



[Home](#)



Microware OS-9[®] Release Notes

Version 1.3



RadiSys.
THE POWER OF WE

www.radisys.com

Revision A • February 2006

Copyright and publication information

This manual reflects version 1.3 of Microware OS-9 for 68k. Reproduction of this document, in part or whole, by any means, electrical, mechanical, magnetic, optical, chemical, manual, or otherwise is prohibited, without written permission from RadiSys Microware Communications Software Division, Inc.

Disclaimer

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

Reproduction notice

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

August 2005

Copyright ©2006 by RadiSys Microware Communications Software Division, Inc.
All rights reserved.

EPC and RadiSys are registered trademarks of RadiSys Corporation. ASM, Brahma, DAI, DAQ, MultiPro, SAIB, Spirit, and ValuePro are trademarks of RadiSys Corporation.

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

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

Contents

Chapter 1: Introduction

Chapter 2: Operating System

Enhancements.....	8
Resolved Problems.....	8

Chapter 3: Host Applications

Hawk Notes	14
Enhancements	14
Resolved Problems	17
Known Issues	25
TECH-CHECK Notes.....	25

Chapter 4: Components

OS-9 Compiler Notes	28
Enhancements	28
Resolved Problems	31
Networking Notes	39
Enhancements	39
Resolved Problems	41
OS-9 Utilities Notes.....	46
Enhancements	46
Resolved Problems	47
MAUI Notes.....	51
Enhancements	51
Resolved Problems	51
Add-Ons.....	52
Enhancements	52

1

Introduction

Microware OS-9 for 68k version 1.3 represents a maintenance and update release to incorporate all of the improvements that have been introduced into the component parts.

2

Operating System

This chapter provides an overview of the changes and improvements made to OS-9 for 68k for version 1.3.

For information about...	Go to this page...
Enhancements	8
Resolved Problems	8

Enhancements

The following list describes general enhancements made to the OS-9 operating system for this release. Where applicable, CustomerFirst (CF) and ClearQuest (RSYS) incidents are included.

- CF 13012: New low-level console service, `rom_fprintf`.
The console services in the low-level ROM structure now has a function pointed to by the `rom_fprintf` field. Previously, the value of `rom_fprintf` was NULL.
- CF 13348: ModMan File Manager components.
`modman` and `mm` descriptor have been added for all processors.
- CF 13465: Support `iovcons`.
The `iovcons` module is now shipped with all processors.

Resolved Problems

The following section describes CustomerFirst (CF) and ClearQuest (RSYS) incidents related to the OS-9 operating system and how they were resolved for the current release.

- CF 10489: `siuirq.c` and `cpicirq.c` routines.
Originally, only the `cpicirq.c` attempted to re-enable interrupts to allow nesting. This allowed all non-CPM interrupt sources to preempt and interrupt the CPM interrupt sources. Code in `cpicirq.c` was modified so that after the appropriate masking is done on the CPM module, the CPM interrupt to the SIU is re-enabled.
- CF 13748: SCSI driver NCR8xx is now detecting the 895A correctly.
Unlike all previous chips, the 895A has a 256-byte read and write FIFO. In addition, the PCI Device ID search has been updated for use with the 895 and 895A.
- CF 13749: NCR8xx low-level driver add-on for 895A.
Three modifications were made to the NCR8 low-level driver for 895A:
 1. The reset time has been changed from 20 μ sec to 350 msec.
 2. Support for 895A has been added.
 3. The chip now resets at termination.
- CF 13757: Set the IRQ Mask in the SIU and re-enable the higher IRQs.
The handling has been updated similarly to `siuirq.c`.
- CF 13781: The `rom` module service `goodmodule()` fails when the boot size equals the module size.
The ROM software was fixed so that if the last module in a ROM-based boot ends on the last byte of the ROM memory, it is still accepted as a valid module.

- CF 13803: The low-level 16450/16550 driver needed baud rates higher than 38,400

The low-level 16450/16550 serial driver now supports baud rates higher than 38,400 up to 115,200. This allows the higher baud rates to be selected in the Wizard low-level serial port configuration.

- CF14122: Ethernet boot finds modules from ROM/Boot image. Adding `nokrs` option to eb booter

Normally, the kernel will have access to all modules found by the low level memory scan in addition to modules found during the high level memory scan. The eb booter has been modified to support an argument, `nokrs`, to prevent the eb booter from passing the ROM module list to the kernel. `nokrs=0` is the default (report modules found to the kernel) and `nokrs=1` prevents the kernel from seeing any modules found by the ROM memory scan. This allows booting using only modules from the ethernet boot without seeing older modules being found first in the ROM. To use from the boot menu:

Select a boot method from the above menu: eb **nokrs=1**

- CF15141: Updates are needed for `dbgextns`.

OS-9 for 68K: Edition 7 of `dbgextns` permits memory in the process descriptor fragment list as well as the SSM map.

OS-9: A user-state version of `_os_chkmem()` has been added to allow the Debugger to read and write memory properly when attached to a module (rather than fork). This requires edition 173 of the kernel and edition 73 of `spfndpdc`.

- CF15214: Update 16550 driver to properly handle buffer overrun.

Previously, the 16550 driver overwrote the data buffer when no more room was available in the buffer. This problem has been corrected.

- CF15480: `sc16550` doesn't allow arbitrary device driver global data.

The parameters to `_os_irq()` used by `sc16550` were only correct if the `drvstatic` variable was the first item in the static storage. This problem has been corrected by passing the correct value to `_os_irq()`. This is not a problem unless you modify the shipping sources for `sc16550`.

- CF15676: Time zone problems.

Corrected standard library TZ time zone conversion for CAT from GMT-60 to GMT+120. Also, the Ultra C Library Reference Manual and Using OS-9 Manual had several errors regarding the supported time zones. Corrected the time zone tables.

- CF15758: PCF may loop forever when trying to delete a file on a corrupt file system.

PCF has been modified to detect this case and resolve the problem.

- CF15821: FPU does not get initialized on MVME162

The `systype.d` file in the MVME162 port directory did not include proper spacing in the extends list between `fpsp` and `fpu`. As a result, FPU was not initialized for this port. The spacing has been corrected for future releases.

- CF15849: The RAMdisk driver source code is misleading in regards to static storage.

The RAMdisk driver source code was fixed to keep all device driver global static storage in a single structure. This is important because RBF expects a structure of type `rbfdrv_stat` to appear as the first item in the static storage. If everything is in a single structure and the first item of the structure is of type `rbfdrv_stat`, then this requirement will be met.

- CF15979: `curses.l` can cause linker errors because it is compiled for short code `curses.l` for OS-9 for 68k was rebuilt using compile options that specify short data and long code references. This should correct any linker errors related to `curses.l`.
- CF16285: `btf.m` and `reach32.m` macros for 68k do not ship with the product. These two macro files were described in the documentation, but not shipped. They are now shipped with the 68k OEM package.
- CF16347: PCF problem deleting files with long file names if full path was specified.

PCF had a problem deleting files in a directory when multiple files with long file names were in a directory and a full path name was specified. The problem only showed when the directory slots consumed over two clusters. This problem has been resolved.

- CF16527: PCF for OS-9/68K returns #000:214 (File Not Accessible) error when trying to open file with the `FAMAPPEND` mode bit set.

The PCF file manager was improperly detecting file access when append mode was specified. This has been resolved with edition 84 of PCF for 68K

- CF17086: The Flash version of the RAMdisk driver crashes if disk is over 1MB
The `init.c` source file for the RAMdisk driver was fixed to allocate sufficient room for writing the initial RAMdisk sectors.
- CF17205: RAM disk driver real-time response needs to be increased.
When copying large blocks of data that could affect real-time performance. The RAM disk driver will now yield, to improve real-time performance.
- CF17364: The `sc16550` driver needs to mask IRQs in some circumstances
The 16550 serial driver (`sc16550`) was modified to optionally mask interrupts so it was not re-entered while handling interrupts.
- RSYS9001: New `sysmbuf` system subcode added for 68k.
A fourth system call subcode was added to `sysmbuf` to provide real-time usage status of mbuf pool.
- RSYS9065: 68k `tftpd` now correctly forks `tftpdc`

The 68k version of `tftpdc` did not correctly fork the `tftpdc` connection handler. This has been corrected.

- **RSYS9115:** Pass through file manager definitions added to 68k `path.h`
Pass through file manager (PTF) definitions were added to `/mwos/OS9/SRC/DEFS/path.h`.
- **RSYS9115:** Universal `getstat/setstat` parameter block added to `srvcb.h`
A universal `getstat/setstat` parameter block was added to `/mwos/SRC/DPIO/DEFS/srvcb.h`. It may be used when the parameters for a sub-code can be easily passed in a couple of integers. When used, it saves the trouble of having to call `_os_chkmem()` and then de-referencing another parameter block structure.
- **RSYS30623:** RBF/PCF used on flash media can cause excessive wear.
PCF was enhanced to include control for `CTRL_NOWRITE` which reduces the number of file descriptor updates to a minimum. This keeps PCF from constantly updating the modified date as writes are performed. The final modification date is written when the file is closed. RBF already has similar support.

