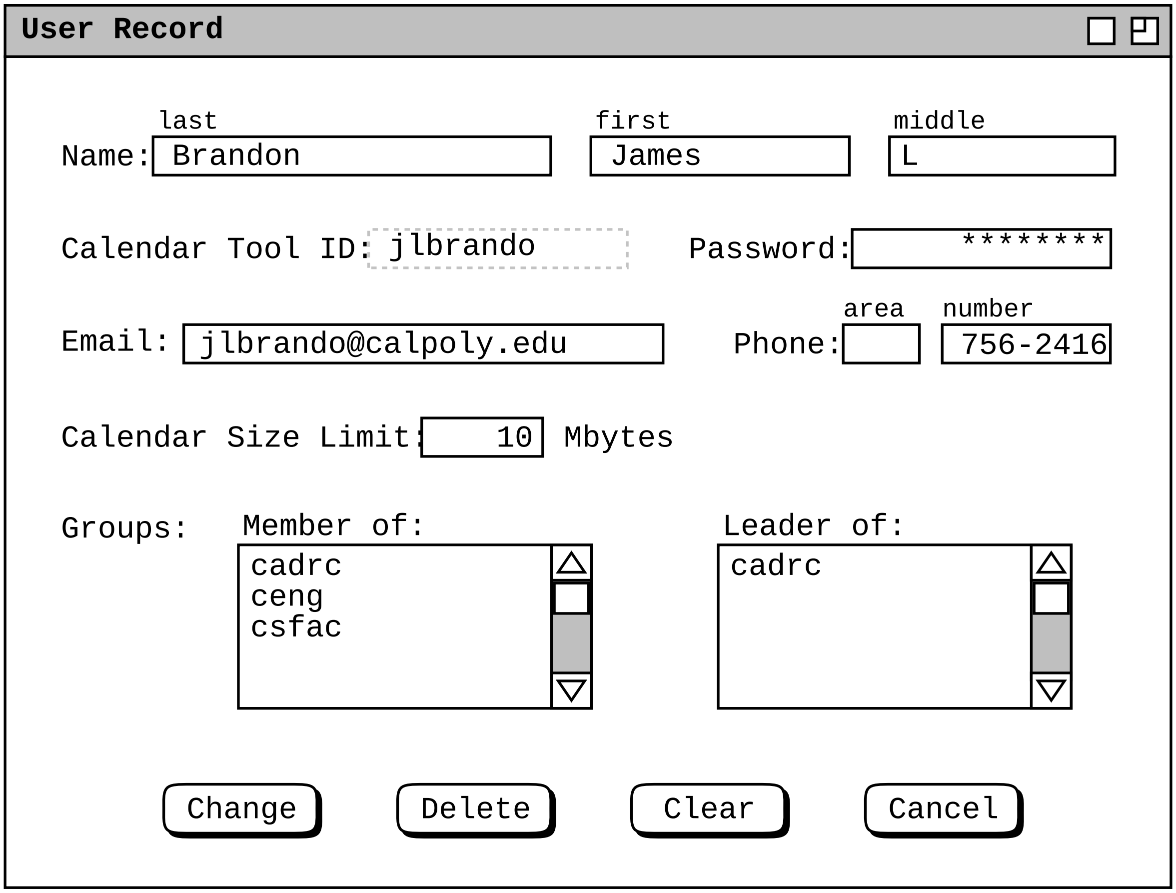
As explained in Section 2.8.8, the central host repository contains a copy of each registered user's calendar and the only copy of each group calendar. As explained further in Sections 2.6.6.1, and Section 2.8.4, the system copies the user's local calendar to the central host when the user initially establishes a host connection and whenever the user saves the calendar while connected.

The reason that user and group calendars on the host are stored in individual files is so the administrator can ascertain the size of each calendar using the normal file–size commands available in the underlying operating environment. In this way, if a user or group file exceeds a size that can be supported on a given host, the administrator can limit the file size by setting the value of the '`Calendar Size Limit`' component in the user and group records. For example, Figure 240 shows the administrator having set the size limit to 10 Mbytes for user James Brandon.

For a typical implementation of the Calendar Tool, it is expected that the size of calendar data files will remain manageably small, unless they contain a particularly large number of scheduled items. Nevertheless, it is necessary for an administrator to be able to cap the size of user calendars as needed on a particular host.

When the administrator sets the size limit for a user's calendar, the user is not notified of the setting unless the current size of the user's host calendar is greater than the limit. The user can view the size limit at any time by viewing her user record in the database.

In the case where the current size of the user's host calendar is greater than a changed size limit, the system notifies the user with an on–screen dialog of the form shown in Figure 241. The dialog explains the change and the effects it has the user's host calendar. The notification is sent immediately after the administer confirms the size limit setting with the '`Change`' command in user record dialog. Also, the item deletion actions explained in the message are performed immediately upon confirmation of the size change. If the user is not currently connected to the host, the notification is queued and subsequently delivered to the user the next time she connects, as explained in Section 2.6.5.4.



User Record

Name: last Brandon first James middle L

Calendar Tool ID: jlbrando

Password: ********

Email: jlbrando@calpoly.edu

Phone: area number 756-2416

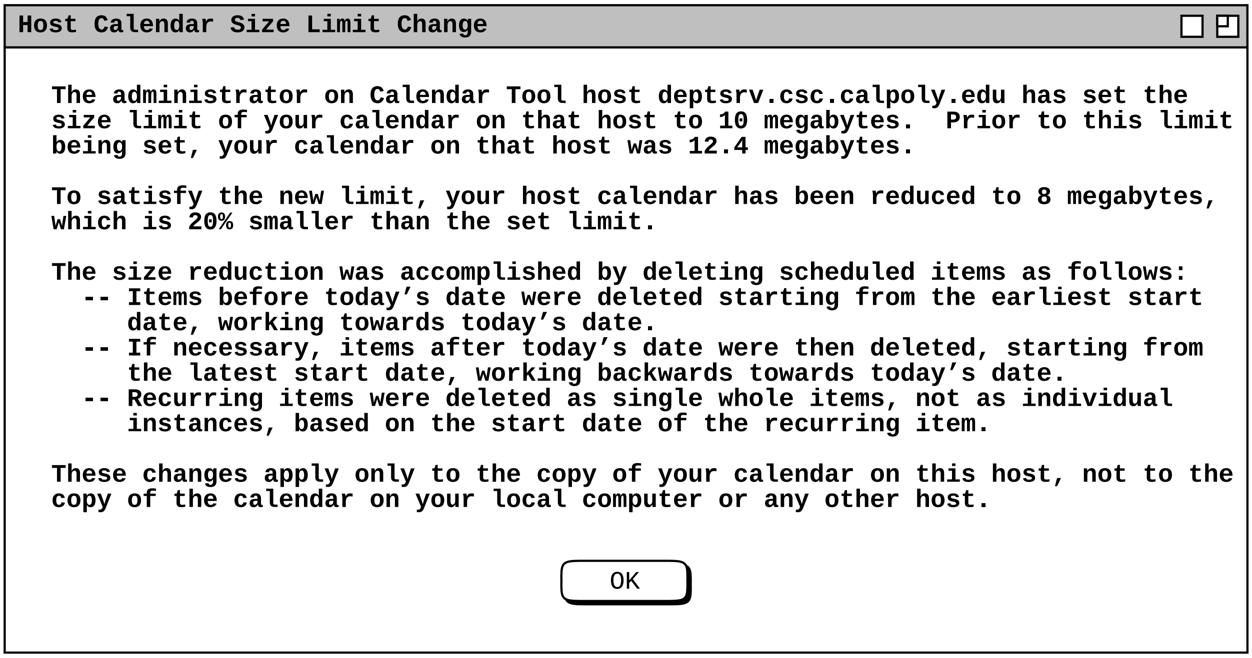
Calendar Size Limit: 10 Mbytes

Groups:

Member of:
cadrc
ceng
csfac

Leader of:
cadrc

Figure 240: Size limit set in a user record.



The administrator on Calendar Tool host deptsrv.csc.calpoly.edu has set the size limit of your calendar on that host to 10 megabytes. Prior to this limit being set, your calendar on that host was 12.4 megabytes.

To satisfy the new limit, your host calendar has been reduced to 8 megabytes, which is 20% smaller than the set limit.

The size reduction was accomplished by deleting scheduled items as follows:
- Items before today’s date were deleted starting from the earliest start date, working towards today’s date.
- If necessary, items after today’s date were then deleted, starting from the latest start date, working backwards towards today’s date.
- Recurring items were deleted as single whole items, not as individual instances, based on the start date of the recurring item.

These changes apply only to the copy of your calendar on this host, not to the copy of the calendar on your local computer or any other host.

Figure 241: Host calendar truncation notification, after size limit change.

The other circumstance that leads to user notification of a size limit is when the user adds scheduled items to a calendar such that the addition causes the host calendar size limit to be exceeded. This case arises when the user saves the local calendar file when connected to the host, or establishes a host connection after having added items locally while not connected. When one of these actions occurs, the system notifies the user with the same form of dialog as shown in Figure 241, but with the text of the first two paragraphs of the message changed to the following:

> **The size of the local calendar just transmitted to the host**
> **exceeds the calendar size limit for the host.**
>
> **To satisfy the required limit, your host calendar has been**
> **reduced to 8 megabytes, which is 20% smaller than the set limit.**

The third and fourth paragraphs of the notification are the same as in Figure 241 As explained in the notification message, the system reduces the size of the user calendar by 20%, in order to achieve a workable calendar size.

The circumstance that leads to an automatic size-limit change in a group calendar is when a group leader schedules a meeting that cause the size of the group calendar to be exceeded. In such a case, the system reduces the size of the group calendar using the same 20%-reduction procedure as for a user calendar. No notification is sent to anyone if the size limit is changed on a group calendar, not even the group’s leader.

2.6.6. Regular User Access to Admin Commands

As noted in the introduction, only the ‘`Connect`’ and ‘`Contact Admin`’ commands are enabled in the regular–user `Admin` menu, prior to establishing a host connection. Once connected, the user can perform all administrative commands in the regular user interface, as presented in the scenarios that follow.

2.6.6.1. Central Host Connections

To connect to a central host, the user selects the ‘`Connect`’ command in the ‘`Admin`’ menu. In response, the system displays the dialog in Figure 242. The host–connection dialog contains a table that associates Calendar Tool central host computers with local calendar files. The rightmost column in the table indicates which hosts are currently connected. By default, there are no pre–defined entries in the table.

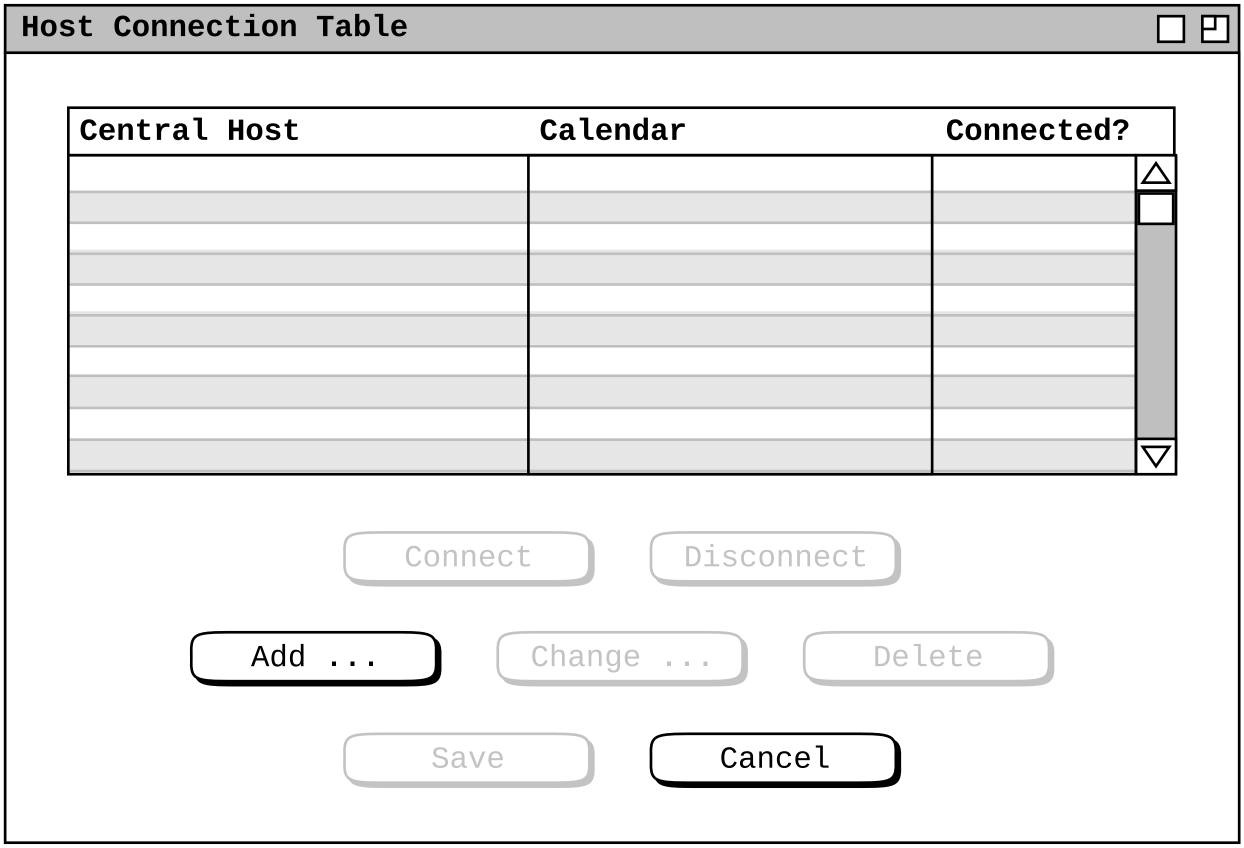


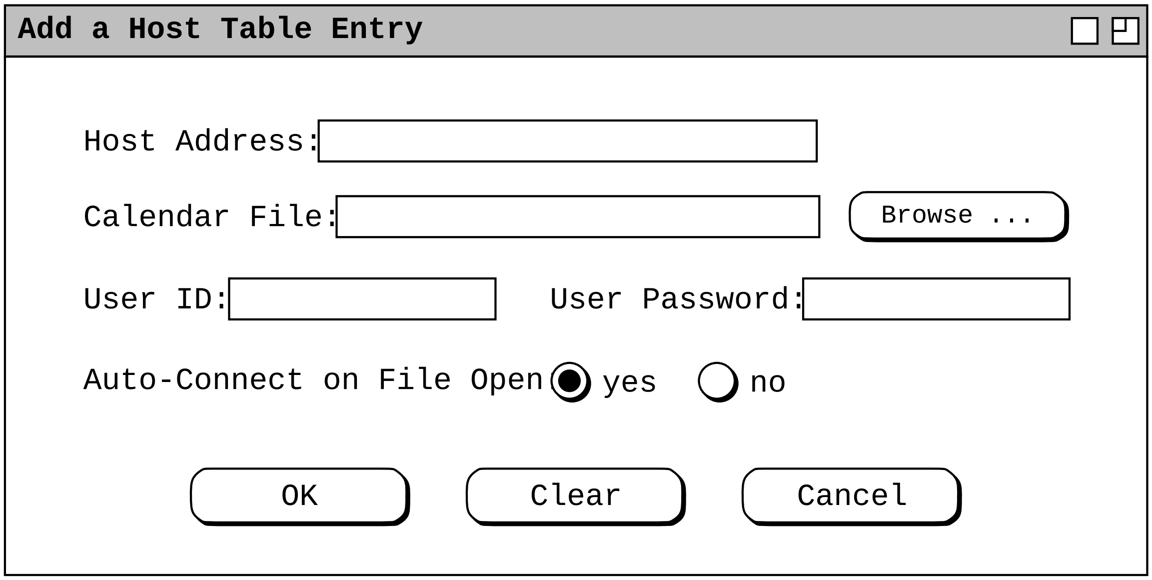
Figure 242: Host connection dialog.

The association of a central host computer with a local calendar file is a fundamental feature of Calendar Tool networking. As explained in the introduction, regular users run the Calendar Tool on local computers, where their calendar files are stored. To support multi–user calendar sharing, there are Calendar Tool central host computers to which users can connect. When a user is connected to a particular host, the associated calendar is the user's "portal" onto that host. When a user connects to the host, a copy of the associated local calendar is sent to the host. When another user schedules meetings on the host, the meeting is penciled in on the associated calendar. Further discussion on these and other aspects of Calendar Tool networking appears in Section .

With an initially empty host table, as in Figure 242, the only enabled command buttons are ‘`Add ...`’ and ‘`Cancel`’. The other buttons become enabled as table entries are added. To add a new entry, the user presses ‘`Add ...`’, in response to which the system displays the dialog in Figure 243. The ‘`Host Address`’ field is the unique network address of the Calendar Tool central host computer to be connected. The format of the address is platform–dependent, based on the type of network to which host computers are attached. IP address is a typical format. ‘`Calendar File`’ contains the name of the local calendar file to be associated with the specified host. If a leading file path is not specified in the calendar file name, then the file is assumed to be in user's Calendar Tool directory, as described in Section 2.8.1.

The ‘`User ID`’ and ‘`User Password`’ fields contain the Calendar Tool ID and password for the user on the specified host. These values must match the ID and password values defined by the host administrator in the user's database record (see Section 2.6.2).

The last field in the add–host dialog is ‘`Auto-Connect on File Open`’. The setting of this field specifies whether or not the system initiates a connection to the host when the associated calendar file is opened. This is the same option value that appears in the ‘`Admin`’ tab of the ‘`Administrative`’ options dialog, described in Section



Host Address:

Calendar File:

User ID:

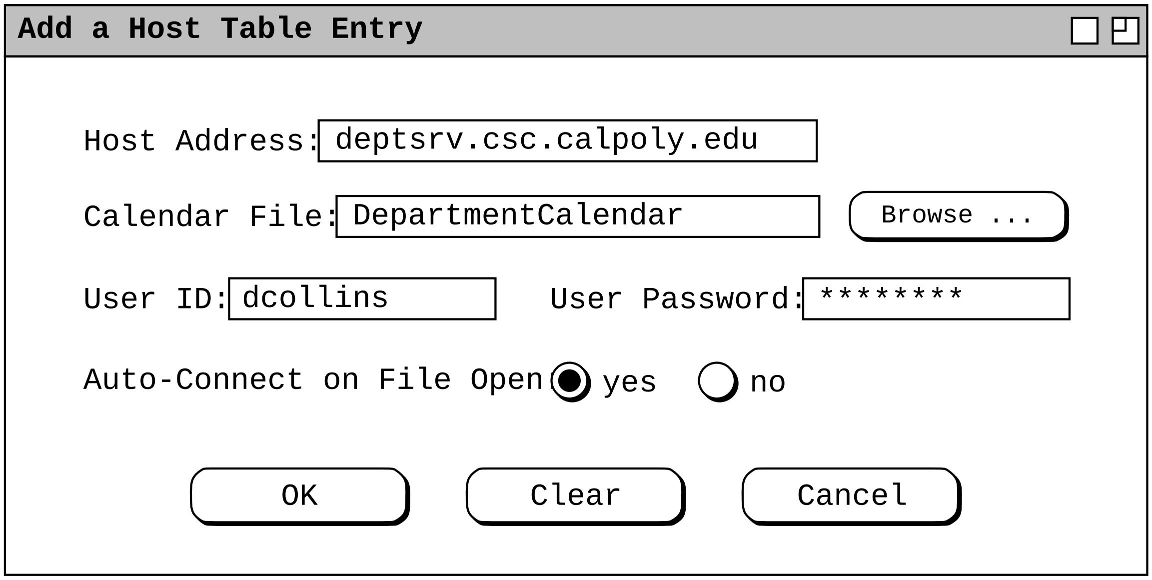
User Password:

Auto-Connect on File Open ● yes ○ no

Figure 243: Add host dialog.

2.7.5. The setting also appears in the add−host dialog, as a convenience. When the user changes the option setting here, the system changes it to the same value in the options dialog, and vice versa.

Figure 244 shows the user having entered typical values for a host table entry. In order to enter the proper ID and password values for the specified host, the user must have been notified by the host administrator when the user record was created, as explained in Section 2.6.2 The ID and password values entered in this dialog become part of the host table entry for the specified host. This information is stored on the user's local computer. When the user



Host Address: deptsrv.csc.calpoly.edu

Calendar File: DepartmentCalendar

User ID: dcollins

User Password: ********

Auto-Connect on File Open ● yes ○ no

Figure 244: Defining a host table entry.

initiates a connection to the host, the ID and password in the local table entry are matched against the ID and password in the user's database record stored on the host. This matching is what validates the connection. The complete details of the connection process are described momentarily.

The user presses '`OK`' in Figure 244 to confirm the addition of the host table entry, after which the system removes the add dialog from the screen and updates the host connection table as shown in Figure 245. Note that the ID, password, and auto−connect values do not appear in the host table display. These values are accessible in the add (and change) entry dialogs. The system does not validate any of the entered values in the add−host dialog, other than to require that all data fields are non-empty. Complete validation is performed when the user initiates a connection, as explained next. Note also that the '`Save`' button is enabled now that an entry has been made. Details of '`Save`' are covered following the discussion of making connections.

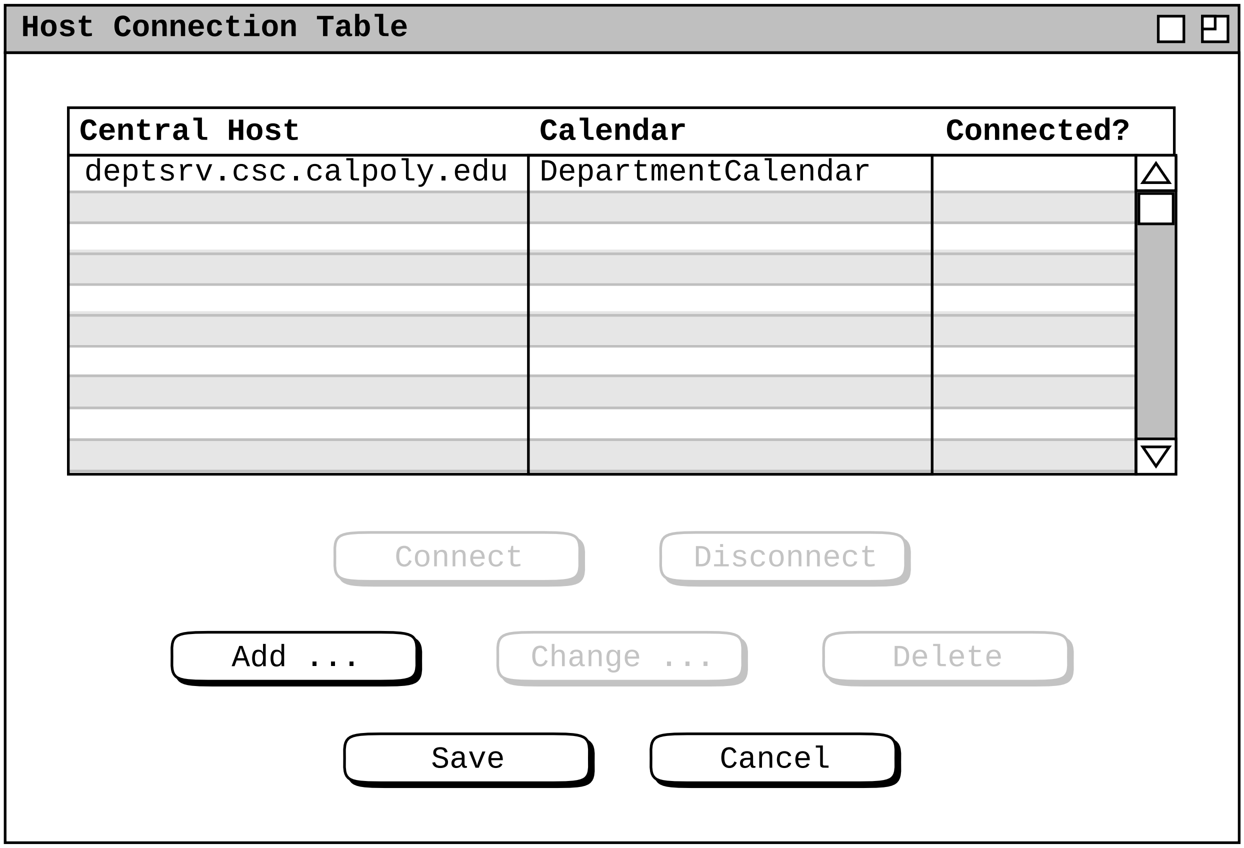
Once at least one host is in the table, the user can initiate a host connection. To do so, the user selects the host by clicking on the desired row in the table. In response, the system highlights the row and enables the '`Connect`', '`Change`', and '`Delete`' buttons, as shown in Figure 246. To initiate the host connection, the user presses the '`Connect`' button. In response, the system performs the following validation actions:

a. verifies that the host is accessible on the network and is currently running a Calendar Tool server (see Section 2.6.5.1);

b. verifies that the user ID and password in the host table match those of a registered user on the host;

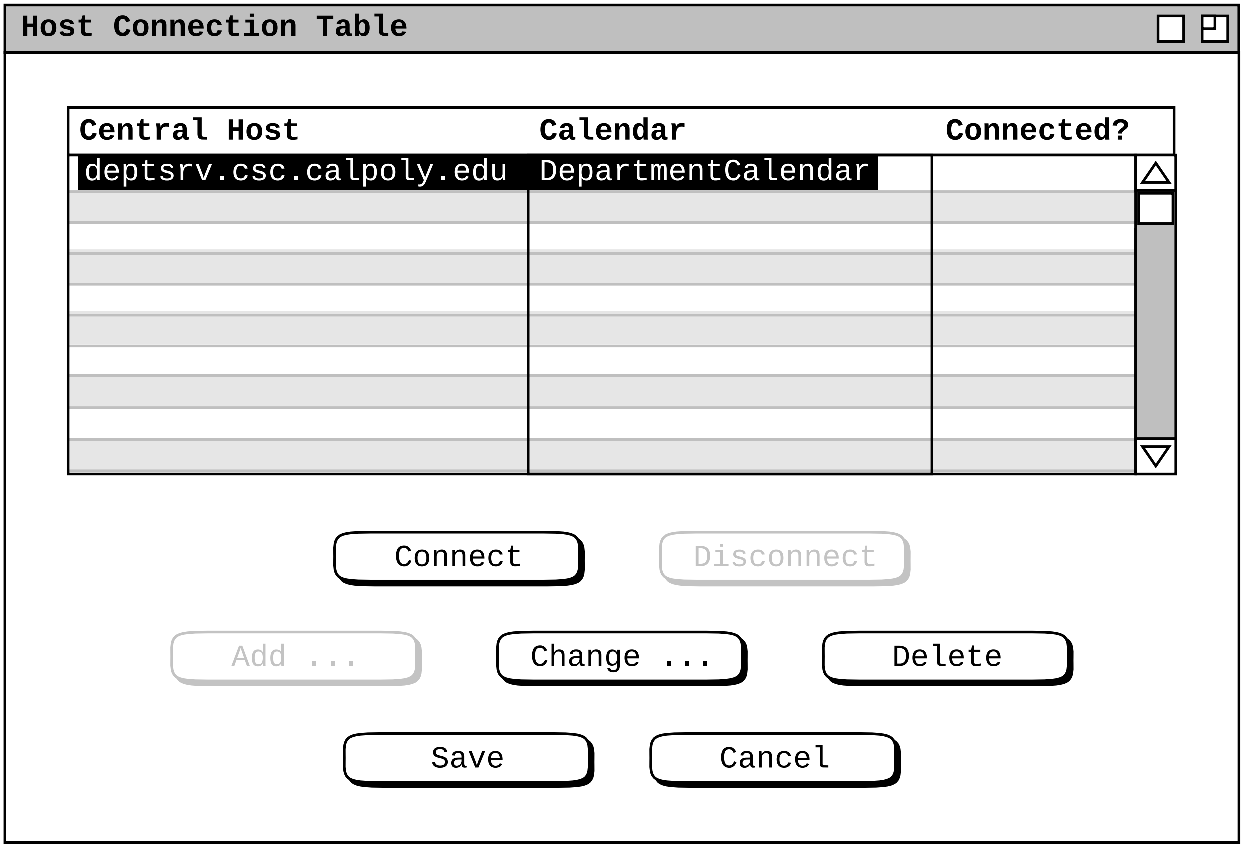
c. verifies that the file associated with the host is an existing Calendar Tool file on the local computer that is readable and writable by the user

If the validation actions succeed, the system proceeds to establish the connection and performs the following



Central Host	Calendar	Connected?
deptsrv.csc.calpoly.edu	DepartmentCalendar	

Figure 245: Host connection table with new entry added.



Central Host	Calendar	Connected?
deptsrv.csc.calpoly.edu	DepartmentCalendar	

Figure 246: Host selected in table.

actions:

a. opens the associated calendar file if it is not already open;

b. copies all new penciled–in meetings from the host calendar to the associated local calendar;

c. copies the entire local calendar, including the new pencil–ins, to the host, thereby making the local and host copies of the calendar identical;

d. processes all pending notifications, as described in Sections 2.4.1.6.8, 2.5.2.2.9, and 2.6.5.4;

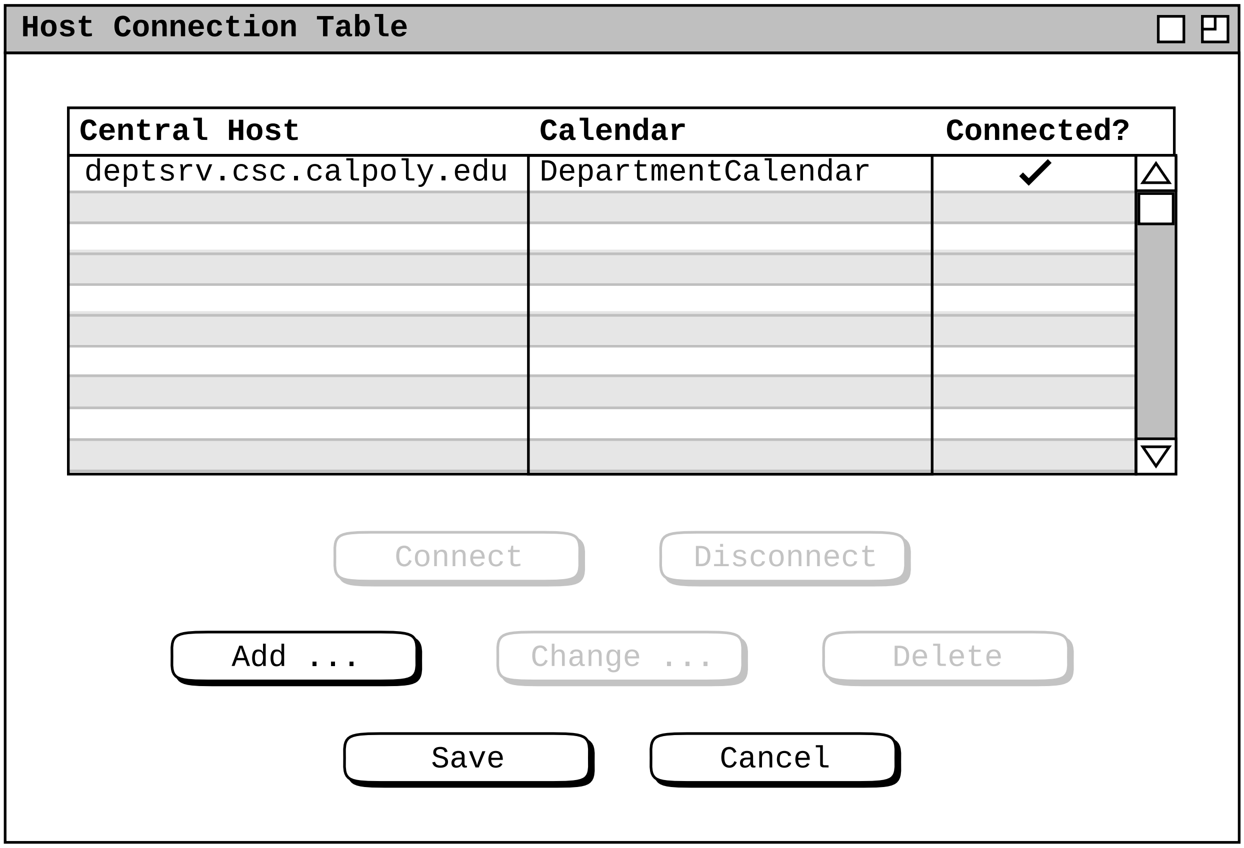
e. in the host table: marks the host as connected; unhighlights the newly connected host; disables the '`Connect`', '`Change`', and '`Delete`' buttons, all as shown in Figure 247.

The system also performs actions b and c any time the user subsequently saves the connected calendar; further details of save commands are covered in Section 2.8.4.

The set of new penciled–in meetings is comprised of all those penciled in on the central host calendar that have not yet been copied to the local calendar. The contents of this meeting set is determined by the setting of the pencil–in acceptance option on the host copy of the calendar when the meeting is scheduled. Details of the pencil–in acceptance option are described in Section 2.4.1.5 The three possible cases for copying new penciled–ins are the following:

o If pencil–in is accepted whenever connected to the host, then the set of new penciled–in meetings at the time of establishing a connection consists of all those meetings penciled in since the user last disconnected from the host.

o If pencil–in is accepted only at initial connection to the host, then the set of new penciled–in meetings at the time of establishing a connection consists of all those meetings penciled in since the user last connected to the host, or since the user last executed the '`Get now`' command in the Scheduling Meetings Options tab (see Section 2.7.3.3), whichever occurred most recently.



Central Host	Calendar	Connected?
deptsrv.csc.calpoly.edu	DepartmentCalendar	✓

Figure 247: Host connection established.

o If pencil–in is never accepted, then the set of new penciled–in meetings is empty at the time of establishing a connection.

If any part of the connection validation process fails, the system issues an appropriate error message, as described in Section .

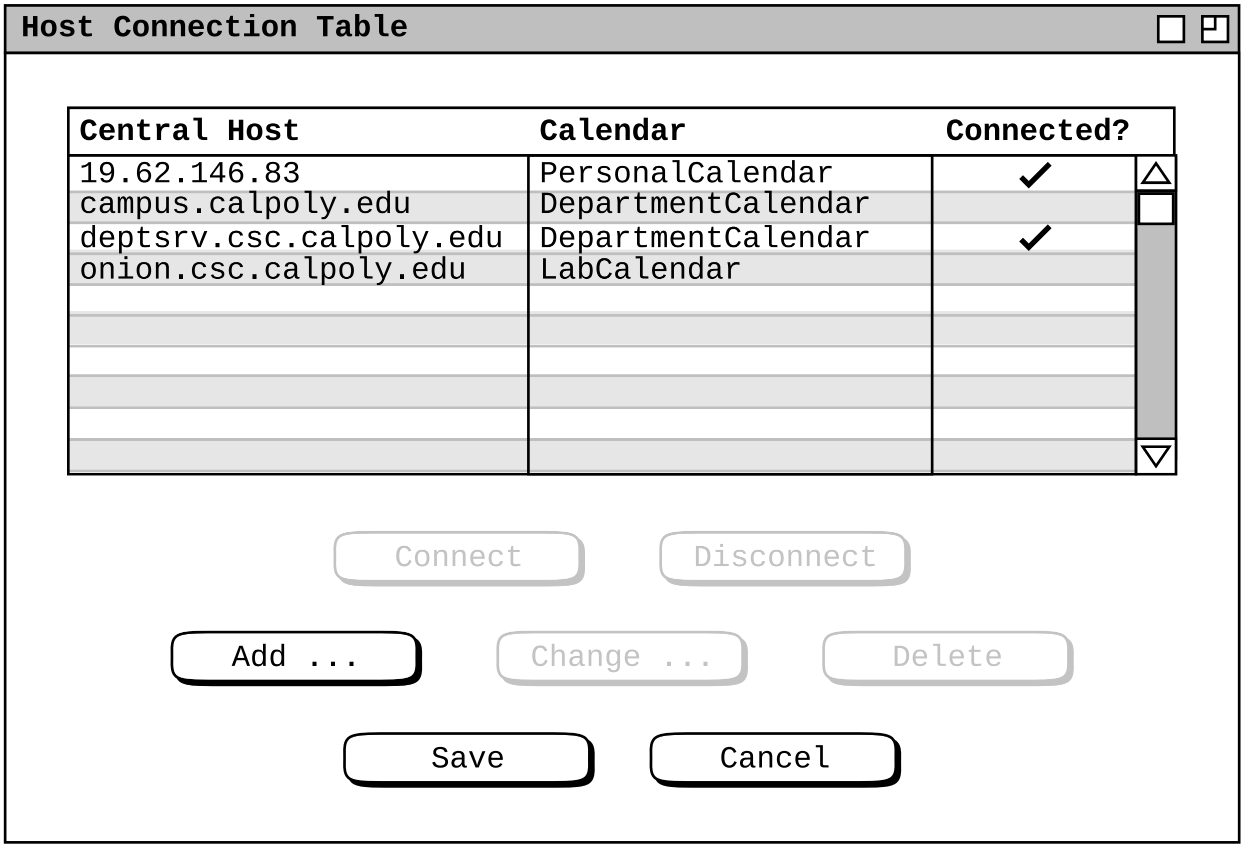
The user can add multiple entries in the host table. For example, Figure 248 shows the user having added three more entries, and having established another connection. The specific rules for multiple calendar/host associations are the following:

a. the same host address cannot appear more than once in the table, where "same" means lexically equal as strings

b. more than one host can be associated with the same calendar, but at most one of the same-calendar hosts can be connected at any given time

Figure 248 illustrates the case where the user has associated the same calendar with two different hosts. The `DepartmentCalendar` is associated with both the `campus ...` and `deptsrv ...` hosts. The utility of such multi–host–to-calendar associations is discussed further in Section . As the user adds each new table entry, the system keeps the table rows sorted by host address in ascending lexical order.

To change or delete a host table entry, the user selects the entry of an unconnected host. In response, the system enables the ‘`Change`’ and ‘`Delete`’ buttons. A host entry cannot be changed or deleted when the host is connected. When the user selects a connected host, the system enables the ‘`Disconnect`’ button, leaving the ‘`Change`’ and ‘`Delete`’ buttons disabled.



Central Host	Calendar	Connected?
19.62.146.83	PersonalCalendar	✓
campus.calpoly.edu	DepartmentCalendar	
deptsrv.csc.calpoly.edu	DepartmentCalendar	✓
onion.csc.calpoly.edu	LabCalendar	

Figure 248: Host connection with additional entries.

When the user presses ‘`Change`’, the system displays the same dialog used to add an entry (Figures 243 and 244), with the word "`Add`" in the window banner replaced with "`Change`". The current values of the selected table entry are displayed in the change dialog. The user can make any changes, except that none of the data fields can be empty. As the user confirms each new table entry change, the system keeps the table rows sorted by host address in ascending lexical order.

When the user presses ‘`Delete`’, the system displays a confirmation dialog with the message "Are you sure you want to delete the selected host table entry?", with an ‘`OK`’ button to confirm the delete and ‘`Cancel`’ button to cancel it.

When the user presses ‘`Disconnect`’ (after having selected a connected host in the table), the system displays a confirmation dialog with the message "Are you sure you want to disconnect from the selected host?", with an ‘`OK`’ button to confirm the disconnect and ‘`Cancel`’ button to cancel it. Disconnection does not close the associated calendar, however the converse is true. That is, closing the calendar associated with a connected host performs a disconnect with that host, as described in Section 2.8.3.

When the user presses the enabled ‘`Save`’ button, the system saves the contents of the host table into the Calendar Tool `Settings` file. Only the host table portion of ‘`Settings`’ is affected by this save, with all other portions of the settings file left unchanged. Details of `Settings` file contents and related commands are covered in Sections 2.8.1 and Unsaved changes can also be saved using the ‘`Save Settings`’ command in the ‘`File`’ menu, as described in Section .

When the user presses ‘`Cancel`’, the system clears any unsaved changes to table entries that are not connected, and removes the dialog from the screen. Unsaved changes to connected table entries are unaffected by ‘`Cancel`’, and reappear in their pre-cancel state if the user re-executes the ‘`Admin->Connect`’ command.

The following is a summary of the button states in the host connection table, based on what if any entry the user has selected:

a. When no entry is selected, including the case where the table is empty, only the ‘`Add`’ and ‘`Cancel`’ buttons are enabled.

b. When an unconnected entry is selected, the ‘`Change`’ and ‘`Delete`’ buttons are enabled; ‘`Add`’ and ‘`Disconnect`’ are disabled; ‘`Connect`’ is enabled if the selected host is not connected with another calendar in the table; ‘`Cancel`’ is enabled.

c. When a connected entry is selected, only ‘`Disconnect`’ and ‘`Cancel`’ are enabled.

d. ‘`Save`’ is enabled whenever the user has edited one or more fields so that the values are different that those most recently saved.

The user cannot select more than one entry in the table at a time.

To enforce the calendar/host association rules fully, the Calendar Tool server program prohibits more than one connection at a time from the same user to the same host. While the intent of the association rules is to prohibit this, they do not cover the following cases:

a. two different central host addresses in the connection table are actually the address of the same host computer, e.g., one is an IP number, the other a domain-qualified name;

b. the same user is running the Calendar Tool program more than once at the same time, on the same local computer or on two different local computers

The definition here of "same user" is based on Calendar Tool user ID.

When a user attempts to establish a second connection to a particular host, the system displays the error dialog described in Section . The Calendar Tool system does not prevent the same user from connecting to two different hosts at the same time. That is, two different servers running on two different hosts do allow a single user to be connected to both hosts at the same time.

2.6.6.2. Defining the Current Host to Which Admin Commands Apply

Once the user establishes a host connection, the system enables the database and password commands in the regular–user ‘`Admin`’ menu. Theses commands are host–specific, in that they apply to the particular central host that is current. For example, when the user selects the ‘`Users`’ command in the ‘`Admin`’ menu, the system displays the user database entries for the current host.

The definition of "current host" is the connected host that is associated with the current calendar. As defined in Section 2.3.6.4, the "current calendar" is the one at the top of the calendars list on which the user is currently operating. Hence, all commands in the regular–user ‘`Admin`’ menu are enabled when the current calendar is associated with a host, and that host is connected. In summary, to make a host current, the user follows these steps:

a. defines an entry in the host connection table;

b. connects to the host;

c. makes the calendar associated with the host the current calendar.

When the current calendar is not associated with a connected host, or there is no current calendar, then only the ‘`Connect`’ and ‘`Contact Admin`’ items are enabled in the ‘`Admin`’ menu.

Since the results of regular–user ‘`Admin`’ commands are host–specific, they are displayed in separate windows for each host. For example, suppose the user views the user database for host A, then makes host B current, then selects the ‘`Admin Users`’ command again. In response, the system displays a separate window for the user database of host B. The displays for all regular–user ‘`Admin`’ commands operate in this manner, with a separate display window for the results of each command, for each host.

The system does not display multiple copies of the same admin window for a specific host and a specific command. For example, there is at most one copy of the user database window open for any given host. The windowing display style for admin commands is fixed, and not affected by the ‘`Windowing Mode`’ setting in the ‘`View`’ menu (see Section 2.3.6.2).

2.6.6.3. Regular User Access to the Admin Databases

A regular user can view but not modify all three of the Calendar Tool databases. To do so, the user selects one of the top three commands in the regular–user '`Admin`' menu. For example, if the user selects the '`Users`' command, the system displays the dialog shown in Figure 249. The regular–user view of the user database displays the same information as the privileged admin user database dialog in Figure 201, with the following two differences:

a. The banner of the display for the regular–user display is suffixed with the name of the host on which the user database resides.

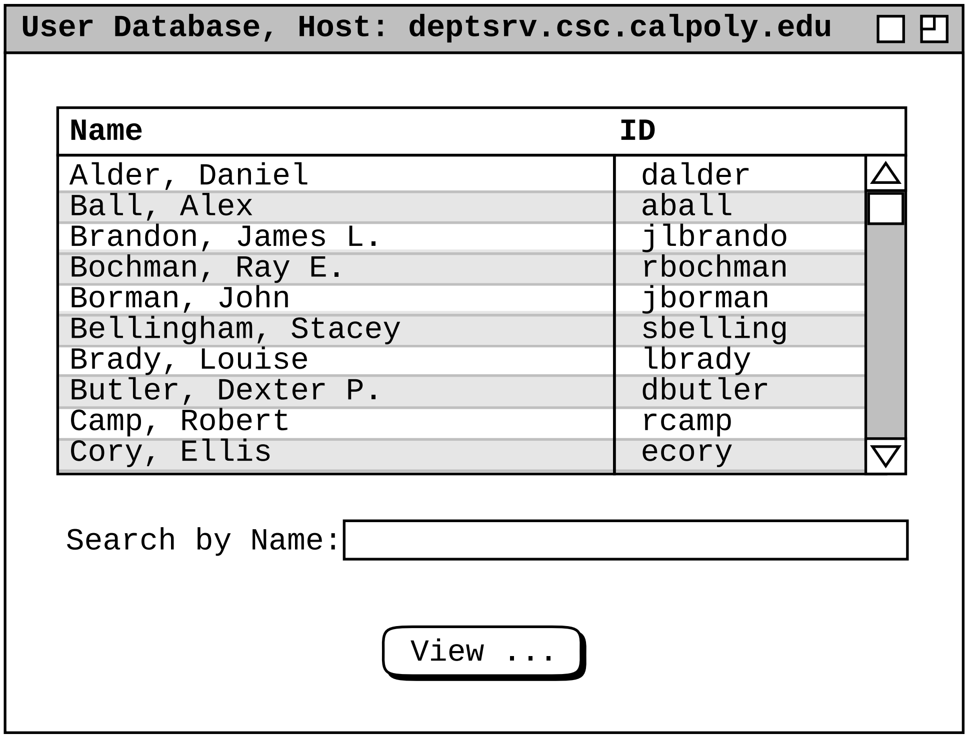
b. The '`Add`' and '`Cancel`' command buttons are missing in the regular–user display of the database.

Aside from these differences, the functionality of the regular–user database display is the same as the privileged admin database dialog, including the searching, sorting, and viewing functionality.

To view a selected user record, the user selects the desired record in the user database display and presses '`View`'. In response, the system displays a dialog of the form shown in Figure 250. This display shows precisely the same information as Figure 206, without any of the command buttons at the bottom of the window. Also, as with the regular–user database display, the banner of the user record display is suffixed with the name of the host on which the user database resides.

In general, all regular–user database display windows have the host name in the window banner. Also, all regular–user database displays lack the following buttons when compared to the privileged administrator dialog versions of the displays: '`OK`', '`Clear`', '`Cancel`', '`Add`', '`Change`', and '`Delete`'. Without '`Add`' buttons, none of the data–entry dialogs for adding database information are available to regular users.

Normally in Calendar Tool display windows, uneditable data fields are shown in light–grey typeface and borders. This is not the case with the regular–user database displays, since none of the fields is ever editable, and they appear more readable in regular type and black borders.

