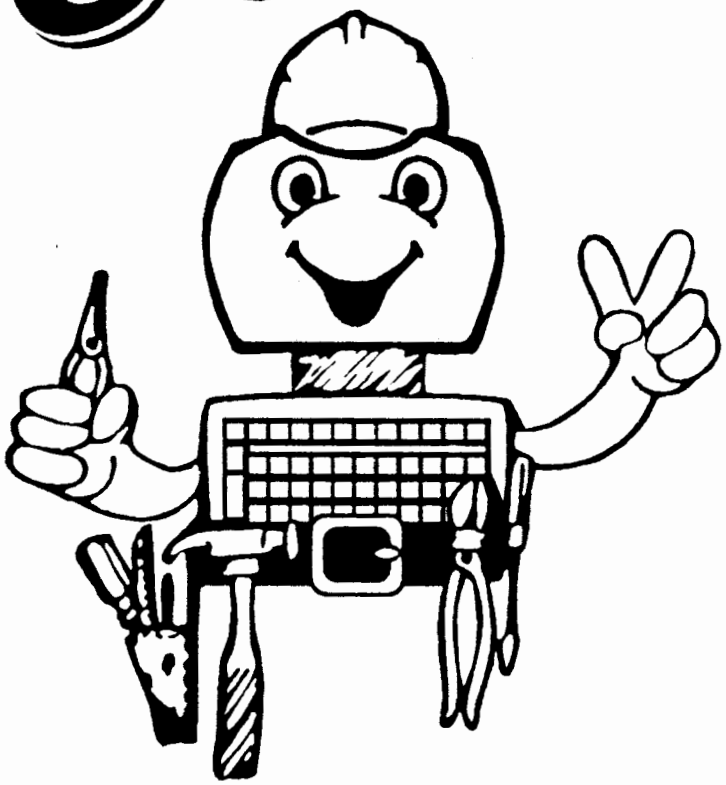
TOOLS II



A set of 27 useful tools for the OS-9 Level II user, written with the sole purpose of making your computing life easier!

We hope that you enjoy your purchase of TOOLS II. Each utility includes its own help system for reminding you of how to use it. To get help on any given command, just type the name of the command without any parameters. If parameters are required, you will be informed of how many and what they mean. If no parameters are required, no help should be needed! The utilities are divided into the following sections:

•Windowing Utilities

These utilities allow you to take advantage of OS9's windowing features.

Brnout	- RAM resident screen saver	Clocker	- RAM resident clock util.
Draw	- General purpose drawing util.	Wmenu	- Full featured menuing
Select	- Window selection util.	Width	- Window width changing

•File Manipulation Utilities

These utilities allow easier use of your OS9 file system.

Bak	- General purpose backup util.	Cdr	- Copy/Delete/Rename
DirCopy	- Directory copy utility	Dname	- Disk naming utility.
Find	- Find a file (somewhere) on a disk.	Go	- Go to any directory easily.

Print - Format and print a file utility.

•File Conversion Utilities

These utilities allow you to make various changes and conversions.

FConvert	- Generic file conversion util.	Fixtabs	- Fix files with TABs.
GSAR	- Global Search And Replace	Search	- Multi-file search utility.

Zero- File purging utility.

•Process Scheduling Utilities

These allow you to take full advantage of OS9's multitasking capabilities.

Alarm	- Have your computer alert you.	At	- Execute AT this time.

Demon - Full featured process scheduling.

•Script File Utilities

These utilities give you more power & flexibility in writing script files.

Ask	- Ask the user a question.	Prompt	- Prompt user for a parameter.

•I/O Port Utilities

These utilities allow you to take full control of your OS9 I/O system.

Autodial	- Autodial your Hayes™ modem.	NetLink	- Network style SCF links.

ReCobbler - Re-save your boot configuration.

•Calculation Utilities

Calc - Full feature command line calculator.

Table of Contents

WINDOWING UTILITIES

Brnout

Syntax: Brnout

Description: Brnout is a utility that will help save your monitor in the event that you leave your computer unattended for some length of time.

When you run Brnout, you are immediately returned to the OS9: command line. Brnout has installed itself and is running in the background. If Brnout is running and there is no keyboard activity for 5 minutes, the entire screen goes black. This will stop an image from being burned into your monitor. To restore the screen, simply press the <CLEAR> key.

