Chronograph

version 1.03

by John A. Schlack

User's Manual
September 1994

Chronograph Contents

The following items should have been included with the *Chronograph* software package:

- *Read Me First*
- *User's Manual* (this document)
- *Getting Started* Folder containing an example cron creation session
- *Registration Form*
- *Cron Master* Application
- *Cron Daemon* Application (standard Macintosh)
- *Cron Daemon PPC* Application (Power Macintosh)
- *Chronograph ƒ* Folder containing *crontab* text file and *contabs* folder

Requirements

This program requires System 7.0 or later to run. It can execute on both standard Macintoshes and Power Macintoshes. The *Cron Master* program is a "fat binary". It has code for both standard Macintosh and Power Macintosh computers in the same program. Since the *Cron Daemon* application will always be running, its size was minimized by including a version for standard Macintosh computers (*Cron Daemon*) and one for Power Macintosh computers (*Cron Daemon PPC*). These programs function exactly the same way and will simply be referred to as *Cron Daemon* .

Introduction

Chronograph is a utility that is similar to the UNIX *cron* facility. Cron is a scheduling utility that permits programs to execute on a regular basis. Once per minute, it checks to see if any jobs require execution. If so, it executes these programs and then becomes idle until the next minute arrives, when it checks for scheduled jobs again.

Chronograph has these features, but is more flexible and easier to use. Any object that can be double clicked in the Finder can be scheduled (except folders and disks). (Note that scheduled items will be called jobs, events, or tasks). Sounds can be played, files opened, or applications and scripts launched. Jobs can be scheduled by the minute, hour, day, day of week, and/or month. One can specify whether to execute a task only once or repetitively.

Since this program has its roots in UNIX land, one might think that using the program is intimidating (or at least difficult to use). Although one may opt to use the cryptic UNIX method for specifying tasks (and this method is the most powerful way to specify tasks), a simple Macintosh interface has been added to make *Chronograph* easy to use.

One may wonder why Chronograph should be used at all. What possible tasks can be scheduled? Daily backups, reminder messages, hourly chimes, and checking for E-mail messages are sample uses.

Installation

Check to make sure all components are present as specified in the *Chronograph Contents* section above. Perform the following instructions for installation.

1) Place the *Chronograph* folder (the folder which contains all of the Chronograph files) in a suitable location on your hard disk (such as a utilities directory)
2) Move the *Chronograph ƒ* folder into your Preferences folder (located in the System Folder).. This step can be skipped if you would like the application to create its own copy.
3) Make an alias of *Cron Daemon* (or *Cron Daemon PPC* for Power Macintosh computers) and place the alias in the Startup folder (located in the System Folder).

Alternatively, move the *Cron Daemon* application to the Startup Folder.

4) Either restart the computer to automatically launch *Cron Daemon* or double click on it to manually start the program (subsequent restarts will automatically launch the program).

Scheduling a Job

There are several methods that can be used to schedule a *Chronograph* job. One may launch the *Cron Master* application and select *New* from the *File* menu or strike the *Add...* button in the *Cron Master* Main Window (see below for details). One can drag-n-drop a set of files onto *Cron Master* in the Finder. Both of these methods create an Event Window for graphical schedule editing (see the *Cron Master* section below).