3

Host Applications

This chapter contains release notes for host applications used with OS-9 for 68k v1.3.

For information about...	Go to this page...
Hawk Notes	14
TECH-CHECK Notes	25

Hawk Notes

The following sections represent changes and updates to Hawk since the last release.

Enhancements

- The Differencing feature has been enhanced. A new Differencing submenu provides access to differencing features, including text files, binary files, and directories.
- The Show References to (selected symbol) dialog and Show Symbols Called or Used by (Function or Class Name) dialogs now provide information on how C, C++, Java or C# symbols are used in your project.
- Regular expression support is available for both strings and numbers, allowing you to more easily identify which code is ChromaCoded.
- CodeSense has been enhanced as follows:
 1. You can select the Advanced Options tab on the **Customize -> CodeSense Global Configuration** dialog to access several settings for enhancing the functionality of CodeSense.
 2. A Maximum Cache Size slider on the CodeSense Advanced Options dialog has been added. This feature allows you to change the number of hits from CodeSense lookups that are retained in memory.
 3. Mark the option **Filter Matches by File Type** on the **Customize -> Language -> CodeSense** dialog to filter lookups for the file type(s) selected on the Language dialog. To see CodeSense information that applies to any file type, disable this option.
 4. CodeSense now fully supports Java packages and C++/C# namespaces.
 5. You can now specify which parser language to use for a given file type or library database.
- Documents opened in Hawk are now accessible from a tab bar across the top of the Edit window.
- Three new check boxes are available from the **Customize -> Environment -> General** tab:
 1. Enable tab virtual space
 2. Enable EOL virtual space
 3. Standard MDI presentation
- The **File View** window (now known as the **Project** tab of the **Project Window**) displays the current project space with all of its member projects contained underneath. Each project can be further expanded to display the member files in hierarchical form.
- The **File -> New (Create New Document)** dialog allows you to start your editing session by populating a new document with one or more code snippets from the

CodeFolio directory. Include file headers, function headers, and more to eliminate repetitive typing, and assure consistent code formatting.

- Microsoft Visual Studio.NET projects (.VCPROJ, .VBPROJ and .CSPROJ) and solutions (.SLN) can now be imported into Hawk.
- The underlying functions for the CUA, BRIEF, vi and Epsilon keymaps are now documented online to illustrate their functionality, parameters, return values and keyboard shortcuts, when applicable.
- Both named and unnamed bookmarks can now be saved in the Bookmark Database when a file is closed, either during a session or at the end.
- CodeWright 7.5 integrated

The latest version of CodeWright from Borland was integrated into the RadiSys Hawk Integrated Development Environment. CodeWright 7.5 includes many improvements in user-interface design and work flow mechanics.

- The **Watch** window should maintain the expressions across Hawk sessions.

Hawk was enhanced to save and restore the Watch expressions across Hawk sessions. The key or the right-click menu can be used to remove Watch expressions from the window.

- **Process I/O** window is not very useful

The **Process I/O** was revamped to include support for:

- clearing the “screen” when debugging starts
- a right-click menu with **Copy** and **Clear** options
- automatically scrolling to the bottom so that new information is visible
- pasting to the window to send information to the target
- The **Command I/O** needs more functionality

The **Command I/O** window now supports a right-click menu with **Copy**, **Clear Screen**, and **Clear Buffer** options. In addition, the **Command I/O** window resizes properly when initially displayed.

- The Hawk debugger needs usability enhancements.

The Hawk debugger was enhanced by adding these usability features:

- an **Exit Debugger** button is not available on the button bar (previously, a menu item had to be used)
- an “**Are you sure?**” dialog is displayed when exiting the debugger to ensure the exit was intentional.
- editing of locals or watches by simply double-clicking on the expression
- The Hawk Project Manager was enhanced.

The Hawk Project Manager has these enhancements:

- the intermediate file directories (ROF and/or I-code intermediates) can be specified as relative to the project's directory. Then, if the project is moved or reused elsewhere, the intermediate file directories move with the project.
- the `build.log` file is created within the project directory, instead of at the current directory at the time of the build
- CF9436: Add client installation.
 Client Installation is now available. Hawk will continue to use the path specified in the `MWOS` environment variable to find all program files and the rest of OS-9. If it exists, the path specified in a new environment variable named `HAWKCLIENT` will be used for all user configuration files. If any of the user configuration files do not exist in the `HAWKCLIENT` path, they will be opened on the `MWOS` path. The product installers have an additional setup type called, "Install Client", which functions similarly to the other options, but displays an additional dialog prompting for the path to the server directory. The only files installed with this option will be the user configuration files installed to the path chosen for `HAWKCLIENT`.
- CF12712: Delete files out of dialog.
 A handler has been added to the **Directory Selection** dialog so the **Delete** key can be used to delete a folder from the list. Once a folder has been deleted, the folder above it will be highlighted.
- CF12735: C++ support for `hawk_control_log` in `s1mlib.h`.
 If the log file is included in a `.cpp` file, the `hawk_control_log` function will have "C" { } code around it.
- CF13004: Enhancement needed for Hawk and memory directories.
`Hawkdbg.dll` and `hawkdbgr.exe` have been updated. Hawk now contains an edit box that allows you to input a module directory to load a module into on both the **Load** dialog and the load portion of the **Connect** dialog.
 Also, edition 71 of `spfndpdc` was modified. If an older version of `spfndpdc` is used and you specify a module directory in which to load a module, the module will be placed in the root directory.
- CF13051: Load all modules.
 An item has been added to a component's **Property** dialog and pop-up menu called **Exclude from Load All**. If you select this item, the component will be skipped when the **Load All** item is executed.
- CF13084: Enhancements for Hawk's `build.log` file.
 The `build.log` file is now saved in the project directory. In the log, the “\r” character that was used for a new line in the command line memo has been replaced with a normal line feed.
- CF13085: The `hawkdbgr.exe` should have a `-z` option to load multiple files.
 Code was added so that a file can be used to input a list of modules to load.
- CF13219: Debug parallel processes in the Hawk Debugger.

Hawk will now start a stand alone debugger if the Hawk IDE is currently debugging a process and **Debug -> Connect** is selected.

- CF13263: Display of data types in Hawk.

If you right-click on a value in the **Locals** or **Watch** window, you can now select two more menu items (**Display Decimal** and **Display Hexadecimal**); each can be used to change the display type of a single entry in either grid.

- CF13603: Add 64-bit support for Hawk.

Code was added so that Hawk can display 64-bit values correctly in the **Locals** and **Watch** windows.

- CF13760: The Hawk Debugger **Connect to module** pull-down menu keeps no record of previous modules.

Code was added so that eight entries are saved in the **Attach to module** edit box.

- CF14012: Add watches in the Debugger via **Watch** window.

This enhancement has been made.

- CF14047: Specify multiple module to attach in Hawk Debugger.

The `-AM` command line option was changed to allow the input of a list of modules.

- CF15153: Hawk GUI cannot specify the same library twice.

Hawk has been enhanced to allow the same library to be selected twice.

- CF15872: Fix VT100 emulation for **Process I/O** window.

The **Process I/O** window didn't handle arrow key sequences properly for VT100 emulation. Now, there is a `hawkterm.txt` file that determines the keymapping of the non-ASCII keys - such as keypad, arrow, delete, etc. The user can edit this file to achieve the desired key sequences.

- CF17322: Hawk now supports environment variables.

Hawk now supports references to environment variables like `$(MWOS)` or `%MWOS%` in file paths or other settings which are passed to compiler command lines.

Resolved Problems

This section provides a list of Hawk-specific CustomerFirst incidents and how they were resolved for the current release.

- CF: Hawk does not allow debugging if a project path contains spaces.

Previously, Hawk would not allow debugging if the pathname to the module to be debugged contained a space. This has been fixed.

- CF5720: Editing the `.mpj` file of a closed project can cause a source file corruption.

The source file is no longer corrupted. In the current version of Hawk, when you save a project with the `.mpj` file open in the editor, the dialog will now state

that the file was changed by an external process and ask you to either reload the file or save it as another name.

- CF5723: The **Build** window does not get cleared when **Close** is selected from the **Project** menu.

Hawk has been updated so that the errors in the **Build** window are cleared before each build. Therefore, if there are errors in a new project and you perform a build, the previous errors will be cleared before any of the new project's errors are displayed. The errors can also be cleared at any time by doing a right click in the output window and selecting **Clear Error File** from the pop-up list.

- CF11187: Hawk cannot launch the Profiler if the `MWOS` environment variable contains spaces.

Quotes have been added around the string that contains the path to the `java.jar` file that is executed to run the Profiler.

- CF11851: System-state Hawk needs the ability to boot a system in stages.

A “go boot” option was added to the Debug menu that has the same functionality as typing in `gb` when booting a system in ROMBug.

- CF12208: When you are debugging, if you perform a step-through command in Hawk using the `int*float` operation, it will result in an incorrect answer.

