

Enhanced OS-9 for 68K Release Notes

Version 1.1



MICROWARE™
Intelligent Products For A Smarter World

Copyright and Publication Information

Copyright ©2000 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 1.1 of Enhanced OS-9 for 68K.

Revision: B
Publication date: September 2000

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, and MAUI are registered trademarks of Microware Systems Corporation. SoftStax, FasTrak, UpLink, and Hawk are trademarks 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: Enhanced OS-9 for 68K V1.1 Release Notes **7**

- 8 Supported Reference Boards
- 8 68030
- 8 68040
- 8 68060
- 8 68328
- 8 68360 (OEM Package Only)
- 9 Enhancements
- 11 New Optional Extras
- 11 Features Removed

Chapter 2: OS-9 Operating System Version 3.2 Release Notes **13**

- 14 ROM_CBOOT
- 16 Kernel
- 18 IOMAN
- 19 File Managers
- 20 IO Drivers

Chapter 3: Hawk Release Notes **21**

- 22 Hawk 2.1 Release Notes
- 22 XML Project Format
- 22 General Improvements
- 29 Hawk Profiler
- 29 On-Line Help System
- 29 System Requirements
- 30 Using the Help System

31	Hawk 2.0 Release Notes
31	Editor
33	Project Manager
33	Debugger
34	Existing Issues
35	Documentation Notes

Chapter 4: SoftStax/LAN Communications Pak Release Notes **37**

38	DPIO
38	Libraries
38	Protocol Modules
39	Descriptors
39	Utilities
40	Ethernet Drivers
40	NFS
41	SPF File Manager
41	General Improvements
44	Enhancements

Chapter 5: SNMP Release Notes **47**

48	Known Issues
----	--------------

Chapter 6: MAUI Release Notes **49**

50	MAUI 3.0 Release Notes
50	MAUI for DAVID
50	Graphic Driver
51	Resolved Issues
52	MAUI 2.4.1 Release Notes
52	General Improvements
54	MAUI 2.4 Release Notes

Chapter 7: OS-9 Compiler Release Notes **55**

- 56 Compiler Version 2.4 Release Notes
- 56 Resolved Issues
- 60 Added Features
- 62 Compiler Version 2.3 Release Notes
- 62 Resolved Issues

Chapter 8: OS-9 Utilities Release Notes **75**

- 76 Updates to Utilities
- 79 General Notes
- 80 os9make Notes
- 81 "for" Loops
- 82 Macro Substitution
- 84 New Dependency Macros
- 85 New Macro Assignment Modes
- 87 Directory Building Rules
- 88 New Dynamic Macro
- 88 Other Changes

Appendix A: Example Port for M68360 QUADS Boards **89**

- 90 Enhanced OS-9 for 68K Version 1.1 Notes
- 90 Overview
- 90 Supported Features
- 90 MC68360 Reference Port
- 91 Rebuilding the QUADS ROM Image
- 94 Creating OS-9 for 68K Bootstrap Code for QUADS
- 95 OS-9 for 68K System Bootstrap
- 95 Creating QE (BOOTP Boot) Code for QUADS
- 96 Booting Procedures Available—<Input>
- 97 LINUX Example BOOTP Setup

98	Customizing ROM Bootstrap
98	QE Boot (BOOTP) Image
98	Starting SoftStax

Product Discrepancy Report

101

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

As part of Microware's policy of continued product development, this Enhanced OS-9 for 68k V1.1 represents a maintenance and update release, to incorporate all the improvements which have been introduced into the component parts.

These release notes cover the changes made to the Enhanced OS-9 package in the time since the Enhanced OS-9 for 68k V1.0 was made available. If upgrading from an earlier version, these release notes should be read in conjunction with the relevant older release notes.

This chapter contains the following sections:

- **Supported Reference Boards**
- **Enhancements**
- **New Optional Extras**
- **Features Removed**



Note

Files for Enhanced OS-9 for 68K Version 1.1 are updates to Version 1.0. Although these files are intended to be installed on top of your existing version of OS-9, it is recommended that you complete a backup of your system before installation.

Supported Reference Boards

This release of Enhanced OS-9 for 68K supports the following reference boards:

68030

- MVME147

68040

- MVME162
- MVME167

68060

- MVME172
- MVME177

68328

- MC328ADS

68360 (OEM Package Only)

- MC8360QUADS

Enhancements

- Support for the Motorola MVME147 board is reintroduced in this release.
- Known timer and alarm issues in the kernel have been fixed.
- The C library support for CPU32 family and 68020 family processors has been improved. This package now provides both (separate) CPU32 and 68020 specific libraries, whereas previous releases had only one set of libraries, to be used for both processor families. This separation, and recompilation, enables the libraries to be better optimized for the processor family, resulting in smaller, more efficient code. This change does not affect the 68000 libraries.



Note

The compiler executive automatically chooses the correct libraries, but CPU32 users requiring custom builds (i.e. not using the compiler executive) should ensure that they use the correct libraries (from the CPU32 directory). The libraries in the 68020 directory will no longer operate correctly on a CPU32 processor. Please see [Chapter 7: OS-9 Compiler Release Notes](#) for additional information.

- Hawk v2.1 is included in this package, offering a significant upgrade. More than 70 enhancements are documented in [Chapter 3: Hawk Release Notes](#). Especially significant is the faster debugging capabilities, making Hawk more usable over SLIP.



Note

As a result of extensive debugger enhancements, the PC resident Hawk components are incompatible with V1.2 (or earlier) OS-9 for 68k target daemons, so the daemons will need to be upgraded on any system that Hawk v2.1 is required to communicate with. Please see [Chapter 3: Hawk Release Notes](#) for more information.

- Hawk now simplifies the process of debugging a forked child process, by starting a new incarnation of the debugger interface when the child is forked.
- The Hawk target system profiler has been completely rewritten, to offer more useful data and features. Additional features include a memory usage and a CPU usage chart.
- Hawk's on-line help system has been updated. Please see the relevant sub-section of [Chapter 3: Hawk Release Notes](#), for details.
- Compiler improvements continue to be made. This release includes the latest Ultra C technology available.
- SoftStax networking has received some new functionality, improved mbuf handling and improved Ethernet performance. Please see [Chapter 4: SoftStax/LAN Communications Pak Release Notes](#) for more information
- NFS has had the 2 Gbyte file limit removed so that it can support file systems up to the RBF maximum of 4 Gbyte.
- Various utilities have been enhanced or improved. `os9make` has had some significant enhancements.

New Optional Extras

- X.25 support is available as an optional extra

Features Removed

- None

Chapter 2: OS-9 Operating System Version 3.2 Release Notes

This chapter describes changes made to the OS-9 operating system between the previous version of Enhanced OS-9 for 68K, 1.0, and the current version, 1.1. It includes the following sections:

- **ROM_CBOOT**
- **Kernel**
- **IOMAN**
- **File Managers**
- **IO Drivers**

ROM_CBOOT

- Resolved issue number 2215: When RomBug catches an user state exception, it incorrectly displays the type of exception.

Modules: ROM_CBoot versions of rombug

The cboot Linked version of RomBug incorrectly displayed the type of exception that had occurred if the `oe` command was used. If the exception was a user state exception, the `oe` command would incorrectly display "System state exception." The problem was corrected and incorporated in Enhanced OS-9 for 68K V1.1 ROM_CBoot ROM images.

- Resolved issue number 3036: Rombug displays incorrect fault address for 68030 stack frames.

Modules: ROM_CBoot Rombug code

The information displayed as the fault address in previous versions was only correct if the fault address was a data reference. The value displayed as the fault address is now also correct if the fault occurred from an instruction fetch.

- Resolved issue number 3300: RomBug does not assert CTS correctly.

Module: SC177

Edition: 24

Sources: /mwoS/OS9/SRC/ROM/SERIAL/iocd2401.a

The console serial driver used in MVME167 and MVME177 CPU boards was modified to assert CTS to allow cables with hardware handshake lines to work with the console driver used with RomBug. For simplicity, hardware handshaking is not implemented, but the CTS line is now set to a state that allows data transmission if a cable with more than transmit, receive, and ground is used to connect the CPU serial ports to a terminal device.

- Resolved issue number 6165: cboot ROM timer makefiles use obsolete header files.

Module: cboot Based Rom Images

Sources: /mwos/OS9/SRC/ROM/CBOOT/TIMERS/bp_tk147.c
 /mwos/OS9/SRC/ROM/CBOOT/TIMERS/bp_tk162.c
 /mwos/OS9/SRC/ROM/CBOOT/TIMERS/bp_tk167.c
 /mwos/OS9/SRC/ROM/CBOOT/TIMERS/bp_tk172.c
 /mwos/OS9/SRC/ROM/CBOOT/TIMERS/bp_tk177.c

The sources were modified to use standard shipping SPF versions of the included defs files rather than the older ISP Version, which has been discontinued from the standard product line. There are no differences in the binaries attributed to these changes.

- Resolved issue number 7792: Missing files in MWOS/OS9/SRC/ROM/CBOOT directory.

Sources: /MWOS/OS9/SRC/ROM/CBOOT/SYSBOOT/diskboot.c
 /MWOS/OS9/SRC/ROM/CBOOT/SYSBOOT/initdata.c
 /MWOS/OS9/SRC/ROM/CBOOT/SYSBOOT/binboot.c
 /MWOS/OS9/SRC/ROM/CBOOT/SYSBOOT/misc.c
 /MWOS/OS9/SRC/ROM/CBOOT/SYSBOOT/romboot.c
 /MWOS/OS9/SRC/ROM/CBOOT/SYSBOOT/sysboot.c
 /MWOS/OS9/SRC/ROM/CBOOT/SYSBOOT/sysboot_glue.c
 /MWOS/OS9/SRC/ROM/CBOOT/SYSBOOT/tapeboot.c

The files listed above were not included in the Enhanced OS-9 for 68K Version 1.0 distribution package although the documentation specified them as part of previous versions of the package. The OEM Version 1.1 distribution now includes the appropriate cboot ROM sources.

- Resolved issue number 8034: VMEchip2 Pre-Scaler wrong for 30 Mhz Hardware (MVME172).

Modules: MVME172 ROM_CBoot ROM images

Sources: MWOS/OS9/68060/PORTS/MVME172/ROM_CBOOT/
 sysinit.a

The initialization code for the board incorrectly set the pre-scaler value for the VMEChip2 to 30mhz. Board registers are now checked and appropriately set value for 25, 30, or 33 Mhz.

Kernel

- Resolved issue: Fixed a race condition in F\$STime system call.

Modules: Kernel (all 68k)
Edition: 360

If a ticker interrupt occurs before the call to the ticker, `init` routine returns. The kernel had not yet set `D_Tick` from `D_TickSec`, which is set by the ticker `init` routine. This change closes a window first seen on systems with free running tickers, for example ones that generate ticker interrupts from power line frequencies.

- Resolved issue: Changed memory allocation to close a possible condition where a request for RAM could return addresses for ROM if the system would likely have returned a no RAM available error.

Modules: Kernel (all 68K)
Edition: 361

- Resolved issue: Closed a condition where the system alarm handler could get into a loop without exiting for up to a second at 05:44:47 AM.

Modules: Kernel (all 68k)
Edition: 362

This would only occur when alarms were set for greater than one day in the future. In this case, an alarm was scheduled for midnight to reset the number of days left on the alarm. This case could cause looping. To avoid the word multiply that caused the looping, processors without the 32-bit multiply instructions (68000/68010/68070/68328 etc.) also set an alarm for Noon to avoid the 5AM looping condition. Processors with 32 bits multiplies (68020,68040, 68060) do not set the additional alarm timeout at noon.