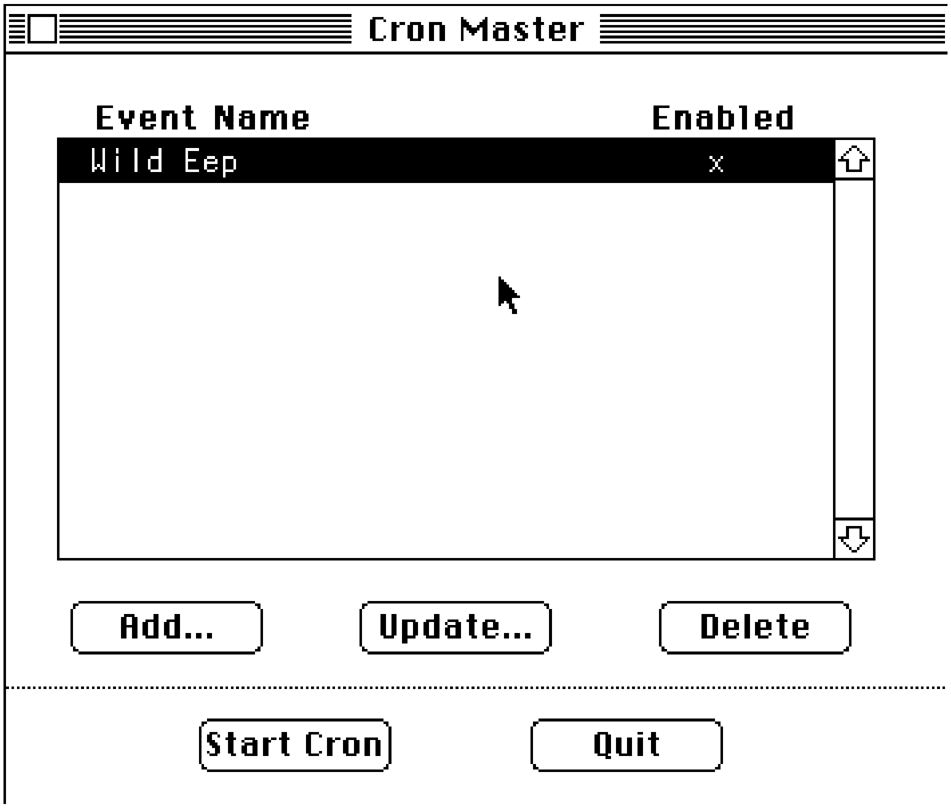
One may use the more cryptic (but powerful) UNIX method. Crons can be entered into a text file named *crontab* which is located in the *Chronograph ƒ* folder (which is in the Preferences folder). For the exact format of this file, see the *crontab File* section below. One may also create a *crontabs* folder. This folder is located in the *Chronograph ƒ* folder. Every text file within this folder is treated as a crontab file. Therefore, one may assign a different file to each system user, permitting each user to specify jobs. Note that all crons in this folder will be executed. The system currently does not check for a user name and only execute those events (but perhaps in the future...).

Cron Master

Cron Master is the program that governs the creation and modification of Chronograph Events. It is designed to provide basic event definition functionality using a graphical user interface. One can also use an advanced mode for exacting control over the event's operation. Note that *Cron Master* is not needed to use *Chronograph* . Events can be defined using the *crontab* files located in the *Chronograph ƒ* directory (in the Preferences folder).

Events can be defined in *Cron Master* by several methods: drag and drop, menu commands, and Add button from the Main Window. Any event creation will display an Event Window that is named after the cron item. This window permits alteration of the job settings.

When starting *Cron Master* , the Main Window will always be displayed. This window shows a list of the current events and whether they are enabled (see Figure 1). Only one item in the list may be selected at any time. Each line of the list displays the job name (name of the file used for the cron) and whether it is currently enabled. For detailed information about the event, either double click on it or select it and strike the Update button. This will display the Event Window for cron viewing and modification.



Event Name	Enabled
Wild Eep	x

Figure 1: Cron Master Main Window

Five buttons are visible in this window: Add, Update, Delete, Start/Stop Cron, and Quit. Descriptions of these buttons follows.

The Add button will display the Standard Get File dialog box, permitting the user to choose any file as the basis for a new cron. The only exceptions are cron job files. After choosing a target file, the Event Window will be displayed (see Figure 2), permitting selection of job execution parameters.

The Update button is only active if an item from the cron list is selected. Clicking update (or double clicking on a job in the list) will display the Event Window with the event's current settings, making job modification simple.

To delete a job, one simply selects the event from the cron list and strikes the Delete button. Alternatively, one can remove the cron's file from the *Chronograph ƒ* folder. Note that the second method does not update the *Cron Daemon* , which will continue executing the cron until the program is restarted. The Delete button is only active if an event is selected from the cron list.

The Start / Stop Cron button will launch *Cron Daemon* if it is not running or kill the program if it is executing. The button will be named Start Cron and Stop Cron if the *Cron Daemon* application is idle or running, respectively. On may need to located the *Cron Daemon* application the first time one strikes the Start Cron button.

This button is useful if one wishes to stop the background program to free up memory or to restart it to recognize changes to *crontab* files.

The final button is the Quit button. Striking this closes the *Cron Master* program.

