OS-9 for 68K Release Notes

Version 1.0

www.radisys.com

World Headquarters
5445 NE Dawson Creek Drive • Hillsboro, OR
97124 USA
Phone: 503-615-1100 • Fax: 503-615-1121
Toll-Free: 800-950-0044

International Headquarters
Gebouw Flevopoort • Televisieweg 1A
NL-1322 AC • Almere, The Netherlands
Phone: 31 36 5365595 • Fax: 31 36 5365620

RadiSys Microware Communications Software Division, Inc.
1500 N.W. 118th Street
Des Moines, Iowa 50325
515-223-8000

Revision A
August 1999

Copyright and publication information

This manual reflects version 1.0 of OS-9 for 68K. Reproduction of this document, in part or whole, by any means, electrical, mechanical, magnetic, optical, chemical, manual, or otherwise is prohibited, without written permission from RadiSys Microware Communications Software Division, Inc.

Disclaimer

The information contained herein is believed to be accurate as of the date of publication. However, RadiSys Corporation will not be liable for any damages including indirect or consequential, from use of the OS-9 operating system, Microware-provided software, or reliance on the accuracy of this documentation. The information contained herein is subject to change without notice.

Reproduction notice

The software described in this document is intended to be used on a single computer system. RadiSys Corporation expressly prohibits any reproduction of the software on tape, disk, or any other medium except for backup purposes. Distribution of this software, in part or whole, to any other party or on any other system may constitute copyright infringements and misappropriation of trade secrets and confidential processes which are the property of RadiSys Corporation and/or other parties. Unauthorized distribution of software may cause damages far in excess of the value of the copies involved.

August 1999
Copyright ©1999 by RadiSys Corporation.
All rights reserved.

EPC, INtime, iRMX, MultiPro, RadiSys, The Inside Advantage, and ValuPro are registered trademarks of RadiSys Corporation. ASM, Brahma, DAI, DAQ, MultiPro, SAIB, Spirit, and ValuePro are trademarks of RadiSys Corporation.

DAVID, MAUI, OS-9, and OS-9000, are registered trademarks of RadiSys Microware Communications Software Division, Inc. FasTrak, Hawk, SoftStax, and UpLink are trademarks of RadiSys Microware Communications Software Division, Inc.

† All other trademarks, registered trademarks, service marks, and trade names are the property of their respective owners.

Table of Contents

Chapter 1: Enhanced OS-9 for 68K V1.0 Release Notes

Enhanced OS-9 for 68K represents a new and improved method of providing system software, languages, tools, communications software, graphics, system enhancements and selected board-specific software in a single product. Enhanced OS-9 for 68K Version 1.0 includes Version 3.1 of the operating system and board level solutions.

These release notes cover the changes made to the OS-9 operating system since the release of OS-9 V3.0, as well as recent changes to the various OS-9 components.

The following sections are included in this chapter:

RadiSys.

MICROWARE SOFTWARE

Enhanced OS-9 for 68K 1.0

The following sections discuss various features in Enhanced OS-9 for 68K for this release.

General Improvements

- \The SoftStax TCP/IP stack is now significantly more efficient than the previous release.
- The OS-9 resident utility set has been improved.
- An RBFTL add on, which uses AMD series C/D PCMCIA flash cards, was tested and set up for 4MB flash card configurations. Custom configurations are possible depending on the part to be used.

Added Features

The following features have been added and introduced to Enhanced OS-9 for 68K for this release:

- The ability to partition SCSI hard disk drives with capacities in excess of 4GB is now available.
- Support for the 68328ADS (Dragonball) has been added to the OEM package.
- An example port for the 68360 has been added to the OEM package.

For More Information

See **Appendix A: Example Port for M68360 QUADS Boards** for more information.

- Hawk has been integrated into the package for the first time. Hawk has received numerous fixes, and exception violation problems have been solved (System state debugging is supported on the 68328ADS Dragonball only).
- Example PCMCIA support (for the 68328) has been included in the OEM pack.
- Example MAUI (graphics) support is now included (for the 68328) in the OEM pack.
- Example Power Management has been introduced (for the 68328) in the OEM package.

Removed Features

- Support for backward compatibility with the Kernigan & Ritchie C compiler has been removed. Existing OS--9 for 68K users are still able to use the `-bc` option in Ultra C, if required, by copying `math.l, cio, math` etc. into this package. New customers cannot compile code in backward compatibility (non-ANSI) mode.
- ISP (Internet Support Pack) has been replaced by SoftStax.
- The print spooler system (`spl`) has been removed.
- RBSCCS support has been removed. RBSCCS was used with RBF Editions less than 65, which only supported 256-byte sectors.

Optional Extras

- SNMP
- TrueFFS (FLASH file system)
- ISDN support

Chapter 2: OS-9 Operating System V3.1 Release Notes

This chapter discusses the issues relevant for version 3.1 of OS-9. The following sections are included:

RadiSys.
MICROWARE SOFTWARE

System Module Updates

The following sections explain updates to certain modules since the release of OS-9 version 3.0.3.

Kernel

- The error in FPU type code for 68060 processors has been fixed (It had been returning 60 instead of 68060).
- The problems in the interrupt polling system have been fixed:
  - When calling an interrupt service routine, `d0.b` did not contain the vector address for the second and subsequent routines called on a vector.
  - If excessive unknown interrupts occurred, kernels that did not have a Master Stack Pointer (MSP) i.e. 68000, 68010, 68060, 68070, CPU32 could crash.
- Problems with excess stack information within exceptions have been fixed.
- `F$FIRQ` now enforces the reservation of `d1.b`. It must now be zero.
- System-state time-slicing is now disabled when calling the clock driver.
- The alarm system has been revised to make it more robust, especially for cases where an alarm is deleted while active.
- Support for the 68040 has been moved into its own directory from the 68020 directory for this release.

System Security Module (SSM)

The following enhancements and fixes were made to the OS-9 for 68K System Security Module (SSM) for the 68040 and 68060 microprocessors. The new flags are set as bit fields of the configuration define for the `M$SysConf` entry in the INIT module.

SSM

- It is now possible to use system state page tables. This feature is selected by setting the INIT module flag `SSM_SysPT`. Tables are generated based on the systems cache list and on the value in DTT0 (1=Use System State page tables, 0=Use only transparent translation registers).
- The ability to use either the 4k or 8k MMU page size is now available. This feature is selected by setting the INIT module flag `SSM_Page8k` (1=8k, 0=4k).
- You now have the ability to choose SSM default system state cache mode as write-thru or copy-back. This feature is selected by setting the INIT module flag `SSM_CMode` (1=copy-back, 0=write-thru). This change allows the use of the single binary `ssm040/ssm060` where two were used previously (`ssm040_cbsup` and `ssm060_cbsup` will no longer be shipped).

SSM Bug Fixes

The internal SSM error recovery paths that did not propagate `E_NoRAM` errors properly are now fixed. This bug resulted in a possible system state bus error during user state page table allocations.

SSM Application Notes

There are new features for the 68040 and 68060 SSMs. These features are the use of system state page tables, 4k or 8k page sizes, and the system state cache mode setting.

OS-9/040/060

OS-9 uses a processor's MMU (memory management unit) to provide the following items:

- hardware-assisted user level per-process memory protection
- definition of logical to physical address mappings.
- definition of cache modes for regions of the systems address space.

The OS-9 memory management model can be broken into two distinctive states: the user state per-process model and the system state model. Applications, in the form of processes, run in the non-privileged user state model and are protected from each other using the MMU. The kernel, drivers, file managers, and system state processes run in the privileged system mode and run with no protection.

The OS-9 SSM and kernel modules can provide this protection using combinations of the available page-based and block-based translation/protection mechanisms on the 68040/68060. The user state per-process model for the 040/060 uses the page-based mechanism. This allows memory protection and cache mode granularity equal to the chosen page size, 4k or 8k, for user state.

The system state model is implemented by the use of three of the four 040/060 transparent translation registers (TTRs) or by using a mix of page-tables and two of the four TTRs. The exclusive use of TTRs for system state provides fast translations, no memory requirement for system state page tables, and minimum 16Meg-cache mode granularity.

Using the mixed TTRs and page-tables for system state results in slightly slower translations in some cases, a usually small system state memory requirement for separate page tables, and cache mode granularity equal to the chosen page size. The advantage of the mixed setting is that the cache mode granularity can be as small as 4k instead of 16Meg. This allows for the definition of small regions in general memory with dissimilar cache modes. This is sometimes necessary with special hardware devices, and can provide regions which do not require cache flushing.

Configuration Options

Designers can optimally configure the SSM module for their system using the INIT module's cache list and flags in the `M$SysConf` field—`SSM_SysPT`(bit5), `SSM_Page8k`(bit7), `SSM_CMode`(bit6), `SSM_NoProt`.(bit4). The contents of the INIT module can be changed by setting flags in the systems `systype.d` file in the CONFIG macro section or by the use of the moded or editmod utilities.

- `SSM_SysPT` (system state page table)

The setting of this flag causes the system state page tables to be generated for regions defined in the system cache list. Page tables are not generated for regions covered by `DTT0`. `DTT0` is set up by the system's boot code and usually defines the IO regions (cache inhibited) for boards. Areas defined in the `DTT0` register bypass table based translation, therefore building tables for this area would be a waste of memory. The main motivations for system state page tables is to relieve MVME162, 167, 172, and 177 ethernet drivers from the need to flush caches and to allow the 172/177 based systems to run using copy-back caching for system access instead of write-thru.

It is recommended that you use `systype.d` system state page tables so that ethernet can be run without flushing.

Note

The generated ethernet driver must allocate its memory from the locally defined color NoCacheRAM and should having flushing compiled off.

The following fragments show what INIT module changes are necessary to allow this to work. They are found in the following file:

`$MWOS/OS9/68060/PORTS/MVME177/systype.d`

```
* System State Cacheing Mode
* Set system state cacheing mode to copyback (no Page tables)
* Current set to 0 by init.a for Write Thru mode for use
* with ethernet
*
ifdef Config
Config set SSM_CMode
endc
```

- `SSM_Page8k` (Use 8k system page size)

  The setting of this flag in the INIT module results in the use of the 8k-page size for states using the page-biased mechanism. Using this larger size should reduce the number of hardware ATC/TLB misses. This reduction can result in faster code execution.

  In addition, the setting of this flag causes `D_BlkSiz` to equal 8k. This means that the system will give user state processes memory in an 8k modulus of the request size when the process local free memory image can't satisfy a request. This may speed up memory allocation, but it may also increase the amount of memory a process uses.

- `SSM_CMode` (Default system state cache mode)

  The setting of this flag in the INIT module results in system state running in copy-back cache mode instead of write-thru mode. This flag provides the same functionality that was previously provided through the use of binaries `ssm0X0_cbsup` and `ssm0X0`. It applies only when the `SSM_SysPT` flag is not set.

  When the system runs without system state page tables, system state memory access is controlled by the combination of `TTRs`, `DTT0` and `DTT1`. This flag's state is reflected in `DTT1`'s cache mode field. Systems using copy-back caching for system state run considerably faster. Copy-back system state caching does not work for all system configurations (hence the introduction of `SSM_SysPT`).

- `SSM_NoProt` (No protection mode)

  The setting of this flag in the INIT module results in the building of user state page tables without the protection features of the MMU enabled. This flag allows precise cache modes to be defined for user state without protection. This is the default mode for the Atomic Kernel.

- Cache lists (Specifies cache modes for regions)

  Cache list entries specify caching modes for the regions of memory a system needs to access. Cache lists are only used by the page-based translation mechanisms. If `SSM_SysPT` is set, all memory, which the system needs to access, must be defined in the cache list. The cache list is defined in `systype.d` and is contained in the INIT module.

For More Information

See the ***OS-9 Technical Manual*** for a description of cache list format.

PCF

- Long file name directory entries are read as directory entries with the file name `.lngf` instead of what pcf was returning before.

  The long file names are still hidden, but now behave like an old version of DOS.

- The bug that caused bus faults in the FD cache has been fixed.
- Empty files and the root directory now both use a first cluster address of 0.

I/O System Updates

The following sections explain updates of the I/O system.

IOMAN

The attach function of IOMAN has been changed to allow device drivers to sleep in their init routines without causing device table entries to be overwritten. Previously, if the `init` routine of a driver performed a sleep, it was possible for the device table entry set up by the init operation on another device to be overwritten.

RBF

- When RBF deallocates sectors from the bitmap as part of a delete, the `SS_FTL` setstat is called with the `FTL_DelBlk` subcode set. This allows Flash File System drivers to be notified that a group of sectors need no longer be managed. If RBF finds that a device does not support the `SS_FTL` setstat, the call will not be made when subsequent deallocations occur.
- The create and delete operations now update the last modified time in the containing directory. Previously creating or deleting a file from a directory would not cause the last modified time to be updated.
- RBF has been fixed to lock the directory where a new file is being created or deleted. Previous versions of RBF would allow two files of the same name to be created in a directory under some circumstances.
- Editions of RBF previous to 104 could, under some circumstances, remove a file from its directory without returning the disk space allocated to it. This problem has been fixed.
- A new utility, named partition, has been added to OS-9 to allow the partitioning of large RBF disk drives.

For More Information

For more information on this utility, see the ***Utilities Reference*** manual.

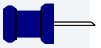
68328 Board

The following represents improvements to the 68328 board for this release:

- The priority of `lmm` to 129 has been increased to avoid unreliable download with kermit. Refer to the example below:

```
$ lmm ^129 /pipe/4 &
$ kermit -rik>-/pipe/4
```

- MAUI does not support 1 to 2 bit expand blit operations.
- The MC328ADS bootcode has been changed to allow the use of the PCMCIA card. This enables the use of linear flash cards using the memory window.
- The MC328ADS memory timings have been reduced to take advantage of the fast SRAM parts being used. This can result in a 2x gain in raw performance.



Note

If you use slower memory, change the `systype.d` file for the appropriate number of wait states and recompile your rom image.

Utility Updates

All of the 68K utilities have been recompiled with the current UltraC compiler. This may have caused the CRC and/or size of these utilities to change even though they have had no source updates. The edition number is a better indication of updates.