To remove Brnout from memory, do a "procs e" to get the Brnout process ID. Then type "kill xx" where xx is the Brnout process ID number.

Notes: Brnout will only work on a COCO 3 system.

Clocker

Syntax: Clocker

Description: Clocker is a utility that displays the current time in the upper right hand corner of the current window. After running clocker, the message "ERROR xxx" will be displayed. xxx is the process number of the clocker program. The ERROR message is the OS9 shell's way of telling you that clocker is still running.

Notes: Clocker requires its own device window. It opens this window via the "/w" window device. If an ERROR 221 occurs it is either because the "/w" module is not in memory or there are no available device windows.

Draw

Syntax: Draw <command> <parameter> <parameter> . . .

The <command> parameter is the name of a COCO drawing command as defined in the OS9 Windowing Manual.

The <parameter> parameters are the data values re quired for the given command. These are defined by the command.

Description: The Draw utility allows you to easily execute OS9 drawing commands from either the OS9: prompt or from within OS9 script files. This utility supports all of the windowing codes defined in the OS9 Windowing manual. These commands are listed below along with their parameters. For more information see the OS9 Windowing manual.

Each command requires a set of parameters to operate. The given drawing command and parameters will be translated into the proper OS9 screen control codes and sent to the screen. If the window is properly set up, this will result in proper actions on the screen. The param eters are usually numeric, but in some cases may be the words "ON", "OFF", "PUT","AND", "OR", and "XOR".

BColor <color #>
Change the current background color.

BoldSw <ON|OFF>
Enable/Disable bold face text.

Border <color #>
Change the current border color.

CWArea <Left> <Top> <Width> <Height>
Change the working area of the current window.

Bar <Xpos> <Ypos>

Draw a bar defined by the line from the current draw position to <Xpos>, <Ypos>.

RBar <Xoffset> <Yoffset>

Same as above with relative coordinates.

Box <Xpos> <Ypos>

Draw a box defined by the line from the current draw position to <Xpos>, <Ypos>.

RBox <Xoffset> <Yoffset>

Same as above with relative coordinates.

Circle <Radius>

Draw a circle at the current position with the given radius.

Ellipse <Width> <Height>

Draw an ellipse at the current position with the given width and height.

FFill

Fill from the current position to an enclosed area.

Line <Xpos> <Ypos>

Draw a line from the current position to <Xpos>,<Ypos>.

RLine <Xoffset> <Yoffset>

Same as above with relative coordinates.

LineM <Xpos> <Ypos>

Same as line, but moves the current position to <Xpos>,<Ypos>.

RLineM <Xoffset> <Yoffset>

Same as above with relative coordinates.

Point <Xpos> <Ypos>
Draw a point (pixel) at <Xpos>,<Ypos>.

RPoint <Xoffset> <Yoffset>
Same as above with relative coordinates.

PutGC <Xpos> <Ypos>
Put the graphic cursor at the specified position.

SetDPtr <Xpos> <Ypos>
Change the current position to <Xpos>,<Ypos>.

RSetDPtr <Xpos> <Ypos>
Same as above with relative coordinates.

Examples:

```
display c
load draw
draw setdptr 320 100
draw line 0 0
draw line 320 0
draw line 639 0
draw line 639 100
draw line 639 191
draw line 320 191
draw line 0 191
draw line 0 100
draw setdptr 160 50
draw box 480 150
```

DefColr

Reset all palettes back to their original default colors.

DfnGPBuf <Group> <Buffer #> <Length>

Define a get put buffer with the given group number, buffer number, and length.

DWEnd

End the current device window.

DWProtSw <ON|OFF>

Enable/Disable the device window protection (disallows device window overlaying).

DWSet <Type> <Left> <Top> <Width> <Height> <Foreground><Background> <Border>

Define a new window's parameters.

FColor <Color #>

Change the current foreground color.

Font <Group> <Buffer #>

Change the current font to the given group and buffer number.

GCSet <Group> <Buffer #>

Set the current window's mouse cursor to the given group and buffer numbers.

GetBlk <Group> <Buffer #> <Left> <Top> <Width> <Height>

Get the rectangle given by the given parameters and store it into the given group/buffer.

GPLoad <Group> <Buffer #> <Format> <Width> <Height> <Numbytes> <Byte> <Byte> . . .

Load a get/put buffer with the given byte data in the format defined by <Format>.