The bug was included with the daemon `spfndpdc`. It was not properly restoring floating point registers when doing a `step`, `next`, `go`, etc. This has been corrected with the current `spfndpdc` module.

- CF12344: Cannot set breakpoint in module source located in a directory with special characters.

The `mwsrdbg.dll` previously used many unix conventions and characters like '+' were not allowed in a pathname. The function that parses the breakpoint command has been updated to allow “+”, “.”, and other special characters in the pathname.

- CF12740: Corrupt source code folder.

Previously, the default size of the text box was 256. This would cause any source paths that were set over the maximum character length to be deleted. The folder is now set to have a maximum of 4000 characters.

- CF12854: After switching from user-state to system-state, the processor selection changes.

The processor selection for a component is now retained if the component type is switched.

- CF12856: Problem setting break points.

Previously, breakpoints were both set and stored in the same manner. For example, if a user typed a function name and then tried to delete the breakpoint by selecting it on the left side of the source window, the filename and line number would not match any of the set breakpoints; consequently, the breakpoint

would be deleted. Hawk has been updated so that it will remember both the function name and the filename and line number of the breakpoint.

- CF12933: The **Rebuild all projects** item does not work.

Previously, if a project had more than one component when the **Rebuild all projects** action was performed, Hawk attempted to advance to the next project before all of the first project's components had completed building, thus causing access violations. The problem has been corrected.

- CF12942: The Hawk Debugger cannot initiate a connection if a parameter includes “@” character.

The function used to parse the string passed to `mwsrddb.dll` during a program fork has been rewritten to allow the @ character to be in the environment variables or command parameters.

- CF13153: Deleting a project in Hawk causes an access violation.

Hawk looked at the function to delete a component when the **Delete** key was pressed while a project was highlighted. The `delete` function now checks if the highlighted item is a component or a project and then calls the correct routine to delete the selected item.

- CF13196: Edit memory in Locals window.

Previously, the function used to edit memory inserted an extra two bytes of data. The function has been updated so that it writes the correct amount of data; thus, modifying a “short” will not change the value of a variable stored right after it.

- CF13209: Hawk **Process I/O** window does not remain scrolled at the bottom.

There is a property of the control `TAdTerminal` called “`ClientOriginRow`”. If you set this value to 1, the terminal window will automatically scroll down so that the top line of the terminal window is visible, regardless of where you set the scrollbar.

- CF13210: Hawk **Process I/O** window adds extra carriage returns.

Previously, Hawk added an extra carriage return each time the Return key was pressed in this window. This problem has been fixed.

- CF13337: The underscore character cannot “prefix” a renamed component.

When you select the **Rename** item by right-clicking on a component, you can now also select component names that begin with an underscore.

- CF13338: Source files of type `*.a` are not visible in the **Create New Component (Unit)** dialog.

The text of the file filter displayed type `*.a`; however, that item was missing from the actual filter component. It has been added; thus, assembly files are displayed with the other source files.

- CF13404: Hawk cannot debug multi-threaded applications.

The x86-specific code in `mwsrdbg.dll` was not reading the pointer to the thread function correctly; this caused an error when the new debugger was launched. This problem has been fixed.

- CF13469: Hawk does not build modified source code.

Previously, the method used to determine whether or not a file needed to be rebuilt did not work. A new method has been implemented.

- CF13493: Hawk is appending `mwhawk.pst` to open source files.

This was a problem in CodeWright 6.6 that was fixed with version 7.0. CodeWright 7.0 has been integrated into this Hawk 2.3.

- CF13505: Hawk displays an error message after the current project is changed.

Previously, switching between two projects in a project space several times caused an access violation. This problem has been corrected.

- CF13594: Selecting the `.cs` file type in the **Customize -> Environment** dialog causes Hawk to hang.

Code was added to `cwstart.dll` to support the `.cs` type in the **Customize -> Language** dialog.

- CF13763: Add missing PowerPC boards to chip selection in Hawk

The MPC 555, MPC 750, MPC 8240, MPC 8260, and PPC 405 processors were not available when creating a component. These processors have been added to the list of available processors for PowerPC.

- CF13812: The Hawk Debugger cannot find the current file.

Previously, Hawk would read the symbol file as it loaded a module and then move to main. If the source path was not set correctly, it would prompt the user to find the file. This allowed the file to display, but setting breakpoints on the file did not work because the symbol information was not updated with the correct path.

A dialog has been added in the latest version of `mwsrdbg.dll` that prompts you to find the file as the symbol information is read. The path to the source path will then be added to the source path debug options so you will not be prompted to find other source files in the same directory. Thus, breakpoints can now be set, as the path information for the symbol now matches the file's path.

- CF13860: It is not possible to browse file paths in the **Project Properties** dialog.

The version of the component used to browse through the folders was not supported under Windows XP. Hawk has been rebuilt with an updated version of the component; this error no longer occurs.

- CF14057: There is no error when an attempt is made to load non-existing files.

Previously, when a target load was performed, the error message was not descriptive enough for incorrectly input filenames. Currently, in the case that the load fails, a check is done to see if the filename the user input exists. If the file does not exist, the message, "Module: <ModuleNameEditBox->Text> could not be found" will display.

- CF14079: The Hawk Debugger bus traps when adding first watch entry.
This problem has been resolved; the bus trap no longer occurs.
- CF14261: An access violation occurs when you right-click in the **Add Watch** window in the Hawk Debugger.
A check has been added to avoid this problem; the pop-up menu now appears and items can be added to the window using the **Insert** menu item.
- CF14305: The **Advanced** button does not work properly in the **Version Control Setup** dialog.
A patch has been included into the current version of Hawk that fixes this problem.
- CF14758: Forking a module in Hawk doesn't handle parameters properly.
Hawk was not parsing the command correctly to get just the name of the module being forked if command line parameters were used. The effect of this was Hawk was unable to find the correct `.dbg` and `.stb` files because it was looking for the wrong things. This has been fixed in version 2.3.4.2 of `mwsrddb.dll`.
- CF14770: Hawk problems when syntax errors are in source code
The Microware error parsing functions were missing from a previous release. This would cause the functions that show where a certain error occurred to not work. The error parsing function has been added back in version 2.3.4.0 of `hawkprj.dll`.
- CF14847: The Hawk Debugger crashes when setting breakpoints in files with a long pathname.
Previously, Hawk did not operate successfully if the pathname was too long on a source filename. This problem has been resolved in `hawkdbg.dll` and `hawkdbgr.exe`.
- CF14865, CF14894: If a connection is lost, Hawk won't allow another right-click debug
There is a flag that is set when a component is being debugged that is used to make sure the debugger is started twice for the same component. This flag was not being reset to false when a connection with the target was lost. This would result in the inability to debug a component again if the connection with the target was lost for some reason. This has been fixed in version 2.3.1.29 of `hawkdbg.dll`.
- CF14866: the "**ident module**" function doesn't work in Hawk.
The "**ident module**" menu item in Hawk now works correctly. Previously, incorrect values were reported or Hawk would lock up in an infinite loop.
- CF14867: When editing register values, there's no cursor
The row width in the registry window was too small to allow for a cursor to be drawn. The width has been increased and the cursor now fits in the row. Because the display of the registers window is always in hexadecimal format the

editing of the register values has been changed so that it is also always in hexadecimal format.

- CF14870, CF14675: Hawk standalone debugger fails to load module (system-state)

`undpdc` and `spfndpc` weren't exiting correctly when a module was loaded. This was causing an access violation in Hawk when a module was loaded. The modules have been fixed in edition 72 and the access violation no longer occurs.

- CF14894: "This component is currently being debugged" error when not debugging.

There is a flag that is set when a component is being debugged that is used to make sure the debugger is started twice for the same component. This flag was not being reset to false when a connection with the target was lost. This would result in the inability to debug a component again if the connection with the target was lost for some reason. This has been fixed in version 2.3.1.29 of `hawkdbg.dll`.

- CF15025: A system-state debugging error, "Invalid class typecast," occurs when changing `mem` value.

This problem has been resolved.

- CF15062: Cannot step out of a signal handler.

In the previous version of Hawk, it was not possible to step out of a signal handler to the source line that was interrupted by the incoming signal. It is now possible to return from a signal handler with the following caveats:

1. If you use `_os_intercept()` to install the handler, you must step over the `_os_rte()`.
2. If you use `intercept()` or `signal()`, the "step next", "step source", and "return from function" options will all work properly.

- CF15251: Hawk fails to execute some programs.

Hawk was failing to set up a forked process standard input correctly. Thus, any program that performed any operations with standard input would fail. This problem was more frequently seen when older MS-DOS based programs were forked from a MS-DOS batch file.

- CF15616, CF15682, CF15704: `_MicrowareErrorInfo` error parser missing. Unable to click on compile error and have source automatically opened.

The UltraC error parser function, `_MicrowareErrorInfo` was missing. It has been added to `hawkprj.dll` version 2.3.4.3. Now, clicking on a compile error line in the output window will open the appropriate source and place you on the offending line of code.