Most of the utilities (68K and WIN) were updated and their edition numbers increased due to improvements in our compiler technology. All warnings generated in the compiling of the utilities are now corrected. Most code problems identified by UltraC were cosmetic; however, there was a detection of some important uninitialized variables and there were a few cases of incorrect parameter passing.

The default stack size of the utilities has been increased as a result of customer requests. If your installation calls for a smaller (or larger) default stack size, use the `fixmod -us=<size> <util>` to modify the utility.

Updated Utilities

The following lists represents changes made to existing utilities for this release:

`copy`	There is a new `-c` option to convert end of line characters between different line terminations.
`dir`	The command line argument parsing, year and file size print, and sector number sign have been fixed.
`events`	The `-h` and `-k` options have been added.
`format`	New prompts for cluster size and number of sectors have been added, as has support for partitioning large disks; in addition, problem error checking auto-sized disks have been corrected.
`kermit`	Several of the problems that could cause memory corruption have been fixed.

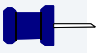
os9make	As of edition #97, os9make supports conditional statements including ifdef, ifndef, ifmake, ifnmake, if, elif, elifdef, elifndef, elifmake, and elifnmake. As of edition #100, os9make supports use of += for incremental macro definitions.
procs	New command line argument to specify individual process IDs.
save	There is now support of the use of -f and -z together.
shell	Several of the problems that could cause memory corruption have been fixed. In addition, a -<sig> option to kill command has been added.
sysgo	sysgo now uses mshell for processing the startup file and as the shell forked on the console port for user interaction. The standard sysgo now looks for the startup file in the SYS subdirectory of the default disk; this provides consistency with other versions of OS-9 and helps with the transition to startup files that can take advantage of mshell specific features. Sysgo_nodisk also uses mshell. Sysgo_shell has been added as a backward compatible version of sysgo that still forks shell and looks for the startup file in the root of the initial disk device.
tapegen	The bug that caused division by zero error has been fixed.
tapestart	The corrupt params in error message when out of memory have been corrected.
tsmon	tsmon no longer bus traps when using the -z (from stdin) option.

umacs	Several of the problems that could cause memory corruption have been fixed.
unlink	The -f option can no longer enter an infinite loop by attempting to unlink members of a module group.
xmode	xmode no longer bus traps when using the -z (from stdin) option or when attempting to change a write protected descriptor.

Added Utilities

The following lists represents utilities added for this this release of OS-9:

- bfed
- chown
- mkdatmod
- mshell
- mv
- partition
- su
- tar
- undel
- what



Note

For more information on all utilities, see the ***Utilities Reference***.

Ports Directories

The following sections explain updates made to and within the Ports directories.

Ports Directories Structure

The ports directories are undergoing changes to get their structure in line with other versions of OS-9. Each Port now has a number of sub-directories corresponding to areas of code that can be rebuilt. Examples are `ROM_CBOOT`, which contains the makefiles to remake the rom image and the rom-based bootfile.

`BOOTS` contains makefiles to build bootfiles that can be booted via hard disk, bootp, tape, or floppy booters. `SPF` contains the makefiles to rebuild items for the SoftStax networking system. Each subdirectory contains makefiles to build descriptors and items for configuring a port's board level solution (BLS) and may also contain makefiles to rebuild drivers in the OEM version of the package.

MVME1XX Ports Directories Overview

Enhanced OS-9 is a Windows hosted cross development environment for coding, compiling, and debugging. Once the package is loaded on the PC, the next step is to bring up the target board for the first time.

The MVME167 boards require ROMs containing the binary image for the board to be installed. Roms are shipped with the MVME167 BLS and can be optionally ordered with the OEM Source Package (OS-9 for Embedded Systems). MVME162, MVME172, and MVME177 boards require "MotBug" eproms to program binary images into the on-board flash memory.

With Enhanced OS-9 for 68K, the binary image sizes on the MVME162, MVME167, MVME172, and MVME177 boards have been increased from 512k to 1 MB. The additional space is divided into two sections.

In the default case, the first 128k are used for the ROM based hardware initialization, ROMBug (if applicable), and the booters used to boot the operating system. The remaining 896 kilobytes are used for an OS-9 Bootfile, which can include a substantial number of modules and utilities. The separate utility section used in the 3.x series ROMs is now included within the bootfile in the ROMs.

The ROM and ROMBug images shipped with Enhanced OS-9 include all the system modules and utilities to initially format a hard disk (if attached), configure the SoftStax/LAN Communications Pak TCP/IP networking stack, and start ftp and telnet daemons. This enables you to bring the board up in a short amount of time. Once the system is running, you can FTP programs to the machine for testing, or include them in the boot image if the system has no disk or other persistent storage available.

INIT

The init modules have been tuned to search relatively small amounts of rom to speed the booting process. Additional ROM space can be searched once the kernel is started by adjusting the `MemType` macros in the init module.

`Init_dd`, which defines a default disk drive as `/DD`, was added. `Init_dd` forks `sysgo` as the first process. The standard `rom.bl` file sets `/dd` to be a ramdisk and uses `init_dd` rather than `init_rom`, which had no default disk defined. Programs that look for password, termcap, and `errmsg` files often check the `/dd` device.

ROM areas (module search lists) have been limited to speed the boot process by refraining from searching areas not used by the initial configurations. If you choose to install additional or larger ROMs, you may need to rebuild their init modules after adjusting the lists in `systype.d`.

`SysMbuf` was added as a `p2` module to initialize at kernel startup time. `OS9P3` was added as another generic `p2` module to initialize at kernel startup time at the end of the Extens list. You can build your own OS9P2 or OS9P3 modules, add them to the boot, and they will be called—without requiring any changes to the init module.

`init_tape` was changed to use `mshell` instead of `shell`.

As described in the SSM section, a config field has been added to the init module to control SSM caching and page table operation in system state. 68040 ports enable copy-back caching in system state. 68060 based ports currently set system state caching to write-thru mode for compatibility with the I82596 Ethernet driver.

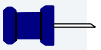
ROM_CBOOT

ROM search lists are set to search the primary address space of FF8xxxxx first and FFAxxxxx second. ROM/Flash based code is designed to co-exist with `MotBug` images on the MVMEXXX board. If the OS-9 ROM image is in the primary socket/address space, the boot process is faster than the previous release.

Booters

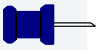
romboot Booter

The generic rom booter now searches the ROM special search list entries for a `kernel` module. If one is found it continues to search for contiguous modules. Once the end of the contiguous modules is found, the size of the boot is passed up to the kernel the same as other boot methods. This contiguous module search is also used for the load RAM boot option.



Note

Since these booters no longer stop at romlist boundary, boots may be able to span multiple ROMs without requiring all of the ROMs to be specified in the searchlist. If the entire space is not included in the search list, and boot searches access a missing rom, the system will not boot. While this method substantially increases boot speed, some additional attention is required when building the ROM boot images so that they do not overflow or exactly fill the available ROM space. The Load Boot from Rom (lr) booter has the same constraints.



Note

Because of caching boundaries and the difference in access speed of RAM versus ROM or Flash memory, Microware recommends you use the Load Bootfile from ROM (lr) boot option rather than the Boot from ROM (ro) option whenever possible.

BootP/TFTP Booter

The `bootp/tftp` booter error handling and messaging has been substantially improved. Booting speed is also improved.

Existing Issues

The following lists represents existing issues for the OS-9 operating system, version 3.1:

- Use of some functions in `clib.il` will generate warnings from `iopt.exe`. These warnings do not affect operation and will be fixed in a future release. Refer to the example below:

```
xcc -td=. -to=OSK -cw -olg -tp=68k,ld,lc -v=. -v=..\..\..\..\SRC\DEFS -v=..\..\..\..\OS9\68000\DEFS
-v=..\..\..\..\OS9\SRC\DEFS -t=1 -s=10 -l=..\..\..\..\OS9\68000\LIB\mauidemo.l
-ill=..\..\..\..\OS9\68000\LIB\HOST3\mauidemo.il  -l=..\..\..\..\OS9\68000\LIB\iff.l
-ill=..\..\..\..\OS9\68000\LIB\HOST3\iff.il  -l=..\..\..\..\OS9\68000\LIB\maui.l
-ill=..\..\..\..\OS9\68000\LIB\HOST3\maui.il  -l=..\..\..\..\OS9\68000\LIB\sys_clib.l
-y=..\..\..\..\OS9\68000\LIB\HOST3\sys_clib.il -l=..\..\..\..\OS9\68000\LIB\clib.l
-y=..\..\..\..\OS9\68000\LIB\HOST3\clib.il -l=..\..\..\..\OS9\68000\LIB\os_lib.l
-y=..\..\..\..\OS9\68000\LIB\HOST3\os_lib.il -l=..\..\..\..\OS9\68000\LIB\sys.l  -r hello.c
-fd=..\..\..\..\OS9\68000\CMDS\MAUIDEMO\hello
iopt: **** warning - possible uninitialized variable 'pending' in _il$15() ****
iopt: **** warning - possible uninitialized variable 'tempsave' in _il$16() ****
iopt: **** warning - possible uninitialized variable 'padloc' in _il$16() ****
iopt: **** warning - possible uninitialized variable 'tempsave' in _il$17() ****
iopt: **** warning - _il$3a() missing return expression in 1 place ****
iopt: **** warning - possible uninitialized variable 'ret' in wcstombs() ****
iopt: **** warning - _flshbuf() missing return expression in 1 place ****
```

- In some VME cards, the ROM boot code may have conflicts with the MVME050 card if installed on the system. At the time of this release, we have not determined if this is a problem with the MVME167 and MVME050 cards or a software problem that may apply to other CPUs with the MVME050. If your system fails to boot with the MVME050 card in the system, try removing the card. Contact Microware Support to determine if a fix is available.
- By default `mshell` has a periodic alarm, which has the side effect of waking from a power manager sleep. Although a future edition of mshell is expected to correct this problem, it is not supported with this release. Following is a list of work arounds:
  - Use the `-nf` option to disable alarm usage.
  - Use shell.
  - Check with the Microware Support web site for an mshell update.
- Due to lack of available hardware at time of release, this package has not been tested with Motorola's newest MVME-A boards. Please contact your local Microware representative for assistance if you experience any difficulties with these boards.

Chapter 3: OS-9 Operating System V3.0.3 Release Notes

This chapter details the changes made to version 3.0.3 of OS-9. The following sections are included:

RadiSys.

MICROWARE SOFTWARE

System Module Updates

The following sections explain updates to certain modules since the release of OS-9 version 3.0.2.

Kernel

- A problem that caused the kernel to crash with an Invalid Stack Frame error has been fixed. This situation was caused due to co-processor mid-instruction stack frames; therefore, it would only affect 68020 and 68030 systems with a 68881 or 68882 floating- point co-processor installed.
- When the Event table contained more than 2047 entries, it was possible for Event entries to become lost.
- The kernel now rejects System Calls (OS-9 for 68K calls) with a system call code greater than 255.
- Support for Power Management functions (`PwrMan`) has been added.
- If you attempted to run a non-68000 kernel on a 68000 processor, the kernel would get an invalid instruction exception when attempting to print out the wrong-kernel message.
- Processes using semaphores could be awakened correctly, but with the `Carry` flag still set, leading to applications thinking that the semaphore operation failed.
- System-state preemption is now explicitly disabled (`D_Preempt`) when the interrupt polling routines are called.
- The system would crash if an `Uninitialized Interrupt (Vector #15)` occurred.
- Trap modules are now unlinked correctly when the last user exits.

- The kernel's default routines for `Protect`, `Permit`, and `ChkMem` no longer perform a simple memory probe on the start address of the region. This makes the kernel's default routine compatible (as far as the caller is concerned) with the operation of the SSM modules. Note that if an invalid address is specified, the bus error still occurs on the actual access.
- A problem with memory leakage and `F$Mem` has been repaired.
- It was possible for non-MSP kernels (all kernels except those for the 68020/68030/68040) to crash the system when an exception occurred in system-state while interrupts were masked. The kernel would consider itself in an interrupt-service context (where exceptions are fatal) when that was not necessarily true. (This case could be provoked, for example, by having a driver mask interrupts, then probe for a non-existing device. The correct response for this type of situation would be to return the error to the caller.)
- The kernel's interrupt service code has been revised for robustness and speed improvements.
- It was possible for a complex name pattern match (using an asterisk, (`*`) or question mark (`?`) in `F$CmpNam`) to cause the system to crash if a task switch occurred at the wrong moment. The process now receives the correct error (`E$StkOvf`) instead.
- If a process's system-stack (located in the process descriptor) overflows (for example, heavy usage of stack in C-language system code) the kernel attempts to abort the process with a stack overflow (`E$StkOvf`) message.

System Security Module (SSM)

The SSM modules for the 68040 (`ssm040`) and 68060 (`ssm060`) have been improved in their management of page-table allocation, both for speed and memory-fragmentation issues. Large address range permits (for example) now complete in approximately half the previous time.

System Definitions

- The following system call numbers have been allocated:
  - F$PwrMan - perform Power Management functions.
  - F$Crypt - perform cryptographic functions.
  - F$HLProto - perform high-level protocol functions.
- SCF baud rates for the following rates have been defined as standard baud rates: 56000, 64000, 31250, 57600, 115200.
- Standard values for PD_PAR fields (stop-bits, parity-type) have been added.
- The following Device Type code numbers have been added:
  - DT_MFM - multimedia file manager.
  - DT_DVM - device manager.

Moded.Fields

Moded.Fields has been updated to reflect the following system changes:

- Append_ mode added to descriptors for SCF, Pipe, and NFM.
- New standard baud rates for SCF and NFM serial descriptors were added.
- New DevCon flags for new RamDisk functionality were added.

Boot Code

The following system changes have been made to the boot code:

- Boot.a has been modified so pc-relative references to (external) vector labels work correctly when spans greater than 32K occur.
- The rompak.m macro has been fixed to repair a problem in the (unused) PAKEND macro.

- The `initdata()` routine of `initdata.c` now properly parses the code and data relocation tables produced by the linker.
- The `Sysboot` glue code (`sbtgueasm.a`) now correctly allocates a properly sized data area when no Rombug debugger is to be built into a ROM that has the CBOOT system.
- The `SysBoot` library routine `hwprbasm.a` has been fixed for offset spans greater than 32K.
- `diskboot.c` now supports ramdisk boots with variable sector sizes.
- The `inetboot.l` library has been updated to provide the following fixes:
  - The client broadcast IP address has been changed to `INADDR_ANY` to support RFC 1122.
  - The port number no longer assigns a negative port number.
  - The client `tftp` code now calculates the block number of the ending bootfile to either be an even multiple of 512 bytes or an even multiple plus one.

