

Release

Notes

for

**OS-9 for 68K
Processors**

Version 3.0.3

microware



Copyright and Publication Information

Copyright ©1996 Microware Systems Corporation. All Rights Reserved. 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 Microware Systems Corporation.

This manual reflects version 3.0.3 of OS-9 for 68K processors.

Revision: A
Publication date: October 1996
Product Number: 1233-0022 / OPS68303IRN

Disclaimer

The information contained herein is believed to be accurate as of the date of publication. However, Microware 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. Microware 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 Microware and/or other parties. Unauthorized distribution of software may cause damages far in excess of the value of the copies involved.

For additional copies of this software/documentation, or if you have questions concerning the above notice, please contact your OS-9 supplier.

Trademarks

OS-9, OS-9000, DAVID, FasTrak, and UpLink are registered trademarks of Microware Systems Corporation. SoftStax is a trademark of Microware Systems Corporation. All other product names referenced herein are either trademarks or registered trademarks of their respective owners.

Address

Microware Systems Corporation
1500 N.W. 118th Street
Des Moines, Iowa 50325
515-223-8000



Table of Contents

Chapter 1: Overview and Installation Notes **7**

- 8 Overview
- 9 Installation Notes
 - 9 New Installation
 - 10 3.0.3 Upgrade Installation

Chapter 2: Version 3.0.1 Changes **11**

- 12 Overview
- 13 System Module Changes
 - 13 Kernel
 - 15 Kernel Patches for Cold Start Cache Disable
 - 17 Init Module
 - 18 Moded.Fields
 - 18 Boot Code
 - 18 boot.a
 - 19 romboot.c
 - 19 BootP Update
- 20 I/O System Changes
 - 20 IOMan
 - 20 SCF
 - 21 Pipeman
 - 21 RBF
 - 21 Floppy Disk Formats
 - 22 SCSI System
 - 24 SCSI 327 Driver Enhancements



25	Utility Changes
26	Internet Changes
26	IP Module
26	Inetboot.I Library
26	ISP Enhancement
27	Ethernet Drivers for MVME162 and MVME167
27	tcp
27	IFSLIP
28	SysMbuf Changes
29	Example Port for 68360 Quads Board Addition

Chapter 3:Version 3.0.2 Changes

31

32	Overview
33	System Module Changes
33	Kernel
34	SysCache
35	SSM
35	System Definitions
36	Moded.Fields
36	Init Module
37	I/O System Changes
37	IOMan
37	Pipeman
37	SBF
38	SCF
38	SCSI Driver
39	NCR53C710 SCSI Driver
39	SBF Device Descriptor
39	SCF Driver
40	Clock Drivers
41	ROM Changes

41	CBoot
41	ROM Common Code
41	RomBug Debugger
42	ROM Serial Drivers
42	rompak.m Macro
42	ROM CBoot
43	ROM Definitions
43	Hardware Definitions
44	System Error Message Changes
45	MVME147 Example Port Changes
47	IF177 Driver Changes
48	68060 Mask Issues

Chapter 4:Version 3.0.3 Changes

49

50	Overview
51	System Module Changes
51	Kernel
53	SSM
53	System Definitions
54	Moded.Fields
54	Boot Code
56	I/O System Changes
56	Real-Time Clock
56	IOMAN
57	RBF
57	RamDisk
58	SCF
59	Pipe
59	SCF Driver
59	BCC332 Example Port
60	SCSI Driver



61 Utility Changes

Chapter 5:Known Problems **65**

- 66 Intel I82596 Driver
- 67 Debuggers
 - 67 System-State Debugger
 - 67 RomBug
- 68 System-State Time Slicing

Appendix A:Example Ports for M68360 Quads Boards **69**

Appendix B:SCSI Driver Configuration **75**

- 76 SCSI Driver Note
- 77 RBTEAC - Format Table Notes
 - 77 Tables For Pin Requirements
- 79 TEAC Tables

Product Discrepancy Report **83**

Chapter 1: Overview and Installation Notes

Overview

The files contained in this release of OS-9 are provided as updates to existing Version 3.0, 3.0.1, and 3.0.2 Developer's Kit packages. These files are intended to be installed **on top of** your existing Version 3.0, 3.0.1, or 3.0.2 file system.

