EZCronGuideFile

COLLABORATORS			
	TITLE : EZCronGuideFile		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		March 28, 2025	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

EZCronGuideFile

1.1 No title

```
                           EZCRON v2.10
                            February 1997
```

1.2 Introduction

EZCron Introduction

This project was started mainly because I had a need to run lightwave between newscasts (I work for a television station) and do automated rendering whenever I wasn’t present.

The program has grown from a simple rexx script to a full fledged GUI oriented program that is easy to use, and doesn't require a knowledge of UN*X.

EZCron actually consists of two programs. The first is the EZCrond (daemon) which runs as a background process transparently to the user. It's very small and pretty fast. EZCrond will sleep between every minute,only awaking to check if there are any events that need be executed. Its CPU usage is minimal!

The second program, EZCron is the user-interface (formally EZCronGUI). This is the Event Manager and allows you to easily add and edit your events. It will also allow you to start and stop EZCrond.

EZCron is the result of many many hours of work and many contributions and ideas from the you, the end users.

1.3 Requirments

SYSTEM REQUIRMENTS

EZCron should run on any Amiga with Workbench/Kickstart 2.04 or better with 512k of ram. This programs/librarys are required to make various parts of the EZCron programs work.

- RexxMast MUST be running

- ARexxPort.library by Andy Cook (included)

- RexxReqTools.library 37.71 or later (included)

- Varexx by Andy Cook (included)

1.4 Getting up and running

Getting Up and Running

INSTALLATION

********IMPORTANT********

As of version 2.00+ of EZCron, a lot has changed. Please DELETE the contents of your old EZCron directory before installing this one. You can also copy the old one into another directory if you like.

****

To run EZCron, you first need to install it. This can be done simply by double clicking the EZCron-Install icon. Or, if you're like me and want to install it manually, see MANUAL~INSTALLATION. The installation script will add a startup command to your s:user-startup sequence. It will also add an assign for EZCron. You must reboot for the assign to take effect.

After you reboot, EZCrond should be running. There are two ways to check. The easiest is to simply double-click on the EZCron icon in the EZCron: directory, and look at the EZCron Status gadget. It will either say "Active", or "Inactive."

1.5 Manual Installation

MANUAL INSTALLATION

- Create a directory called EZCron.

- Copy the EZCrond program to your EZCron directory.

- Copy the EZCron program and EZCron.info to your EZCron directory.

- Copy the contents of the libs directory to your libs: directory.

- Copy the contents of the s directory to your s: directory (s: is located in sys:s and is assigned by the kickstart ROM).

- Copy rexxc/varexx and vxc to your sys:rexxc directory.

The following directories MUST be present in your EZCron directory.

- Copy the prefs directory to your EZCron directory.

- Copy the rexx directory to your EZCron directory.

The following MUST be added to your s:user-startup file:

```
;Added for EZCron
assign EZCron: [yourdevice]:EZCron
run >nil: rx >nil EZCron:EZCronD Start
;End EZCron
```

If you are running OS3.1 this assign can be like this:

```
;Added for EZCron
assign EZCron: [yourdevice]:EZCron
run <>nil: rx EZCron:EZCronD Start
;End EZCron
```

That should do it. Just reboot and run EZCron when you want to add, edit or remove events.

1.6 Buttons

LISTING OF BUTTONS AND FUNCTIONS

E	Event
A	Arguments
S	Sfx
D	Date
DateRange	Date Range
T	Time
Time Range	Time Range
Add	Add Event
Test	Test Event
Del	Delete Event
Mod	Modify Event
TEdit	Edit with text editor
New	New Config File
Purge	Purge old events
E/D	Enable/Disable Event
MkDef	Make Default Event
Refresh	Refresh List/Status
Clock	Run a clock
Prefs	Set/Change Preferences
Control	Set/Change Preferences

1.7 e

- E (Create a new event)

This is the gadget you use to create a brand new event. This will bring up a requester that asks you what type of event you would like. There are two types:

Program Event:

This is the event type you would select for running external events. An external event could be an executable program, a rexx script or a dos script etc. If you select the program event, it will bring up a file requester for you to choose the program you wish to execute. Make sure a full path is given, or the program may not execute properly! When a program is selected, the CURRENT time and date are filled in the time and date fields. You will need to change these to reflect the time and date you wish your program to run. Also, be sure to TEST the event to make sure it works properly before you depend on the EZCrond (daemon) to start it automatically.