KilBuf <Group> <Buffer #>
Kill a get/put buffer (remove it from memory).

LSet <PUT|AND|OR|XOR>
Define the current drawing mode.

OWEnd
Close an overlay window.

OWSet <SaveSwitch> <Left> <Top> <Width> <Height> <Foreground> <Background>
Open an overlay window with the given parameters.

Palette <Palette #> <Color Value>
Change the given color palette # to the given color value (0-63).

PropSw <ON|OFF>
Enable/Disable the proportional print spacing.

PSet <Group> <Buffer #>
Change the get/put buffer used for all drawing commands.

PutBlk <Group> <Buffer #> <Left> <Top>
Put the specified get/put buffer onto the screen.

ScaleSw <ON|OFF>
Enable/Disable automatic scaling (to 640/200).

Select
Change to a different window. Use redirection to decide which window.

TCharSw <ON|OFF>
Enable/Disable the transparent character switch.

Arc3P <Width> <Height> <X1> <Y1> <X2> <Y2>
Draw an arc defined by <Width> and <Height> and clipped by the line from x1,y1-x2,y2.

Wmenu

Syntax: Wmenu [-c] -m] <MenuFile>

The optional -c parameter specifies that the screen should be cleared before the menu is displayed.

The optional -m parameter specifies that the menu window should be closed before every menu option.

Description: The Wmenu utility allows you to create sophisticated menus to execute any OS9 program. The menu is defined by a menu file. This file is a text file with all the information needed to create the menu.

The WMenu program will read the file, then open an overlay window and display the menu choices on your screen. The user can use the arrow keys to select the appropriate menu choice, or he may press an number key to directly access a menu option. When an option is selected, all of the commands defined for that option will be executed.

The menu file is defined as follows:

<menu title>
<X> <Y> <Width> <Height> <Fgrn> <Bgrn> <Border>
<Options>
<Menu Option 1>
<Option 1 Commandline 1>
<Option 1 Commandline 2>
...
<Menu Option 2>
<Option 2 Commandline 1>
<Option 2 Commandline 2>
...

The first line is the menu title. This title will be displayed at the top of the menu's window.

Example:
OS9 System Menu

The Second line defines the menu parameters. These are as follows:
<X>- X Position of upper left corner of menu window.
<Y>- Y Position of upper left corner of menu window.
<Width>- Width of menu window
<Height>- Height of menu window
<Fgrn>- Menu Foreground Color
<Bgrn>- Menu Background Color
<Border>- Menu Border Color
<Options>- The number of menu options in the file.

Note that the height of the menu must include 2 spaces for a border as well as 2 spaces for "padding" around the menu options. If there is not enough space for all menu options on the screen, the options will all appear on the same line.

Example:
20 6 40 12 0 2 4 10

The rest of the menu file defines each of the menu options. Each option consists of an option name and some number of commandlines to execute for that option.

Option names can be any text, but must not be wider than the width of the menu.

Example:
2> List a file

Option commandlines follow the menu option that they are to be executed for, and are indented at least one space. Each of these commandlines can contain any valid OS9 commandline, plus some additional special codes.

Example:
1> List current directory
dir

One special code is the '%' character. You can use the '%' character to enclose a prompt. When the menu is executed, the user will be prompted for the information. For more information, see the description of the "Prompt" command.

Example:
1> List a directory
dir %Enter pathname%

Another special code is the ':' character. If you BEGIN a commandline with a ':', the menu program will termi nate with that program. A ':' on a line by itself will simply terminate the menu program without executing any other programs. One major use for the ':' character is to change from one menu to another. You can use the ':' character with the "WMenu" command to have one menu call another.

Example:
9> Exit
:echo Goodbye

The following is an example of a complete menu file. You can use this example to create other menus.

```
OS9 System Menu
20 4 40 15 0 2 4 6
1> List current directory
dir
2> Change current directorychd
%Enter new directory%
```

```
3> List a file
list %Enter file to list%
4> Copy a file
copy %Enter source file% %Enter dest file%
5> Delete a file
del %Enter file to delete%
6> Exit

:echo Goodbye.
```

Select

Syntax: Select <window> </1

The <window> parameter is the name of the window to select.

Description: The Select command allows a window different from the current window to be selected. This is most useful inside of script files, where a new window can be created and selected. It can also be useful when a window cannot be reached with the clear key. This can sometimes happen when running terminal programs in a window.