- Resolved issue: The possibility existed that while an alarm delete was in progress, the alarm could be activated immediately after a system tick but before returning to the process deleting the “alarm execution thread.”

Modules: Kernel (all 68k)
Edition: 363

If the alarm was one-time-only, or chose to delete itself, the remainder of the original alarm delete in progress could fail. In addition, if a new alarm had been created and given the same ID as the one being deleted, it was possible the original alarm delete could delete the new alarm. These potential problems have been eliminated by better coordination between the system call code and the alarm execution thread.

IOMAN

- Resolved issue number 8157: Dump of IOMan in 68K 3.1.

Modules: Ioman(_dev) ioman(_atom)
Edition: Edition 37 (Same edition, smaller size)
Sources: NO Changes

The IOMan modules in the Enhanced OS-9 for 68K V1.0 Package were released with the in-house conditionals set. This generated additional checking code used for testing purposes and an embedded message in the module stating it was an "INHOUSE Version - NOT FOR DISTRIBUTION." The modules were recompiled with the conditionals set for the standard distribution version. The module Edition was not changed. The Version 1.1 modules can be identified by their sizes, which are smaller than the Version 1.0 modules.

File Managers

- Resolved issue number: 7990: pcf #82 is calling RomBug if disk is write protected.

Editions 82 and 83 (a previous release) were created with debugging flags set during compilation. The current edition 83 module included in the Enhanced OS-9 for 68K V1.1 release was rebuilt correctly and does not include the debugging test code.

- Resolved issue numbers 8852, 2003, 6681, 6824: Pipeman problem (ed. 56 vs. ed. 62) Unexpected `E_EOF` error from pipe operation.

Modules: Pipeman
Edition: 63

Improvements were made internally in pipeman to better handle cases where pipes are being opened and closed rapidly and signaling of data ready is required. Improved tracking of the number of readers and writers was also added. Transfer locking was added to protect a critical piece of code used by both reading and writing processes to prevent a possible case of data corruption. Repaired a case where a signal could potentially be sent to a non-queued process upon close of paths to a pipe. Added additional check to handle cases where both buffered and direct data transfers are occurring on the same pipe.

IO Drivers

- Resolved issue number 3120: Iniz of nonexistent device locks `rbvccs` driver.

A routine was added to the `SCSIxxx` drivers to probe for the hardware at the address passed in from the device descriptor. The driver now returns an `EOS_Hardware` error at attach time if the hardware is not found. This causes the logging data module to also be removed properly when attempts are made to attach a non-existent SCSI device.

- Resolved issue number 6965: OS-9 for 68K SCSI Incompatibility.
The SCSI device drivers supplied in the OS-9 for 68K Version 1.0 release accidentally broke compatibility with previous versions. This has been corrected so that device drivers from older, third-party systems, will again operate without change.
- Resolved issue number 8727: FFS Software - CF8685.

Sources:

`/mwoS/OS9/SRC/IO/RBF/DRVR/FLASHDVR`

The example driver provided with the True Flash File system did not have a working example makefile. The package now contains a makefile to make the example driver. The driver may still need to be adapted for the particular hardware or additional drivers written for other chips, but the example is now more complete for the AMDflash driver.

Chapter 3: Hawk Release Notes

This chapter describes changes made to the Hawk Integrated Development Environment between the previous version of Enhanced OS-9 for 68K, 1.0, and the current version, 1.1. It includes the following sections:

- [Hawk 2.1 Release Notes](#)
- [Hawk 2.0 Release Notes](#)

Hawk 2.1 Release Notes

The following sections represent new information regarding the release of Hawk 2.1.

XML Project Format

The Hawk project file (*.mpj) format has been changed from the ini file format to the xml file format. The conversion of existing projects will be seamless. An editor is provided with Hawk 2.1 for editing the project files manually.



WARNING

Opening an old Hawk project in the new version of Hawk (version 2.1) will make that project incompatible with previous versions of Hawk because Hawk will convert the project to xml format.

General Improvements

- Hawk now looks in the \MWOS\dos\bin directory (the MWOS location is read from the system's environment variable settings) for hawkdbg.dll and hawkprj.dll instead of looking at the libpreload registry keys as in previous versions.
- A button for the Profiler has been added to a toolbar on the project panel.
- Issue 2521: Support for the “Go with Boot Staging” command, gb, has been added to the Windows debugger. The RomBug command gb has been added to Hawk.

- Issue 4176: Previously, aborting a system state connection did not work. Aborting a system state connection using the `STOP` option when the debugger is not active was not supported. Currently, Hawk can handle the abort of a system state connection.
- Issue 4178: Previously, it was not possible to view properties of the module and **OTHER** from the symbol browser. Hawk would crash if the user selected to view the properties of the module or **OTHER**.

This crash no longer happens, and if the properties of the module are selected, Hawk completes an ident of that module on the target system and displays the information. The **ident** feature has also been added to the target drop-down menu.

- Issue 4190: A **Clean all** option has been added to Hawk. It is accessed through the project's pop-up menu. Currently, the incremental link option is not available with the compiler; it will be added to Hawk at a later date.
- Issue 4276: Previously, there were problems debugging modules of different files with the same source file name. This problem has been corrected; Hawk has been changed to use the full path names when searching for files.
- Issue 4297: A **Load all** option has been added to Hawk. It is accessed through the project's pop-up menu. When selected, all of the modules in a project are loaded to the target specified in the module's properties.
- Issue 4948: Previously, Hawk did not support building of C++ libraries containing templates. This problem has been resolved in Hawk by doing a prelink for both I-Code and O-Code C++ libraries.
- Issue 5137: Previously, Hawk did not support moving a project. This has been resolved; a dialog has been added that is called when Hawk detects a project has moved. The dialog gives the user two options to resolve relative sources. The options are listed below:

Assume the sources remain in their original location.

Assume the sources were moved relative to the new project location.

- Issue 6248: Previously, Hawk would cause the PC to hang when multiple `PreLoadLib` dlls were specified. Currently, Hawk can handle this task provided there is more than one set of registry entries for `PreLoadLib`.
- Resolved issue numbers 6924, 7007: The Help file `HAWKWT_10.HLP` appears to be missing from the products. The Help system no longer requires this file.
- Resolved issue number 6985: Hawk 1.2 for 68k is installing old and wrong `cpu.1` for 68020 Product.

An unusable version of `cpu.1` was placed in the `MWOS/OS9/68020/LIB` directory on Enhanced OS-9 for 68K V1.0 distributions. The file has been removed from the distribution.



Note

If the file `cpu.1` exists in your `MWOS/OS9/68020/LIB` directory, it should be deleted.

- Resolved issue number 7009: Hawk libraries not updated.
- Resolved issue numbers 7242, 7276: Enhanced OS-9 for 68K Evaluation Installation will not complete. The `dbgextns` module was incorrectly left out of the distribution. The module must be initialized via the `init` module's extension list or via the `p2init` utility prior to using Hawk.
- Issue 7397: Previously, Hawk's New Project/Unit dialog had a bad default button. The default button selected in the final stage of creating a project was the **Next** button, which was not enabled in that particular dialog. The default button has been changed to be the **Finish** button.

- Issue 7414: Previously, Hawk batch file units did not run properly on Windows 95. Hawk now creates the command line using the `COMSPEC` environment variable. When the full path of `command.com` on 95 and 98 is used, the batch file executes correctly and captures the I/O.
- Issue 7421: Previously, Hawk's project builder passed unsupported option `-bepg` to `be68k`. The 68k back end has been modified for this release so that it now accepts the `-pg` option.
- Issue 7514: Previously, it was impossible to build single source modules with Hawk. Now there is an option to exclude a unit from the link. In the **Properties** window for any given unit, you should see a check box with the caption "Exclude from Link." If this box is checked, the unit is still treated as a dependency of the component, but it will not appear on the command line when that component is linked (or a `libgen` is performed).

In addition, a change was made to add an option that associates a unit as the "Unit Root Psect" of a component. This is an easy way to add a unit to a component and then specify that the added unit is the Root Psect. In the **Properties** window for a component, there is a combo box called "Unit Root Psect". It contains a list of the units in that component. Select one of these to be your Root Psect. Note that this option takes precedence over the Root Psect option on the **Link** tab of the **Properties** window.

- Issue 7639: Hawk turns off all optimization by default for components. When a new component is created, the properties are synchronized with that of its parent. The parent may either be another component (if the new component is a child), or the project (if the new component resides at the root). After the properties are synchronized with the parent, the properties in `HAWKDEFS.INI` are applied based on component type. `HAWKDEFS.INI` resides in the Windows directory and can be freely modified by the user.
- Issues 7677 and 7679: When changing the stack size, Hawk previously used increments that were too large. For this release, Hawk has been changed to use more reasonable units.

- Issues 7753 and 7047: Previously, Hawk had overwritten settings for *.c, *.cpp, *.h extensions without warning in older versions. Instead of overwriting current settings, the current version now asks you if you would like to overwrite their current settings.
- Issue 7822: Previously there was no help available in Debug->Memdisplay.

In the current version of Hawk, if you click **Help** from the memory window, it displays context sensitive help.

- Issue 7823: Previously, a Hawk help file system failure occurred.
In the current version of Hawk, when Help --> Index--->Contents---Debugging--->Attach to process is selected, error 129 no longer appears.
- Issue 7859: Previously, it was not documented that the target system must have the modules pipe and pipeman in memory to use the Hawk Debugger. This has been added to the current documentation.
- Issue 7904: Previously, the Hawk help files were shipped "As Is" from Premia Corporation and documented the Codewriter 6.0 editor only. The non-Hawk-related functionality has been removed from the documentation in this release.
- Issue 7907: Previously, the **Add existing project** dialog in the file view window called the incorrect function. Currently, the menu calls the correct function.
- Issue 8071: Previously, the Hawk Assembler GUI displayed only assembly listed in the **show address** mode. Currently, the default for this window displays mixed source in **show function name** mode.
- Resolved issue number 8109: The following `dbgextns` modules were missing from the Enhanced OS-9 for 68K Version 1.0.

Modules:

`/MWOS/OS9/68000/CMDS/BOOT0BJS/dbgextns`

`/MWOS/OS9/68020/CMDS/BOOT0BJS/dbgextns`

The `dbgextns` modules were not included on the Enhanced OS-9 for 68k Version 1.0 Distribution. These files are now included in this release.

- Issue 8118: Previously, if Hawk exited when a project was not open it had tried to destroy unopened windows, which resulted in multiple `memory block freed twice` errors. The new version of Hawk checks to make sure the window exists before it tries to destroy it, which solves the problem.
- Issue 8122: Previously, component names that began with an underscore character were not displayed in the project window. This has been fixed. The current release of Hawk allows a component name to be up to 48 characters. The component name may contain any characters. The output name of the component is validated to allow alpha-numeric, '.', '_', or '\$' characters. Any invalid characters are replaced with an underscore.
- Issue 8125: Previously, there were several problems with Hawk's help system. The help system has been updated to be current and all of Hawk's menus and indexes access the correct files.
- Issue 8147: Previously, if you were debugging code and pressed **step** on the last line of source code error (106:006) would be displayed. `Mwsrddbgb.dll` has been modified so that it exits correctly, without reporting an error, when a step is done while on the last line of source code in a program.
- Issue 8188: Previously, Hawk only allowed the user to input numbers when editing registers, which meant the value of a floating point number could be changed only to an integer. The new version of Hawk allows users to enter numbers and decimal points in the floating point registers window.
- Issue 8191: Previously, Hawk required the user to hit **Enter** twice after changing the value of a variable in the locals or watch window. For this release, these windows have been modified so that the user only needs to hit **Enter** once.
- Issue 8287: On the previous **Connect** dialog, if you chose to attach to a module, a **Browse** button was made available to select the appropriate module. This button was removed due to the fact that the module to be entered in the box was intended to be one currently loaded on the target. Entering a module name with a full path was never intended and consequently did not work.