Reminder Event:

When you select REMINDER, it will bring up a string requester that allows you to enter a text string. Once set, this will also automatically set the Event field to TEXTREMINDER. This must remain TEXTREMINDER in order for the program to understand its task. When TEXTREMINDER is selected, the CURRENT time and date are filled in the time and date fields. You will need to change these to reflect the time and date you wish your text reminder to be displayed.

NOTE: You must click the ADD button after creating the event.

1.8 t

- T (Time)

** PLEASE NOTE THAT THIS WORKS ON A 24 HOUR CLOCK **

This gadget brings up a window containing a list of gadgets which allow you to select the time format you wish. The options follow with their explainations:

Once - Run the even only once. ie: 13:22

Minute - Run the event every minute of every hour

Five Min - Run the event every five minutes of every hour.

Ten Min - Run the event every ten minutes of every hour.

Quarter - Run the event at 00, 15, 30, & 45 minutes of every hour.

Thirty - Run the event at 00, and 30 minutes of every hour.

Hourly - Run the event every hour on the hour.

*Special - Run the event via wildcards. The default is set for every hour at a specified minute. The hour place will be replaced with the pounds sign.
ie: ##:33 would run the event every hour at 33 minutes after the hour.

Also, you may manually change the minute field to the wildcard as well.
ie: 15:## would run the event every minute from 15:00 to 15:59.

A third option is to set it for ##:##. This would run the event every minute of every hour. This would be the exact same as selecting 'Minute' in the time field.

1.9 timerange

- Start Time/End Time

These gadgets represent a time range you may want an event to execute within. If you wish your event to run from noon to 3pm you would enter 12:00 in the Start Time gadget and a 15:00 in the End Hr. gadget. Please note that events that run from PM to AM MUST be set up as two seperate events. EZCrond (daeman) works on a 24 hour clock internally. If, for example you wish your event(s) to run from 11pm to 3am you would have to set up your first event to run from 23:00 to 24:00 and then your second event to run from 00:00 to 03:00. I am VERY aware that there is not really a 24 in the 24 hour format, but it has to be done this way in order to work properly. In other words, Midnight has to ALWAYS be entered as 24 when used as an end time and 00 when used as a start time. Of course, you could enter 23:59 as the end time on one event and 00:00 as the start on another.

```
 Wildcard Support

Currently, EZCrond and EZCron does not support wildcards within date ranges.
```

1.10 d

```
 - D (Date)

 This gadget brings up a window containing a list of gadgets which allow
you to select the date format you wish.  The options follow with their
explainations:

   Once         - Run an even only once. ie: 13:22

   Daily        - Run an event every day.

   Monthly      - Run an event one day a month. The actual field would
                  be 'Monthly-00' where '00' is the date. ie. Monthly-14
                  would execute the event every month on the 14th.

   Sun-Sat      - Run an event on the given day of the week.

   Weekdays     - Run an event on Saturdays and Sundays only

   Weekends     - Run the event Mon-Fri only.

   EveryNNDD    - This is a strange one. This option will allow you
                  to run events one given days in weeks of the months.
                  ie: 'Every_3_Fri' would run the event every third
                  friday of every month. 'Every_1_Mon' would run
                  the event every first Monday of a month.

   Range        - This option will allow you run events within a range
                  of dates.

   Special      - This is where it gets fun. This allows you to use
                  wildcards for your event dates.

                  The first field of wildcards represents the month.
                  The next field of wildcards represents the day.
                  The last field of wildcards represents the year.

                  When selected, the date window will contain '##/##/##'.
                  This means that the month, day and year are all set
                  for 'on' and the event would run every day of every
                  month of every year.

                  If you wanted the event to execute every day during
                  March of 1997, if would look like  this: '03/##/97'.

                  If you wanted the event to execute every month on the
                  12th in 1997, you would change it to this: '##/12/97'.
                  This would be the same as if you had used 'Monthly-12'.
```