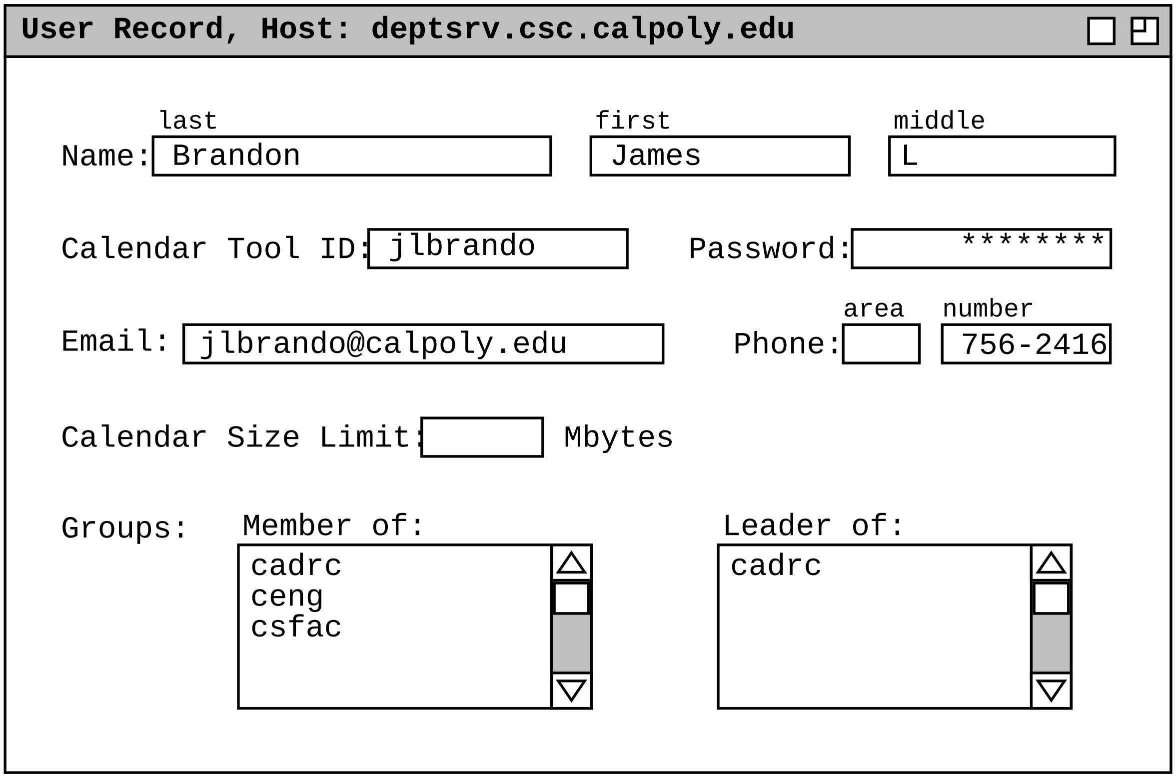


User Database, Host: deptsrv.csc.calpoly.edu

Name	ID
Alder, Daniel	dalder
Ball, Alex	aball
Brandon, James L.	jlbrando
Bochman, Ray E.	rbochman
Borman, John	jborman
Bellingham, Stacey	sbelling
Brady, Louise	lbrady
Butler, Dexter P.	dbutler
Camp, Robert	rcamp
Cory, Ellis	ecory

Search by Name:

Figure 249: Regular user viewing the Users database.



User Record, Host: deptsrv.csc.calpoly.edu

Name: last: Brandon first: James middle: L

Calendar Tool ID: jlbrando

Password: ********

Email: jlbrando@calpoly.edu

Phone: area: number: 756-2416

Calendar Size Limit: Mbytes

Groups:

Member of:
cadrc
ceng
csfac

Leader of:
cadrc

Figure 250: Regular user viewing a user database record.

Except for the password change command described in the next section, a regular user cannot edit any user, group, or location database record, including the user's own. The user can request that an administrator make a change by contacting the administrator with the '`Contact Admin`' command, described in Section 2.6.6.5.

2.6.6.4. Changing the User Password on a Central Host

A registered user's password is the only piece of information stored on the central host that the user can change directly. To change a host password, the user first makes current the host on which the change is to occur, as explained in Section 2.6.6.2. Then the user executes the '`Change Password`' command in the regular–user '`Admin`' menu. In response, the system displays the dialog in Figure 251.

To effect the password change, the user enters the current password, a new password, and then reenters the new password to confirm. As all password values are typed, they are echoed as asterisk characters, not as the actual characters typed. When the user presses '`OK`', the system confirms that the current password is correct and that both versions of the new password are identical. If so confirmed, the system changes the password in the user's database record on the host, as well as in all affected entries in the host connection table (see Section 2.6.6.1). An *affected entry* in the host connection table is one in which the current host is listed as the value of host address in the table entry.

If the current password is incorrect or the versions of the new passwords differ, the system displays an error dialog to that effect, upon dismissal of which the user can try again.

Some clarification on the storage of user password values is in order. For a regular Calendar Tool user, the value of a central host password is stored both on the central host as well as on the user's local computer where the Calendar Tool program runs. The password on the central host is stored in the user's database record. The host database record contains the *defining value* of the password, in that the host password value must be matched when the user initiates a host connection. A password value is also stored on the user's local computer in the host connection table,

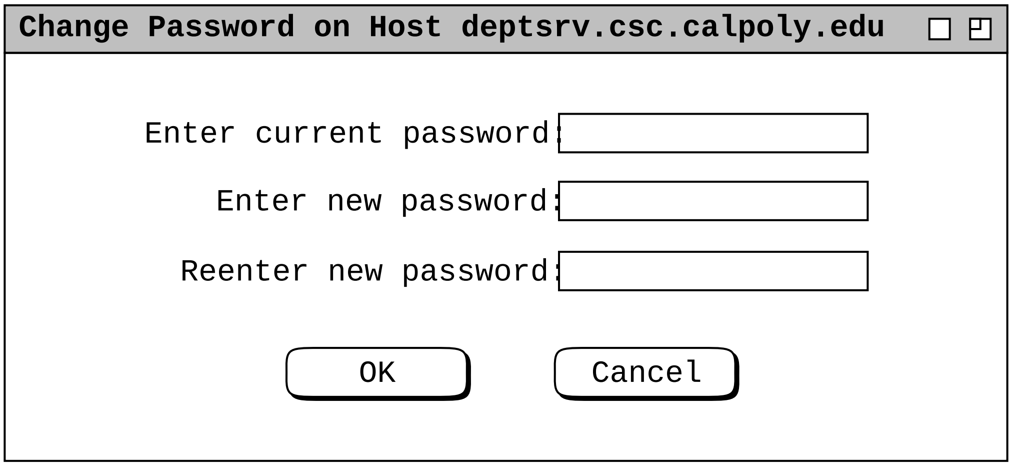


Figure 251: Regular user password change dialog, for a central host password.

as described in Section 2.6.6.1. An entry in the local host connection table contains an *accessing value* of the password, in that this value is the one matched against the host–stored password value in order to validate a connection.

In terms of physical storage, the host password is stored in the user's database record, which is in turn stored in the `Users` component of the host central repository, as described in Section 2.8.8. On the user's local computer, the password is stored in the host–connection table, which is saved permanently in the settings file, as described above. Whenever a password is stored on a file, it is suitably encrypted to prevent readability within the contents of the file.

2.6.6.5. Contacting an Administrator

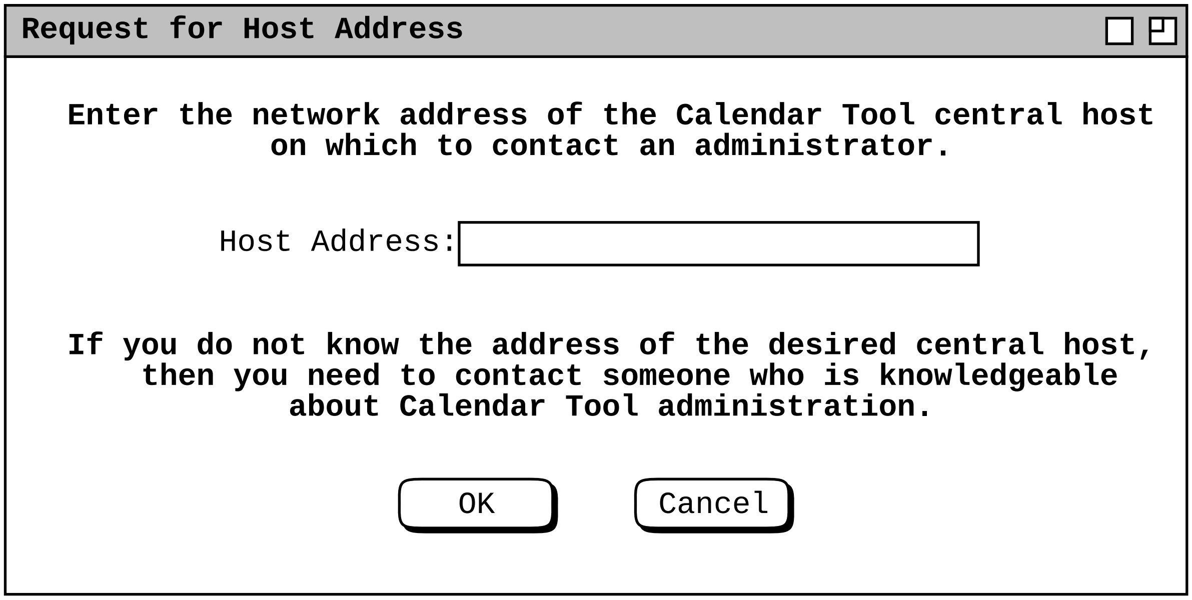
The regular Calendar Tool user may need to contact an administrator for a number of reasons. The reasons include requesting a change to the user's database record; requests to be added or deleted from a Calendar Tool group; or requests for general information about the operation of the Calendar Tool in a particular installed environment. To make any such request, the user selects the '`Contact Admin`' command in the regular–user '`Admin`' menu.

As noted earlier, '`Contact Admin`' is enabled whether or not there is a current host connection. '`Contact Admin`' is still host–specific, like all other `Admin` commands. The system requests a host from the user if no host is current when '`Contact Admin`' is executed. Specifically, if the user executes '`Contact Admin`' when no host is current, the system displays the dialog in Figure 252. The user must enter a valid central host network address in order to proceed. In this context, a valid address is that of a network–accessible computer that is running a Calendar Tool central host server. If the user enters an invalid address, the system displays a failure dialog with the message "The entered host address is not valid.", in response to which the user can dismiss the failure dialog and re–enter a valid host address. There is no limit to the number of retries the user may perform.

When the user enters a valid host address in the request dialog, or when a host is current in the first place when the user executes '`Contact Admin`', the system displays the dialog in Figure 253. When the user presses '`Contact`', the system launches an external email program, with an empty email message addressed to the indicated email address. The name and location of the external email program are set as a user option, as described in Section 2.7.5. After the external email program is launched, the system removes the contact dialog. The email program and Calendar Tool program then execute entirely independently and asynchronously. All subsequent communication through the external email program is outside of the control of the Calendar Tool.

Figure 254 shows the body of a typical email message that a user sends to an administrator.

The administrator's response to this particular request is described in the change scenario of Section 2.6.2. The user is notified of the change as discussed there, in particular Figure 209. Since users contact an administrator with an external email program, an administrator may reply to the contact directly via external email. Such communication

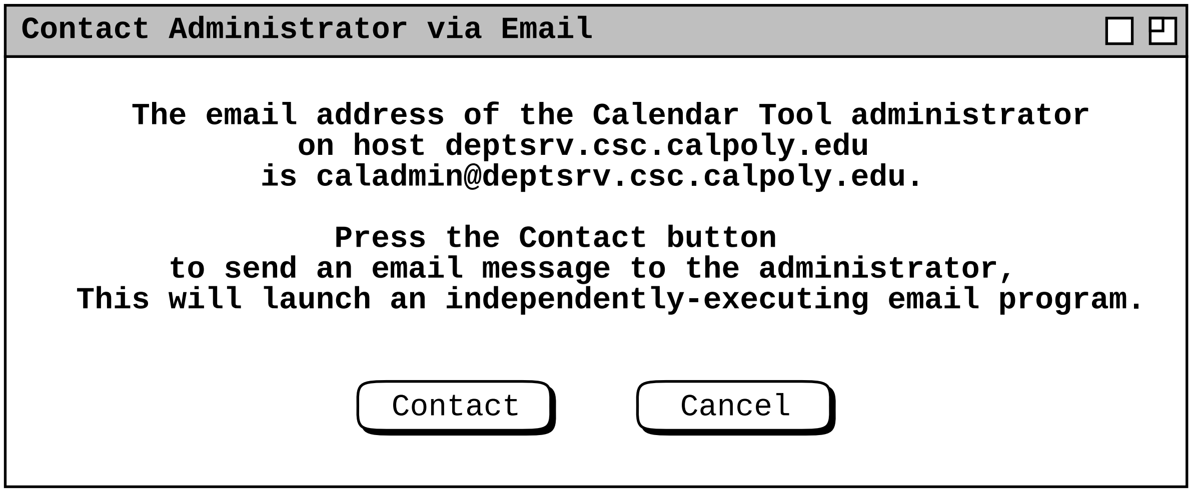


Enter the network address of the Calendar Tool central host on which to contact an administrator.

Host Address:

If you do not know the address of the desired central host, then you need to contact someone who is knowledgeable about Calendar Tool administration.

Figure 252: Host request dialog.



The email address of the Calendar Tool administrator
on host deptsrv.csc.calpoly.edu
is caladmin@deptsrv.csc.calpoly.edu.

Press the Contact button
to send an email message to the administrator,
This will launch an independently-executing email program.

Figure 253: Contact admin dialog.

```
Dear Admin:

I've changed my email address and phone number since the last time I used the
calendar program.  They are jim@onion.csc.calpoly.edu and (805) 756-1897.
Also, I've forgotten my password.  Can you tell me what it is or change it to
something new.

Thanks,
    Jim Brandon
```

Figure 254: Typical admin request.

is outside the control of the Calendar Tool. For managerial requests such as the preceding from user Brandon, the administrator need not reply externally to the request, since the Calendar Tool sends a change notification as part of the change process.

The following is a summary of the administrative communication services provided by the Calendar Tool and Calendar Tool Administration programs:

a. Regular users can contact administrators using the '`Contact Admin`' command, which in turn launches an external, independently–executing email program.

b. An administrator can send general notifications to all users with the '`Notify Users`' command; users receive these notifications in on–screen dialogs, email messages, or both (see Section 2.6.5.4).

c. Calendar Tool Administration sends notifications to affected Calendar Tool users when the administrator performs functions that require notification, as described in Sections 2.6.2 through 2.6.5.

2.6.7. Limitations on Administrator Privileges

As described in the preceding scenarios, Calendar Tool administrators have full access to the three Calendar Tool databases. They also have complete execution control over the Calendar Tool central host server.

Administrators have limited control over user calendars stored in the central host repository. By setting the '`Size Limit`' field in a user database record, the administrator can limit the number of items stored in the central host copy of a user's calendar. The administrator can also move or delete the files in which user calendars are stored on the host.

Administrators have no direct control over the internal content of user calendars stored in the host repository. Specifically, administrators cannot add or change individual scheduled items in the calendars. The only deletion control is indirect, through the setting of the calendar size limit and the systematic deletion of items that may lead to, as explained in Section 2.6.5.5. Per the file encryption requirement described in Section 3.1, administrators cannot meaningfully view the contents of user calendar files, except by running the Calendar Tool in the normal way on the host computer. This prevents administrators, or any other users of the central host computer, from viewing the private portions of user calendars stored in the host repository.

Administrators have limited and indirect control over Calendar Tool operations on users' local computers. Specifically, when a local user is connected to a central host, notifications from an administrator are sent to and received by the Calendar Tool program running on the local computer. When using the Calendar Tool Administration program, administrators have no control whatsoever on Calendar Tool files stored on any users' local computers, or on the contents of any of these files.

A Calendar Tool user who happens to be an administrator may run the regular–user Calendar Tool on a host computer to manage her own personal calendars and to schedule meetings for other users. When running the regular Calendar Tool in this way, the user is not considered to be a administrator at all. Further, the physical computer that happens to be a Calendar Tool central host is considered to be the user's *local* computer when running the regular Calendar Tool program. As explained in Section the distinction between central host computer versus local client computer is a logical distinction, in that the same physical computer can serve both functions. The precise definition of "administrator" is one who is running the Calendar Tool Administration program on any computer where the administration program happens to be installed.

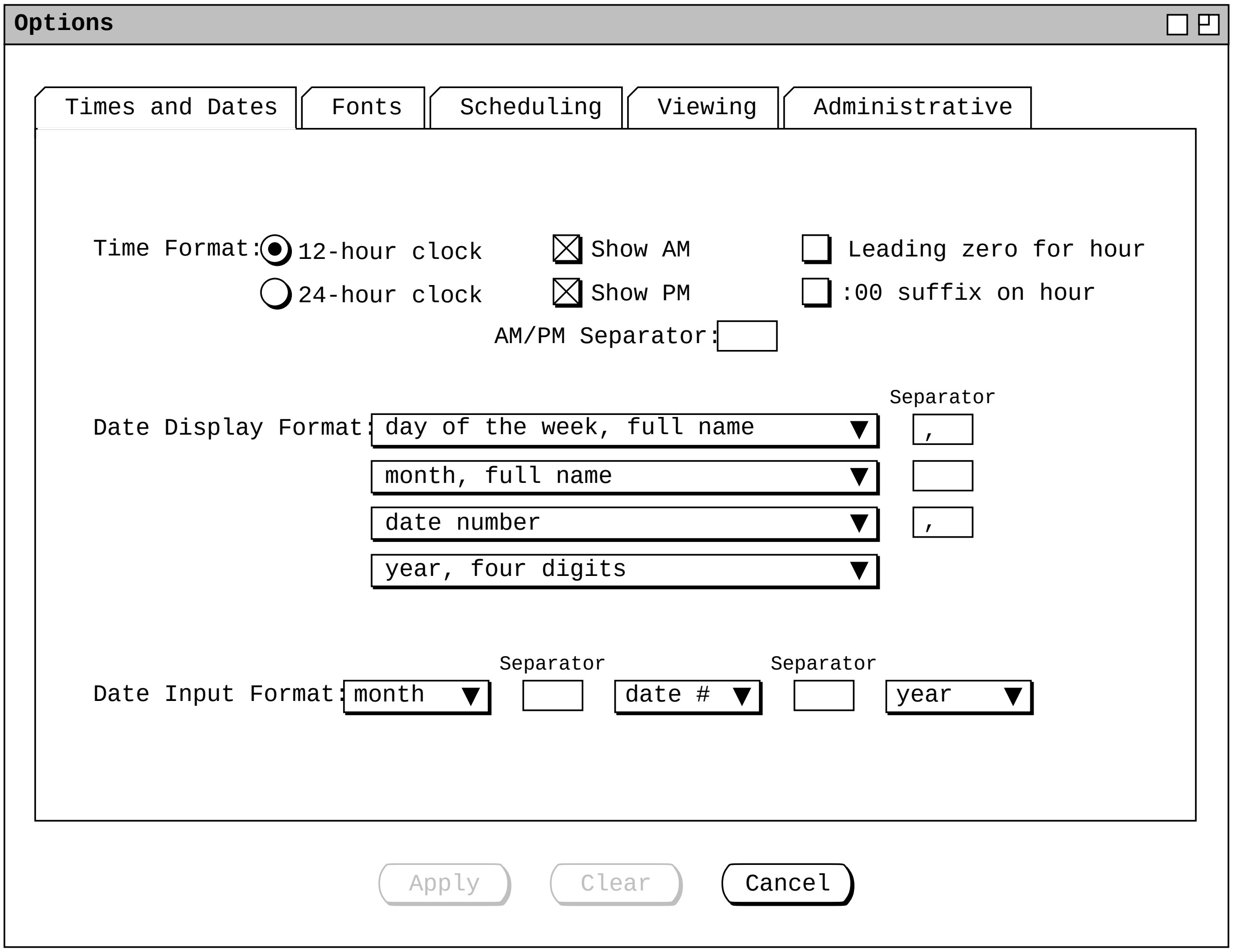
2.7. Options

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.

There are three options-setting dialogs -- two for the regular user of the Calendar Tool and one for the administrative user of the Calendar Tool Administration program. The regular user options are covered first, followed by options for the administrative user.

For both the regular and administration interfaces, the top group of items in the ‘`Options`’ menu correspond one-for-one with tabs in an options dialog. When the user selects one of the menu items, the system displays the dialog with the corresponding tab selected. For example, when the regular user selects the ‘`Times and Dates`’ item in the ‘`Options`’ menu, the system displays the dialog shown in Figure 255. 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, and the effect of the buttons applies to all tabs simultaneously. The ‘`Apply`’ button is used to apply any option changes entered in any of the tabs. ‘`Apply`’ is initially disabled, becoming enabled when the user changes the options settings in one or more



Time Format: ● 12-hour clock ○ 24-hour clock

☒ Show AM

☒ Show PM

☐ Leading zero for hour

☐ :00 suffix on hour

AM/PM Separator:

Date Display Format:	Separator
day of the week, full name ▼	,
month, full name ▼	
date number ▼	,
year, four digits ▼	

Date Input Format: month ▼ Separator: date # ▼ Separator: year ▼

Figure 255: Options dialog.

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 '`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 all unapplied changes and removes the dialog from the screen, without applying any changes.

Most option settings for the regular user 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.

For the regular user, there are four global option settings that are not calendar-specific, but instead apply to the Calendar Tool operating session as a whole. The dialog for these options is displayed using the '`Global`' command in the options menu. These global options are described in Section 2.7.6.

2.7.1. Times and Dates

The time and date options provide formating 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 255. 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 the user selects 24-hour clock, the system disables the show AM, show PM, and AM/PM separator fields, but leaves the settings of these fields unchanged. If the user subsequently re-selects 12-hour clock, the AM/PM checkboxes and separator field are re-enabled, and retain the settings from when the 12-hour setting was last in effect.

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 11 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.

All four date-display list boxes have the same list of items, which is shown in Figure 256. The list items specify the 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 12 shows the three alternatives. In all cases, the day name is capitalized.

Date Context	Components
Date text fields in all dialogs and displays	month, date, year
Date label at the top of an item view	day, month, date, year
Date label at the top of a day view	day, month, date, year
Date label at the top of a week view	month, date range, year
Date label at the top of a month view	month, year
Month name within year view	month
Date columns in list views	month (short or number), date, year

Table 11: Date display contexts.

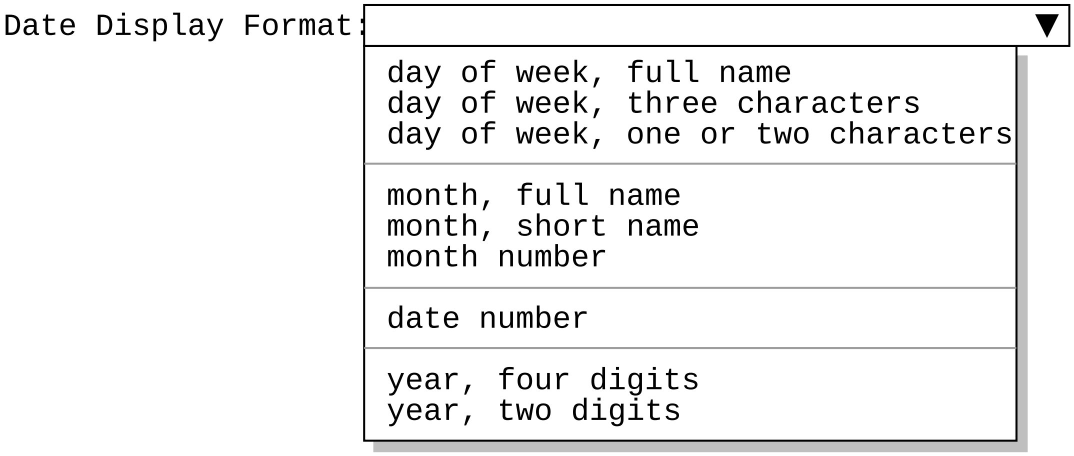


Figure 256: Date format combo box lists.

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 12: Day of the week formats.

The fourth through sixth items specify the month name. The full name is the complete spelling of the month, capitalized. The short name is the first three letters of the month, capitalized. The month number is between 1 and 12.

The seventh list item is for the date number. It is an integer between 1 and 28 to 31, depending on the number of days in the selected month.

The last two date-display 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, which ever 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, month, date, and year. The values in the three boxes in the ‘`Separator`’ column 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 formated 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 .

Figure 257 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 258 shows the result of applying these changes on selected display excerpts. These excerpts illustrate all of the date contexts listed in Table 11.

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 259. 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.

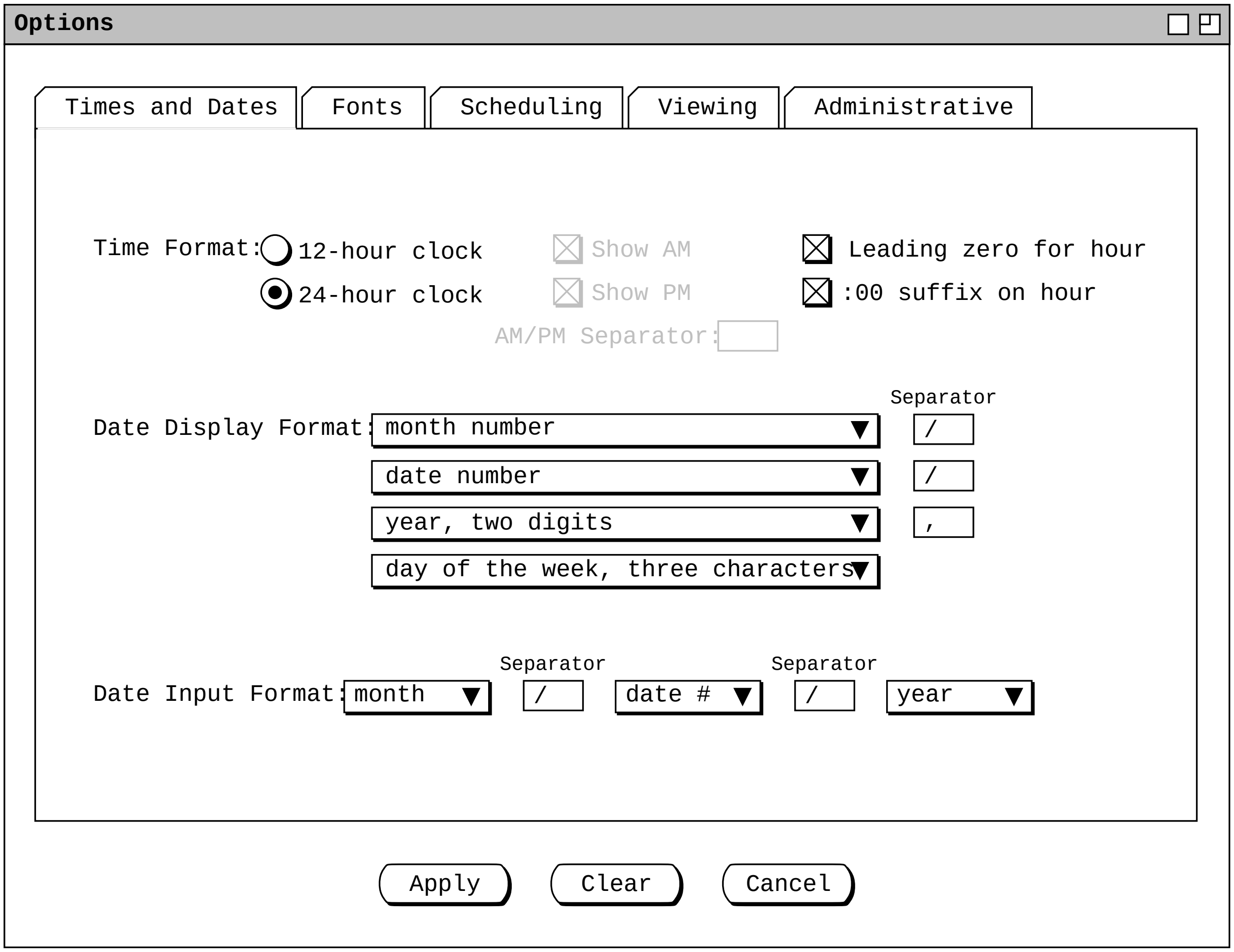
The same two data-entry fields are available for both forms of text. The ‘`Name`’ and ‘`Size`’ menus are platform-specific lists of available fonts and their sizes. With only a few exceptions, the same relative font size is used throughout all Calendar Tool display windows. The exceptions are the following:

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

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 coloring being black except for the coloring defined by categories, as defined in Section 2.5.5.

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

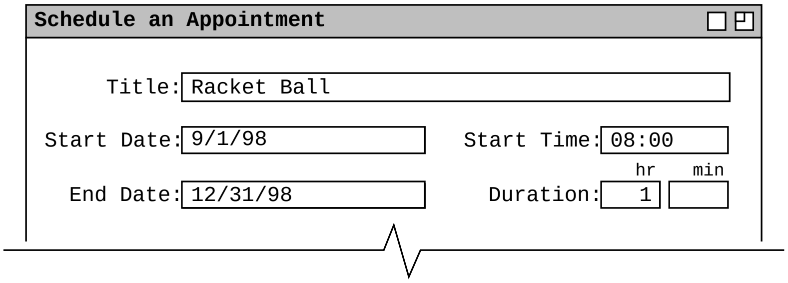


Time Format: ○ 12-hour clock ☒ Show AM ☒ Leading zero for hour
● 24-hour clock ☒ Show PM ☒ :00 suffix on hour
AM/PM Separator:

Date Display Format: month number ▼ Separator /
date number ▼ /
year, two digits ▼ ,
day of the week, three characters ▼

Date Input Format: month ▼ Separator / date # ▼ Separator / year ▼

Figure 257: Changes to time and date options.



Schedule an Appointment

Title: Racket Ball

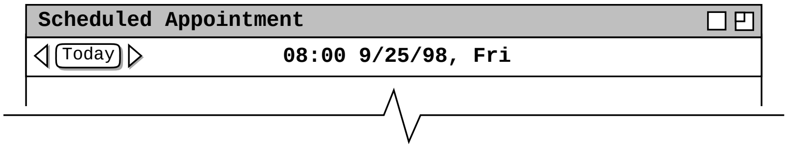
Start Date: 9/1/98

Start Time: 08:00

End Date: 12/31/98

Duration: 1 hr, min

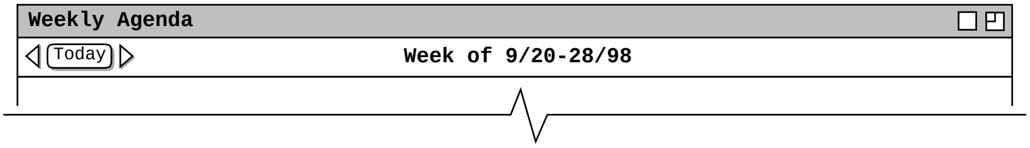
a. In a scheduling dialog.



Scheduled Appointment

08:00 9/25/98, Fri

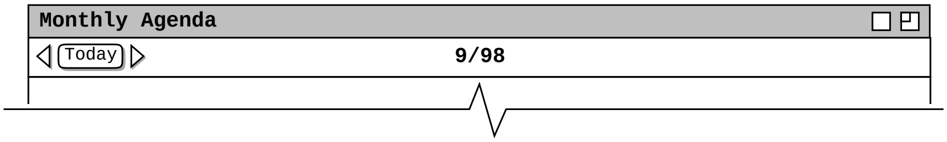
b. In an item-level view (same format in day view).



Weekly Agenda

Week of 9/20-28/98

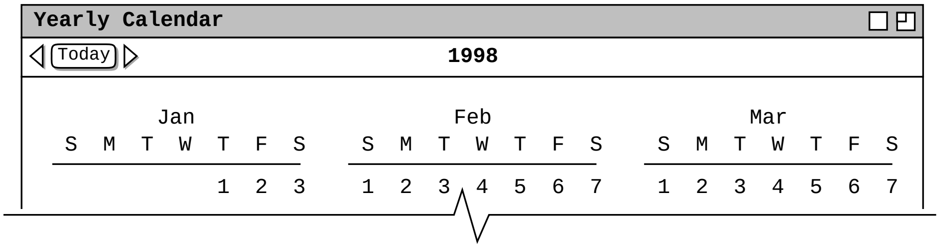
c. In a week view.



Monthly Agenda

9/98

d. In a month view.



Yearly Calendar

1998

Jan

S	M	T	W	T	F	S
				1	2	3

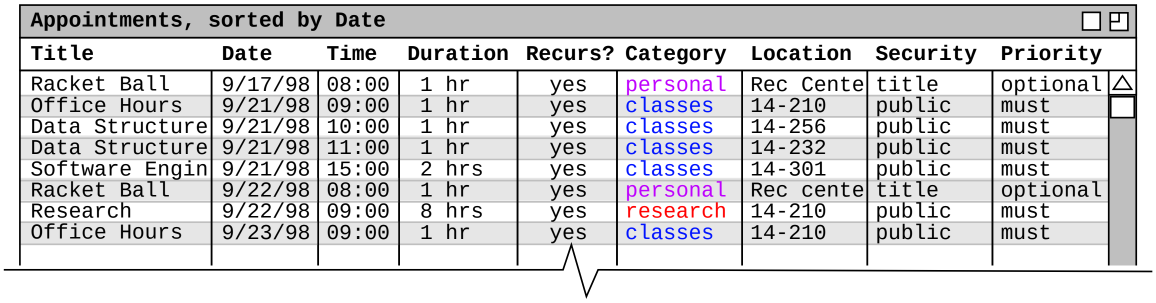
Feb

S	M	T	W	T	F	S
1	2	3	4	5	6	7

Mar

S	M	T	W	T	F	S
1	2	3	4	5	6	7

e. In a year view.

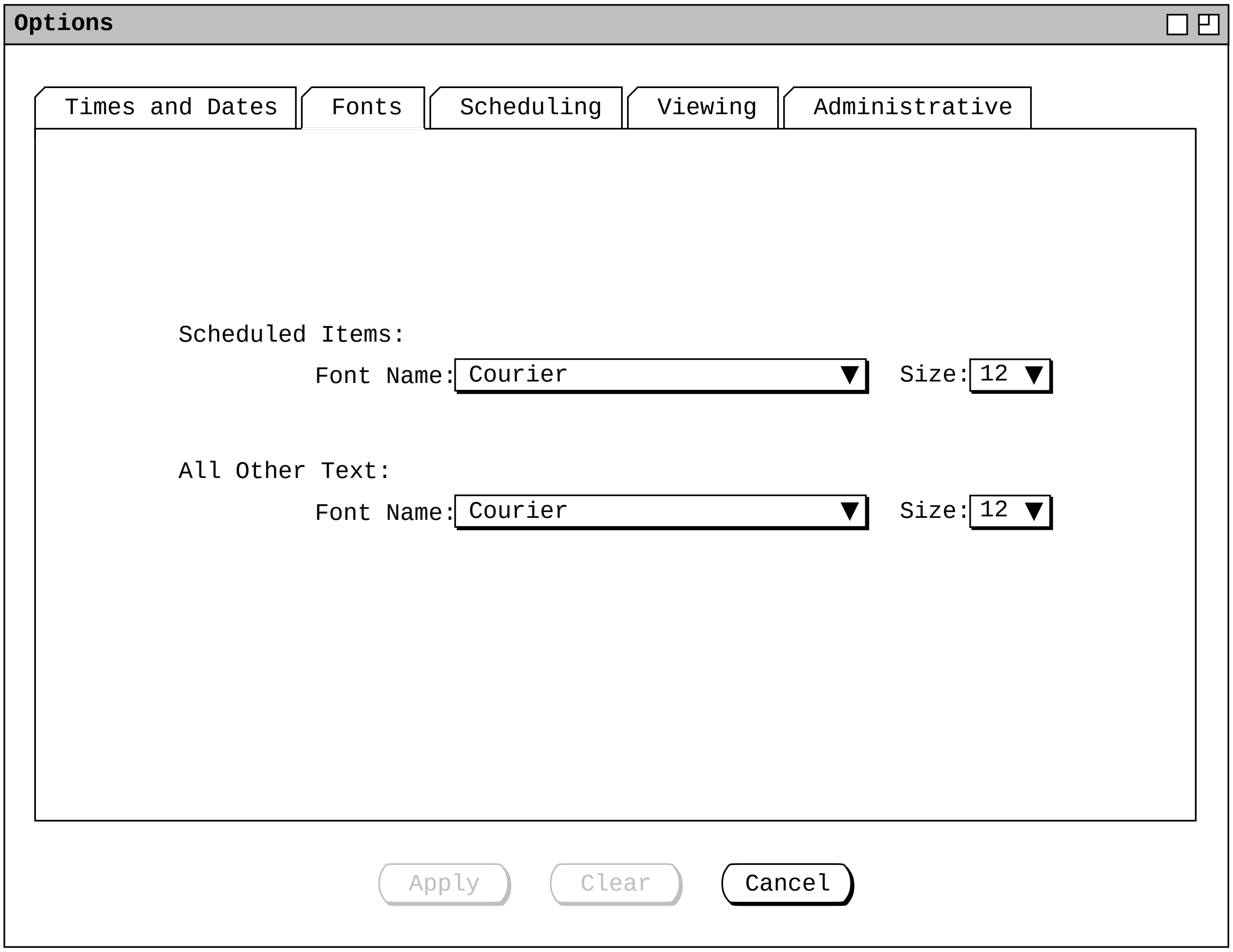


Appointments, sorted by Date

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 258: Effects of time and date option changes on different displays.



Scheduled Items:
Font Name: Courier Size: 12

All Other Text:
Font Name: Courier Size: 12

Figure 259: Fonts options tab.

the Calendar Tool.

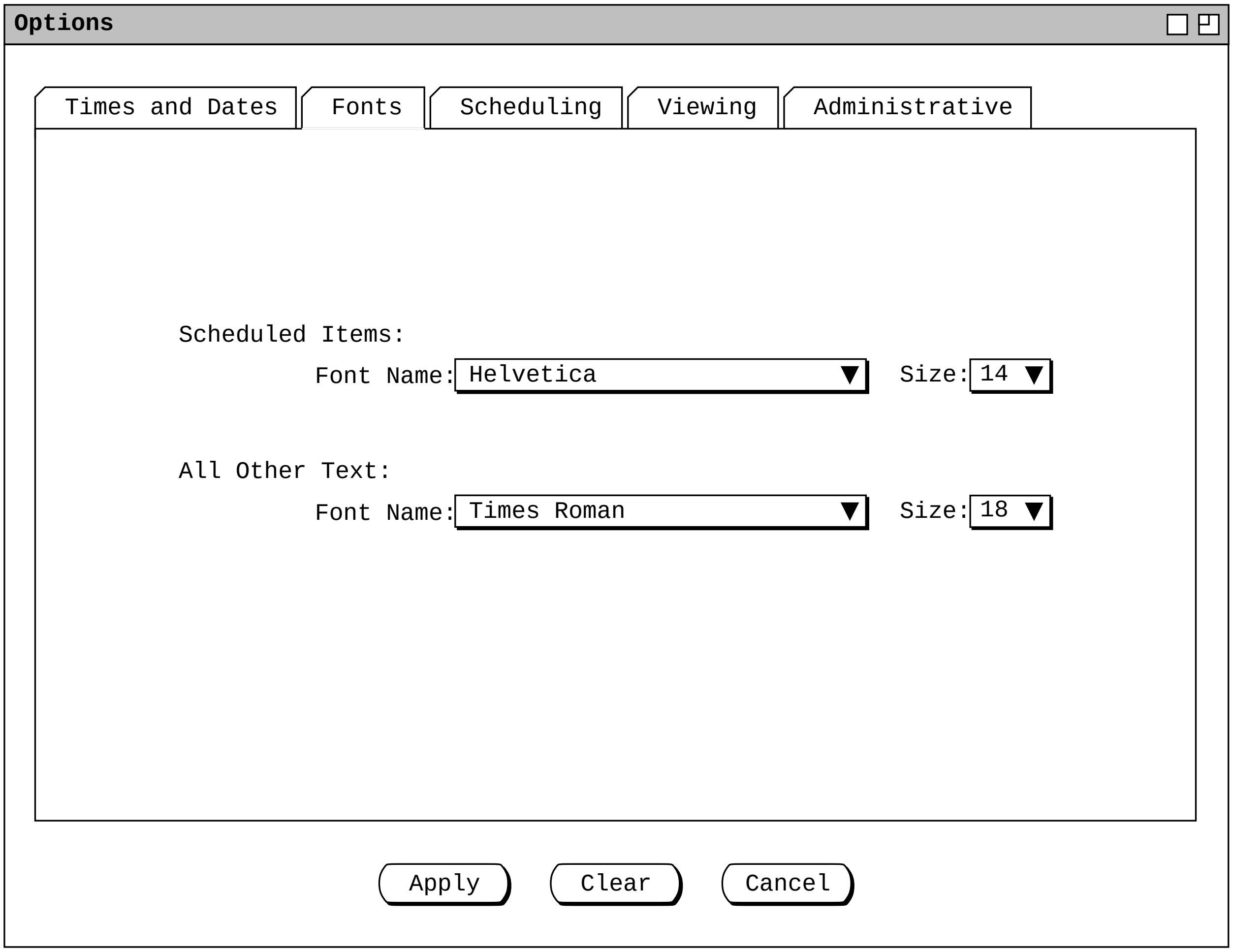
Figure 260 shows the user having changed font option settings. Figure 261 shows the result of applying these changes on the monthly view shown originally in Figure 21. The scheduled item text is displayed in Helvetica-14 and all other text is in Times-Roman-18. Font changes may involve size changes to some or all of the components in a display window. Details of window (re)sizing are covered in Section .

2.7.3. 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 262. 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

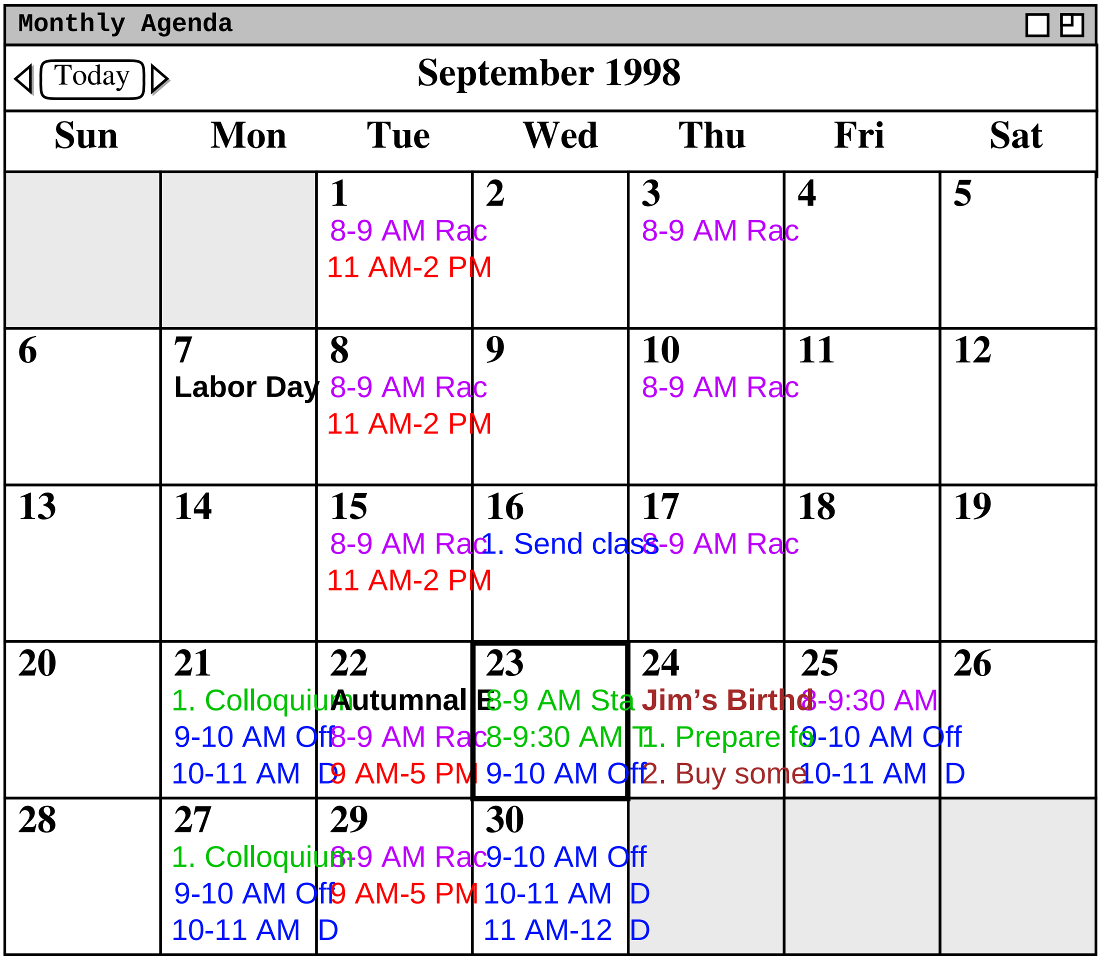
The '`Defaults`' tab allows the user to control the default values that appear in scheduling dialogs when they are initially displayed and when the user presses '`Clear`'. To set the default values for a scheduling dialog, the user presses one of the four dialog-type buttons to bring the dialog up. When 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 263.