- Issue 8413: Previously, when you selected Hawk -> Help -> Configure Index File, files were listed that were not shipped. The index (`mwhawk.idx`) has been updated in this release to include the proper files.
- Issue 8548: Previously, upon running Hawk for the first time, you would see an error dialog saying, "System or Project Initialization Failed." This occurred because the `mwhawk.ini` file that was shipped had "ClearCase" listed in the VCS entries. This has been removed from `mwhawk.ini` in the current release.
- Issue 8566: `mwhawk.tdx` is the index file used by the Hawk Assist feature. It previously referenced files that did not ship with the product. The index file has been updated.
- Issue 8670: Previously, if Hawk failed when you attempted to load a module, no error message was displayed. It would appear as if the module was loaded although it was not. In this release, if Hawk fails to load the module or fails for another reason, an error message is displayed.
- Issue 8844: Previously, the **Group** and **User** edit boxes only allowed for three digits of input. Currently, five digits can be input; this allows for numbers up to 65,535.
- Issue 8849: Previously, Hawk crashed if you pressed **Build/Rebuild** without having a project space open. This problem has been fixed for the current release.
- Issue 8991: Previously there was an error in the Hawk **Component** menu. When a project space was opened, and you right-clicked a unit, the **Component** menu would pop up. If **Remove from project** was selected using this menu, the whole component would be removed. In the current release, this has been corrected.
- Issue 9011: Previously, SoftStax and MAUI functions were not available with the Hawk Assist feature. These are now available.

Hawk Profiler

- A Memory Usage and CPU Usage chart have been added.
- Completely re-written and easier to use. Some features that were added include selection of target, saving profile data to a text file, and displaying function data for multiple modules at the same time.
- The profile data is now stored in a table format. This fixed column alignment resolves problems with the previous profiler.
- The Hawk Profiler was rewritten as a Java application. To use the Profiler, you must have a Java Runtime Environment (JRE) installed on your system. The Hawk Profiler was tested against the Sun JRE version 1.2.2, which is provided on this product CD.

On-Line Help System

The on-line help system was improved to reflect an upgraded version of the code editor and new compiler and debugger functionality. The following sections describe the on-line help system.

System Requirements

The OS-9 on-line help system includes both of the following types of help files. The requirements for each are also listed.

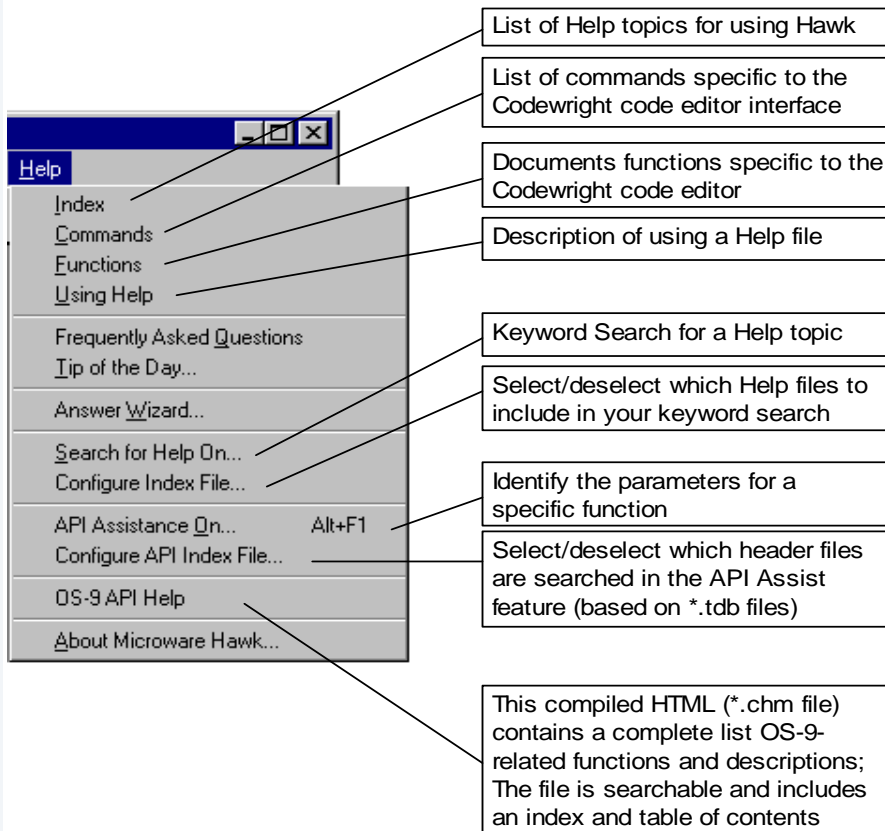
- Standard WinHelp 4 (.hlp) files. To view, these require Windows 95 or later and the `RoboEx32.dll` file (this file is shipped and installed with Enhanced OS-9 products).
- Microsoft Compiled HTML Help (.chm) files. To view, these require Windows 98 or Windows 2000. Windows 95 and Windows NT users must have Internet Explorer 4.0 (Service Pack 2). This is available for free from the Microsoft web site.

NOTE: All information provided in .chm format is also provided in PDF format.

Using the Help System

The OS-9 Help system is accessible from the Hawk integrated development environment user interface. **Figure 3-1** shows the Hawk Help menu and the Help options it contains.

Figure 3-1 Hawk Help Menu



In addition to the Help menu, most dialog boxes in Hawk will include a Help button. These are usually located in the lower right-hand corner of the dialog box. Selecting the Help button automatically selects the Help topic associated with the particular dialog box.

The F1 key no longer accesses the OS-9 API help topics because they have been converted to .chm format.

Hawk 2.0 Release Notes

Editor

CodeSense	CodeSense provides word completion and function parameter help. Graphical symbols distinguish parts of code, and “goto buttons” take you to their definition.
Project Spaces	Project spaces enable you to view multiple projects at one time without physically changing which project is the “current” project.
CodeFolio	Click and drag your most commonly used code stored in a folder or file view. This enables a quick placement, macro expansion, and prompted input as you drop.
Multiple Clipboards	Multiple clipboards and scrap buffers give you added room for storing code segments for use later in your editing sessions.
Command Line Editing	Enhanced command line editing enables greater flexibility in inserting text on the command line.
Clip View	View the contents of your clipboards or scrap buffers.
Updated Language Dialog	Associate View Setups or Lexers to your file type for added editing power.
ChromaCoding Lexer	Augment or change the support for an existing language. Lexers enable you to define comments, keywords, numbers and strings for your language without having to write a DLL.

Window Manager	The Window Manager dialog enables you to selectively tile or set display attributes for individual windows or documents.
View Setups	View Setups both consolidate and expand the most common customizations into a single dialog. Control color effect, font usage, outline effects and column wrapping.
API Macros	Use the interactive commands from the Hawk API to achieve a quick C-like macro language.
Enhanced Word Wrap	Hawk provides you with true soft word wrapping; word wrap without reformatting your code.
Auto-hide Toolbars	Give yourself more editing space by using auto-hide toolbars.
Customize Menu	All of Hawk's customization features are located in the <code>Tools -> Customize</code> menu.
Enhanced Auto-Save	Gain additional file specification options and the ability to limit the size of files that are saved.
Popup Menu Editor	This may be accessed by selecting the Edit this menu option on most Hawk popup menus.
Hex Mode Editing	Expanded hex editing features include: hex character can be copied, cut and pasted into documents. You may edit on either the hex or ASCII side in hex view, and the cursor position is highlighted on the other side to help you track your location.

Project Manager

The following project manager issues have been resolved:

- Issues 6072, 5395: The Extra Stack property has been fixed to default to 2K. This had been defaulting to 512K.
- Issues 4982, 7418: The order of components is retained when the project file is saved. The components may be reordered through both a mouse and keyboard (<ALT>Arrow) interface.
- Issue 7417: Units may be reordered to provide greater control the build process. A new dialog has been created (Unit Maintenance) to enable the user to enter new units, delete current units, and reorder all units.
- Issues 7278, 6825: The Exclude from Build option is fixed.
- Issue 7082: The Search control on the Standard toolbar now works correctly. If you enter text to search and hit Enter, the search will take place.
- Issue 7229: The Edition # property is retained when the project is saved.
- Issue 2943: The Serial and Telnet windows are now standalone toolbars rather than tabs on the Output toolbar. They may be displayed and docked individually.

Debugger

- The speed of the debugger has been increased.
- A new standalone debugger (`hawkdbgr.exe`) has been created to handle the debugging of forked child processes. This debugger may be run as a separate process.
- Issue 6941: All debugger data windows, except Process I/O, are disabled when the debugger is communicating with the target.

The Process I/O window is no longer a tab in the Output toolbar. It is a separate window.

- Issue 7241: The Hawk debugger now looks at the environment variable `TARGETUID` to determine what user ID should be used for the process on the target system. The format of a user ID is `ggg.uuu` where `ggg` is the group number and `uuu` is the user number.
- Issue 6921: A variable with the volatile type will no longer confuse the debugger symbol browser.
- Issue 7279: The Target Load and Unlink dialogs now remember up to 8 previous targets used.
- Issue 4184: The debugger used to have a problem when the user selected to Reset the target. The debugger will now gracefully handle the reset of the target.
- Issue 2946: When the Watch or Locals windows are resized, the second column will automatically resize to completely fill the window.
- Issue 6816: The debugger now allows the setting of complex breakpoints. A complex breakpoint usually contains a condition in addition to a location.

Existing Issues

Due to the debugger enhancements, the debugger daemons need to be updated on the target OS-9 machine. The Hawk debugger will sense any incompatibilities and inform you before continuing.

If the user presses the Stop button while the Hawk Debugger is waiting for input, the connection will be lost, Hawk is likely to be temporarily frozen, and the target daemons will be left dangling. This is a limitation of OS-9.

Documentation Notes

The Hawk documentation, in PDF format, includes the following manuals:

- ***Getting Started With Hawk***
- ***Using Hawk***
- ***Using Hawk Macros***

These documents focus on using the Hawk integrated development environment to develop Hawk projects in OS-9—including using the editor, debugger, and compiler.

The Hawk documentation, in standard WinHelp format, includes the On-line Help System, which is accessed from the Hawk interface.

The On-line Help System focuses exclusively on using the Hawk editor, which is a derivative of the Codewright 6.0 editor from Premia Corporation. The On-line Help System may contain references to Codewright functionality that has been removed from Hawk. It may also not include functionality added to Hawk for this release.

Chapter 4: SoftStax/LAN Communications Pak Release Notes

This chapter describes changes made to the SoftStax and LAN Communications Pak networking software between the previous version of Enhanced OS-9 for 68K, 1.0, and the current version, 1.1. It includes the following sections:

- **DPIO**
- **Libraries**
- **Protocol Modules**
- **Descriptors**
- **Utilities**
- **Ethernet Drivers**
- **NFS**
- **SPF File Manager**
- **General Improvements**
- **Enhancements**

DPIO

- The DPIO device list entry now matches the non 68k versions. In particular the driver global pointer (`v_dr_globals`) and file manager global pointer (`v_fm_globals`) fields are now initialized properly. (Issue Numbers 8524 and 8525).