1.11 daterange

```
  - Date Range

  The date range feature allows you to set a range of dates for a given
 event. If, for instance, you wish to run an event every day from
 January 1, 1997 to March 14, 1997, you would enter 01/01/97 as the
 start date and 03/14/97 as the end date. Note that the event will
 NOT run on the end date specified.

 Wildcard Support

  The date range also supports wildcards. This drastically opens up the
 possibilities of date ranging. By using wildcards, one could set up a
 single event to execute at given times throughout a month, week or year.

  Keep in mind that a range will override any date setting you may have.

  This is difficult to explain so I will use examples to try and make it
 easier to understand.

 Date:      ##/##/97 or Daily
 Startdate: ##/14/97  This would run an event every month of 1997 on the
 Enddate:   ##/18/97  14th through the 18th only.

 Date:      ##/##/97 or Daily
 Startdate: 01/##/97  This would run an event every day thoughout January
 Enddate:   02/##/97  and Feburary of 1997 only.

 Date:      Every_3_Fri
 Startdate: 01/14/97  This would run an event every third Friday from January
 Enddate:   06/##/97  14th through the last day of June.

 Date:      Saturday
 Startdate: 01/14/97  This would run an event every Saturday from January 14th
 Enddate:   05/##/97  through the last day of May.
```

1.12 ed

```
  - E/D (Enable/Disable Event)

 (NEW v1.75+) This switch is the Enable/Disable gadget.  It does just what
 it says, it enables and disables events.  This will add a '!' charactor to
 the beginning of the event name.  The Daemon will not execute events marked
 in this manner.

  To use this feature, simply select the event you wish to enable/disable and
 then click the E/D gadget.  A '!' charactor will be added or removed from
 the event field.  NOTE:  Events may STILL be tested when disabled.
```

1.13 mkdef

- MkDef (Make Default Event)

When EZCron makes a new s:cron.config file, it must always create one event within the new file (EZCrond (daemon) would simply exit if no events existed). EZCron will create a default event of your choice simply by clicking on an event in the listview window, or by creating an event, then clicking the MkDef gadget. You could set the default event to be dated sometime in the past so that the default event never actually runs. Note: If you do make the default event a past date, beware that the Purge gadget WILL delete it.

1.14 a

- A (Event Arguments)

This brings up a ReqTools file requester for adding program file arguments. Be sure that your paths are complete.

If the argument is not a file, you may type the argument in manually in the text field.

1.15 s

- S (Sound Effect/Sample)

This gadget brings up a ReqTools file requester for adding an optional sound effect to be played with a given event. Once again, make sure a full path in included here. The Sample Player Prefs must have some sort of audio player set like playsound, Play16, etc.

1.16 add

- Add (Add New Event/Copy Event)

This gadget allows you to add your new event or copy an existing one. To create a new event, you must add all of your parameters FIRST then click the ADD gadget to add it to the list. If coping an existing event, select the event, then click the ADD button. This will copy the selected event to a new one at the bottom of the list.

1.17 del

- Del (Delete Event)

Deletes the selected event from the list. There is no safty net here, If you accidently click on the DEL button, the event is gone for good.

1.18 mod

- Mod (Modify Event)

This will modify the current selected event. First select the event you wish to modify. Make the necessary changes. Click the Mod gadget.

1.19 purge

- Purge (Purge Event List)

This gadget will delete old, one time events. It will not delete any events that are still active. ie: events containing the fields Daily or Monthly-xx would not be deleted. It will also delete events that had a date range set on them if the range has expired.

1.20 new

- New (New Config)

When clicking this gadget, EZCron will create a new s:cron.config file with a 'dummy' event contained in it. There MUST be at least one event in the config file for the daemon to function (This default event can be set using the AddDef gadget). This event may be a date already passed if you don't wish to be bothered with an active event.

This gadget will copy the current config file to s:cron.config.old and create a new one in its place. If you happen to do this by accident, just delete the newly created config file and rename cron.config.old back to cron.config.