This Version 3.0.3 Update contains all updates for the OS-9 Developer's Kits since the Version 3.0 release of OS-9, in that it contains all files released with the Version 3.0.1 and 3.0.2 updates as well as additional file updates since Version 3.0.2. Whether you are updating from Version 3.0, 3.0.1, or 3.0.2 of OS-9, you will only need to use this package to bring your system up to Version 3.0.3 status.

If this is the first 3.0 Version of the OS-9 Developer's Kit you have received, you may be interested in the changes explained in all the chapters of these release notes.

If you have already installed Version 3.0.1 of the OS-9 Developer's Kit and are upgrading to Version 3.0.3, you should refer to **Chapters 1, 3, 4, and 5** of these release notes.

If you have already installed Version 3.0.2 of the OS-9 Developer's Kit and are upgrading to Version 3.0.3, you should refer to **Chapters 1, 4, and 5** of these release notes.

Installation Notes

The ***OS-9 Developer's Kit Version 3.0.3 Installation Instructions*** included in your distribution package contain instructions intended to aid in the installation of the OS-9 Developer's Kits. Instructions are included both for customers who have never previously licensed Version 3.0 and those who wish to update to the latest release.

New Installation

If this is the first OS-9 Version 3.0 Series Developer's Kit you have received, please note the following special installation instructions:

-
- Step 1. Install the ***V3.0 OS-9 Developer's Kit*** first.
 - Step 2. Install the ***V1.3 OS-9 C Toolkit*** or ***V1.0 OS-9 C++ Toolkit*** next.
 - Step 3. Install the ***V3.0.3 Developer's Kit Update***.
 - Step 4. If you want to use ISP Version 2.1, you must install it after the ***V3.0.3 Developer's Kit Update***, even if you have previously installed ISP Version 2.1.
 - Step 5. Specify the same destination directory (`install to`) pathlist for each archive.
-

3.0.3 Upgrade Installation

If you have already installed Version 3.0 of the Developer's Kit and are now upgrading to Version 3.0.3, you should note these special installation instructions.

The files contained in this release of OS-9 are provided as updates to existing Version 3.0 Developer's Kit packages. These files are intended to be installed **on top of** your existing Version 3.0 file system.



Note

The Version 3.0.3 Developer's Kit is intended to be installed over a Version 3.0, 3.0.1, or 3.0.2 Developer's Kit. You must also have Version 1.3 or Version 1.0 of the Toolkit installed before you install the Version 3.0.3 Developer's Kit update. Also, you must install, or re-install, ISP Version 2.1 after you have installed the **V3.0.3 Developer's Kit Update**.

Chapter 2: Version 3.0.1 Changes

Overview

The following notes detail the changes made in the Version 3.0.1 update release of OS-9/68000. If this is the first release of OS-9 you have received, you may want to read all of the chapters in these release notes to get a complete list of software changes since the OS-9 Version 3.0 release.

System Module Changes

Kernel

Changes to the kernel include:

- 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$PModule`) 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 on 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

The following changes have been made 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

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$SigReset` more than 5 times, the system may become damaged. The solution for this problem has been provided in the following two ways:
 - Use the Version 3.0.1 kernel. This kernel contains the updated `F$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` will ONLY work on Version 3.0 kernels. If you attempt to use it with any other version of the kernel the initialization part of the module will fail 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$NoClk` (No Clock) as the error when the `Init` module does not specify any clock module offset. Additionally, if the clock start-up during kernel cold start fails, the development version kernels issue a warning message to the console.
- The kernel cold start could become confused when locating the `Init` module's module names beginning with the string `init` (for example, `initsys`) are now no longer confused with `init`.
- Kernels supporting the 68882 Floating Point Co-Processor (for example, 68020 and 68030 kernels) now correctly recognize a potential co-processor protocol violation when a floating point exception occurs.
- The expansion stack fields (`P$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.

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.