Notes: The </1 is only required if Select is used from within a script file.

Width

Syntax: Width <80/40>

Description: The width command allows you to easily change the working width of your current window. The width command will change the current window type to 40 or 80 columns, depending on what parameter is given.

Notes: The width command always makes the window a full screen window with a black background and a white foreground.

FILE MANIPULATION UTILITIES

Bak

Syntax: Bak <sourcedir> <destdir> [-r]

The <sourcedir> parameter specifies the source directory of files to back up.

The <destdir> parameter specifies the destination directory to which the files will be backed up.

The optional -r parameter specifies whether or not Bak will backup sub-directories.

Description: The Bak utility allows you to backup one directory to another (or one device to another). Bak will optionally recurse sub-directories.The procedure Bak uses to backup files is as follows:

1. Bak first reads the source directory for the name of a file to back up.
2. Bak then checks the destination directory for the same file.
3. If the file does not exist in the destination directory it will be copied.
4. If the file does exist, the file dates are checked and the most recent file is used, i.e. if the source file date is later than the destination file date, the file is copied.

If the -r parameter is used, Bak will detect each sub-directory as it is backing up files and back up the files in the sub-directory as well.

Notes: The copy command must be in the current execution directory.

Cdr

Syntax: Cdr

Description: Cdr (Copy/Delete/Rename) is a utility that allows the easy copying, deleting and renaming of files (the three most-performed file operations).

Upon execution of Cdr, a list of the files in the current directory will be displayed. Under this list is a menu that looks like the following:

<x>change dir <q>uit <c>opy <d>elete <r>ename <l>ist a file:

If you press 'x' you will be prompted: "Enter new directory:" upon which you can enter the name of a new directory. After typing the new directory name and pressing <ENTER>, that directory will become the current directory and a list of the files in that directory will be displayed.

If you press 'q' you will be returned to the OS9: prompt.

If you press 'c' you will be able to copy files. You will first be prompted "Source filename:". At this prompt, type the name of the file to copy. After entering the source filename, you will be prompted "Destination filename:" at which point you can enter the name of the new (copied) file. After pressing <ENTER> the file will be copied.

If you press 'd' you will be able to delete a file. You will be prompted "Enter filename:" at which you can type the name of the file you wish deleted. After pressing <ENTER>, the file will be deleted.

If you press 'r' you will be able to rename a file. First you will be prompted "Source filename:" at which you can type the name of the file that you want to change the name of. You will then be prompted "Destination filename:" at

If you press 'l' you will be able to list a file. You will be prompted "Enter filename:" at which you can type the name of the file to list. After pressing <ENTER>, the file will be listed to your screen.

After each of these commands is executed, the files in the current directory as well as the menu prompt are re-displayed.

Notes: Note that the "dir", "copy", "del", "rename", and "list" commands must be in the current execution directory (CMDS). If they are not, an ERROR 216 may occur.

Note that when prompted for filenames, a full OS9 pathlist may be given so that files can be copied, deleted, and renamed from other than the current directory.

DirCopy

Syntax: DirCopy <sourcedir> <destdir>

The <sourcedir> parameter is the name of the source director from which to copy.

The <destdir> parameter is the name of the destination director to copy the source directory into. If the destination directory doe not exist it will be created.

Description: The DirCopy command will copy an entire directory to another directory.

All files in the directory specified by <sourcedir> will b copied into the directory specified by <destdir>.

Notes: The copy command must be in the current execution directory.

Dname

Syntax: Dname <devname> "<name>"

The <devname> parameter is the name of the drive in which the disk to change the name of is. The <name> parameter (enclosed in double quotes) is the new name to give the disk.

Description: The Dname command allows you to change the name of a disk without re-formatting the disk.

Find

Syntax: Find <filespec> [<path>]

<filespec> is a wildcard file specification to look for.
<path> is an optional starting directory in which to look.

Description: The Find utility will find a file on a disk or in a given directory (or sub-directory).

The filespec is a wildcard file specification to be matched.

The optional path, if specified, will restrict the search to only that directory and any of its sub-directories.

Find will display a list of all files that match the given file specification. If you wish to terminate the search after a single match has been found, press the <BREAK> key.

Go

Syntax: Go <directory alias>

The <directory alias> parameter is an alias name that you give to an actual directory name.