1.21 tedit

- TEdit (Call Text Editor)

This will call a text editor of your choice (defaults to c:ed) and loads the s:cron.config file into it for manual editing of the cron.config file. You may select an editor of your choice in the prefs settings.

NOTE: ADVANCED USERS

1.22 test

- Test (Test Event)

This will test the selected event to make sure everything works just fine before you 'trust' it to work on its own.

1.23 control

- Control (EZCronD Control)

This manually starts and stops the EZCrond (daemon). It simply toggles it on or off. Just remember to start it if events are due to take place. If you used the installer, EZCrond will be started from your startup-sequence automatically.

1.24 refresh

- Refresh

This simply refreshes the events listing and the Daemon control status. You may need this if you edit the cron.config file with a text editor that seperates itself from the parent process, for example.

1.25 clock

- Clock

This gadget will pop open a small AmigaDos digital clock.

1.26 prefs

- Prefs

This gadget takes you to the prefs window where you may set prefs for an audio player and for an optional text editor.

PREFS WINDOW

- Choose Text Editor

This will choose a text editor and saves its settings. For use with the TEdit gadget.

- Choose Sample Player

For adding a sample player to the prefs file. This must be set if you wish to use sound effects with events.

- Set Test Console

When testing an event, by default, a con: window will pop up for debugging purposes. This prefs setting decides the attributes and redirection values of the tested event. This could be changed to prt: if desired and the results would be printed. Also, if you wish for no output whatsoever, you can replace this with an

```
empty string.

 * Clock

 This simply allows you to choose whether you want a digital or
analog clock.

 * cron.config path

 This setting will allow you to put your cron.config file (the one the
daemon reads) in a location of your choice.  The default is t:cron.config.
Note that if you choose a place on your HD, it will spin your drive once a
minute when the daemon 'wakes' up to check its events so the ideal place
for this file is ram:, sd0:, rad:  etc.

* - Designates new additions or changes since previous release.
```

1.27 Tutorials

```
                             TUTORIALS

 A few tutorials are given below to help you understand how to edit and add
events.



ADDING A NEW PROGRAM EVENT:

    Click the 'E' button at the top left. A Requester will pop up asking if
    want a Program or Reminder event.

    Click on PROGRAM. A file requester will then pop up. Find the program you
    wish to run and click ok. The program, current date and current time should
    appear in their respective fields.

    If this event is one that you only wish to run once, click in the date
    gadget and change the date to whatever date you want (remember, the format
    is mm/dd/yy). Otherwise, click on the 'D' button and select and alternate
    selection.

    Next, edit the time field to whatever time you desire, or use the 'T' button
    to select and alternate selection.

    Click the 'ADD' button. The new event should appear in the listing.

ADDING A NEW REMINDER EVENT:

    Click the 'E' button at the top left. A Requester will pop up asking if
    want a Program or Reminder event.

    Click on REMINDER. A string requester will then pop up. Type in the text you
    want displayed and click ok. The reminder string, current date and current
    time should appear in their respective fields. Also, the top event field ↩
       should
    have displayed "TEXTREMINDER"
```

If this event is one that you only wish to run once, click in the date gadget and change the date to whatever date you want (remember, the format is mm/dd/yy). Otherwise, click on the 'D' button and select and alternate selection.

Next, edit the time field to whatever time you desire, or use the 'T' button to select and alternate selection.

Click the 'ADD' button. The new event should appear in the listing as TEXTREMINDER.

COPY AN EVENT:

Click on the event you wish to copy. Click the 'ADD' button. The newly copied event should now appear at the bottom of the event list.

MODIFY AN EVENT:

Click on the event you wish to modify. Make the necessary modifications. Click the 'MOD' button. The changes should now be applied.

DELETE AN EVENT:

Click on the event you wish to delete. Click the 'DEL' button.

CREATE A DEFAULT EVENT:

Click on the event you wish to become the default event. Click on the MkDef button. Now, whenever you create a new config file, this event will be the first one created.

CREATE A NEW CONFIG FILE (s:cron.config)

Click the 'NEW' button. If you already have a config file, it will ask if you want to overwrite it. Select 'YES'. You will now have a new config file with the default event in it (See CREATE A DEFAULT EVENT).