I/O System Updates

The following sections discuss I/O system updates and changes.

Real-Time Clock

All real-time clocks were modified to ensure they would function properly through the turn of the century. All clocks were made consistent in their handling of the century roll-over.

Specifically, an epoch of 70 was chosen and used such that any year read from the RTC that is less than the epoch is added to the base year 2000, and any year greater than or equal to the epoch is added to 1900. Thus, the interpreted years range from 2000-2069,1970-1999 as the RTC year value ranges from 00-99.

In addition, the following RTC modules were updated in this release:

- `rtc1216E.a`
- `rtc146818.a`
- `rtc48T02.a`
- `rtc48T08.a`
- `rtc58274.a.`

IOMAN

A problem in `F$SchBit`, which caused the bitmap search to terminate early when at the bitmap end, has been fixed.

RBF

- A problem with premature media full error (`E$Full`) cases has been fixed.
- The `Date` field of the File Descriptor's `Last Modified` and `Date Created` fields has been formalized as the number of years since 1900. This alleviates any rollover issues when the year 2000 occurs. RBF media dates are now unambiguous to December 31, 2155.

RamDisk

- The `ramdisk` driver now supports sector sizes greater than 256-bytes. Valid sector sizes now match RBF's supported sizes (256 - 32K).
- The maximum media size of a ram disk is now 4GB. Previously, ram disks were limited to 32MB.
- RAM disks may now be re-formatted using the `Format` utility.
- Format-enabled ram disk descriptors are now available (for example, `r0fmt.a`) to support formatting of ram disks. The new ram disk descriptors are similar in usage and functionality to those used for hard disks.

Note

The new `ramdisk` driver operates correctly with old descriptors provided you are not attempting to use any of the following new features of the ram disk driver:

- Re-format the ram disk
- Media size greater than 32MB.
- Sector sizes greater than 256 bytes.

If you are using any of these features, you need to make new descriptors.

The initial creation of a RAM disk is still performed in the same manner as previously (In other words, by performing a `CHD` to the RAM disk, or performing an `Iniz` on the device).

SCF

- SCF now checks for deadly I/O signals when allocating a device.
- The default SCF descriptors now add `Append_` to the mode byte.

Pipe

- The default `Pipe` descriptors now add `Append_` to the mode byte.
- `PipeMan` now maintains the FD Date information in a similar manner to RBF.

SCF Driver

The range of signal values treated as deadly to blocking I/O calls has been increased in `sc68562.a` to include all signal values up to `S$Deadly`. This was done to address the discrepancy reported in PDR 94163.

BCC332 Example Port

The `systype.d` file was modified to remove the extraneous `endc DEVTAPE` in the `DiskR0` macro. This was done to correct the problem reported in PDR 95025.

SCSI Driver

- The `RBTEAC` driver and boot code have been improved as follows:
  - Source code is now ACSI compliant.
  - Media can have physical sector numbers starting with `0`.
  - Media formats are now table-driven. This allows easier changes for new formats and new drive firmware revision updates. See ***Appendix B, SCSI Driver Configuration*** for details on the table formats.
- `NCR53C710` scsi drivers have been improved as follows:
  - More robust operation and error recovery features were added, especially when running with fast SCSI drives.
  - Faster operation has been achieved.

Utility Updates

All the utilities were recompiled with a newer version of Ultra C.

`binex`	A `-z[[=]<file>]` option has been added.
`build`	`build` has been fixed to work properly when used from a procedure file.
`cfp`	`cfp` has been changed to use a more recognizable temporary file name. This makes it easier to identify the temp file name when the `-nd` option is used.
`copy`	`copy` has beenfixed a memory allocation problem and corrected to properly copy a file from a pipe device.
`dir`	`dir` has been fixed to print dates correctly after the year 2000. The sector number and file size printouts for sectors/files greater than 2 GB have been fixed as well.
`diskcache`	`diskcache` has been fixed to properly close its standard I/O paths. It has also been fixed to free its data structures more safely.
`format`	`format` has been fixed to work properly for disks larger than 2 GB.
`free`	`free` has been fixed to more correctly handle NFS devices.
`frestore`	`frestore` has been fixed to handle the year 2000.
`fsave`	`fsave` has been fixed to handle the year 2000. The summary printout has been changed to handle backups greater than 2GB.
`help`	`help` has corrected infinite loop when help requested on built-in shell commands.

ident	ident has been added -a option to print alternative information for quick (-q) output.
irqs	irqs has changed interpretation of system data structures for the Version 3.0.3 implementation of fast IRQs.
login	login has been added -n option to support FTP and enable user-defined replacements of login technology. It has also been fixed to handle the year 2000.
make	make has added C++ support. make no longer discards the line after an include line.
merge	The -z file handling has been fixed to trim whitespace from head and tail of each line and to continue after reading a blank line.
mfree	mfree has been fixed to support printing of any number of segments.
os9gen	os9gen has been fixed to support device names of exactly 6 characters and to support disks greater than 2GB.
pr	pr has been fixed to support the year 2000.
romsplit	romsplit has been fixed to properly support -x.
setime	setime has been fixed to support the year 2000.
tapegen	tapegen has been fixed to support the year 2000.
tmode	New baud rates from 3.0.3 have been added for tmode.
tsmon	tsmon has been fixed to support the year 2000.
xmode	New baud rates from 3.0.3 have been added for xmode.

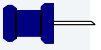
Existing Issues

The following sections explain current issues associated with various parts of the OS-9 operating system.

Intel I82596 Driver

The internet driver for the Intel I82596 supplied with this package will not operate correctly when running on a 68060 with supervisor-state accesses set for `CopyBack` cache mode. `WriteThrough` mode for supervisor-state does operate correctly.

The init module for 68060 based boards has been set so the system will use write-thru system state caching. The MVME 68040 boards have had their system state caching mode set to copy-back. System State write-thru caching is selected via the Config word having a value of `0`, which is the default.



Note

This issue ONLY affects 68060 systems (for example, 68040 systems are unaffected). This issue does NOT affect the ability to set user-state accesses to `CopyBack` mode. User-state cache modes for the 68060 are set in the same manner as those for all other M68K processors, that is by using the CacheList entries in the `init` module.

System-State Debugger

The current release of the System-State debugger (`SysDbg`, edition #79) does not operate correctly on the 68060. Contact your Microware sales representative for more information. In the interim, Microware recommends that you use RomBug for your system-state debugging work.

RomBug

The RomBug debugger is known to have the following problem when running on a 68000-core processor (for example, 68000 and 68328). The soft breakpoints feature does not work. This prevents breakpointing within code in ROM.

System-State Time Slicing

The example code in Chapter 1 of the ***OS-9 Technical I/O Manual*** for dealing with system-state time-slicing is incorrect, in that it does not check for the case of system-state time-slicing being disabled completely. (This is an `init` module option.)

Therefore, the following updated code example is offered as a better example for file managers to follow when determining whether a process can be time-sliced or not:

```
* Preempt - allow process pre-emption, and check for
*           pre-emption pending
*
* Passed:   (a4) = current process descriptor ptr
*           (a5) = caller's register stack ptr
*           (a6) = system global data ptr
*
* Returns:  nothing
*
* Destroys:  nothing
*
Preempt: subq.l   #8,a7                Get space for status save & R$d0 result
*                                           (if sleeping)
         move.w   sr,0(a7)             save ccr status
         subq.l   #1,P$Preempt(a4)     allow pre-emption
         bne.s    Pre99                ..not currently allowed, exit quickly
         btst.b   #TimOut,P$State(a4)  process time-out set ?
         beq.s    Pre99                ..no; keep running
         tst.l    D_Preempt(a6)        is system-state t/s allowed?
         bne.s    Pre99                ..no; exit

* process pre-emption taken
*
         movem.l  d0-d1/a5,-(a7)       save regs
         lea.l    (3*4)+4(a7),a5       set fake frame ptr (R$d0 created above)
         moveq.l  #1,d0                give up time-slice
         OS9svc   F$Sleep
         movem.l  (a7)+,d0-d1/a5       restore regs
Pre99    move.w   0(a7),ccr            restore Carry status
         addq.l   #8,a7                toss scratch
         rts                           return
```

Chapter 4: OS-9 Operating System V3.0.2 Release Notes

This chapter details the changes made to the 3.0.2 version of OS-9. The following sections are included:

RadiSys.

MICROWARE SOFTWARE

System Module Updates

The following sections discuss the system module updates for version 3.0.2 of OS-9.

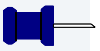
Kernel

The following changes have been made to the kernel since Version 3.0.1:

- Support has been added for the 68060.

For More Information

For more information on 680060 support, see the **68060 Mask Issues** section.



Note

Microware has attempted to support the 68EC060 and 68LC060, but this support has not been tested due to lack of available silicon/hardware.

- A broadcast signal to `F$Send` now sends the signal to the highest process ID.
- System-state time-slicing checks are now more reliable.
- The semaphore (P operation) code now correctly indicates an error when it is awakened (prematurely) by a signal. Previously, the error code was being returned correctly (`E$BSig`) but the `Carry` was not being set.
- User-ram checks which result in overflow are now correctly noticed.

- The Atomic kernel now correctly notices the `PrivAlm` flag in the `Init` module's `Compat` flags.
- Alarms which activate while being deleted no longer cause problems.
- The system now returns an error (`E$NoClock`) if an attempt is made to set a (tick-count) alarm when there is no system clock.

Note

This error only occurs once the system cold-start has been completed.

- The `MBAR` patch location (CPU32 kernels only) now contains the correct default value.
- Semaphore operations now validate the operation (P or V) correctly.
- Problems with returning memory fragments (when using the Standard Allocator) have been fixed.
- The `Sleep(1)` call problem that allowed a process' state to indicate `s` (`Sleep` queue) when the process is actually in the `Active` queue, has been fixed.

SysCache

Support has been added for 68060 processor caches.

SSM

Added support for 68060, using the `ssm060` module.

System Definitions

The following system definition changes have been made and are reflected in the appropriate system libraries (for example, `sys.l`) and C header files.

The following new error definitions have been defined:

- `E$UEA`: Unimplemented Effective Address (68060)
- `E$UII`: Unimplemented Integer Instruction (68060)
- `E$ReqDenied`: Request denied by far end

The following new GetStat/SetStat function codes have been defined:

- `SS_SockMan`: SockMan `GetStat/SetStat` with sub-function code.
- `SS_SPF`: SPF version of `SS_PMOD`.
- `SS_Maui`: Maui `GetStat/SetStat` with sub-function code.

The following new system call has been defined:

`F$Flash`: FLASH file manager operations.

The following process state values (as displayed by `procs`) have been defined:

- `Q_Sema`: `p` - process performed a semaphore P operation.
- `Q_MBuf`: `m` - process awaiting free MBuf buffer.

Definitions for the 68060 FPU state context have been defined.

A new module type value has been defined for Configuration Description Block modules (`CDBData`). This type shares the same value as the Configuration Status Descriptor (`CSDData`).

Moded.Fields

`scsi_wide` descriptions for RBF and SCF descriptors have been added.

A 68681 `DevCon` description for `sc68341` driver has been added.

An `Append` mode description for NFM device descriptors has been added.

Init Module

The `Init` module has been updated for 68060 support and now reflects the latest (Version 3.0.2) release version of OS-9 for 68k.

I/O System Updates

Changes have been made to the following device managers:

- IOMan
- Pipeman
- SBF
- SCF
- SCSI Drivers
- Clock Drivers

IOMan

An I/O operation that is I/O Queue'd no longer aborts the I/O operation when any signal is received. Only I/O deadly signals (signal values less than 32) abort the I/O operation.

Pipeman

System-state preemption checks are now more reliable.

Various problems relating to send-signal-on-data-ready (`SS_SSig`) have been repaired.

SBF

SBF now saves `a6` prior to calling a driver.

SCF

System-state preemption checks are now more reliable.

SCF now updates the device driver static storage special characters after calling the driver for `SetStat(SS_Opt)`. Previously, SCF updated these fields prior to calling the driver, and this would cause some drivers to fail to notice the change (for example, attempting to disable XOn-XOff characters).

If a driver, in servicing a `GetStat` call, changed the caller's `D1` register image (`R$d1`), SCF could be tricked into thinking another type of `GetStat` call was in progress, causing possible system problems. SCF now prevents this from occurring.

SCSI Driver

A number of updated files have been included in this release for various SCSI drivers, both for the high-level (RBF and SBF) device drivers as well as the low-level (Hardware) subroutine modules.

A number of updates are provided to correct a pathname case problem when referencing the `reg.h` file. This issue affects cross-development platforms (for example, UNIX and DOS) only.

The `RBVCCS` and `RBSCCS` RBF drivers have been updated to support the case of `MEDIUM|NOTREADY`.

The `RBTEAC` RBF driver now supports the FD55 and FC-5 configurations.

The low-level module for the WD33C93 device has been modified to be ANSI compliant. SCSI ID checks are also now performed by this module.

For More Information

The ID checks relate to `scsi_wide` support. For more information, refer to **Chapter 5: OS-9 Operating System V3.0.1 Release Notes**.

NCR53C710 SCSI Driver

The low-level module for the `NCR53C710` device has been modified to be ANSI compliant and properly support SCSI-II Fast drives. The sources have been significantly restructured and moved to the `MWOS/SRC/IO/SCSI/NCRSIOP` directory to maintain common sources with other parts in the same or related family. The makefiles for the module remain in `MWOS/OS9/SRC/IO/SCSI/SCSI53C710`.

SBF Device Descriptor

The `sbfdesc.a` Device Descriptor file now supports the `scsi_wide` flag in the `ScsiOpts` field.

SCF Driver

The `sc68681.a` driver has been updated to provide support for the 68341 processor, external clocks (X16) and 38400 baud when running on a 68340/68340/68349 mpu.

The `sc8x30.a` driver has been updated to support indirect data access.

The `sccd2401.a` driver has been updated to support 68060 operation and repair a problem with the initialization of the Timer Period Register (TPR).