Libraries

- All SoftStax libraries now ship 68020 and CPU32 versions in addition to 68000.
- The `recv` and `recvfrom` socket library functions now support the flags parameter. The Supported flags are `MSG_PEEK` and `MSG_WAITALL`. These options also require the new protocol modules to work correctly.
- The socket library functions `getsockname()` and `getpeername()` now return the correct information after doing a non-blocking accept. This fix also requires new protocol modules.
- The `setsockopt()` function in the socket library now accepts either an `int` or a `u_char` as a parameter for the `IP_MULTICAST_TTL` and `IP_MULTICAST_LOOP` socket options. This fix also requires new protocol modules (Issue Number 7738).
- A problem in the mbuf library that could corrupt mbufs with the `SPF_NOFREE` bit set was fixed. (Issue Number 8355).

Protocol Modules

- Mbuf usage has been improved and socket programs will now consume less mbuf space per active connection.
- A problem where in some situations `sptcp` would retransmit bad data, potentially causing undetected data corruption, has been fixed. (Issue Number 7335).

- An ICMP error in response to a TCP SYN packet could potentially cause a continuous stream of SYN packets being retransmitted. This problem has been fixed. (Issue Number 4888).
- A problem with `sptcp` incorrectly sending a FIN too early has been fixed.
- Much of `spenet` was rewritten to improve performance. This also fixed known problems with mbuf leaks and arp flooding. (Issue Numbers 6718 and 2734).

Descriptors

- A `make clean` now correctly causes SPF device descriptors to be rebuilt. It is no longer required to do a purge in order to force a rebuild.

Utilities

- There is now a Windows hosted version of `rpcgen` in addition to the resident version. (Issue Numbers 8509 and 8655).
- The Windows hosted version of `rpcdump` now correctly prints the contents of an `rpcdb` module. (Issue Number 8840).
- The `dhcp` client now works with the Windows NT DHCP server. Also a potential bus error when adding DNS servers returned from the server was fixed.
- Ping no longer exits immediately when receiving ICMP packets other than the expected response. (Issue Number 9138).
- An `ftp mget` now works correctly if the local directory is on a PCF device.
- The `ftp` server now supports `-f` and `-e` options to specify alternate directory listing commands. Also, directory listings are always sent in ASCII mode even if the current mode is binary.

- The ftp server no longer fails with an `EADDRINUSE` error when receiving simultaneous directory requests. (Issue Number 6785).
- The bootp server (`bootpd`) now works correctly with multiple Ethernet interfaces. (Issue Number 8645).
- The tftp server (`tftpd`) now supports the block and file size options specified in RFCs 2347, 2348, and 2349.

Ethernet Drivers

- The `sp82596` driver has been fixed so it no longer hangs or crashes the system under heavy packet load. (Issue Number 8637).
- The MTU for the QUADS port is now correctly set to 1500 instead of 1496. (Issue Number 8532).
- The `spquicc` driver has had several changes. An mbuf leak when transmitting packets was fixed, multicast support was added, and several other minor bug fixes. In addition it has been changed to make it easier to port to other boards. (Issue Numbers 7725 and 2733).

NFS

- Resolved issue number 7746: Incorrect (and missing) module paths in 68K bootlist files.

Sources:

```

/MWOS/OS9/68040/PORTS/MVME162/BOOTLISTS/*
/MWOS/OS9/68040/PORTS/MVME167/BOOTLISTS/*
/MWOS/OS9/68040/PORTS/MVME172/BOOTLISTS/*
/MWOS/OS9/68040/PORTS/MVME177/BOOTLISTS/*

```

The `nfs`, `nfs_devices`, and `nfsnul` file locations were referenced in the `/MWOS/OS9/68020/CMDS/BOOTOBJS` directory. The references have been changed to `/MWOS/OS9/68020/CMDS/BOOTOBJS/SPF` directory.

- The 2GB file system limit for NFS has been removed. File systems up to the RBF maximum of 4GB are now supported. (Issue Numbers 7894, 8602, 8855, and 8902).

- `mountd` now prints an error instead of crashing with a bus error if it is unable to resolve the name returned from `gethostname()` into an IP address. (Issue Number 8904).
- The NFS server no longer maintains exclusive access to a file for 30 seconds after a write. The file can now be accessed (read only) on the host immediately after a write operation by `nfsd`. (Issue Number 3062).

SPF File Manager

- The receive thread now inherits its priority from the starting process rather than being fixed to 128.
- Added `FMCALLUP_TIMER_START`, `FMCALLUP_TIMER_STOP`, and `FMCALLUP_TIMER_RESTART`. These macros take the same parameters as, and are intended to replace, the `timer_start`, `timer_stop`, and `timer_restart` functions in `sptimer.l`. For compatibility, the library is still available and now uses the file manager callups.
- Fixed a problem where the spf timers could become very inconsistent when using a combination of one shot and cyclic timers.

General Improvements

- Issue 2602: The FTP server (`ftpd/ftpd.c`) now supports a `-f` and `-e` option which can be used to allow a web browser to correctly view an OS-9 target.
- Issue 2731: A memory leak in `spenet`'s `dr_term` has been fixed.

- Issue 2734: `spenet` has been changed to prevent `arp` flooding. At most 1 `arp` request for a particular address will be sent per second. Also, the length and resolution of the timer that controls removing `arp` entries is configurable in the `enet` descriptor. The default values are 60 seconds for the timer interval, 20 intervals (i.e. 20 minutes) to remove completed entries and 3 intervals to remove incomplete entries.
- Issue 2769: `Ifconfig` no longer requires UDP. If it can not open a UDP socket it will try a TCP socket, and if that fails it will try a raw socket.
- Issue 2784: `sptcp` was incorrectly adding an additional `maxseg / 8` when opening the congestion window. It now uses the correct value of $(\text{maxseg}^2) / \text{cwnd}$.
- Issue 2789: `ipstart` now prints a warning instead of an error if no protocol modules (`sptcp`, `spudp`, `spraw`, or `sproute`) are successfully started. This is most often caused by a missing descriptor so it also reminds the user to check and make sure they have the correct drivers and descriptors loaded.
- Issue 2801: Routed will no longer crash if an interface is added via `ifconfig`.
- Issue 2891: Fixed a problem with `select` and TCP sockets. There was a small window of time where a socket being closed by the remote end would not break an application out of a `select` call. This hole has been fixed and now `select` should return in all situations.
- Issue 4842: The `ftp mget` command fails to work if the current directory is on a PCF device. This was caused because the temporary file created to hold the file list did not conform to the 8.3 naming standard. This has been changed to a shorter filename and now `mget` works correctly.
- Issue 5111: The check in `sptcp` to retract the congestion window was incorrect. The old check was: `tp->t_dupacks > tcprexmitthresh` but has now been changed to the correct: `tp->t_dupacks >= tcprexmitthresh`.

- Issue 5337: The `mkdir` command in the ftp server (`ftpd`) returns an `EOS_PARAM` error. This has been fixed and now the `mkdir` command works correctly.
- Issue 5386: The TCP/IP stack runs out of mbufs and stops working under high load. It is not an error that in a situation where you have a mismatch between the size of the mbuf pool, and the amount of traffic, that all TCP traffic will pause. However, this pause should only last until TCP connections begin timing out. In some situations connections timing out would not free their allocated mbufs and the pause turned into a deadlock situation. This is now fixed and any terminated connection will return its mbufs and the system will continue to run normally.
- Issue 5605: The header files `auth.h`, `auth_unix.h`, `os9.h`, `pmap_clnt.h`, `pmap_prot.h`, `pmap_rmt.h`, `rpc.h`, `rpc_msg.h`, `svc.h`, `svc_auth.h`, `types.h` and `xdr.h` in the `MWOS/SRC/DEFS/SPF/RPC` directory have been modified to work correctly when compiling with C++ code.
- Issue 6564: The java function `getLocalAddress()` and socket function `getsockname()` do not return the correct value. When doing a non blocking connect the value returned from a subsequent call to either of these functions returns an incorrect value. This has been fixed and they now return the correct value in all situations.
- Issue 7466: Parsing problem because of extra semi-colon in `sys/cdefs.h` when using C89 or ANSI modes. The stray semi-colon has been removed from the `cdefs.h` header definition of `__END_DECLS`.

- Resolved issue number 8023: `idbgen` for 68000 is generating bus error (000:102).

Modules: `idbgen`
Edition: `236`

Changed the data reference model to use the 68k, `ld` (long data) compiler option to allow for referencing relatively large `intedb` data structures during the creation of the module.

Modules: `idbgen`
Edition: `237`

Change creation attributes on `inetdb2` to add write permission so the module can be modified for `hostname` and other host specific fields when the module is loaded into RAM.

Enhancements

- `sptcp`: The `mbuf` to `mbuf` data copy in `sptcp` has been removed. Now the only time the data is copied is when it is copied from the users buffer into an `mbuf`. This results in a substantial improvement in TCP performance.
- `spenet`: Most of the data path was rewritten resulting in a significant speed increase.

If `spenet` does not have enough space to contain the Ethernet header, it allocates a new `mbuf` and copies the data into it. Previously it did not leave any free space for additional header required by the hardware drivers. Now it leaves the amount of free space requested by the hardware drivers `TXOFFSET` variable.

- `arp`: This uses `spenet`'s new `arptab` structure and will no longer work with previous versions of `spenet`.

- `ftpd/ftpd`: This now sets the `SO_REUSEADDR` socket option on the listening socket. This allows the server to be restarted even if some sockets are in the `TIME_WAIT` state.

In some error situations the server would send duplicate error messages and get into a confused state. These duplicates have been removed.

- `ping`: Now exits immediately instead of waiting when `sendto()` returns an error such as `EHOSTUNREACH`.



Chapter 5: SNMP Release Notes

This chapter describes changes made to the Microware implementation of SNMP between the previous version of Enhanced OS-9 for 68K, 1.0, and the current version, 1.1. It includes the following sections:

- **Known Issues**

Known Issues

- The `target.tpl` file in `mwos/SRC/SPF/SNMP/src` must be edited before running any compiles. One, and only one, of the supported `TARGET =` lines must be uncommented.
- All SNMP makefiles use `bmake` rather than `os9make`. To make all the subagent examples, use the appropriate makefile in the `mwos/SRC/SPF/SNMP/src/emanate/sub/examples` directory.

For example, to build them for SNMP version 1, use the following command line:

```
bmake -f Makev1.os9
```

The appropriate makefile in the following directory must be run before making any example subagents:

```
mwos/SRC/SPF/SNMP/src/mibs/common
```

Chapter 6: MAUI Release Notes

This chapter describes changes made to MAUI (the Microware Graphics and Audio package) between the previous version of Enhanced OS-9 for 68K, 1.0, and the current version, 1.1. It includes the following sections:

- [MAUI 3.0 Release Notes](#)
- [MAUI 2.4.1 Release Notes](#)
- [MAUI 2.4 Release Notes](#)

MAUI 3.0 Release Notes

MAUI for DAVID

Support for the **MAUI for DAVID** and **MAUI for DAVIDLite** profiles have been discontinued in an effort to simplify the product. Most DAVID customers used the standard full MAUI profile and none of the handicapped versions. Customers that wanted JAVA were required to use the full version.

Applications compiled against the 2.4 or later versions of **MAUI for DAVID** or **MAUI for DAVIDLite** should work without needing anything from recompile with the new shared libraries (`maui`, `mt_maii`).

Graphic Driver

The graphic driver logical unit static storage structure has changed to correct an occasional memory access violation in `gfx_update_display()`; this occurred when two or more processes had the graphics device open. This change requires that the driver and descriptor be recompiled. New descriptors are not compatible with old drivers nor are new drivers compatible with old descriptors. To take full advantage of this update, OEM customers should update their custom drivers as indicated below.