Scheduled Items:
Font Name: Helvetica
Size: 14

All Other Text:
Font Name: Times Roman
Size: 18

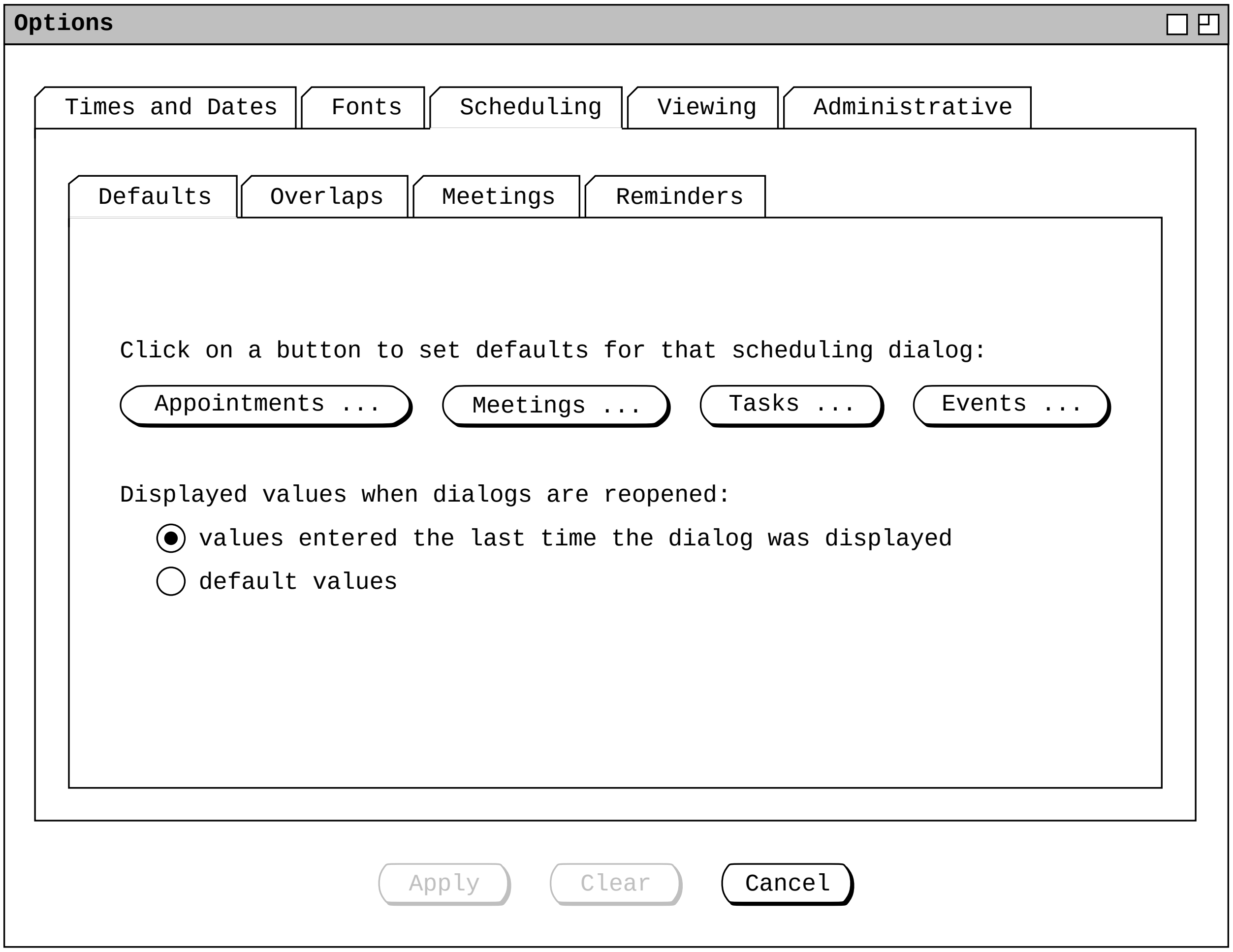
Figure 260: Changes to font options.



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 1. Send class	17 8-9 AM Rac	18	19
20	21 1. Colloqui 9-10 AM Off 10-11 AM D	22 Autumnal E 8-9 AM Rac 9 AM-5 PM	23 8-9 AM Sta 8-9:30 AM T 9-10 AM Off	24 Jim's Birthd 1. Prepare fo 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 261: Monthly view with font changed applied.

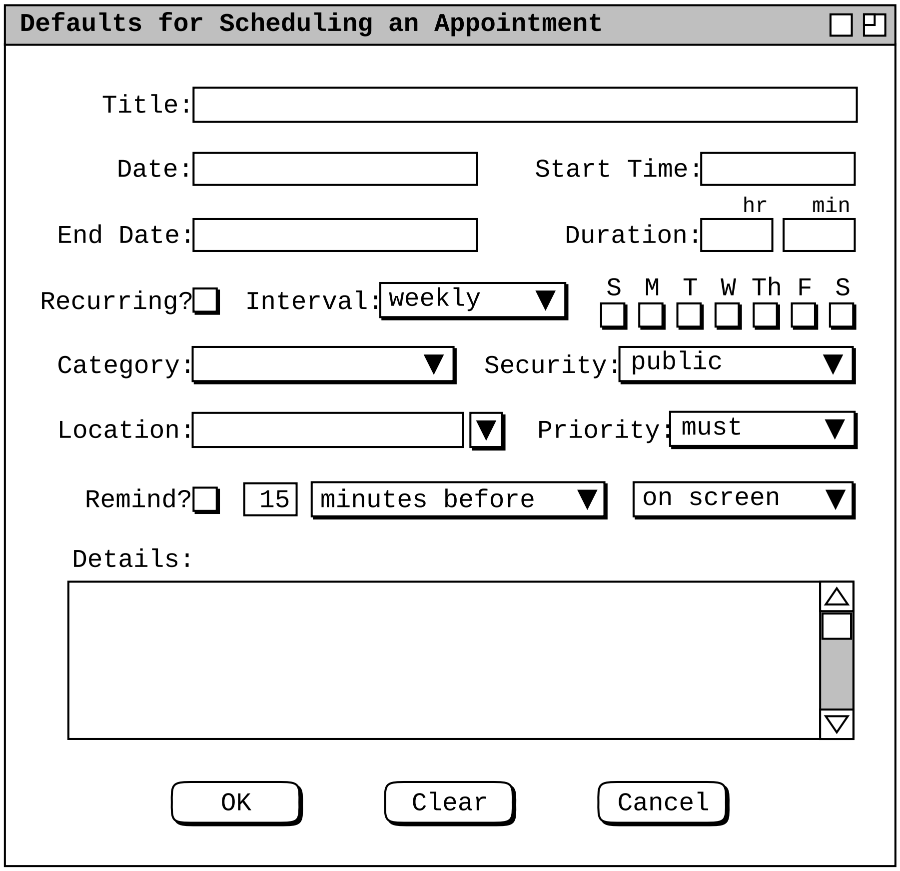


Click on a button to set defaults for that scheduling dialog:

Displayed values when dialogs are reopened:

● values entered the last time the dialog was displayed
○ default values

Figure 262: Scheduling options tab.



Defaults for Scheduling an Appointment

Title:

Date:

Start Time:

End Date:

Duration: hr min

Recurring? ☐

Interval: weekly

S ☐ M ☐ T ☐ W ☐ Th ☐ F ☐ S ☐

Category:

Security: public

Location:

Priority: must

Remind? ☐ 15 minutes before on screen

Details:

Figure 263: Appointments scheduling defaults dialog.

This dialog has exactly the same format as the regular appointment scheduling dialog shown in Figure 5. 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:

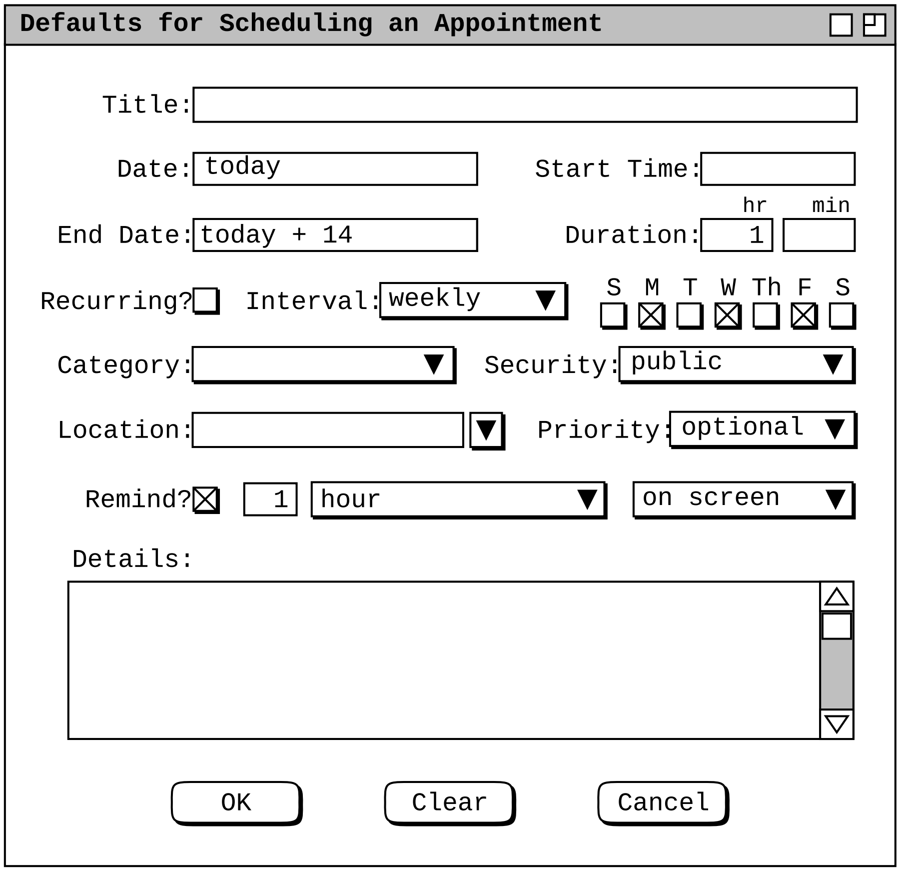
a. The word "`today`" can appear in date fields of a default-setting dialog, meaning the default value is dynamically replaced with the current date whenever the scheduling dialog is opened.

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.

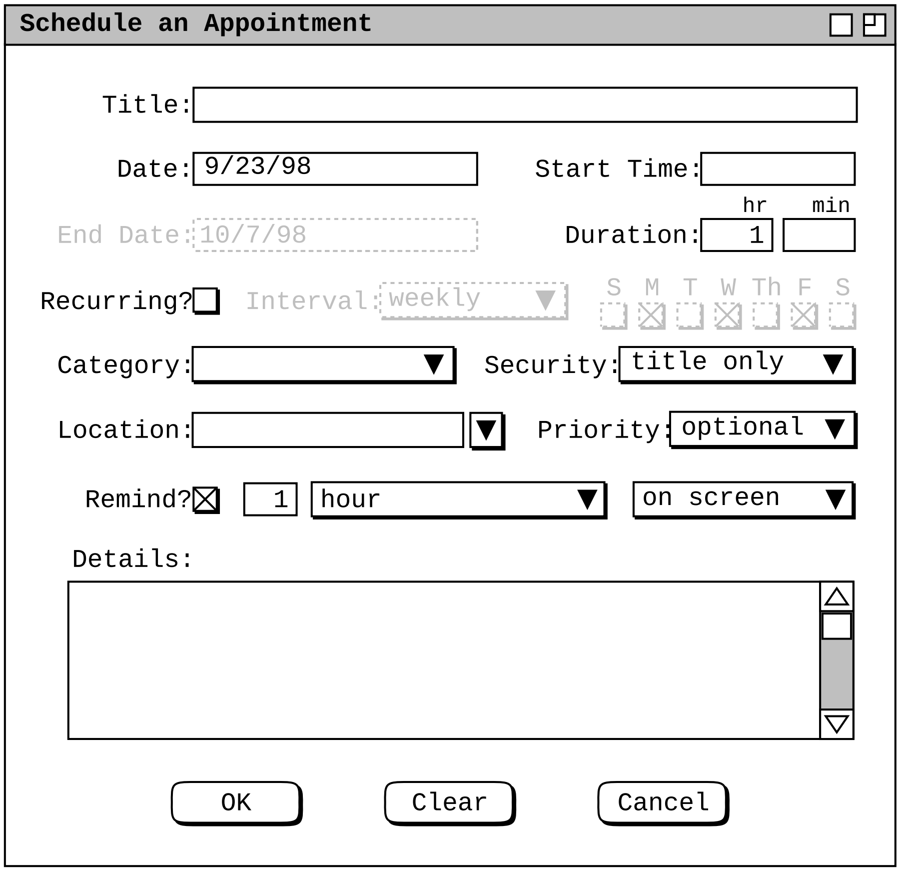
Figure 264 shows the result of the user setting some appointment dialog defaults. The "`today`" value in the '`Date`' field defaults to today's date. If the appointment is recurring, the '`End Date`' value of "`today + 14`" defaults to the today's date plus 14 days. The duration default is set to 1 hour. 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`'.

When the user presses '`OK`' in Figure 264, 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 265 shows the default state of the appointments dialog after the user has applied the settings in Figure 264. The



Title:
Date: today
Start Time:
Duration: 1 hr, min
End Date: today + 14
Recurring? ☐
Interval: weekly
S ☐ M ☒ T ☐ W ☒ Th ☐ F ☒ S ☐
Category:
Security: public
Location:
Priority: optional
Remind? ☒ 1 hour on screen
Details:

Figure 264: Appointments scheduling defaults dialog with values filled in.



Schedule an Appointment

Title:

Date: 9/23/98 Start Time:

End Date: 10/7/98 Duration: hr 1 min

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

Category: Security: title only

Location: Priority: optional

Remind? ☒ 1 hour on screen

Details:

Figure 265: Appointments dialog with updated default settings.

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 265 also reflects the changes to date format options made in Figure 257.

The built-in default values for all scheduling dialogs are defined when the calendar tool is configured for installation (see Section). Table 13 shows the initial default values for all fields in all four types of scheduled item. Figures 5, 83, 123, and 127 illustrate the initial default settings for each type of scheduled item.

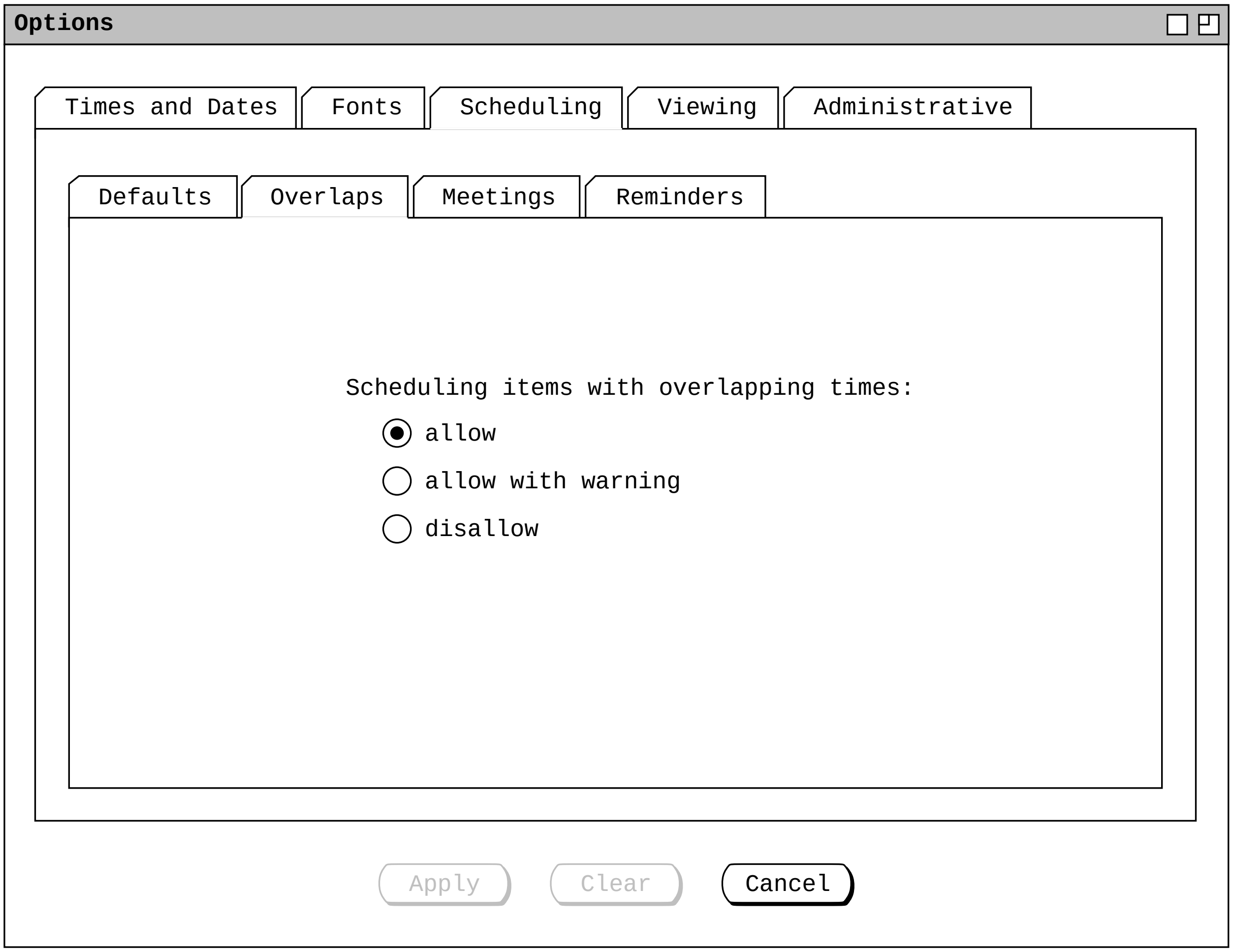
The two radio buttons in the scheduling defaults tab (Figure 262) are used to select the values that appear in scheduling dialogs when they are reopened after having been opened one or more times previously. The first selection specifies that values in reopened dialogs are those that were entered by 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.

2.7.3.2. Scheduling Overlaps Options

When the user selects the ‘`Overlaps`’ tab, the system updates the options dialog as shown in Figure 266. 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 the rules for item uniqueness

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 13: Default field values for all four types of scheduled item.



Scheduling items with overlapping times:

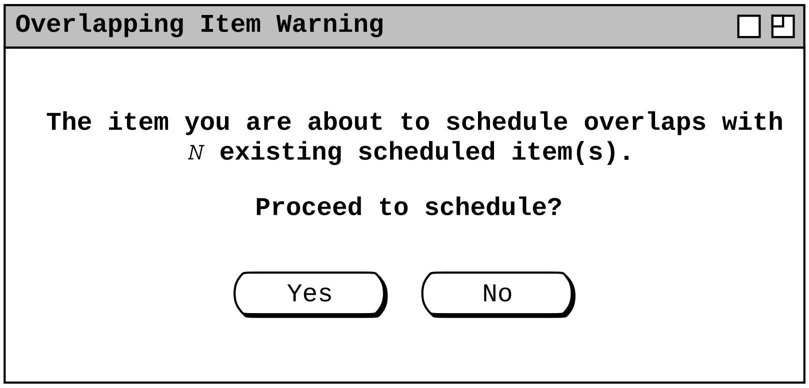
● allow

○ allow with warning

○ disallow

Figure 266: Scheduling overlaps options tab.

in Section 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 267. The number *N* is the number of existing scheduled items with which the new item overlaps, and the text



The item you are about to schedule overlaps with *N* existing scheduled item(s).

Proceed to schedule?

Figure 267: Overlapping item warning.

"**`item(s)`**" is singular if $N = 1$, plural otherwise.

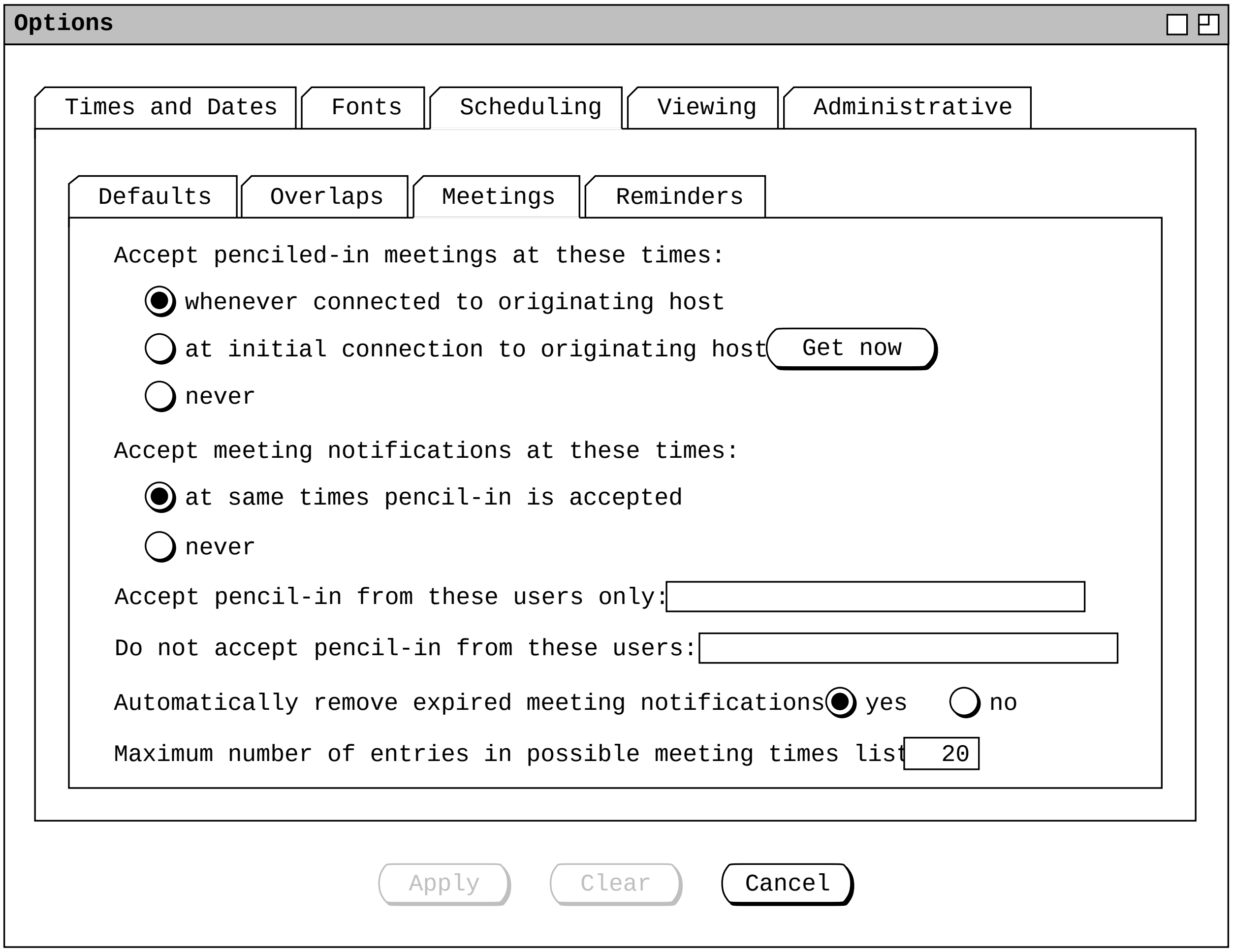
If ‘`disallow`’ is selected for overlapping items, then the system treats the scheduling of such items as an error. The details of scheduling error messages are covered in Section Complete details on the effects of allowing overlapping items are covered in Sections 2.3.1. and

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 2.5.2.2.3.

2.7.3.3. Scheduling Meetings Options

When the user selects the ‘`Meetings`’ tab, the system updates the options dialog as shown in Figure 268. The effects of the first two option settings, ‘`Accept penciled-in ...`’ and ‘`Accept meeting notifications ...`, are explained fully in Section 2.4.1.5.

The third and fourth meeting-scheduling options control from whom penciled-in meetings and notifications are accepted. Both fields contain comma-separated lists of user IDs, per the syntax described in Section . If both ID lists are empty, then penciled-in meetings, and the accompanying notifications, are accepted from all users, except as



Accept penciled-in meetings at these times:

● whenever connected to originating host

○ at initial connection to originating host

○ never

Accept meeting notifications at these times:

● at same times pencil-in is accepted

○ never

Accept pencil-in from these users only:

Do not accept pencil-in from these users:

Automatically remove expired meeting notifications ● yes ○ no

Maximum number of entries in possible meeting times list 20

Figure 268: Scheduling meetings options tab.

excluded by a '`never`' setting in the previous two meeting-scheduling options. If one or more user IDs are specified under '`Accept pencil-in from these users only`', then penciled-in meetings are accepted only from those listed users and from no others. If one or more user IDs are specified under '`Do not accept pencil-in from these users`', then meetings are not accepted from those listed users, but from all other users not otherwise excluded by the other option settings. The contents of the '`Accept ...` and '`Do not accept` ID lists must be mutually exclusive, that is the same ID may not appear in both lists. The effect of excluding a user with these settings means that any meetings scheduled by those users are automatically declined, without ever appearing on central host or local calendars.

These 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 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 269. 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.

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

The '`Default Redisplay Settings`' in this options tab apply to the pop-up reminder dialog that appears when the time for the reminder has come. Default settings for the time the reminder dialog appears initially are defined in the scheduling defaults options, described in Section 2.7.3.1.

2.7.4. Viewing Options

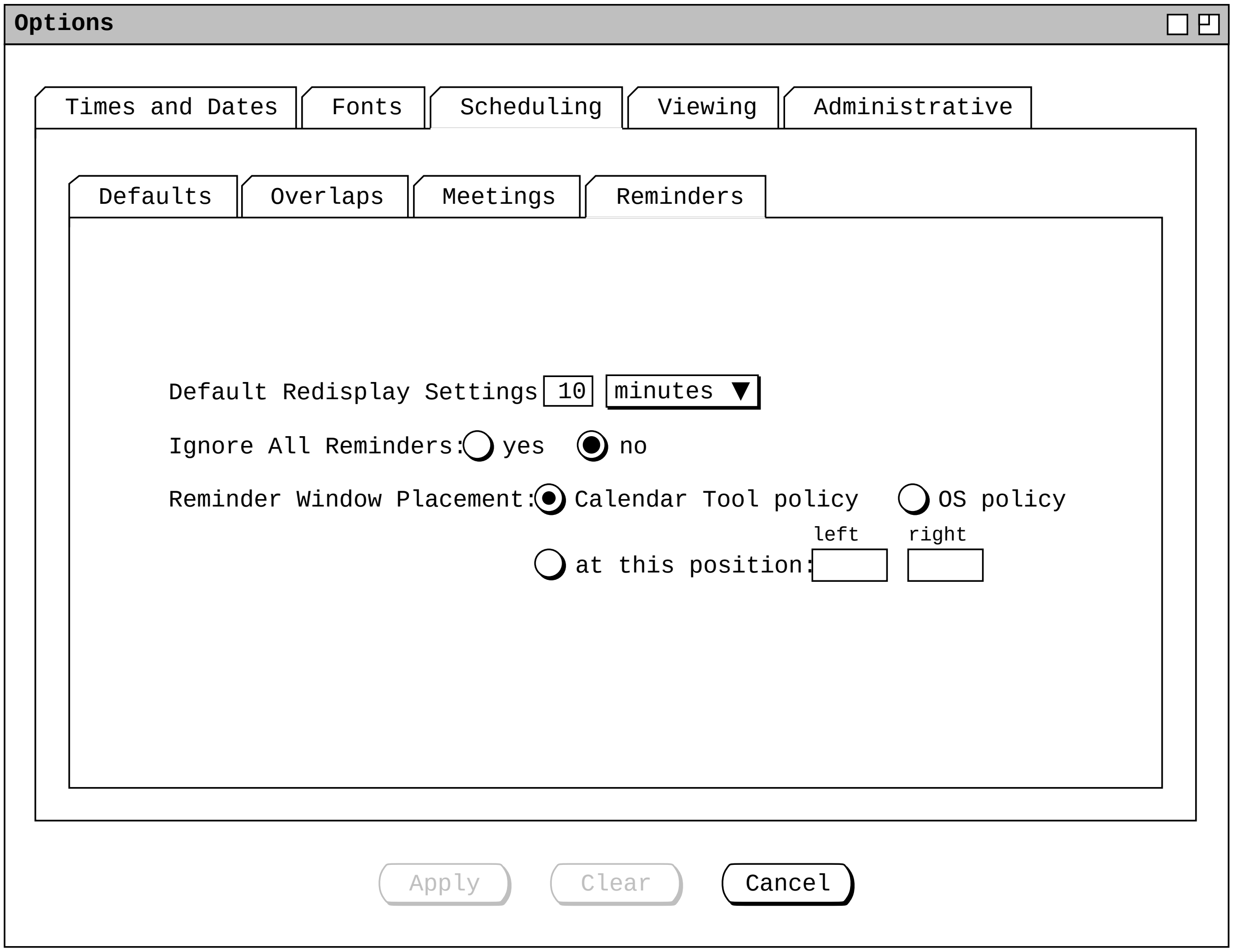
The viewing options provide control over the commands in the '`View`' menu. When the user selects the '`Viewing`' tab in the options dialog, the system updates the display as shown in Figure 270. The second tier of viewing tabs provide option settings for each each of the five calendar viewing levels, 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 .

The viewing options for the day through year levels are described and illustrated fully in Sections 2.3.1.1 through 2.3.1.4, respectively. Figures 271 through 274 show the presentation of these settings in the the options dialog. Only title, day range, and width/height apply to list style.

Details of list, window, and miscellaneous options follow in the next three subsections.

2.7.4.1. Viewing Lists Options

Viewing lists options provide control over the commands in the '`View Lists`' submenu. When the user selects the '`lists`' tab within the '`Viewing`' tab, the system updates the display as shown in Figure 275. All five subtabs for list viewing are similar in format. Each contains a subset of the information in the custom lists dialog, presented in Section 2.3.3.6. The settings in each subtab allow the user to define the default format for each of the five types of lists -- appointments, meetings, tasks, events, and all items. The specific meaning of the subtab settings is as



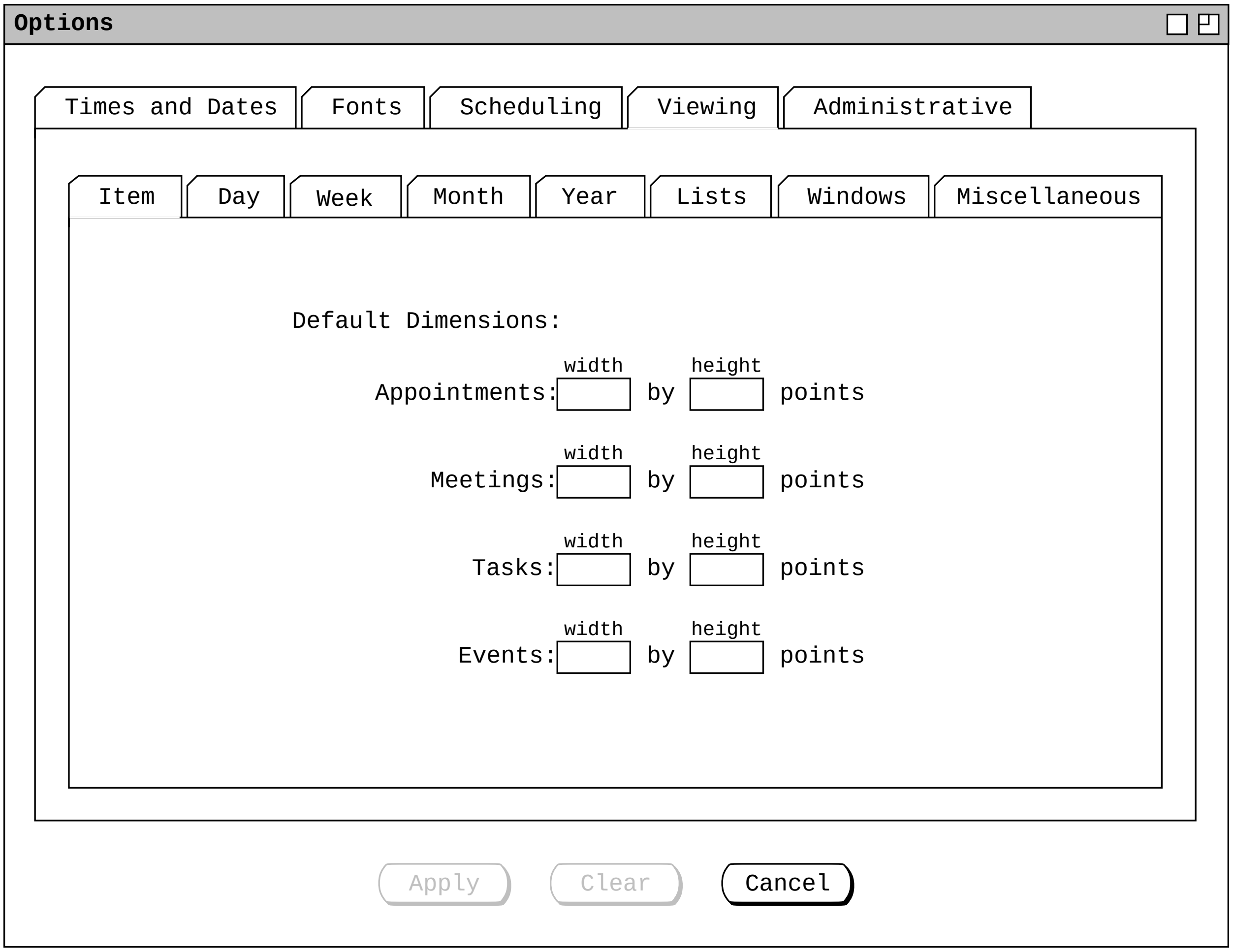
Default Redisplay Settings 10 minutes ▼

Ignore All Reminders: ○ yes ● no

Reminder Window Placement: ● Calendar Tool policy ○ OS policy

○ at this position: left ☐ right ☐

Figure 269: Scheduling reminders options tab.



Default Dimensions:

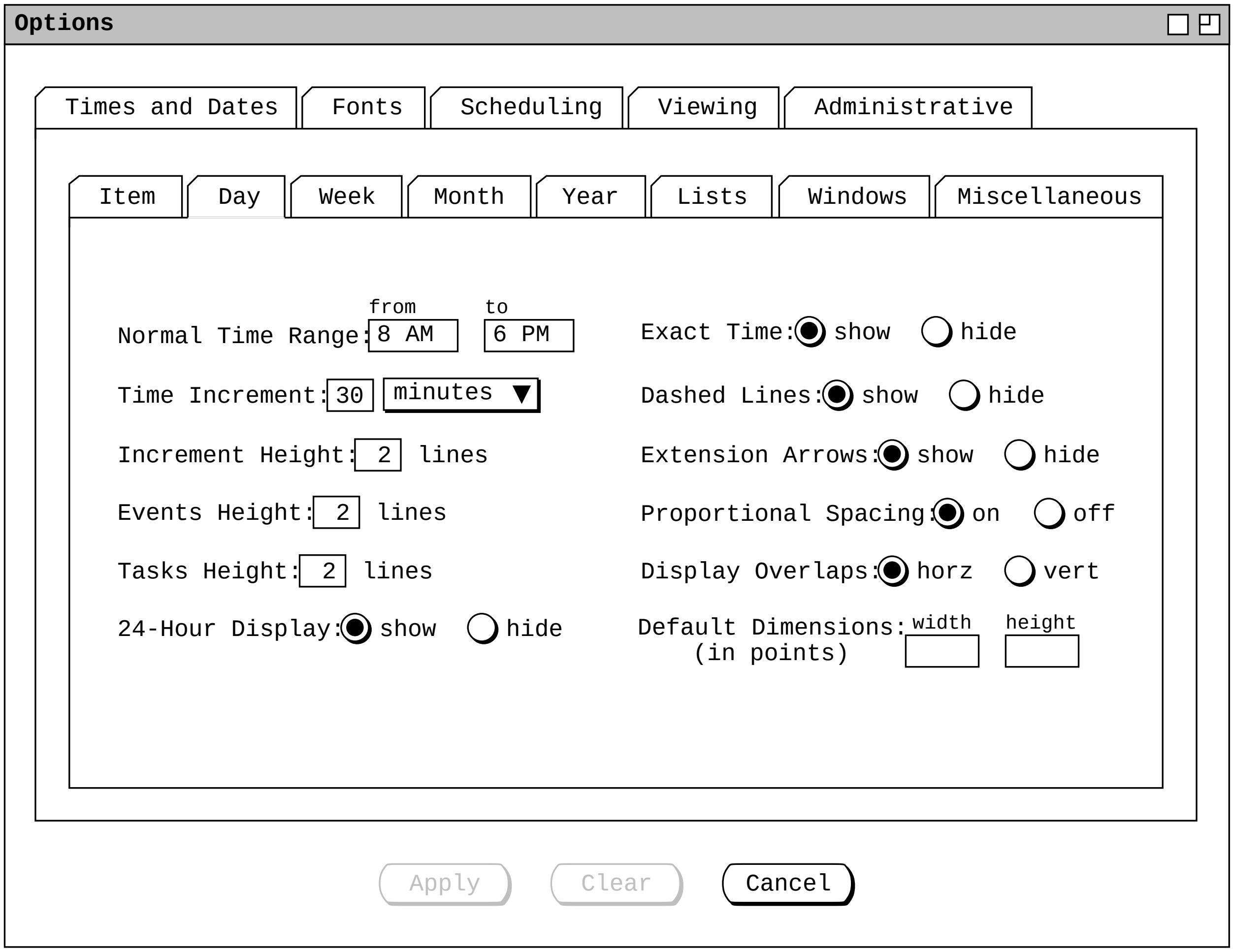
Appointments: width ☐ by height ☐ points

Meetings: width ☐ by height ☐ points

Tasks: width ☐ by height ☐ points

Events: width ☐ by height ☐ points

Figure 270: Viewing options tab.



Normal Time Range: from 8 AM to 6 PM

Time Increment: 30 minutes ▼

Increment Height: 2 lines

Events Height: 2 lines

Tasks Height: 2 lines

24-Hour Display: ● show ○ hide

Exact Time: ● show ○ hide

Dashed Lines: ● show ○ hide

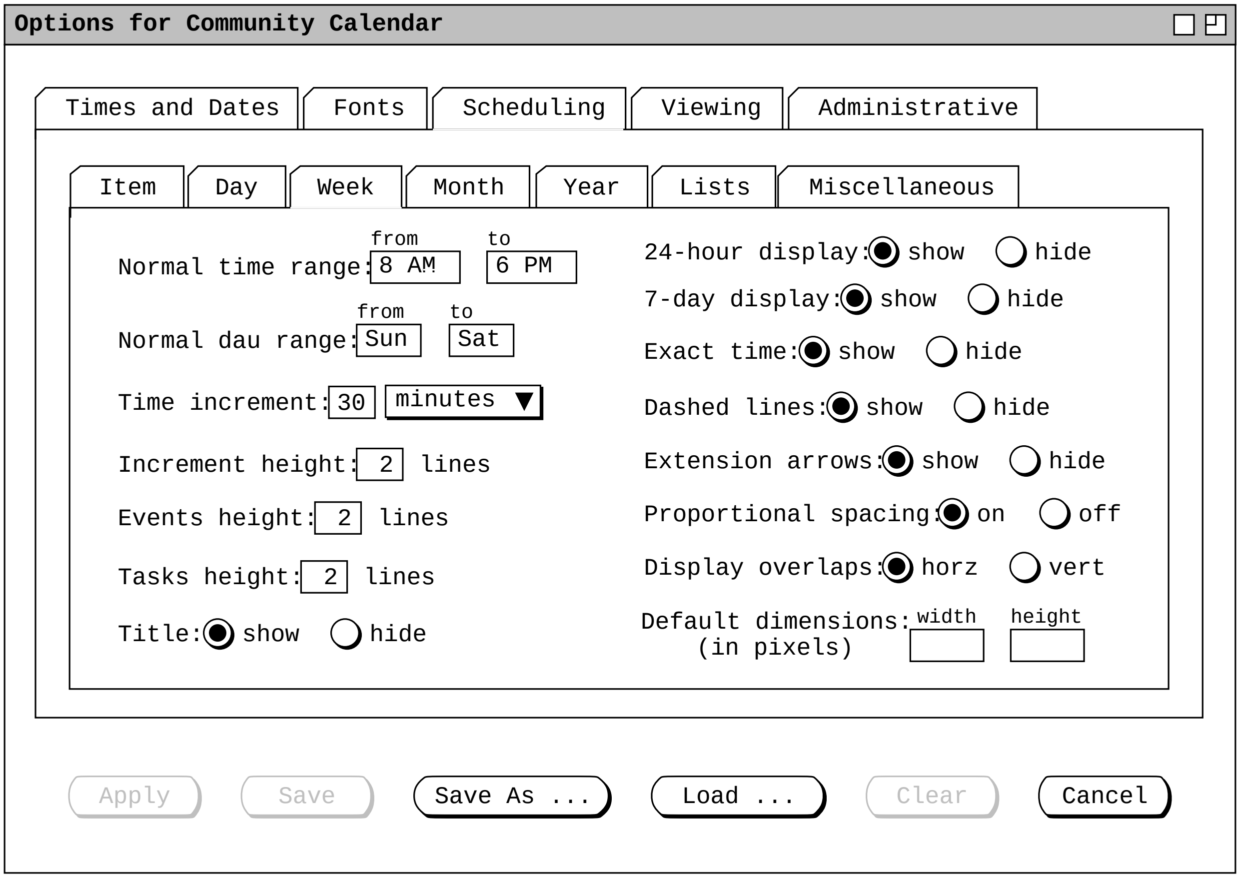
Extension Arrows: ● show ○ hide

Proportional Spacing: ● on ○ off

Display Overlaps: ● horz ○ vert

Default Dimensions: (in points) width height

Figure 271: Viewing day options tab.



Normal time range: from 8 AM to 6 PM

Normal dau range: from Sun to Sat

Time increment: 30 minutes ▼

Increment height: 2 lines

Events height: 2 lines

Tasks height: 2 lines

Title: ● show ○ hide

24-hour display: ● show ○ hide

7-day display: ● show ○ hide

Exact time: ● show ○ hide

Dashed lines: ● show ○ hide

Extension arrows: ● show ○ hide

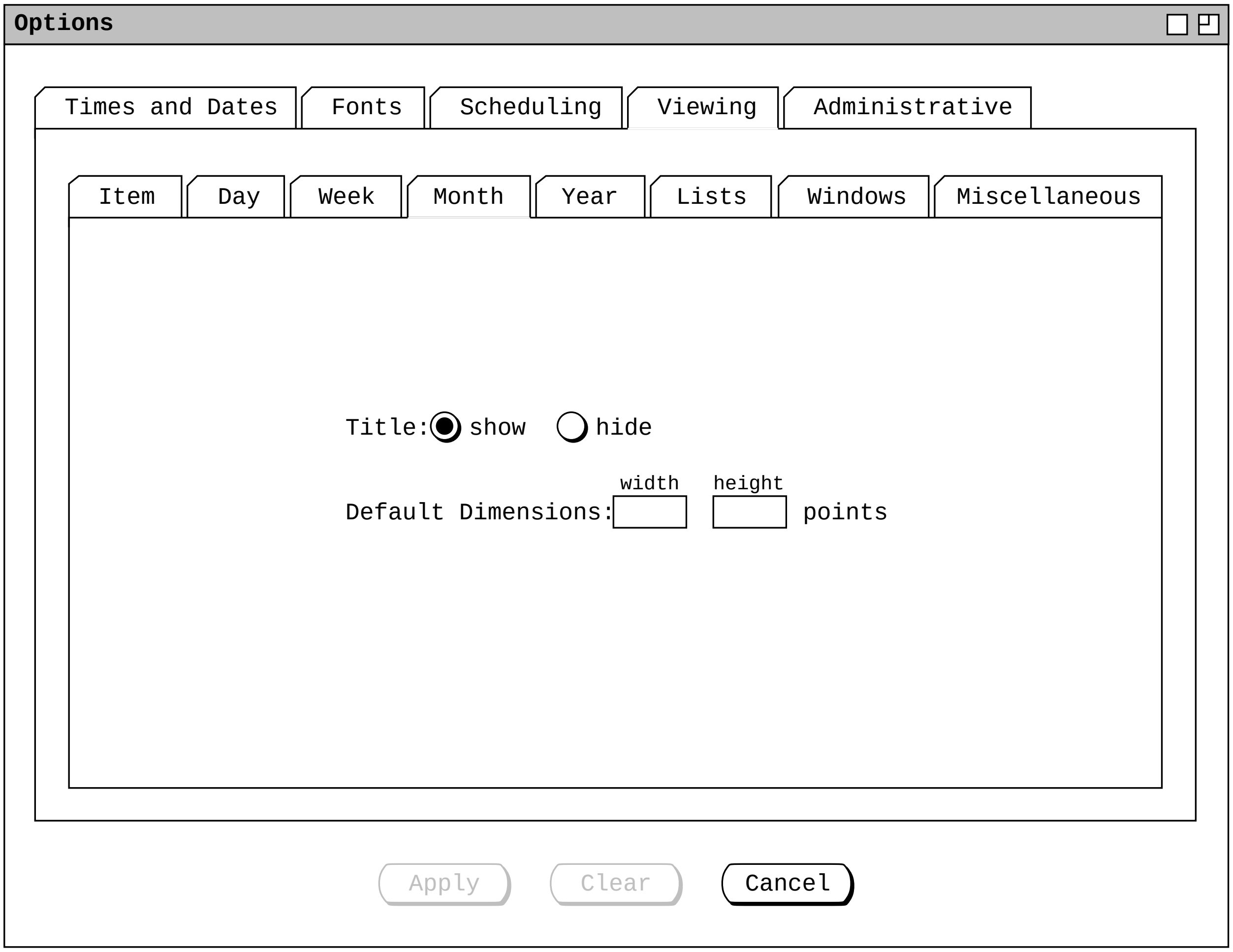
Proportional spacing: ● on ○ off

Display overlaps: ● horz ○ vert

Default dimensions: width height
(in pixels)

Make other options-viewing
windows the same size
as this one.

Figure 272: Viewing week options tab.



Title: ● show ○ hide

Default Dimensions: width ☐ height ☐ points

Figure 273: Viewing month options tab.