Moded.Fields

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

Boot Code

Changes have been made to the following boot code files:

- `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 Update

An example `BOOTP` database is provided in the directory `MWOS/OS9/SRC/IO/INET/UTILS/BOOTP/BOOTPD`.

I/O System Changes

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.

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.

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.

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)	
<code>pcdos580</code>	512	80	15	360rpm	500KB/s	1.2MB	2.0MB
<code>pcdos380</code>	512	80	18	300rpm	500KB/s	1.44MB	2.0MB
<code>pcdos380ed</code>	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- and low-level drivers for SCSI have been modified as follows:

- 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`, `rbscs`, 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 Enhancements

This driver has been updated as follows:

- 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 Changes

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

Updated `cs1` modules are included for proper operation of the `fixmod` and `ident` utilities.

Internet Changes

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 BootP client use Sun Solaris 2.3 as a BootP server.

ISP Enhancement

The header file `<MWOS/SRC/DEFS/INET/un.h>` has been added in support of `AF_UNIX` sockets.

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 Changes

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 Addition

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** of these release notes.

2

Version 3.0.1 Changes



Chapter 3: Version 3.0.2 Changes

Overview

The following notes detail the changes made in the Version 3.0.2 update of OS-9 since the Version 3.0.1 update. Details of the changes made at Version 3.0.1 are explained in the **Version 3.0.1** chapter of these release notes.

System Module Changes

Kernel



For More Information

See the *68060 Mask Issues* section for more information on 68060 support.

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

- Added support for the 68060.



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$Send` will now send the signal to the highest process ID.
- System-state time-slicing checks are now more reliable.
- The semaphore (P operation) code now correctly indicates an error when it is awakened (prematurely) by a signal. Previously, the error code was being returned correctly (`E$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

Added support 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

Added `scsi_wide` descriptions for RBF and SCF descriptors.

Added `68681 DevCon` description for `sc68341` driver.

Added `Append` mode description for NFM device descriptors.

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.

I/O System Changes

The following changes have been made to the 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) now 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.

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 Changes

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:

- Added support for 68060 processor.
- Repaired problems with short branches going out-of-range when `MBUGTRC` was defined.
- Improved the memory testing routines 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 `iodcd2401.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 `nvr.am.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 `nvr.am.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:

- Resolving definition conflicts with `mcecc.d`
- Replacing 33Mhz definitions with 32Mhz definitions
- Additional 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 Changes

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 Changes

The `systype.d` file has been updated for:

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

The `sysinit.a` file has been updated to convert the `BLOAD` 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:

- Modified to provide for the definition of `RAMLOAD` instead of `BLOAD`.
- 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`).

3

Version 3.0.2 Changes



- Modified to check utilities area size (`rom_utilities.make`).
- Modified to fix title comments.

IF177 Driver Changes

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 except for changes required for ISP Version 2.1 development.

Due to directory location and definition changes, the source form will be provided in ISP Version 2.1.

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 4: Version 3.0.3 Changes

Overview

The following notes detail the changes made to OS-9 since the Version 3.0.2 update. Details of the changes made at Version 3.0.2 and 3.0.1 are explained in the **Version 3.0.2** and **Version 3.0.1** chapters of these release notes.

System Module Changes

The following changes have been made to the kernel since Version 3.0.2.

Kernel

The following changes have been made to the 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, and thus 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 unfindable.
- The **kernel** now rejects System Calls (OS-9 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$Mem` has been repaired.
- It was possible for non-MSP kernels (in other words, all kernels except those for the 68020/68030/68040) to crash the system when an exception occurred in system-state while interrupts were masked. The **kernel** would consider itself in an interrupt-service context (where exceptions are fatal) when that was not always the case. [This case could be provoked, for example, by having a driver mask interrupts and then probe for a non-existing device. The correct response for this type of situation would be to return the error to the caller.]
- The **kernel's** interrupt service code has been revised for robustness and speed improvements.
- It was possible for a complex name pattern match (using an asterisk, (*) or question mark (?) in `F$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.

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 changes have been made to 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

Changes to the boot code include:

- `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.