DYNAMICALLY CHANGE CRON.CONFIG FILES

As you can probably imagine, there is a LOT of power in the ability to alter config files on the fly, without being present. There could be many reason to want to do this. One of which is to reduce the size of very large cron.cofig files that only have events which execute at certain times on a daily basis. ie, you have 20-25 events that execute before noon each day, but only 10 events which execute after noon. Huge cron.config files can start to get into valuable processing time each minute if they are too large. By splitting them up, you could speed up this process considerably. This example will only show how to use two config files (not including the master config), but you could actually have as many as you like.

I. Copy the s:cron.config file:
   1. Copy s:cron.config to s:cron.config1.
   2. Copy s:cron.config to s:cron.config2.

II. Edit the s:cron.config1 file:
1. Run your favorite text editor and load s:cron.config1 into it.
2. Delete any lines containing any events that don’t execute until after 12:00PM noon but before 12:00 midnight (excluding hourly events that execute all day).
3. Save the file as s:cron.config1.

III.Edit the s:cron.config2 file:
1. Run your favorite text editor and load s:cron.config2 into it.
2. Delete any lines containing any events that execute after 12:00 AM midnight excluding hourly events that execute all day.
3. Save the file as s:cron.config2.

IV. 1. Add the included script (EZCron:rexx/Swapconfigs.rexx) as a program event. Be sure to use rx as the program and use the script as the argument. The script number (1 or 2) is also a required argument.

2. You should have ’rx’ as the event; ’SwapConfig.rexx 1’ as the argument minus the quotes.

3. Set time date to ’daily’ or ’##/##/##’ Set the time to ’12:00’

4. Click the ADD button to add it to the event list.

5. Click the ADD button again to make a duplicate of the event in the list.

6. Change the argument string to ’SwapConfig.rexx 2’
Change the time string to ’00:00’

Thats it. Now at 12:00 noon and 12:00 midnight, your cron.config files will actually get changed automatically. Also, because of the way the script works, whenever you run EZCron (Event Manager), you will always be editing on the currently active cron.config file. Be careful not to add any events to a config that may not be the active one at the time the event should execute. Also, if you do modify your active config file whenever swapping is active, you MUST copy the current cron.config back over the either cron.config1 or cron.config2.

1.28 Author

EZCron is emailware

EZCron is free. If you like it and use it, I would like to hear from you. Or if you have any suggestions for improvements and new features, I like to hear them. Also, of course, if you find any bugs, I would like to know where they are. My EMail address is:

hines_j@iolinc.net
or
wdtv@wdtv.com

Also, visit my Internet Home Page for the latest version of EZCron and for other interesting stuff.

http://www.iolinc.net/~hines_j

Also, If you are a Star Trek fan, check out the demo of Trek-The Guide. It is a full featured database with pics, sfx and synopsis for each episode with full cross references and misc facts.

It is available on Aminet.

Thanks must go to the following people:

Gene Heskett	(For his huge help in figuring out some of these algorithms. . .and for beta testing)
Mike Herschede	(Another Great Beta Tester)
Karl Swisher	(For some pretty good ideas)
Andy Cook	(Author of VArexx for which the GUI wouldn't have been possible)
Robert Wilson	(For his contribution of his very nice and useful scripts for Miami and YAM).

LEGAL STUFF

1.29 Troubleshooting

TROUBLESHOOTING

EZCROND WILL NOT RUN ON STARTUP:

Make sure that the line 'run >nil: rx >nil: ezcron:ezcrond start' is in your s:user-startup sequence.

Make sure you have a valid s:cron.config file. It will not run without it! A valid cron.config file should contain at least 1 event in the following format:

```
command,args,time,date,startdate,enddate,TimeRangeStart,TimeRangeStop,SFX, ReminderString
```

EXAMPLES:

```
rx,EZCron:rexx/TalkingClock.rexx,hourly,daily,-,-,--,--,-,-
TEXTREMINDER,-,22:00,range,01/01/97,12/31/97,--,--,-,Remember to email the author .
```

EZCRON WILL NOT RUN AT ALL:

Make sure all the included librarys are installed in Libs:.
Make sure that sys:rexxc is in your path. I have found some systems that don’t seem to carry their path over to shell processes too well.