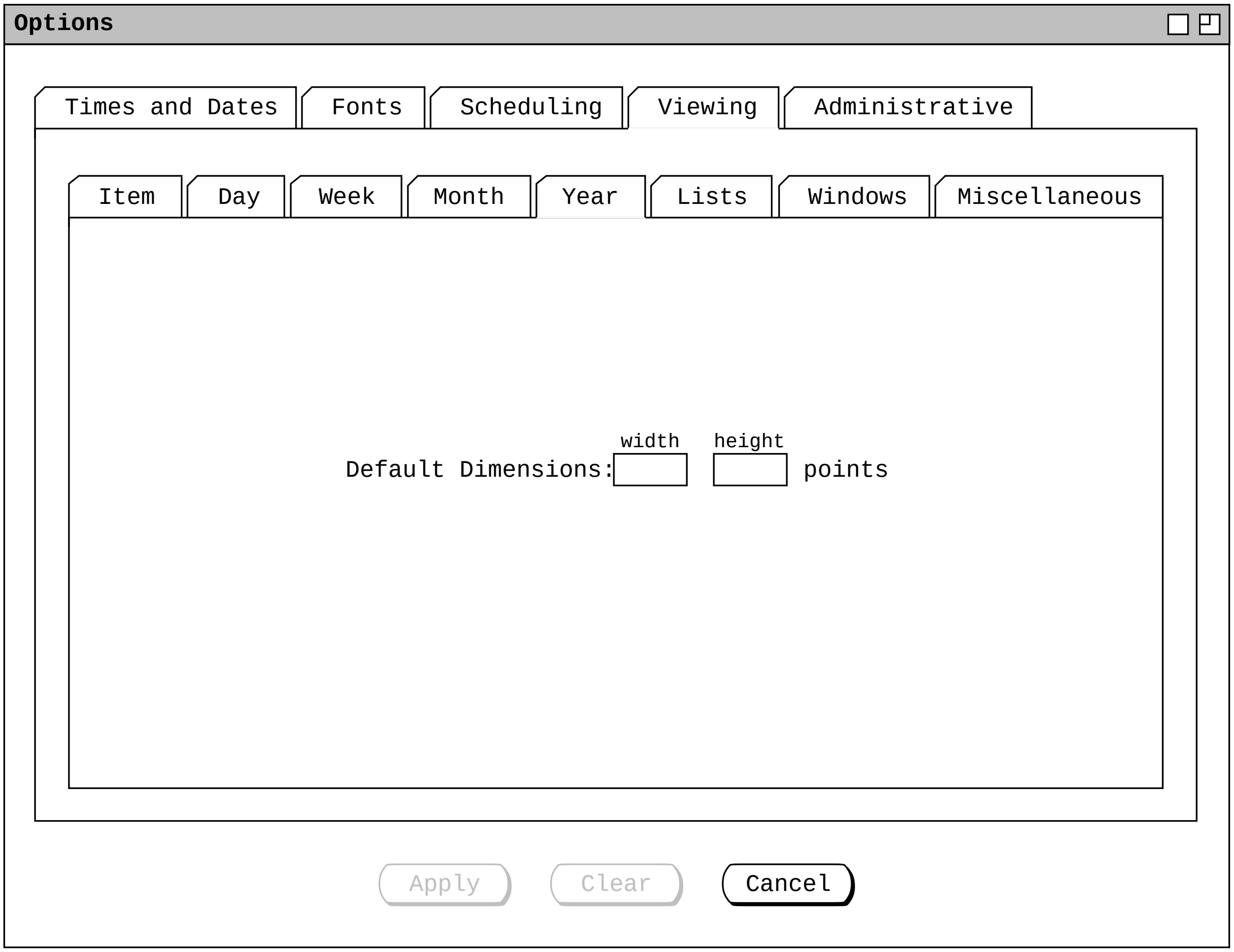
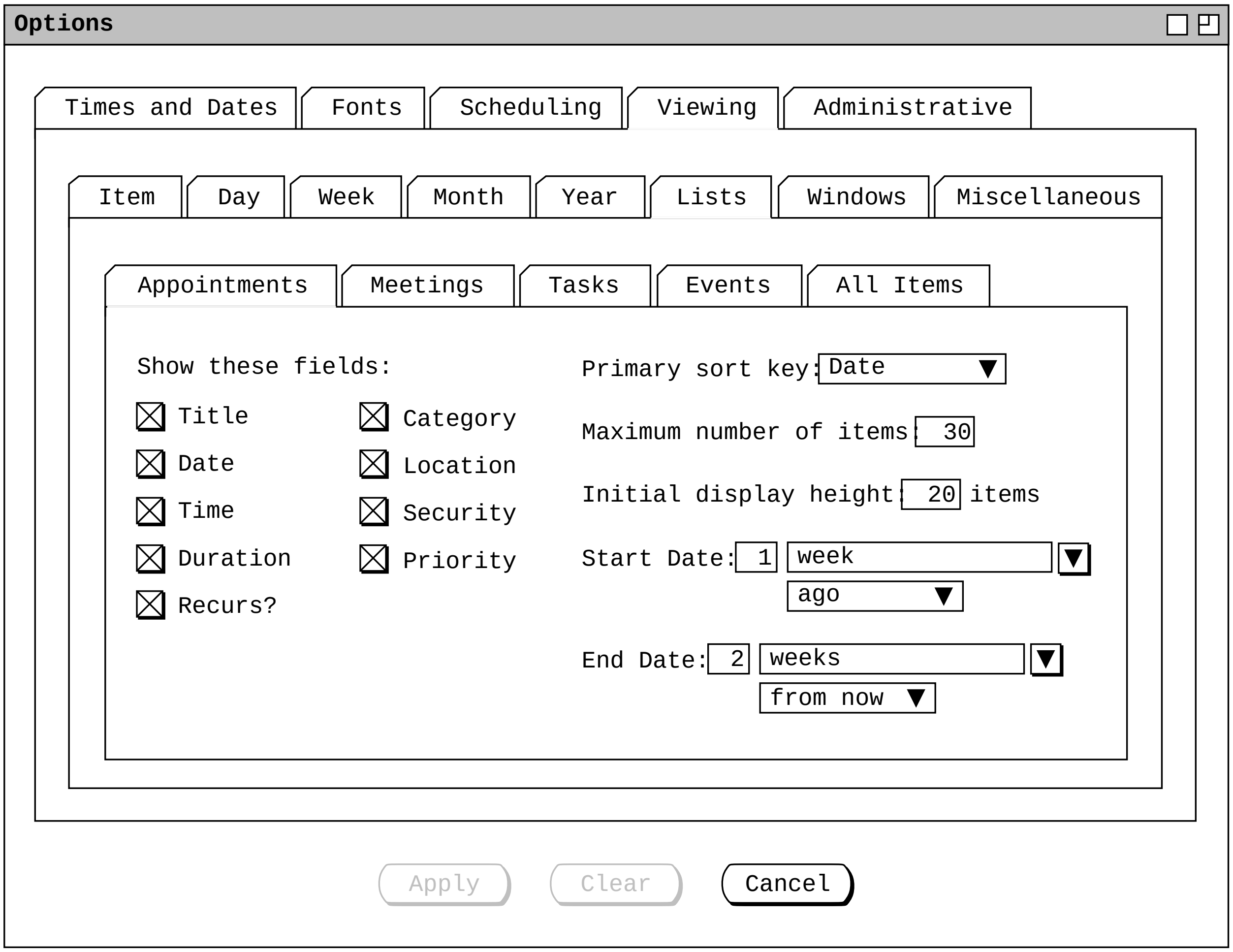


Figure 274: Viewing year options tab.



Show these fields:

☒ Title
☒ Date
☒ Time
☒ Duration
☒ Recurs?
☒ Category
☒ Location
☒ Security
☒ Priority

Primary sort key: Date ▼

Maximum number of items: 30

Initial display height: 20 items

Start Date: 1 week ▼ ago ▼

End Date: 2 weeks ▼ from now ▼

Figure 275: Viewing lists options tab.

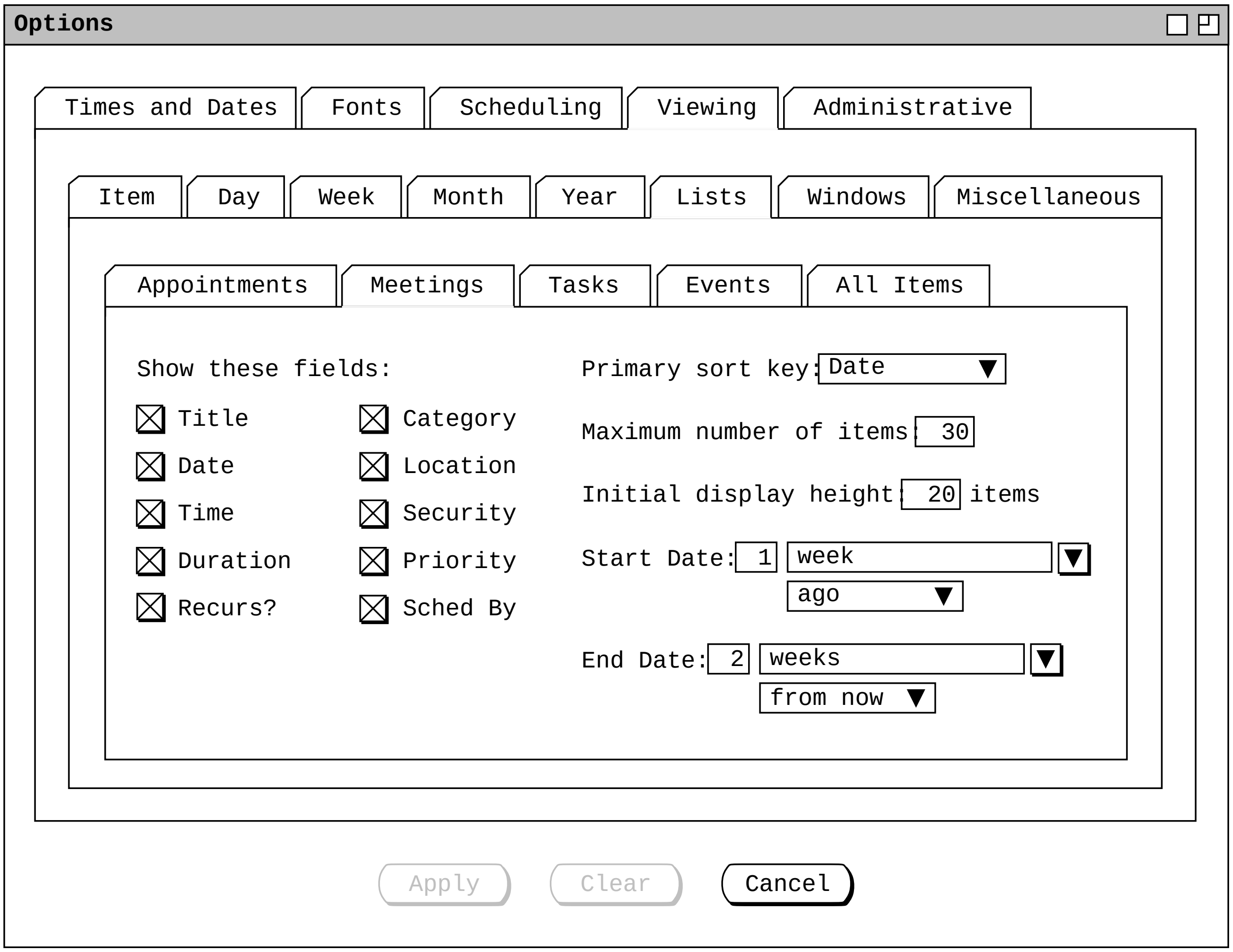
described in Section 2.3.3.6. The difference between the list viewing options compared to the custom list settings is that the options apply to all lists that are displayed by the top five the commands in the '`View Lists`' menu (Figure 37), whereas the custom list settings apply to a single specific custom list. To be clear, the list viewing options apply to all lists displayed with the top five '`View Lists`' menu commands, never to a custom list. In contrast, the view settings for any particular custom list apply to that list only, not to any list displayed with the top five menu commands, nor to any other custom list.

Figures 276 through 279 show the other four subtabs for viewing lists options.

2.7.4.2. Viewing Windows Options

When the user selects the '`Windows`' tab within the '`Viewing`' tab, the system updates the display as shown in Figure 280.

The '`Initial Windowing Mode`' specifies the default setting for the window modes described in Section 2.3.6.2. The items in the windowing-mode option menu are the same as in the windowing-mode submenu shown in Figure 71 -- Two-Window, Per-Level, and Multi-Window. The default is Per-Level. The effect of each of these settings is described fully in Section 2.3.6.2.



Show these fields:

☒ Title
☒ Date
☒ Time
☒ Duration
☒ Recurs?
☒ Category
☒ Location
☒ Security
☒ Priority
☒ Sched By

Primary sort key: Date ▼

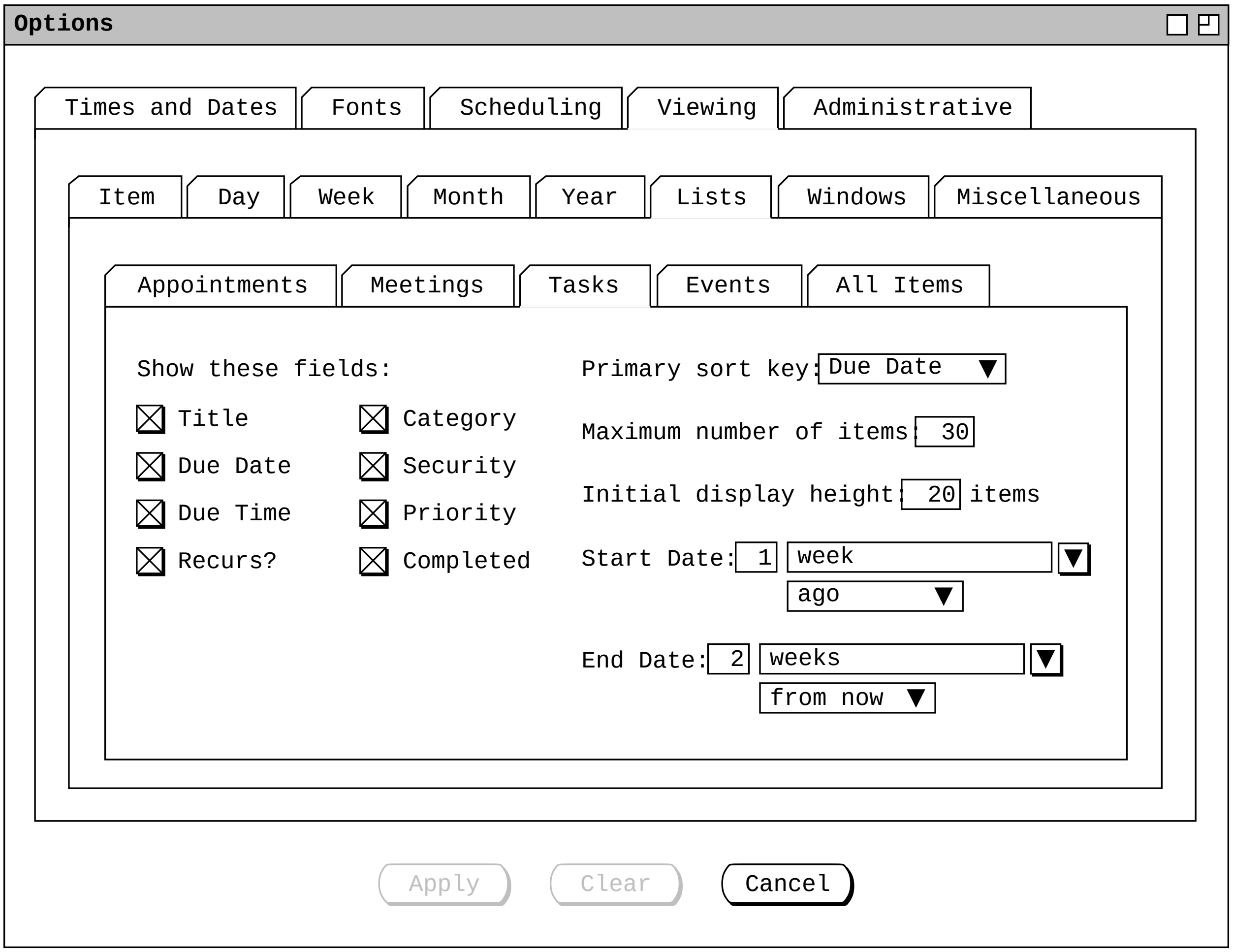
Maximum number of items: 30

Initial display height: 20 items

Start Date: 1 week ▼ ago ▼

End Date: 2 weeks ▼ from now ▼

Figure 276: Viewing meetings lists options tab.



Show these fields:

☒ Title
☒ Due Date
☒ Due Time
☒ Recurs?
☒ Category
☒ Security
☒ Priority
☒ Completed

Primary sort key: Due Date ▼

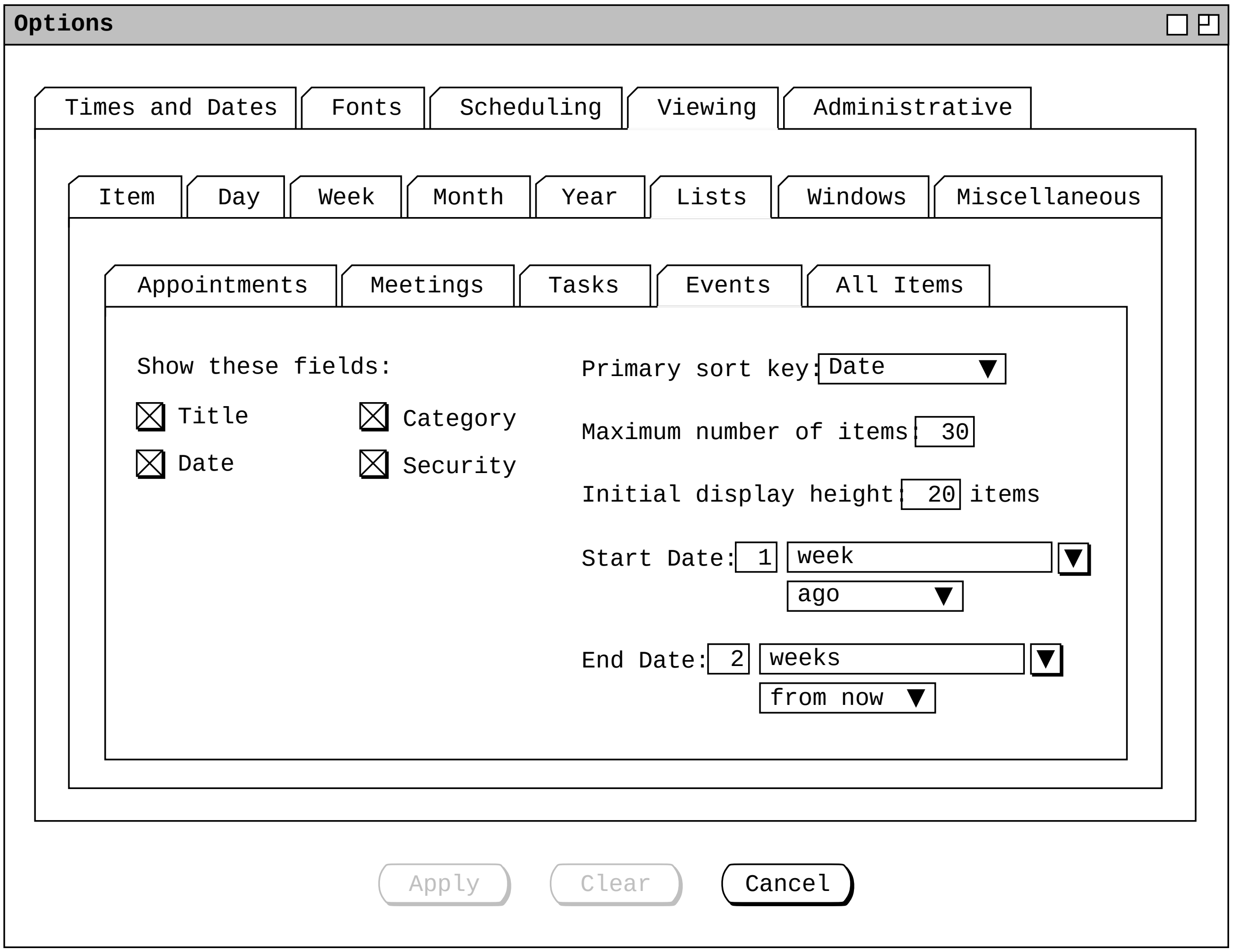
Maximum number of items: 30

Initial display height: 20 items

Start Date: 1 week ▼ ago ▼

End Date: 2 weeks ▼ from now ▼

Figure 277: Viewing tasks lists options tab.



Show these fields:

☒ Title
☒ Category
☒ Date
☒ Security

Primary sort key: Date

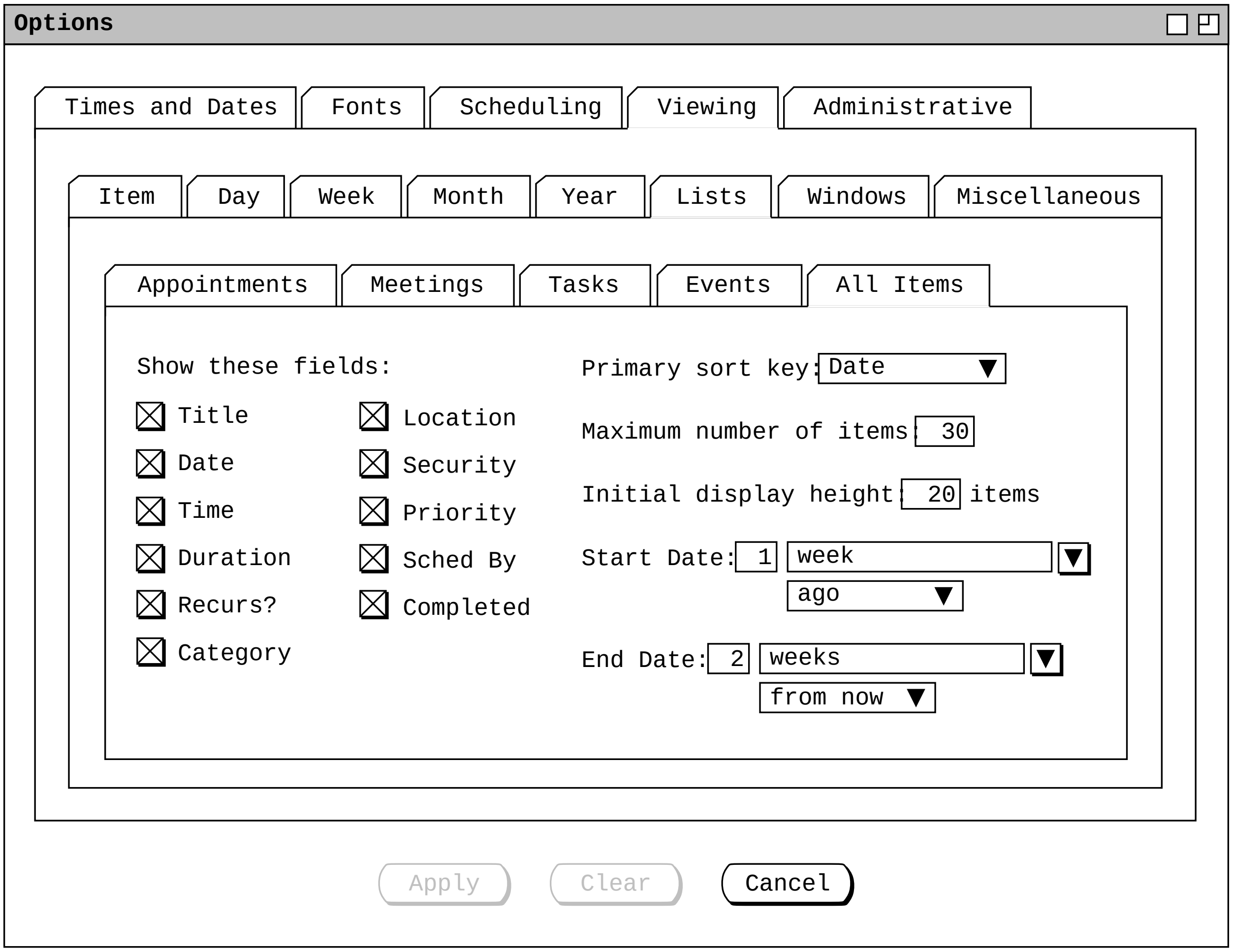
Maximum number of items: 30

Initial display height: 20 items

Start Date: 1 week ago

End Date: 2 weeks from now

Figure 278: Viewing events lists options tab.



Show these fields:

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

Primary sort key: Date ▼

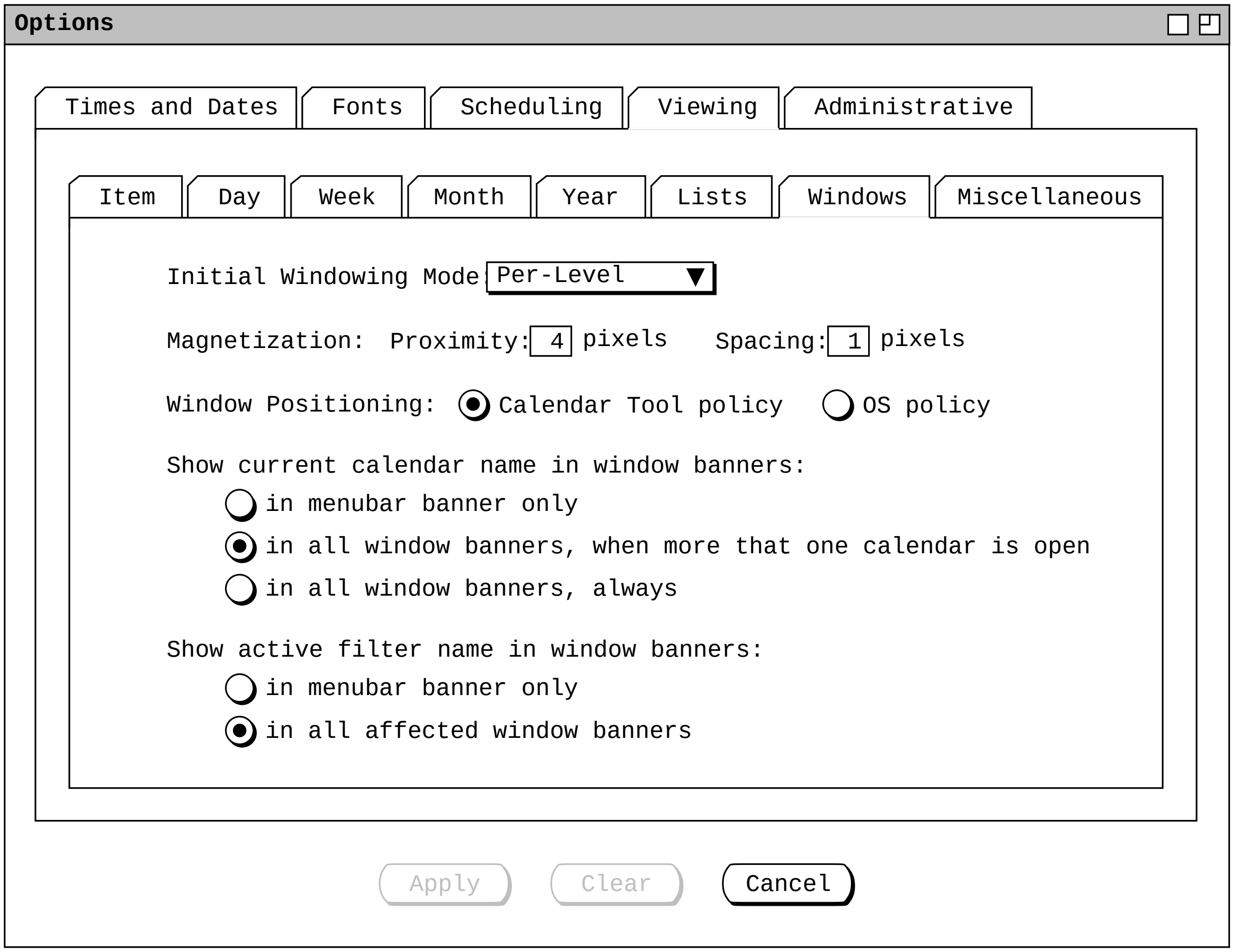
Maximum number of items: 30

Initial display height: 20 items

Start Date: 1 week ▼ ago ▼

End Date: 2 weeks ▼ from now ▼

Figure 279: Viewing all-items lists options tab.



Initial Windowing Mode: Per-Level

Magnetization: Proximity: 4 pixels Spacing: 1 pixels

Window Positioning: ● Calendar Tool policy ○ OS policy

Show current calendar name in window banners:
○ in menubar banner only
● in all window banners, when more that one calendar is open
○ in all window banners, always

Show active filter name in window banners:
○ in menubar banner only
● in all affected window banners

Figure 280: Viewing windows options tab.

The ‘`Magnetization`’ option specifies the default proximity and spacing for the magnetization features described in Section 2.3.6.3.

The ‘`Window Positioning`’ radio buttons specify the way successive display windows appear on the screen. When the ‘`Calendar Tool policy`’ is on, the window placement is as follows:

a. After the menubar, the next display window is placed with its top border ten pixels below the bottom of the menubar, and with its left border aligned with the left border of the menubar.

b. Thereafter, the top left corner of each successive display window appears down and to the right of the most recently displayed previous window. The downward and rightward spacing is two times the height of the display window banner, which value is dependent on the operating environment in which the Calendar Tool is running.

c. Whenever the downward and/or rightward displacement would cause the top left corner of the window to be off the screen, the window is placed so that its top left corner is at the top left corner of the screen. Successive windows thereafter are placed down and to the right.

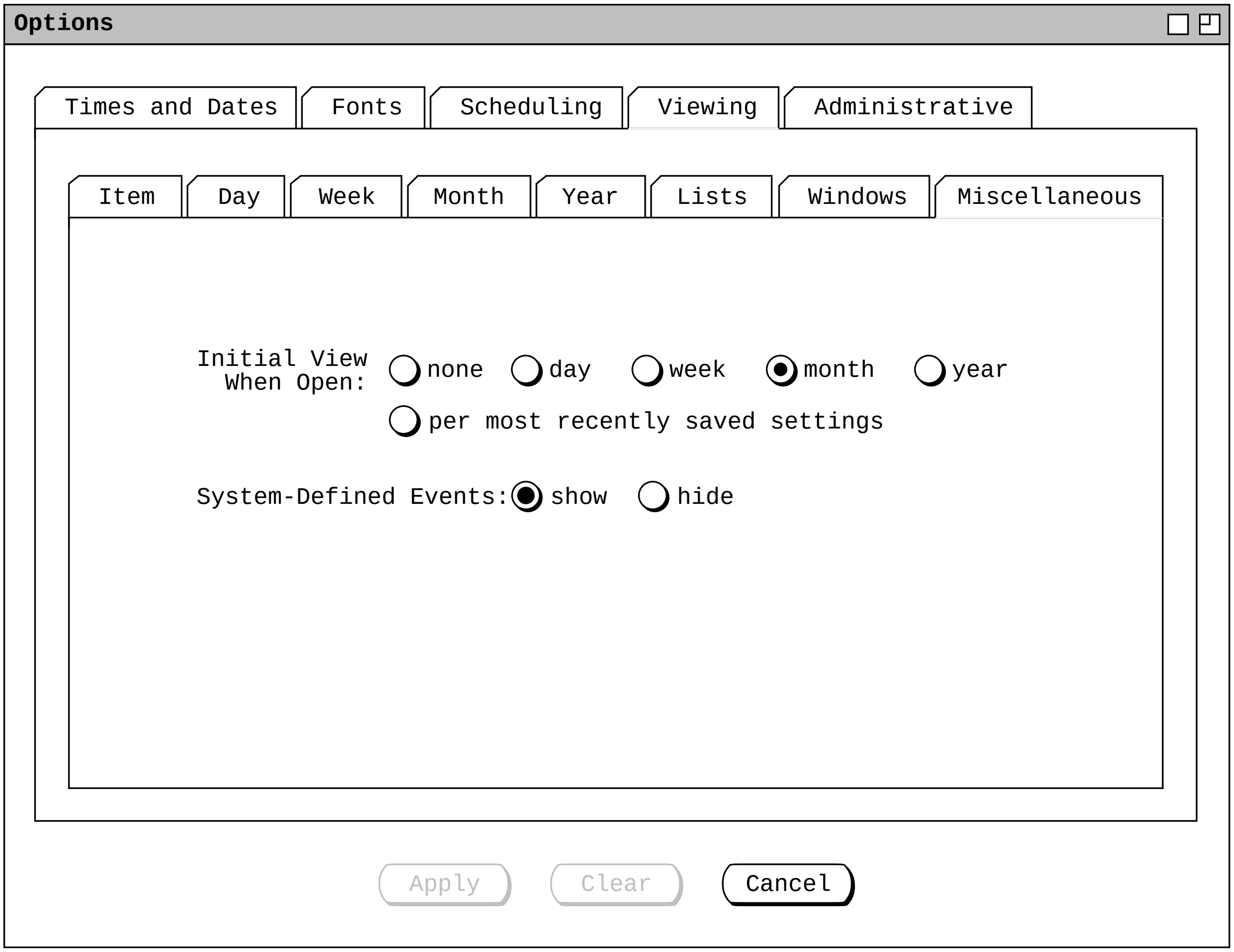
When the ‘`OS Policy`’ radio button is on, successive display windows are placed per the placement policy of the window manager of the underlying operating system. Specific control options for this placement policy must be set at the operating system level, if possible.

The three radio buttons under '`Show current calendar name in window banners`' specify when the name of the current calendar appears in display window banners. Placement of the calendar name within the banner is described in Section 2.3.6.5. 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.

The two radio buttons under '`Show active filter name in window banners`' specify in which display window banners the currently active filter name appears, if a filter is active. Placement of the filter name within the banner is described in Section 2.3.6.5.

2.7.4.3. Viewing Miscellaneous Options

When the user selects the '`Miscellaneous`' tab within the '`Viewing`' tab, the system updates the display as shown in Figure 281. The '`Initial View When Open`' radio buttons allow the user to select which calendar view(s) appear initially when the user opens the calendar with the '`File->Open`' command (Section 2.8.2), or when the calendar name appears in the '`names of open calendars`' list in the `Settings` file that the system loads at start-up (Sections 2.8.1 and 2.8.5). The default for the initial-view option is the month view. If '`none`' is selected, there is no initial view for this calendar, i.e., when the calendar is opened it becomes current, its name appears in the `Calendars` list, but there is no initial display window. If '`Day`' is selected, the calendar is opened with the daily view of today's date. If the '`week`', '`month`', or '`year`' radio button is selected, the calendar is



Initial View When Open: ○ none ○ day ○ week ● month ○ year
○ per most recently saved settings

System-Defined Events: ● show ○ hide

Figure 281: Viewing miscellaneous options tab.

opened at that level, with today's date highlighted. When '`per most recently saved settings`' is selected, the initial view(s) are those saved in the '`state of open windows`' setting for this calendar; this setting is saved whenever the user executes '`File->Save`', as described in Section 2.8.4.

Typically, when the user selects '`per most recently saved settings`', the initial view for the calendar is the same as when the user last closed the calendar or last exited the Calendar Tool. This is the case because it is presumably typical that the user saves the calendar before closing it or exiting.

When the user invokes the Calendar Tool with one or more calendar files as arguments, the effects of the initial view settings at start-up time are as shown in Table 14. Section covers details of invoking the Calendar Tool with calendar file arguments.

The radio buttons for '`System-Defined Events`' specify whether to show or hide such events in all calendar and list views. System-defined events are holidays and other pre-defined calendar dates, as defined in Section .

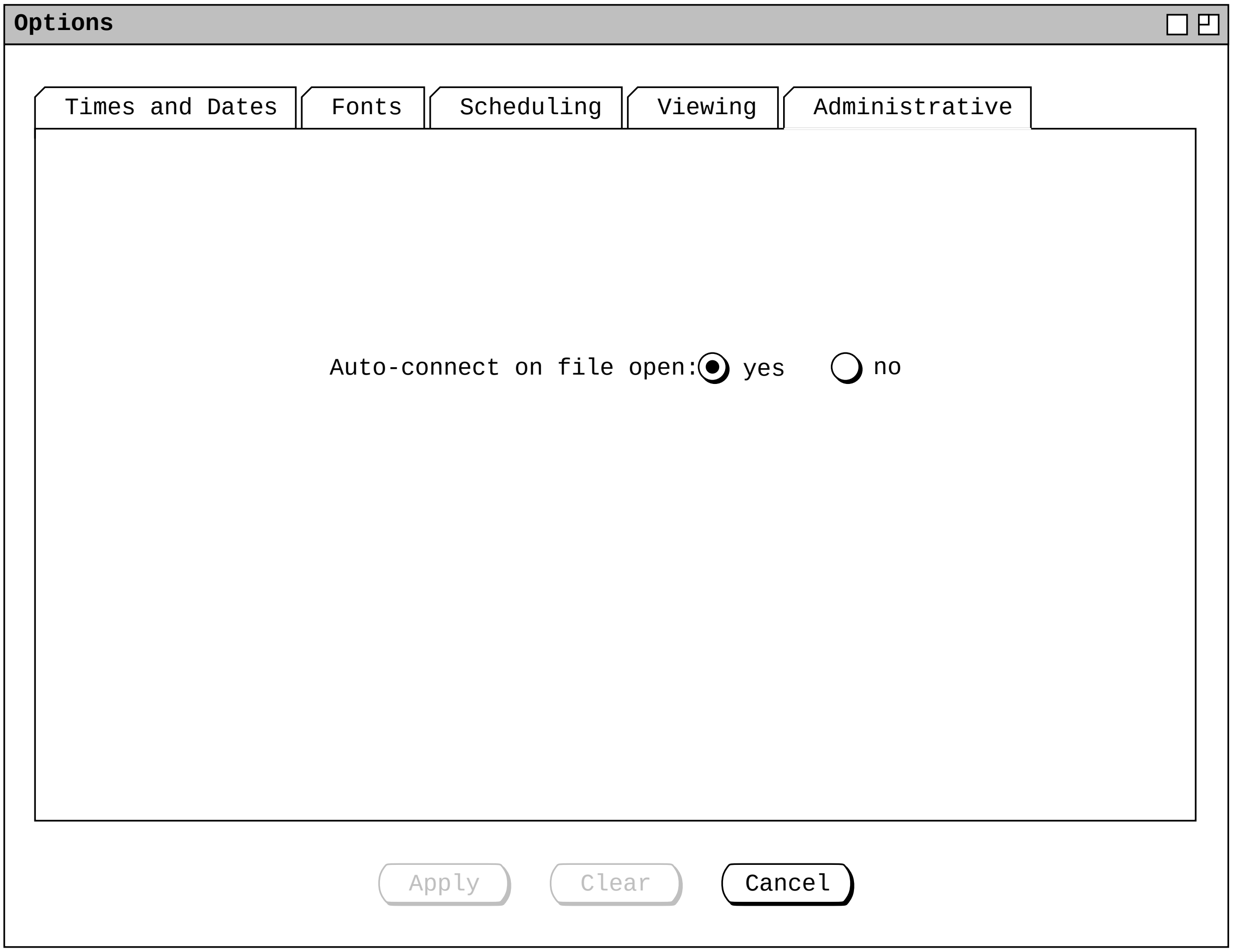
2.7.5. Administrative Options, for the Regular User

When the user selects the '`Administrative`' tab in the options dialog, the system updates the display as shown in Figure 282. The '`Auto-connect on file open`' option controls whether the system attempts to establish a connection with the calendar's central host computer whenever a calendar is opened. When the default setting of '`yes`' is selected, every time the user executes '`File Open`' for a calendar, the system tries to connect to the central host associated with that calendar, if there is an associated host. Details of host association and connection are covered in Section 2.6.6.1. If the user selects '`no`' for the auto-connect option, then the system does not establish a host connection when a calendar is open, requiring that the user establish the connection explicitly with the host '`Connect`' command, as described in Section 2.6.6.1.

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 in the regular-user Calendar Tool. For administrative users running the Calendar Tool Administration program, there are additional administrative options, as described in Section 2.7.7.

Initial View Setting	**Effect with respect to calendar file arguments**
`none`	the presence of arguments overrides the `none` setting; all argument files are opened at the month view level, with today's date highlighted; windows are placed on the screen per the current window placement option setting
`day`, `week`, `month`, or `year`	each argument file is opened at the selected view level
`per most recently saved settings`	all argument files are first opened at the month view level, with today's date highlighted, per current window placement option; then the settings are used to further populate the initial display, except that if the settings include the month view for any already-opened argument calendar, that month-view setting is ignored
`view status as of most recent close`	same effect as '`per most recently saved settings`'

Table 14: Effects of initial view settings when tool is invoked with file arguments.



Auto-connect on file open: ● yes ○ no

Figure 282: Administrative options tab.

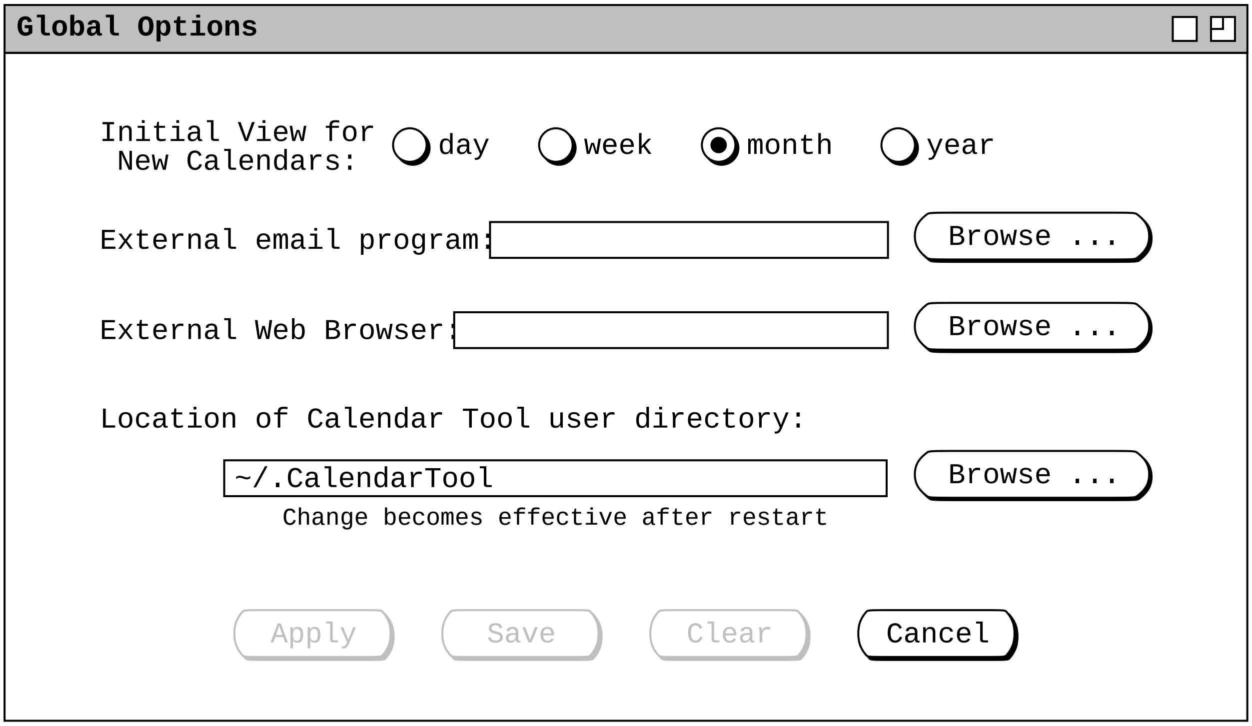
2.7.6. Global Session-Wide Options

When the regular user selects the ‘`Global`’ command in the ‘`Options`’ menu, the system displays the dialog shown in Figure 283.

‘`Initial View for New Calendars`’ specifies the initial view when the user creates a calendar with the ‘`File->New`’ command, or when the system creates a new calendar after having been invoked from the operating environment with no file argument(s). If ‘`Day`’ is selected, a new calendar is opened with the daily view of today’s date. If any of the other three radio buttons is selected, the new calendar is opened at that level, with today’s date highlighted.

The option for ‘`External email program`’ allows the user to select the application program to be used by the Calendar Tool for the purposes of electronic mail communication. The only circumstance in which an external email program is launched from the Calendar Tool is when the user executes the ‘`Contact Admin`’ command, as described in Section 2.6.6.5. When possible, the Calendar Tool queries the underlying operating environment to determine the identity of the user’s preferred email program. When such determination is made, the system inserts the name of the program as the default value in the ‘`External Email Program`’ field. If the user’s preferred email program cannot be determined, the field is blank.

The option for ‘`External Web Browser`’ allows the user to select the application program to be used by the Calendar Tool for the purposes of web browsing. The only circumstance in which an external web browser is



Global Options

Initial View for New Calendars: ○ day ○ week ● month ○ year

External email program:

External Web Browser:

Location of Calendar Tool user directory:

~/.CalendarTool

Change becomes effective after restart

Figure 283: Global options dialog.

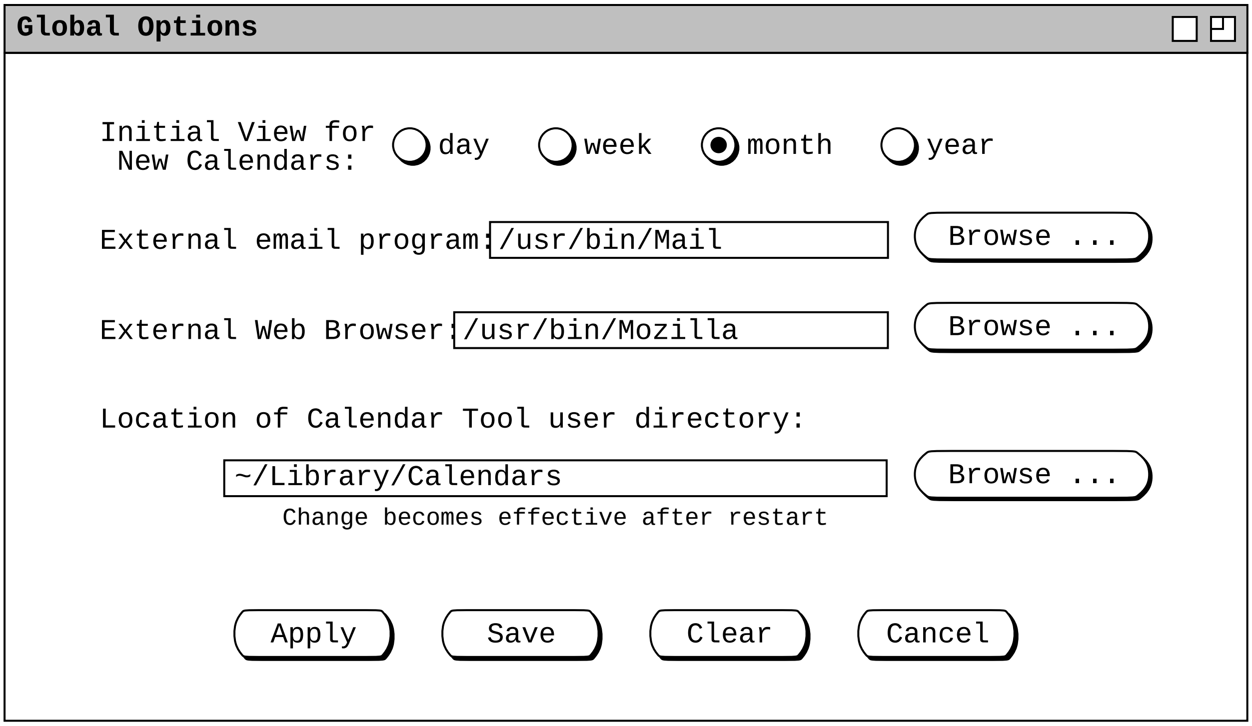
launched from the Calendar Tool is when the user chooses to view the minutes associated with a meeting, as described in Section 2.5.1.2. When possible, the Calendar Tool queries the underlying operating environment to determine the identity of the user's preferred web browser. When such determination is made, the system inserts the name of the web browser as the default value in the '`External Web Browser`' field. If the user's preferred web browser cannot be determined, the field is blank.

The option for '`Location of Calendar Tool user directory`' allows the user to change the location of that directory. As explained in Section 2.8.1, the default location is "`.CalendarTool`" under the user's home directory. This default location appears as the initial setting in the options dialog. Once applied, a change in directory location takes effect after the user exits and restarts the Calendar Tool. This is the only option that requires a restart.

A change of directory location does does *not* change the location of files in the original user directory. The files must be moved or copied as desired, using the normal file management commands of the underlying operating environment. The system does not validate the directory until the restart is performed. If the changed-to directory is invalid or contains invalid files, the system responds as described Section 2.8.12.