The Event Window is a dynamic window which displays the settings for a specific cron job. Any time a new cron is created or one wishes to view or modify an event, an Event Window is created. Thus many Event Windows can be on screen at the same time. There is only one Main Window, however.

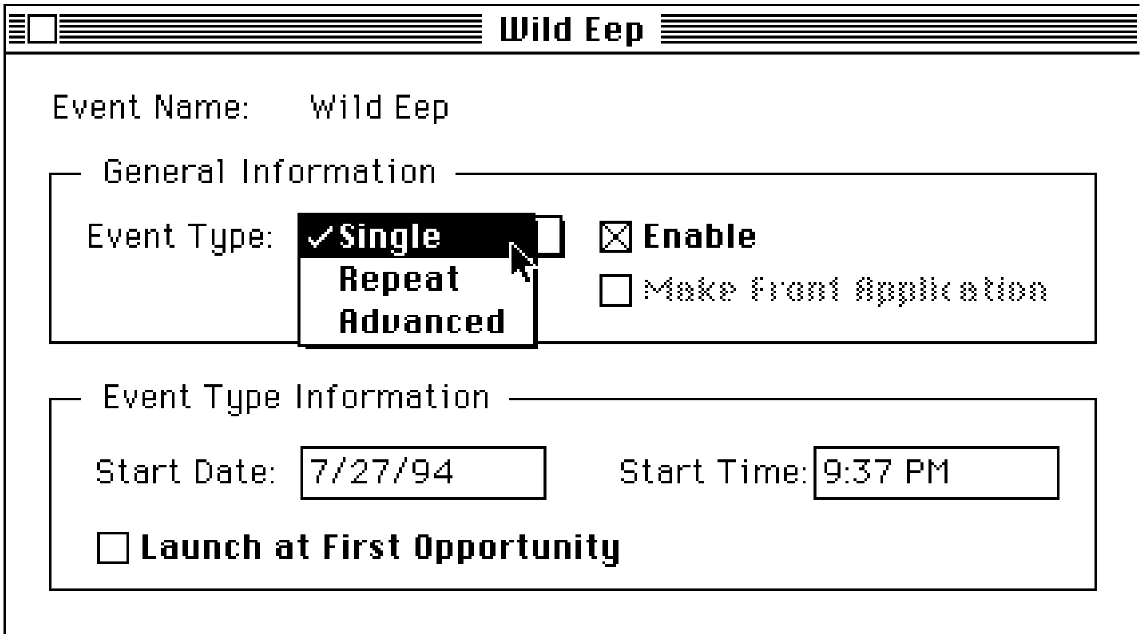
Certain basic information is displayed in the Event Window. This includes the job name, event type, and whether it is currently enabled. The top half of figure two displays the basic information.

There are three event types: single, repeat, and advanced. Single is a one shot deal. The job is scheduled to execute at a specific date and time. Once it does so, the event will be automatically deleted. Repeat events are jobs the execute periodically. The frequency of execution and termination conditions can be specified by the user. The advanced event type is the most flexible. It has a command line interface and permits the user to specify the execution times and frequency using the standard UNIX definition method.

The "Enable" check box will toggle between a active and disabled state for the event. When enabled, if the execution time arrives, the job will activate. If not enabled, the job will not execute.

The final general check box is "Make Fron Application". This applies only to crons whoses basis are applications. When the cron activates, the target application will be made the front-most if this item is selected.

The most basic event type is the Single event type (smaple Event Window is shown in figure 2). In addition to the general information, the start date and time are specified in the appropriate editable text boxes. In addition, one may select to execute the job at any time after the start date/time is reached (provided the event is enabled). This is useful if *Cron Daemon* was not running when the execution time was reached (computer was off, *Cron Daemon* was killed, etc.). Note that once the "single shot" executes, the event will be automatically deleted.