Try running it from a shell. Just type ’rx ezcron:ezcron’
The Shell is actually the best method of trouble-shooting since any EZCron errors will be reported to the shell it was run from.
If run from WorkBench, all error messages are suppressed.

EZCROND (DAEMON) WILL NOT RUN AT ALL:

NOTE: EZCrond can also be started from a shell. Just cd to EZCron: then type run <>nil: rx EZCrond start.

Any errors should be reported to the shell it was started from.

If none of these solutions work, please contact me and describe the problem. My email address is hines_j@iolinc.net.

Make sure you have ARexx installed and running.

1.30 ARexx Scripts

Included ARexx Scripts (and how to use them)

There are a few useful ARexx scripts included that you may find useful. You can find them in the the EZCron/rexx directory. There are auto-online/Getmail scripts for AmiTCP and Thor here as well as for Miami and YAM.

I will list them here and briefly descibe what they do and how to utilize them.

AutoGetMail:

This simple script utilizes both AmiTCP v4.+ and Thor. All it does is start your AmiTCP/bin/startnet script and gets online...then it runs Thor:bin/gettcp and gets your mail and any newsgroups you may be subscribed to. The only setting that is adjustable, and you must adjust this, is the ThorMailBox variable at the top of the script. Just change this to reflect YOUR Thor mailbox.

LWave_EZCron.rexx:

This script will control Lightwave 3.5+ and start rendering a given project when executed. It is even smart enought to figure out which frame to start on if there are previously rendered frames in the directory. There are variables that need to be set at the top of the script and they are pretty much self explainatory.

Reminder.rexx:

This script is used internally by EZCron. It is used to control any textreminders you may have set in EZCronGUI. There are no user varibles to set and this script should be left as is.

TalkingClock.rexx:

This script does exactly what you think....its a talking clock.

First, you MUST have speak: mounted. If you are not familier with Speak:, see you AmigaDos manuals. Also, Speak: is not included with with AmigaOS starting with v2.1.

TalkingClock.rexx by Gene Heskett ©1997

MailCheck.rexx:

This fine script was written by Robert Wilson and is for automatically getting on line with Miami and getting your mail using YAM.
See MailCheck.docs

MailCheck.rexx by Robert Wilson ©1997

1.31 Cron.config: An Overview

Cron.config: An Overview

The cron.config has changed considerably in the past few versions. Old config files ARE NOT COMPATABLE with this new format. You may use this information to modify your existing config file, or just create a new one with EZCron (recommended).

*** FOR THOSE YOU WOULD RATHER MAKE THE CONFIG USING A TEXT EDITOR ***

```
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~cut here~~~~~~~~~~~~~~~~~~~~~~~~~~
; This is the cron.config file. The semi-colon is a comment marker.
; You may use this to remove commands or make comments. You may also
; comment (disable) out events by using the '!' charactor at the head
; of the line. This will allow you to view and modify events that don't
; actually execute.
;
; The structure of the file is as follows and each field is seperated by a comma:
; ---------------------------------------------------------------------------
;  1st entry: The path & program you wish to run
;  2nd entry: The program arguments
;  3rd entry: The time to execute. This may also be set to 'Minute' and 'Hourly'
;  4th entry: The Date to execute. This may also be set to 'Daily', a day of the
;             week such as 'Wednesday', 'Monthy-00' or 'Range'. The Montly must
;             have the hyphen and a two digit number in the same format as written
;             below.
;  5th entry: The start date.
;  6th entry: The end date.
;  7th entry: Time Range start time.
;  8th entry: Time Range end time.
;  9th entry: The Sound FX File to run with the event
; 10th entry: If  event is a textreminder, this field will contain the text.
; ---------------------------------------------------------------------------
; EXAMPLES
; ========
```

```
TEXTREMINDER,-,22:00,range,01/01/97,12/31/97,--,--,-,Remember to email the author.
!TEXTREMINDER,-,23:14,every_4_fri,-,-,--,--,-,This is the 4th friday event
ezcron:rexx/TalkingClock.rexx,-,hourly,daily,-,-,--,--,-,-
!Inet:InetScripts/AutoGetMail,-,15:00,daily,-,-,--,--,-,-
c:execute,amitcp:Start_AmiTCP,12:00,weekdays,-,-,--,--,-,-
System2.1:WBStartup/LineMan,-,hourly,daily,-,-,02:00,22:00,Data:SoundSamples/ ↩
   Sandro_3_4_5.8svx,-
c:play16,Data:SoundSamples/DoubleBeep,quarterly,daily,-,-,--,--,-,-
```