- CF15638: Hawk/Hawk Debugger Process I/O Window carriage return and backspace problems.

Corrected problems in the Hawk and Hawk Debugger Process I/O window that would eat or double backspace and carriage returns. This was due to a third

party vt100 emulation package used by the **Process I/O** window. Reverted to a simple TTY package to correct the problem.

- CF15673: Hawk does not terminate `spfn̄p̄d̄c` on target after a module load. After a Hawk module load, the `spfn̄p̄d̄c` process on the target would not terminate. This has been corrected.
- CF15675, CF16120, CF16231: Hawk help no longer comes up when **F1** or **Alt-F1** are pressed.

CodeWright has removed support for the API Assistance that came up when **Alt-F1** was pressed. As a result, Hawk no longer has this functionality.

Hawk now ships with the following help files found in the `MWOS\DOS\BIN` directory:

- `compiler.chm`
- `os.chm`
- `gfx_audio.chm`
- `network.chm`

When a function is highlighted with the mouse and **F1** is pressed, the associated help file will be opened with help describing the function.

- CF15707: CodeSense not working with Hawk v2.3.4.2
This issue was resolved by integrating the latest version of CodeWright into Hawk.
- CF15972: Several messages are coming up when debugging threads.
When debugging threads, debug message boxes were displayed on the screen. These messages have been removed.
- CF16047: System state debugging through a gateway does not work.
The `llarp` library improperly implemented hooks for gateway/route support for low level drivers to use. The code has been modified to resolve the problem.
- CF16484: The Profiler can not show symbol information for ARMv4 big endian, XScale, nor SH-5m.
These newer module header sync codes have been added to the profiler's `.stb` file parser so that the symbols will now be displayed correctly.
- CF16501: Hawk does not properly open source files to lines with errors
If errors were discovered when compiling in Hawk, double clicking on an error line or using the "Goto next message line" button in Hawk would not always go to the line containing the error and would report "Unable to locate file" on the status bar. This was caused by the use of two error parsers in Hawk. `_MicrowareErrorInfo` is the only error parser used in Hawk now, and it will properly open the source file to the line containing the error.
- CF16502: The Hawk debuggers print that signal zero was received when it was really non-zero.

The Hawk debuggers now correctly print the signal value that the remote process received. This was only a problem when the remote process did not have a signal handler.

- CF16679: Hawk does not regenerate dependencies when relative paths are used

Hawk did not regenerate dependencies when relative paths were used, and the current working directory (cwd) was not set to the project directory. Hawk now searches for files relative to the cwd and the project directory.

- CF16701: Hawk changes the current working directory (CWD) when adding a library.

Hawk has been modified to not change the CWD when adding a library to a component. Requires `HawkPrj.dll v2.3.4.11`.

- CF16941: The Hawk debuggers incorrectly display `long long` values for x86 and SH-4 targets

The Hawk debuggers have been fixed to correctly display 64-bit integer values for both the SH-4 and x86 target types. Previously, the target integer value's words were incorrectly swapped prior to display.

- CF17053: Hawk cannot debug on a Pentium 4 target

The Hawk debuggers (`mwsrdbg.dll`) were fixed to understand the CPU type returned by a Pentium 4 system.

- CF17306: Double-clicking a stack-frame in the stack traceback window does not put the debugger on the correct line.

The Hawk debugger was fixed to place the cursor on the correct line when double-clicking a stack traceback entry.

- CF17345: Hawk project files could sometimes reset

Switching certain compiler settings in Hawk would cause the project to be reset to defaults. This was caused by the use of parenthesis in two `hawkdata.xml` tags. Changing settings so these tags were used would create an invalid project `.xml` file which could not be parsed and was therefore reset to defaults. The master `hawkdata.xml` file has been updated to use only valid XML tags.

- RSYS30723: Hawk's CPU window should be an Output tab

Hawk 2.4 was implemented such that the CPU window (the window for docking all the various debugger windows into) is implemented as a tab in the Output folder control. This consumes much less screen real estate than the previous implementation as a button bar.

- RSYS30356: Register windows can show old values for registers.

Hawk was fixed to ensure that the initial display of a register window (MPU, FPU, MMU, or special) shows the current values of the registers rather than out-of-date values.

Known Issues

- CF14222: The Hawk Debugger only allows nine paths to source code directories.

All of the Borland controls suffer from the same limitation when it comes to the length of a string that can be saved. Refer to the Borland help file for a description of this problem.

TECH-CHECK Notes

- CF 14584: Tech Check gets memory access error in certain circumstance
In certain cases TechCheck would get memory access errors when trying to save an incident report as text. This problem has been fixed in version 1.1.3.2 of `tcheck.exe`.
- CF 14717: Remove UK and French support offices from Tech-Check
The UK and French support offices were removed from TechCheck in version 1.1.3.1.

4

Components

This chapter contains processor-independent release notes for OS-9 components.

For information about...	Go to this page...
OS-9 Compiler Notes	28
Networking Notes	39
OS-9 Utilities Notes	46
MAUI Notes	51
Add-Ons	52

OS-9 Compiler Notes

The following sections contain release notes for the OS-9 Ultra C/C++ compiler.

Enhancements

The following list describes general enhancements made to the Ultra C/C++ compiler for this release. Where applicable, CustomerFirst (CF) and ClearQuest (RSYS) incidents are included.

- The following processors now support 64-bit integer operations using the non-standard data type `long long`:
 - PowerPC
 - ARMv4 (both big- and little-endian)
 - SH-3/SH-4
 - SH-5
 - x86/Pentium
 - MIPS64
- CF8155: XCC: Better makefile-compatible dependency generation is needed.

Dependency generation in Ultra C/C++ is not adequate. The Ultra C/C++ Compiler has been enhanced to emit makefile dependencies that can now be used in an `os9make` makefile. Refer to the documentation for the `-pm` option in the *Using Ultra C/C++* manual for more information.

- CF15546: Ultra C should support `pthread_mutexattr_getkind_np` and `pthread_mutexattr_setkind_np`.

The Ultra C threading libraries were modified to include support for these two functions. Ultra C now supports three different kinds of mutexes:

1. POSIX standard, non-recursive
2. POSIX super-set, recursive
3. high-speed, non-recursive

Refer to the Ultra C Library Reference and Using OS-9 Threads manuals for more information.

- The message queue facility was expanded to include inter-process communication.

The `mq.1` library was enhanced to include support for message queues that allow one process to communicate with another. Previously, message queues were limited to being within a single, threaded application. Now, a message queue whose name begins with slash (/) will be visible to all processes and threads on the system. Inter-process message queues support virtually all the same features as inter-thread message queues: POSIX compliance, `_MQ_O_IMMEDIATE` and `_mq_notify_write()`. Inter-process message queues do not support `SIGEV_THREAD` notifications.

The inter-process communication is facilitated by a new file manager (`msgman`) and device descriptor (`mq`). This is a general-purpose message passing file manager; similar in functionality to `pipeman`. It supports both named and unnamed message queues.

- CF16377: Ultra C/C++ does not accept `switch` statements with 64-bit expressions

Ultra C/C++'s 64-bit integer support has been enhanced to include support for 64-bit `switch` statements. They are implemented internally using 32-bit constructs.

- CF16586: The Ultra C/C++ front-end is adhering too closely to the ANSI standard.

The front-end has been relaxed in these ways:

- The `long long` data type is now allowed in all modes (K & R, ANSI extended, and ANSI) without any diagnostic messages. The diagnostic messages can be restored by passing an option to the front-end via the executive: `-fe-diag_[warning|error]=450`
- C++ style comments (`//`) are now allowed in C code unless `--cpp_comments_notallowed` is passed to the front-end.
- Empty translation units (source files) are now allowed in all three modes. The diagnostic messages can be restored by passing an option to the front-end via the executive: `-fe-diag_[warning|error]=96`
- e500 and 8540 targets were added

The Ultra C/C++ executive (`xcc`) was modified to have support for the e500 and 8540 targets. These targets have unique defines (e500 defines `_MPFE500` and 8540 defines both `_MPFE500` and `_MPFPPC8540`). In addition, `mkdatmod`, `idbgen`, `rpcdbgen`, `editmod`, and Hawk were modified to include these targets.

- CF16919: Some C++ objects are slow to create and destroy

The Ultra C/C++ compiler header files and libraries were updated to speed up the implementation of mutexes internal to the C++ libraries. This resulted in a roughly 50X speed improvement.

- CF17382: OS-9 should ship with the `.stb` files for `cs1` and `mt_cs1` to aid in debugging

The shipping images of OS-9 have been modified to include the symbol files (`.stb`) for `cs1` and `mt_cs1`. This should help when problems occur in the C shared libraries.

- CF17416: Daylight saving time start and end time will change starting in 2007.

The Ultra C libraries handling for daylight saving time was updated to reflect that, in the United States, it starts the second Sunday in March and ends the first Sunday in November in 2007 and years after.