The '`Apply`', '`Clear`', and '`Cancel`' buttons in the global options dialog behave the same as in the multi-tab option dialog described in the preceding sections. The '`Save`' button becomes enabled when the user edits one or more dialog fields. For example, Figure 284 shows the user having entered values for the email program, web browser, and location of the user directory.

When the user presses the enabled '`Save`' button, the system saves the contents of the dialog into the Calendar Tool `Settings` file. Only the global settings portion of '`Settings`' is affected by this save, with all other portions of the settings file left unchanged. Details of `Settings` file contents and related commands are covered in Sections 2.8.1 and 2.8.5. Unsaved changes can also be saved using the '`Save Settings`' command in the '`File`' menu, also described in Section 2.8.5. The '`Save`' button is disabled when the user has made no edits, or changed edited fields back to their most recently-saved values.



Global Options

Initial View for New Calendars: ○ day ○ week ● month ○ year

External email program: /usr/bin/Mail

External Web Browser: /usr/bin/Mozilla

Location of Calendar Tool user directory:

~/Library/Calendars

Change becomes effective after restart

Figure 284: Changes to global options.

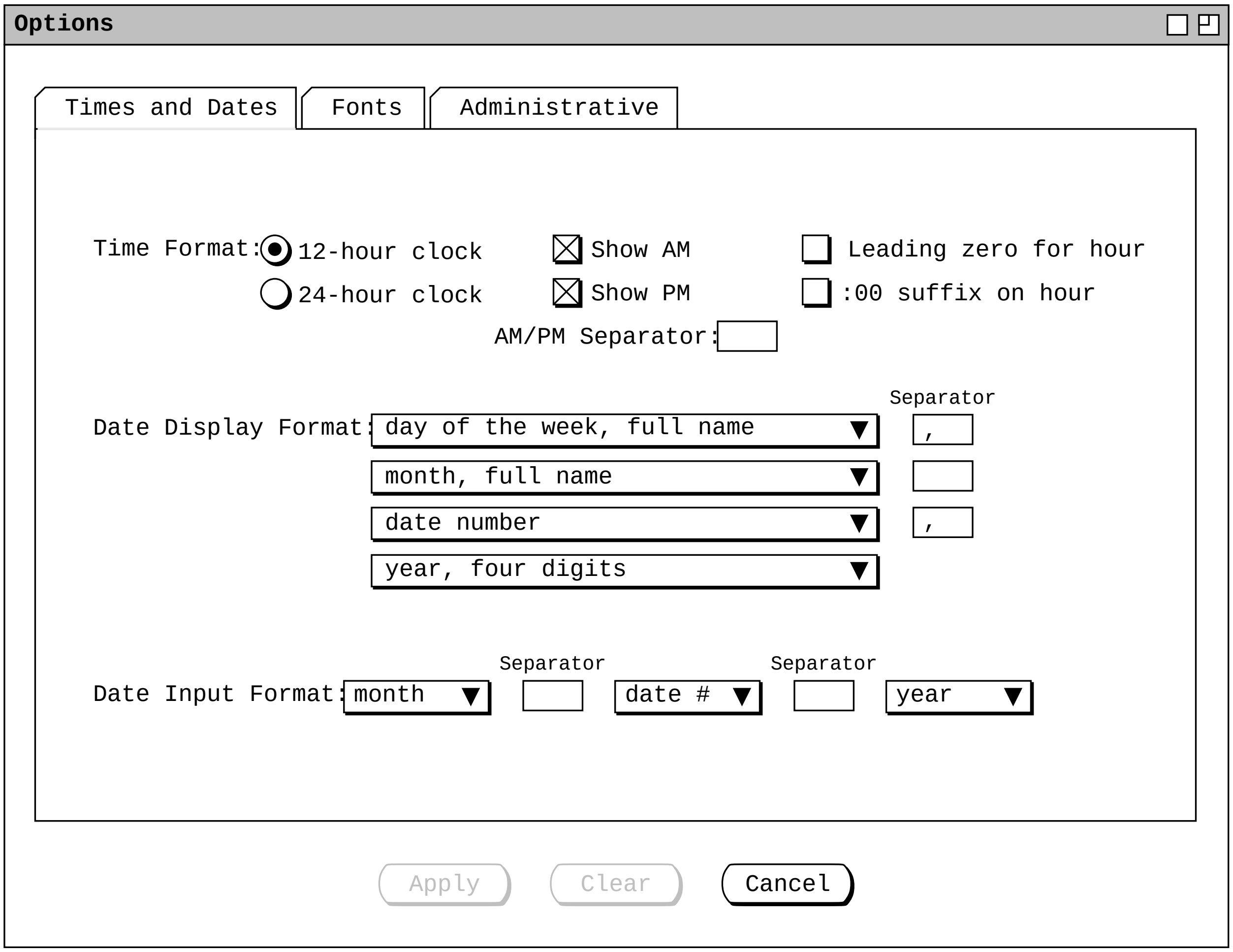
When the user applies or saves the global options, the system validates that the '`Location of Calendar Tool user directory`' is an existing directory that is readable and writable by the user. If this validation fails, the system displays an appropriate error message, as described in Section . The system performs no validation of the email program or external web browser when options are applied or saved. If these are not valid executable programs, the error condition will arise when the user attempts to invoke the programs in the contexts of their use, which are, respectively, contacting a Calendar Tool administrator (Section 2.6.6.5) and viewing the minutes for a meeting (Section 2.4.1.1). If an execution error arises in those contexts, the system displays an appropriate error, as described in Sections

and

2.7.7. Options in the Calendar Tool Administration Program

When the administrative user of Calendar Tool Administration selects the '`Times and Dates`' item in the administrative '`Options`' menu, the system displays the dialog shown in Figure 285. Compared to the options dialog for the regular Calendar Tool, there are two fewer tabs. The three tabs in Figure 285 correspond to the three items in the administrative options menu, shown in Figure 4. The '`Scheduling`' and '`Viewing`' tabs are missing, since the administration program does not perform the calendar scheduling or viewing operations that are available in the Calendar Tool.

The content of the '`Times and Dates`' tab in the administrative options is identical with that tab in the regular-user options. These options also have exactly the same functional effect as in the regular-user interface. The appearance of times and dates in the Calendar Tool Administration program is more limited than in the regular-user Calendar Tool. In particular, the only place that times and dates appear in the administrative interface is in the location bookings, described in Section 2.6.4. This means that the time and date option settings are less significant for the administrative user, though again, their effect where visible is the same as for the regular user.



Time Format: ● 12-hour clock ○ 24-hour clock

☒ Show AM

☒ Show PM

☐ Leading zero for hour

☐ :00 suffix on hour

AM/PM Separator:

Date Display Format:	Separator
day of the week, full name ▼	,
month, full name ▼	
date number ▼	,
year, four digits ▼	

Date Input Format: month ▼ Separator: date # ▼ Separator: year ▼

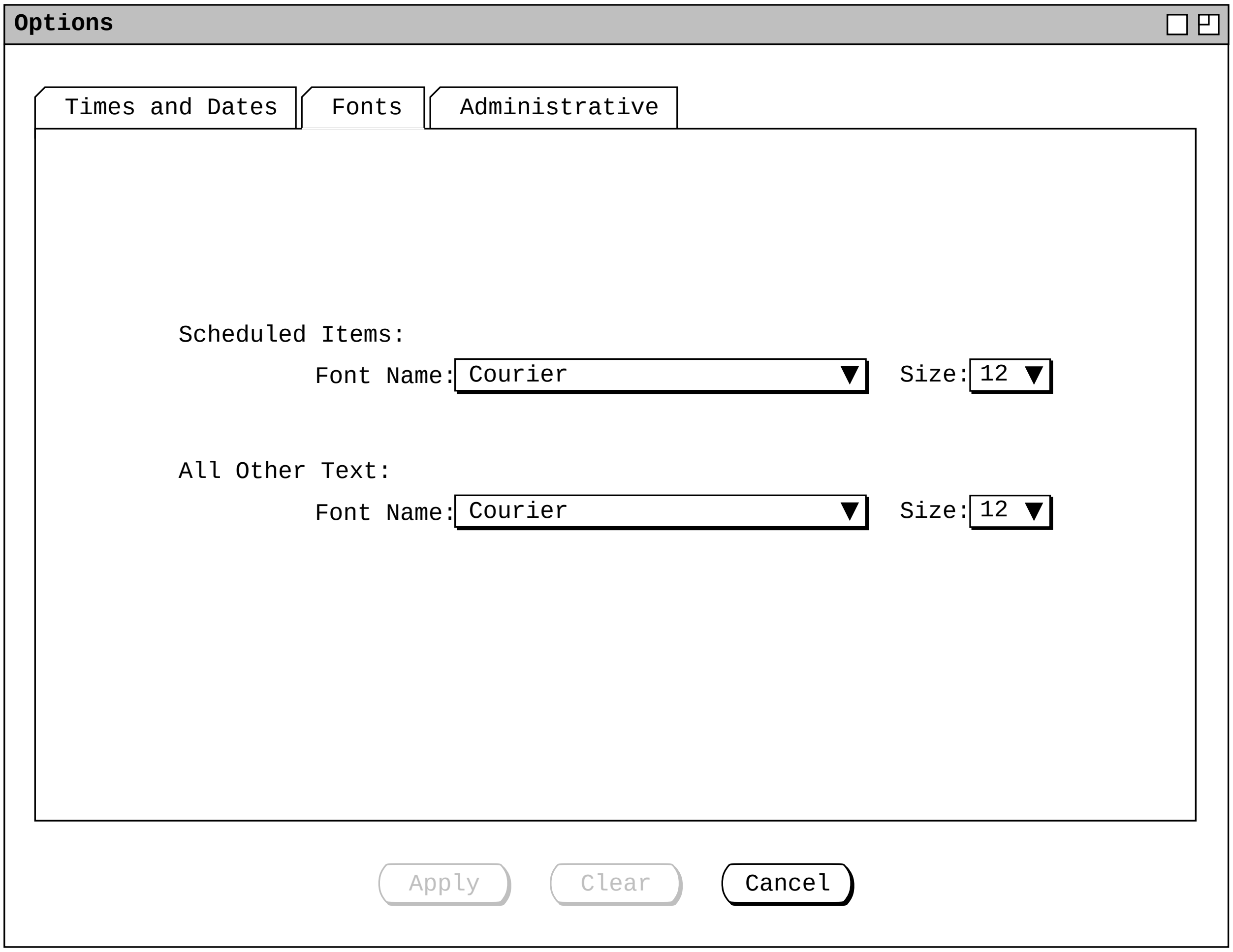
Figure 285: Options dialog for Calendar Tool Administration.

When the administrator selects the '`Fonts`' tab in the administrative options dialog, the system updates the display as shown in Figure 286. This tab is again identical in content and function to the one in the regular-user interface. Since the appearance of scheduled items is very limited in the administrative interface, the effect of the scheduled-item font setting is limited. In particular, the only place where scheduled items are shown in the administrative interface are in the location booking for a meeting (see Figure Figure 229).

When the administrator selects the '`Administrative`' tab in the options dialog, the system updates the display as shown in Figure 287. Here the option settings are fundamentally different than for the regular-user interface.

The '`Default calendar size limit`' specifies the default maximum size of all user and group calendars on the central host. The default can be overridden for individual users or groups by changing the size limit for those individuals, as explained in Section 2.6.5.5.

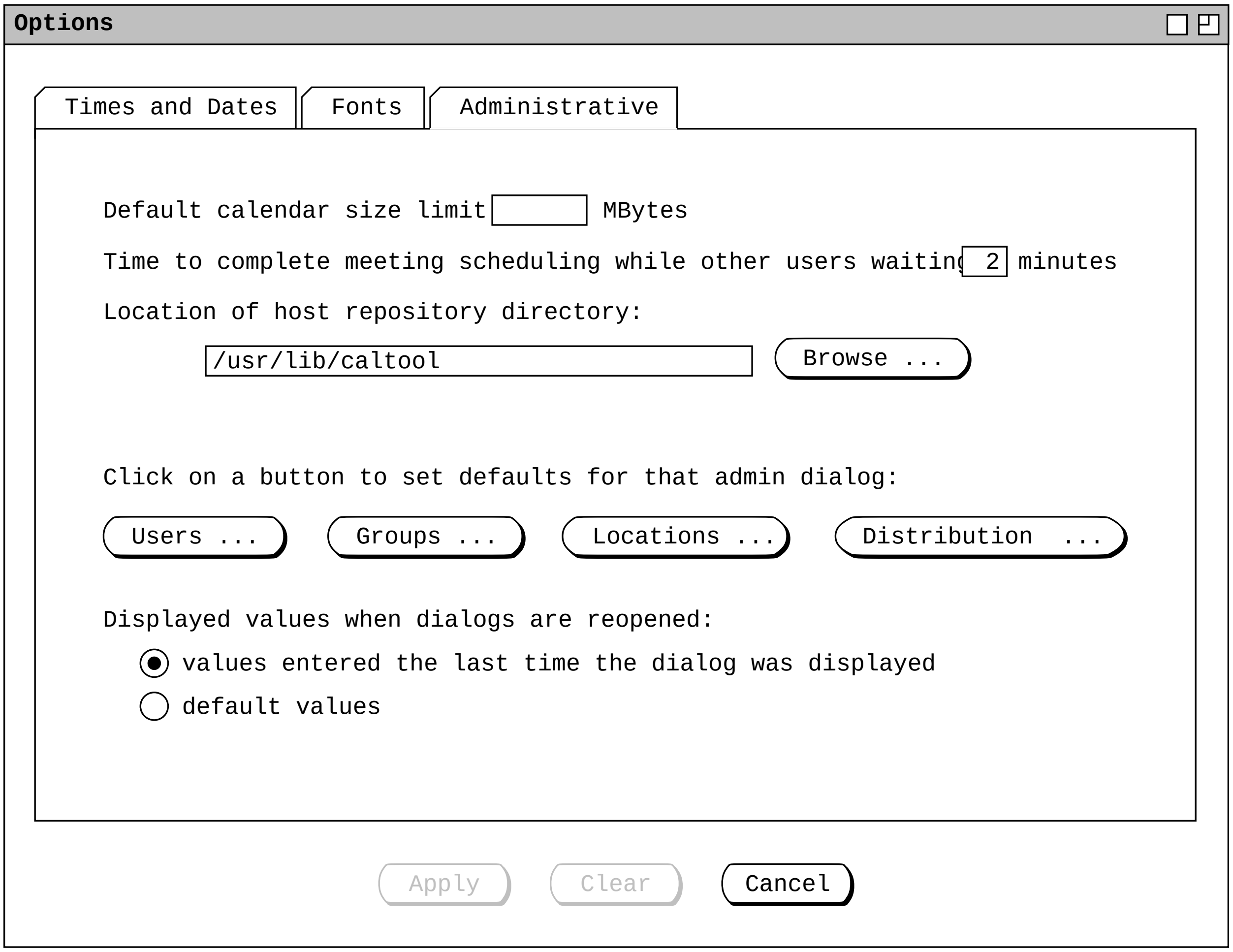
The option for '`Time to complete meeting scheduling while other users waiting`' is explained in Section 2.4.1.6.9. The administrator can change the time from the default of 2 minutes, if desired. The value must be between 1 and 15 minutes, inclusive.



Scheduled Items:
Font Name: Courier
Size: 12

All Other Text:
Font Name: Courier
Size: 12

Figure 286: Fonts options tab in the administrative options.



Default calendar size limit ☐ MBytes

Time to complete meeting scheduling while other users waiting 2 minutes

Location of host repository directory:

/usr/lib/caltool

Click on a button to set defaults for that admin dialog:

Displayed values when dialogs are reopened:

● values entered the last time the dialog was displayed
○ default values

Figure 287: Administrator version of the Administrative options Tab.

The option for '`Location of host repository directory`' allows the administrator to change the repository location. Details of the central host repository and its location are covered in Sections 2.6 and 2.8.8. Figure 287 shows a typical default value for the repository location.

Once applied, a change in directory location takes effect after the administrator exits and restarts the Calendar Tool Administration program. This is the only administrative option that requires a restart.

Changing the location of the repository does *not* change the location of the repository files and subdirectories. The files and subdirectories must be moved or copied as desired, using the normal file−management commands of the underlying operating environment. The system does not validate the directory until the restart is performed. If the changed-to directory is invalid or contains invalid files, the system responds as described Section 2.8.12.

The default setting buttons under '`Click on a button to set defaults for that admin dialog`' operate in the same manner as the default-setting dialogs described in Section 2.7.3.1. That is, when the administrator selects one of the default-dialog buttons, the system displays a normal administrative dialog that is augmented for the purposes of setting default values. The two radio buttons under '`Displayed values when dialogs are reopened`' have the same function as for regular-users, as explained at the end of Section 2.7.3.1.

The built-in default values for all administrative dialogs are defined when the calendar tool is configured for installation (see Section). These values are illustrated in the figures of Sections 2.6.2 through 2.6.5 that show the initial state of each administrative dialog.

2.7.8. Restoring Built-In Default Settings

When the Calendar Tool and Calendar Tool Administration program are initially installed, they have built-in default settings for all options. The values for these built-in defaults are those that appear in each of the preceding figures that show the initial state of the option tabs, and the global options dialog.

The built-in settings are overridden by any applied changes in the options dialogs. The user (regular or administrator) can restore the built-in defaults with the ‘`Restore Defaults`’ command at the bottom of the ‘`Options`’ menu (both regular-user and administrative). When the user executes this command, the system restores all options to their built-in settings and applies the changes. In the case of the global options, the system both applies and saves the changes. The result is that all tabs are restored to their default states.

If the user wants to retain changed option settings before restoring defaults, the user must execute a ‘`Save`’ or ‘`Save Settings`’ command in the ‘`File`’ menu. These commands save applied option settings in the current calendar or a settings file, as described in Sections 2.8.4 , 2.8.5 , and 2.8.9. Restored defaults are not automatically saved. To do so, the user must execute a save command.

2.7.9. Operating Environment Conventions

Some operating environments may have specific conventions for the organization of the functionality described in this 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 Preferences 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

The scenarios in this section cover full details of file-related commands, some of which have been introduced in preceding sections. There are two file command menus to be covered, one in the regular-user Calendar Tool and the other in the Calendar Tool Administration program. Regular-user file commands are covered in Sections 2.8.1 through 2.8.7. Administrator file commands are covered in 2.8.8 through 2.8.11. The concluding Sections 2.8.12 and 2.8.13, cover general topics relevant to both regular–user and administrative files.

2.8.1. Files Used by the Calendar Tool

There are two types of files used by the regular-user Calendar Tool -- calendar files and settings files. Calendar files contain the following information:

- *o* Scheduled items, i.e, appointments, meetings, tasks, and events
- *o* Calendar-specific settings, consisting of:
  - -- categories (Section 2.5.5)
  - -- custom lists (Section 2.3.3.6)
  - -- filter settings and custom filters (Section 2.3.4.3)
  - -- options (Section 2.7)
  - -- page setup and print settings (Section 2.8.6)
  - -- windowing mode (Section 2.3.6.2)
  - -- state of open windows (Section 2.3.6)

All of this information is stored in a single calendar file for each calendar created by the user. In this way, a single file encapsulates scheduled items and all other calendar-specific information that the user has defined for that calendar. "For that calendar" means that each calendar has its own setting information, but no settings for other calendars that may be open at the same time. In particular, the calendar-specific information for the state of open windows means the windows open for a particular calendar, not for any other open calendars. The calendar-specific nature of the different types of information is explained further in the sections cited parenthetically in the immediately preceding list.

The other type of Calendar Tool file contains settings that exist independent from any individual calendar. These settings are used as the defaults when a new calendar is created and to record setting values that apply to a whole Calendar Tool session, not just an individual calendar. The contents of a settings file are the following:

o Calendar-specific settings:

    -- categories
    -- custom lists
    -- filter settings and custom filters
    -- options
    -- page setup and print settings

0. windowing mode

    -- state of open windows

o Session-wide settings:

-- global options, described in Section 2.7.6
-- settings defined in the host-connection table, described in Section 2.6.6.1.
-- the names of all open calendars, i.e., the names in the `View->Calendars` submenu described in Section 2.3.6.4

The calendar-specific settings in a settings file are a separate version of the same information that is contained in each calendar. Their existence in the settings file is for permanently recording settings used in new calendars, and to allow settings to be conveniently copied among different calendars. The session-wide settings define information that applies across all calendars during a Calendar Tool execution session. Section 2.8.5. describes how settings files are saved and loaded.

Calendar and settings files may be stored in any file directory accessible to the user. However, there is a specific directory called the *Calendar Tool user directory* used for the following purposes:

o the location of the `Settings` file opened when the Calendar Tool starts up
o the initial default directory for the '`File->Open`', `File->Save As'`, '`File->Load Settings`', and '`File->Save Settings`' dialogs
o the location of calendars named in the connection table, when the calendars have no leading pathname in the table
o the location where users presumably store their calendar files, though this is not required

Whenever the Calendar Tool is invoked, it opens a single settings file to initialize the session-wide settings for the ensuing session. The name of this file is "`Settings`", and it must be located in the Calendar Tool user directory at the time the '`New`' command is executed. The user directory also has a default location, determined at installation time as described in Section .

The user can save additional copies of setting files, which are useful for copying settings from one calendar to another. However, the user cannot change the name or relative location of the default file named `Settings` that the Calendar Tool opens initially at start up. That is, the file must be named "`Settings`" and it must be located immediately within the Calendar Tool user directory.

The user *can* change the location of her own Calendar Tool user directory, as an option. Doing so is described in Section 2.7.6.

There are no naming requirements for calendar or settings files, other than the requirement for the fixed name and location of the "`Settings`" file. In particular, there are no requirements for filename extensions. If the user chooses to give calendar files an extension, or if a particular operating environment recommends or requires that application files have known extensions, then the recommended extension for calendar and settings files is "`.cal`". If this extension conflicts with that used by some other application, then the installation process can assign a particular extension to be used in a particular operating environment.

The internal data format of calendar and settings files is not specified in these requirements, and therefore left to the implementors. In particular, there are no requirements that these files be editable as plain text or other external data format understandable to application programs other than the Calendar Tool. It is required that the Calendar Tool be able to identify calendar and settings file as such, so they can be distinguished from each other and from other types of files. The specific means to accomplish this identification is left to the implementors.

2.8.2. New and Open

When the user executes the ‘`New`’ command in the ‘`File`’ menu, the system responds by creating a new calendar and displaying its initial view. The default initial view is at the month level, with today’s date highlighted. The user can change the default initial view for new calendars with the view option described in Section 2.7.6.

The name of a new calendar is "`unnamed`", until the user saves it with a different name. Each successive application of `New`, with no intervening save, creates a calendar named "unnamed[*N*]", for $N = 2$ to the number of unnamed calendars.

A new calendar has no scheduled items except for system-defined events. The user can change the visibility of system-defined events with the option setting described in Section 2.7.4.3. The setting effects only whether system-defined events are visible in calendar displays, not their physical presence in the calendar. Section describes exactly what the system-defined events are.

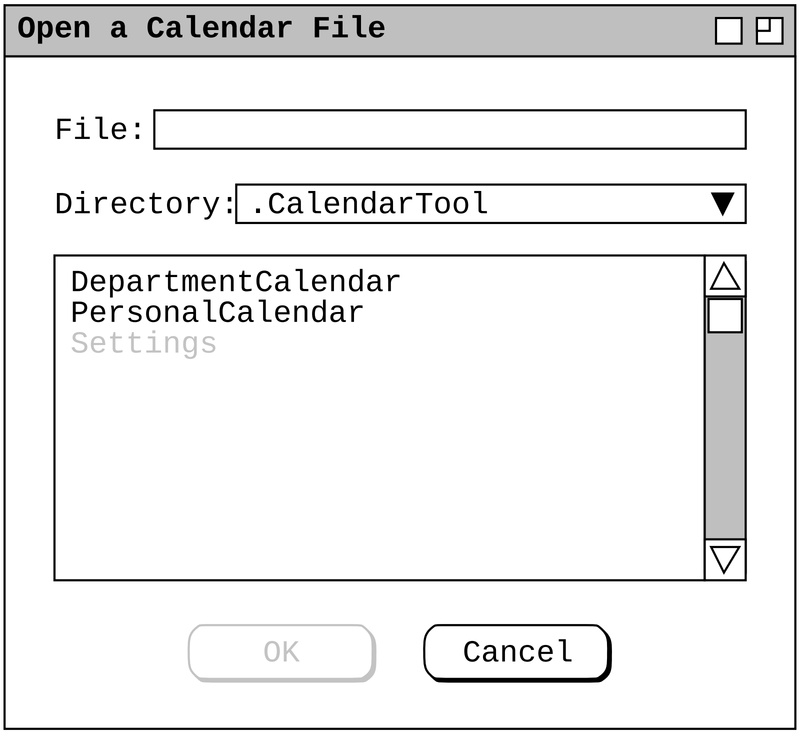
The calendar-specific settings for each new calendar are those defined in the `Settings` file in the Calendar Tool user directory. If there is no `Settings` file, settings have the built-in default values defined when the Calendar Tool is installed. Table 15 shows the built-in default settings for all new calendars, in the absence of a `Settings` file. A new calendar becomes the current calendar, and the system performs all of the actions associated with making a calendar current. These actions are described in Section 2.3.6.4. Before the user saves a new calendar, it has no associated file. This affects the behavior of file-save operations, as discussed in Section 2.8.4.

When the user selects the ‘`Open`’ command in the ‘`File`’ menu, the system displays a dialog of the form shown in Figure 288. The dialog reflects the state of the `.CalendarTool` directory after the user has saved two calendar files -- `DepartmentCalendar` and `PersonalCalendar`. The ‘`Directory`’ pulldown lists the components in the file path to the current working directory. As explained in Section 2.8.1, the default directory path for the file-open dialog is the Calendar Tool user directory for the current user.

The user may type the name of the file to open or select it from the scrollable list of file names. When the user selects the pulldown arrow in the ‘`Directory`’ field, the system displays the directory path, as shown in Figure 289. The user may select from the directory list to navigate to another directory, whereupon the system displays its

Type of setting	**Initial value**	**Where illustrated**
categories	only holidays defined	Figure 7
custom lists	none defined	Figures 37 and 38
filter settings and custom filters	all items shown, no custom filters defined	Figures 51 and 52
options	various	figures in Section 2.7 showing initial values in options-setting tabs
page setup and print options	platform-dependent defaults	Figures 304 and 305
windowing mode per-level	Figure 71	
state of open windows	monthly view shown,	Figure 1

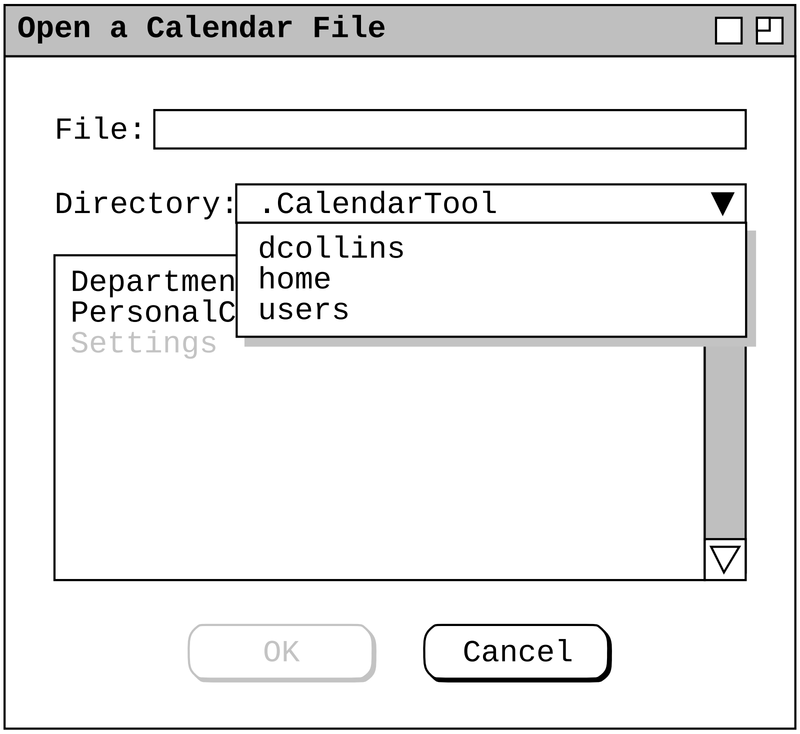
Table 15: Built-in default settings for new calendars, absent a Settings file.



Directory: .CalendarTool

DepartmentCalendar
PersonalCalendar
Settings

Figure 288: File open dialog.



Directory: .CalendarTool

Departmen
PersonalC
Settings

Figure 289: File open dialog with directory path displayed.

files and subdirectories in the scrollable list. The names of readable calendar files and readable subdirectories are shown in regular type in the file list. The names of all other files are shown in grey type, indicating they are disabled for selection. This means the user cannot open an unreadable calendar file, an unreadable directory, or any type of non-calendar file. The user can load a settings file, but this is done with the '`Load Settings`' command, described in Section 2.8.5

The user may navigate into a subdirectory by selecting its name in the list and pressing '`OK`'. The '`OK`' button is enabled when there are one or more subdirectories in the scrollable list. When the user presses '`OK`' for a selected directory, the system displays its contents in the scrollable list.

When the user selects a specific file in the list, the system enters its name in the '`File`' field and enables the '`OK`' button, if it was not already. When the user presses '`OK`', the system opens the selected file if it is not already open. Once opened, the system displays the file's initial view, per its most recently saved settings, including in particular the option setting for its initial view window(s) (see Section 2.7.4.3). The opened calendar becomes the current calendar, and the system performs all of the actions associated with making a calendar current, as described in Section 2.3.6.4. If the selected file is already open, pressing '`OK`' in the open dialog has the same effect as selecting the calendar's name in the '`Calendars`' list, described in Section 2.3.6.4. Double clicking on a calendar file name or directory name is a shortcut for selecting that name and pressing '`OK`'.

The user may type the name of a calendar file or directory in the '`File`' field. As the user types, the scrollable list is scrolled to the alphabetic position in the list nearest but not after the place where the thus-far typed name appears or would appear in the list. If the user types the complete name of an existing calendar file or readable directory, the effect is the same as when the user selects the name by clicking in the list. If the user types the name of a non-existent file, the user may press '`OK`', whereupon the system creates a new file of the entered name. This is effectively a short cut for executing '`File->New`' followed immediately by '`File-Save`'.

Since unreadable files, unreadable directories, and non-calendar files are disabled in the file-open dialog list, the user cannot select one of these. However, the user may type the name of an unreadable file, unreadable directory, or non-calendar file in the '`File`' field. If the user does so, the system disables the '`OK`' button, since these files or directories cannot be opened. If the user continues typing so that the name is not an unreadable calendar file, unreadable directory, or an existing non-calendar file, the system re-enables the '`OK`' button.

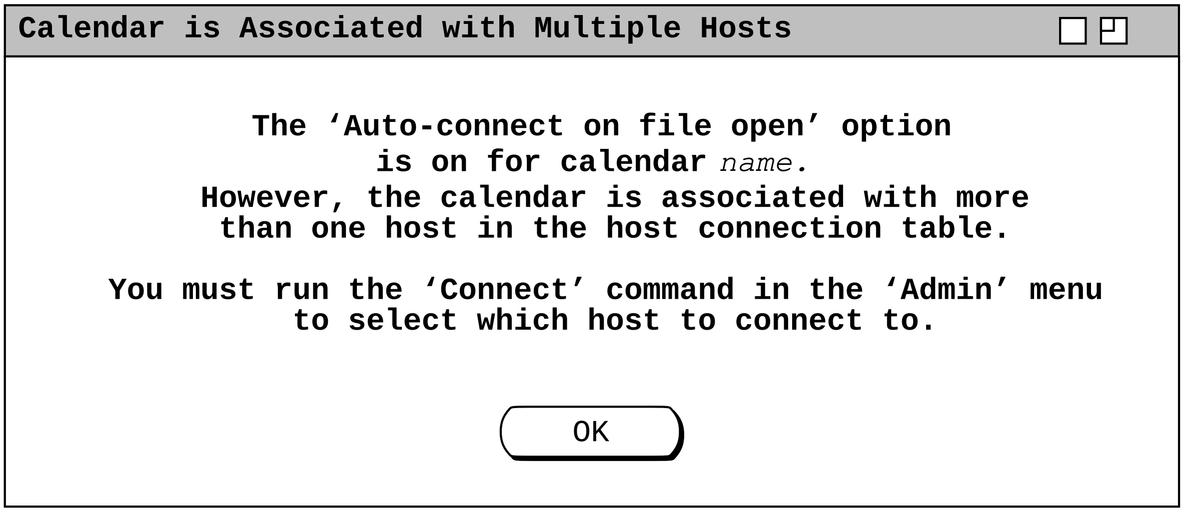
When a calendar is opened from a saved file, the name of the calendar is synonymous with its root filename. The root filename is the name of the file, less any leading file path and less any filename extension. For example, the name of the calendar stored in the file "`/users/home/dcollins/.CalendarTool/DepartmentCalendar.cal`" is "`DepartmentCalendar`". The Calendar Tool provides no command to change the name of a calendar independent from its filename. In particular, the only way the user can name an unnamed calendar is to save it to a file.

A new or just-opened calendar has today's date highlighted with a darkened border, as illustrated in Section 2.3.1. No date is selected as a viewing target, in the sense of selection defined in Section .

If an opened calendar is associated with one host in the host connection table and the '`Auto-connect on file open`' option is on, then the system initiates the host connection. The connect operation proceeds as described in Section 2.6.6.1, as if the user had pressed the '`Connect`' button in the host connection table. The auto-connect option is described in Section 2.7.5. If an opened auto-connect calendar is associated with more than one host, the system displays the dialog shown in Figure 290. The *`name`* string is the name of calendar just opened. The dialog informs the user that since the calendar is associated with more than one host, the user must run the '`Admin->Connect`' command in order to select the host to connect to.

The '`New`' and '`Open`' commands are always enabled in the file menu. The following is a summary of the when the '`OK`' button is enabled in the file-open dialog:

a. when the user has selected the name of a readable calendar file in the scrollable list
b. when the user has selected the name of a readable directory in the scrollable list
c. when the user has typed anything in the '`File`' field, except for the complete name of an unreadable calendar file or existing non-calendar file



The 'Auto-connect on file open' option
is on for calendar *name*.
However, the calendar is associated with more
than one host in the host connection table.

You must run the 'Connect' command in the 'Admin' menu
to select which host to connect to.

Figure 290: Select from multiple associated hosts.

As explained in Section 2.3.6 zero or more new or existing calendars can be open at any time the Calendar Tool is running. That section explains the way that the windows of multiply open calendars are managed and displayed.

Since '`New`' and '`Open`' are always enabled, the user may execute these commands at any time while the Calendar Tool is running. This means that the user can have an unbounded number of calendars open. If taken to an extreme, this could conceivably cause resource problems in some operating environments, as discussed in Section

2.8.3. Close and Close All

The '`Close`' and '`Close All`' commands are enabled in the '`File Menu`' if there is at least one open calendar, disabled otherwise. '`Close`' applies to the current calendar; '`Close All`' applies to all open calendars.

When the user selects the '`Close`' command, the system performs the following pre-close steps for the calendar to be closed, i.e., the current calendar:

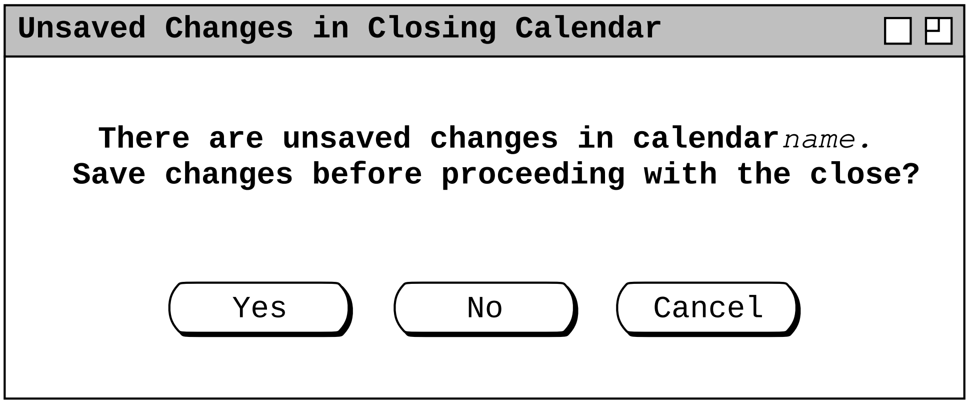
a. checks if the calendar has unsaved changes
b. checks if there are any pending edits in dialogs associated with the calendar
c. checks if the calendar is connected to a central host computer

Once these steps are completed satisfactorily, the system proceeds to perform the close operation. Details follow.

For the first pre-close step, the system checks if the current calendar has any unsaved changes. The precise definition of an unsaved change is given in Section 2.8.4. If there are one or more unsaved changes, the system displays the dialog shown in Figure 291. The *`name`* string is the name of the current calendar to be closed. If the user presses '`Yes`', the system proceeds as follows:

a. If the calendar to be closed has not yet been saved to a file, the system displays the file save-as dialog shown in Figure 294. If the user confirms a legal save-as operation in that dialog, the system proceeds to the next pre-close step. If the user presses '`Cancel`' in the save-as dialog, the system returns to the offer-to save dialog, where the user may press '`No`' or '`Cancel`' (or '`Yes`' if she wants to return again to the save-as dialog).
b. If the calendar to be closed has previously been saved to a file, the system performs the file save operation described in Section 2.8.4, and proceeds to the next pre-close step.

If the user presses '`No`' in the offer-to-save dialog, the system proceeds to the next pre-close step without saving the calendar, thereby losing any unsaved changes. If the user presses '`Cancel`' in the offer-to-save dialog, the system cancels the close operation entirely, without saving, without performing further pre-close steps, and leaving the calendar open and unchanged. The system also removes both the offer-to-save and file-close dialogs from the screen.



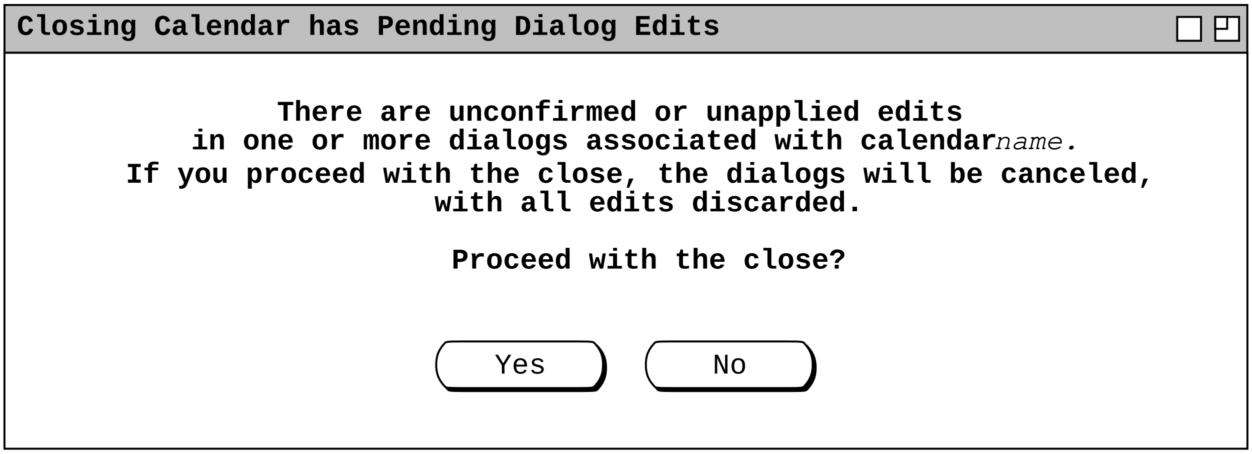
There are unsaved changes in calendar*name.*
Save changes before proceeding with the close?

Figure 291: Offer to save calendar before closing.

If the calendar to be closed has no unsaved changes, the system proceeds to the next pre-close step, without displaying the offer-to-save dialog. Calendars with no unsaved changes are those that have been saved prior to close, as well as other-user and group calendars, since the latter two types of calendar cannot be changed.

For the second pre-close step, the system checks if there are any pending edits in calendar-specific, non-modal editing dialogs for the calendar to be closed. Section defines which Calendar Tool dialogs are of this type. A pending edit is any unconfirmed or unapplied change the user has made in one or more of these dialogs. if there any such pending edits, the system displays the dialog shown in Figure 292. If the user answers '`Yes`' in the pending-edits dialog, the system discards any pending edits, cancels all pending dialogs, and removes the dialogs from the screen. When the system responds to the '`Yes`' response or if the pending-edits dialog was not displayed, the system proceeds to the last pre-close step.

If the user answers '`No`' in the pending-edits dialog, the system cancels the close operation without performing the last pre-close step, leaves the calendar open, and leaves all pending edit dialogs open and unchanged. If the calendar had been saved as a result of the first pre-close step, it remains saved, i.e., the save operation is not undone.



There are unconfirmed or unapplied edits
in one or more dialogs associated with calendar*name.*
If you proceed with the close, the dialogs will be canceled,
with all edits discarded.

Proceed with the close?

Figure 292: Warning when there are unconfirmed or unapplied dialog edits.

For the third and final pre-close step, the system checks if the calendar being closed is currently connected to a central host computer (see Section 2.6.6.1). If so, the system displays the dialog shown in Figure 293. If the user answers ‘`Yes`’ in the connected-to-host dialog, the system disconnects the host and updates the host connection table accordingly, as described in Section 2.6.6.1. When the system responds to the ‘`Yes`’ response, or if the connected-to-host dialog was not displayed, the system proceeds to complete the close.

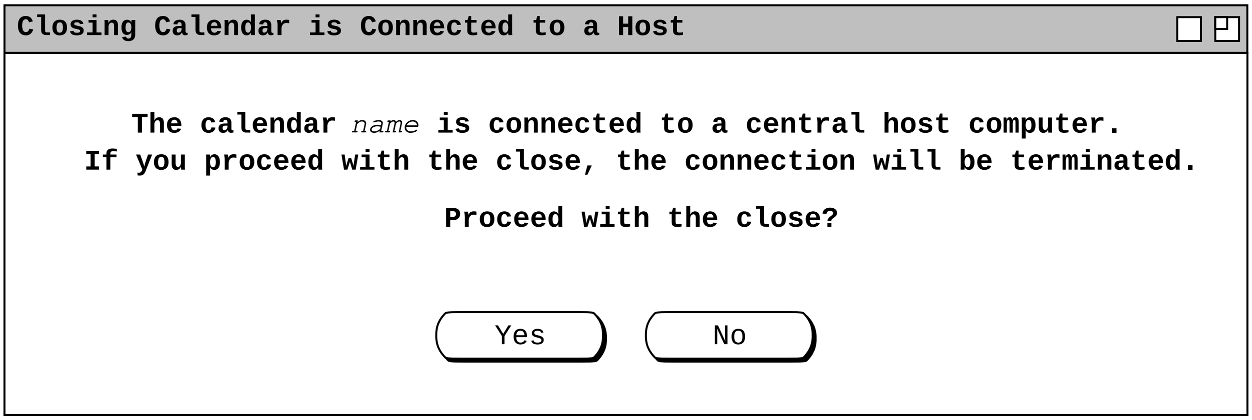
If the user answers ‘`No`’ in the connected-to-host dialog, the system cancels the close operation, leaving the calendar open and connected. If the calendar had been saved as a result of the first pre-close step, it remains saved, i.e., the save operation is not undone. If any pending-edit dialogs where canceled and closed as a result of the second pre-close step, the dialogs remain so.

When the pre-close steps have been completed without a cancellation, the system completes the close operation by closing the current calendar, removing its name from the `Calendars` list, and removing all windows associated with the calendar from the screen, including any remaining dialogs (those without pending edits). Section 2.3.6.4 describes details of the `Calendars` list, calendar-to-window association, and changing the current calendar. When the close operation has been completed or canceled, the system removes all close-related dialogs from the screen.

When the user selects ‘`Close All`’ in the ‘`File`’ menu, the system executes a close operation for all open calendars, in the same manner as described above for ‘`Close`’. The offer-to-save dialog is displayed in turn, as necessary, prior to closing each calendar. If the user presses ‘`Cancel`’ in the offer-to-save dialog for any calendar, the system cancels the close operation for that calendar and all remaining open calendars. ‘`Cancel`’ does not affect any already-closed calendars for which the user previously answered ‘`Yes`’ or ‘`No`’ in the offer-to-save dialog; these calendars remain closed. The pending-edits and connected-to-host dialogs are also displayed in turn, as necessary. Since there is no ‘`Cancel`’ button in the pending-edits or connected-to-host dialogs, the ’`Close All`’ operation proceeds with the next calendar to be closed, whatever the user’s decision is in either of these warning dialogs. A calendar remains open if the user answers‘`No`’ in either the pending-edits or connected-to-host dialog.

When the system completes a ‘`Close`’ or ‘`Close All`’ for the last open calendar, it closes the ‘`Find`’ dialog and ‘`Goto Date`’ dialog, if either or both are open. As described in Section , these two dialogs can be open only when at least one calendar is open.

Closing the last open calendar does not automatically exit the Calendar Tool. As explained in Section 2.3.6.4, the Calendar Tool runs with zero or more calendar files open.



Closing Calendar is Connected to a Host

The calendar *name* is connected to a central host computer.
If you proceed with the close, the connection will be terminated.

Proceed with the close?

Figure 293: Warning when closing calendar is connected to a central host.

2.8.4. Save, Save As, and Save All

A new calendar is not associated with a file until it is saved for the first time. For new not-yet-saved calendars, the '`Save`' and '`Save As`' commands have exactly the same effect. When the user selects either of these commands for a not-yet-saved calendar, the system displays the dialog shown in Figure 294. This dialog shows the initial state of the `.CalendarTool` directory before the user has saved any files. The `Settings` file is shown in the scrollable list, since it is installed by default during the normal Calendar Tool installation process, described in Section . The system does not pre-enter "`unnamed`" in the '`File`' field, to discourage the user from naming a calendar "`unnamed`". The system does not however prevent a recalcitrant user from doing so.

As in the file-open dialog, the user may navigate to another directory by selecting from the '`Directory`' pulldown list and by selecting directory names in the scrollable list. To save to a file, the user types its name in the '`File`' field. The system enables the '`OK`' button when the user begins typing, and it remains enabled unless the user does one of the following:

- *o* erases all typing
- *o* enters the name of a currently open calendar file
- *o* enters the name of an unwritable calendar file
- *o* enters the name of an unreadable directory
- *o* enters the name of an existing non-calendar file

The system disallows saving in any of these circumstances (only the first of which happens to be possible in Figure 294). If the user continues typing to change these circumstances, the system re–enables the '`OK`' button. The system does allow saving a calendar into an existing calendar file, which case is described shortly.