Note

The `inetboot.l` library with these changes has been provided in binary form only. The sources for this library require an ISP V.2.1 base which does not ship with OS-9 V.3.0 packages. Therefore, the sources provided in the OS-9 Version 3.0.3 Developer's Kit do not include these fixes.

I/O System Changes

Real-Time Clock

Changes to the real-time clock include:

- 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.
- The following RTC modules were updated in this release:
`rtc1216E.a`, `rtc146818.a`, `rtc48T02.a`, `rtc48T08.a`
and `rtc58274.a`.

IOMAN

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

RBF

Changes to RBF include:

- 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 will alleviate any rollover issues when the year 2000 occurs. RBF media dates are now unambiguous to December 31, 2155.

RamDisk

Changes to Ram Disk include:

- 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

Changes to SCF include:

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

Pipe

Change to Pipe include:

- 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 following changes were made to the 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 Changes

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

- `attr`
(OS-9000 only) Fixed so that the `-g` options are now correctly ignored when `-n` is used. `-g` options are not used on OS-9. The OS-9 version of `attr` ignores these options for compatibility with OS-9000 procedure files, for example. (PDR 95359)
- `binex`
Added a `-z[[=<file>]` option.
- `build`
Fixed to work properly when used from a procedure file.
- `cfp`
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`
Fixed a memory allocation problem and corrected to properly copy a file from a pipe device.
- `dir`
Fixed to print dates correctly after the year 2000. Fixed sector number and file size printouts for sectors/files greater than 2 GB.
- `diskcache`
Fixed to properly close its standard I/O paths. Fixed to free its data structures more safely.
- `fixmod`
Added recognition of more Microware module formats (OS-9000/PowerPC and OS-9000/386 Version 2.0).

- `format`
Fixed to work properly for disks larger than 2 GB.
- `free`
Fixed to more correctly handle NFS devices.
- `frestore`
Fixed to handle the year 2000.
- `fsave`
Fixed to handle the year 2000. Fixed summary printout to handle backups greater than 2 GB.
- `help`
Corrected infinite loop when help requested on built-in shell commands.
- `ident`
Added `-a` option to print alternative information for quick (`-q`) output.
- `irqs`
Changed interpretation of system data structures for the Version 3.0.3 implementation of **fast** IRQs.
- `login`
Added `-n` option to support ftp and enable user-defined replacements of login technology. Fixed to handle the year 2000.
- `make`
Added C++ support. `make` no longer discards the line after an include line.
- `merge`
Fixed `-z` file handling to trim whitespace from head and tail of each line as well as continuing after reading a blank line.
- `mfree`
Fixed to support printing of any number of segments.

- `os9gen`
Fixed to support device names of exactly 6 characters and to support disks greater than 2 GB.
- `pr`
Fixed to support the year 2000.
- `romsplit`
Fixed to properly support `-x`.
- `setime`
Fixed to support the year 2000.
- `tapegen`
Fixed to support the year 2000.
- `tmode`
New baud rates from 3.0.3 added.
- `tsmon`
Fixed to support the year 2000.
- `xmode`
New baud rates from 3.0.3 added.

4 Version 3.0.3 Changes



Chapter 5: Known Problems

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.

As the current version of SSM does not implement system-state page tables (and thus fine resolution of supervisor-state cache modes), the following options are available for I82596 operation with a 68060:

- Implement a system-state page table module allowing the MBuf pool to be placed in `WriteThrough` memory (the remaining memory can then be any mode you desire).
- Choose the `SSM060` module instead of the `SSM060_CBSUP` module for the system's memory management functions. User-state cache modes can still be controlled via the `Init` module's `CacheList` entries.



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.

Debuggers

System-State Debugger

The current release of the System-State debugger (`SysDbg`, edition #79) does NOT operate correctly on the 68060. You should 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.)