Graphic drivers that made the `if (gfxdev->shared == dvatch->lustat->topdev->shared)` test, or one similar (often in `uptdpy.c`), should change to the test below (or one similar):

```
if (gfxdev->shared == dvatch->lustat->topdev_shared)
```

Resolved Issues

- An address violation in `_gdv_set_topdev()` of `GX_COMM/gdv_dev.c` has been fixed by protecting the call with a semaphore in open and close. The system state open/close calls were interfering with the user state update display call.
- Issue 8931: The rasterization problems with `drw_arc()` and `drw_circle()` have been fixed.
- Issue 7736: `GX_VGA` to use `gfx_get_cm_name()` has been corrected.
- Issue 7955: Enhancements have been made to reduce the size of `BLT_CONTEXTS` that do not use expansion. This reduces run-time memory usage and speeds up manipulation of the `BLT_CONTEXT`.
Each `BLT_CONTEXT` is also almost 2K smaller and the expansion table is only to be calculated when it is needed to do an expansion blit. This provides added savings because other objects (`ANM_OBJECT`, `DRW_CONTEXT`, `TXT_CONTEXT`, etc.) allocate one or more `BLT_CONTEXTS` each.
- The callbacks, `maui_vfprintf()` and `maui_fflush()` have been created from a shared library. This allows the MAUI shared library to print to `stderr` without pulling in the `fprintf()` code. Functions such as `mem_list_segments()` and `mem_list_overflows()` now work as documented in the shared library.

If you do not want the shared library to print or do not want to incur the code size overhead of the print functions in the binary of the application, simply define the following functions in the application:

```
#include <stdio.h>
int    maui_vfprintf(FILE *fp, const char *fmt, va_list ap)
{
    return 0;
}
int    maui_fflush(FILE *fp)
{
    return 0;
}
```

MAUI 2.4.1 Release Notes

General Improvements

- Issues 7446, 7492: The graphic driver common code has been modified to correct warnings associated with `GDV_SET_VPINTEN`, `GDV_CREATE_CURSOR`, `GDV_DESTROY_CURSOR`, `GDV_SET_CURSOR`, and `GDV_SET_CURSOR_POS`.
- Issue 2124: The `GDV_HWBLT_*` function prototypes have been added to the graphic driver common code to eliminate warnings.
- Issue 2124: `mem_list_overflow()` now returns error code `EOS_MAUI_DAMAGE` if a memory overflow is detected.
- Issue 2133: New `demo_init()` and `demo_term()` functions have been added to the MAUI demo library.

All `*_init()` call error handling and `*_error()` calls have been rewritten to reduce code size.

`win_init()`'s failure has been corrected to call `gfx_term()` and `drw_term()` on error.

Enhanced 68k version of CDB API to work with SSM.

- Issue 7338: Previously, `cdb_get_copy()` caused exception if the MAUI shared library was not installed. This issue has been corrected.
- Issue 4911: The `drwarc()` bug that caused bus trap when drawing an arc off the bottom of the screen has been corrected.

Also, the default behavior for `EOS_MAUI_NOTFOUND` errors from `cdb_get_ddr()` has been corrected to no longer print a text error message. In addition, the error type for `EOS_MAUI_NOTFOUND` has been changed from `MAUI_ERR_NON_FATAL` to `MAUI_ERR_WARNING` and the CDB APIs default debug print level has been changed so that `MAUI_ERR_WARNINGS` are not printed.

- Issue 2119: Read and write entry points to `mfm` have been added, as have both the sound and graphic driver common and sample code.

Names of functions reported by some of the CDB API errors have been corrected.

- Issue 7308: Modified `auplay.c` and `aurecord.c` sound demos to try to find the audio device name from the CDB.
- Issue 8970a: Blitting problems associated with `blt_expand_block()` and `blt_expd_next_block()` when using expand mode `BLT_MIX_RWT` on little endian processors have been corrected.
- Issue 8970b: `IO_BLT` graphic drivers common code (`gdv_blt.c`) that did not properly return `EOS_MAUI_NOHWSUPPORT` for expand using RWT mode have been corrected. If OEM customers wish to add support for expand using RWT in their driver, they can remove the test in `gdv_blt.c`'s `set_dispatch()` and add the required support in `gdv_expd.c`. Expand using RWT is not in the standard `IO_BLT` driver code in the interest of size and speed.

MAUI 2.4 Release Notes

- 1- to 2-bit expansion in driver `IO_BLT`.
- True Color swap bit in `GFX_CM` allows better support of 16- and 32-bit color hardware.
- `gfx*_dev_attribute()` calls allow setting of brightness, contrast, hue, saturation, sharpness, gama, white balance, etc. on some hardware.
- Improvements in oval and arc drawing.
- Corrected copy block problem that could result in data access outside of a drawmap.

Chapter 7: OS-9 Compiler Release Notes

This chapter describes changes made to the Microware Compiler Ultra C/C++ between the previous version of Enhanced OS-9 for 68K, 1.0, and the current version, 1.1. It includes the following sections:

- [Compiler Version 2.4 Release Notes](#)
- [Compiler Version 2.3 Release Notes](#)

Compiler Version 2.4 Release Notes

Resolved Issues

The following list explains issues concerning Compiler 2.3 and discusses how these issues are resolved:

- Resolved issue number 2384: The `_os_id()` function call was using 1024 bytes of stack space. This amount was a significant portion of the default stack space of an application. This function has been modified to only use 740 bytes of data, and this data is not located on the stack, rather in the global data area.
- Resolved issue number 3659: The linker incorrectly assigns data fields in a module when remote and non-remote initialized data are used. The linker correctly reports when remote initialized data exists (as well as non-remote initialized data) when it is not allowed for certain module types. In addition, when creating a raw binary, the compiler reports the total size of initialized data that exists (including remote).

Edition 151 or later of `l68` fixes this problem.

- Resolved issue number 3990: Previously, the assembler could only handle up to 65,535 references in a file. If more were present, an incorrect assembly file was generated, which resulted in the linker/libgen generating an error. Currently, the assembler has been updated to handle more than 4 billion references in a single file.
- Resolved issue number 4276: Previously, there have been problems debugging modules that contain the same source file name, but different paths. Previously, the Hawk debugger brought up the first instance of the file it encountered. If you attempted to attach two modules that have source code files with the same name (such as `entry.c` for SPF drivers), you would find that the second one uses the sources from the first. This has been updated so that the compiler and Hawk each use full path names.

Edition 34 of `cpfe` (or later) and Hawk version 2.1 (or later) are required to fix this problem.

- Resolved issue number 4853: The compiler verbose output and error messages from various phases were incorrectly intermingled in the middle of the output. This issue has been corrected so that the compiler now places errors at the end of verbose output.

In Hawk, Edition 66 or later of `xcc` fixes this problem.

- Resolved issue number 5015: Previously, it was not possible to redirect output when pre-processing from a DOS command prompt. `Cpfe` edition 31 or greater fixes this problem.
- Resolved issue number 5536: The `%n` conversion specifier does not work if you have an exact format match (if you hit `EOF` on your input).

`Clib.l` dated August 24, 2000, or later fixes this problem.

- Resolved issue number 5934: There was a problem in the `writenv()` function of `unix.l`, where it would abort the gather operation when an element of length 0 was present in the array. `Unix.l` dated August 26, 2000, or later fixes this problem.
- Resolved issue number 6094: `#ident` causes the front end of the compiler to terminate with an internal error. If a source file contains a `#ident` directive and is compiled in ANSI extended mode, the following internal error is generated:

```
At end of source: internal error:
C2Istmt.c:SetPragmas: no pragmas!
```

In ANSI mode, the above error is correct.

The following program further demonstrates this problem:

```
#ident "This is who I am" int f(int p) { return p; }
```

Edition 32 or later of `cpfe` correctly flags this as an illegal preprocessing directive.

- Resolved issue number 6292: The compiler should generate statically linked `fpu` emulation code. However, supporting this type of emulation code can be expensive. Generic floating point software emulation is supported with the compiler version 2.3. To specify the software emulation of floating point instead of the standard hardware emulation, use the `,fp` target suboptions (such as, `-tp=ppc,fp`).

Edition 67 or later of XCC, Edition 2 or later of `ficcode`, and its appropriate libraries, are required to fix this issue.

- Resolved issue number 6439: The `getwd()` function does not work with PCF file systems.

`Unix.1` dated August 26, 2000, or later fixes this problem.

- Resolved issue number 6606: `iopt` leaves code with extraneous pointers. Use the code below as an example:

```
static void func(int *a0, int *a1, int x)
{
  if (x)
  {
    *a0 = 1;
    *a1 = 2;
  }
  else {
    *a0 = *a1 = 0;
  }
}

int callit(int a, int x)
{
  int a0, a1;
  func(&a0, &a1, x);
  return a + a0 + a1;
}
```

`iopt` leaves all pointers where it should have deleted pointer symbols and referred to the symbols directly.

Edition 95 or later of `iopt` fixes this problem.

- Resolved issue number 6977: `iopt` code motion hoists function calls above global references. If the function call hoisted modifies the global in question, the generated code does not function properly.
Edition 96 or later of `iopt` fixes this problem. Edition 97 improves this fix.
- Resolved issue number 6989: RFI: add support for more time zones [as done for OS-9 (non-68K)].
Additional time zones were added to `oslib.l` to match those known by OS-9 (non-68K) platforms. Note: OS-9 for 68K keeps system time as the current local time in most implementations.
- Resolved issue number 7026: `iopt` can terminate with a segmentation fault if enough optimizations are suppressed. If CSE is not suppressed and other optimizations are suppressed, `iopt` can result in a segmentation fault. An example of such a command line is shown below:

```
iopt -o -he -cd -cl -ct <file>.i
```


Edition 98 or later of `iopt` fixes this problem.
- Resolved issue number 7372: The `_os_open()` function prototype now specifies the pathname as a `const char *` instead of just `char *`.
- Resolved issue number 7515: Previously, the assembler did not allowing a '-' character in the filename.
Assembler version 205 fixes this issue.
- Resolved issue number 7649: `cpfe` consumes all system memory in debug mode.
If a source file has a large structure with more than 10,060 fields and has debugging enabled, the compiler front end consumes all memory on the system and eventually produces the following error:

```
I-code buffer exhausted.
```


Edition 33 or later of `cpfe` fixes this issue.

- Resolved issue number 7800: There were problems replacing the default `sysgo` with a CSL linked application. If a large number of command line options were passed to the application, it crashed.
CSL edition 22 or above fixes this problem.
- Resolved issue number 8514: The linker does not complain when linking 68K libraries and libraries from other processors.
Edition 156 or later of `l68` fixes this problem.
Assembler version 204 or later fixes this problem.

Added Features

- A command line switch has been added to all compiler executables; the command line switch prints out the versions of these compiler executables. To see version information regarding all compiler phases, type the following command.

```
%xcc -tp=68k -ver
```

- OS-9 for 68K is now shipping CPU32 versions of the compiler libraries. These libraries are located in the following directory:

```
$(MWOS)/OS9/CPU32/LIB
```

These are automatically selected when CPU32 is specified as the processor type.



Note

The CSL module is now also shipped for the CPU32 family in `mwos\OS9\CPU32\CMDs\`. The CSL module in `mwos\OS9\68020\CMDs\` is now a true 68020 version and should NOT be used on CPU32 family processors. Boot lists for CPU32 family ports should use the CPU32 CSL module and not the 68020 version of CSL.

- Resolved issue number 3659: 168 not reporting errors about initialized data in driver.

Modules: I68 (all versions)
Edition: 151

168 will now correctly report when remote initialized data exists as well as non-remote initialized data, when it is not allowed for certain module types. When creating a raw binary, 168 will report the total size of initialized data that exists, including remote data.