Clock Drivers

The `rtc48t08.a` clock driver has been updated to support the 68060 processor.

The `rtc58274.a` module `Attribute/Revision` field is now generated correctly.

The `rtc_example.a` file `Attribute/Revision` field is now generated correctly, and now contains additional routines for Binary/BCD conversions.

ROM Updates

The following ROM changes have been made since Version 3.0.1.

CBoot

The `boot327.h` header file has been repaired to fix a problem with the pathname case of the `reg.h` file inclusion.

`BOOTSCCS` has been updated to support the case of `MEDIUM|NOTREADY`.

ROM Common Code

`Boot.a` has been updated as follows:

- Support for the 68060 processor has been added.
- Problems with short branches going out-of-range when `MBUGTRC` was defined have been repaired.
- The memory testing routines have been improved so that parasitic capacitance issues are not a problem with non-serialized access (68040 and 68060) processors.

RomBug Debugger

RomBug now supports the 68060 processor.

Note

Support for the 68EC060 and 68LC060 processors has not been tested as yet, due to a lack of available hardware and silicon.

The single-port download functionality now operates correctly.

Tracing with the 68341 MPU no longer provokes a format error.

ROM Serial Drivers

The `io68681.a` driver has been updated to support the 68341 processor and external (x16) clocking.

The `iocd2401.a` driver has been updated to support the MVME177 (68060) processor board. Problems with chip reset delays and input echo have also been repaired.

The `ioz8530.a` driver now supports indirect data access.

rompak.m Macro

The `rompak.m` macro has been updated to repair problems when it is used in a ROM set that contains no initialized data.

ROM CBoot

The `sysbtasm.a` file has been updated to add an exit jump missed during the conversion of `CBOOT` to Ultra C.

`BOOT53C94` has been updated to support the Motorola IOP platform.

ROM Definitions

The `nvram.d` file has been updated to conditionally define `NVR_VER_MIN` and `NVR_VER_MAX` only if they are not already defined. The responsibility for defining these values have been moved to the applicable port-specific `systype.d` files.

The `nvram.h` file has been updated for ANSI standard function prototypes and usage with C++.

Hardware Definitions

The `mcecc.d` file has been updated to include register and bit definitions for scrubber operations.

The `mcchip.d` file has been updated for the following reasons:

- resolving definition conflicts with `mcecc.d`
- replacing 33Mhz definitions with 32Mhz definitions
- adding definitions of the upward-compatible MC2 chip

The `mcchip.h` file has been updated for compiling with Ultra C and Ultra C/C++.

The `ipic.d` file has been updated for additional definitions of the upward-compatible IP2 chip.

System Error Message Updates

The `ErrMsg` and `ErrMsg.short` support files have been updated to support the latest additions to system error codes (`E$ReqDenied`, `E$UIA`, `E$UII`).

MVME147 Example Port Updates

The systype.d file has been updated for the following reasons:

- convertion of the BPLOAD conditional to RAMLOAD.
- correction of the number of blocks/buffer in tape descriptors.
- addition of the ROM utilities area to the colored memory list for all but hard disk boots. Increased MDirSz accordingly.
- correction of DrvNum for various /h1 descriptors.
- correction port addresses for SCSI device descriptors.

The sysinit.a file has been updated to convert the BPLOAD conditional to RAMLOAD.

The syscon.c file has been updated to add the parameter missing from the checknvram() call.

The reconfig.c file has been updated to fix the default booter list construction, allow for more than 9 booters, and correct the validation for a 32MB board memory size.

The BOOTFILES/rom.bl file has been updated to not include the cache and ssm modules in a ROM by default.

Makefiles have been updated in order to complete many tasks:

- modified to provide for the definition of RAMLOAD instead of BPLOAD
- modified to adopt the TYPE macro replacing TGTDIR (for ROM makefiles only)
- restructured to use include files (bsp.com and bsp_rom.com) for macro definitions common to both source (OEM) and end-user (BSP) releases
- partitioned to simplify BSP packaging
- modified to simplify testing (rom_bootfile.make)
- modified to check utilities area size (rom_utilities.make)
- modified to fix title comments

IF177 Driver Updates

`If177` ed #34 is provided in binary and source form for development.

Binaries for `if177_206` and `iestat_207` have been added to the `MWOS/OS9/68060/PORTS/MVME177/CMDS/BOOTOBJS` directory.

There are minor differences in `if177` edition #34 and edition #206.

68060 Mask Issues

The 68060-specific kernels supplied with this package have been evaluated with the 01F43G mask of the PC/XC68060 part. Earlier mask versions of the processor are not supported by this release, and if you have an early version we recommend that you contact your Motorola distributor to obtain updated parts.

Due to problems encountered on the 01F43G mask with superscalar operation, running the 68060 at full speed provokes problems detailed in the Motorola Errata (see Rev 1.5, Item I14). To eliminate these problems, the 68060-specific kernels supplied in this package **detunes** superscalar operation by setting Bit 5 in the 68060's PCR. This has the effect of making the system run approximately 3-5% slower than possible.

Motorola's Errata indicates that these problems will be fixed in the next mask (0F84W), but this mask has not yet been evaluated by Microware.

Chapter 5: OS-9 Operating System V3.0.1 Release Notes

This chapter details the changes made for version 3.0.1 of OS-9. The following sections are included:

- **System Module Updates**
- **I/O System Updates**
- **Utility Updates**
- **Internet Updates**
- **SysMbuf Updates**
- **Example Port for 68360 Quads Board**

RadiSys.
MICROWARE SOFTWARE

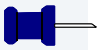
System Module Updates

The following sections explain the system module updates.

Kernel

- The Version 3.0 kernel recognized only one (the last) block of shared memory. This problem has been repaired and the current limit of shared blocks is now ten (as documented in the version 3.0 Release Notes).
- When the system was configured to disable system state time-slicing (`Init` module `SysConfig` flag), loading large modules could cause the load to monopolize the system. Processes that were time-sliced in system state may also exhibit excessive time-slicing. These problems have been repaired.
- The Primary Module pointer (`P$PModul`) for the system process (kernel) now correctly points at the kernel module header, not the kernel module name string.
- A `Sleep(1)` call was returning register `d1` unchanged. `d1` is now returned correctly with a value of zero (`0`).
- When the Version 3.0 CPU32 kernel is used on 68331, 68332, 68F333, and 68334 processors, the kernel may hang during cold start. This problem is caused by the kernel attempting to initialize the system global `D_MBAR` (Module Base Address Register) to the value of the system's I/O block. The problem only occurs on designs which do not return a bus error for accesses to CPU-Space address 0x3ff00.

  If your system behaves in this manner, the `Init` module's `Compat` byte should be set so bit 0 (the `NoMOVES` flag) is set to a one (`1`). The kernel checks this flag, and if set, the default I/O address of 0x00fff000 is set into `D_MBAR` (and no access to the CPU-Space address is made).



Note

This problem is particular to the 68331, 68332, 68F333 and 68334 processors (for example, those CPU32-family processors that do not have a true MBAR register). For CPU32 processors that actually have an MBAR register, the Version 3.0 release notes regarding use of `D_MBAR` still hold true. The processors that have the MBAR register include the 68330, 68340, 68341, 68349, and 68360.

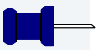
If the system software on a non-MBAR system (for example, a TPU driver on a 68332) wishes to use the value in `D_MBAR`, and the default `MBAR` value is not correct, then two methods are available to accomplish this:

- Write a P2-style module to change `D_MBAR` to the correct value. This is the preferred method for OEMs and end-user's.
- Obtain a patch from Microware if P2-modules prove unusable.
- NVRAM support has been improved. It is now possible to unlink individual modules (or module groups) that reside in NVRAM memory blocks without having to unlink **all** modules in that block.

  System cold start is now faster for the 68020, 68030, and 68040 processors because kernel cold start now makes use of the on-chip caches for these processors. The actual improvement realized depends upon the actual system configuration (for example, size of RAM memory, size of ROM memory area, and number of modules in that area) being used. Increases of 25% to 500% have been measured during testing of this feature.
- The enabling of caches during cold start is designed to be totally transparent to all existing software, and thus it should not require any revision of existing code.
- If, for some reason the enabling of caches causes problems with the kernel start-up (this would usually indicate a problem with the system hardware), patches can be made to the kernel to disable caching during cold start, as described below.

Kernel Patches for Cold Start Cache Disable

- To disable caching during cold start, the following patch can be applied to the kernels indicated. The cache use is disabled by setting bit 6 of the offset indicated in the table below. Once applied, the kernel's module CRC should be updated using the `Fixmod` utility (or equivalent).



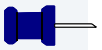
Note

The following offsets apply to all kernels for the indicated processor type. Thus, `aker020s`, `dker020s`, `aker020b`, and `dker020s` would all be patched at offset 0x48.

Processor	Offset	Existing	Change To
68020	0x48	0x00	0x40
68030	0x4c	0x00	0x40
68040	0x4c	0x00	0x40

- If a process used `F$SigReset` more than five times, the system may become damaged. The solution for this problem has been provided in the following two ways:
  - Use the Version 3.0.1 kernel. This kernel contains the updated `F$SigReset` code that allows repeated use of `F$SigReset`, or
  - Use the supplied `p2sigreset` system expansion (P2) module in conjunction with a Version 3.0 kernel.

  The preferred method is to use Version 3.0.1, but the P2-style method is provided so that existing V3.0 systems can implement this fix without requiring a kernel update.



Note

P2SigReset only works on Version 3.0 kernels. If you attempt to use it with any other version of the kernel the initialization part of the module fails with an E$BadRev error.

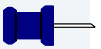
- F$UnLink now checks for a NULL pointer. This could cause a bus trap on systems which do not have memory at 0.
- F$STime now returns E$NoClk (No Clock) as the error when the Init module does not specify any clock module offset. Additionally, if the clock start-up during kernel cold start fails, the development version kernels issue a warning message to the console.
- The kernel cold start could become confused when locating the Init module's module names beginning with the string init (for example, initsys). This problem is now corrected.
- Kernels supporting the 68882 Floating Point Co-Processor (for example, 68020 and 68030 kernels) now correctly recognize a potential co-processor protocol violation when a floating point exception occurs.
- The expansion stack fields (P$ExpStk, P$ExpStkSz) of a process descriptor are now cleared when a process performs a Chain operation.
- A potential bus trap situation has been cured when a process attempted a fork, the system lacked memory to link the trap handler and the system had no memory at address zero (0).
- The kernel's memory search/test routines have been improved to deal better with **out-of-order** execution MPUs (for example, 68040 and above) and memory blocks that are not installed and fail to return a bus error.

Moded.Fields

Moded.Fields has been updated to support the changes to the Init module.

Init Module

The `NoMOVES` flag has been added to the `Compat` byte. See the *Kernel* section for further details.



Note

This flag replaces the obsolete `SlowIRQ` (defined for Version 2.X systems) flag.

The revision information has been updated to reflect the Version 3.0.1 release.

Boot Code

- `boot.a`
- `romboot.c`
- `BootP` database

boot.a

Support for processors in the range 68310 to 68299 is now provided. These processors are 68000-core processors.

The memory search/test routines have been improved to deal better with **out-of-order** execution MPUs (68040 and above) and memory blocks that are not installed and fail to return a bus error.

romboot.c

The parameter list on the first call to `goodmodule()` was corrected. Without this change, the kernel might not be found in some ROM/memory configurations.

BootP

An example `BootP` database is provided in the ***Using LAN Communications Pak*** manual.

I/O System Updates

Changes have been made to the following device managers and drivers:

- IOMan
- SCF
- Pipeman
- RBF
- SCSI

IOMan

If the module directory expanded during the loading of a module group, a bus error could result. This problem has been repaired.

IOMan cold start now returns the correct error code if a problem occurs during IOMan start-up.

Pipeman

A problem with signals has been corrected. This problem occurred if a reader process had used `SS_SSig` (Send Signal on Data Ready) and the writer process blocked because the buffer was full. In this case Pipeman would wake up the reader process with a wake up (`S$Wake`) signal instead of the `SS_SSig` value specified.

SCF

It was possible for the process preemption field (`P$Preempt`) value to be incremented when calling a device driver. This confused drivers as to whether it was safe to perform preemption. This problem has been corrected.

RBF

A problem existed when a `ChgDir` was made to a drive whose logical drive number (`PD_DRV`) was invalid. This caused RBF to get a bus error.

When a disk was formatted so the media-size/cluster-size combination resulted in a bitmap using all available sectors, an attempt to write a large file (for example, greater than 8M byte) as the first file on the disk resulted in a false **media full** error. This has been fixed.

Floppy Disk Formats

Microware has added support to RBF descriptor generation to support the following PC formats:

- 5 1/4" High-Density (1.2M formatted)
- 3 1/2" High-Density (1.44M formatted)
- 3 1/2" Extra-Density (2.8M formatted)

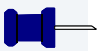
These formats can be invoked using the following disk macros and macro parameters when generating RBF descriptors via the `rbfdesc.a` descriptor generator:

Descriptor Sources:`d0_hd.a, d1_hd.a`, etc.

`d0_ed.a, d1_ed.a`, etc.

Disk Macro Names:`DiskPCD0, DiskPCD1`, etc.

Macro Param #6	Sect Size	Trks	Sec/Trk	Rotate Rate	Transfer Rate	Capacity (formatted/unformatted)	
pcdos580	512	80	15	360rpm	500KB/s	1.2MB	2.0MB
pcdos380	512	80	18	300rpm	500KB/s	1.44MB	2.0MB
pcdos380ed	512	80	36	300rpm	1MB/s	2.88MB	4.0MB



Note

These disk formats use physical disk formats identical to equivalent DOS formats, but this does not imply that the RBF file system is DOS-compatible.

SCSI System

- The high-level drivers previously validated/qualified the descriptor's SCSI options (`pd_opt`), so unknown flags were stripped from the field prior to being passed to the low-level module. This checking is now optional for high-level drivers. They may pass the SCSI options field unchanged or they may qualify it as necessary. The majority of the high-level drivers now pass the field unchanged (for example, `rbvccs`), although some (`rb5400`) still need to qualify the SCSI options to prevent bus lockups.

  Passing the SCSI options field unchanged allows for future options to be implemented without high-level driver updates. It is now the responsibility of the low-level driver to validate and then accept/reject unknown SCSI option flags.