- The `stat()` and `fstat()` functions should support `st_ino`.

The `stat()` and `fstat()` functions were enhanced to provide a meaningful value for the `st_ino` field of the `stat` structure. This can be used to determine the uniqueness of multiple files on the same device.

- RSYS33816: There should be a way to declare that an assembly language escape does not call any functions, maintaining the “leaf” status of the containing function.

Ultra C/C++ for all processors now supports the assembly language escape pseudo function `__reg_nofunccall`. This can be used for the stated purpose. This has the benefit of generating more efficient assembly language for trivial functions containing assembly language escapes.

For example, this simple PowerPC function:

```
void *getsp(void)
{
    void *ret;
    _asm(" mr %0,sp", __reg_r3(__obj_assign(ret)));
    return ret;
}
```

Translates to this assembly language (fully optimized):

```
=getsp:
    mflr r0
    stw r0,4(r1)
    stwu r1,-8(r1)
    lwz r0,_stbot(r2)
    cmplw cr0,r0,r1
    ble+ =_L0
    bl _stkhandler
=_L0
    mr r3,r1
    addi r1,r1,8
    lwz r0,4(r1)
    mtlr r0
    blr
```

If `__reg_nofunccall` is used:

```
void *getsp(void)
{
    void *ret;
    _asm(" mr %0,sp", __reg_r3(__obj_assign(ret)),
        __reg_nofunccall());
    return ret;
}
```

that same function translates into this assembly language:

```
=getsp:
    mr r3,r1
    blr
```

Although this pseudo function is available for all processors, it only benefits ARMv4/StrongARM, XScale, SH-3, SH-4, and PowerPC processors.

Resolved Problems

This section gives a description of Customer First (CF) and ClearQuest (RSYS) incidents related to the Compiler and how they were resolved for the current release.

- CF 8178: The ARM back-end performs incorrect data area layout, which results in linker errors.

The ARM back-ends were fixed to recognize the fact that there are three different lengths of data area references for performing data area layout. This eliminates said linker errors.

- CF 11223: The `xcc.exe` file does not compute the correct ROF name in C89 mode.

Previously, the executive computed an incorrect ROF file name. It now computes the correct ROF file name.

- CF 11607: The `-qi` option for `xcc.exe` does not work.

The executive no longer puts `implicit_include` on all front-end command lines for C++ source files. Instead, it now only places it on the front-end command line if `-qi` is used.

- CF 12706, 12709: MIPS `const` qualified pointer is incorrect.

Previously, on MIPS processors, the `const` qualified pointer to `const` qualified objects (where the destination of the pointer was offset `0x7ff0` from the base of the module) was incorrectly interpreted as NULL. This problem was corrected; the front-end has been modified to initialize NULL `const` qualified pointers with `-0x8000`. This ensures that NULL `const` qualified pointers are unambiguous.

- CF 12759: `_os_modaddr` isn't getting translated

The trans modules (user/system state address space translation) for MIPS3000, MIPS32, MIPS64, SH-3, and SH-4 no longer fail to translate the input parameter to `_os_modaddr()`. This was causing modules to be reported as not found when they actually existed.

- CF 12957, 14091: The `mkdir()` and `ev_creat()` functions do not pass parameters correctly.

The following functions from `sys_clib.1/sys_cs1.1` were fixed to fill `errno` properly: `alm_atdata()`, `alm_atjul()`, `alm_set()`, `creat()`, `ev_creat()`, and `mkdir()`. In addition, `lseek()`, `ibrk()`, `ebrk()`, `getstat()`, and `setstat()`.

- CF 13021: The 64-bit integer left shift operator does not work.

Previously, the front-end failed to promote left shift operators to the required width. This led the back-end to produce incorrect code. The front-end was fixed to perform left shifts in the proper width.

- CF 13159: The `module.h` file's subroutine numbers are incorrect.

The related values and comments related have been corrected. A new `module.h` file has been created to resolve this issue.

- CF 13291: The `mktemp()` function does not work correctly.

The `mktemp()` function will now correctly set `errno` when it runs out of filenames to return. This function also ensures that the directory specified in the prefix exists before it returns any filenames. It will now handle any pattern ending in three or more “X's”. Also, `mktemp()` now returns at least 256 unique file names in a given directory.

- CF 13378: The 64-bit “+=” can cause a bad mnemonic to be emitted for PowerPC.

A typographical error in the PowerPC back-end has been fixed to allow the handling of a 64-bit `ident += const` construct, where `ident` is in a register. Programs containing this and other similar constructs can now be compiled.

- CF 13388: The front-end receives a General Protection Fault (GPF) after some `inline _asm()` statements.

The front-end received a GPF when it encountered a string constant in a macro invocation after an `inline _asm()` statement. This problem has been corrected.

- CF 13512: The short-data model on the ARM processor seems to use too-large of a symbol on an add instruction.

The executive was fixed to ensure that the files are passed to the back-end in the same order they are passed to the linker during data area layout performance. The linker no longer prints errors related to symbol offsets in this situation.

- CF 13564: The wrong prototypes for the `_os_exec()` and `_os_fork()` functions are compiled.

The prototypes for `_os_exec()` and `_os_fork()` were fixed so they could be successfully compiled in C++ mode.

- CF 13570: `_OPT_PROTOS` is undefined if certain header files are used.

The following header files no longer define `_OPT_PROTOS`: `UNIX/os9def.h`, `UNIX/os9time.h`, `UNIX/stat.h`, `UNIX/stringlist.h`, and `UNIX/times.h`.

- CF 13591: Using the `signal()` function in C++ delivers an error.

The `signal()` function is now properly prototyped so it can be compiled in C++ mode.

- CF 13743: The assembler cannot handle illegal expressions.

Illegal expressions, such as `==label` and `'xxx`, were not handled well by the RISC assemblers. The `==label` expression resulted in no code generation for the line, while `'xxx` resulted in a GPF in the assembler. These problems have been corrected.

- CF 13869: Fix failed support for 64-bit integers on PowerPC.

Previously, several features of 64-bit integer support failed for PowerPC. These problems have been corrected.

- CF 13978: RISC assembly optimizers fail to support the `lo()`, `hi()`, and `high()` functions in all contexts.
The RISC assembly optimizers were fixed to allow `lo()`, `hi()`, and `high()` in all valid situations.
- CF 13999: Add PCF support to `unix.l`'s `utimes()`.
The `utimes()` function in the `unix.l` C library was modified to include support for updating the times for a file on a PCF device (MS-DOS format).
- CF 14074: Arrays of `const` pointers do not work with `const` structures.
Previously, the code generated to handle `const` qualified pointers within arrays within structures was not working correctly. The front-end now correctly generates code for such data structures.
- Previously, the code generated to handle `const` qualified pointers within arrays within structures was not working correctly. The front-end now correctly generates code for such data structures.
- CF 14960: PCMCIA utility for ThinClient doesn't work
The `pcmcia` would cause user permission errors when it was run (000:164). To remedy the problem it has been recompiled with the `group.user` option set to 0.0.
- CF10556: The `__obj_assign()` function is not properly recognized by the Compiler.
The Compiler generated an “uninitialized” warning for the variable passed in `__obj_assign()`. This problem has been resolved in edition 46 of `cpfe`.
- CF12466: `fstream` file open fails with EINVAL for read and write files.
This problem has been corrected. The problem stemmed from the differences between UNIX and OS-9 `O_XXX` modes.
- CF13838: There is an internal error in Compiler.
Previously, the C/C++ front-end contained an internal error under certain circumstances when processing C++ code. The internal error resulted from an infrequent instance of circular data types. This problem has been resolved.
- CF14182: I-code-linked threaded applications can have stack recursion problems.
This problem has been corrected.
- CF14803: The Hawk Debugger cannot display types that are not directly used in a source file.
The Ultra C/C++ Compiler front-end was modified to ensure that all type information is copied to the `.dbg`, regardless of whether or not the compiled source file used the type. This can lead to larger `.dbg` files, but they have enhanced type visibility.

- CF15078: `__rwstd::__stl_tree_mutex` unresolved in threading C++ standard library.

The threading C++ standard library (`mt_cpplib`) now correctly resolves `__rwstd::__stl_tree_mutex`.

- CF15140: `strftime()` does not handle the specified buffer size correctly. `strftime()` has been fixed to ensure that it correctly returns zero when the specified buffer size is insufficient for the number of characters generated. Previously, it could overwrite data appearing beyond the end of the specified buffer size.
- CF15161: Ultra C can have problems with complex right-shift expressions involving constants.

If this expression involving the integer `x`:

```
(short)(x >> 16) >> 8
```

was seen by the I-code optimizer, it was translated into the following expression:

```
(short)x >> 24
```

This is not a correct transformation since information for the right-shift is lost by the cast of `x` to `short`. This problem has been corrected.

- CF15235: `snprintf()` and `vsnprintf()` do not handle threading well.

`snprintf()` and `vsnprintf()` have been re-implemented to handle threading correctly. They now correctly use locks and do not overwrite any I/O data structures that other threads might be using.