- Resolved issue number 8111: Please remove the `conv_lib.l` text file in `68020/lib` and `cpu32/lib`.

Modules: /MWOS/OS9/68020/LIB/conv_lib.l
 /MWOS/OS9/68020/LIB/lock.l
 /MWOS/OS9/CPU32/LIB/conv_lib.l

These files contained a warning message that the user should use the corresponding `lib` file in the `68000/LIB` directory. They are no longer part of the distribution.

Compiler Version 2.3 Release Notes

Resolved Issues

The following issues have been resolved:

- Issue 2204: Program compiled with `-c` crashes with bus error.

The following program ends with a bus error when compiled with the `-c` option. Edition 24 of `cpfe` fixes the problem.

```
#include <stdio.h>
typedef struct {
    const char *str;
} stru;
main()
{
    stru data[] = {
        {"abc"},
        {"123"},
    };
    printf("c:%c\n", data[0].str[0]);
    exit(0);
}
```

- Issue 2373: `intercept()` and `signal()` do not work together.

This problem has been corrected. The order of calling `intercept()` and `signal()` is irrelevant. Signals will be dispatched to the proper routine regardless of the order. The latest version of `clib` and `sys_clib` libraries fixes the problem.

- Issue 2397: Tables of function pointers in code do not work.

Edition 24 of `cpfe` fixes the problem.

- Issues 2413 and 2545: Long file names do not work using `readdir` in `sys_clib.l`.

`readdir()` has been updated to support PCF Long FileNames. The latest `sys_clib` library fixes this problem.

- **Issue 2428:** `getwd()` does not work correctly for PCF.
`getwd()` now uses `opendir()/readdir()/closedir()`. It was also modified to be non-recursive, to not `malloc/free` memory on each loop, and to not make an assumption as to the maximum path length to be returned. It will also abort if caught in a recursive loop, rather than crashing like the predecessor. The latest `unix.1` library fixes this problem.
- **Issue 2547:** `getopt` should use `writeln` instead of `write`.
When `getopt` prints an error message, it uses `write`. However, SCF does not process the `\n` as an end of line character, which causes problems in later messages. This should be changed to `writeln`. The use of `write()` to print error messages in `getopt()` was replaced with `_os_writeln()`, which comes from `os_lib.1`. Error messages should print correctly on SCF devices now. The latest `unix` library fixes this problem.
- **Issue 2577:** `strerror` should not cause problems with `errno`.
Both `strerror()` and `perror()` avoid writing over the value in `errno` at function call time. `errno` is saved and restored so that it is constant across the call to `strerror()` and `perror()`. The latest `clib` library fixes this problem.

- Issues 2649 and 5443: Additional version information requested for the compiler.

The compiler will now show its versioning information if a `-tp=<target>` option is used. However, it does not list files provided. Refer to the below example:

```
xcc -tp=ppc
Microware Ultra C/C++ Compiler. Version 2.3
Copyright 1999 Microware
PC version
xcc      Edition  67, Nov 10 1999
cpfe     Edition  32, Nov 10 1999
ilink    Edition  11, Nov 10 1999
iopt     Edition  97, Nov 10 1999
ficode   Edition   2, Nov 10 1999
beppc    Edition  59, Nov 10 1999
optppc   Edition  65, Nov 10 1999
appc     Edition  56, Nov 10 1999
prelink  Edition   9, Nov 10 1999
linker   Edition 151, Nov 10 1999
```

This is supported with version 2.3 or later of Ultra C/C++.

- Issue 2743: `cpfe` internal error.

The following code results in a `cpfe` internal error when compiled:

```
/* foo.c */
main()
{
    char    *a;
    int     *b;
    int     n;
    n = a - b;
    /* "n = a - (char *)b;" is OK */
}
```

This has been fixed in `cpfe`, Edition 30 or later.

- Issue 2931: C-library function `BussError (102)` when there is insufficient memory.

This error can be duplicated on any OS-9 machine by starting a shell with most of the memory and then by running any utility which prints out data. Some amount of trial is necessary depending on the system. This error case can happen on any C FILE type access on this error path.

The latest release of the `clib` library fixes this problem.

- Issue 2954: Use directive defaults to non default directories.

All assemblers (`os9` and `os9000`) will default to looking in `/dd/defs` for files included using the `use` directive in the absence of any other suggestions (for example the `-u=<directory>` command-line switch or the `CDEFS` environment variable). This is a poor solution on cross-compilation systems (for example Windows).

The default `use` directive search `dir`, in absence of a given search path (via the `DEF envvar` or the `-u` switches) is as follows: For resident systems the default `dir` remains `/dd/defs`. For cross-hosted systems the default `dir` is `/mwos/<OS>/SRC/DEFS` where `<OS>` is `OS-9` for 68K and `OS-9000` for all others.

This was fixed in the following editions of the assemblers:

r68: 203

a386: 60

others: 53

- Issue 2955: `cpfe` converts calls to functions as calls via a function pointer.

In the absence of a function prototype, `cpfe` is transforming calls to a function as if they were called via a function pointer.

This was fixed in `cpfe`, as early as Edition 28.

- Issue 2968: Unnamed bitfield, initialization causes internal error.

The following program causes `cpfe` to terminate with an internal error:

```
struct bob {
    unsigned short a;
    unsigned int :16; /* note: un-named */
    unsigned long b[32];
} a_bob = { 1, {1, 2} };
```

This was corrected in Edition 23 of `cpfe`.

- Issue 2997: Assembly optimization: `loc` tracking misses some gratuitous moves. The location tracking code does not process the following very well (code is ppc):

```
nop
li r2,15
mr r1,r2
mr r2,r1
nop
```

The second `mr` could be eliminated but it is not (actually, copy propagation will eliminate it, so this should be done with copy propagation disabled). However, if the `li` is replaced with a `MEMORY` load, the operation completes:

```
nop
lwz r2,0(r3)
mr r1,r2
mr r2,r1
nop
```

The second `mr` is eliminated.

This has been fixed in Edition 63 or later of the assembly optimizers (`opt68k`, `opt386`, `optppc`, `optarm`, `optsh`, `optmips`, and `optsparc`).

- Issue 3039: GPF in DOS version of linker.

The linker may terminate after emitting "value too large for field"-type errors.

This was fixed in Edition 149 or later of the linker.

- Issue 3055: The `is*()` functions in `ctype.h` should process large ints.

The `is*()` routines in `ctype.h` do not operate well with values outside the range of an unsigned `char`.

The `is*()` functions declared in `<ctype.h>` have been modified to check for invalid input (values that are not representable as an unsigned `char` and not equal to the value of the macro `EOF`) and return 0 for such input.

The macros defined in `<ctype.h>` do not have such safeguards. As a result they are faster. Since the macros are the default, to use the functions you must explicitly use the `undef` preprocessor directive or ensure that the macro expansion does not occur (by placing the function name within parentheses).

The latest release of the `clib` library contains this update.

- Issue 5563: `scanf` function handles `%n` incorrectly in some cases.

The `%n` conversion specifier doesn't work if you have an exact format match (in other words, if you hit `EOF` on your input).

The latest release of the `clib` library fixes this problem.

- Issue 5586: C libraries have internal `iopt` warnings.

When `-cw` (get verbose warnings) is used, certain functions in the C libraries may give warnings, which can not be controlled by the programmer.

The latest release of the compiler libraries fixes this problem.

- Issue 5662: `clib.1` - LIBM version of math routines incorrectly generates exceptions.

Certain math functions might generate exceptions given certain error conditions. For example:

```
pow(0.0, -1)
```

The ANSI standard is as follows:

```
<math.h>
```

4.5.1 Treatment of Error Conditions

The behavior of each of these functions is defined for all representable values of its input arguments. Each function shall execute as if it were a single operation, without generating any externally visible exceptions. This allows exceptions to occur, but they must be handled by default.

The latest release of the compiler libraries fixes this problem.

- Issue 5719: `iopt` can incorrectly move the wrong code during common code motion.

It's possible for `iopt` to incorrectly move statements around when doing common code motion optimization. For example:

```
irq_disable();
do {
    check_something();
    irq_disable();
} while (cproc == 0);
can get changed to:
irq_disable();
{
    check_something();
} while (cproc == 0);
```

It meant to complete the task below:

```
goto inloop;
toploop:
    check_something();
inloop:
    irq_disable();
    if (cproc == 0)
        goto toploop;
```

Disabling common code motion with `iopt`'s `-h` option works around the problem. Other factors may also hide it as well.

Edition 89 or later of `iopt` fixes this problem.

- Issue 5790: `iopt` terminates on some functions.

Certain (largely dead code) functions can cause `iopt` to GPF on Windows. For example, this source file causes a crash:

```
void func(void)
{
    extern int glob;
    switch (glob) {
        case 1:
        case 2:
        case 3:
            break;
    }
    return;
}
```

Edition 90 or higher of `iopt` fixes this problem.

- **Issue 5869:** `cpfe` - creates bad implicit prototypes for functions defined in `_asm()`

Given the following code (`ick.c`):

```
#ifndef NOASM
_asm("function: ");
#endif
int func(int x, int y)
{
    char buf[64];
    if (function(x, buf))
        return x + y;
    return 0;
}
```

Compiled with the following (the target is irrelevant):

```
xcc -tp=68k -efe ick.c -cw -fd=ick1.i
xcc -tp=68k -efe ick.c -cw -dNOASM -fd=ick2.i
```

The front-end is looking through the `_asm()` (in the first example) for labels that appear to be function definitions (labels that start with an `isalpha()` char and contain only `isalnum()` chars and end with a `:` -- this misses those with `'_'` and will also catch variable definitions) and generating what amounts to a prototype for these "functions".

Fixed in edition 28 of `cpfe.exe`. No longer attempts to track "functions" defined in `_asm()` statements.

- Issue 5892: `xcc` - allow overriding of target suboptions with default suboptions.

The compiler executive allows you to override previous options with new ones (when reading left to right). The following demonstrates this ability:

```
> xcc -bh x.a -eas -tp=68k -tp=ppc
```

Include file paths:

```
\mwos\src\DEFS
\mwos\OS9000\src\DEFS
\mwos\OS9000\PPC\DEFS
appc -a=_UCC -a=_MAJOR_REV=2 -a=_MINOR_REV=2 -a=_SPACE_FACTOR=1
-a=_TIME_FACTOR=1 -a=_OS9000 -a=_MPFPOWERPC -a=_FPFPOWERPC -a=_BIG_END
-o=x.r -q -u=\mwos\src\DEFS -u=\mwos\OS9000\src\DEFS
-u=\mwos\OS9000\PPC\DEFS x.a
```

Note that the target is PowerPC, not 68k.

However, this ability is not provided for the target suboptions.

For example:

```
> xcc -bh -xio -ebe x.i -tp=armv4,vld -tp=ppc
```

Include file paths:

```
\mwos\src\DEFS
\mwos\OS9000\src\DEFS
\mwos\OS9000\PPC\DEFS
beppc -o=x.o -px -m=0k x.i
```

Note that the target changed (`beppc` is used instead of `bearm`), however the `bearm` option `-px` (which corresponds to ARMV4's `,vld`) is still present. `beppc` cannot function with this option.

You can determine the default target suboptions by asking for them explicitly. Note, however, that if you use `-tp=ppc,vld` the executive merely ignores the suboption (because it doesn't exist for PowerPC), but keeps the `-px` on the command line.