- In a similar manner, the high-level drivers were checking the SCSI target ID and rejecting all IDs greater than 7. With SCSI-3's proposal for IDs ranging up to 31, this checking by the high-level driver is no longer correct. It is now the responsibility of the low-level driver to range check the target ID. All Microware low-level drivers have been set to check for IDs in the range 0 - 7 - thus the net effect has been to move the range check from the high-level module to the low-level module. Microware has not implemented any WIDE support in the current driver examples due to a lack of testing hardware.

  The above changes affect almost all high- and low-level drivers for SCSI.

- The `rbvccs`, `rbsccs`, and `rbteac` drivers have been updated so the write-verify function is performed correctly. Previous versions verified too many blocks, thus making the verify operation take longer than necessary.

SCSI 327 Driver

- The driver now supports synchronous operation on boards having Version 2.7 firmware.
- The driver sources have been changed to support ANSI style compilation using UCC.

Utility Updates

The `fixmod` and `ident` utilities have been updated to recognize OS-9000 PowerPC modules.

Updated `csl` modules are included for proper operation of the `fixmod` and `ident` utilities.

Internet Updates

The following changes have been made to the internet modules, libraries, products, and drivers.

IP Module

Routing table maintenance was improved to try to find the lowest hopcount route when a route is requested. Add route will not add a longer hopcount route when a shorter hopcount route exists. This avoids routes entering a looping situation.

The `rtabupdate()` algorithm was also improved to fix a problem in IP edition 85 which allowed an outdated route to get stuck with a hopcount of 16 instead of getting deleted.

Inetboot.l Library

`Inetboot.l` library has been modified so it can let an OS-9 for 68k BootP client use Sun Solaris 2.3 as a BootP server.

Ethernet Drivers for MVME162 and MVME167

Following improvements have been made in edition #34 of the `if162` and `if167` drivers:

- Restarting the Receive Unit after lockups during heavy traffic has been improved.
- Handling of unexpected chip behavior has been improved. The `recv_unused` and `rbd_headlost` fields have been added to the `iestate` (chip statistics) structure to keep track of such behavior. The `iestat` utility has been updated to display these fields.

tcp

An interaction between the persist and retransmission timers causing tcp-based connections to hang has been corrected.

IFSLIP

The problem that occurred when trying to execute `slipstat` from a remote SLIP session has been corrected.

The `slipstat` output display has been reorganized.

SysMbuf Updates

`SysMbuf` has been modified to use events when a process blocks for lack of mbufs. The separate mbuf queue is no longer used. A problem causing failure to reactivate a process waiting for mbuf availability has been corrected.

A possible register usage problem has been corrected.

Example Port for 68360 Quads Board

An example port for the 68360 Quads Board has been added in the directory `MWOS/OS9/CPU32/PORTS/QUADS` and is explained in Appendix A: Example Port for M68360 QUADS Boards of these release notes.

Chapter 6: Hawk 1.2 Release Notes

For More Information

For complete information on how to use Hawk, refer to the following documentation.

Using Hawk

Using Hawk Macros

The Hawk On-line Help System (accessed from the Hawk Integrated Development Environment).

RadiSys.

MICROWARE SOFTWARE

Architecture

The graphical, PC hosted, sections of Hawk have been completely re-designed to reduce the numbers of exception faults and to facilitate easier future maintenance. Other improvements have been made to the host code, in order to improve debugger performance: especially code used for stepping through individual lines of the debugged application.

Documentation

The documentation and on-line help have both undergone significant improvements.

Changes made since the previous product release include:

1. Eliminated Using Hawk Profiler (incorporated into Using Hawk)
2. Eliminated Hawk Programming Reference (now in Help file only)
3. Eliminated Using Hawk Utilities (now in Help file only)

The resulting manual set includes:

1. Getting Started with Hawk
2. Using Hawk
3. Using Hawk Macros

Product Improvements

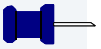
The following issues are believed to have been resolved:

- Hawk better handles variables which have been optimized away to nothing
- General protection fault in compiler front end has been fixed
- Header files will be included in next OS-9/68k product release
- Line number of error is now available in assembler
- Hawk no longer uses an absolute path and does not assume code is on drive C:
- The Error button in the toolbar now works
- “run” and “stop” sometimes entered a mode where they only worked every second time
- Problems with the rebuild button in Project view have been fixed
- Hawk is now able to recover, if the target is rebooted
- Toggling of read-only mode now works correctly
- General protection fault in Linker has been fixed
- Numerous access violations have been fixed
- Hawk no longer has problems moving and docking windows
- The FasTrak manual has been replaced by the Hawk manual
- Access violation in Hawk when using absolute pathnames now solved
- Environment variables can now be separated by carriage return
- A general protection fault in cpfe is fixed
- The watch window no longer limits an array to just 32 elements
- RegDBDeleteKey error removed from InstallShield
- Telnet to OS-9/StrongARM now works correctly
- Button/menu problems fixed

- Packaging changed to add definitions files and solve 8.3 file name problems
- A problem with closing one window and another also closing has been fixed
- Access violation problems when importing project have been solved
- Missing help file is now included
- Bounds checking has been added to array accesses
- A certain key sequence could cause Hawk to enter infinite loop
- The debugger can now find files in sub-directories
- Hawk is now able to attach to subroutine modules
- Hawk no longer has compiler's -t and -s options swapped
- A problem in dependency creation has been fixed
- Mis-handling of PowerPC r0 during copy propagation optimization has been fixed
- Left-right scrollbar has been added to “build” window
- Hawk no longer “freezes” after certain combined key presses
- Updates to assembly source code window during animation have been improved
- Problems with default Hawk environment has been fixed

Chapter 7: LAN Communications Pak 3.3 Release Notes

These notes describe release 3.3 of the LAN Communications Pak.



Note

For hardware specific information, please see the release notes for the board you are interested in.

RadiSys.

MICROWARE SOFTWARE

Enhancements/Changes Since LAN Communications Pak v3.2

Enhancements

The LAN Communications Pak is now year 2000 compliant.

Multicast support has been added to the core protocol stacks, as well as the slip and PPP drivers. Refer to the release notes for a particular board for information on multicast support for specific hardware drivers.

Interface drivers now inform IP of their capabilities, such as multicast or broadcast support, rather than depending on them being specified correctly in the `inetdb` modules or on the `ifconfig` command line. If it is not desirable to use all the capabilities of a driver, new options have been added to `ifconfig`, `ndbmod`, and `idbgen`. For example, if you do not wish an ethernet interface to support broadcasting, you can specify the `iff_nobroadcast` option.

ifconfig

- Added `iff_nobroadcast`, `iff_nomulticast`, and `iff_nopointopoint` options to override driver default values.
- No longer requires UDP. Now also tries TCP or RAW sockets before giving up.
- Repeated changing of an interface's netmask no longer results in an `EOS_FULL` error.

ndbmod

- Added `iff_nobroadcast`, `iff_nomulticast`, and `iff_nopointopoint` options to override driver default values.

netstat

- Added `-ia` option to print all multicast groups joined on each interface
- Prints `igmp` statistics using either the `-s` or `-p igmp` options.

ping

- Increased the size of the receive buffer to handle packets bigger than 48K. Now able to specify values up to 65507 using the `-s` option.

routed

- No longer crashes when a new interface is added by `ifconfig`.

idbdump

- Now prints `mw_flags` in the interfaces section. The override flags are: `iff_nopointopoint`, `iff_nobroadcast`, and `iff_nomulticast`.

idbgen

- Modules are now created as a multiple of 8 bytes in size to fix alignment problems on some platforms.
- Added `iff_nobroadcast`, `iff_nomulticast`, and `iff_nopointopoint` options to override driver default values.

rpcdbgen

- Modules are now created as a multiple of 8 bytes in size to fix alignment problems on some platforms.

exportfs/showmount/rusers

- The date printing is now year 2000 compliant.

rusersd

- Now uses 1970 as its time epoch instead of the OS-9 version 1.x time epoch of 1980.

pppd

- Fixed hole where we can receive a signal before we are ready to process it.

netdb.l

- Fixed problem with the trap module not handling the stack properly in certain situations.
- If a DNS server returns an ICMP port unreachable error, the next listed DNS server is tried rather than returning an error.

socket.l

- The backlog parameter to listen was ignored and assumed to be the old maximum of 5. It is now passed correctly, and the maximum has been changed from 5 to 128.

- The `accept` function properly sets the foreign port and address so subsequent `getpeername` functions work correctly.

rpc.l

- Now compiled with long code and long data for all targets.

spenet

- Now supports IP multicast packets.
- Correctly sets `arp_op` to `ARP_REPLY` when responding to proxy `arp` requests.
- Fixed mbuf leak when receiving packets from broken hardware drivers.

spip

- Multicast support has been added.
- Support for IGMP version 2 has been added.
- Now able to add interfaces and routes from an `inetdb` module in ROM.
- Correctly counts input bytes on the loopback interface. This value is now correct when viewed with `netstat`.
- The `checksum` function is now written in assembly for ARM4 targets. This is the common function also used by `SPTCP`, `SPUDP`, and `SPRAW`.
- If only an interface's netmask changes and not its address, the request is not passed to `spenet`.
- Applied patch for CERT advisory CA-98.13.
- The 500ms IP timer now only runs when IP fragments are present in the IP reassembly queue.

sptcp

- The retransmit timer is set if TCP can not get an mbuf to prevent certain deadlock situations when running low on available mbuf space.
- If all TCP sockets are closed its 500ms and 200ms cyclic timers are stopped. This allows the system to suspend if power management is enabled.
- The TCP congestion window uses an improved algorithm.

spudp

- Support for multicast socket options added.
- Incoming data is now delivered correctly to multiple recipients, even if it is in an mbuf packet chain.

spraw

- No longer corrupts memory when binding a RAW socket to a specific IP address.
- Incoming data is now delivered correctly to multiple recipients even if it is in an mbuf packet chain.
- Connected RAW sockets now work.

sproute

- Verifies size of user-supplied buffer before copying interface information into it.

spslip

- Added support for IP multicasting.
- No longer corrupts occasional non-TCP packets by attempting to compress them.

sphdlc/splcp/spipcp

- Now allows multiple PPP stacks to coexist by changing the `lun` number in the descriptors.

spipcp

- Added multicast support.

Known Issues

Installation

Care should be taken when installing the LAN Communications Pak on a 68000, PPC, or 80x86 platform which previously had ISP installed. ISP (Internet Support Pak) is the BSD v4.2+ TCP/IP stack provided in the past.

The two networking packages can not be run on the same machine at the same time, nor can the utilities for one run with the other. In addition, several utilities such as `telnet` and `ftp` share the same names. The installation program will list the conflicting utility names. All conflicting utilities are in `MWOS/<OS>/<PROCESSOR>/CMDS`. The ISP versions should either be renamed or moved to another directory.

Running any utility with the `-?` option will tell you what networking stack it is for. The LAN Communications Pak utilities will say SPF for the IP stack. If the utility does not list any IP stack, then it is an older ISP utility.

NFS client and server

- The NFS server only supports RBF devices. PCF file systems can not be exported at this time.
- The `rename()` function returns `000:208` Unknown Service Routine when used on an NFS disk. The workaround is to use the `_ss_rename()` function. Append mode is not supported.
- Mount disk points are limited to 2 Gigabyte media size.
- NFS does not support file locking.
- The `create()` function acts like `creat()` for the NFS server software.

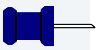
Chapter 8: MAUI 2.3 Release Notes

This chapter provides release note information for MAUI 2.3. It also includes release note information for MAUI 2.2.

RadiSys.

MICROWARE SOFTWARE

MAUI 2.3 Release Notes



Note

Because of the limited resolution of the MC68328ADS LCD, many of the standard MAUI demo applications are not appropriate. The following list details the applicability of each of the standard MAUI demos:

The following applications are bit-depth independent:

- aloha
- fdraw
- gxdevcap
- hello
- inp
- msginfo
- msgrdr & msgwrtr
- sfont

The following demo applications are not supported on the MC68328ADS:

- fcopy - uses the CDi IFF file format which does not support 1 bit
- jview - does not support the low bit-depth
- auplay - no audio device
- aurecord - no audio device
- showimg - uses the CDi IFF file format which does not support 1 bit
- windraw, winink, and winmgr (a.k.a winmgrdemo) - bit depth issues

Bug fixes

- Corrected IOBLT copy from an interlaced source
- A `gfxdev` parameter was added to `gfx_clone_vport()` to correct a problem where the cloning application would attempt to access the graphic device using the `path_id` of the application that originally created the viewport.

MAUI 2.2 Release Notes

New Features

- Color map control - these functions provide a consistent interface for MAUI application developers for color management in windowing. This includes bug fixes and enhancements for Java applications to enable transparency.

  `win_alloc_cmap_cell`

  `win_alloc_cmap_colors`

  `win_get_cells_params`

  `win_get_cmap_params`

  `win_set_color_match`

- Cursor control - these functions provide application developers the ability to programmatically move the cursor to desired screen positions plus the ability to hide and show the cursor on the fly.

  `win_set_cursor_pos`

  `win_set_cursor_state`

- Added new Blt and Drw functions to support overlapping copy. As a result, the graphic memory requirement for maui_win was reduced by half.

  `blt_copy_oblock`

  `drw_copy_oblock`

- Common VGA driver codec - this architecture allows quicker ports to new VGA devices and lower maintenance costs.
- Extended coding method definition - this supports hardware requiring large line alignments.
- Graphic driver common code enhancements to facilitate hardware BLIT acceleration and other BLIT customizations.

- Graphic hardware cursor support in GFX API and `maui_win` - these functions can result in a performance boost for platforms that support graphics hardware cursors. Hardware cursor support eliminates the significant overhead of redrawing the cursor and backgrounds beneath the cursor.

  `gfx_create_cursor`

  `gfx_destroy_cursor`

  `gfx_get_cursor_cap`

  `gfx_set_cursor`

  `gfx_set_cursor_pos`

  `maui_win` - updated to take advantage of hardware cursor
- Shared library support for StrongARM, SH3, and SPARC.