- CF15541: `editmod` fails to ignore illegal `#` directives in unincluded text.

`editmod` has been corrected to properly ignore any illegal `#` directives within text that is not included. For example, this sequence will no longer result in an illegal directive error:

```
#if 0
#unknown_directive
#endif
```

- CF15576: `optmips` can incorrectly common tail merge code.

The MIPS assembly language optimizer was fixed to compare all arguments to branch instructions, rather than just the destination label, when considering equality. This prevents creating branches to code that has different semantics than the prior code.

- CF15732: Complex expressions containing many comma operators can lose expressions.

The C/C++ front-end has been fixed to properly handle expressions containing many comma operators. Previously, depending on the configuration of the expression tree, parts of an expression could be lost in the translation from C/C++ source code to I-code.

- CF15885: The `functional` C++ header file doesn't compile correctly.
The `functional` C++ header file was fixed to be syntactically correct. Previously, the `mem_fun1_t` and `mem_fun1_ref_t` template types would instantiate with type errors.
- CF15935: `iopt` needs to better optimize calls to functions returning struct.
`iopt` was fixed to eliminate useless structure assignments that happen on the right-hand-side of other assignment statement. For example, for automatic structure variables `a` and `b` and function returning structure `c()`:

```
a = b = c();
```

If `b` was not used after this assignment, `iopt` failed to remove the assignment. `iopt` now correctly turns this expression into:

```
a = c();
```

and deletes the variable `b`.
- CF15955: String objects don't follow stream width, justification, or fill values for the stream.
The "put to" (`<<`) operator on string class objects was fixed to properly respect the justification, width, and fill values for the output stream. Previously, the string data was written without any of these settings.
- CF16035: `mem_fun_ref_t` templates don't compile when non-pointer types are used.
The `functional` header file was modified to allow such template instantiations to compile and work correctly.
- CF16113: ARM big-endian code generation is wrong for some narrowing in-memory conversions.
The ARM big-endian back-ends have been fixed to correctly handle memory references resulting from narrowing conversions of objects.
- CF16205: Constant pointer (CP) points to wrong location
When a child thread was linked with CSL and forked on processors with a CP, the CP would incorrectly point to the CSL code rather than the user module. `mt_csl` has been modified to properly calculate the CP when a child thread is forked.
- CF16260: I-code and CSL linked, threaded applications that use `intercept()` can not get signals
If an application is I-code linked with the threading CSL I-code libraries and does not call the `signal()` function explicitly and contains a reference to a function that has a reference to `_glob_sem` (e.g. `strcspn()`) then arriving signals would crash the application. This problem in `mt_csl.il` has been corrected.
- CF16368: `cpfe.exe` gets an application error when compiling extremely large files.

When compiling very large files, `cpfe.exe` could get an application error. This was caused by an overflowed number (negative) being used as an index number of an array). The number is now kept within bounds. `cpfe.exe` edition 51 resolves this problem.

- CF16427: SH-3/4 `switch` statements don't work when they contain large negative cases

The SH-3 and SH-4 back-ends were generating incorrect code for `switch` statements containing large negative numbers. This has been corrected.

- CF16522: Ultra C/C++ can use unpredictable instructions on ARM processors

The ARM back-ends (`bearm.exe`, `bearmbe.exe`, and `bearmv5.exe`) were fixed to no longer use a register as both a writeback base register and a listed register in `ldm` or `stm` instructions (e.g. `ldmia r7!, {r7-r8}`).

- CF16530: `pthread_kill()` cannot be used again after `longjmp()` is used to exit signal handler

The `pthread_kill()` function was fixed so that if a thread sends a signal to itself and uses `longjmp()` to exit the signal handler then `pthread_kill()` can be used again. Previously a lock was held such that no thread could return from `pthread_kill()`, among other things like `pthread_exit()`, `pthread_join()`, etc.

- CF16581: Exception related signal reception can cause application to crash

The C libraries were fixed to ensure that the mechanism used to generate exception related signals for a process does not crash. Previously, it did not work correctly if the exception happened with a different global data pointer that the main application (e.g. in the `cs1` module).

- CF16583: `dup2()` causes the calling program to crash if it has more than 32 paths

The `dup2()` function was fixed to correctly reallocate an internal buffer when more than 32 saved, open paths are required to perform the `dup2()`.

- CF16617: `stwcx` instruction errata for PowerPC 405

As per the errata, the `_os_sema_p()` and `_os_sema_v()` code was modified. Before the `stwcx` instruction, the `dbct` instruction was used. This should ensure the cache's store access will work correctly if an interrupt were to occur during the `stwcx` instruction.

- CF16634: Mutex problem with C++ libraries in threaded applications

It was discovered that the standard C++ libraries had a mutex problem with threaded applications. If an application using mutexes created a new mutex at the same location of a previously used mutex that was not destroyed and still contained a valid sync code, the new application would not initialize the mutex, and could try to use pointers in the mutex that were invalid. This resulted in random corruption and invalid data references.

A semaphore has now been placed around the mutex initialization, and threads competing to initialize the mutex will clear the structure before initializing the structure.

- CF16678: The Ultra C/C++ front-end crashes if `_asm()` statements appear in a namespace block

The front-end was fixed to not generate a protection fault in this situation. Further, it was fixed to not generate repeated duplicate `_asm` symbol errors in this same situation.

- CF16693: The `offsetof()` macro cannot be used in a constant integral expression in strict ANSI mode.

The `stddef.h` header file was modified to use a built-in compiler operator `__INTADDR__(x)` that takes an expression (possibly containing address-of or pointer casts) and returns a `size_t` value for the resulting constant. Now, `offsetof()` may be used as a bitfield size, enum initializer, array dimension, or case label in strict ANSI mode.

- CF16795: The `iterator` class does not have a not equals operator for a `const_iterator`.

The `list` header file was modified to include a not equals operator for a `const_iterator`.

- CF 16796: C++ problem with modified `new` operator

`cpfe.exe` has been updated to distinguish between a placement `new()` and an overridden `new()` and should now handle `new()` with extra/different parameters.

- CF16868: `cpfe.exe` can crash when `-c` is used

The Ultra C/C++ front-end (`cpfe.exe`) no longer crashes when checking pointer casts for targeting a string constant. The general protection fault was avoided by ensuring that the casted ampersand is on an identifier for a string constant and not something else.

- CF16870: The 386 back-end can emit code that has `imul.b` instructions that destroy valid data in `%ah`

The 386 back-end (`be386.exe`) has been fixed to no longer destroy valid data in `%ah` when `imul.b` is used.

- CF16871: `cpfe.exe` can crash on certain pointer casted ternary operators

The Ultra C/C++ front-end (`cpfe.exe`) no longer crashes when faced with a pointer casted ternary operator with a compile-time-known outcome. The general protection fault was avoided by ensuring that the ternary operator remained in the tree after folding.

- CF16916: Certain 64-bit memory stores on PowerPC result in illegal assembly language

The PowerPC back-end has been updated to no longer generate incorrect assembly language when the results of certain bitwise operations are stored to the stack in a 64-bit automatic.

- CF16941: `double` values can not always be converted to `unsigned long long` values

The C runtime libraries (`os_lib`) have been corrected to more accurately translate `double` values to `unsigned long long` values. This problem effected ARMv4, XScale, SH-3, SH-4, and PowerPC. Currently, `double` values in the range (shown here in hexadecimal) `0x0` to `0x2000000000000000` can be correctly converted. The results of conversions outside this range are undefined.

- CF16950: Very long command lines can cause `xcc` to crash.

`xcc` was fixed to avoid internal buffer overflows when there are more than 4096 characters of command-line options.

- CF17002: `ioctl()` is not suitable for non-networking file managers/drivers.

The `ioctl()` `unix.1` library function was fixed to pass unknown `ioctl`'s to non-networking file managers/drivers using the normal `setstat` protocol. Refer to the `sg_codes.h` and `srvcb.h` header files for information about the macro for the `setstat` code (`SS_IOCTL`) and the structure of the parameter block (`ss_ioctl_pb`).

- CF17066: 64-bit functions can return the wrong value on PowerPC

The PowerPC back-end (`beppc.exe`) was fixed to ensure that both `r3` and `r4` are maintained until the function returns.

- CF17331: `mktime()` update to honor `tm_isdst` field.

For time zones with Daylight Saving Time, `mktime()` in `clib.1` has been updated to honor the `tm_isdst` field when setting the ambiguous hour that repeats when time rolls back an hour. For example, when specifying 1:30am on the day that time rolls back at 2am in the USA, `tm_isdst=0` will be non-DST 1:30 (after the roll back to normal time), and `tm_isdst=1` will be DST 1:30 (before the roll back).

Networking Notes

The following sections include the release notes for the current versions of SoftStax and LAN Communications.

Enhancements

The following list describes general enhancements made to SoftStax and Lan Communications for this release. Where applicable, CustomerFirst (CF) and ClearQuest (RSYS) incidents are included.