Note that this handicap does not seem to infect the assembler phase:

```
> xcc -bh -eas x.a -tp=68k,sc -tp=sparc
```

Include file paths:

```
\mwos\src\DEFS
\mwos\OS9000\src\DEFS
```

```

\mwos\OS9000\SPARC\DEFS
asparc -a=_UCC -a=_MAJOR_REV=2 -a=_MINOR_REV=2 -a=_SPACE_FACTOR=1
-a=_TIME_FACTOR=1 -a=_OS9000 -a=_MPFSPARC -a=_FPFSPARC -a=_BIG_END
-o=x.r -q -u=\mwos\SRC\DEFS -u=\mwos\OS9000\SRC\DEFS
-u=\mwos\OS9000\SPARC\DEFS x.a

```

Note that the `-j` switch is not passed to `asparc` (it would be if the target were still 68k). The compiler now correctly resets target suboptions to their defaults.

Edition 64 or later of `xcc` fixes this problem.

- Issue 6053: `iopt` can corrupt semantics of structure copy.

`iopt` does not seem to acknowledge that when a composite is copied and one version of the composite is modified that it should keep both copies as the user wrote.

The code that demonstrates this is as follows:

```

int glob;
typedef struct {
    int lX, lY;
} int_2d_t;
typedef struct {
    int_2d_t stOrg, stDim;
} im_win_t;
int
lCopyStruct(im_win_t stWinProcess)
{
    im_win_t stWinCpy;
    glob += stWinProcess.stDim.lY;
    stWinCpy = stWinProcess;
    stWinCpy.stDim.lY /= 2;
    glob += stWinCpy.stDim.lY;
    glob += stWinProcess.stDim.lY;
    return 0;
}

```

`iopt` uses `stWinProcess` for every composite reference shown.

Edition 91 or later of `iopt` fixes this problem.

- Issue 6238: `xcc`'s new preprocessing options do not function correctly.

The implementation was changed to correctly parse the file name and allow the mode to be in lower case. In addition, the `-P` option in C89 mode now can be followed by a mode option to get similar functionality. The C89 modes are upper case.

Edition 65 or later of `xcc` fixes this.

- Issue 6242: Preprocessing to files does not function properly, it always goes to `stdout`.

Edition 31 or later of `cpfe` fixes this problem.

Chapter 8: OS-9 Utilities Release Notes

This chapter describes changes made to the Microware utilities between the previous version of Enhanced OS-9 for 68K, 1.0, and the current version, 1.1. It includes the following sections:

- **Updates to Utilities**
- **General Notes**
- **os9make Notes**

Updates to Utilities

<code>copy</code>	<p>Issue 8869: If a destination file already exists, the copy utility will prompt the user to overwrite the file. The utility will no longer terminate badly if the user answers "n" for no.</p> <p>There is a new <code>-c</code> option to convert end of line characters between different line terminations.</p>
<code>del</code>	<p>Enhancement 7765: This enhancement now allows command line options to be specified in <code>-z</code> files. This is mainly to support <code>os9make</code> on DOS, where the command line length is limited and options may need to be passed by file.</p>
<code>dir</code>	<p>The command line argument parsing, year and file size print, and sector number sign has been fixed.</p>
<code>events</code>	<p><code>-h</code> and <code>-k</code> options have been added.</p>
<code>format</code>	<p>New prompts for cluster size and number of sectors, new support for partitioning large disks, and corrected problem error checking auto-sized disks have been added.</p>
<code>kermit</code>	<p>Several of the problems that could cause memory corruption have been fixed.</p>
<code>login</code>	<p>You can now use this as an environment to set shell.</p> <p>This now works without a current directory (for disk-less systems).</p>
<code>mfree</code>	<p>Issue 8614: The link count of the <code>init</code> module now correctly decrements when <code>mfree</code> has finished using it.</p>

<code>mshell</code>	<p>Issue 8264: A path completion problem (a potential buffer overrun) sometimes caused <code>mshell</code> to report that it has run out of RAM. It would then terminate. This has been corrected.</p> <p>Issue 7694: DOS end-of-line (CRLF) characters are currently supported, along with the standard OS-9 end-of-line (CR) character.</p> <p>Several of the problems that could cause memory corruption have been fixed and a <code>-<sig></code> option to kill command has been added.</p>
<code>os9make</code>	<p>Issue 7993: Given an incomplete macro expansion, e.g. <code>\$(MAC</code>, recent versions of <code>os9make</code> might crash. This has been corrected.</p>
<code>paths</code>	<p>This can now display more than 1024 processes.</p>
<code>procs</code>	<p>A new command line argument to specify individual process IDs has been added.</p>
<code>rpcdump</code>	<p>The <code>ed#26</code> bug, where it could not identify 80386 <code>rpcdb</code> modules, has been fixed.</p>
<code>save</code>	<p>This now supports the use of <code>-f</code> and <code>-z</code> together.</p>
<code>shell</code>	<p>Several of the problems that could cause memory corruption have been fixed and a <code>-<sig></code> option to kill command has been added.</p>
<code>sysgo</code>	<p>This utility now uses <code>mshell</code> for processing the startup file and as the shell forked on the console port for user interaction.</p>

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, which 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.

`tsmon`

Fixes have been made to this utility to allow it to use environment to set shell.

This utility no longer bus traps when using the `-z` (from `stdin`) option.

`umacs`

Several of the problems that could cause memory corruption have been eliminated.

`unlink`

The `-f` option can no longer enter an infinite loop by attempting to unlink members of a module group.

`xmode`

This utility no longer bus traps when using the `-z` (from `stdin`) option or when attempting to change a write protected descriptor.

General Notes

- Issue 2686: All OS-9 DOS utilities (`ident`, `fixmod`, `os9del`, etc.) now support wildcards on the command line.
- All of the OS-9 utilities have been recompiled with the current UltraC compiler. This may have caused the CRC and/or size of these utilities to change even if they had no source code updates. The edition number is a better indication of updates.
- The default stack size of the utilities has been decreased, and some utility stacks have 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.

os9make Notes

- Edition #138 of `os9make` corrects a bug when the `-b` option is used. Previous versions of `os9make` did not report missing dependencies. As a result, `os9make` was failing to detect and report when a dependency did not exist (no file or directory) nor a rule to build it existed. Following is an example:

```
cd ..\SRC\DPIO\SPF\DRVR\SPPROTO; os9make -nn "MOPTS=-nn"
"AOPTS=" "COPTS=" "EOPTS=" "LOPTS=" HOST=mdos
"DOCMD="TARGET=k68k SUBTRGT=_build _build
os9make: can't open "nulltrg".
```

This was caused by the makefile referencing `nulltrg` but not defining it:

```
$(TRGTS): nulltrg
        -$(MAKESUB) -f=$@.mak
```

This can be solved by either adding a `nulltrg` rule:

```
$(TRGTS): nulltrg
        -$(MAKESUB) -f=$@.mak
nulltrg : .
        $(COMMENT)
```

or by changing the `nulltrg` to `..`:

```
$(TRGTS): .
        -$(MAKESUB) -f=$@.mak
```

Another common error in older versions of `os9make` was the use of implied built-in rules even when using the `-b` and `-bo` options. Following is an example:

```
...
SDIR = ../SRC
ODIR = ../CMDS
...
myfile : mysource.c
        $(LINK) -fd=$@ $(SDIR)/mysource.c
```


The source was in `$(SDIR)` but the dependency considered this to be in the current directory. This rule caused the make to always run. To correct this add `$(SDIR)` to the dependency line as follows:

```
...
SDIR = ../SRC
ODIR = ../CMDS
...
myfile : $(SDIR)/mysource.c
        $(LINK) -fd=$@ $(SDIR)/mysource.c
```

In addition, the following enhancements were made to `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.
- As of edition #132, `os9make` supports use of loops.

"for" Loops

New "for" loop functionality has been added. Refer to the below example:

```
_purge: nulltrg
for NAME in $(TRGTS)
    -$(CODO) $(NAME)
    -$(DEL) $(NAME)
    -$(CODO) $(NAME).stb
    -$(DEL) $(NAME).stb
endfor
```

As is apparent from the above example, if you have a large number of targets, this can substantially reduce the size of your makefile. The above "for" can also be nested. Refer to the below example:

```
_purge: nulltrg
for NAME in $(TRGTS)
for NAME2 in $(NAME) $(NAME).stb
    -$(CODO) $(NAME2)
    -$(DEL) $(NAME2)
endfor
endfor
```

Macro Substitution

Be careful to not indent the contents of if and for statements. Standard make indentation rules have not changed. New macro substitution syntax can greatly simplify makefiles.

Current OS-9 makefiles require that you specify both a RELS line and an IRELS line (if both types of object are being generated). Now it is possible to specify a single SRCS line and let generic make rules handle the redundant lines.

Refer to the below example to start:

```
CRELS = $(RDIR)/dodate.r $(RDIR)/doname.r $(RDIR)/domake.r \
        $(RDIR)/domac.r $(RDIR)/data.r $(RDIR)/misc.r \
        $(RDIR)/rule.r
CICODE = $(IDIR)/dodate.i $(IDIR)/doname.i $(IDIR)/domake.i \
        $(IDIR)/domac.i $(IDIR)/data.i $(IDIR)/misc.i \
        $(IDIR)/rule.i
```

The above example could be rewritten as follows:

```
CSRCS = dodate.c doname.c domake.c domac.c data.c \
        misc.c rule.c
CRELS = $(CSRCS:%.c=$(RDIR)/%.r)
CICODE = $(CSRCS:%.c=$(IDIR)/%.i)
```

Combining this with the “for” loop above, almost all redundant lines of a makefile can be eliminated. This results in a smaller, easier to read, and easier to maintain makefile.

In addition to the `% = %` modifier, there are several other new rewriting functions. There are others for performing useful operations on the source macro. For instance, to select just the last component of a file path, use the `T` function as follows:

```
$(CRELS:T)
```

The result (from the example above) would be as follows:

```
dodate.r doname.r domake.r domac.r data.r misc.r rule.r
```

Multiple modifiers may be specified and are processed in order.

Macros can be used anywhere within the modifier format strings, for even "macroized" substitution processing.

The list of macro modifiers includes the following:

E	Replace each word with its suffix		
	TEST/test.c	:E	c
H	Replace each word with everything but the last component		
	TEST/test.c	:H	TEST
L	Lowercase each word		
	ARMv4 Mc68000	:u	armv4 mc68000
M	Select words matching a pattern		
	t1.c t1.h t2.c	:M*.c	t1.c t2.c
N	Select words not matching a pattern		
	t1.c t1.h t2.c	:N*.c	t1.h
R	Remove suffix from all words		
	TEST/test.c t1 t2.c	:R	TEST/test t2
S	Substitute old string for new string (sed syntax)		
	testing the rest	:S/t/T/g	TeSting The resT
	testing the rest	:S/t/T/1	Testing The resT
	testing the rest	:S/t/T/	Testing The resT
	testing the rest	:S/est\$/EST/	testing the rEST
	testing the rest	:S/^est/EST/	testing the rest
	testing the rest	:S/^t/T/	Testing The rest
U	Uppercase each word		
	armv4 Mc6800	:u	ARMV4 MC68000
<p>% = % UNIX prefix/suffix syntax (percent is the wildcard pattern character which is left intact on the rewrite)</p>			
	t1.c t1.h t2.c	:%.c=% .r	t1.r t2.h t3.r
	DIR1/t1.c t2.c	:DIR1/%.c=% .r	t1.r t2.c
	t1.c t2.c	:%.c=RDIR/% .r	RDIR/t1.r RDIR/t2.r

Putting the macro substitution together with "for" loop processing results in a very efficient makefile.

In the generic case, you need only modify the CSRCS line and standard rules take care of the rest.