Optimizations

- RAM - the dynamic footprint requirement is reduced in windowing applications. For example:

  Brutus - 320 x 240 = 76K
- Speed improvements - 10% faster in CaffeineMarks 3.0.
- Screen flicker - Corrected/reduced screen flash and cursor flicker due to extra window expose events when creating, moving and resizing windows.
- `drw_arc` - arc drawing works and is much faster

Demos

- Improved readme notes

Several new demos, including the following:

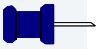
- New JPEG library viewer.
- New `gxdevcap` utility displays device capabilities information about each graphic device on the system.
- `demo_set_timeout()` now uses `signal()` instead of `intercept()` to eliminate signal handler conflicts.

Bug fixes

- ANSI compliance/Compiler warnings
- `drw_oval_arc()` draws a correct oval
- `drw_ellipse()` in dash mode no longer strays from bounding box
- Eliminated unnecessary draws of cursor in windowing
- CLUT of top graphic device is now restored following close or restack of device
- Fixed drawing problems that were causing artifacts in the Java Gas Pump demo
- `blt_copy_block` now reads the correct number of bytes—which was causing unexplained crashes.
- `win_move_win()` no longer exposes to much area, cutting down the amount of UI refresh needed.
- The check for topdev is now correct for cloned devices
- Corrected identity crisis of `drw_set_context_origin()` and `drw_set_context_draw()`. Their error messages claimed they were `DRW_SET_CONTEXT_CLIP`

Chapter 9: SoftStax 2.2.1 Release Notes

This chapter provides release note information for SoftStax 2.2.1.



Note

This product was formerly titled SPF Base Package.

RadiSys.

MICROWARE SOFTWARE

SPF File Manager

A number of minor updates and bug fixes have been made to the SPF File Manager.

Updates

- Added capability for `select()` call to return if data is already waiting on a path.
- The SPF file manager chains application data for datagram I/O option path (`spf_popts.pd_iopacket` type is `IO_PACKET_TRUNC`), and sends it down to the protocol driver.
- The SPF file manager checks the TCP read status (`pd->pd_item.error_state`) before going to sleep to wait for data within `read()` function, and returns if there is an error state.

Bug Fixes

- The SPF file manager checks the persistent incrementing events using `notify_type->ntfy_class` instead of `notify_type->ntfy_on` in `DR_FMCALLUP_NTFY`.
- The SPF file manager correctly updates the read queue count.
- Removed user state process permission for reading and allocating System Mbuf.
- `Dr_use_cnt` has been corrected to count exact number of use count.
- The stack cleanup procedure removes mbufs in the SPF receive thread queue when the network stack is torn down.

- In the case of the stack initialization error at a path open time, the file manager now call cleanup for the bottom driver (normally the hardware driver), and the hardware driver should be removed from the OS-9000 system. However, because of the nature of the DPIO, the hardware drivers' attach count will be one more than actual count, and the hardware drivers `dt_term` won't be called for the OS-9 system. For the OS-9 system, iniz the devices before running applications is recommended.
- The SPF file manager will return `SUCCESS` when partial data has been sent out and there is an `EWOULDBLOCK` error occurred on an ASYNC mode (non-blocking) path.

SoftStax Notes

Headers

item.h

- Added `ITE_ON_DR_DEFINE` for `notify_type.ntfy_on` for driver specific notification type.
- Added ethernet address class `ITE_ADCL_ENET` on `addr_type.addr_class`.

item_pvt.h

Added `ITE_GETADDR` definition.

mbuf.h

- Added `MBT_SENDTO` definition for `mbuf.m_type` to distinguish UDP `sendto()` and `send()`.
- Added `M_PREPEND` macro to prepend space size requested to mbuf if there is enough space.

spf.h

- Added `SPF_PR_SOCKET` definition to the Microware protocol type.
- Added `spf_pdstat.pd_flag` to support UDP `sendto()`.
- `pd_ioasync` path option bug is fixed: `IO_SYNC` is 0, `IO_ASYNC` is 1, `IO_WRITE_ASYNC` is 3, and `IO_READ_ASYNC` is 5.
- Added `SPF_PR_USB` definition.
- `lu_prottype` in `spf_luopts` structure (Logical Unit Option) is moved to after `lu_optsize` to accommodate `u_int16` definition.

Descriptor File

stat.c

`lu_prottype` field in `spf_luopts` was moved to after `lu_optsize`, and `lu_rsv2` field (renamed from original `lu_prottype`) has 0 value. All descriptors in the SPF Base Pak v2.2 were remade with the new `stat.c` and are shipped in SoftStax Base Pak v2.2.1.

desc.tpl

`os9make clean` does nothing, and `os9make purge` removes the target descriptor. It is necessary to purge the target descriptor in order to remake it.

Libraries

dbg_mod.l

The `dbg_mod.l` for ARMV3 didn't line up the debug module static area correctly. This is fixed by the compiler and the pointer to the static area is .rr7+64.

item.l

Separated functions out of single library source to provide optimal libraries. Previously, all functions were in a common C file and would be embedded in the final application. By separating each function into its own file, only functions required would be placed in the final application. Added check for NULL pointer in the following functions.

```
error_code ite_ctl_connect(path_id path, Addr_type our_num, Addr_type
   their_num, Notify_type npb);
error_code ite_ctl_connstat(path_id path, Device_type dev_info);
error_code ite_ctl_resadd(path_id path, Notify_type npb);
error_code ite_ctl_rcvrasgn(path_id path, Addr_type telno, Notify_type npb);
error_code ite_dev_getmode(path_id path, u_int16 *mode);
error_code ite_ibreschg_asgn(path_id path, Notify_type npb);
```

The following ite_mpeg functions were removed from the item library:

```
error_code ite_pgm_psiget(path_id path, Ite_psi_pb psi_pb);
error_code ite_pgm_psirmv(path_id path, Ite_psi_pb psi_pb);
error_code ite_pgm_psimask(path_id path, Ite_psi_pb psi_pb);
error_code ite_pgm_notify_asgn(path_id path, Ite_psi_pb psi_pb,
   Notify_type npb);
error_code ite_pgm_send_notify(path_id path, Ite_psi_pb psi_pb);
error_code ite_pgm_notify_rmv(path_id path, Ite_psi_pb psi_pb);
error_code ite_pgm_get_pref(path_id path, Ite_stream_pref pref);
error_code ite_pgm_set_pref(path_id path, Ite_stream_pref pref);
error_code ite_pgm_info(path_id path, Ite_pgm_tbl pgm_table, Notify_type npb);
error_code ite_pgm_info_rmv(path_id path, Ite_pgm_tbl pgm_table);
error_code ite_pgm_set(path_id path, Ite_pgm_tbl pgm_table, Notify_type npb);
error_code ite_pgm_abort(path_id path);
error_code ite_flush_pat(path_id path);
error_code ite_set_mbanchor(path_id path, Mbuf *mb_anchor);
```

mbuflib.l

When the `SPF_NOFREE` bit is set, the following functions will reinitialize the mbuf header.

```
Mbuf  m_free(Mbuf mb);
int32 m_free_m(Mbuf mb);
Mbuf  m_free_p(Mbuf mb);
void  m_free_q(Mbuf *queue);
```

OS-9 Modules

mbinstall

Fixed to restore `stdin`, `stdout` and `stderr`.

mbdump

Fixed to restore `stdin`, `stdout` and `stderr`.

SysMbuf

Removed user state accesses. SysMbuf call for `mbdump` now returns pool address instead of bit map address.

OS-9000 Modules

mbdump

Updated help text and formatted bit map display.

SysMbuf

Access/Creation is limited to a system state process. Speed optimization for mbuf allocation is added to PPC and ARM.

Loopback Driver (sploop)

If a path registers a notification for far-end hangup, and the connected path hangs up, the far-end hangup notification registered path now handles the path close properly. A path which has an established connection returns `EOS_DEVBSY` for `ite_ctl_answer` call.

Compiler Optimization Issue for System State Process

Any spf drivers using `item.l`, `ppstat.l` and `sptimer.l` need to recompile with `-ao=sc` optimization option due to an icode linking issue in system state modules.

makefiles

Cosmetic changes were made in makefiles for WINDOWS support.

Compiler Warning Issue

Cosmetic changes were made to most of the networking-related source files to remove compiler warnings. This may have caused the CRC to change.

Appendix A: Example Port for M68360 QUADS Boards

This appendix describes the example port for M68360 QUADS boards. The following sections are included:

- **Enhanced OS-9 for 68K Quads Board Port Guide**

RadiSys.
MICROWARE SOFTWARE

Enhanced OS-9 for 68K Quads Board Port Guide

Overview

While the QUADS 68K-based board is no longer available, the port itself does contain a valid working CPU32 based example.

The QUADS port may be used as an example to bring up custom OEM based CPU32 based hardware.

There is no hardware beyond serial and ethernet connections in the QUADS port. OEM's who use this board as a basis for new design work and encouraged to also refer to other example ports for additional support of devices. The MVME177 port is one such port. It is located in the following directory:

```
\mwos\OS9\68060\PORTS\MVME177
```

Support for onboard ethernet and serial connections is provided.

Support for 768k RAM disk is provided. The RAM disk may be used for transferring OS-9 modules to the target system via FTP or Kermit.

Creating OS-9 68K Bootstrap Code for QUADS

To create bootstrap code for QUADS, complete the following steps:

Step 1. Change to the `\mwos\OS9\CPU32\PORTS\QUADS\ROM_CBOOT` directory and type the following on the command line:

```
os9make
```

Step 2. Burn the following image into a 27C010 EPROM:

```
\mwos\OS9\CPU32\PORTS\QUADS\CMDS\BOOTOBJS\ROMBUG\rombugger
```

If you do not want to use ROMBUG, you can burn the following image instead: \mwos\OS9\CPU32\PORTS\QUADS\CMDS\BOOTOBJS\NOBUG\rommer

Step 3. Connect a terminal to the board (such as Hyperterminal). Select the following settings:

9600 Baud 8Bits No Parity.

Step 4. Insert EPROM in socket U48.

Step 5. Apply power to the board. From here, the OS-9 bootstrap message is displayed. Type the following on the command line to bring up the bootmenu options:

g

OS-9/68K System Bootstrap

Once you have completed the steps above, the screen displays the following messages:

```
<Called>
Searching special memory list for symbol modules...
dn: 00000008 00002000  00000000 00000000   00000000 00000001
FFFFE000 004078F0
an: 00000A3E 00000480  00100000 00100000   00408900 00400400
FFFFFFFF 004078F0
pc: 00000940  sr:2700 (--S--7-----)t:OFF
usp:00000000   ^ssp^
0x00000940   >43FAFB82         lea.l 0x4C4(pc),a1
RomBug: g

BOOTING PROCEDURES AVAILABLE - <INPUT>

Boot from Ethernet ----------- <qe>
Boot from ROM ---------------- <C>
Restart the system ----------- <Q>

Select a boot method from the above menu:
```

Creating OS-9 68K QE (BOOTP Boot) Code for QUADS

To create 68K QE code for QUADS, complete the following steps:

Step 1. Obtain a suitable bootp server. Linux bootp server will work fine or a Windows based bootp server may be used if available. (See LINUX Example BOOTP Setup).

Step 2. Create a QE BOOT image by typing the following on the command line:

```
cd \mwos\OS9\CPU32\PORTS\QUADS\BOOTS

os9make
```

The created OS-9 QUADS boot image is placed in the following location: `\mwos\OS9\CPU32\PORTS\QUADS\CMDS\BOOTOBJS\BOOTFILES\qe_netboot.bf`

The above file should be copied to the bootp server machine or directory.

Step 3. Select the `QE` option from the QUADS OS-9 68k bootstap menu to boot.

Booting Procedures Available - <INPUT>

Upon selecting the qe option, the following window is displayed:

```
inetboot: (Edition 13) using Quads Ethernet @ 0x22C00 MTU=1500
inetboot: msgpcntl=404f70:16, ttcntl=404f72:16
inetboot: Bootp/Tftp sequence try: 1/3
bootp: 0:53:39:33:36:30 broadcasting for server..try 1/6
GOT BOOTP RESPONSE from server 192.168.0.13!
My IP address will be: 192.168.0.5 (0xc0a80005)
My tftp bootfile is: /tftpboot/os9boot
My bootfile size is: 1285 (512-byte) blocks
My subnet mask is: 255.255.255.0
tftp: fetching 1285 blocks
tftp: server response 192.168.0.13 port 1026
A valid OS-9 bootfile was found.
$
```

LINUX Example BOOTP Setup

To set up BOOTP using LINUX, follow the example steps below:

Step 1. Create `tftpboot` directory by typing the following in the command line:

```
mkdir /tftpboot
```

Step 2. FTP a boot image to Linux server and place in `tftpboot` directory. The name for this example is `os9boot`.

Step 3. Set permissions on bootfile by typing the following in the command line:

```
chmod 777 os9boot
```

Step 4. Set up the bootp table:

`/etc/bootptab` <= add entry for QUADS board.

.quads:ht=ethernet:hd=/tftpboot:bf=os9boot:bs=auto:\

sm=255.255.255.0:to=18000:ip=192.168.0.5: \

ha=005339333630:

Step 5. Enable `tftp` server

/etc/inetd.conf <= make sure tftp is enabled

tftp dgram udp wait root /usr/sbin/tcpd in.tftpd

Re-boot the Linux system if change is required here.

Step 6. Start bootp server by typing the following in the command line:

```
bootpd
```

If you prefer to debug, type the following in the command line:

```
bootpd -d7
```

Step 7. Apply power to QUADS board and select `qe` option. The bootfile should load and the `$` prompt should be displayed.

Customizing

This section discusses customizing parts of the boot.

ROM BOOTSTRAP

For customizing the ROM bootstrap code refer to the following directory:

```
\mwos\OS9\CPU32\PORTS\QUADS\ROM_CBOOT
```

You may customize the bootmenu by changing `syscon.c` and or `nvdftbl.a` files. Be sure to re-run `os9make` once the changes are made. Please refer to the following directory for switch settings and other definitions:

```
\mwos\OS9\CPU32\PORTS\QUADS\systype.d
```

QE BOOT (BOOTP IMAGE)

Edit the following file and remove or add the required modules.

```
\mwos\OS9\CPU32\PORTS\QUADS\BOOTLISTS\qe_netboot.bl
```

For More Information

See QE BOOT (BOOTP IMAGE) for information on re-making the boot image.