- `spf.h`
`SPF_SS_ROUTEMSG` now adds a route with `rt_msghdr`.
`spf_pps_idata` is changed to `spf_pps_resv`.
- IP functions have been moved to `spfuncs.h` and the legacy LAN Communications stack's IP functions have been moved to `spfuncs4.h`.
- `sproto`
The `pullup` function has been added in `dr_downdata()` for hardware drivers that do not handle mbuf packet chain.
- CF 13615: `mbdump` should provide free lists.
The `mbdump` utility should have an option that prints an mbuf allocation table. The `-a` command line option was added to provide this functionality.
- The `sendto()` function supports item address structure for the legacy LAN Communications stack.
- CF13945: Add FTP support.
`rt_msg_sdl_index_update()` was added into SPIP edition 70.
- CF14129: Insert hardware address from `spenet` into the route table.
This enhancement has been made. Now, when SPIP receives `RTM_ADD` via `IP_SS_RTMSG`, it adds a host route entry in its route table and use the hardware address instead of IPv4 address to send mbuf to `spenet`. SPIP removes a host route entry when it receives `RTM_DELETE` via `IP_SS_RTMSG`.
- CF14354: Add multicast packet routing support for IPv4.
This enhancement has been made in edition 8 of `netstat`, edition 82 of `spip`, and edition 16 of `spraw`.
- CF15086: TCP/IP eliminates predictable multi-tasking response times.
TCP/IP is now processing regularly; it now operates with an extremely small time interval while the network is running. The software preemption depends on tick rate.
- CF15237: `sysmbuf` is not saving or restoring all 32 bits of the status register.
The OS-9 module `sysmbuf`, in an attempt to mask interrupts, was failing to save and restore the upper 16 bits of the machine's status register. This problem has been resolved.

- CF15244: `nfsc` can corrupt memory when UID/GID mappings are used.
`nfsc` has been updated to create a data module of the appropriate size when UID/GID mappings are used. Previously, the created data module was small, which could have lead to corruption of nearby memory.
- CF15261: Add support for `blksize` and `tsize` in `tftp`.
 To improve download performance, `tftp` has been enhanced to support `blksize` and `tsize` as described by *RFC 2347*. The `blksize` command allows larger or smaller blocks of data. Larger blocks are an advantage when few, if any, blocks have to be retransmitted. Use the `tsize` command when you get files to pre-allocate the disk space, which improves the speed of the writes. This enhancement is valid starting with edition 2 of `tftp`.
- CF15367, 15397: Add support for block sizes to `tftpd` that are greater than 1428.
 In edition 215 of `tftpd`, the following updates have been made to resolve this issue:
 1. The block size limit has been increased from 1428 to 8192.
 2. `tftpd` has been modified to allocate the memory buffers dynamically, rather than strictly to conserve memory.
 3. `tftpd` has been modified to use `mempy` instead of byte copies in binary mode, allowing for faster transfers.
- CF15252: Apply better security against ISN attacks.
 NetBSD1.5.1-based TCP does not have strong security against Initial Sequence Number attacks. Enhanced SoftStax TCP to be NetBSD1.6-based since it uses MD5 and SHA1 (Secure Hash Algorithm 1).
- CF16072: PKMAN update with `getstat` call for Write Ready.
 PKMAN has been enhanced with an additional `getstat` call for Write Ready (`SS_WRDY (0x99) & SS_WREADY (0x201)`). This call is very important for the MGR Graphical User Interface.
- CF16699: Add gratuitous ARP support
 Edition 63 of `spenet` has been enhanced to support gratuitous ARP. A gratuitous ARP is sent out when a new IP address is assigned if the `enet` descriptor has defined `GRATUITOUS_ARP` to be 1.
- `ifconfig`
 Stop and start network interfaces are now supported.
- `mbdump`
 The `-a` display allocation table option has been added.
- `ip0_router`
 The `spip` descriptor has been added.

- The following NFS modules have been eliminated: `dird`, `on`, `rcopy`, `rdir`, `rex`, `rexdc`, `rldd`, `rload`, `rpcgen`, `rpr`, `rsort`, `sorted`, `spray`, and `sprayd`.
- `netdb_local.1` has been eliminated.
- `recvmsg()` and `sendmsg()` socket library functions have been added. In addition, the `flags` parameter for `send()` is now supported.
- `spip`
Multiple rejecting network routes have been added at a start time.
According to RFC 1122, 127.0.0.0/8 should not leave the node 127.0.0.0
`netmask 0xff000000 127.0.0.1`.
- CF17017: All the baud rate generators for the SMCs should be supported.
SMC's can use BRG1,2,7,8 for clocking. The original sources did not list all of the clock sources. All of the BRG's for the SCC & SMCs may be need.
`unit_stat->v_brg` should reflect the number for the BRG used.

Resolved Problems

This section discusses problems that were resolved for SoftStax and LAN Communications. Where applicable, CustomerFirst (CF) and ClearQuest (RSYS) incidents are included:

- CF13932: `beam` fails with an exception.
This problem has been corrected; additional stack size has been added.
- CF14142: Local name resolution does not work without DNS servers configured.
The local name resolution now works even if DNS are not configured; this update is located in `gethnamaddr.c`.
- CF14212: The `route` display hangs the system.
When the path is closed, per-path-storage memory returns to the system. However, `os9_pp` is still in the socket that is pointing to that memory. When the packet moves to that socket, it attempts to send according to the released per-path-storage. `del_pp()` cleans up the socket's `os9_pp` and `do_updata()` does not send packet if there is no per-path-storage associated with it. These changes are reflected in edition 18 of `sptcp`, edition 16 of `spudp`, edition 11 of `sproute`, and edition 14 of `spraw`.
- CF14449: A problem in `fmstart.r` causes SPF to crash the system.
Previously, when entering the `__DeInitFM` function of the SPF, the system crashed just before the `__RmDevlist` call. This happened because the test for a valid pointer was not done. Code has now been added to check to make sure that the pointer is valid.
- CF14896: `route` information does not update after reconnecting to the existing interface.

After establishing routes with two interfaces, one interface was disconnected. Soon, the routes updated to use the other interface. However, when the disconnected interface was reconnected, the route information did not recover.

- CF15096: The NFS file manager leaks memory on aborted delete requests.
The OS-9 for 68K and OS-9 NFS file managers have been fixed to no longer leak memory on aborted delete requests. Generally, delete requests abort when the target file is open by a process on the mounted machine.
- CF15467: `rpclib` expected `select()` to return `EINTR`.
OS-9 `select()` returns (via `errno`) `EOS_SIGNAL` if a signal interrupts the call. The UNIX version of `select()` returns `EINTR`. It was determined that we could not break compatibly with existing OS-9 applications and change the behavior of `select()`. Instead, `rpclib` was enhanced to test for both `EOS_SIGNAL` and `EINTR`.
- CF16149: Multiple `sptcp4` updates.
Corrected problems with memory corruption in `TF_NODELAY` cases. Also resolved problem referencing a NULL pointer.
- CF16257: Frequent outgoing connection via TCP/IP is crashing the system.
Changed `in.h` to move the ephemeral port range from 1024-5000, to 49152-65535. Also changed the loop in `in_pcb.c` to only loop through the port range once and then exit with an `EADDRNOTAVAIL` error if no free port is found.
- CF16393: Networking stack can incorrectly respond to UDP broadcast packets.
`spenet` edition #62 or better has been corrected to maintain the broadcast/multicast status of a packet as it flows up the stack.
- CF16444: `select()` now returns `EOS_SIGNAL` when an unexpected signal arrives and no timeout was specified.
Previously, no error was returned when an unexpected signal arrived during a blocking `select()` call. Now, -1 is returned and `errno` is set to `EOS_SIGNAL`. Third-party source applications may be expecting `errno` to be `EINTR` in this case.
- CF16446/RSYS30080: Many out-of-sequence TCP packets allow a denial-of-service attack
The TCP driver (`sptcp`) was changed to only allow a specified amount (default = 24K) of data to accumulate in the TCP re-assembly queue. This prevents an unlimited accumulation of re-assembly data.
- CF16489/RSYS30106: address aggregation problem, RIPV2
More recent code from the BSD source was integrated into `routed` to correct a problem with aggregation.
- CF16532: Taking the SLIP interface down does not free all the memory for 68K
The `spslip` driver was fixed so that the kernel could correctly reclaim all the memory allocated when the `spslip` system-state threads were terminated.
- CF16616: System reboots when taking down PPP stack for 68K

When deinitializing the 68K PPP stack, a privilege violation exception (8) would occur, causing the system to reboot. The problem was found and resolved in the `sphdlc` driver.