Description: The Go command allows you to set up an alias list for directory names, and change to those directories easily. Go opens the file "/dd/sys/go.file" and searches that file for

the given alias. If the alias name is found, a "chd" is performed to the corresponding directory. If the alias entry is not found, you are allowed to add the new alias to the file for later use.

The "/dd/sys/go.file" file contains entries like the following:

source /d0/source
BASIC09 /d0/source/basic09

You can edit this file with any editor and add your own entries. You can also add entries by calling go with an entry that doesn't exist. If an alias is given to Go that is not in the file, you will be prompted:

Alias not found...Do you want to add it?

If you enter 'Y' to this prompt, you will be prompted for the name of the directory to associate with this alias.

Print

Syntax: Print [-l <pagelen>] [-p <port>] [l <pagelen>] [-b] [-a] [-n] <filename> <filename> ... [&]

The optional -l <pagelen> parameter allows you to specify the length of the printed page. After this number of lines is printed, the page will be ejected from the printer. The default page length is 64 lines.

The optional -p <port> parameter is the name of the port to print to. If the name of a port is not given, /p is assumed.

The optional -b parameter specifies that Print should eject a page BEFORE each file is printed.

The optional -a parameter specifies that Print should eject a page AFTER each file is printed (the default).

The optional -n parameter specifies that Print should not eject a page before or after each file.

The <filename> parameter(s) are the names of the files to print. The ampersand (&) can be used to send printing to the background.

Description: The Print command allows you to print one or more files to the printer (or any port) with proper page break formatting.

FILE CONVERSION UTILITIES

FConvert

Syntax: FConvert [-t] <inpat> <outpat> <file> <file> ...

The optional -t parameter allows you to run FConvert in test mode only. In this mode, the changes will not be saved to your file.

The <inpat> parameter is an input pattern matching string to be searched for inside of each file. The syntax of a pattern matching string is explained in detail below.

The <outpat> parameter is an output pattern matching string to be used as a replacement for the input pattern. Everywhere that the input pattern is found in the file, it will be replaced by this output pattern.

The <file> <file> ... parameters are a list of files to which the convertions will be done.

Description: The FConvert command will convert a given list of files replacing the input pattern <inpat> with the output pattern <outpat>.

Patterns may contain either text or hexadecimal numbers. Text can be any alphnumeric text such as "dog" or "cat", etc. The text does not have to be enclosed in quotes. If quotes ARE used, the entire pattern must be enclosed inside of the quotes. Hexadecimal numbers can be entered into the pattern matching string by prefixing the hexidecimal number with a '$' character. The hexidecimal number must be exactly two digits. If only one digit is needed it must be preceeded with a 0 (as in $03). If you would like the '$' character to actually appear in your pattern match, use the hexidecimal equivalent $24.

A pattern match may also be empty, to signify this use two double quotes enclosing nothing ("").
NOTE: Empty patterns are only legal for <outpat>.

Examples: **FConvert dog moose myfile.txt**

Convert all instances of "dog" to "moose" in"myfile.txt"

FConvert $0d$0a $0d msdosfile.txt

Convert all carriage return/line feed pairs to carriage returns. This is a conversion often needed on text files downloaded from MSDOS systems.

FConvert $09 "" tabsfile.txt

Convert all tabs in the input file to nothing (strip them out). Note that double quotes enclosing nothing can be used to signify an empty pattern.

FConvert 87cd cd87 modsfile

Convert all 87cd pairs to cd87 pairs. Since an $87cd pair is used to mark the beginning of a module file, this would corrupt all module headers in the file.

FConvert cd87 87cd modsfile

Convert all $cd87 pairs to $87cd pairs. This will un-corrupt the file corrupted with the above conversion. Note how this can be used to disable and enable modules in a module file.

Fixtabs

Syntax: Fixtabs <filename> <tabsize>

The <filename> parameter is the name of the file for which to fix the tabs.The <tabsize> parameter is the number of spaces between tab stops.

Description: The Fixtabs command will process a file and replace tabs with the appropriate amount of spaces to make all tab stops align properly.

Notes: Fixtabs does more than just convert every tab to <tabsize> spaces, it calculates the number of additional spaces required to bring the text to the correct tabstop.For example,

suppose the file contained two spaces and a tab such as "<spc><spc><tab>", and the <tabsize> is given as 8. The tab character would be converted into 6 spaces to compliment the other two spaces.