STARTING SoftStax (SPF) - EXAMPLE

```
$ ndbmod create inetdb3 11 400 0 160 0 0 0 100 0 400 65 256
$ ndbmod interface add enet0 address 192.168.0.5 netmask
255.255.255.0 binding /spqe0/enet
$ ipstart
$ ping 192.168.0.13
PING 192.168.0.13 (192.168.0.13): 56 data bytes
64 bytes from 192.168.0.13: ttl=255 time=5 ms
$ ftp 192.168.0.13
ftp> bin
ftp> get dir
ftp> quit
$ load -d dir free
$ dir -u
dir
```

Boot Image Modules—qe Boot

The following table lists the models included in the boot image of the default (qe) boot.

Table A-1 mdir -e

Address	Size	Owner	Perm	Type	Revs	Ed #	Lnk	Module Name
00408c00	27024	0.0	0555	Sys	a000	363	2	kernel
0040f590	5660	0.0	0555	Sys	a000	37	1	ioman
00410bac	414	1.0	0555	Sys	8000	50	0	init
00410d4a	12848	0.0	0555	Sys	a000	18	1	fpu
00413f7a	296	0.0	0555	Sys	a000	11	1	tk68360
004140a2	2280	0.0	0555	Fman	a000	51	2	scf
0041498a	174	0.0	0555	Driv	a000	1	0	null
00414a38	120	0.0	0555	Desc	8000	4	0	nil

Table A-1 mdir -e

Address	Size	Owner	Perm	Type	Revs	Ed #	Lnk	Module Name
00414ab0	3324	0.0	0555	Fman	a000	63	0	pipeman
004157ac	102	0.0	0555	Desc	8000	3	0	pipe
00415812	2132	0.0	0555	Driv	a000	5	2	sc68360
00416066	122	0.0	0555	Desc	8000	7	2	term
004160e0	9638	0.0	0555	Fman	a000	104	1	rbf
00418686	1284	0.0	0555	Driv	a000	24	1	ram
00418b8a	144	1.0	0555	Desc	8000	26	1	dd
00418c1a	144	1.0	0555	Desc	8000	26	0	r0
00418caa	210	1.0	0555	Prog	8000	10	1	sysgo
00418d7c	2866	0.0	0555	Sys	a000	30	1	sysmbuf
004198ae	3032	0.0	0555	Fman	e000	6	0	pkman
0041a486	4764	0.0	0555	Driv	e000	20	0	pkdvr
0041b722	108	0.0	0555	Desc	a000	4	0	pk
0041b78e	124	0.0	0555	Desc	a000	6	0	pks
0041b80a	17480	0.0	0555	Fman	a001	263	0	spf
0041fc52	2218	0.0	0555	Prog	a001	263	0	spf_rx
004204fc	51060	0.0	0555	Driv	a000	78	0	spip
0042cc70	662	0.0	0555	Desc	8000	4	0	ip0
0042cf06	21446	0.0	0555	Driv	a000	74	0	sptcp

Table A-1 mdir -e

Address	Size	Owner	Perm	Type	Revs	Ed #	Lnk	Module Name
004322cc	668	0.0	0555	Desc	8000	5	0	tcp0
00432568	7752	0.0	0555	Driv	a000	11	0	spudp
004343b0	664	0.0	0555	Desc	8000	6	0	udp0
00434648	5806	0.0	0555	Driv	a000	12	0	spraw
00435cf6	660	0.0	0555	Desc	8000	1	0	raw0
00435f8a	11410	0.0	0555	Driv	a000	7	0	sproute
00438c1c	654	0.0	0555	Desc	8000	1	0	route0
00438eaa	45148	0.0	0555	Trap	8000	226	0	netdb
00443f06	8010	0.0	0555	Driv	a000	37	0	spenet
00445e50	676	0.0	0555	Desc	8000	18	0	enet
004460f4	7220	0.0	0555	Driv	a000	21	0	sp360
00447d28	1070	0.0	0555	Desc	8000	14	0	spqe0
00448156	1728	0.0	0111	Data	8000	9	0	inetdb
00448816	3674	1.0	0555	Prog	c001	5	0	ipstart
00449670	18010	1.0	0555	Prog	c001	214	0	telnet
0044dcca	8130	0.0	0555	Prog	c001	217	0	telnetd
0044fc8c	14246	0.0	0555	Prog	c001	217	0	telnetdc
00453432	29398	1.0	0555	Prog	c001	221	0	ftp
0045a708	7650	0.0	0555	Prog	c001	227	0	ftpd

Table A-1 mdir -e

Address	Size	Owner	Perm	Type	Revs	Ed #	Lnk	Module Name
0045c4ea	21090	0.0	0555	Prog	c001	227	0	ftpdc
0046174c	17508	1.0	0555	Prog	c001	221	0	idbdump
00465bb0	16240	0.0	0555	Prog	c001	16	0	ndbmod
00469b20	20276	1.0	0555	Prog	c001	11	0	route
0046ea54	7394	1.0	0555	Prog	c001	204	0	hostname
00470736	25916	0.0	0555	Prog	c001	4	0	netstat
00476c72	9022	1.0	0555	Prog	c001	9	0	ping
00478fb0	2168	0.0	0555	Prog	c001	10	0	break
00479828	2578	1.0	0555	Prog	c001	23	0	date
0047a23a	2796	1.0	0555	Prog	c001	14	0	devs
0047ad26	2614	1.0	0555	Prog	c001	25	0	echo
0047b75c	2176	1.0	0555	Prog	c001	21	0	iniz
0047bfdc	5830	1.0	0555	Prog	c001	11	0	irqs
0047d6a2	17170	1.0	0555	Prog	c001	12	0	kermit
004819b4	5344	1.0	0555	Prog	c001	28	1	mdir
00482e94	4114	1.0	0555	Prog	c001	29	0	mfree
00483ea6	6718	1.0	0555	Prog	c001	29	0	procs
004858e4	4198	1.0	0555	Prog	c001	40	0	setime
0048694a	83850	1.0	0555	Prog	c001	135	1	mshell

Table A-1 mdir -e

Address	Size	Owner	Perm	Type	Revs	Ed #	Lnk	Module Name
0049b0d4	3754	1.0	0555	Prog	c001	30	0	pd
0049bf7e	46238	1.0	0555	Trap	c009	22	2	csl
004a741c	4374	1.0	0555	Prog	c001	37	0	attr
004a8532	3984	1.0	0555	Prog	c001	29	0	load

File List

```
\mwos\OS9\CPU32\PORTS\QUADS\defsfile
\mwos\OS9\CPU32\PORTS\QUADS\makefile
\mwos\OS9\CPU32\PORTS\QUADS\nv.h
\mwos\OS9\CPU32\PORTS\QUADS\systype.d
\mwos\OS9\CPU32\PORTS\QUADS\systype.h

\mwos\OS9\CPU32\PORTS\QUADS\CMDS\BOOTOBJS\sc68360
\mwos\OS9\CPU32\PORTS\QUADS\CMDS\BOOTOBJS\term
\mwos\OS9\CPU32\PORTS\QUADS\CMDS\BOOTOBJS\tk68360
\mwos\OS9\CPU32\PORTS\QUADS\CMDS\BOOTOBJS\ROMBUG\bootfile
\mwos\OS9\CPU32\PORTS\QUADS\CMDS\BOOTOBJS\ROMBUG\initext
\mwos\OS9\CPU32\PORTS\QUADS\CMDS\BOOTOBJS\ROMBUG\rombug
\mwos\OS9\CPU32\PORTS\QUADS\CMDS\BOOTOBJS\ROMBUG\rombugger
\mwos\OS9\CPU32\PORTS\QUADS\CMDS\BOOTOBJS\ROMBUG\utilities
\mwos\OS9\CPU32\PORTS\QUADS\CMDS\BOOTOBJS\NOBUG\bootfile
\mwos\OS9\CPU32\PORTS\QUADS\CMDS\BOOTOBJS\NOBUG\initext
\mwos\OS9\CPU32\PORTS\QUADS\CMDS\BOOTOBJS\NOBUG\rom
\mwos\OS9\CPU32\PORTS\QUADS\CMDS\BOOTOBJS\NOBUG\rommer
\mwos\OS9\CPU32\PORTS\QUADS\CMDS\BOOTOBJS\NOBUG\utilities
\mwos\OS9\CPU32\PORTS\QUADS\CMDS\BOOTOBJS\SPF\sp360
\mwos\OS9\CPU32\PORTS\QUADS\CMDS\BOOTOBJS\SPF\spqe0
```

\mwos\OS9\CPU32\PORTS\QUADS\CMDS\BOOTOBJS\SPF\inetdb
\mwos\OS9\CPU32\PORTS\QUADS\CMDS\BOOTOBJS\SPF\inetdb2
\mwos\OS9\CPU32\PORTS\QUADS\CMDS\BOOTOBJS\SPF\rpcdb
\mwos\OS9\CPU32\PORTS\QUADS\CMDS\BOOTOBJS\INITS\init
\mwos\OS9\CPU32\PORTS\QUADS\CMDS\BOOTOBJS\BOOTFILES\
qe_netboot.bf

\mwos\OS9\CPU32\PORTS\QUADS\BOOTLISTS\rom.bl
\mwos\OS9\CPU32\PORTS\QUADS\BOOTLISTS\rom_utilities.ml
\mwos\OS9\CPU32\PORTS\QUADS\BOOTLISTS\qe_netboot.bl

\mwos\OS9\CPU32\PORTS\QUADS\SPF\makefile
\mwos\OS9\CPU32\PORTS\QUADS\SPF\ETC\interfaces.conf
\mwos\OS9\CPU32\PORTS\QUADS\SPF\ETC\makefile
\mwos\OS9\CPU32\PORTS\QUADS\SPF\SPQUICC\DEFS
\mwos\OS9\CPU32\PORTS\QUADS\SPF\SPQUICC\makefile
\mwos\OS9\CPU32\PORTS\QUADS\SPF\SPQUICC\options.h
\mwos\OS9\CPU32\PORTS\QUADS\SPF\SPQUICC\spfdesc.mak
\mwos\OS9\CPU32\PORTS\QUADS\SPF\SPQUICC\spfdrvr.mak
\mwos\OS9\CPU32\PORTS\QUADS\SPF\SPQUICC\target.c
\mwos\OS9\CPU32\PORTS\QUADS\SPF\SPQUICC\DEFS\spf_desc.h

\mwos\OS9\CPU32\PORTS\QUADS\INIT\defsfile
\mwos\OS9\CPU32\PORTS\QUADS\INIT\init.make
\mwos\OS9\CPU32\PORTS\QUADS\INIT\makefile

\mwos\OS9\CPU32\PORTS\QUADS\SCF\defsfile
\mwos\OS9\CPU32\PORTS\QUADS\SCF\makefile
\mwos\OS9\CPU32\PORTS\QUADS\SCF\scf_descriptors.make
\mwos\OS9\CPU32\PORTS\QUADS\SCF\scf_drivers.make

\mwos\OS9\CPU32\PORTS\QUADS\SYSMODS\clock.make
\mwos\OS9\CPU32\PORTS\QUADS\SYSMODS\defsfile
\mwos\OS9\CPU32\PORTS\QUADS\SYSMODS\makefile

```
\mwos\OS9\CPU32\PORTS\QUADS\ROM_CBOOT\defsfile
\mwos\OS9\CPU32\PORTS\QUADS\ROM_CBOOT\initext.a
\mwos\OS9\CPU32\PORTS\QUADS\ROM_CBOOT\makefile
\mwos\OS9\CPU32\PORTS\QUADS\ROM_CBOOT\nvdftbl.a
\mwos\OS9\CPU32\PORTS\QUADS\ROM_CBOOT\rom.make
\mwos\OS9\CPU32\PORTS\QUADS\ROM_CBOOT\rom_bootfile.make
\mwos\OS9\CPU32\PORTS\QUADS\ROM_CBOOT\rom_common.make
\mwos\OS9\CPU32\PORTS\QUADS\ROM_CBOOT\rom_image.make
\mwos\OS9\CPU32\PORTS\QUADS\ROM_CBOOT\rom_initext.make
\mwos\OS9\CPU32\PORTS\QUADS\ROM_CBOOT\rom_port.make
\mwos\OS9\CPU32\PORTS\QUADS\ROM_CBOOT\rom_serial.make
\mwos\OS9\CPU32\PORTS\QUADS\ROM_CBOOT\rom_utilities.make
\mwos\OS9\CPU32\PORTS\QUADS\ROM_CBOOT\rombug.make
\mwos\OS9\CPU32\PORTS\QUADS\ROM_CBOOT\syscon.c
\mwos\OS9\CPU32\PORTS\QUADS\ROM_CBOOT\sysinit.a
\mwos\OS9\CPU32\PORTS\QUADS\ROM_CBOOT\oem.com

\mwos\OS9\CPU32\PORTS\QUADS\BOOTS\makefile
\mwos\OS9\CPU32\PORTS\QUADS\BOOTS\qe_netboot.make
```

MC68360 Reference Port

The reference port for the Motorola MC68360 processor is based on the Motorola M68360QUADS Board. This board contains dual XC68360 processors. The reference port contains drivers for the on chip periodic interrupt timer, scc based rs232 IO, and quicc Ethernet controller. Boards with either the AMD7992 or the Motorola 68160 Ethernet interface chips are supported. In OEM packages, source code is provided for the supported devices.

The bp_tk360.l and bp_quads.l libraries may be remade by executing make as shown below.

Library	Directory	Command
bp_tk360.l	MWOS/OS9/SRC/ROM/CBOOT/TIMERS	make -f=bp_tk360.make
bp_quads.l	MWOS/OS9/SRC/ROM/CBOOT/NETWORK/BOOTQUICC	make

The makefile in the ports directory (MWOS/OS9/CPU32/PORTS/QUADS) calls additional makefiles to recreate the modules and ROM code for the reference port. These makefiles also call additional makefiles. A list of makefiles is shown below.