1.32 history

```
                                EZCron

** EZCron is The Graphic User Interface Prefs for EZCrond, ©1994-1997 Jim Hines.
** 12-13-95 Added Speech Function to test subrtn.
** 12-22-95 Added Purge Function.
** 12-24-95 Added multi-windowing on prefs window
** 12-31-95 Added multi-windowing cyclegads on time and date fields
**      Changed the main gui interface window and moved the new config
**      button to the edit screen. Removed the Quik Add button altogether.
** 01-05-96 Added Weekdays, Weekends and Quarterly support.
**      Added full online help subroutine.
** 01-07-96 Added Default Event Variable Handling. Opens prefs file at beginning
**      and rereads it with the 'addefault' command.
**      added online help for def event button.
** 08-25-96 Changes the cron.config file to reside in envarc: and RAM:ENV
**      Great idea Paul. ;)
** 09-02-96 Moved the active config file to ram:t as the env: was causing HD
**      activity with such programs as HDEnv. The cron.config is now BACK
**      in s: but copied to t:
** 09-09-96 Started work on the range stuffs
** 09-10-96 Hourly time range routine added.
** 09-11-96 Fixed a few bugs and tightened the code up a bit.
** 09-18-96 Added the every 5 minutes time field.
** 10-10-96 Fixed a nasty clock bug in the I/O button which caused the GUI to exit ↩
   .
**      This occured after being in the event editor, then exiting to the
**      control panel and then quiting the daemon.
** 01-07-97 Added the ENABLE/DISABLE EVENT function
** 01-08-97 Rewrote the GUI section entirely. Now smaller and faster since the
**      main loop is much much smaller!!!
** 01-09-97 You can now test disabled events. Line 210
** 01-14-97 More speed rewrites: seperated the waitforpkt() from the eventparse()
**      shortening the main loop even more.
**      Added Ten_Min and Thirty_Min to the time fields.
** 01-18-97 Fixed a nasty bug; If you tested a disabled event, then enabled
**      the event, the argument 1 field would get copied to the other 3.
**      TOOK OUT THE DELAYS()
**      Added a check for an empty cron.config file. Used to cause a startup error ↩
   .
** 01-20-97 Fixed a bug in the testevent(). It was lacking the double quotes which ↩
    was
**      causing problems.
** 01-25-97 Reworked the event gadget. It now will ask what type of event you want ↩
   .
** 01-27-97 Fixed a bug in Purge:.
**      Fixed a bug in Prefs:.
** 01-29-97 Started on the Date Range function.
** 01-30-97 DateRange function implemented, but still a little buggy.
** 02-01-97 Now supports unlimited arguments.
**      Now using commas between fields. Old config is no longer compatable
**      as of v1.84e
** 02-04-97 Eliminated the online help system. With the new GUI, its much easier
**          to understand and it was felt that it was no longer needed.
** 02-06-97 ReqTools requesters will now open under the mouse pointer.
** 02-07-97 Rewrote the update() routine. It will no longer put modified events
**          at the bottom of the list.
** 02-07-97 Reduced the deleteevent() and adddefault code majorly.
**      Added a comma after the last field.
**      Cleaned up extra, unneeded copy to t: routines.
** 02-08-97 Blank date range double dashes (--). Now it's a single dash (-).
** 02-09-97 Added the shour to the time menu.
** 02-09-97 added the hour wildcard '##'. ie: '##:15'. AF (advanced feature #1)
** 02-10-97 Renamed EZCronGUI to EZCron
** 02-12-97 Renamed the MAINWINPORT to EZCRON_PREFS.
**      EZCrond can now auto update the daemon status window.
** 02-14-97 Renamed EZCron to EZCrond and EZCronGUI to EZCron.
**          Sorry for the confusion, but they really needed renaming.
**          The ARexx ports were also renamed to EZCROND and EZCRONPREFS.
** 02-16-96 Added a new prefs option: Test Routine Redirection.
**          Changed cprefs to defev.prefs.
** 02-17-97 Public Release v2.00.
** 02-21-97 Change clock to analogue.
** 03-04-97 Added clock prefs.
** 03-05-97 Added cron.config path prefs.
** 03-06-97 Added rexx command quit.
** 03-06-97 Added rexx command copy (Don't ask).
** 03-09-97 Public Release v2.10.
```