Thus, 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
```

Appendix A: Example Ports for M68360 Quads Boards

The table below shows a list of the directories that are included in the example port and briefly describes the contents of each directory.

Directory	Directory Contents
MWOS/OS9/CPU32/PORTS/QUADS	Master QUADS Ports directory
MWOS/OS9/SRC/IO/SCF/DRVR	Source directory for sc68360
MWOS/OS9/SRC/IO/INET/DRVR/QUICC	Source directory for qedvr Ethernet driver and stat program
MWOS/OS9/SRC/SYSMODS/GCLOCK	Source directory for system tick driver: tk68360
MWOS/OS9/SRC/ROM/LIB	ROM libs for BootP driver and timer routines
MWOS/OS9/SRC/ROM/SERIAL	Source directory for ROM console serial driver: io68360.a

A

Example Ports for M68360 Quads Boards



MWOS/OS9/SRC/ROM/CBOOT/TIMERS

Source directory for the
BootP timer:

bp_tk360.l

MWOS/OS9/SRC/ROM/CBOOT/NETWORK/BOOTQUICC

Source directory for the
Ethernet boot driver:

bp_quads.l

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. Binaries and source code are provided for the supported devices.

The bp_tk360.l and bp_quads.l libraries may be remade by executing make as shown in the table 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. Refer to the table below for a list of makefiles used.

Makefile Directory	Makefile Called	Action
all.make:	clock.make	creates tk68360 ticker driver

<code>init.make</code>	creates term scf descriptor
<code>scf_descriptors.make</code>	creates term scf descriptor
<code>scf_drivers.make</code>	creates sc68360 scf drive
<code>qedvr.make</code>	creates qedvr Ethernet driver
<code>isp.make</code>	calls makes in ISP directory to create socket descriptor, ipconfig modules, and qe0 descriptor
<code>rombug.make :</code>	creates ROM object containing RomBug
<code>rom_common.make</code>	creates rom_common.l (vectors, boot, nvdftbl)
<code>rom_serial.make</code>	creates rom_serial.l console driver library (io68360)
<code>rom_port.make</code>	creates rom_port.l port specific library (sysinit, syscon)

A

Example Ports for M68360 Quads Boards



<code>rom_image.make</code>	creates combined ROM image as distributed with BSPs
<code>rom_initext.make</code>	creates <code>initext.r</code> containing end user definable initialization extensions to the ROM
<code>rom_bootfile.make</code>	creates <code>boot</code> as included in BSP ROM sets This make is not used on the Quad board due to limited ROM space but may be used to create BSP style ROMs on boards with adequate ROM space.
<code>rom_utilities.make</code>	creates <code>utilities</code> as included in BSP ROM sets This make is not used on the Quad board due to limited ROM space but may be used to create BSP style ROMs on boards with adequate ROM space.

<code>rom.make :</code>	creates ROM object without RomBug
<code>rom_common.make</code>	creates <code>rom_common.l</code> (vectors, boot, <code>nvdftbl</code>)
<code>rom_serial.make</code>	creates <code>rom_serial.l</code> console driver library (<code>io68360</code>)
<code>rom_port.make</code>	creates <code>rom_port.l</code> port specific library (<code>sysinit</code> , <code>syscon</code>)
<code>rom_image.make</code>	creates combined ROM image as distributed with BSPs
<code>rom_initext.make</code>	creates <code>initext.r</code> containing end user definable initialization extensions to the ROM
<code>rom_bootfile.make</code>	creates boot as included in BSP ROM set
<code>rom_utilities.make</code>	creates utilities as included in BSP ROM sets

A

Modify files in the ISP subdirectories to set IP addresses for the board.



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

B

SCSI Driver Note

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

RBTEAC - Format Table Notes

Support for the TEAC SCSI floppy drive has been a complex issue. This is mainly due to the differences seen in drives released to other 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 programming than one released in Japan and perhaps Germany or other areas of the world. A study of some Japanese versions have 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 and OS-9000 use the same table for support of both the high-level drivers and the boot system.

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

OS-9 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 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_fmfts"
 * structure.
 */

typedef const struct teac_fmfts {
    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_fmfts, *Teac_fmfts;

#define MAX_TEAC_FMFTS 100

teac_fmfts 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_ED, /* Media Type 1000 */
},

```


B SCSI Driver Configuration