Table A-2 Quads PORT Directory Makefiles

Makefile Called	Makefile Called	Makefile Called	Comment
INIT/makefile			
	INIT/init.make		Creates Init module
SYSMODS/makefile			
	SYSMODS/clock.make		Creates tk360
SCF/makefile			
	SCF/scf_drivers.make		Create sc360 driver
	SCF/scf_descriptors.make		Create Term descriptor

Table A-2 Quads PORT Directory Makefiles (continued)

Makefile Called	Makefile Called	Makefile Called	Comment
SPF/makefile			
	SPF/ETC/makefile		Creates inetdb, inetdb2
	SPF/SPQUICC/makefile		
		SPF/SPQUICC/spfdrvr.mak	Creates spquicc
		SPF/SPQUICC/spf_desc.mak	Creates spqe0
ROM_CBOOT/makefile			
	ROM_CBOOT/rom.make		Creates CMDS/BOOTOBJS/ROM/rommer ROM image without rombug
		ROM_CBOOT/ROM_common.make	Creates rom_common.l (vectors, boot,nvdftbl)
		ROM_CBOOT/ROM_serial.make	Creates rom_serial.l console driver library (io68360)
		ROM_CBOOT/ROM_port.make	Creates rom_port.l port specific library (sysinit, syscon)
		ROM_CBOOT/ROM_image.make	Creates CMDS/BOOTOBJS/ROM/rombug (BLS CBOOT core distribution file)
		ROM_CBOOT/ROM_initext.make	Creates CMDS/BOOTOBJS/ROM/initext User sysinit extension binary
		ROM_CBOOT/ROM_bootfile.make	Creates CMDS/BOOTOBJS/ROM/bootfile rom based bootfile
	ROM_CBOOT/rombug.make		Creates CMDS/BOOTOBJS/ROM/rombugger ROM image with rombug
		ROM_CBOOT/ROM_common.make	Creates rom_common.l (vectors, boot,nvdftbl)
		ROM_CBOOT/ROM_serial.make	Creates rom_serial.l console driver library (io68360)

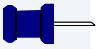
Table A-2 Quads PORT Directory Makefiles (continued)

Makefile Called	Makefile Called	Makefile Called	Comment
		ROM_CBOOT/ROM_port.make	Creates rom_port.l port specific library (sysinit, syscon)
		ROM_CBOOT/ROM_image.make	Creates CMDS/BOOTOBJS/ROMBUG/rombug (BLS CBOOT core distribution file)
		ROM_CBOOT/ROM_initext.make	Creates CMDS/BOOTOBJS/ROMBUG/initext User sysinit extension binary
		ROM_CBOOT/ROM_bootfile.make	Creates CMDS/BOOTOBJS/ROMBUG/bootfile rom based bootfile

The IP address is set with `ifconfig`.

For More Information

See the ***Using LAN Communications Pak*** for more information.



Note

The makefile `MWOS/OS9/CPU32/PORTS/QUADS/SPF/SPQUICC/makefile` makes the sp360 driver and descriptor.

Note

The QUADS board uses software to set the Ethernet hardware addresses. When multiple QUADS board are in use on the same network, appropriate Ethernet addresses MUST be assigned to each board. The addresses are declared in the file `nvdftbl.a` which becomes part of the code within the ROM set for each board. QUADS boards should have a power supply which provides a full 5 Volt power source measured at the board side of the protective fuses on the board. Low voltages may produce erratic performance with the Ethernet interface.

Appendix B: SCSI Driver Configuration

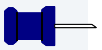
This appendix discusses the SCSI driver configuration. It includes the following sections:

RadiSys.
MICROWARE SOFTWARE

SCSI Driver

ZIP™ and JAZ™ drives inter-operate with the RBVCCS driver when both synchronous operation and disconnect support are disabled.

RBTEAC - Format Table



Note

This support if available for OEM package customers only.

Support for the TEAC SCSI floppy drive has been a complex issue. This is mainly due to the differences seen in drives released to different countries. To add to the complex issue of drive differences is the lack of information on what TEAC is doing in other areas of the world. For example; a FC-1 released in the USA will have different pin programing than one released in Japan and perhaps Germany or other areas of the world. A study of some Japanese versions has provided some insight as to the differences.

The high-level driver used after the system boots does an **inquiry** of the drive to determine the origin of the drive. The **inquiry** also helps to identify the setup required for the drive model for formats other than OS-9 Universal Format.

Unlike the high-level driver, the **booter** code has not done an **inquiry** of the drive. This has caused some problems even when using OS-9 Universal Format.

With **FC-5** drives, this problem has become even more complex to support world-wide.

Tables For Pin Requirements

With edition 29 of the RBTEAC driver, the support has been simplified through tables. These tables describe the pin differences between different models. To add support for a new drive, the table only need to be updated, the code does not have to change.

OS-9 for 68k and OS-9000 use the same table for support of both the high-level drivers and the boot system.

Booters for both OS-9 for 68k and OS-9000 will do an **inquiry** of the drive for proper setup.

OS-9 for 68k still only boots OS-9 Universal Format by default; however, you may change the boot table to boot higher density formats.

OS-9000 includes boot table formats to 2.88 MB for the TEAC SCSI Floppy driver.

Using the new table format, you can now add new drivers more easily. All that is required is that you determine the pin requirements and add the information to the drive tables. You will also need to re-compile the sources, both high-level and booters. The table is located in `MWOS/SRC/DEFS/IO/SCSI/teactab.h`.

Technical information related to OS-9 Universal Format prior to edition 29:

Level	trdy	pin1	pin4	pin2	pin34	medtype
OS-9 Boot	00	00	00	05	02	16
OS-9 High-Level-USA	00	00	00	05	02	16
OS-9 High-Level-WORLD	00	00	00	0D	02	16

Technical information related to OS-9 Universal Format edition 29:

Drive	trdy	pin1	pin4	pin2	pin34	medtype
FC-1 GF : USA	00	00	00	05	02	16
FC-1 GF : WORLD	00	00	00	0D	02	16
FC-1 HF : USA	00	00	00	05	02	16
FC-1 HF : WORLD	00	00	00	0D	02	16
FC-1 JHF: USA	00	00	00	05	02	16
FC-1 JHF: WORLD	00	00	00	0D	02	16
FC-1 JGF: WORLD	00	00	00	0D	02	16
FC-1 HGF: WORLD	00	00	00	D	02	16
FC-5 HGF: WORLD	80	03	07	0D	02	16

TEAC Tables

```
/*
 * Teac drive differences are now handled via tables
 * to add support for a new Teac drive simply include the
 * pin setup information as described in the "teac_fmts"
 * structure.
*/

typedef const struct teac_fmts {
     char       const     *dseries;           /* drive series FC-1 FC-5 */
     char       const     *dtype;             /* drive type "HF" "JHF" */
     u_int8     const     fc_firm;            /* firmware origin 0=USA 1=WORLD */
     u_int8     const     trdy;               /* ready state */
     u_int8     const     pin4;               /* pin1 and pin4 */
     u_int8     const     pin34_250;          /* pin2 and pin34 - rate 250 */
     u_int8     const     medtype_250;        /* media type - rate 250 */
     u_int8     const     pin34_500;          /* pin2 and pin34 - rate 500 */
     u_int8     const     medtype_500;        /* media type - rate 500 */
     u_int8     const     pin34_1000;         /* pin2 and pin34 - rate 1000 */
     u_int8     const     medtype_1000;       /* media type - rate 1000 */
} teac_fmts, *Teac_fmts;


#define MAX_TEAC_FMTS 100

teac_fmts const teac_desc[] = {

     /* Spare type : ( User Patchable ) */

     { "XX-X", "XXX", 0xa0,
          0xa1, 0xa2, 0xa3, 0xa4, 0xa5, 0xa6, 0xa7, 0xa8,     },

     /* GF type : USA Version (untested) */

     { "FC-1", "GF ", 1,
          0,                                 /* trdy */
          (MS_PIN4 << 4)  | MS_PIN1,         /* pin 1 & pin 4 */
          (MS_PIN34 << 4) | MS_PIN2LD,       /* Pin2 and Pin 34 Rate:250 */
          MS_LD,                             /* Media Type 250 */
          (MS_PIN34 << 4) | MS_PIN2HD,       /* Pin2 and Pin 34 Rate:500 */
          MS_HD,                             /* Media Type 500 */
          0xff,                              /* (NA) Pin2 and Pin 34 Rate:1000 */
          0xff,                              /* (NA) Media Type 1000 */
          },

     /* GF type : World Version (tested) */

     { "FC-1", "GF ", 0,
          0,                                 /* trdy */
          (MS_PIN4 << 4)  | MS_PIN1,         /* pin 1 & pin 4 */
          (MS_PIN34 << 4) | MS_PIN2HD,       /* Pin2 and Pin 34 Rate:250 */
          MS_LD,                             /* Media Type 250 */
          (MS_PIN34 << 4) | MS_PIN2HD,       /* Pin2 and Pin 34 Rate:500 */
          MS_HD,                             /* Media Type 500 */
          0xff,                              /* (NA) Pin2 and Pin 34 Rate:1000 */
          0xff,                              /* (NA) Media Type 1000 */
          },

     /* HF type : USA Version (tested) */

     { "FC-1", "HF ", 1,
          0,                                 /* trdy */
          (MS_PIN4 << 4)  | MS_PIN1,         /* pin 1 & pin 4 */
          (MS_PIN34 << 4) | MS_PIN2LD,       /* Pin2 and Pin 34 Rate:250 */
          MS_LD,                             /* Media Type 250 */
          (MS_PIN34 << 4) | MS_PIN2HD,       /* Pin2 and Pin 34 Rate:500 */
          MS_HD,                             /* Media Type 500 */
          0xff,                              /* (NA) Pin2 and Pin 34 Rate:1000 */
```

```
     0xff,                                  /* (NA) Media Type 1000 */
     },

/* HF type : World Version (tested) */

{ "FC-1", "HF ", 0,
     0,                                     /* trdy */
     (MS_PIN4 << 4)   | MS_PIN1,            /* pin 1 & pin 4 */
     (MS_PIN34 << 4) | MS_PIN2HD,           /* Pin2 and Pin 34 Rate:250 */
     MS_LD,                                 /* Media Type 250 */
     (MS_PIN34 << 4) | MS_PIN2HD,           /* Pin2 and Pin 34 Rate:500 */
     MS_HD_ER,                              /* Media Type 500 */
     0xff,                                  /* (NA) Pin2 and Pin 34 Rate:1000 */
     0xff,                                  /* (NA) Media Type 1000 */
     },

/* JHF type : USA Version (untested) */

{ "FC-1", "JHF", 1,
     0,                                     /* trdy */
     (MS_PIN4 << 4)   | MS_PIN1,            /* pin 1 & pin 4 */
     (MS_PIN34 << 4) | MS_PIN2LD,           /* Pin2 and Pin 34 Rate:250 */
     MS_LD,                                 /* Media Type 250 */
     (MS_PIN34 << 4) | MS_PIN2LD,           /* Pin2 and Pin 34 Rate:500 */
     MS_HD,                                 /* Media Type 500 */
     (MS_PIN34 << 4) | MS_PIN2HD,           /* Pin2 and Pin 34 Rate:1000 */
     MS_ED,                                 /* Media Type 1000 */
     },

/* JHF type : World Version (tested) */

{ "FC-1", "JHF", 0,
     0,                                     /* trdy */
     (MS_PIN4 << 4)   | MS_PIN1,            /* pin 1 & pin 4 */
     (MS_PIN34 << 4) | MS_PIN2HD,           /* Pin2 and Pin 34 Rate:250 */
     MS_LD,                                 /* Media Type 250 */
     (MS_PIN34 << 4) | MS_PIN2LD,           /* Pin2 and Pin 34 Rate:500 */
     MS_HD_ER,                              /* Media Type 500 */
     (MS_PIN34 << 4) | MS_PIN2HD,           /* Pin2 and Pin 34 Rate:1000 */
     MS_ED,                                 /* Media Type 1000 */
     },

/* JGF type : World Version (tested) */

{ "FC-1", "JGF", 0,
     0,                                     /* trdy */
     (MS_PIN4 << 4)   | MS_PIN1,            /* pin 1 & pin 4 */
     (MS_PIN34 << 4) | MS_PIN2HD,           /* Pin2 and Pin 34 Rate:250 */
     MS_LD,                                 /* Media Type 250 */
     (MS_PIN34 << 4) | MS_PIN2HD,           /* Pin2 and Pin 34 Rate:500 */
     MS_HD,                                 /* Media Type 500 */
     (MS_PIN34 << 4) | MS_PIN2HD,           /* Pin2 and Pin 34 Rate:1000 */
     MS_ED,                                 /* Media Type 1000 */
     },

/* FC-1 HGF type : World Version (untested) */

{ "FC-1", "HGF", 0,
     0,                                     /* trdy */
     (MS_PIN4 << 4)   | MS_PIN1,            /* pin 1 & pin 4 */
     (MS_PIN34 << 4) | MS_PIN2HD,           /* Pin2 and Pin 34 Rate:250 */
     MS_LD,                                 /* Media Type 250 */
     (MS_PIN34 << 4)     | MS_PIN2HD,       /* Pin2 and Pin 34 Rate:500 */
     MS_HD_ER,                              /* Media Type 500 */
     0xff,                                  /* (NA) Pin2 and Pin 34 Rate:1000 */
     0xff,                                  /* (NA) Media Type 1000 */
     },
```

```
    /* FC-5 HGF type : World Version (tested) */

    { "FC-5", "HGF", 0,
         0x80,                                  /* trdy */
         (MS_PIN4_LS << 4)  | MS_PIN1_1_6M,     /* pin 1 & pin 4 */
         (MS_PIN34 << 4) | MS_PIN2HD,           /* Pin2 and Pin 34 Rate:250 */
         MS_LD,                                 /* Media Type 250 */
         (MS_PIN34 << 4)      | MS_PIN2HD,      /* Pin2 and Pin 34 Rate:500 */
         MS_HD_ER,                              /* Media Type 500 */
         0xff,                                  /* (NA) Pin2 and Pin 34 Rate:1000 */
         0xff,                                  /* (NA) Media Type 1000 */
         },

    /* WARNING: DO NOT DELETE THE FOLLOWING ENTRY. */

    { "END ", "END ", 0, 0, 0, 0, 0,     0, 0, 0, 0,     }

};
```

Product Discrepancy Report

To: Microware Customer Support

FAX: 515-224-1352

From:_____

Company:_____

Phone:_____

Fax:_____Email:_____

Product Name:

Description of Problem:

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Host Platform_____

Target Platform_____

RadiSys.

MICROWARE SOFTWARE