Gsar

Syntax: Gsar [-i] [-p] <search text> <replace text> <filespec> [<filespec> ...]

The optional -i parameter informs Gsar to ignore case differences between the search string and the file text.

The optional -p parameter informs Gsar not to prompt you for each replacement.

The <search text> is any text to search for in the file. If the search text contains characters that may interfere with the operation of the command line (such as (;) (&) (<) (>) etc., double quotes can be used to delimit the text.

The <replace text> is any text with which you would like the <search text> replaced. As with <search text> it may be delimited with double quotes if necessary.

Description: The Gsar command is a Global Search And Replace (G.S.A.R.) utility. It will replace all instances of <search text> with <replace text> in the files specified by <filespec>. Note that filespec may contain the wildcard characters '*' and '?'.

Examples: Gsar "dog" "cat" animals.*
Gsar "hit me" "smack me" painful_?.txt

Search

Syntax: Search [-c] <text> <filespec> [<filespec> ...]

The optional "-c" parameter can be used to disable case sensitivity. If case sensitivity is disabled, matches will occur whether the text is upper or lower case.

The <text> parameter specifies the text to search for. If the text contains special characters (such as ; or & or <SPACE>) it can be enclosed in double quotes (").

The <filespec> parameter(s) specify what files to check. This file specifications can include wildcard characters '*' and '?'. The '*' character represents any string of characters in a filename, and the '?' character represents any single character.

Description: The Search command allows you to search for text in a group of files.

The Search command will search in each of the files that matches each of the <filespec>s for the given <text>. If the text is found, the line on which the text was found will be displayed along with the line number and the name of the file in which it occurred. The text that was found will be highlighted so that it can be easily seen.

Zero

Syntax: Zero <filename>

The <filename> parameter is the name of the file to make 0 length.

Description: The Zero command will change the size of a given file to 0. Often it is necessary for a file to be stripped of all data, but not completely deleted. This is usually useful when a file accumulates data from day to day, but occasionally needs to be cleared. This command will make the given file's size 0, but will not delete the file.

PROCESS SCHEDULING UTILITIES

Alarm

Syntax: Alarm <count> <tone>

The <count> parameter determines how many "rings" the alarm gives. Each "ring" consists of five short bursts of <tone>.
The <tone> parameter determines the pitch of the alarm bursts.

Description: The Alarm command gives an audible alarm defined by <tone>. Both <count> and <tone> should be given in the range of 0-255.

Notes: This command can be used in conjunction with "At" to alert yourself to some event while you're working on your computer (like maybe its time to go to bed!!).

At

Syntax: At YY/MM/DD HH:MM <commandline> &

The YY/MM/DD HH:MM is the date/time specification of when to execute the commandline.

The <commandline> is any valid OS9 commandline that would normally be typed at the OS9: prompt except that semicolons (;) are not allowed.

Description: The At command allows you to execute programs later (unattended). You specify the time to execute the program in standard Year/Month/Day (YY/MM/DD) Hour:Minute (HH:MM) notation. Hours are given in 24 hour notation (i.e. 10:00PM = 22:00 and Midnight = 00:00).

If the date is given as 00/00/00, the commandline will be executed at the given time EVERY day.

Notes: If you do not use the ampersand (&) at the end of the command, it will hang your current terminal (or window) until the specified time, when the command will be executed. Pressing <BREAK> will return you to you prompt, but will NOT schedule the commandline for execution. At uses very little CPU time, as it only wakes up about once every 60 system ticks (about 1 second). This is very little as far as CPU time is considered.

Demon

Syntax: Demon [-s] [-d] [-l] [-z] [-k] [-a]

Description: The Demon command provides you with a sophisticated background task manager (demon). A demon program allows you to schedule programs to be executed at some later time. Demon allows you to do this as well as allowing you to have a program executed over and over again at a given time.

The -s parameter informs Demon that you want to schedule a task. After the -s parameter, you may enter the date, time, and event to schedule.

The date is specified in YY/MM/DD format. Two digits for year, month, and day.

The time is given in HH:MM format. The hour (HH) is a number between 00 and 23 (00=midnight, 23=11pm). The minute (MM) is a number between 00 and 59.

The event is any valid OS9 commandline (what could be typed at the OS9: prompt). This includes OS9 script files as well as OS9 commands.