Wild Eep

Event Name: Wild Eep

General Information

Event Type: ✓Single

☒ Enable

□ Make Front Application

Event Type Information

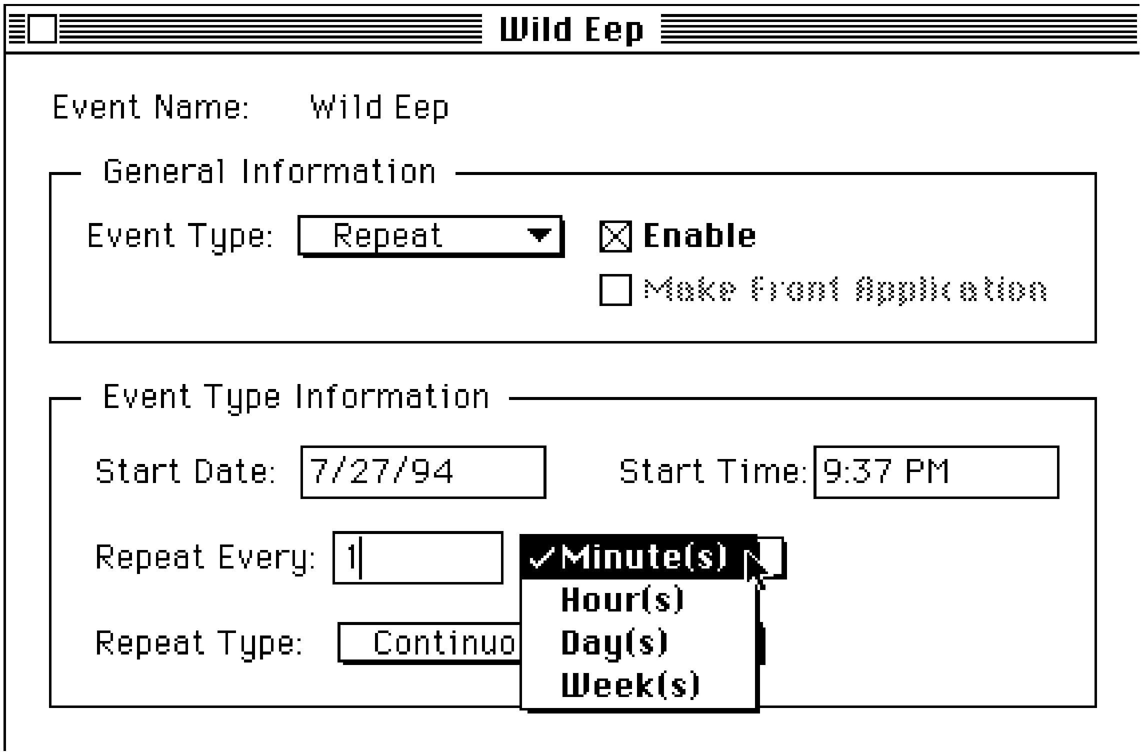
Start Date: 7/27/94

Start Time: 9:37 PM

□ Launch at First Opportunity

Figure 2: Event Window (Single Event)

The most used event type will probably be the Repeat event type. This job will execute periodically and may or may not have termination conditions. If termination conditions exist, the event will be automatically deleted once those conditions are met. Sample Event Windows for Repeat event types are shown in Figure 3 and 4.