Figure 295 shows the user having entered the name "`DepartmentCalendar`" for saving. When the user presses '`OK`', the system saves the entire contents of the calendar onto the named file. All scheduled items and all calendar-

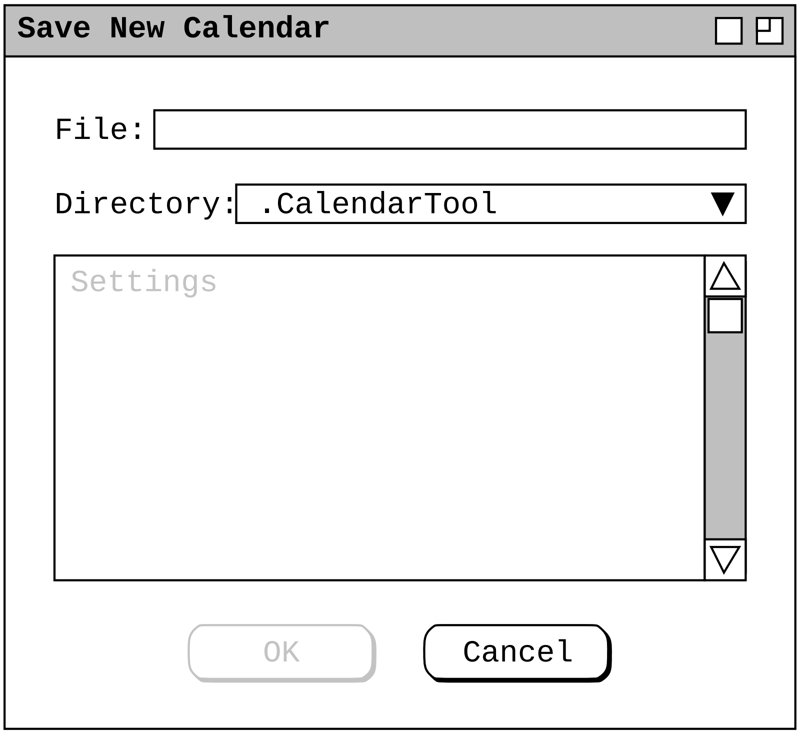
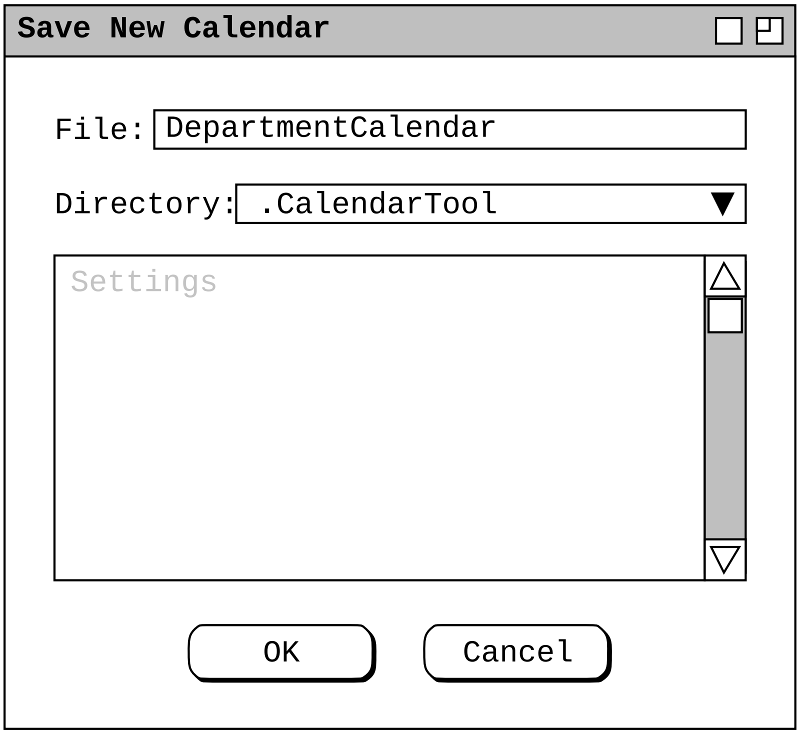


Figure 294: Initial save dialog for a not-yet-saved calendar.



File: DepartmentCalendar

Directory: .CalendarTool

Figure 295: Initial save dialog filled in.

specific settings are saved. After completing the save, the system removes the save dialog from the screen. If the user enters the name of an existing calendar file and presses ‘`OK`’, the system asks for overwrite confirmation, as described below.

When saving a new calendar onto a new file, it is not possible for the saved-to calendar to yet be connected to a central host computer, since to make such a connection the calendar must already have been open. Hence in this case of saving, no copying to a central host is ever performed. Such copying is performed when saving an existing connected calendar, as described shortly.

When the current calendar is already associated with a file, the ‘`Save`’ and ‘`Save As`’ commands differ. The ‘`Save`’ command is enabled whenever the current calendar is associated with a file, and the calendar has unsaved changes. A calendar has unsaved changes when the user has performed one or more of the following actions since the file was opened or the last confirmed save operation, whichever has occurred most recently:

a. confirmed any scheduling or category-changing command in the ‘`Schedule`’ menu
b. confirmed any change to custom lists
c. confirmed any change to filter settings or custom filters
d. executed any command in the view menu that changes the state of the display screen
e. changed the windowing mode
f. moved, iconified, or closed any window associated with the calendar
g. applied any option changes
h. confirmed any change to page setup or print settings
i. loaded settings

If the user undoes any of these actions such that the state of the calendar returns to the same state as most recently saved, then the calendar no longer has unsaved changes. The "state" of the calendar is defined as the values of all scheduled items and calendar–specific settings. It is noteworthy that the definition of unsaved change includes any and all actions that change scheduled items *or* calendar-specific settings.

When the user selects the enabled ‘`Save`’ command in the ‘`File`’ menu, the system saves the scheduled items and settings for the current calendar on the file associated with the calendar. If the file is connected to a central host computer, the system also copies the calendar file to the central host, as explained in Section 2.6.6.1. All scheduled items and calendar-specific settings are copied to the host. If a calendar is associated with a central host but not connected to it, the save operation does perform the copy.

The ‘`Save As`’ command is enabled in the ‘`File`’ menu whenever the current calendar is not that of another user or group. ‘`Save As`’ is enabled whether or not the current calendar has unsaved changes. When the user selects ‘`Save As`’ for a new calendar, the system displays the dialog shown in Figure 294, and the system actions are those described for that figure.

When the user executes ‘`Save As`’ for a calendar that is associated with a file, the system displays the dialog shown in Figure 296. This is the same dialog as shown in Figure 294, except the banner is "`Save Calendar As`" instead of "`Save New Calendar`". Figure 296 shows the state of the `.CalendarTool` directory after the user has saved two calendars. The names of writable unopened calendar files and readable subdirectories are shown in regular type in the list. The names of all other files are shown in grey type, indicating they are disabled for selection. This means that the user cannot save to an already open calendar file, an unwritable calendar file, or to any type of existing non-calendar file. The user can navigate through directories as explained above. The user can navigate to a readable but unwritable directory to see the files there. The existence of a writable calendar file in an unreadable or unwritable directory renders the file itself unwritable.

For a calendar that has an associated file, the purpose of ‘`Save As`’ is to save a copy of the calendar on a file other than the one with which the calendar is already associated. To perform the save, the user types the name of the desired file in the ‘`File`’ field, or selects the name of an existing calendar file in the scrollable list. The system enables the ‘`OK`’ button when the user begins typing, and it remains enabled except under the circumstances

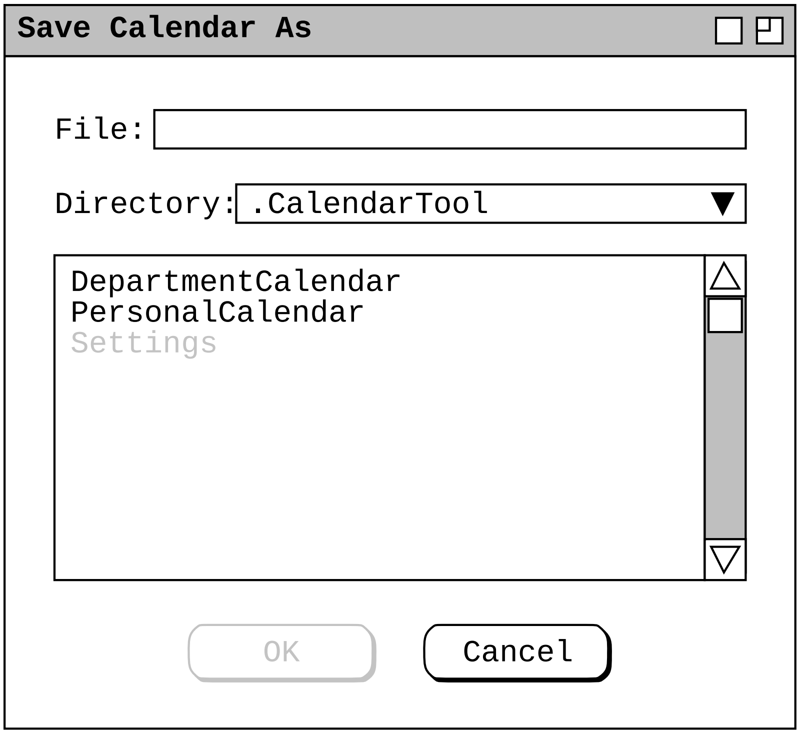


Figure 296: File save-as dialog.

enumerated above after Figure 294. When the user selects an enabled name in the list, the system enters the name in the `File` field.

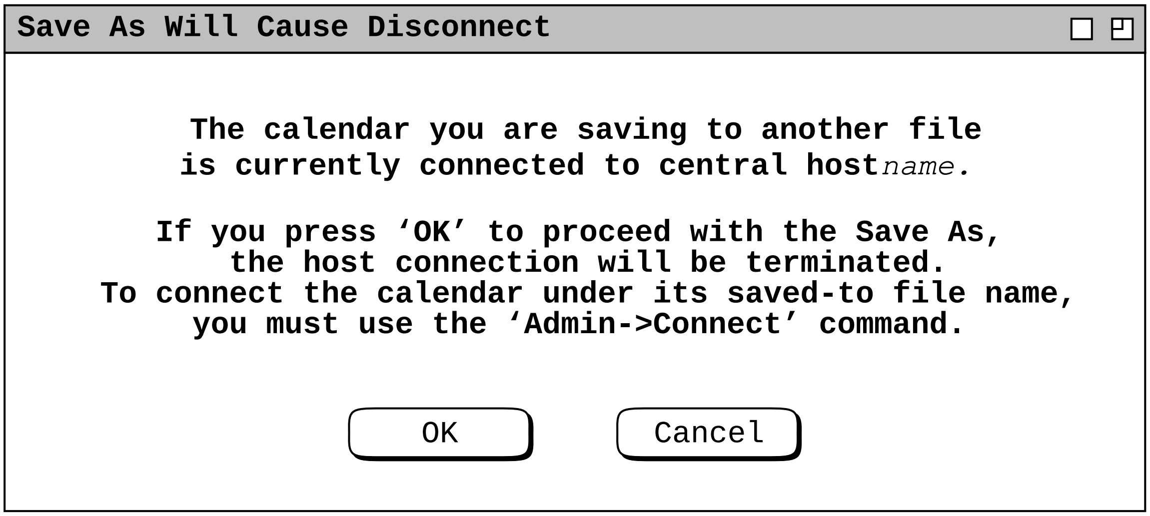
To save the calendar onto a new or existing calendar file, the user presses `OK`. If the calendar to be saved is connected to a central host, the system displays warning shown in Figure 297 before completing the save-as operation. The *name* string is the name of the host to which the current (pre-save-as) calendar is connected. If the user presses `OK` in the disconnect warning dialog, the system disconnects the calendar under its current name and proceeds with the save-as. As the dialog message explains, the user must run the `Admin->Connect` command to re-connect the calendar under its new name. If the user presses `Cancel`, the system cancels the save-as operation, leaving the current calendar connected and otherwise in its pre-save-as state.

To proceed with the save-as, the system checks if the file to be saved to exists. If it does not exist, the system completes the operation by saving a complete copy of the current calendar to the new file. If the saved calendar had unsaved changes, those changes are saved in the new saved-to file, but not in the original file. The original file remains in the same state as it was prior to `Save As` being executed. The saved calendar remains the current calendar, but with the new name instead of the old name. The system changes the name of the calendar in all of the contexts in which the original name appears. These contexts are described in Section 2.3.6.

When the user executes `Save As` to an existing calendar file, the system displays the warning dialog shown in Figure 298. If the user presses `OK`, the system proceeds with the save, fully overwriting the previous contents of the file. The states of the original and saved-to files are as described above for when `Save As` saves to a new file. The system also changes the name of the current calendar in the same way as for `Save As` to a new file. If the user presses `Cancel` the system does not perform the save, returning to the `Save As` dialog where the user can enter a different file name. In either case, the system removes the overwrite warning dialog.

The user may save a new or existing calendar to a file that is listed in the host connection table (in the `Calendar` column). In such a case, the system does not initiate a connection to the associated host as a result of the save. This is the case even if the option `Auto-connect on file open` is on. This option only applies to `File->Open`, not to `Save As`. To connect to a host of a saved-to calendar, the user must execute the `Admin->Connect` command.

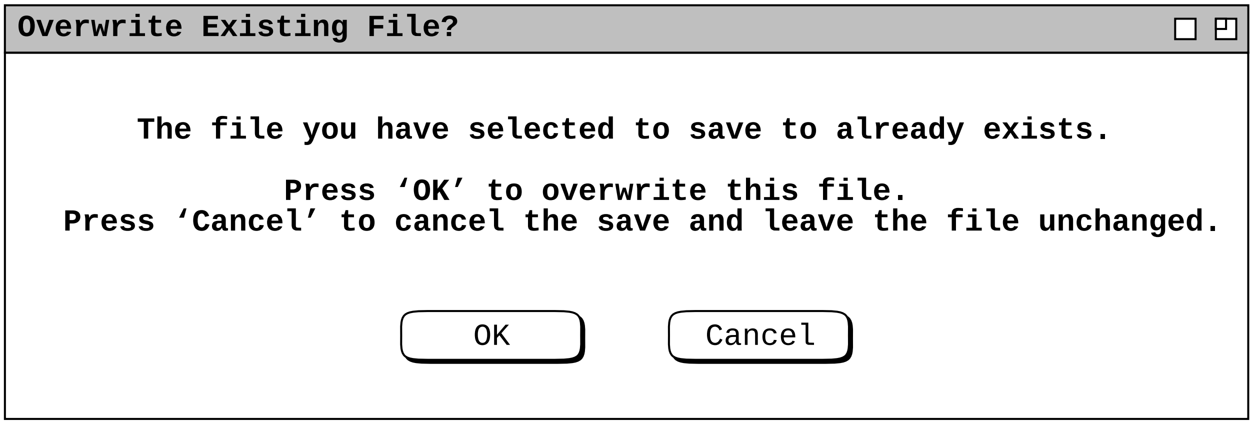
The user can type or select the calendar's current name in the save-as dialog, although this defeats the ostensible purpose of the `Save As` command. If the user does so and presses `OK`, the system saves the calendar onto the file with which is currently associated, in the same manner as if the user had executed the `Save` command.



The calendar you are saving to another file
is currently connected to central host*name.*

If you press 'OK' to proceed with the Save As,
the host connection will be terminated.
To connect the calendar under its saved-to file name,
you must use the 'Admin->Connect' command.

Figure 297: Save-as operation will cause disconnect.



Overwrite Existing File?

The file you have selected to save to already exists.

Press 'OK' to overwrite this file.
Press 'Cancel' to cancel the save and leave the file unchanged.

Figure 298: Confirm overwrite of an existing calendar file.

When the user selects '`Save All`' in the '`File`' menu, the system executes a '`Save`' for all calendars with unsaved changes. If any of the calendars is not yet saved to a file, and the user presses '`Cancel`' in the save-new-calendar dialog (Figure 294), the system cancels the save for that calendar only.

The following is a summary of the enabled/disabled states of the save commands:

a. '`Save`' is enabled when the current calendar is new and has no associated file, i.e., it has not yet been saved. Save is enabled unconditionally for a new calendar, even if the user has performed no changes to it since it was created. In this way, the user can save a completely empty new calendar.

b. '`Save`' is also enabled when the current calendar has an associated file which has unsaved changes.

c. '`Save As`' is enabled whenever at least one calendar is open, and the current calendar is not that of another user or group. '`Save As`' is enabled whether or not the current calendar has unsaved changes.

'`Save`' and '`Save As`' are disabled otherwise.

None of the save operations affects the state of pending edit dialogs. Unconfirmed or unapplied edits are not saved, and the states of all dialogs associated with a calendar are unchanged by a save of the calendar.

As described above, the system attempts to disallow saving to an unwritable calendar file. However, if the user changes the permissions of a calendar file externally, before or during Calendar Tool execution, it is possible for the user to execute a '`Save`' or '`Save As`' to an unwritable file. The circumstances where this can happen are described in Section 2.8.12. When the user does execute a '`Save`' or '`Save As`' to an unwritable file, the system responds with an appropriate error message, as described in Section . If this error is reported for a file during a '`Save All`' sequence, the system continues with the '`Save All`' after the user dismisses the error message.

2.8.5. Loading and Saving Settings

As outlined above, Calendar Tool settings are stored in the standard `Settings` file, as well as in other named files in which the user chooses to save them. The commands to save and load settings allow the user to bundle different settings to be used in different types of calendars. For example, the user can define a collection of work-related settings, save them in a settings file, and then load them for use in work-related calendars. Users or administrators can share such settings files with other users, so that a particular user community has a common collection of settings. Administrators can create useful settings files for inclusion when the Calendar Tool is installed, as described in Section

Since calendar files contain their own calendar-specific settings, the sharing functionality provided by settings files could be accomplished in part by sharing just calendar files themselves. However, settings files provide the advantage of being able to copy settings from one calendar file to another, without adding or deleting scheduled items in either calendar. Also, since calendar files do not contain session-wide settings, the only way to save and load these are through independent settings files.

The system loads the standard `Settings` file under the following circumstances:

a. when the user starts up the Calendar Tool from the operating environment
b. when the user invokes the '`File->Load Settings`' command and selects the `Settings` file to be loaded

When the system loads `Settings` at start up, it uses the session-wide component of the settings file to initialize global options, initialize the host connection table, and to determine which calendars to open at start-up. The initialized global options are those used in the ensuing session of the Calendar Tool, and those presented to the user when she executes the '`Options->Global`' command. The initialized host connection table is that which is used during the ensuing session, and that which is presented to the user when she runs the '`Admin->Connect`' command. The list of which calendars, if any, to open at start-up is in the '`names of open calendars`' component of the settings file. Further discussion of the Calendar Tool start-up procedure is in Section .

The system does not use the calendar-specific component of the `Settings` file at start-up. Rather, for each calendar it opens at start-up, it uses the calendar-specific settings stored in that calendar's file. This includes the calendar-specific settings for viewing options and the setting for the '`state of open windows`' for each calendar. These settings determine which windows are opened for each start-up calendar. The calendar-specific component of the `Settings` file is used when the user creates a new file, as described in Section 2.8.2.

When the user runs the '`Load Settings`' command in the '`File`' menu, the system displays the dialog shown in Figure 299. The user may select via checkboxes which parts of a settings file to load, and from which file to perform the load. All of the checkboxes are on by default. When the filter settings checkbox is on, the system automatically checks and disables the categories checkbox. This is because custom filter definitions may be defined in terms of calendar-specific categories. If the user unchecks the filters checkbox, the categories checkbox is enabled for independent checking or unchecking.

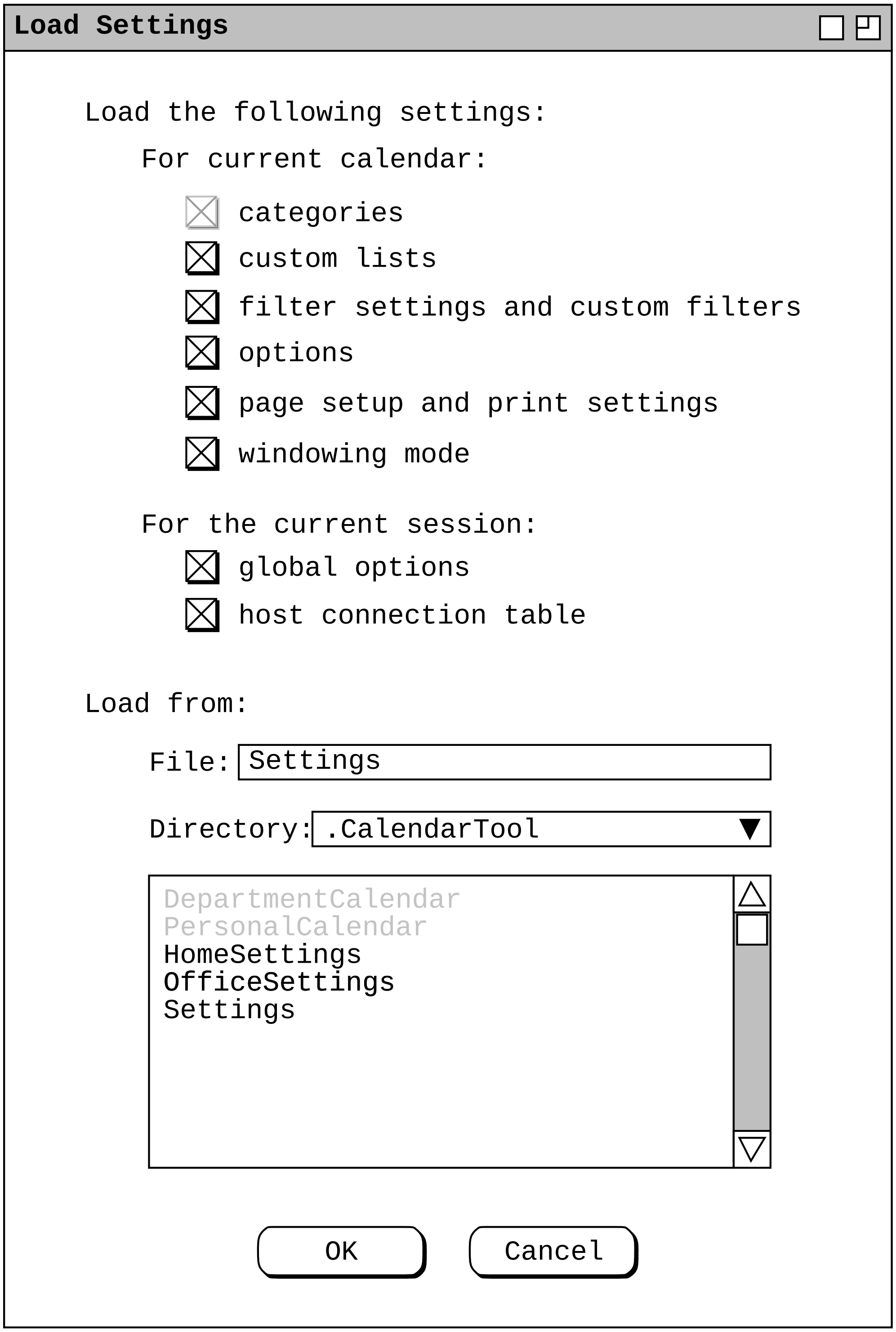
If the user invokes '`Load Settings`' when there is no current open calendar, the system unchecks and disables all of the calendar-specific checkboxes under '`For the current calendar`'. In this case, only the settings for global options and the host connection table can be loaded. If the user invokes '`Load Settings`' when one or more calendars is connected, the '`host connection table`' checkbox is unset and disabled. This means that the host connection table cannot be loaded from settings when there any active connections.

The default file in the load-settings dialog is the standard `Settings`, in the Calendar Tool user directory. The dialog shown Figure 299 reflects the user having previously saved two settings files in addition to the pre-installed "`Settings`". Only Calendar Tool settings files and directories are enabled in the scrollable file list. The user can type, navigate, and select in the dialog in the same manner as described for the '`Open`' dialog in Figure 288.

Figure 300 shows the user having chosen to load a subset of the settings stored the file "`OfficeSettings`". When the user presses '`OK`', the system displays the confirmation dialog shown in Figure 301. The dialog explains that loading settings may cause changes to the calendar and its display windows, and that changes will be made without further confirmation. The dialog further explains that any currently unconfirmed or unapplied settings edits will be canceled. What this means precisely is that if the user has edited by not applied or confirmed changes to settings that are being loaded, the dialogs in which the edits have been performed are canceled. This cancellation applies only to dialogs for settings checked on in the load-settings dialog. Edit dialogs for unchecked settings are not canceled. For example, if the user has edited a category definition but not yet confirmed the edits, and the '`categories`' checkbox is on, the system cancels the categories editing dialog and loads the settings. On the other hand, if the user has unapplied edits to calendar-specific options, but the '`options`' checkbox is off in the load-settings dialog, then the options dialog is unaffected by the settings load, and the dialog remains open with the still unapplied edits.

If the users answers '`OK`' in the load-settings dialog, the system performs the following actions:

a. copies the values for calendar-specific settings from the settings file into the current calendar; this is done for the settings with the checkbox turned on; unchecked settings values are not copied, leaving them unchanged in the current calendar
b. confirms and applies all setting changes



Load Settings

Load the following settings:

For current calendar:

☒ categories
☒ custom lists
☒ filter settings and custom filters
☒ options
☒ page setup and print settings
☒ windowing mode

For the current session:

☒ global options
☒ host connection table

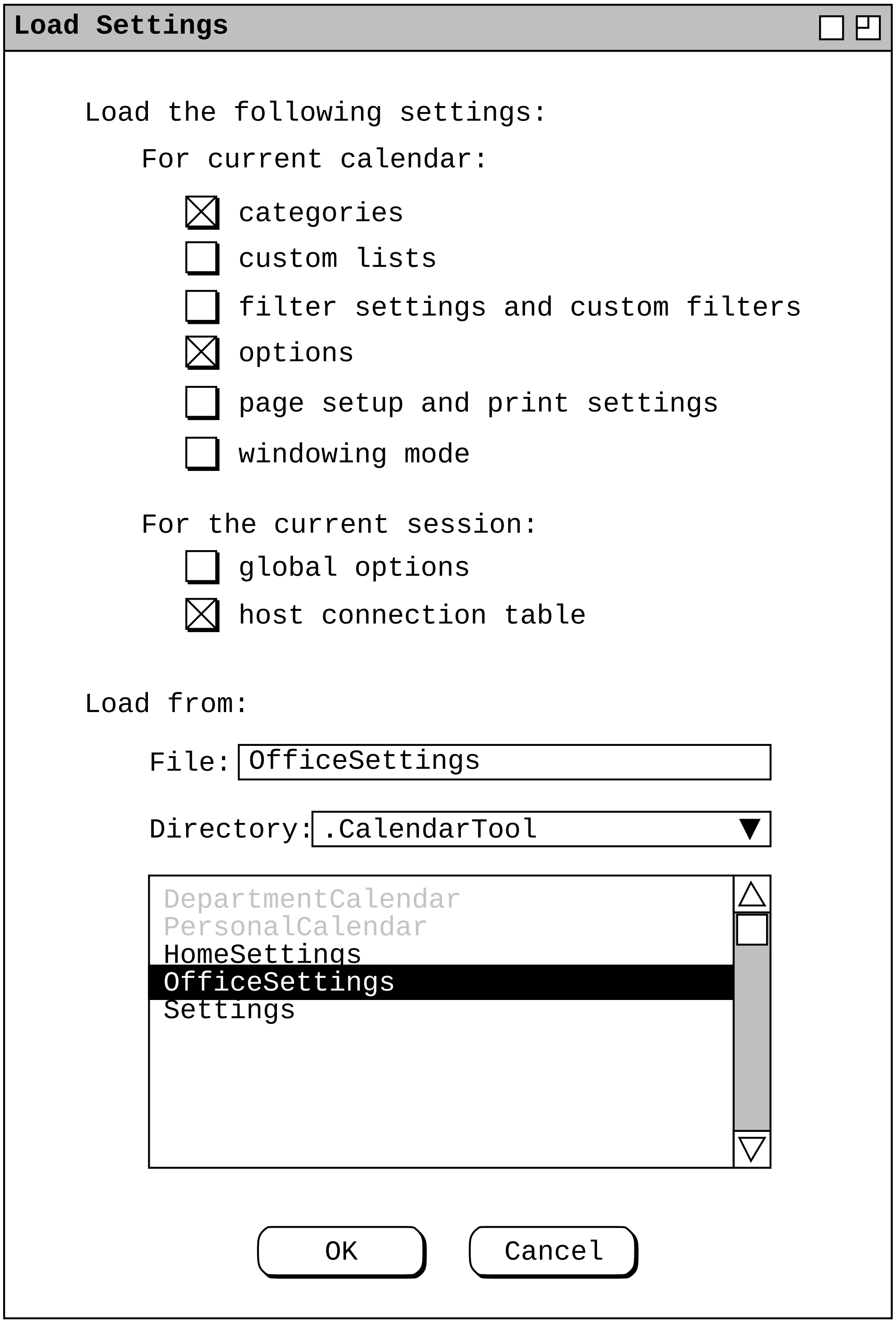
Load from:

File: Settings

Directory: .CalendarTool

DepartmentCalendar
PersonalCalendar
HomeSettings
OfficeSettings
Settings

Figure 299: Load-settings dialog.



Load Settings

Load the following settings:

For current calendar:

☒ categories
☐ custom lists
☐ filter settings and custom filters
☒ options
☐ page setup and print settings
☐ windowing mode

For the current session:

☐ global options
☒ host connection table

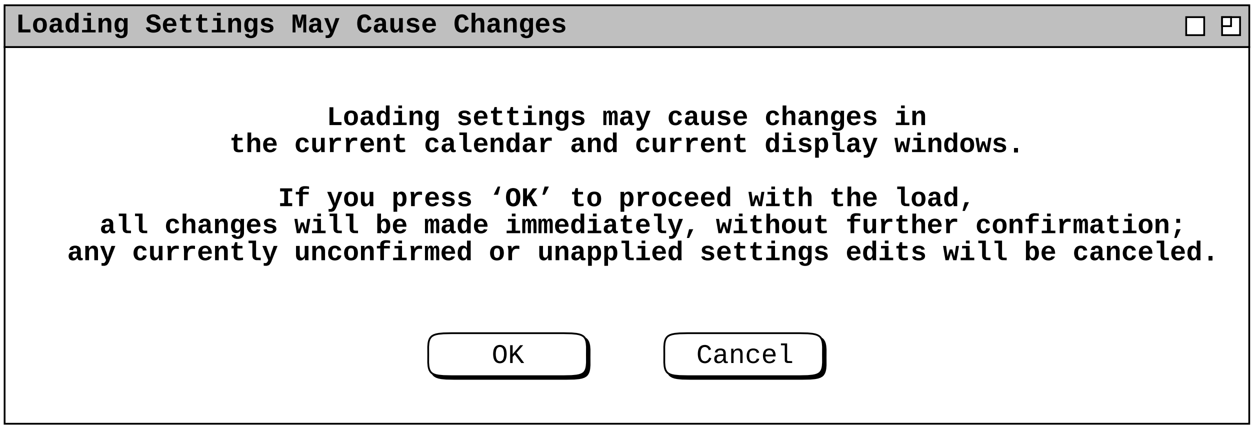
Load from:

File: OfficeSettings

Directory: .CalendarTool

DepartmentCalendar
PersonalCalendar
HomeSettings
OfficeSettings
Settings

Figure 300: Load-settings dialog filled in.



Loading Settings May Cause Changes

Loading settings may cause changes in the current calendar and current display windows.

If you press ‘OK’ to proceed with the load, all changes will be made immediately, without further confirmation; any currently unconfirmed or unapplied settings edits will be canceled.

Figure 301: Confirm load settings.

c. performs all processing that results from changed, added, or deleted calendar-specific settings

d. re-initializes the session’s global options and host table from the settings file; this is performed when the associated checkbox is on; if either or both checkbox is off, the global settings and/or connection table are left unchanged for the current session

When the system performs copying of calendar-specific settings, the values from the settings file completely replace all of the calendar-specific settings values in the current calendar. If values copied from the settings file are the same as those in the current calendar, then those values are effectively unchanged.

The effective change of values is particularly relevant to the definition of categories, custom lists, and custom filters. If any of these definitions are exactly the same in the settings file and the current calendar, then these definitions remain effectively unchanged in the calendar. Any definitions in the settings file that are not the same as the current calendar cause a change, addition, or deletion to the calendar. Table 16 specifies precisely what constitutes the same definition, a changed definition, a definition to be added, and a definition to be deleted.

Once calendar-specific settings values are loaded, the system confirms and applies all loaded values. In particular,

- *o* categories, custom lists, and custom filter definitions are confirmed and applied
- *o* changed option settings are applied
- *o* page set up and print settings are confirmed
- *o* the effects of a change to the windowing mode takes effect when the user executes the next affected viewing command

The system then performs all processing that results from changed, added, or deleted settings values. If this

Type of Definition	**Same**	**Change**	**Addition**	**Deletion**
category	same name and color	same name with different color	new name in settings file that is not in calendar	name in calendar that is not in settings
custom list and custom filter	same name and values	same name with one or more different values	new name in settings file that is not in calendar	name in calendar that is not in settings

Table 16: Comparison of definitions in settings file versus current calendar.

processing would require further user confirmation, the system proceeds as if the confirmation is affirmative (‘`OK`’ or ‘`Yes`’). The sections of the requirements that specify the processing associated with each type of calendar setting are those cited parenthetically in the list at the beginning of Section 2.8.1.

Changes to category definitions and options are the only setting changes that may result in immediate changes to scheduled items in the current calendar. The category changes that have immediate effect are those described in Section 2.5.5 that result from color changes or deletions of category definitions. Changes in meeting-scheduling options can have immediate effect if the current calendar is connected to a host and the loaded options change the setting of when pencil-ins are accepted to ‘`whenever connected to the originating host`‘.

Changes to custom lists, filters, options, and windowing mode may result in changes to the appearance of scheduled items in display windows, but not in the items themselves. Changes to page setup and print settings take effect only when the user executes the ‘`Page Setup`’ and ‘`Print`’ commands.

When the system re-initializes the host connection table, it completely replaces the contents of current the session-wide table with the entries in the settings file. The "session-wide table" is that in use in the current Calendar Tool session, as it appears when the user runs the ‘`Admin->Connect`’ command. To avoid undue disruption, the checkbox to load host connection settings is unset and disabled when one or more hosts is connected in the current Calendar Tool session. Changes to the host connection table may lead to changes in scheduled items being downloaded from the host. However, these changes are not immediate when settings a loaded, because no hosts can be connected when the host connection table is loaded from a settings file. Complete discussion of the host connection table is in Section 2.6.6.1.

It is noteworthy that two parts of the settings file are missing from the load−settings dialog. These are the the calendar-specific setting for the ‘`state of open windows`’, and the session−wide setting for the ‘`names of open calendars`’. These settings are used, and are only manageably meaningful, when the settings file is loaded at start−up, or in the case of open windows, when a calendar is opened.

When the user selects the ‘`Save Settings`’ command, the system responds by displaying the dialog shown in Figure 302. The figure reflects the state of the Calendar Tool user directory before any settings files have been saved. The dialog is very similar in content to that for loading settings, with the addition of the two settings noted just above as missing from the load dialog. The user may select via checkboxes which parts of a settings file to save, and to which file to perform the save. All of the checkboxes are on by default. When the filter settings checkbox is on, the system automatically checks and disables the categories checkbox. This is because custom filter definitions may be defined in terms of calendar-specific categories. If the user unchecks the filters checkbox, the categories checkbox is enabled for independent checking or unchecking.

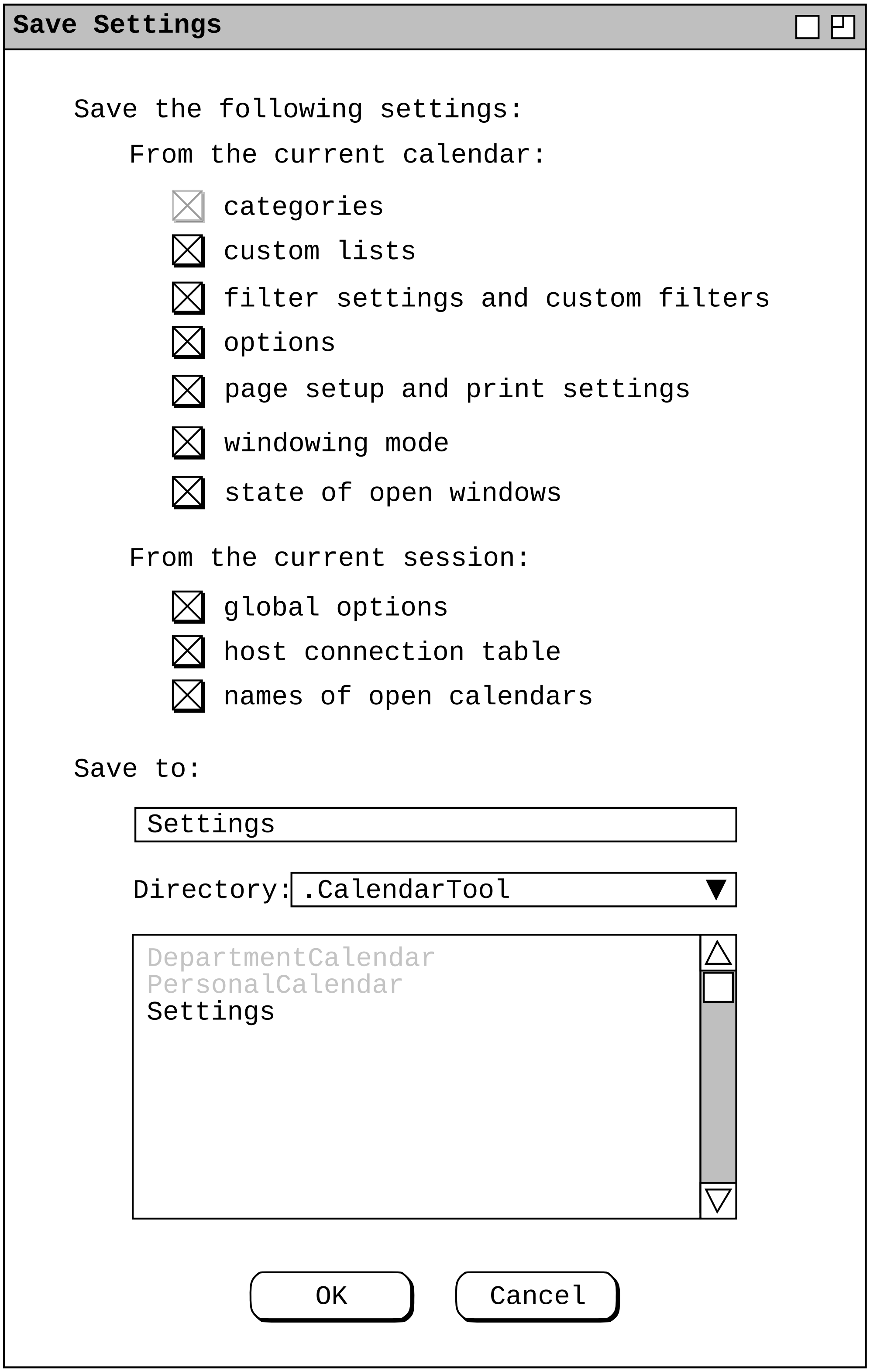
If the user invokes ‘`Save Settings`’ when there is no current open calendar, the system unchecks and disables all of the calendar-specific checkboxes under ‘`From the current calendar`’. In this case, only the session-wide settings can be saved. If the user invokes ‘`Save Settings`’ when one or more calendars is connected, the settings can still be saved, but the connected state is ignored. That is, the host connection table is always saved with the ‘`Connected?`’ column blank.

The default file in the save-settings dialog is the standard `Settings`, in the Calendar Tool user directory. The values shown Figure Figure 302 reflect the initial state of the dialog, with the pre-installed "`Settings`" file, before the user has saved to any other settings files. Only Calendar Tool settings files and directories are enabled in the scrollable file list. The user can navigate and select in the dialog in the same manner as described for the ‘`Save As`’ dialog in Figure 296. The ‘`OK`’ button is enabled except when the user does one of the following:

- *o* erases all typing
- *o* enters the name of an unwritable settings file
- *o* enters the name of an unreadable directory
- *o* enters the name of an existing non-settings file

The system disallows saving in any of these circumstances. If the user continues typing to change these circumstances, the system re−enables the ‘`OK`’ button.

Figure 303 shows the user having chosen to save a subset of the settings to the file "`OfficeSettings`". When the user presses ‘`OK`’, the system performs the following actions to complete the save:



Save Settings

Save the following settings:

From the current calendar:

- ☒ categories
- ☒ custom lists
- ☒ filter settings and custom filters
- ☒ options
- ☒ page setup and print settings
- ☒ windowing mode
- ☒ state of open windows

From the current session:

- ☒ global options
- ☒ host connection table
- ☒ names of open calendars

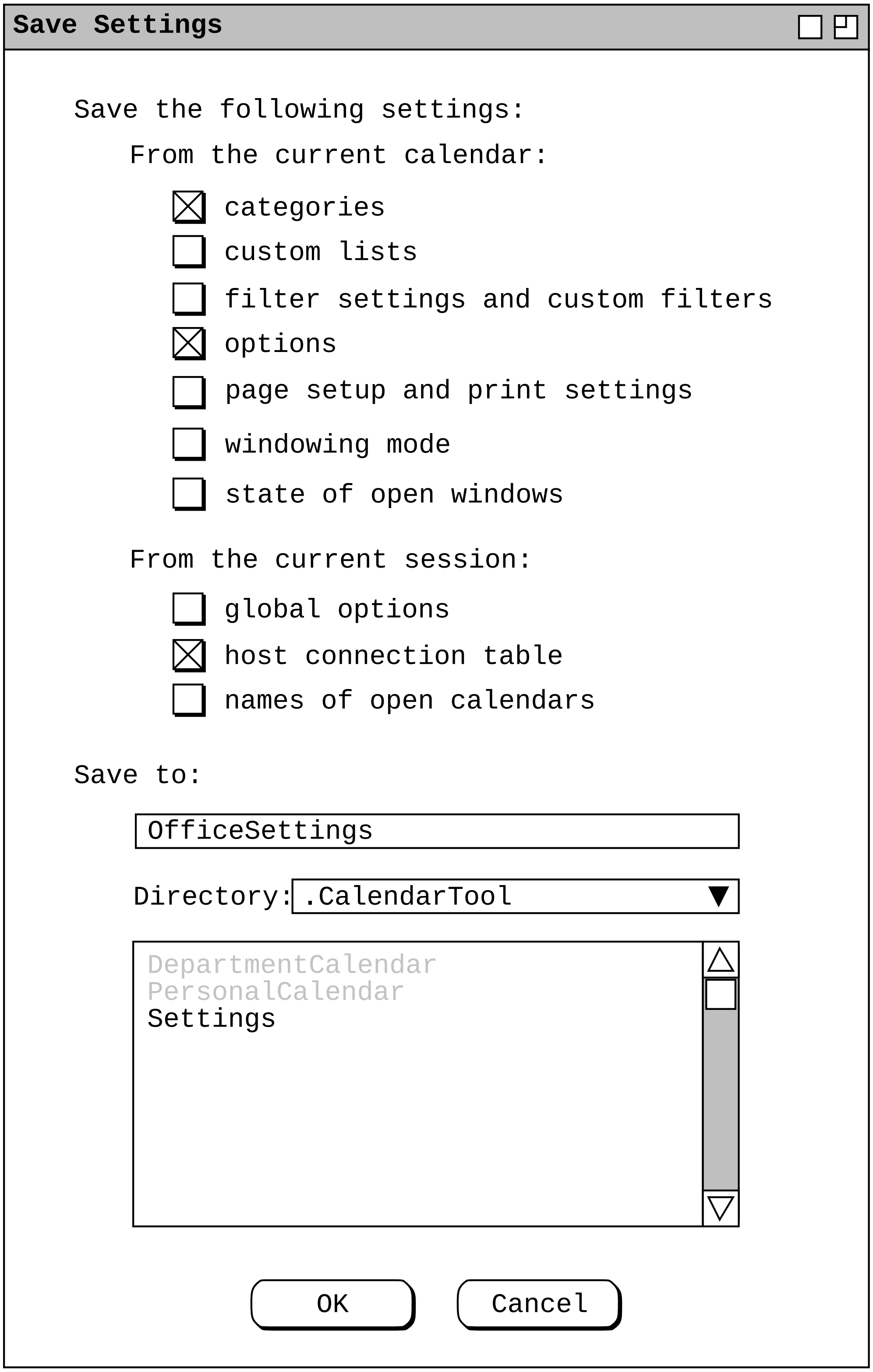
Save to:

Settings

Directory: .CalendarTool ▼

DepartmentCalendar
PersonalCalendar
Settings

Figure 302: Save-settings dialog.



Save Settings

Save the following settings:

From the current calendar:

☒ categories
☐ custom lists
☐ filter settings and custom filters
☒ options
☐ page setup and print settings
☐ windowing mode
☐ state of open windows

From the current session:

☐ global options
☒ host connection table
☐ names of open calendars

Save to:

OfficeSettings

Directory: .CalendarTool

DepartmentCalendar
PersonalCalendar
Settings

Figure 303: Save-settings dialog filled in.

a. copies the checked calendar-specific settings from the current calendar to the selected settings file

b. copies the checked session-wide settings to the selected settings file

- for the unchecked calendar-specific settings, the system does one of the following:
  i. if the saved-to settings file exists, the unchecked settings in the file are not changed
  ii. if the saved-to settings file does not exits, the system copies the built-in default values to the settings file; Table 15 defines the values of built-in calendar-specific settings

c. for unchecked session-wide settings, the system copies the built in values, which are as shown in Figure 283 for global options, and are empty for both the host connection table and list of open calendars

In saving the state of open windows, the system saves all physical details of all windows associated with the current calendar. These details are the window content, size, position, and state of iconification. In saving calendar view windows, the system uses the specific absolute dates in the display, not dates relative to today's date. For example, if there is a month view for September in an active window, the system saves the state of that window as open for that specific month. If the user subsequently opens the calendar with the saved settings in, say, October, the month view opened from the saved settings is still September. The system always highlights today's date in any calendar view that contains today's date. However, if a view saved in the settings contains today's date when it is saved, but not when the settings are later loaded, the view cannot and does not have today's date highlighted. The point again is, views saved in the '`state of open windows`' setting are saved with the absolute dates displayed at the time of saving, and the dates are not adjusted relative to today's date when the settings are subsequently loaded. If the user wants the initial view of a calendar to be relative to today's date, she can use a viewing option other than '`per most recently saved settings`', as described in Section 2.7.4.3.

The system does not display an overwrite warning if the user saves to an existing settings file. This is because the user may frequently save settings to an existing file, in particular the `Settings` file in the Calendar Tool directory. In this sense, overwriting an existing settings file is more a normal than exceptional activity.

When the user saves settings to the standard `Settings` file, the saved values become those that are used as the defaults when the Calendar Tool starts up, and as the defaults for new calendars. In this way, the current calendar serves as the basis for the standard default settings. If the user wants different values for the standard defaults, she can change the current calendar and save again to the standard `Settings` file.