EZCronD

```
/* $VER: EZCrond by Jim Hines v1.85 ©1995-1997 All Rights Reserved
** 10-23-95 Added requester to verify making a NEW cron.config file. See newrtn:
** 10-25-95 Fixed Bug in the above and added renaming of the old config file. See ↩
   newrtn:
**      renamed from ACron to EZCron. GUI started on this date too.
** 11-30-95 added the text string reminder support and the sfx functions
** 12-05-95 Fixed case sensitivity problem with weekly, monthly & Daily routines
** 12-13-95 Added Speech Function
** 01-04-96 Added Weekdays, Weekend and Quarterly support functions
** 01-05-96 Fixed a bug in the Weekdays routine.
** 01-11-96 Cleaned up the code a bit
** 01-17-96 Added a new routine.  ie every_3_wed
** 08-25-96 Moved the cron.config file to envarc: / RAM:ENV
**      Good idea Paul. ;)
** 09-02-96 Moved the active config file to ram:t as the env: was causing HD
**          activity with such programs as HDEnv
** 09-10-96 Hourly time range routine added.
** 09-11-96 Fixed a nasty bug in time range and majorly rearranged the timer code.
**          Version bumped to 1.70. Thanks Gene!
**      Finally renamed the rexx port from ACron_Rexx to EZCron_Rexx.
** 09-18-96 Added the every 'Five_Min' time field.
** 10-03-96 Actually made it work (What took me so long?)
** 01-07-97 The daemon now also ignores the '!' charactor as a comment. This is  ↩
   used
**          by EzCronGUI to disable events without removing them from the list.
** 01-14-97 Now supports Ten_Min and Thirty_Min events.
** 01-20-97 Fixed a speak: bug. It was lacking the double quotes which was
**          causing problems.
** 01-25-97 Rewrote the quarterly routine.
** 01-28-97 Fixed a bug I made when I rewrote the quarterly routine.
** 01-30-97 Started work on the date range routine.
**          Changed the TimeRange so it will now accept minutes.
** 02-09-97 added the hour wildcard '##' support. ie: '##:15'. AF1 (advanced  ↩
   feature #1)
** 02-11-97 Added the minute wildcard '##' support. ie '22:##'
**      Added the Date wildcard support. ie: ##/##/## would be = DAILY
**                         ##/11/## would be = MONTHLY-11
**                         02/##/97 would be every day in FEB of 1997
** 02-12-97 EZCrond can now auto update the daemon status window.
**      If present, signals the EZCron (Prefs) program when starting/stoping.
** 02-13-97 Moved the timerange routine below the wildcard routines.
**          Empty startdate and enddate fields are now '##/##/##' instead of '-'
**          Empty starttime (rng1) and endtime (rng2) is now '--:--' rather than  ↩
   '-'.
**          Renamed EZCron to EZCrond and EZCronGUI to EZCron.
**          Sorry for the confusion, but they really needed renaming.
**          The ARexx ports were also renamed to EZCROND and EZCRONPREFS.
** 03-04-97 Reworked the every_##_dd and Monthly routines a bit.
**          Got rid of the void_event check in the wildcard time check.
** 03-05-97 Reworked and shortened considerably the wildcard routines.
**      Added movable cron.config file by reading prefs file.
** 03-09-97 Public Release v2.01.
*/
```