For example, based on the example above:

```
CSRCS = dodate.c doname.c domake.c domac.c data.c \
      misc.c rule.c
CRELS = $(CSRCS:%.c=$(RDIR)/%.r)
CICODE = $(CSRCS:%.c=$(IDIR)/%.i)

for SRC in $(CSRCS)
$(SRC:%.c=$(RDIR)/%.r): $(SRC)
    $(RCOMPILER) $(SRC)
$(SRC:%.c=$(IDIR)/%.i): $(SRC)
    $(ICOMPILER) $(SRC)
endfor
```

This example reduces a huge bulk of the makefile down to a few lines. The only line you may need to touch is the first.

New Dependency Macros

The operation in the above example is made easier and more efficient by specifying the same `$*.c` conversion in the dependency line as in the build commands which follow the line.

`$*` represents, in make syntax, to take the current object being built and strip off the directory path and suffix. Any characters following the asterisk are then appended to the new string.

Therefore, `$*.c` on `RDIR/test.r` will result in `test.c`.

`$@` represents the current object being built.

For example, we used to have the following:

```
$(RDIR)/dodate.r: dodate.c
    $(RCOMPILER) dodate.c
$(IDIR)/dodate.i: dodate.c
    $(ICOMPILER) dodate.c
$(RDIR)/doname.r: doname.c
    $(RCOMPILER) doname.c
$(IDIR)/doname.i: doname.c
    $(ICOMPILER) doname.c
```

for each individual source file (which could get quite lengthy). The above may also be written more efficiently as:

```
$(RDIR)/dodate.r $(IDIR)/dodate.i: dodate.c
$(RDIR)/doname.r $(IDIR)/doname.i: doname.c
$(RELS):
    $(R_COMPILE) $*.c
$(IRELS):
    $(I_COMPILE) $*.c
```

You would still require another line in the body of the makefile for each individual set of targets/source. One solution that might seem practical is shown below:

```
$(RELS): $(SRCS)
    $(R_COMPILE) $*.c
$(IRELS): $(SRCS)
    $(I_COMPILE) $*.c
```

This solution works but will also build every one of your targets any time a single source file is modified. This is because each individual object on the left hand side would depend upon all objects on the right side.

This is where the new syntax takes effect, as shown below:

```
$(RELS): $*.c
    $(R_COMPILE) $*.c
$(IRELS): $*.c
    $(I_COMPILE) $*.c
```

This results in dependencies that vary depending on the object whose rules are being defined. In fact, it surpasses the use of "for" in some cases. For instance, the "for" loop example from above becomes:

```
CSRCS = dodate.c doname.c domake.c domac.c data.c \
      misc.c rule.c
CRELS = $(CSRCS:%.c=$(RDIR)/%.r)
CICODE = $(CSRCS:%.c=$(IDIR)/%.i)
$(CRELS): $*.c
    $(R_COMPILE) $*.c
$(IRELS): $*.c
    $(I_COMPILE) $*.c
```

New Macro Assignment Modes

Following are two common OS-9 macro assignments:

```
MACRO = value1
MACRO = value2
```

With UNIX `make`, if `MACRO` were later expanded, it would provide a string of `value2`. With `os9make`, if `MACRO` were later expanded, it would provide a string of `value1`.

This is the fundamental difference (in terms of compatibility) between `os9make` and most other `make` programs. Each model is equally valid and useful.



Note

`os9make -l` does macro assignments in UNIX style.

`+=` will append a string to the current value of a macro in `os9make`. For example:

```
MACRO += value2
```

If `MACRO` were later expanded, it would provide `value1 value2`.

`?=` has been added, which forces the assignment of a macro even if it has already been defined. In `-l` mode, this would assign the macro only if it were not previously assigned.

Therefore,

```
MACRO = value1  
MACRO ?= value2
```

If `MACRO` were later expanded by `os9make`, it would provide a string of `value2`. If `os9make` were in `-l` mode, it would provide a string of `value1`.

Another need is to be able to set environment variables directly from a makefile. Often third party applications need environment variables set to function properly. The `@=` operator performs this. For example:

```
PATH @= $(MWOS)/DOS/BIN
```

Directory Building Rules

OS-9 makefiles attempt to create the `RELS` directory each time you perform a build. This is a function of the rule used to create the directory, which typically look as follows:

```
$(TRGT): _mkdir ...
_mkdir: nulltrg
        $(MAKDIR) $(RDIR)
        $(MAKDIR) $(IDIR)
nulltrg: .
        $(COMMENT)
```

In this case, `nulltrg` is doing nothing but adding a command to execute. Old versions of `os9make` required this step. Current versions of `os9make` do not require this step. Following is an equivalent example:

```
$(TRGT): _mkdir ...
_mkdir: .
        $(MAKDIR) $(RDIR)
        $(MAKDIR) $(IDIR)
```

This rule gets executed almost every time because when you modify the current directory (create a temp file for instance), the modify time on `."` is updated. Since the `_mkdir` rule depends on `."` it is considered out of date. This does not make sense when directories are either there or not there. Worse since the `$(TRGT)` above depends upon the `_mkdir`, it may always be relinked whenever the `_mkdir` build commands are executed—even when the directory already exists.

The new functionality allows that if a directory (which already exists) is the object and has no dependencies, the build rules are ignored. Therefore, you can now write the following:

```
$(TRGT): $(RDIR) $(IDIR) ...
        ...
$(RDIR) $(IDIR):
        $(MAKDIR) $@
```

You might then consider a makefile that always executes makes in a set of directories. In that case, use the dependency of `."` For instance:

```
TRGTS = FM DRVR
$(TRGTS): .
        $(CHD) $@; $(MAKE) -f=$(MAKENAME)
```

To conditionally call makes you can use "for" loops and "if exists" as follows:

```
TRGTS = FM DRVR
for NAME in $(TRGTS)
if exists($(NAME)/$(MAKENAME))
$(NAME): .
    $(CHD) $@; $(MAKE) -f=$(MAKENAME)
endif
endfor
```

New Dynamic Macro

In addition to the predefined macros \$@, \$*, and \$?, the following macro is now supported:

\$<	Enumerate all the files upon which an object depends. For example:
	foo : main.c bar.c
	\$(xcc) \$< -f=\$@

Other Changes

- `ifexists` and `ifnexists` are now supported for symmetry similar to `ifmake` and `ifnmak`.
- The `-n` option is now passed down through cascading makes.

Appendix A: Example Port for M68360 QUADS Boards

This appendix describes the example port for M68360 QUADS boards.

Enhanced OS-9 for 68K Version 1.1 Notes

This section describes changes to the QUADS board port new in Enhanced OS-9 for 68K version 1.1.

Overview

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

It is assumed that the QUADS port may be used as an example when working with custom, OEM-based CPU32-based hardware.

The QUADS port is considered basic because of the lack of hardware I/O beyond serial and Ethernet. OEMs who use this board as a basis for new design work are encouraged to also refer to other example ports for addition support of devices. The MVME177 port is one such port, which can be found in the following location:

```
\mwoS\OS9\68060\PORTS\MVME177
```

Supported Features

- Support for onboard Ethernet and Serial interface 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.

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.

Rebuilding the QUADS ROM Image

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

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

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-1 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-1 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/spfdrvrvr.mak	Creates spquicc
		SPF/SPQUICC/spf_desc.mak	Creates spqe0
ROM_CBOOT/makefile			
	ROM_CBOOT/rom.make		Creates CMDS/BOOTOBS/ROM/rommerROM image without rombug
		ROM_CBOOT/ROM_common.make	Creates rom_common.l (vectors, boot,nvdfbtb)
		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/BOOTOBS/ROM/rombug (BLS CBOOT core distribution file)
		ROM_CBOOT/ROM_initext.make	Creates CMDS/BOOTOBS/ROM/initext User sysinit extension binary
		ROM_CBOOT/ROM_bootfile.make	Creates CMDS/BOOTOBS/ROM/bootfile rom based bootfile
	ROM_CBOOT/rombug.make		Creates CMDS/BOOTOBS/ROM/rombugger ROM image with rombug
		ROM_CBOOT/ROM_common.make	Creates rom_common.l (vectors, boot,nvdfbtb)
		ROM_CBOOT/ROM_serial.make	Creates rom_serial.l console driver library (io68360)

Table A-1 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/BOOTOBS/ROMBUG/rombug (BLS CBOOT core distribution file)
		ROM_CBOOT/ROM_initext.make	Creates CMDS/BOOTOBS/ROMBUG/initext User sysinit extension binary
		ROM_CBOOT/ROM_bootfile.make	Creates CMDS/BOOTOBS/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.

Creating OS-9 for 68K Bootstrap Code for QUADS

- Step 1. Change to the `\mwos\OS9\CPU32\PORTS\QUADS\ROM_CBOOT` directory and type `os9make`. (Optional)
- Step 2. Next burn the following image into a 27C010 EPROM:
`\mwos\OS9\CPU32\PORTS\QUADS\CMD5\BOOTOBS\ROMBUG\rombugger`
If ROMBUG is not desired you may burn the following image instead:
`\mwos\OS9\CPU32\PORTS\QUADS\CMD5\BOOTOBS\NOBUG\rommer`
- Step 3. Connect a terminal to the board (from a Windows 95, 98 or NT host, you can use Hyperterminal).
Select `9600 Baud`, `8 Bits`, and `No Parity`.
- Step 4. Insert EPROM in socket U48.
- Step 5. Apply power to the board. At this stage the OS-9 bootstrap message is displayed. Type the `g` command to bring up the bootmenu options.

OS-9 for 68K System Bootstrap

Following is an OS-9 for 68K Bootstrap message:

```
<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 QE (BOOTP Boot) Code for QUADS

Complete the following steps to create OS-9 for 68K QE (BOOTP BOOT) code for the QUADS board.

-
- Step 1. Obtain a suitable bootp server. The Linux bootp server or a Windows based bootp server may be used if available. A Linux example appears below.
 - Step 2. Create QE BOOT IMAGE by entering the following commands:


```
> cd \mws\OS9\CPU32\PORTS\QUADS\BOOTS
> os9make (Optional)
```

The created OS-9 QUADS boot image is placed in the following directory:

```
\mws\OS9\CPU32\PORTS\QUADS\CMS\BOOTOBS\BOOTFILES\qe_netboot.bf
```

Step 3. Copy this file to the bootp server machine/directory.

Step 4. Use "QE" option from the QUADS OS-9 68k bootstrap menu to boot.

Booting Procedures Available—<Input>

Following is an example boot menu:

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

Select a boot method from the above menu: qe

```
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

Following is an example BOOTP set up procedure:

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

```
mkdir /tftpboot
```

Step 2. ftp the boot image to Linux server and place in `tftpboot` directory. Make sure the name of the image is `os9boot` for this example.

Step 3. Set the following permissions on bootfile.

```
chmod 777 os9boot
```

Step 4. Setup the following 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 with the following commands:

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

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

Re-Boot Linux system if change was required here.

Step 6. Start the bootp server with the following command:

```
bootpd
```

If debugging is desired, use the following command:

```
bootpd -d7
```

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

Customizing ROM Bootstrap

To customize the ROM Bootstrap code refer to the following directory:

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

You can customize the bootmenu by changing the `syscon.c` and/or `nvdftbl.a` files. Be sure to re-run `os9make` once the changes are made.

Refer to `\mwos\OS9\CPU32\PORTS\QUADS\systype.d` for switch settings and other defines.

QE Boot (BOOTP) Image

To modify the QE Boot (BOOTP) image, edit the following file:

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

Remove or add the modules required. See [Creating QE \(BOOTP Boot\) Code for QUADS](#) for information on re-making the boot image.

Starting SoftStax

Following is an example of starting SoftStax.

```
$ 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
```



Note

Another example of starting SoftStax using `ndbmod` is available in the following location: `mwoS\src\sysMODS\startspf.ndbmod`.

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 _____