Confirming the '`Save Settings`' command results in a change to the selected settings file, but not to the current calendar. Hence, the system performs none of the change-related processing associated with loading settings shown in Table 16.

Table 17 summarizes all of the ways that the user can save settings to a calendar file or to a settings file.

2.8.6. Page Setup and Print

Printing from the Calendar Tool is based strictly on the content visible in active Calendar Tool windows. Window content appears on the printed page exactly as it appears on the screen. "Exactly" means the content of the printed page is the same in appearance and absolute size as it appears on the screen. Scrolling lists and other scrollable components of the display are not expanded in the printed output. The printed content excludes the window banner, any bordering other than a single black line, and any content that is not visible within the window frame on the screen. This means there is no page numbering, header, footer, or other page labeling.

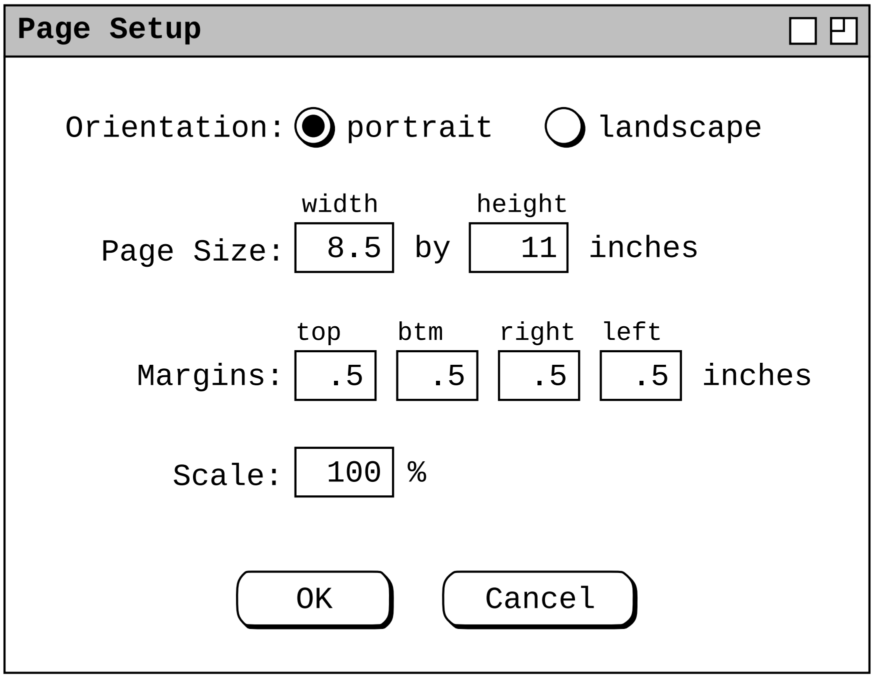
When the user selects the '`Page Setup`' command in the '`File`' menu, the system displays the dialog in Figure 304.

The '`Orientation`' setting defines the width-by-height orientation of pages. With the default selection of '`portrait`', pages are laid out with the `width` dimension running horizontally and the `height` dimension running vertically. When '`landscape`' is on, the `width` dimension runs vertically and the `height` dimension horizontally.

The '`Page Size`' values define the size of a page in inches. The '`Margins`' values define the unprintable area between the edge of the page and the edge of the printed content. When possible, the Calendar Tool queries the underlying operating environment to determine page size and margin settings for the currently selected printer.

Command	Type of saving performed
‘`Save`’ command in the ‘`File`’ menu	All calendar-specific settings in the current calendar are saved to the file associated with current calendar; there is no change to any settings file, and in particular, session-wide settings are not saved anywhere.
‘`Save As`’ command in the ‘`File`’ menu	All calendar-specific settings in the current calendar are saved to the selected calendar file; there is no change to any settings file, and in particular, session-wide settings are not saved anywhere.
‘`Save Settings`’ command in the ‘`File`’ menu	Checked settings are saved to the selected settings file; unchecked settings are unchanged in an existing file, defaults in a new file; no calendar file is affected.
‘`Save`’ button in the global options dialog	Global options are saved to the `Settings` file in the user’s Calendar Tool directory; all other settings are unchanged in the `Settings` file; no calendar file is affected.
‘`Save`’ button in the host connection dialog	The host connection table is saved to the `Settings` file in the user’s Calendar Tool directory; the ‘`Connected?`’ column is fully blank in the saved table; all other settings are unchanged in the `Settings` file; no calendar file is affected.

Table 17: Summary of ways to save settings.



Orientation: ● portrait ○ landscape

Page Size: width 8.5 by height 11 inches

Margins: top .5 btm .5 right .5 left .5 inches

Scale: 100 %

Figure 304: Page setup dialog.

When such determination is made, the system inserts the values as the defaults in the page setup dialog. If the system cannot make this determination, the default width and height values are 8.5 and 11 inches, the default margins are all .5 inch.

The user can manually adjust the page size and margin settings to accommodate the paper size and margin requirements of different printers. In doing so, it is the user's responsibility to enter page size or margin values that can be accommodated on a particular printer, such that printed content is not cropped on the page.

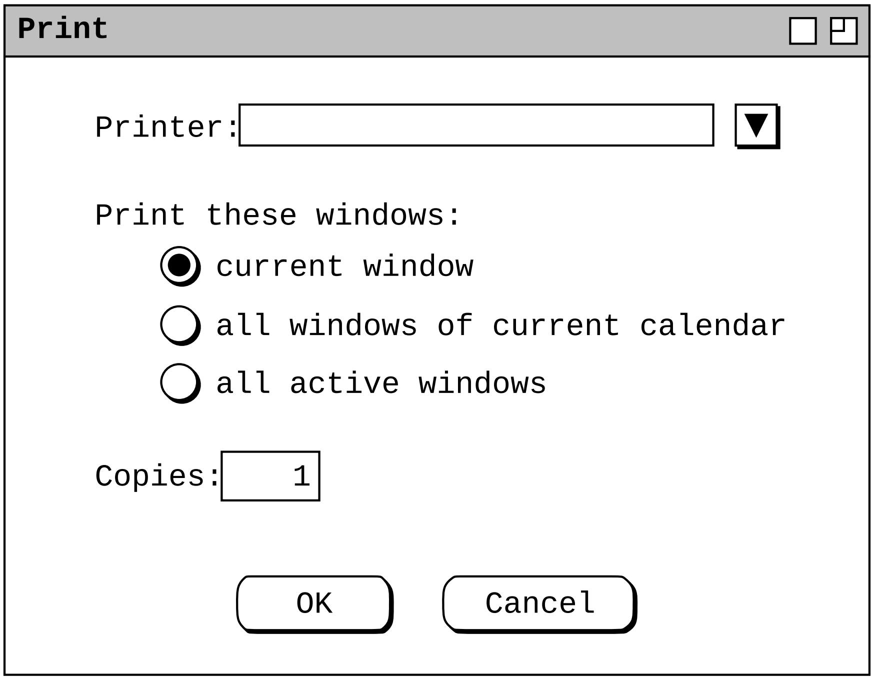
The '`Scale`' setting defines the scale of the printed content relative to the printed page. The default is 100%.

When the user selects the '`Print`' command in the '`File`' menu, the system displays the dialog in Figure 305. The user selects the printer to which output is directed from a list of available printers. When possible, the Calendar Tool queries the underlying operating environment to determine the identity of the user's preferred printer. When such determination is made, the system inserts the name of the printer as the default value in the '`Printer`' field. If the user's preferred printer cannot be determined, the field is blank.

The user can select to print the current window only, all windows of the current calendar, or all active windows. Per Section 2.3.6, an active window is any that appears in the '`View->Windows`' submenu. Iconified (i.e., minimized) windows are considered active. The successive print order of multiple windows is based on the top-to-bottom order of windows in the '`View->Windows`' submenu.

The user may select the number of copies of each window to print, the default being 1. The '`Collated`' checkbox is enabled if the user selects more than 1 copy. If '`Collated`' is off (the default), the system prints all copies of the same window in immediate succession, before printing the next window. If the user turns '`Collated`' on, the system prints one copy of each window in succession before beginning the next copy of the each window.

When the user presses '`OK`' in the print dialog, the system proceeds with the printing. If a window fits on a single page based on the page setup parameters, the system prints the window centered on the page. The system does not print more than one window per page, even if more than one window could fit on a single page. If a window is too large to fit on one page, the system prints as much per page as possible, printing the multiple pages of the same window in row-wise order. The multiple pages of a single window are all left and top justified to the defined margins, not centered as for single-page windows.



Printer:

Print these windows:

● current window

○ all windows of current calendar

○ all active windows

Copies: 1

Figure 305: Print dialog.

There may well be additional functionality for page set up and printing in specific operating environments. This includes the ability to set other printer parameters, preview printed pages, and define other forms of page layout. Implementors must provide all such available functionality. Implementors are required to provide only generic page layout functionality, not content-specific layout. Contentwise, layout is at most one window per page, as described above.

2.8.7. Exit

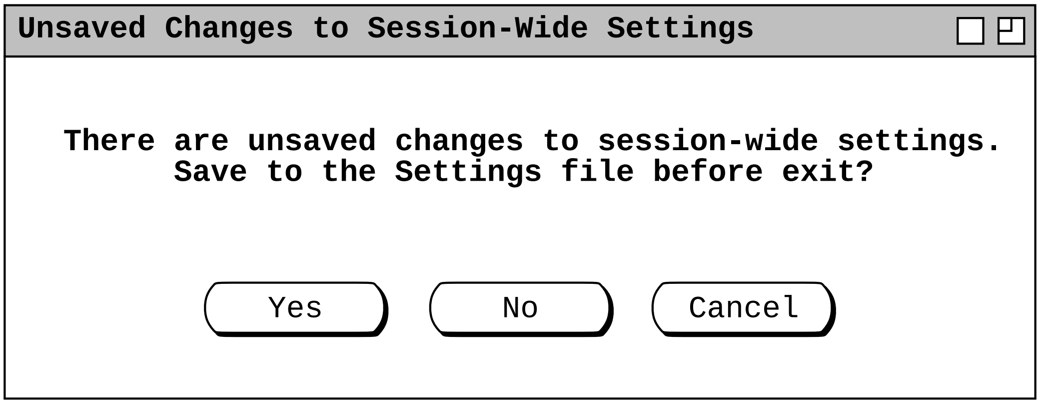
When the user selects the ‘`Exit`’ command in the ‘`File`’ menu, the system begins the exiting process by executing the ‘`Close All`’ command. The associated processing for ‘`Close All`’ ensues, including any necessary user interaction through the warning dialogs. This is described in Section 2.8.3. If the user responds at any point such that the ‘`Close All`’ is canceled for all remaining calendars, the ‘`Exit`’ operation is also thereby canceled.

Once the ‘`Close All`’ has run to completion, the system checks if there are unsaved session-wide settings. If so, the system displays the dialog shown in Figure 306. If the user answers ‘`Yes`’, the system saves all session-wide settings to the `Settings` file in the user’s Calendar Tool directory. If the user answers ‘`No`’, the system does not save session-wide settings, thereby losing any unsaved changes. If the user presses ‘`Cancel`’, the systems cancels the ‘`Exit`’ command and removes the offer–to–save–settings dialog from the screen, leaving all calendars closed but the Calendar Tool still running.

When the user answers ‘`Yes`’ or ‘`No`’, or if the session-wide warning dialog was not displayed, the system proceeds with the ‘`Exit`’ operation by terminating the current session of the Calendar Tool, thereby returning the user to the underlying operating environment.

2.8.8. Files Used by Calendar Tool Administration

Sections 2.6, and , contain general discussions about the data repositories on Calendar Tool central host computers. This section presents specific details of the directory and file structure of the repositories.



There are unsaved changes to session-wide settings.
Save to the Settings file before exit?

Figure 306: Offer to save session-wide settings before exit.

A central host data repository consists of the following storage components:

- *o* a user database, holding Calendar Tool user records but no user calendars
- *o* a group database, holding Calendar Tool group records
- *o* a location database, holding meeting location records and other-usage booking records
- *o* a directory containing user and group calendars, one calendar per file
- *o* a `Settings` file containing administrator password, email address, and options

The internal structure of repository directories and files is implementation−dependent. The externally−visible names and organization of the files are shown in Table 18.

When the Calendar Tool Administration program is installed, initial versions of the repository files and directories are created, as described in Section The file and directory names shown in Table 18 are fixed. The three databases are either single files or directories organized in some implementation−specific manner. The Calendars directory must contain one file for each user and group calendar, with user and group ID as the root filename. The internal structure of the files and possible filename extensions is left to the implementors.

The fact that the databases may be a file or a directory allows the implementors to use a database representation that is either in a single file or multiple files in a parent directory. Independent of this level of implementation detail, the implementors must meet the following specific requirements:

a. the three databases must be in three separate files or directories

File or Directory	Usage
Users	user database file or directory
Groups	group database file or directory
Locations	location database file or directory
Calendars	directory of user and group calendar files, with each root file name identical to the Calendar Tool user or group ID of its owner
Settings	administrative settings

Table 18: Central Host Data Files.

b. the names of the files or directories must be those in Table Table 18
c. user and group calendars must each be stored in a separate file

The purpose of these requirements is to allow the administrator readily to assess the size of databases and calendars using the normal file sizing functionality of the underlying operating environment.

The repository files and directories are themselves stored in a single directory, specified initially when the Calendar Tool Administration program is installed. This superdirectory is the root of the entire central host repository. If there are conventions in a particular operating environment for where such application–specific files should be stored within the file system, then such conventions can be followed during installation. After installation, the administrator can change the location of the repository directory, if necessary. This is done by changing the administrative option setting described in Section 2.7.7. The repository directory may contain other files at the discretion of the administrator. In particular, an administrator may choose to create settings or sample calendar files, to be distributed to users as a convenience. These files are created with the regular-user Calendar Tool, run by an administrator who has privileges to save files in the repository directory.

2.8.9. Administrative Save and Save Copy

When the administrator executes the '`Save`' command in the administrative file menu, the system saves all changes made to any data since the last save. Changes to "any data" mean any data added, changed, or deleted in any of the administrative dialogs discussed in Sections 2.6.2 through 2.6.5 of the requirements. As discussed in Section 2.8.8, administrative data are stored in fixed files in a fixed location; these are the files to which data are saved.

To save a copy of all repository data, the administrator uses the '`Save Copy`' command. In response, the system displays the dialog shown in Figure 307. The administrator selects the parent directory into which a copy of all repository files is to be saved. The administrator can navigate and select in the dialog in the same manner as described for the '`Save As`' dialog in Figure 296. The '`OK`' button is enabled whenever a writable directory is selected in the scrolling list. For example, Figure 308 shows the administrator having selected to save a copy of the repository in the directory "`/usr/local/caltool/Backup`", which in his case is a subdirectory of the

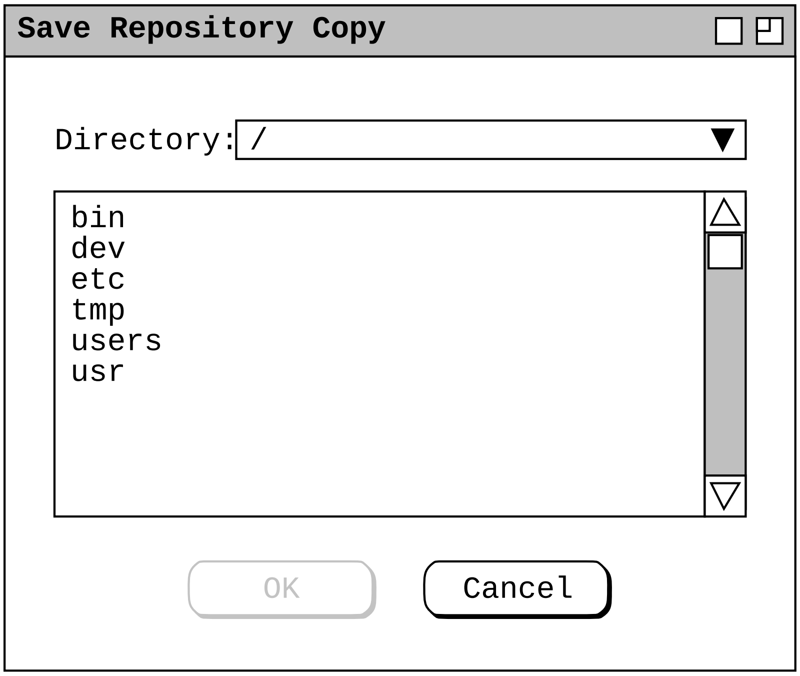


Figure 307: Save-copy dialog.

repository itself.

When the administrator presses '`OK`', the system saves a complete copy of all repository files in the selected directory. A "complete copy" means that all directories and files listed in Table 18 are fully copied. No other files that happen to be within the repository superdirectory are copied. If there are one or more files or directories within the destination directory of the same name as the copied files, the system provides no overwrite warning. Existing directories are written into, existing files are fully overwritten.

The save-copy command does not affect the files in the fixed repository location, and does not change the location of the fixed repository. When the administrator subsequently executes the '`Save`' command, data are saved to the fixed repository.

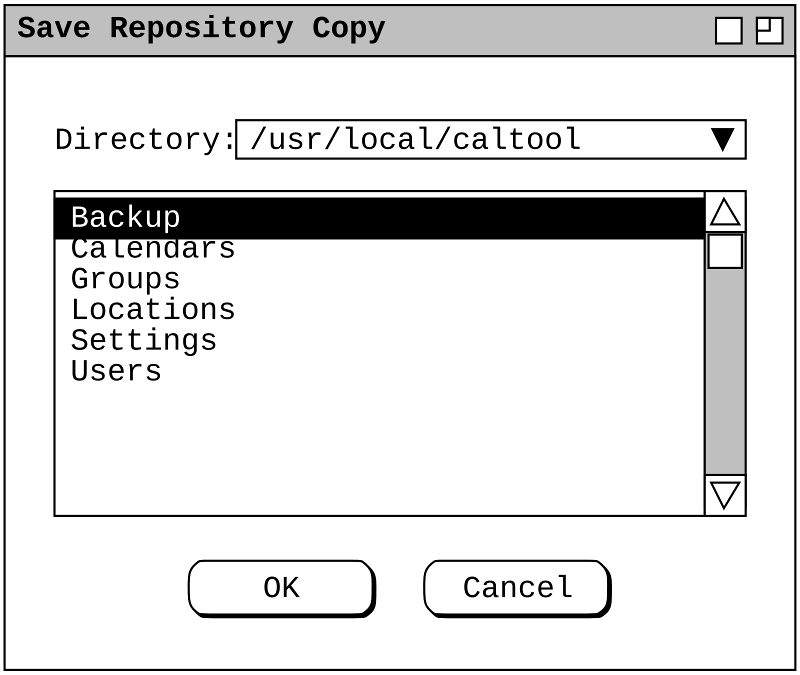
2.8.10. Administrative Page Setup and Print

Administrative page setup and print commands provide precisely the same functionality as in the regular-user Calendar Tool, described in Section 2.8.6. As with Calendar Tool printing, output is based strictly on the content visible in active administrative windows. The page-setup and print dialogs are the same.

2.8.11. Administrative Exit

When the administrator executes the '`Exit`' command, the system checks if there are any unsaved changes to repository files or pending dialog edits. If either or both of these conditions is true, the system displays the dialogs of Figures 309 and 310, respectively.

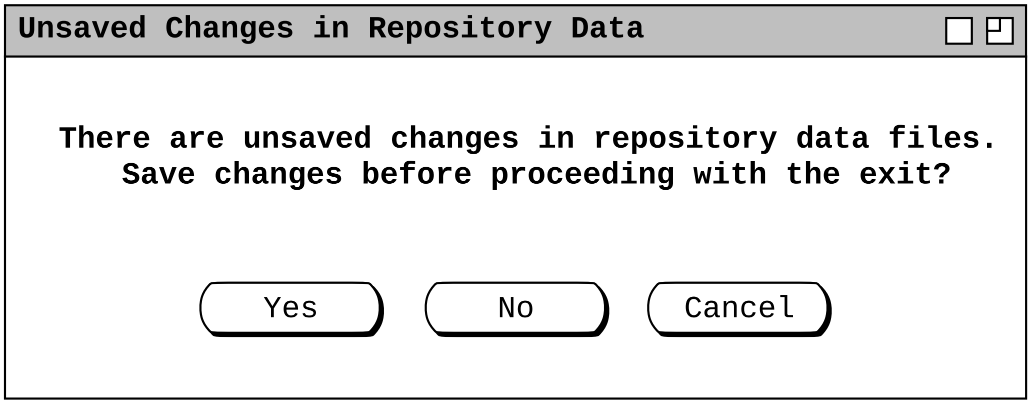
If the administrator answers affirmatively in both cases, or if the dialogs were not displayed, the system proceeds with the '`Exit`' by terminating the current session of the Calendar Tool Administration program, thereby returning the administrator to the underlying operating environment. Termination of the administration program does not terminate the Calendar Tool server program. The server remains running asynchronously until the administrator executes the '`Stop Server`' command, as described in Section 2.6.5.1.



Directory: /usr/local/caltool

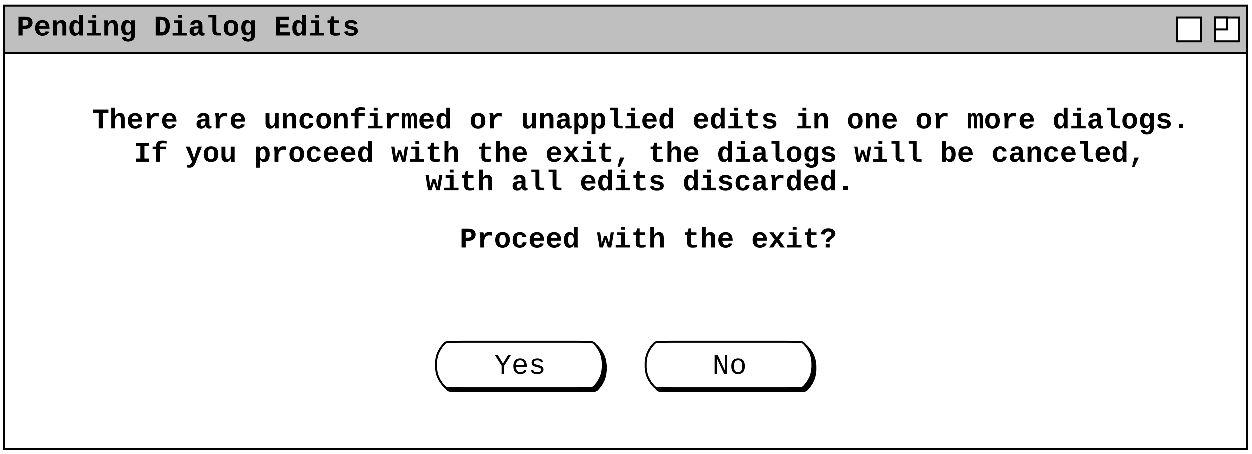
Backup
Calendars
Groups
Locations
Settings
Users

Figure 308: Save-copy dialog filled in.



There are unsaved changes in repository data files.
Save changes before proceeding with the exit?

Figure 309: Offer to save repository data before closing.



There are unconfirmed or unapplied edits in one or more dialogs.
If you proceed with the exit, the dialogs will be canceled,
with all edits discarded.

Proceed with the exit?

Figure 310: Warning when there are pending edits in administrative dialogs.

2.8.12. External Changes to Calendar Tool Files

As explained in the preceding scenarios of this section, both the Calendar Tool and Calendar Tool Administration program normally attempt to disallow opening and saving ineligible files. However, if a user changes or deletes files when the Calendar Tool or Calendar Tool Administration are running, normal operation may be disrupted. Section enumerates the conditions under which file-related errors can occur and the associated error messages.

2.8.13. Operating Environment Conventions

Different operating environments may have conventions for many aspects of file handling. There may be conventions for file naming, including characters allowed in file names, the length of file names, and the designation of a users "home" directory. There may be varying functionality for file-selection dialogs, filespace navigation, and other basic features of file management. Environments may also have conventions for the location of installed files and directories, such as a user’s Calendar Tool directory.

Implementors may follow required or appropriate conventions, but must implement the features described in these requirements insofar as is possible. In particular, the feature of the user being able to type a file name in the file−open dialog must be retained, even if it is not standard in a particular operating environment. Also, the user must be able to change the location of the Calendar Tool user directory, even if the operating environment recommends the placement of such directories in a particular location.

2.9. Details of Edit Commands

6dec04 Note: See the 205 Milestone 10 version of this file for a quick finish to things. The ideas here need to be merged into the M10 version, or vice versa.

Organizationally, this section will probably need to be broken up into several files.

The scenarios in this section cover full details of the general ediing commands that appear on the `Edit` menu. The `Edit` menu in the Calendar Tool Administration program is a proper subset of that in the regular-user Calendar Tool. The adminitrative version of the menu lacks only the `Find` command. Otherwise, the commands on the two menus provide the same functionality. The operational context differs between the regular-user versus adminitrative edit commands. These contextual differences are eplained as appropriate in the scenarios that follow.

of which have been introduced in preceding sections. There are two file command menus to be covered, one in the regular-user Calendar Tool and the other in the Calendar Tool Administration program. Regular-user file commands are covered in Sections 2.8.1 through 2.8.7. Administrator file commands are covered in 2.8.8 through 2.8.11. The concluding Sections 2.8.12 and 2.8.13, cover general topics relevant to both regular–user and administrative files.

2.9.1. Undo, Redo, Repeat

Can undo the last confirmed typing in any editable text field. A typed edit is confirmed when the user presses the TAB key or moves the cursor to another element in the display other than the one in which the edit took place.

Can also undo the most recent confirmation of

Only a single level of undo/redo is available.

2.9.2. Cut, Copy, Paste, Delete, and Select All

These edit commands operate strictly on text content within display windows. In particular, the commands do not apply to scheduled items as a whole, or to any other editable data elements as a whole.

2.9.3. Find

[Sketch: Among the more important items to define clerly in this section are the details of how Edit.Find works. We should check out the details of how it works in other systems.]

The system performs the search through all text fields in all scheduled item instances, and list displays. Does not

```
Search For: <reg expr>

Start Date:                         End Date:

X All Items     _ Visible Items Only


(Find/Next)  (Previous)  (Done)
```

Figure 311: Find dialog.

apply to any non-text fields in any windows item or list displays. Also does not apply to any other windows, in particular not in editing dialogs or any of the admin displays. *Enum precisely what fields it applies to; good way to do this may well be to say it applies to the display windows generated from the following commands*

2.9.4. Text Command Interface

Use the extant `protoj` convention about static methods named for the menus.

```
Execute commands:
    -- multi-line command interpreter --

Load command file:
    -- file chooser --
```

Figure 312: Command dialog.

3. Non-Functional Requirements

6dec04 Note: See the 205 Milestone 10 version of this file for a quick finish to things. The ideas here need to be merged into the M10 version, or vice versa.

Loose items to include where they go:

o Date stamp of what the user sees as the application program cannot change when the user changes the location of the Cal Tool user directory. Need to define a bit more precisely "what the user sees", but it means the thing with the name or icon of the application that the user sees when browsing the file space, which can be either a single file (e.g., `.exe`) or a directory tree (e.g., `.app`).

3.1. Security and Privacy

Cover the fact that user calendar files in the central host repository must be encrypted so there contents are not visible by any other means than through the calendar tool, this to prevent users on the central host computer, including administrative users, from seeing private calendar items. The implementors may not elect to exclude private items from the host copy of a calendar, since the requirements in Section 2.6.6.1 indicate that upon copying from local calendar to host, the copies of the calendar must be identical.

3.2. Performance

We may want to address certain key performance issues of interest to implementors, but not tell them how to do their thing. Here's a list (only one item in the list as of 25ul04):

a. Whether local-to-host copying is full or incremental is up to the implementors; the of the local-to-host copy operation is that the two calendars must be identical when the copy operation is complete, and some reasonable performance standards must be met, where we need to figure out what "reasonable" is.

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:

o system-related,
o process-related, and
o 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.

3.3. System

System-related non-functional requirements cover performance, operational environment, and general system characteristics.

3.3.1. 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.

3.3.2. Operational Environment

There are no specific requirements for the hardware or software platform on which the Rolodex Tool must operate. As discussed in Section , 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.

3.3.3. 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 loose 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.

3.4. 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.

3.5. 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:

- *o* Owners: the owners of the Biggie Burger restaurant for whom the system is being specified
- *o* Customers: the food-ordering restaurant customers who will use the food ordering subsystem.
- *o* Managers: the restaurant management staff, including the owners, who will use the inventory control subsystem.
- *o* Developers: the team or individual who will procure and/or develop the system to meet the specifications set forth in this document.
- *o* Analysts: the authors of this specification document.

3.6. System Performance Constraints

- *o* 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.
- *o* 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.
- *o* 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.

3.7. Qualitative system characteristics

- *o* 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.
- *o* Simplicity is also a desired characteristic for the inventory control subsystem.

3.8. 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. Rationale

Here's an idea for organizing this section -- have subsections for each 2.X-level section, as well as others that may have gone away and don't fit in the current 2.X structuring. As far as the text goes, maybe we can leave it pretty raw and informal, treating it like a (b)log of thoughts along the way. I think this might work, and lead to getting the fucker done before death.

4.1. Clean Up (I Believe) of Section on Recurring Items and Instances

The following bit of doodoo was at the top of "Recurring Items and Instances" section, and it's now been addressed:

> ***NOTE:*** This needs to be updated to reflect the new idea of *effectively-scheduled* instances. As described here, it suggests the idea of *physically-scheduled* instances, which is not what we want. Here's a scketch:
>
> *o* a recurring item has a changed-instance list and a deleted-instance list
>
> *o* the elements of these lists are instance dates
>
> *o* in the case of a changed instance, there is another item added to the calendar, with the changes in it
>
> *o* the changed-instance items are still considered part of the original recurring item, for the purposes of an all or all-future operations, where these operations are invoked from either a changed instance or unchanged instance of the recurring item
>
> *o* when a change to start date, end date, and/or recurring details is made, all effectively-deleted changed-instance items are removed from the calendar, and changed-instance and deleted-instance dates are removed from the changed and deleted lists; in this way, the lists only contain instance dates that are within the start/end date ranges and are covered by the recurring settings

Now, and mostly then, English description does not really imply *physically-scheduled* instances, but rather is suitably non-committal to any particular implementation. And at the same time, I believe it is adequately precise for the requirements.

Now, if we want to go with a model that uses the idea of changed-instance and a deleted-instance lists, then that may be OK. The deal is this, we should not impose too specific a limitation on the implementors when it comes to how many instances are actually instantiated at any given time (actually, this may be wrong, depending on how things work out regarding the user's need to know the *size* of a calendar, measured in the number of calendar items). What we need to do is put some reasonable limits on the sizes of the memory footprint and saved-file size of a calendar, so that neither of these goes unbounded due to physically allocating too many instances of recurring items. Potentially tricky, but interesting business, this, presumably some of which ends up in the non-functionals about size.

4.2. For Next Check-In

o The following point got axed (hopefully permanently, when we decided not to have ID be part of a calendar anywhere, but to get the ID when necessary from the connection tab, when the current calendar is connected to a host:

> When user X opens a cal file copied from user Y, including one created by an admin in a distribution, user X becomes the user of record for that calendar at the point of opening, i.e., the opening user's ID is made part of the opened calendar and will be saved as such when and if the user does a subsequent File->Save.

o `SaveAs` and `Load` are gone from all aux data dialogs they're currently in.

o There's a `Save` (but no load) in the host connection table dialog; the rationale for this being here but not in the other aux data dialogs is that host connections are tool-wide as opposed to cal-specific, and hence not saved with the regular File->Save; the `Save` in the host connection dialog is a convenience, since it can be accomplished within the '`File->Save Settings`' dialog.

4.3. Bits from and Related to File, During Push to Finish

A big pilo crap follows, spewed originally in ".sh 4 "Local Files"". It's indented to distinguish this particularly largish crap pile for the rest. (Indenting crap piles seems to be an evolving convention). Each paragraph is also annotated with where it's been dealt with.

Say that the implementors can figure out how/where exactly to define this (the location of the user dir), including one or more of these ways: ... . The effect must be that when the cal tool is subsequently invoked, it looks for the Settings file and non-path-prefixed calendar files in the specified directory. *This is pretty fucked up, but it's been dealt with in the development overview subsection on implementation considerations.*

Hmm hmm, I think having one-by-one naming for each file is probably (way) overkill, cfing Emacs single .emacs file; also, wait a mother-fucking minute -- we may have a huge motherfucking chicken and egg problem with the whole local files thing altogether, since where the fuck does this setting get saved; i.e., it would seem there needs to be at least one *known* place to look for things, unless we want to modify the app itself, which sounds fucked up; what I think we're getting at is the the local files determination needs to be made at installation/configuration time, not dynamically. *Dealt with by having only one Settings file read at start up; chicken and egg problem dealt with in developer_overview.implementation_considerations.*

Hmm, maybe we want to let the user specify one-by-one where each of the different files goes, rather than having the idea of a specific known directory. This is probably the more flexible way to do it, if it works OK. One fucking way or another, we need to fully figure out what known places the Calendar Tool needs to be aware of and what the full ramifications of knowing these places are. *Dealt with by having one Settings file and, again, in implementation considerations subsection.*

Local Files dialog now has two items: path for local file directory and name of default calendar file to open when the app is launched as an app as opposed to from a file. If the user specifies a value for the default start-up calendar and then invokes the cal tool from a file, the file from which the tool is invoked overrides the default start-up calendar. *Dealt with in "Files Used by the Calendar Tool" subsection and future work item about command-line arg to specify start-up settings file. The idea of a single default calendar to open has been extended to the "open calendars" part of the session-wide settings.*

24jun04 NOTE: Since we've nixed the "CommunityCalendar" idea, what we can say about calendar files and the local files directory is that the directory so specified is the default place where calendars will go. Given the current set up of things, all this really means is that that dir is the default path when calendar file chooser is initially opened. Related to this, we need to figure out if the file-chooser dialogs have temporal memory, as in they come up in the place that they were most recently OK'd (but not Canceled) on. I think the answer to this is almost certainly yes. *Dealt with, as aluded to, in the "new setup" of things. Temporal memory in dialogs dealt with in specific subsection in data entry details.*

NOT to all of the following paragraph; coverage of '`File->Local Files`' since it's been nuked as a command and moved to Admin options; the priv admin '`File->Host Files`' has been moved to the priv `Admin` menu. So, what this section covers is what's in the files (a review, most likely), but not these two commands, since they're not on a file menu anymore. *Dealt with as the paragraph itself explains.*

Need to cover '`File->Local Files`' command here or elsewhere, as well as the privileged admin '`File->Host Files`' command. With '`File->Local Files`' regular users should be able to set the path to where the local files are. When creating a config, admins can set the default value for the local files. *Dealt with fine in the new scheme of things; in particular, admins don't do configs per se anymore, but can set location of user dir by creating a settings file using the regular calendar tool.*

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. "`Intialization`": the command file read in by the CalendarTool whenever it starts up
e. "`FileInfo`": file containing settings established with the '`File->Connect`' and '`File->Local Files`' commands.

Dealt with by consoldating everything into cal-specific and session-wide settings, in cal files and settings file, resp.

Given the nixing 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. *Dealt with as the paragraph itself explains; also, the matter of deleted or corrupeted files in general is dealt with in File subsection on this topic, as well as Error Conditions section as appropriate.*

The next NOTE was correct -- all pretty much obsolete bullshit. Went through it and found a couple thigs that helped firm things up in the final version of the File section, so put a motherfucking fork in it.

NOTE: I'm pretty sure that all of the subsubsections that follow are bullshit, but may have some useful info. Check them out one last time to be sure.

.sh 4 "Save As and Load in Editing Dialogs"

9jul94 ADDITIONAL NOTE: OK, I'm just about ready to fuck the SaveAs/Load shit in all of dialogs where they appear. This will nicely simplify things, and users who give a fuck can --- OK, fuck me, we're going around in circles again. If we lose the SaveAs/Load shit in the dialogs, then there's no particularly convenient way to get, say the options, unbundled from a calendar file. One way might be to provide a "Clear all items" command, say in edit, that could be used to get rid of everything but the aux data, and then a new calendar could get another calendar's aux settings by copying the first calendar into the new one and then nuking all the items in the new one. This really anin't too bad, given the complication it avoids and the fact that we can still get done what we'd like to get done. To whit:

o ...

OK, fuck me hard, one more time, way up the motherfucking ass. Let's just stick with what the fuck we have, including the fucking-ass complicated new row of buttons in the connections table, explain it clearly and get the fuck on with it.

No, but now fuck me again. Probably the bottommost line here is how useful the aux data file saveas/load bullshit is, and I think the answer is honestly not very fucking much useful. For me, I'm likely to have a few calendars at most, and not be spending my motherfucking life building caltool aux data files. So, I think except probably for Options, which are a big pile of do-do, we'll nix SaveAs/Load. If I like the categories, lists, or filters so fucking much in one calendar file, then I can copy the file, do a "clear all items" and get the fuck on with it.

Or one more motherfucking possibility is to have a 'Save Settings' command in the file menu that can be used to save all of the non-scheduled-item data in one motherfucking Settings file (maybe even including options). We'll also need a File->LoadSettings to go along with this. This way I can save all of the motherfucking settings in one place and load them up if I give a fuck about them for some other calendar. In this scheme, we can fuck the explicit `Options` file, having a configured distribution of the tool with the options settings in the tool itself, and provide a 'Restore Defaults' button in the options dialog, instead of SaveAs and Load. ALSO (IMPORTANT) -- nuke File->SaveConfig in favor of a new Options.Viewing.Misc setting that allows 'From last time' (or something like it) for the 'Initial View' option (in Figure 283).

9jul04 NOTE: (1) Itemize the dialogs in which these commands appear; (2) explain why only the host-connection dialog has '`Save`' in addition to '`Save As`', whereas all the others have only '`Save As`', the explanation being that the host-connection dialog is the only one that is editing non-calendar-specific data that is stored someplace other than the current calendar).

NOTE: Adjust the following fodder pulled from the options section to apply uniformly to all four types of aux data (which will have been clearly defined in the intro of this section). 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 .

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 313. 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 . Section . 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 314 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 coy 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 315. The names of valid options files appear in normal typeface in the directory list; all other types of files appear in greyed typeface. Section 2.8.2 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 316 shows the user having selected to load from the "`Personal Options`" file. Figure 317 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 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.)

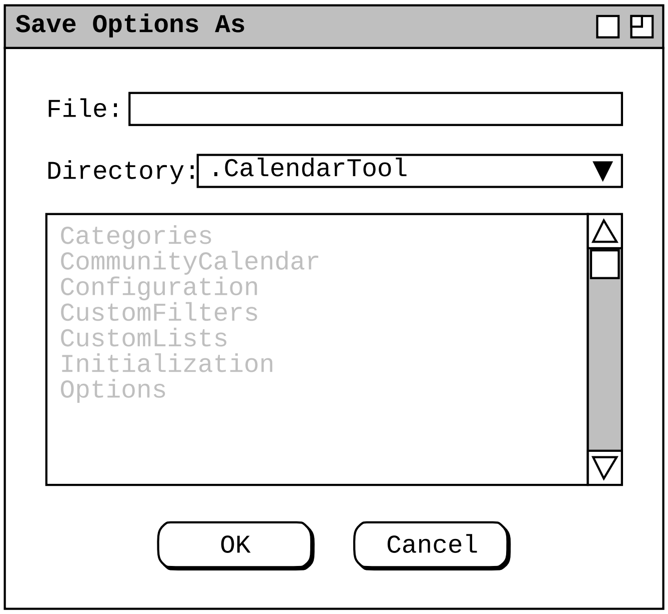
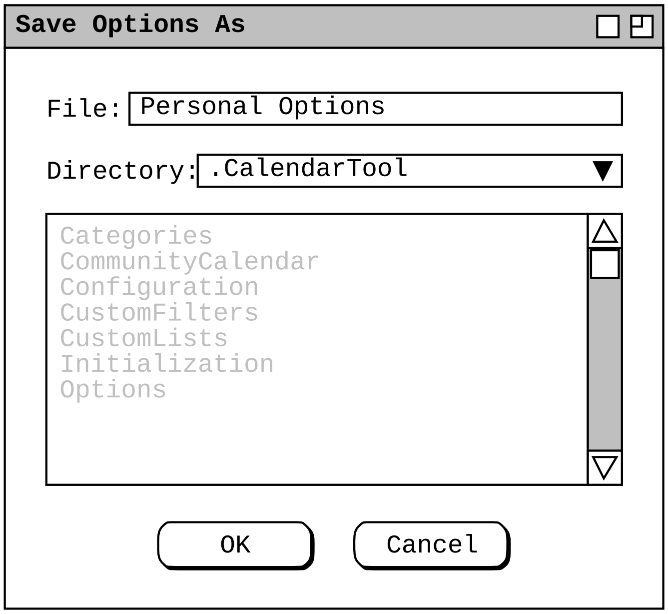


Figure 313: Options save-as dialog.

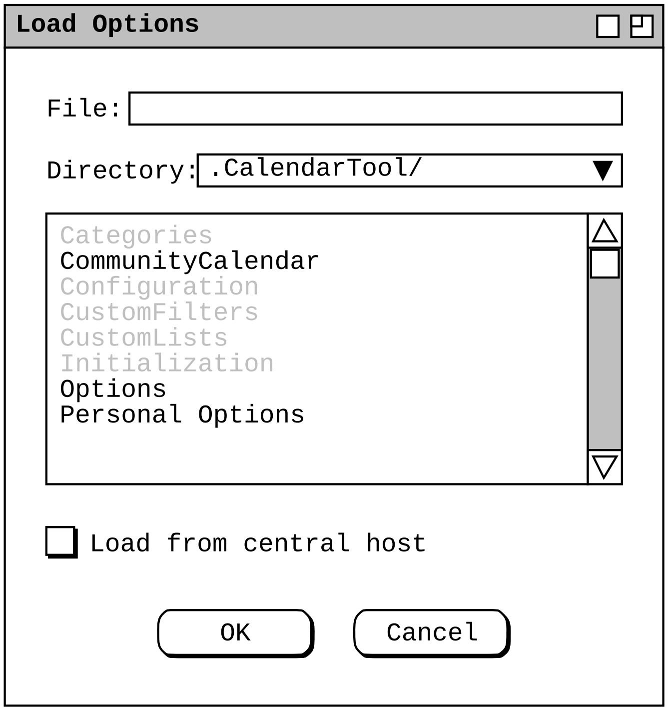

Save Options As

File: Personal Options

Directory: .CalendarTool

Categories
CommunityCalendar
Configuration
CustomFilters
CustomLists
Initialization
Options

Figure 314: Saving options to a PersonalOptions file.


Load Options

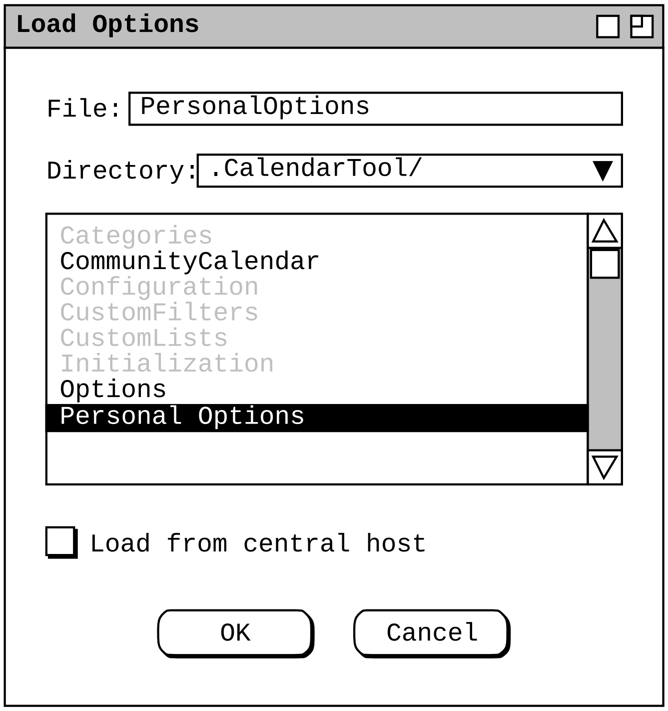
File:

Directory: .CalendarTool/

Categories
CommunityCalendar
Configuration
CustomFilters
CustomLists
Initialization
Options
Personal Options

☐ Load from central host

Figure 315: Options load dialog.



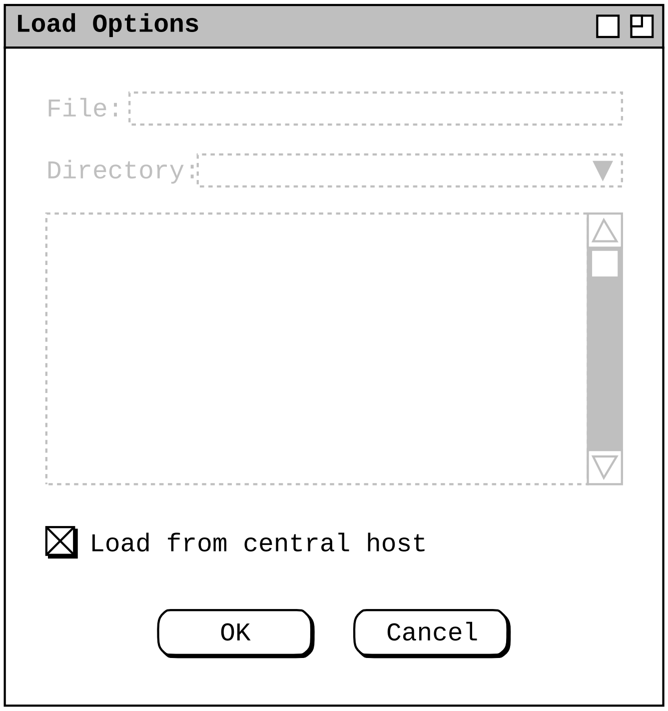
File: PersonalOptions

Directory: .CalendarTool/

Categories
CommunityCalendar
Configuration
CustomFilters
CustomLists
Initialization
Options
Personal Options

☐ Load from central host

Figure 316: Loading from the Personal Options file.



File:

Directory:

☒ Load from central host

Figure 317: Loading options from the Calendar Tool central host.

If there are unapplied or unsaved option changes when the user presses '`Load ...`', the system issues the warning shown in Figure 318, 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 for an overview. So if we allow load from cal file but not save-as to cal file, here's how we accomplish save-into: open the cal 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). 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 Section 2.8.2 and

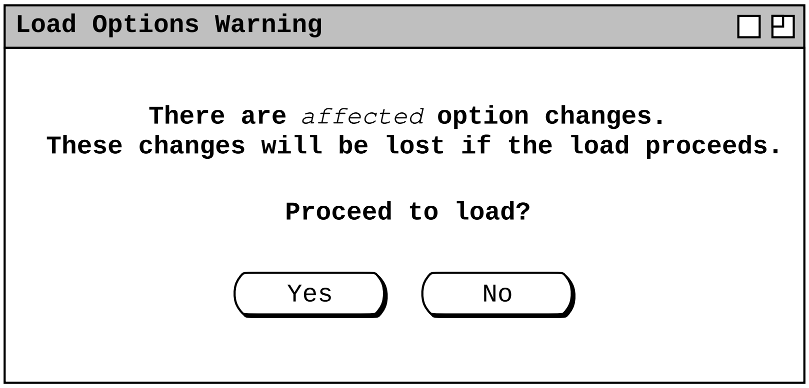
describe how file dialogs display inapplicable file types and the details of overwriting extant files.