If you do not enter these parameters on the command line you will be prompted for the information. If you enter zeros for parts of the date and time, the task will be executed for ALL dates and times that match without the zero parameter.

Example:

Demon -s 00/12/25 08:00 Alarm 10 100
Execute the alarm command every christmas morning at 8am.

Demon -s 00/00/01 17:00 Alarm 10 100
Execute the alarm command on the first of every month at 5pm.

The -d parameter allows you to delete an entry that was previously scheduled. After entering this option a list of the current entries and their ID numbers will be listed. You will be asked which ID number to delete. That entry will then be deleted.

The -l option will display a list of all currently scheduled tasks.

The -z option will zap (delete) all current entries in the table.

The -k option will kill the Demon background processor. This will stop all tasks from being executed, but will not delete the tasks from the task table. You can use the -a option to start tasks executing again.

The -a option will enable the Demon background processor. This will cause all tasks to be loaded from disk if they are not already in memory, and be executed as scheduled. Use Demon with this option in your startup file to load and run previously scheduled tasks. You can also use this option after using the -k option to disable the background processor.

Notes:

The Demon command consists of 3 parts, the "Demon" command, the "Demonite" background processor, and the "DemonDat" data module.

The Demon command is the only command that you should deal with. The Demon command allows you to schedule tasks, execute the background processor (demonite), delete tasks, etc.

The Demonite background processor is a program that runs in the background and periodically checks to see if any tasks need executing. If they do, Demonite will execute them. You should NEVER execute or kill demonite by yourself. This could cause the system to be placed in an improper state, and you would have to recopy the DemonDat module from your master disks. To execute and kill the Demonite background processor, use the -a and -k options of Demon. NOTE: Demonite is automatically executed when you schedule your first task.

The DemonDat module is a data module that contains all of the information for all scheduled tasks.

All three of these modules must be in the current execution directory when you run "Demon".

Tasks will only be executed during the exact minute that they are scheduled. If background processing is disabled, or the computer is turned off when a task is scheduled, that task will be deleted the next time Demon is executed and will never get run.

SCRIPT FILE UTILITIES

Ask

Syntax: Ask "<question>" <yes command> : <no command> </1

The <question> prompt is a prompt that will be displayed to the user. Make sure to put something indicating that either a Y or N should be pressed.

The <yes command> is an OS9 command or script file to execute if the user presses Y in response to the question.

The <no command> is an OS9 command or script file to execute if the user presses N in response to the question.

Description: The Ask command allows you to ask a user a question, then execute a different command based on their yes or no (y or n) response.

Examples: Ask "Delete the files? (Y/N):" del file1 : echo File NOT deleted.</1

Ask "Enter [Y]es or [N]o:" echo yes : echo no </1

Notes: The </1 at the end of this command is only required if the command is used from within as script file.

Prompt

Syntax: Prompt<sub commandline>%<prompt>%<sub commandline></1

The first <sub commandline> is the first part of the commandline that you specify.

The <prompt> enclosed in percent signs (%) is a prompt that you give to the user. Whatever the user types will replace the %<prompt>% in the actual command line.

The second <sub commandline> is the last part of the commandline that you specify.

Description: The prompt command allows you to let the user substitute some part of a commandline himself (presumably from within a script file).

Examples: prompt dir %Enter pathname to list:% </1

Prompts the user for a pathname, then does a dir on the pathname entered.

prompt copy %File to save:% /dd/sys/backupfile </1

Prompts the user for a file to save, then copies that file to "/dd/sys/backupfile"

Notes: The </1 is only required if the prompt command is used within a script file. This will, however, be its biggest use.

I/O PORT UTILITIES

Autodial

Syntax: Autodial [-p] [-t <timeout>] [-r <prefix>] <number> [<port>]

The -p optional parameter tells autodial to use pulse dialing instead of the default tone dialing.

The -t optional parameter allows you to specify a time-out delay after which to re-try the number. If none is specified, a defaut of 30 seconds is assumed.

The -r optional parameter allows you to specify a your own dialing prefix for your modem. If you do not specify your own, either ATDT or ATDP (depending of whether -p was used) will be used.

The <number> parameter is the phone number to dial. This number can have embedded parenthesis and dashes as well as special characters recognized by your modem (see your modem manual for details on its dial strings).

The <port> parameter is the name of the port to use. If you do not specify what port to use, /t2 will be used.