Event Name: Wild Eep

Event Type: Repeat ☒ Enable □ Make Front Application

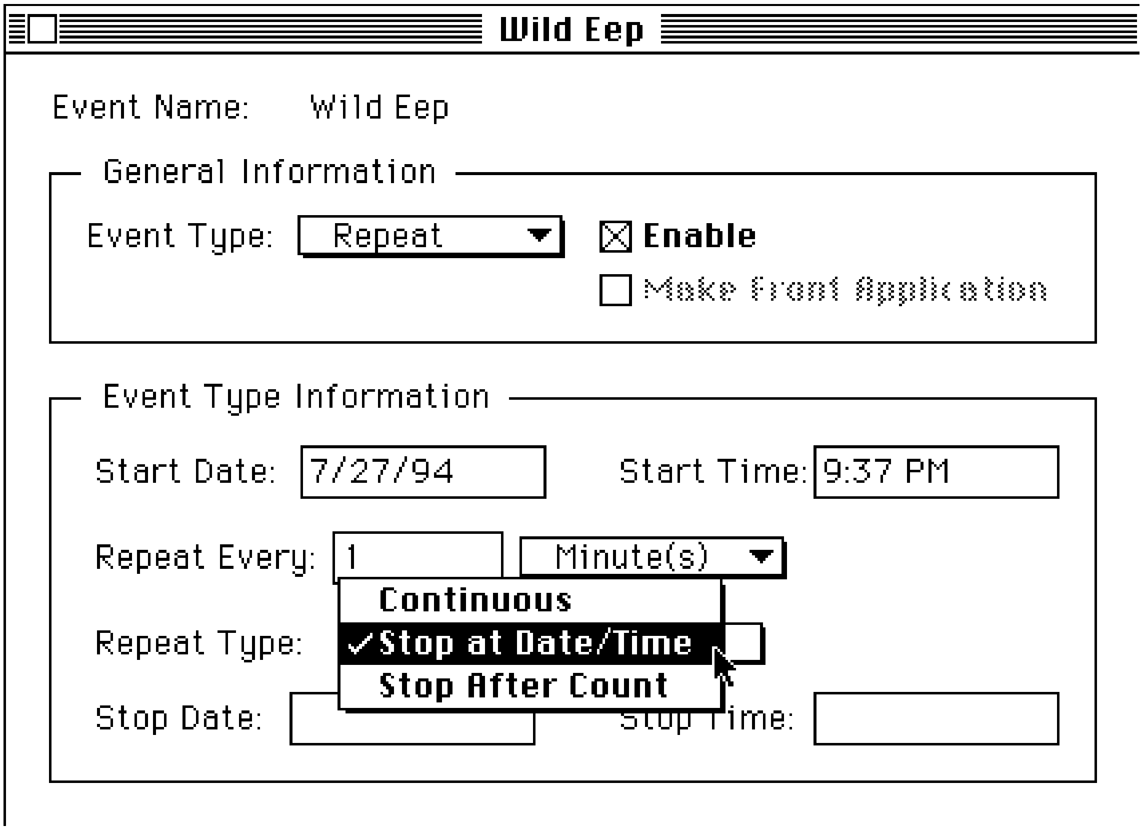
Start Date: 7/27/94 Start Time: 9:37 PM

Repeat Every: 1 Minute(s)

Repeat Type: Continuo

Figure 3: Event Window (Continuous Repeat Event)

All repeat events must specify a start date and time. The repeating job will not begin until the specified date and time is reached. The other common element of Repeat events is the frequency ("Repeat Every" line from the Event Window). One may select a time base of minutes, hours, days, or weeks. Once chosen, the user enters how much time must pass before the event executes again. For example, suppose Hour is chosen as the time base and 12 is entered as the count. Thus, the event will be repeated every 12 hours. Note that if more exacting control is required, one must use the advanced event type.



Wild Eep

Event Name: Wild Eep

General Information

Event Type: Repeat

☒ Enable

□ Make Front Application

Event Type Information

Start Date: 7/27/94

Start Time: 9:37 PM

Repeat Every: 1 Minute(s)

Repeat Type: Stop at Date/Time

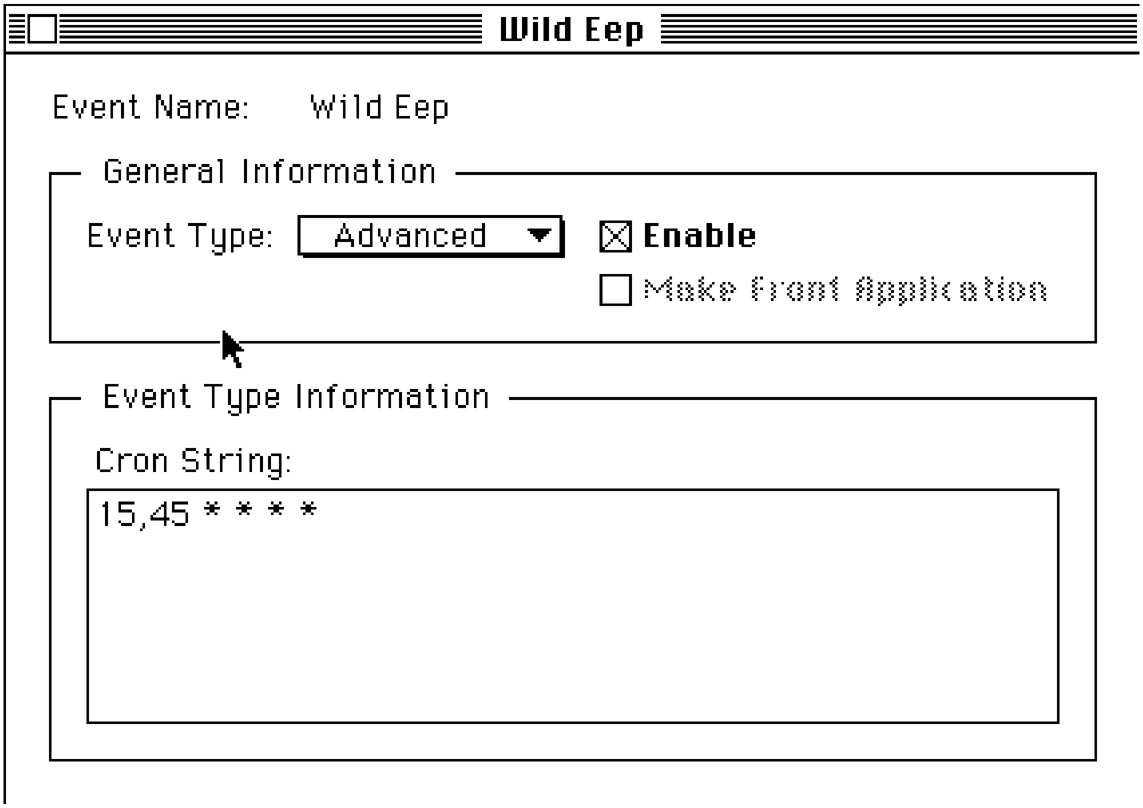
Stop Date:

Stop Time:

Figure 4: Event Window (Terminating Repeat Event)

The repeat type specifies the termination conditions. Three types exist: Continuous, Stop at Date/Time, and Stop After Count. Continuous means that the event will continue to repeat periodically until the user stops it. Stop after Date/Time permits the user to enter a date and time to stop executing the event. Once that time is reached, the job will be deleted. Finally, one can specify a number of times to execute the job. Once the execution count is reached, the job will be removed.

The final event type is the Advanced event (see Figure 5). The user must specify the repeat conditions in a format similar to UNIX cron descriptions. Note that only the five time sets must be specified. The item to execute was already chosen (by the Add button, drag and drop, etc.). See the *crontab File* section for detailed information about specifying the five time parameters.



Wild Eep

Event Name: Wild Eep

General Information

Event Type: Advanced

☒ Enable

□ Make Front Application

Event Type Information

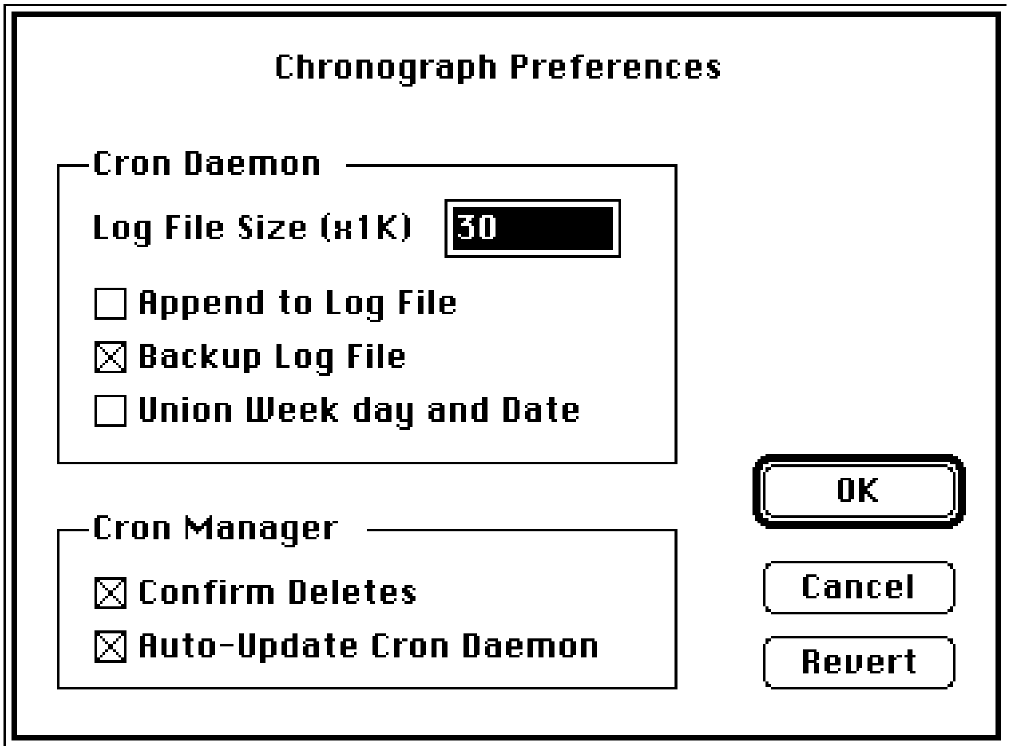
Cron String:

15,45 * * * *

Figure 5: Event Window (Advanced Event)

Preferences

The preferences for both *Cron Daemon* and *Cron Master* are controlled through the Preferences Dialog Box in the *Cron Master* application (see Figure 6).



Chronograph Preferences

Cron Daemon

Log File Size (x1K) 30

☐ Append to Log File

☒ Backup Log File

☐ Union Week day and Date

Cron Manager

☒ Confirm Deletes

☒ Auto-Update Cron Daemon

Figure 6: Preferences Dialog Box

"Log File Size (x1K)" permits the user to specify the maximum size of the *Cron Daemon* 's log file (in multiples of 1024 byte blocks). Each time a job is executed or an error occurs, an entry is made in the log file (see the *Cron Daemon* section). Limiting its size prevents the file from growing forever.

"Append to Log File" will append messages to the current log file whenever *Cron Daemon* is restarted. Leaving this control unchecked causes a new log file to be created each time *Cron Daemon* is launched.

If "Backup Log File" is selected, then any time the current log file reaches its file limit or a new log file needs to be created, the current one is saved as a backup for further reference (and the current backup will be deleted). Otherwise, if a new log file must be created, the current one is discarded. It is suggested that for debugging purposes, this item should be checked.

"Union Week Day and Date" is a preferences item designed only for advanced event types (UNIX style cron definitions). It is only required if both the week day and day of month fields are specified.

If selected then only one of those two fields needs to match for the job to execute. If not checked, then both fields must match. For example, suppose "0 17 10,20 * 1" is specified as the job execution times. If checked, then the job will execute at 5 PM on the 10th and 20th of every month along with every Monday. If not checked, the job will only execute at 5 PM on Mondays where the day of the month is the 10th or 20th.

"Confirm Deletes" will display an Alert Box any time the Delete button is struct, querying the user whether to actually delete the selected event. Set this check box to help prevent accidental cron deletions.

"Auto-Update Cron Daemon" will send messages to the *Cron Daemon* application any time preferences are altered or jobs are added, modified, or deleted. It is recommended that this check box be set so that *Cron Daemon* does not need to be restarted for modifications to take effect.

crontab File

These files provide a UNIX-like method for specifying a job. The file must be a text file. It can either be named *crontab* and located in the *Chronograph ƒ* folder (inside the Preferences folder) or have any name and be located in a folder named *crontabs* , which resides in the *Chronograph ƒ* .

Any non-blank line is treated as a job definition. The only exceptions are lines that begin with '#'. These are treated as comment lines. Note that as opposed to UNIX systems, a job must be completely described on one line. It cannot continue on the next line. Also, the path name to the file to execute must be enclosed in quotes (discussed below).

Below is a sample crontab line. It has 5 fields (space or tab separated) that define the schedule. These fields contain either a number, asterisk, or list of numbers which are comma separated (no spaces). It is followed by the full path name to the application or file to schedule. The full path MUST be enclosed in quotes. Therefore, no quotation marks may appear inside the full path name.

The first field represents minutes after the hour (0 to 59 valid). The second field is the hour of day in 24 hour format (0 to 23). The third field is the day of the month (0 to 31). The fourth field is the month of the year (1 to 12). The last field is the day of the week (0=Sunday to 6=Saturday). An asterisk means "every".