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 2.8.7.

2jul04 NOTE: Be sure to cover the (possibly subtle) distinction between option saving from within the cal tool admin program versus the regular-user cal tool. Viz.,

a. When an admin does regular File->Save, user-related option values always go in the `Options` file, since



Load Options Warning

There are *affected* option changes.
These changes will be lost if the load proceeds.

Proceed to load?

Figure 318: .

there is no calendar in/with which to store options calendar-specifically; I suppose we could say that options are "distribution-specific", but it seems this would not work well at all, since there's no distributional analog of the "current calendar".

b. Otherwise, the file-related option dialog buttons should behave the same for the admin as for the regular user. The usage semantics may differ a bit, and be worthy of explaining, as in the reason the admin does an options-dialog ‘`Save As` is to create (conveniently) a reusable options "bundle", that may be useful generically for different custom distributions. As with the regular-user version of the options dialog, ‘`Save As`’ is no big deal, since it can be accomplished pretty easily at the OS level via file copy. However, it's worth having, and for fuck's sake, even if it isn't, it's not at this point going to get nuked from the multitude of option screen shots we have.

And here's a blast form the not-so-distant past about the now-gone ‘`File->Local Files`’ command.

.sh 4 "File Connect and Local Files"

The ‘`Connect`’ command was described in Section 2.6.6.1 on administrative commands. When the user selects the ‘`Local Files`’ command, the system displays the dialog shown in Figure 319.

The following has been dealt with fully, including nixing the idea of being able to overwrite non-calendar files:

.sh 4 "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.

Done and expanded upon:

.sh 4 "Calendar File Contents"

a. scheduled items -- initially empty
b. categories -- initially holidays
c. custom lists -- initially empty
d. filter settings and custom filters -- all shown, custom filters empty
e. windowing mode -- initially per-level
f. options -- initially obtained from default options file (`Options` by default) (hmm, there's potentially a chicken-and-egg problem lurking here; figure it the f out)
g. windows

Here's a slew of seriously funked up bullshit about saving and loading settings (hold on a minute, we're still seeing if it's really seriously funked up): ... OK, I've held on and checked things out. Some of it isn't so funked up after

Stick in the ‘`Default Calendar Tool Directory`’
item from the Admin tab of the Options dialog.

Also add field for entering default calendar to open on fileless app launch.

Figure 319: Local files dialog.

all, but things have been fixed in the latest version of the file section, and here's the stuff to which I refer:

> *12jul04 -- OK, we need to add to the '`Load Settings`' dialog the option to load them from the current host, which replaces the short-lived but seemingly uninspired '`Options->Load from Host`' command.*
>
> It's more than just the window configuration since it includes what files are open. I think the easiest way to define exactly what save configuration 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 configuration that is absolute datewise, or 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 a la 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 line 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.

Another 4aug03 NOTE. OK, I think I want to take back what I say in the next paragraph. I think it's appropriate to require that each user calendar be stored in a separate file in a known (settable) dir, with some reasonable convention for file name, such as the user's ID plus ".cal" extension.

4aug03 NOTE (if not mentioned elsewhere): Implementors can choose how they implement the central host repository, e.g., having a separate cal file for each user or having all user cals in the same file, or some other implementation. There is no requirement that a user be able to view the contents of a calendar in the central repository as a raw file in any form. Operationally, the user has her own local calendar as a file with a known name. On the remote host, the publicly-visible *contents of* that calendar are made available to other users through the calendar tool, via the operational effects of connecting and file saving described herein. Again, however, there is no requirement for the user to see a specific calendar file on the central host.

Nixed the auto update of host tab in the next bit, but did address the issue of warning on save-as when current cal is connected:

> '`Save As`' needs to issue a warning when the current calendar is associated with a connected host. The warning should allow the user to stay connected, with the host connection table automatically updated.

Decided against the following, since user can move/remove Settings file if she doesn't want it used on File->New, or load a cleared version of settings at any time:

> The '`Load Settings`' dialog also has a radio buttons for '`load settings on File->New`' versus '`Use installed default settings on File->New`.

I was thinking about a adding an admin option for '`Auto-save tool-wide settings on exit`' but decided it wasn't necessary because (a) there are only two such settings (open cals and connect tab), (b) open cals ain't that important, (c) there's already a convenience Save button in connect tab dialog. Also, I don't think it's necessary to warn the user on exit if the only unsaved tool-wide setting is current open cals, since that's not really a big deal given all the other ways that the user can save this stuff, and it's certainly not critical to operation if it's missed when the user wanted it not to be. The emacs hack would say to add the yes/no option warn-on-exit-if-open-calendar-names-setting-not-saved.

Most likely nix the following bulleted list.

File-related commands allow the user to perform these functions:

o create new calendars
o open existing calendar files
o close one or all open calendars
o save calendars
o save the configuration of the Calendar Tool
o set the default file directory where Calendar Tool files are stored
o print calendars
o exit the Calendar Tool

These commands are described in the scenarios that follow. Before the scenarios, a general description is given of the different types of files used by the Calendar Tool, and the types of data stored in the files.

Print crap:

> The printing functionality in the initial version of the Calendar Tool is limited ... . *Make it just screen shots.*

OLD: *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.

NOT SIMPLE ENOUGH: *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.*

Exit crap:

At this point, it looks like, oh please say it's so, that Exit can do exactly what Close All does in terms of offering to save. Hope the fuck so. Well, at least one additional thing Exit needs to check is if there have been changes to either of the session-wide settings since settings were last saved (including the save in the global options and connection table dialogs), i.e., open calendars and connection table.

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.

If the user exits or closes a calendar with notifications that have not yet been responded to, the system displays a dialog of the form:

```
There are meeting notifications you have not yet processed.  Do you want to
review or discard the unprocessed notifications?

                 Review    Discard
```

If the user chooses 'Review' and then grows tired of it before all notifications are processed (where processing means pressing one of the three buttons on the bottom, or explicitly closing the notification dialog (which is equivalent to Delay)), then she can select Exit of Close again (whichever of which started the whole thing, or either) and then hit Discard. (Or use ... (fell the fuck off, but who gives a fuck at this point.))

4.4. Bits from Options that May Be Useful, or May Not

The change in location of the repository directory does not affect any of the contents of the repository files or subdirectories. All of these files and subdirectories are moved in tact from their previous location to the new location.

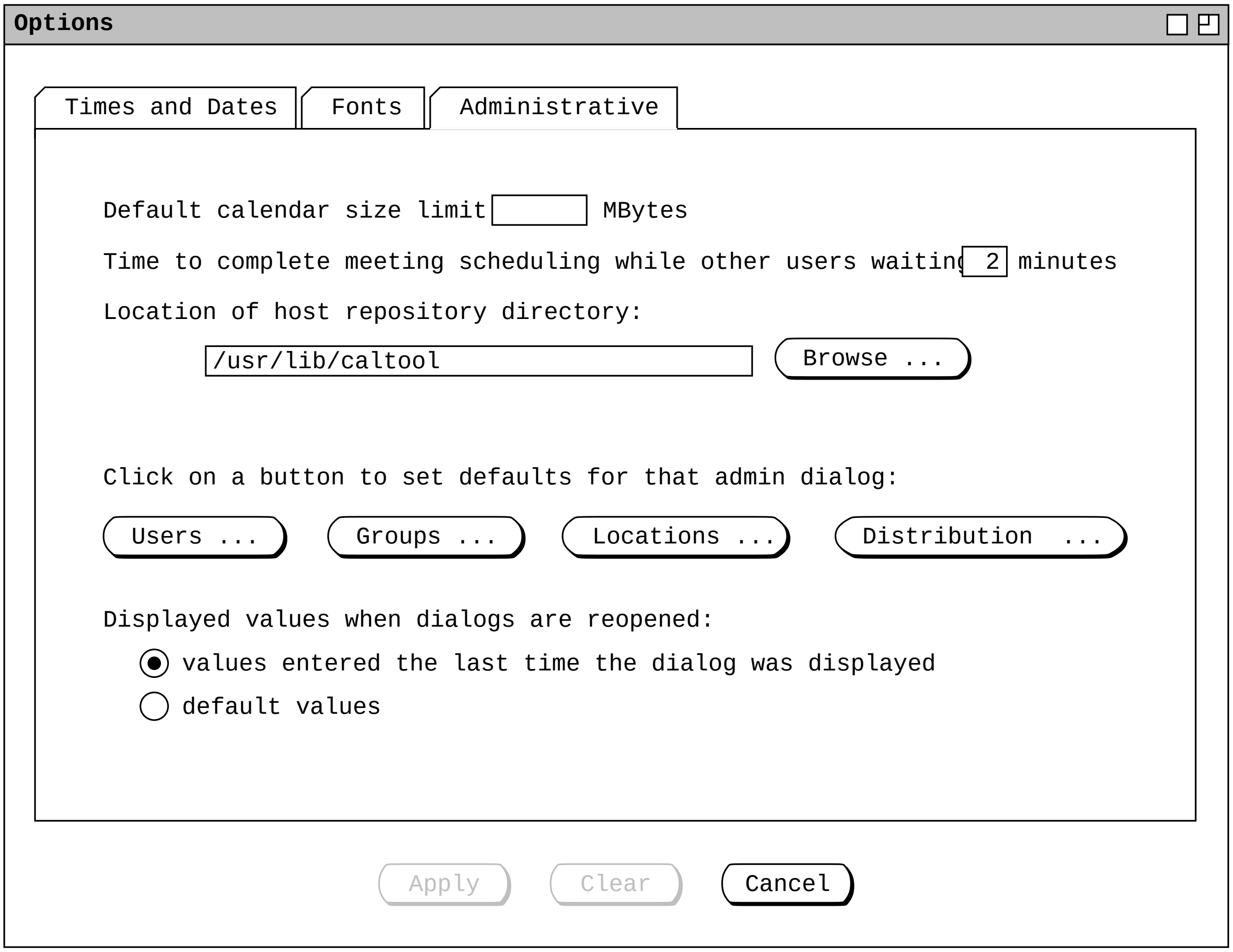
When the administrator applies a change to the repository location by pressing '`OK`' *(should be apply)*, the system verifies that the specified file path is valid and readable and writable by the Calendar Tool Administration program. If it is not, the system reports an error as described in Section . To avoid operational disruption, the '`Host Files`' command is disabled when the server is running.

NOTE 12jul04 -- OK, this section [admin options] is pretty well messed up. The deal is that the long-held idea that the admin would define user options by using a mock version of the user prefs is outta here. Instead, we'll have admin-relevant options tabs for '`Times and Dates`' (which turn out in fact to be relevant in an number of admin dialogs), '`Fonts`' (which are pretty much always relevant options in any app that does not want to be a pain in the neck by not letting users change display fonts), and '`Administrative`' (which presumably will stay the same as the current '`Administrator`' subtab, with whatever recent mods are necessary, at least one mod being the likely addition of the a place to set the host dir, to parallel the new way things are in the regular user UI, where '`File->Local Files`' is now gone). And the way the admin will define user options is to invoke the regular Calendar Tool, and do the normal options and save-settings things there. Even though this may not be quite as convenient as the old way (and this is debatable), it's way fucking less convoluted than having what was turning into a major pain in the neck of having two ways for the admin to set user options. One of the key discoveries that tore it was trying to figure out where the fuck the options-containing settings file would go when the admin did a '`File->Save`' in the admin program. We probably could've figured something out, but it was definitely getting fucked up to have the two ways of admins setting user options. So, get to it and fix this.

For an administrative user running the Calendar Tool Administration program, the '`Options`' command menu and top-level tabs of the options dialog are the same as for the regular Calendar Tool user. Physically, the only difference between the administrative version of the options versus the regular-user version is that the '`Administrative`' tab has two subtabs, as shown in Figure 320. The '`User`' subtab has exactly the same format as the full '`Administrative`' tab for the regular user, shown in Figure 282. Details of the '`Administrator`' subtab shown in Figure 287 are explained shortly.

Functionally, the administrator version of the options dialog differs fundamentally from the regular-user version. In the administrator version, none of the option settings, except those shown in Figure Figure 287 applies to the Calendar Tool Administration Program itself. Rather, when an administrator makes changes to options values, the changes are recorded for use in creating customized Calendar Tool distributions, as described in Section . The point is, none of the option settings for regular-user commands apply to the Calendar Tool Administration program, since the administration program does not perform calendar scheduling or viewing. Hence, except for the one administrator–specific subtab, administrator–supplied option settings apply to custom distributions of the Calendar Tool, not to the administration program itself. In a given custom distribution, these options serve as the installed default settings until changed by the regular user.

In keeping with the functionality just described, the '`Apply`' button in the administrator version of the options dialog is only enabled when the administrator has made a change within the '`Administrator`' subtab. Otherwise, the option dialog buttons in the administrator version function the same as in the regular-user version of the options dialog, as described in the preceding scenarios.



Default calendar size limit ☐ MBytes

Time to complete meeting scheduling while other users waitin 2 minutes

Location of host repository directory:

/usr/lib/caltool

Click on a button to set defaults for that admin dialog:

Displayed values when dialogs are reopened:

● values entered the last time the dialog was displayed

○ default values

Figure 320: Administrator version of the Administrative options Tab _.

The ‘`Administrator`’ subtab in Figure 287 applies specifically and exclusively to the Calendar Tool Administration program.

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.*

Here’s a potentially interesting bit of going around in circles regarding whether to have options to allow the user to change ID and password:

> *Add options for default caltool user ID and password. These are the ones tried first for connect. NOT -- IDs and passwds are now in connect dialog. NOT NOT -- they’re not in the connect dialog, so we do want options here for default ID and passwd. NOT NOT NOT -- the user doesn’t get to change his ID himself, and the password is chaged using Admin->Change Password, q.v.*

And for an example of a brain-damaged option (at least now given that the only way to contact an admin is via the Admin menu):

> The option to ’`Hide Admin menu when not connected`’ controls when the ‘`Admin`’ menu appears ... .

.sh 5 "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.

Well farg me, the hoped-for "Final Sketch" in the next paragraph wont fucking work -- take a mf look at options-admin-admin.idr. So, do this:

o In the Admin version of the options dialog, have everything be exactly the same as the user, except for the Administrative tab, which will have two subtabs -- User and Administrator

o The User subtab is identical with the Admin tab in the user options dialog, and like all of the other options tabs is used by the administrator to set options to be stored in a distribution.

o The Administrator subtab is what's currently in options-admin-admin.idr, cleaned up as appropriate; in particular, we need to note that only the top tier of the three DB dialogs can have options set (e.g., not any of the sub-dialogs like the ones for location bookings); also, change the Central Host button to a Distribution button, since the latter dialog is now one with enough stuff in it to make default settings useful (albeit probably only minimally so).

o No options sharing, since the only one that was shared was email app, and now we'll clarify that caltool admin doesn't launch an email app.

o Much if not all of the crap below about global options and "inheritance" is obsolete; here's the way I think it should now work:

-- All of the user options set by an admin are made available to users via a created distribution, and in particular via the Options file that is packaged with that distribution.

-- There is no 'Defaults' button in the user options dialog; if the user wants to change the distribution-supplied options for possible later reversion thereto, he must do the following:

- save the original `Options` file that comes with the admin-supplied distribution, either with the '`Save As ...`' options dialog command, or via a regular OS file save
- use the '`Load ...`' options dialog command to re-load the saved original `Options` file after any options changes have been made and saved (with an intervening, or move/copy the original saved file back into the file named "`Options`".

Final Sketch:

o Have options be *exactly* the same as for regular users.

o Option values set by the admin are used as the config defaults, as described in the Distribution subsection.

o The only option setting that applies to the admin program as well as to the config settings is the default email program, which in the case of the admin program itself applies to the email program that is used to send admin messages to the user, as described in the Admin section.

• As far as dialog defaults go for the Admin program, the only behavior is that all dialogs are reopened with the values entered the last time the dialog was displayed. For the program, there no other way to set specific default values for any of the admin edit dialogs. When an admin dialog is initially displayed or cleared, the initial values are as shown in Table 19.

NOTE: There needs to be a place for privileged admin-specific options, in particular these:

- default size limit for user calendars; int text field with empty default, meaning no limit
- defaults for db dialogs, using the same style as the defaults for scheduling dialogs

whatever

Table 19: Default values for all admin-dialog data fields.

- Time to Complete Meeting Scheduling While Others Are Waiting: int minutes (default 2)
- possibly other stuff in the privileged admin menu

OK, it looks like the entire admin tab needs to be expanded for versions of the options dialog that is displayed from within the caltool admin program. It almost certainly needs subtabs, more-or-less correspondent to the items on the privileged admin menu.

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.

Here's some fodder axed on 11aug03 from the Admin section:

.sh 3 "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.

.sh 4 "Administrative Options"

.sh 5 "Root System Administrator"

.sh 5 "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 CANCELED for recurring (maybe, or probably better just to nuke it or maybe make this an option).

c. Can add and remove members for group.

4.5. Pretty Close to the Last Pile 'o Due Due from Admin

4aug03 Update to the 17jul03 update: We've now got it all figured out, with the new setup for the user record with ID and password, and the way host connection now happens.

17jul03 Update: I don't think the following issue has been adequately addressed yet. To whit, there needs to be some way for a user on a local machine to be validated as a legitimate (registered) user on a Calendar Tool central host. The, I think, so-far unspecified way I've had in mind for this to happen is that the system performs the validation by matching the value specified in the '`Computer User ID`' field of the Calendar Tool user record with the value of the value of the '`Calendar Tool ID`'. The problem with this is that if the user is coming from some kind of local computer without a user ID, or that's not passworded, or has some funky convention for user ID names, then this might not work. There's also the issue of security when accessing the central host. With the local-id-matches-caltool-id scheme, there's only the security access at the local machine, which may be considered weak.

In thinking about this, I think there's also a problem with the whole idea of listing the IDs of the local host computers in the caltool user record, particular in the case of access from a machine with a non-static IP address. As I recall, the reason for this is to allow the central host to send notifications to particular users on *particular* machines, whether or not the cal tool is running on those machines. The idea that this is nice stems, it seems, from Claris' ability to pop up a reminder even if Organizer is not running. However, I think there's a significant difference between this and the ability of a central host to send a meeting notification to some other machine altogether. In the Claris reminder case, it can be implemented with an entirely local "at" daemon of some kind, or the Windoze equivalent. In case of the remote meeting notification, the communication has to go across the wire to some known network address via a PUSH.

So, first off, I now have a significant question about whether the "any time" option is really viable for meeting (and other) central-host-generated options. In order for this to work, either the central host needs to have IP addresses to send to, with a daemon running on the other end, or when a local machine starts up, or when a particular local user logs in, the local machine has to fire up a daemon the communicates with a central host, including performing the user validation that we mentioned above, in order to be able to receive notifications without the cal tool running. On a multi-user machine, this could be a huge problem. Given this, I'm thinking that it's reasonable to say that notifications can only happen when the cal tool is running and when a user is explicitly connected to one or more hosts. So, in order to check if a meeting has been scheduled, one must explicitly connect to the host from which the meeting originates.

So, I'm thinking we need to have a new, and if we're lucky, simpler scheme where the user must provide an ID and password when connecting to any particular central host.

4aug03: And so it is.

4aug03 reply to 26jul03 consideration in the next paragraph: NOT. We require connect with a particular calendar, period. If this is inconvenient for people who want to "just browse", tough shit. Bottom line -- a calendar is a necessary part of the entry ticket to a central host, along with, of course, a registered ID.

26jul03: consider allowing central host connection without a calendar, which means user can look around at other people's stuff but not receive any updates. Hence, all of the places where we say that the user must be connected, we mean "must be connected with a particular calendar".

When the user executes the ‘`Connect`’ command in the ‘`File`’ menu, the system displays the dialog 321.

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. A calendar can be associated with two or more hosts, but at most one of them can be connected at any given time. If the user tries to connect to a host with an associated calendar that is currently connected to another host, the system displays the dialog in Figure XXX.

Figure 322 shows the user ... . When the user presses connect, the system ... .

If the user has specified a default user ID and password in the options (see Admin Options section), then those are the ones sent to all hosts, i.e., they show up as the default type-ins in the connect confirmation dialog. This means it’s nice etiquette in a particular multi-host environment to use the same user IDs on hosts where it’s reasonable to expect that the same user will be connecting.

The specific rules for calendar/host association are the following:

a. a calendar may be associated with zero or more hosts, at most one of which can be connected at any given time;
b. each host in the connection table must be associated with exactly one calendar (which per the preceding rule may a calendar that is associated with other hosts)

(We need to define precisely the terms *associated* and *connected*.)

Here’s exactly what happens when the user connects:

a. local cal copied to and replaces host cal, *except for* pending meetings scheduled since last connection (call these "new pendings")
b. new pendings are added to local cal (and remain in central host cal)
c. all notifications queued on host are processed, per Section . I.e., if user accepts notifications (whenever connected or at initial connection), then the notifications are all displayed.

To add a new host, press ‘`Add`’ to which the system responds by bringing up and 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. [The following in-line-editing of change doesn’t work because 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.

OLDER: 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

NOTE: Offer-to-save happens on exit if unsaved edits.

```
OLD:
      Central Host Computer ID: host name, with admin-settable default

                        Status: connected/disconnected


               Connect    Disconnect    Clear    Cancel
```

Figure 321: shit.

verification is successful, the system establishes communication with the chosen host and sends the user's community calendar file there.

The 'Central Host' dialog looks like Figure 322. -- **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.

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.

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 that describes the files and say that a normal file chooser 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 to some typical default location (platform-specifically) at initial installation.*

4.6. Nixed in Favor of Explicit Dialog Settings

But, alas, the next paragraph is full of shit, since we have now in fact put (most of) this stuff explicitly in dialog(s).

During process of creating a distribution, consider allowing the admin to set up a pre-defined connection table and pre-defined local-files value. I'd say the best, and most orthogonal way to handle this is to have the admin define an Initialization file that's included with the distribution. This is in fact a good idea in general, viz., a distribution is defined a copy of the app itself, plus a set of standard files. I think this provides a good amount of flexibility and makes good use of the standard files setup, i.e., it's in keeping with the philosophy that the standard files fully define a cal tool config, without any hidden values buried inside the app itself or in some obscure file (the latter being Microsoftesqe, it would seem).

4.7. What's on the Host

8aug03. It's not just the "public portions" of a user calendar, whatever those might have been, but all of the user's calendar. The viewing commands take care of not showing to other users calendar items or parts thereof that aren't supposed to be visible. An added benefit, though perhaps a weak one, is that the user can consider the central host copy of the calendar to be a full backup.

4.8. More nukation from admin, here re. purging and capping

NOT, it's part of the File->-- *THIS NEEDS TO BE ADDED TO THE COMMAND MENU.* --

```
Host Computer ID: (read only)    Status: (up or down)
```

Figure 322: Central host dialog for a regular user.

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:

- select user by ID or with radio button '`all`'
- 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 discretion. 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.

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.

4.9. Nuked from Change/Delete Section 28 Jul 03

NOT: 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.

... This item-level view contains exactly the same data values as Figure 167.

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 two lines of the text have been changed by the scheduler.

4.10. Nuked from Admin Section 27 Jul 03 through 11 Aug 03

Some more old crap:

.sh 4 "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. Can*not* 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.

Next several paragraphs are older variants of '`Contact Admin`'.

Other notes about '`Contact Admin`':

- Dialog simply displays name of current host (which is host to which current calendar is connected (with msg "Current calendar is not connected to a central host")) along with email address of admin of current host (with msg "no admin email is available for this host" if there is no email address.
- Caltool uses same technology as web browser to launch user's selected e mail program when the user presses

’Send Message’ in the dialog.

Here’s an older version of a user-to-admin message:

A user can look up the electronic mail address of any administrator using the ‘List Admins’ and ‘Users’ commands in the ‘Admin’ menu. Figure 323 shows a typical mail message sent by a user requesting an administrator to make some changes to the user’s Calendar Tool user information.

Some old crap:

> A regular user cannot change any of the values in a user database information record [in the DB dialogs; password can be changed indirectly using the password command.] 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.

OK, a big bunch of stuff has been nuked from the admin section, including a lot blather on the design of the central host. If you’re interested, check out admin.me version 1.23, which is a check point with all of the blather before it was nuked. But fuck it, I’ll put the juicier bits of fodder here so I don’t have to go dredge it out of the CVS log. To whit:

> *So fuck it again, I didn’t feel like putting anything here. Go see v 1.23 if you care (I already have a couple times). Diffing v 1.23 and 1.24 is probably a pretty good way to focus on the juicy bits I was thinking about putting here.*

IWe need to remove the ID and Passwd columns, which were recently added, since they should not be stored locally, but rather on the host. So, if we want to change the host password, we need to be connected, which means that leaving the `Password` command in the regular-user `Admin` menu is correct.

```
Central Host           ID  Passwd        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
```

```
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 directory: c:\home\calendar
```

Thanks.

Figure 323: User requesting calendar tool information changes.

`Connect     Disconnect     Change Password`

`Add Host     Change     Delete     Save     Clear     Cancel`

.sh 3 "User Communication with Administrators"

NOT. 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 adminstrator via the Calendar Tool. To do so, the user must look up the adminstrators' 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.

.sh 4 "Listing Administrators"

Describe that '`List Admins`' shows who all the admins are, Figure 324.

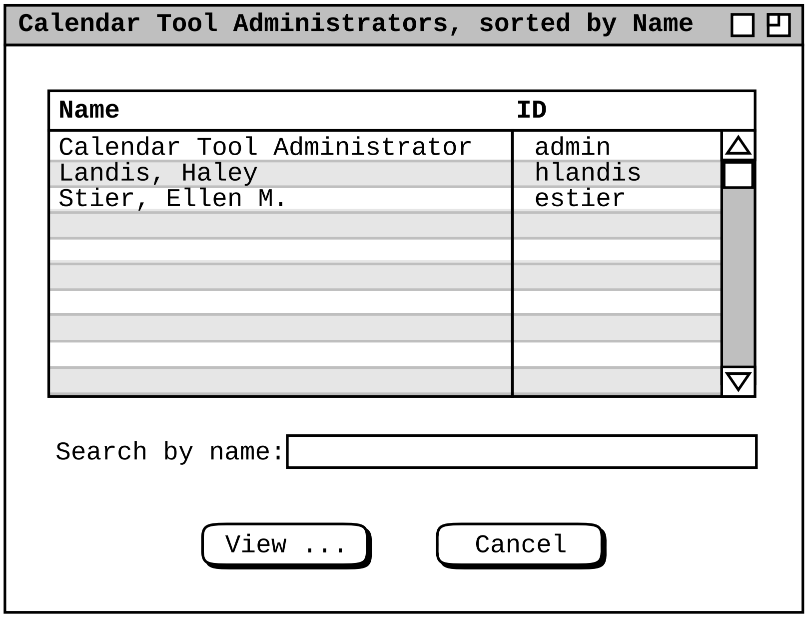
The '`Validate`' button is used to validate user information that is external to Calendar Tool administrative control. This is the information entered in the '`Email`' through '`Computer User ID`' fields. Details of the validation are covered in Section

[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.

In the scenarios to follow, privileged administrator access is covered in Sections 2.6.1 through Section . Access to administrative commands by regular users is covered in Section 2.6.6. In the scenarios, the term "administrator"



Name	ID
Calendar Tool Administrator	admin
Landis, Haley	hlandis
Stier, Ellen M.	estier

Figure 324: List of Calendar Tool administrators.

refers to a user with administrative privileges running the Calendar Tool Administration program. The term "user" refers to a regular user running the regular Calendar Tool program. Also, the term "privileged `Admin` menu" refers to the command menu in the Calendar Tool Administration interface (Figure 4). The term "`Admin` menu" refers to the menu accessible from the regular user Calendar Tool interface (Figure 2).

There are two contexts in which administrative commands appear -- in the admin program and after the user establishes a connection in the regular UI.

When the user establishes a connection with a remote host, the '`Admin`' appears on the menubar. Prior to establishing a connection, or after close all connections, the '`Admin`' menu is not in the menubar. Since establishing a connection is a precondition to working with the administrative commands, it's described here.

TODO:

a. Remove '`Admin Privileges`' radio buttons from add/edit user dialogs, replacing them with a separate admin-editing dialog. NOT to what follows -- there are not admin users any more, just an admin password for any given installation of CalTool Admin. ***NOW OUT***: The latter dialog is accessed via an '`Edit Admins ...`' item in the privileged admin menu, which replaces the current '`List Admins ...`'. This means that the non-privileged `Admin` menu has a '`List Admins ...`' item whereas the privileged `Admin` menu has '`Edit Admins ...`', which is cool.

b. Change '`Item Limit`' to '`Size Limit`', to avoid malicious flooding of a user calendar with an excessively long '`Details`' field value.

c. Update things as necessary throughout to eliminate admins as regular users, leaving a UNIX-style admin that exists solely by virtue of knowing the admin password, which in this case is used to invoke the CalTool Admin program on some host. As with UNIX admins, it's up to them to arrange sharing of the password(s), including whether they want different passwords for different hosts. This updating entails:
   i. Changing the '`List Admins`' item in the regular-user `Admin` menu to '`Contact Admin`', which brings up a dialog allowing the user to submit a message to the admin of the Calendar Tool central host to which the current calenndar is connected. This command is disabled when the current calendar is not connected to a central host. What's probably nice is to have the dialog come that says that in order to contact an admin, the user has to be connected, and offering to op the '`Central Host`' dialog so the user can connect where desired. But this is unnecessary now, since the entire Admin menu only appears when connected. Cool that the banner has the current host name, BTW.
   ii. Removing '`List Admins`' entirely from the priv admin menu, as described above.

4.11. Registered Users

OK, the evidently unspoken rationale for having local host user ID be part of the cal tool user record was this. The local machine would provide the user-level access structure, including passwording, and then the cal tool would rely on this in the sense the mapping from local ID to cal tool ID was the only cal tool validation necessary, the user having (presumably) entered a password to gain access to the local machine. The perceived advantage of this is that the user on a local machine does not have to log in twice, once to the machine and then again to the cal tool.

There are at least a couple problems with this structure however. The first of which was noticed pretty early on. That is that it's a pain for the admin to have to have multiple local computer potentially listed in the user records. At the same time, it's a pain, or evern unrealistic to require that the user IDs be the same on all connecting local machines. Another problem, which is just now coming to light, is that it is in some circumstances unrealistic to expect a specifi local-side login protocol that the remote cal tool server can rely on. What comes to mind most clearly is the access-from-internet-cafe configuration, where the ID of the connecting machine cannot possibly be known (in advance) to the remote caltool server.

Given these problems, the user model that is better is to require both user ID and password in the caltool user record. The inconvenience of double login can be avoided by allowing the user to specify a default cal tool user ID and password as an option. These will be automatically supplied when a user connects to a remote host. Also, admins on a number of cal tool hosts in a given organization can take care to give each user the same cal tool ID on all hosts, so a single default will work for a given user connecting from one or more local machines. The bottom line here is that the cal tool does not rely on any local-machine login protocol to validate that a cal tool user can connect to a remote host.

The other thing that has to be changed in this new user model is the idea that the regular user interface to the cal tool can know the (local) user ID at start-up time, therefore know the cal tool ID (since it's the same), and therefore know whether to start up the registered or non-registered UI. The deal (now) is that launching the cal tool locally does not in and of itself make a connection to a remote host, even though it might seem to do so by virtue of the auto-connect-on-open option. The important point is that launching the cal tool locally cannot necessarily mean that a remote host connection is automatically being made, and even if it is that a particular single cal tool ID is to be used for the connection. Therefore, automatically matching a local ID to a cal tool ID at start up, or even explicitly requiring al login at cal tool start up, is not generally useful. The login is part of the connection process, not the start up process. Given this, there either need to be two separate "regular" user apps, or we can just forget about the unregisted version, just leaving all of the would-be missing features disabled. Another alternative might be to have the entire Admin menu not appear until a connection, but that can't work with exactly the current Admin menu, since it's got the connect command on it (indirectly). So, it's either two separate apps, or one with Admin permenantly disabled. It's not all that hard to provide two, so what the heck, I think we should just do it.

CONCLUSION: OK, here's the deal. We've moved the '`Central Host ...`' command from the '`Admin ...`' menu to the '`File`' menu, and renamed it '`Connect`'. Also, we've moved '`Local Directory ...`' to the file menu, renaming it '`Local Files ...`'. The point is to avoid the chicken-and-egg problem noted above where we can't get to connect if the '`Admin`' menu is initially not in the menubar. With these changes, we can say that the '`Admin`' menu does not appear until at least one successful connection is established. Further, the DBs and admins to which the '`Admin`' menu apply at any given time are the host to which the current calendar is connected. If the current calendar is not connected to a host, then the admin menu is disabled, but still on the menubar. The '`Admin`' menu disappears from the menubar when all connections to cal tool central computers are severed.

7jul04 NEW CONCLUSION: Fuck the chicken-and-egg bullshit around the `Admin` menu. Just leave it the fuck up all the time, with the DB commands and passwd command disabled unless the current calendar is connected. This means we move the `Connect` command back to the `Admin` menu where it belongs. The fact that it belongs in the `Admin` as opposed to `File` menu was brought pretty clearly home when I tried to write a decent intro paragraph to the **Flie** section, whereupon it became pretty darn clear that `Connect` is only weakly related to file processing. Furthermore, the fact that I've seen fit to explain the connect command fully in the **Admin** section makes it quite clear indeed that it should be an `Admin` as opposed to `File` menu command.

4.12. "View ..." versus "Edit ..." as Button and Menu Item Names

"View ..." is used when there is more information to see about a particular item of data. The displayed dialog may allow editing, with `Change` and `Delete` commands, but without an `Add` command. "Edit ..." is used when the when the displayed dialog has all editing commands, include `Add`. "Edit ..." is the menu item for categories, lists, and filters. "View ..." is used as the menu or button for other viewing/editig dialogs, including meeting notifications, meeting minutes, and a few others.

4.13. The Explicit Delay Button in Meeting Notifications

It's clearer than just having the user close the window to delay. It was decided that a Delay button is not necessary in the item-level view of a not-yet-accepted item since the user isn't necessarily there to make an accept/decline decision the way she is in the notification dialog.

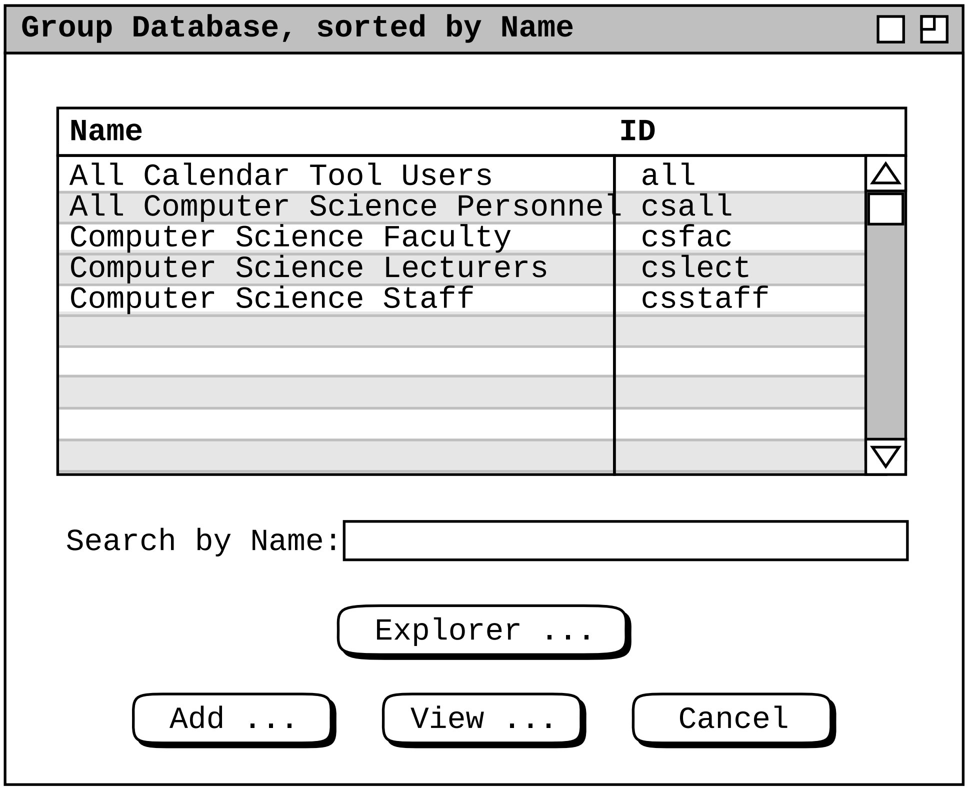
4.14. Group Explorer

Thought about adding a tree-style group/user explorer, but it's a bit much given that the group structure is potentially a graph, not a tree. E.g,

```
campuslect = cslect
csdept = csfac, cslect, csstaff
cslect = b,c,f
```

While this could be rendered in tree, e.g., via some kind of copying rule, it's more trouble than its worth. Here's some verbiage that was going to go in the Section 2.6.3, including a screenshot of the updated group db GUI.

... 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 325. This dialog has the same basic format as that for the user database in Figure 201, with the addition of the '`Explorer ...`' button. Two groups are built-in to the Calendar Tool: `all` and `admins`. These groups consist of all registered users and all administrative users, respectively.



Name	ID
All Calendar Tool Users	all
All Computer Science Personnel	csall
Computer Science Faculty	csfac
Computer Science Lecturers	cslect
Computer Science Staff	csstaff

Explorer sketch:

```
all
csall
  csfac
    cslect
      l1
      l2
      ...
    csregular
      r1
      r2
      ...
  csstaff
```

Note that all group is not a supergroup but a flat list of all registered users.

Figure 325: Group database dialog with explorer.

When the user presses '`Explorer ...`', the system display the group viewing window shown in Figure 326. The explorer shows all groups and users in a browsable herarchy. The groups and users are listed by ID. The default initial configuration of the explorer is the group hierarchy fully explanded, ...

This is the point at which I figured things were going to get out of hand here. Smart thinking, actually.

4.15. 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. Constrast this to the essentially negligable 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'd.

4.16. Fodder from Wincow 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 fargs" 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 cals) may be OK, given that chrono for cals 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 cals just fine, it being "most recently visited (i.e., current)" based on window order. I.e., the chrono order of the cals can be defined as a suborder of the cals 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

This would be a standard kind of tree/graph-based explorer.

Figure 326: Group explorer.

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 bout 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.)

4.17. 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 presuambly 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 assciation rules, the possibilities being:

a. ... enumerate ’em

Desribe things, when you feel like it.

4.18. 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 "`~`".

4.19. 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.

4.20. 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 subgroups by clicking on a parent group.

c. The default ..., fell off.

4.21. 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 .

4.22. Hopefully Final Nukes for Options

```
Admin
  Default shared user calendar file name: [default = "Calendar"]
  Home directory leading path:
```

4.23. 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 lccal 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 simplication 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.

4.24. 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 thes 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 permenant 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 unediable 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, indepednent 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 327.

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 327. 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 328 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, 10-point Times-Roman bold font is selected. Figure 329 shows the result of the new font selections for the monthly view shown originally in Figure 21.

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.

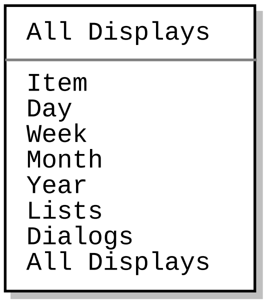


Figure 327: Font options menu.

To appear.

Figure 328: Font option values selected for month-level displays.

To appear.

Figure 329: Monthly view with user-selected fonts.

4.25. 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 considersaions 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 scenarions, 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 hiearchy.
c. Do the detailed scenarions for the options dialog.
d. Refine in earlier sections as necessary.

4.26. More Potential Rationale Fodder Cleansed from Rapidly Maturing Options Setion

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 rate) case that we want to do it for a bunch of calendars).

4.27. 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.

4.28. 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 no where 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 exectues '`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 currntly 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 explantory 29jul02 remark that preceeds 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 leas 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 embeded in every calendar file, and as such, not that amenable to formating 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 inconsisten 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 exaple:

```
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 pane to deal with in a options script file.

24jul02 -- Try this *(29jul02 -- italics after each point indicate whether the point has been incorporated (OK) or nixed (NIXED) from the requirments)*:

- 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 seprate 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 knowldege 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 (!).)*

4.29. 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 porcessing 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

> 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 dialoag 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.]

4.30. 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 struture. This includes a dicussion 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.

4.31. 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 accidently 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

4.32. More on Admin Decisions

Removed the '`List Admins`' command in favor of having an a'`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 adminstrator 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 ...* .

4.33. Fodder Nuked from Admin Section

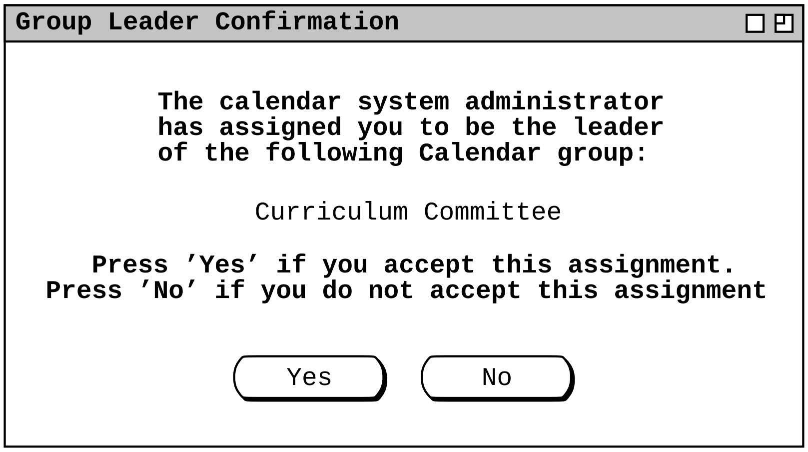
.sh 4 "Group Membership Requests"

An individual user should be able to:

a. see what group she's a member or
b. request (demand?, force?) removal from the group
c. request (demand?, force?) membership in a new group

4.34. Detailed Explanation of Recurring Date Change Restrictions

Recall that all intances 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 calender 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. Coversely, 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.



The calendar system administrator has assigned you to be the leader of the following Calendar group:

Curriculum Committee

Press 'Yes' if you accept this assignment.
Press 'No' if you do not accept this assignment

Figure 330: New group leader confirmation.

4.35. Rationale for 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.

4.35.1. 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.

4.36. From the task scheduling section

The following issues were all dealt with as appropriate.

..., subject to the following restrictinons:
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 appoinemts, 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)

4.37. 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 .

4.38. Possible fodder for meeting change/delete section rationale

[in the sched delete confirmation dialog] The user may edit any data fields accept '`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 331 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 precentage 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.

```
You are the scheduler of this meeting.  Do you want all attendees to be
                    notified of the change?
```

Figure 331: Changing or deleting a meeting as leader.

4.39. 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.

4.40. (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 excutes '`View Item`'. For example, Figure 332 shows the user having selected the October 1 meeting instance in a month-level display.

4.41. 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.)

4.42. 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 2.8.2 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 extern program for viewing minutes, such as a web browser. Selection of such an external viewing program is explained in Section .

It is entirely the responsiblility of the scheduler to name the files in such a way that users can discern which minutes file applies to a particular meeting occurance.

4.43. 2aug01 -- Nixing the notificaiton 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 priveleges 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.

4.44. 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

4.45. 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 Parms" 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 possbles 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 addtions to user options, the user can control when she sees alert -- at starup, 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.

4.46. 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 that 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 that all those with times, per latest details of list sorting. Keep it simple, dickhead.*

4.47. 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.

4.48. Relative Importance of Actual Tool versus Pedagogical Exmaple Goals

Some frank and 100% truthful discussion goes here.

4.49. 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 priorty; (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.

4.50. Refinement Example

See (and discuss here or elsewhere).

Start Date: 1 week ago ▼ End Date: 2 weeks from now ▼

4.51. 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.

4.52. 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.

4.53. 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 whit, 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 man 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.

4.54. 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.

4.55. 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.

4.56. 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.

4.57. 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.

4.58. 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.*

4.59. 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 whit, 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.

4.60. 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.

4.61. 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 an 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.

4.62. Canonical Modeling Form

Interestingly, it looks as if the cannonical 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.

5. Developer Overview

5.1. Issues to cover, if not addressed elsewhere

NOTE: I'm leaning towards having at least what's calle the "bootstrap problem" just below dealt with in the "Platform Dependencies" subsection of Installation, rather than here. The reason is that it's a user-visible issue. If I've not said it yet, what goes here in development overview are *only* issues that are fully invisible to the user.

- There's the bootstrap problem of where to store the location of Cal Tool user directory. Leave this up to the implementors, but suggest a fixed known file rather that modification of the application itself. There are presumably platform-dependency issues here, e.g., in OS X, the "application itself" is acctually an entire directory hierarchy. The thing is, I personally think it would be nice if the date stamp of what appears externally to the OS user as "the application itself" does not change every time the user changes the location of the Cal Tool user directory.
- ... OK, rock on, dude.

OK, it seems like we've embarked in the above list on 17jul03 note (below) to discuss "Implementation Considerations", which is fine. To get this fucker done, we can just dump here whatever we've thought of along the way, and be sure to keep separate requirements from suggestions and guidelines. It might in fact be a great idea to keep this section completely *requirements free*, in that we can say that *everything* in this section is a suggestion, recommendation, or potentially useful observation for the implementors, but not a requirement. To this end, I've put the requirment for the date stamp of the app not changing to non-functionals, and we can cite that requirement here in the lower-level implementation discussion.

17jul03 note: Consider adding a subsection on "Implementation Considerations", wherein we discuss all of the thoughts we've had on potential implementations, i.e., the one's we've run through in our heads to make sure that what we're thinking about is possible. It may not be necessary to even have such a subsection title explicitly, just include such considerations here as a matter of course. Pedagogically, we do need to mention that such considerations are part of what should go here, in general, unless the implementation is 100% straightforward (which one can scarsely ever imagine being the case). Part of these considerations can also be explicit mention of what the implementors are free to do, even if such mention is not strictly necessary given the standard rule that if the user can't see it, the implementors can implement it any way they want. E.g., a case in point in the cal tool is that the implementors are free to implement the central host calendar as a single file, or multiple files, such as one per user, since the user cannot expect to find his calendar in file form on the central host, but can only get access to it via running the calendar tool. This is not to say that an implementor is prohibited from putting it on some file that the user might be able to see, as long as visibility constraints are met.

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 form 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.

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, includin 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_idenity)
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

Final Todo Items