Description: The autodial utility allows you to have the computer automatically dial your Hayes compatible modem.

The auto-dial utility will dial the number, then pause. If the person answers the phone, pick up your phone and hit the <ENTER> key to stop the program. The modem will release the line and you can talk to the person.

If the line is busy or has no answer, autodial will continue re-dialing the number (this is what is really useful). When you hear their phone ringing, then proceed as described above.

If you wish to have the number re-dialed immediately, press the 'R' key while autodial is counting down.

Notes: The port given by the <port> parameter must be "Xmode"ed to the proper baud rate before the autodial utility is used. For a 1200 baud modem the command is "Xmode <port> baud=3", for 2400 baud the command is "Xmode <port> baud=4".

Netlink

Syntax: NetLink [<port1>][<port2>]

The <port1> and <port2> parameters are OS9 device names. They can be any valid OS9 character based port (no disk drives). If only 1 port is specified, the current port becomes port 2. If no ports are specified, NetLink goes into interactive mode. The interactive mode is explained below.

Description: Netlink allows you to logically link two OS9 ports to create a network style link.

What NetLink does is take any input from <port1> and send it to <port2> and vice versa. The transfer of data in this fashion can be very useful. Here are some examples:

Suppose you have another computer connected to this one via a NULL modem to your serial port /t2. You also have a printer connected to this computer's printer port /p. You could type 'NetLink /t2 /p &'. This would allow you to use the printer connected to this computer with your other computer!

For another example, suppose you have another computer connected to this computer's port /t2, and a modem connected to port /t1. Now you call into this computer from a remote location, but want to access your other computer. You could simply type 'NetLink /t2' and you

would instantly connect with the other computer. (Note that some BBS software or Tsmon must be running on both systems to make connection).

The NetLink interactive mode can be entered several ways. If you don't specify any ports, NetLink goes interactive automatically. If you are already communicating with a port, you can type 3 minus signs (-) VERY SLOWLY to enter interactive mode. Note that these minus's must be typed VERY SLOWLY.

Once in the interactive mode there are 3 commands, 'C' for connect, 'Q' for quit, and '?' for help. The 'C' command requires one parameter, the port to connect to (Example:C /t2). Help and quit require no parameters.

ReCobbler

Syntax: ReCobbler <devname>

Description: The ReCobbler command allows you to easily save modifications to your current OS9 system.

The <devname> parameter is the name of the device on which the current bootfile (the last one booted from) is saved.

The ReCobbler command can save any system modification that does not effect the size of modules in memory. This includes Xmode, Dmode, debug, and modpatch.

You can use ReCobbler after using Xmode and Dmode, to make those new device parameters permanent.

You can use ReCobbler after using Debug or modpatch to save the patches in memory back to your OS9 Boot file.

Technical Description:

ReCobbler opens the bootfile on the given device (<devname>/OS9Boot) and checks each module. For each module in the boot file, it links to the corresponding module in memory. If the two modules are the same size, the one in memory is used to overwrite the one on disk. This overwriting is done in the same place on the disk. If the module sizes do not match, a message is displayed and ReCobbler moves on to the next module.

Notes:

When re-saving modules to your boot disk, the message "Module is not sharable!" will be displayed for the Init module. This is simply letting you know that patches to the Init module CANNOT be re-saved.

It is possible to ReCobbler to a different boot than the one that is currently in memory, but this is ill advised. If module sizes differ (from in memory and on disk) they will not be re-saved.

CALCULATION UTILITIES

Calc

Syntax: Calc "<expression>"

The <expression> parameter is some "in-fix" notation expression to calculate.

Description: The Calc utility is a command line calculator that you can use to calculate an expression.

The Calc utility supports addition (+), subtraction (-), multiplication (*), division (/), parenthesis '()', ASCII conversion ('), and hexidecimal conversion ($).

Examples:

Calc "(5 + 3) * 2"	Result: 16
Calc "5 + 3 * 2"	Result: 11
Calc "$ff"	Result: 255
Calc " 'A"	Result: 65
Calc "$ff + 'A"	Result: 320

Notes: The double quotes can be omitted if the expression does not contain any offending characters. Offending characters include spaces and quotes ('). For best results, always enclose the expression in double quotes (").

Copyright Notice

COCOPRO! 1334 BYRON AVENUE
YPSILANTI, MI 48198
(313)481-DAVE(3283)