Example Line:

```
* * * * * "HardDrive:Folder:SubFolder:...:ApplicationOrFile"
```

Example Schedule Fields:

30 5 * * 1	5:30 AM every Monday
30 5 * * *	5:30 AM every day
30 * * * *	30 minutes past each hour, each day
0 17 * * 1,2,3,4,5	5:00 PM Monday through Friday
* * 15 * *	every minute of every hour on the 15th day of every month

Troubles may arise if both day of the month and week days are defined for a job. These can be handled in one of two ways. Either both must match up [intersection] (such as if day of the month was 10,20 and week day was 1 [Monday], then the job will only execute on Mondays that have the day of the month equal to 10 or 20). Alternatively, one may execute the job is either match [union] (in the previous example, the job executes every Monday, in addition to the 10th and 20th of the month). The method to use is defined in the preferences. The default is the intersection method.

Cron Daemon

This is a background-only application. This program or an alias to it should be placed in the Startup folder so that Chronograph is enabled at startup. *Cron Daemon* has no interface and once launched, continues to execute until the computer is turned off.

Every minute, this program checks to see if any jobs are scheduled to execute for that particular minute. If so, those tasks are carried out.

If you really need to kill this program without restarting, you may either use AppleScript to send a quit command to *Cron Daemon* or use the *Stop Cron* button in *Cron Master* .

Note that this is an application, not an extension or control panel. As such, it requires the front-most application to be well behaved and provide background applications with processor time. If you regularly run applications that do not yield time (some games, very computationally intensive programs, or programs written before System 7), *Cron Daemon* may not receive enough CPU time to operate effectively (i.e. it may not execute jobs as expected or at all). This situation cannot be resolved until the Macintosh gets a preemptive multitasking Operating System (last rumored to be in 1996 - but don't hold your breath).

Cron Daemon keeps copies of the jobs in memory (to reduce file I/O, especially useful for PowerBooks). Therefore, changing the *crontab* file or manually deleting jobs from the *Chronograph ƒ* directory will have no effect on cron execution until *Cron Daemon* is restarted. *Cron Master* can dynamically update *Cron Daemon* if the proper preferences options are set up (see the *Preferences* section). The only way for modifications to a *crontab* file, to become active are through a restart.

Cron Daemon keeps a log file to help track down problems. The file is named *Chronograph Log* and is located in the *Chronograph ƒ* folder. Every time a job is executed or an error or significant event occurs, the data is written to the log file. This data file can grow quite large if many jobs execute repeatedly. Two mechanisms have been implemented to reduce the file size: log size limits and creating a new log file on *Cron Daemon* startup.

One may specify the maximum size of the log file through the preferences in 1K increments. One may also specify whether a new log file should be created on startup or if one should append to the current one (also part of the preferences). Finally, a backup of the last log file can be kept. Whenever a new log file needs to be created, the current log file will be backed up to the file named *Chronograph Log Backup* .

Registration

Chronograph is Shareware. If you use this application, please pay the Shareware fee of $15. Much effort went into creating this program and your support of Shareware authors will help continue the flow of useful and innovative Macintosh software.

You may also purchase the source code for $45. It requires Metrowerks Code Warrior release 3.5 (code is written in a combination of C and C++). Note that you may not release software that competes against *Chronograph* based on this code, nor may you provide a copy of the source code to anyone without written permission from myself (John A. Schlack). You may use the code in your own projects (which you may release) or to add features that you desire to *Chronograph* (but you may not release modified versions of *Chronograph*).

After you register, you will receive a disk with a version of Chronograph that does not display the Shareware registration message. Users who purchased the source code will also receive that on the disk. **Due to shipping costs, users outside the continental United States must add $2 for shipping and handling. All registration fees must be paid in U.S. currency.**

You may use the Registration form that accompanied this software or simply send the registration fee to:

John A. Schlack
406 Newgate Court, Apt. A1
Andalusia, Pa. 19020
USA

Register Online!

For users of CompuServe, you can register this software online! Simple type "GO SWREG" and use the registration ID of 3308. After you register, I will E-mail a "thank you" message and send you the registered version by snail mail.

Bugs / Questions / Comments

Chronograph was developed entirely on a Power Macintosh computer and then compiled to run on a non-PowerMac. Please E-mail bug reports to the program's author via E-mail. Every attempt will be made to correct problems (within financial and time constraints). Note that if you have problems running *Cron Daemon PPC* on your PowerMac, try using *Cron Daemon* .

You may also send questions or comments to the following E-mail addresses:

CompuServe:	70252,143
America Online:	John40
Internet:	John40@aol.com

I hope you find *Chronograph* useful! Remember to register.