- `spenet`
 - `deventry` in the mbuf will now determine the incoming interface if the `lu_pathdesc` is set.
 - An Ethernet header has been created to pass hardware addresses to the `spenet` packet.
 - The `ip_rtmsg_pb` parameter block is now used for calling the `IP_SS_RTMSG` `setstat`.
- CF 12697: Incomplete mbuf search leading to premature `ENOBUFS`
 The TCP/IP stack was fixed so that it no longer prematurely return `ENOBUFS` in rare circumstances related to non-blocking sockets.
- CF 13962: The `spquicc` driver does not set Mbuf flags correctly.
 In some situations, the `spquicc` driver sets the Mbuf flags field incorrectly. This problem has been fixed.
- CF 13235: TCP/IP can consume all Mbuf space.
 If an application was writing to a socket and getting responses, but failing to read the responses, it could cause the TCP/IP stack to consume all the mbuf space in an inefficient manner. The `spip` module has been fixed to more efficiently store the responses so that they no longer consume an unnecessarily large amount of space.
- CF 13249: TCP does not respond if no ACK bit is set.
 Some broken implementations of TCP do not set the ACK bit when sending TCP packets. The input routine of `sptcp` has been changed so an ACK is sent in response to packets with no data and no ACK bit set.
- CF 12580: The transfer rate is very slow when the `TCP_NODELAY` flag is used.
`sptcp` now better handles the timing consideration when transferring small amounts of data when the `TCP_NODELAY` option is set.
- CF 12944: Lan Communications memory leak.
 The `accept()` socket function leaked two lock descriptors each time it was called. This has been fixed; the lock descriptors are now freed when the socket closes.
- CF 13139: `SPF_PPSTAT_INIT` is not filled in the per-path storage structure.
 This problem was the result of inadequate documentation for the functions in `ppstat.l`. These functions are now documented.
- CF 13262: SPF is not passing an error marked in Mbuf from the driver up stack.
 The SPF file manager was returning `SUCCESS` in some situations where an error was encountered. SPF was fixed to pay more attention to the error flag on inbound Mbufs.

- CF 12612: BREAK and IP options are not handled by `telnetdc`.
The telnet daemon (`telnetdc`) did not handle the BREAK and IP telnet options. These options are now processed and a BREAK option will send a SIGQUIT signal to the child process. The IP option sends a SIGING signal to the child process.
- CF 12789: The DHCP client is confused by BOOTP responses and floods DISCOVER packets.
The `dhcp` utility is no longer confused by BOOTP responses. It has been fixed to better classify responses.
- CF 12792: The DHCP client "leaks" 32K bytes of Mbuf space.
This problem has been corrected; `dhcp` now both sleeps for the IP address lease time and watches for incoming data on the socket. Any incoming data not related to DHCP is discarded, thus freeing the mbuf space.
- CF 13422: `telnetd -t` option does not auto-logout after a specified time.
The `-t` option is ignored by `telnetd` if it is followed by an equal sign (i.e `-t=5`). This problem has been corrected.
- CF 13522: The DHCP client can override local settings with those returned by server.
The `-override` option was added to `dhcp` allowing existing network configurations, such as DNS name servers, to be overridden by `dhcp` server-supplied information.
- CF 11091: NFS server problem.
The `mount` utility was fixed to not unlink from a module twice. This caused an exception that was visible on the MIPS platform.
- CF 12629: The NFS server returns an error when a file is opened.
If an application attempts to open an existing file mounted on the OS-9 NFS server with "write" permission, it receives an `000:203` error from `fopen()`, even when file permission and UID allow it.
- CF 13450: Ethernet boot memory leak.
Previously, sockets were not being released for reuse after timeout. `llbootp` has been modified to release sockets and free memory when no server is found.
- CF17013: `connect()` can crash system (IPv4)
`spip` for the IPv4 stack has been updated to prevent a situation where a `connect` could cause a crash.
- CF17140: TCP data can be corrupted when small MTU and/or window sizes are used
`sptcp` was fixed to no longer transmit corrupted data when small MTU and/or window sizes are used.
- CF17551: The IPv4-only stack can transmit information that violates the reception window of the receiver.

`sptcp` was fixed to avoid transmits that violate the receivers window requirements. This problem was most prevalent when MTU and/or window sizes were small.

- CF16591: `get_eaddr()` for the `sp860t` driver should be in `target.c` instead of the shared code.

The implementation of `get_eaddr()` was moved from the shared code (`init.c`) to the port-specific source file `target.c`. Refer to the `8XXFADS/SPF/SP860T/target.c` for an example of its implementation.

- CF17364: The `pppd` utility does not print some information correctly.

The compressed state and slot usage in the PPP stack is now correctly reflected in the debug printouts of the `pppd` utility.

- CF17364: OS-9 for 68K needs to use the modern PPP stack

The makefiles were modified to build the newer PPP stack modules (`splcp` and `spipcp`) for OS-9 for 68K instead of relying on old functionality.

- CF17364: `mount` needs to watch out for NULL pointers
- The NFS `mount` command was modified to ensure that strings that are sometimes NULL are not passed to `strlen()`.
- When a `read()` on a socket is interrupted by a signal, the `read()` call returns -1 and set `errno` to `EOS_SIGNAL`.
- If a connection is terminated from the remote side, `_os_read()` now returns an `EOS_EOF` error.
- There is no `recvmsg()` or `sendmsg()` support on the legacy LAN Communications stack.
- `do_read()` returns errors from Mbufs correctly in `IO_PACKET`.

OS-9 Utilities Notes

The following section represents changes and updates to OS-9 utilities for this release.

Enhancements

The following list describes general enhancements made to the OS-9 utility set for this release. Where applicable, CustomerFirst (CF) and ClearQuest (RSYS) incidents are included.

- CF 12355: The `mshell` utility has been enhanced to allow the `print()` directive to send output to `stdout` instead of `stderr`. This enhancement was implemented as a command line option (`-w/-nw`).

By default, the `print()` directive will write to `stderr`. If `-w` is specified on the command line of in a script file, it will write to `stdout`. This new option is also inherited through the `_SHELLPARAMS` environment variable.

- CF 13814: The `mshell` utility should have a `%status` variable.

The `mshell` utility was enhanced to include support for a shell variable called `%status`. It contains the exit status of the most recent command executed. Refer to the chapter on `mshell` in the Utilities Reference manual for more information.

- CF15240: `fdisk` can't be run from a batch file.

The `fdisk` utility previously required the user to press 'ESC' to move between screens and exit. This did not work well if `fdisk` was run from a batch file. The `fdisk` utility now allows the user to select either 'ESC' or 'Q'. In addition, the user no longer has to press ENTER after making menu choices.

- CF15560: Enhancement request for a utility to print the current boot file.

Enhanced `bootgen #29` for OS-9 and `os9gen #33` for OS-9 for 68K. Added `-v` to print current boot file information and `-vv` to print identification sector information.

- CF15820: `mshell` suspends background output after first input character at a prompt

`mshell` was enhanced to allow background process output regardless of how many characters had been entered at the prompt. `mshell` now uses a non-blocking read for all characters read from the console (if `-nf` is in effect, all output is blocked).

- `fixmod` enhanced to optionally keep every copy of a given module

`fixmod` was enhanced to include a new `-c` option that indicates that it should avoid overwriting any previously written copy of a module found in a merged module file. This is useful when breaking a bootfile up into separate modules and the same named module, presumably with a different module revision, appears more than once in the file.

- `padrom` enhanced to have a “reverse” pad option

`padrom` was enhanced to include a new `-s` option to shorten the length of a file. This can be used to strip off padding or other file contents.

Resolved Problems

This section describes CustomerFirst (CF) and ClearQuest (RSYS) incidents related to the OS-9 utilities and how they were resolved for the current release.

- CF 9265: PCF does not allow removal of write permissions.

The case statement for setting the permissions of directories was returning `read/write` each time, regardless of what permissions were set with the `attr` command. This has been modified to return only `read` if the command `attr -nwnngwnpw DIR` is issued. The directory's attributes are correctly set from there.

- CF 12852, 13213: `os9make` under Windows gets long command lines wrong.

Under early versions of Windows/DOS, there is a command line length limit of 128 characters. In order to preserve compatibility with these old versions of Windows/DOS, if there is a command line that exceeds 128 characters, `os9make` would break up the command line and put it into a temporary file, then execute the command with the `-z` option pointing to the temporary file. Although this works with many OS-9 utilities, there are OS-9 utilities that do not support the `-z` option, and no non-OS9 utilities have the option. Because of this, the work-around to break up the command line is no longer valid.

Under newer versions of Windows, the command line length limit was extended; however, this limit varied among versions of Windows, so there was no way of discovering the limitation. This has been resolved; the `-c` and `-c1` options have been added to `os9make` under Windows. The `-c` option ignores a command line length limit, while the `-c1` option allows a specific command line length to be specified by the user.

- CF 12861: The `os9delldir.exe` file in `MWOS/DOS/BIN` does not work.

`os9delldir` was added to the cross-hosted utilities. The `-x` option is unavailable, but the `-q` option is always on.

- CF 13078: `mkdatmod -r` is not extracting information from a file.

`mkdatmod` will extract the information from a module if you specify the size of the original file using the `-s` option.

- CF 13684: Redirection is not complete in `mdir`.

Previously, not all modules were listed by `mdir`. The `_os_exit` calls were modified in `mdir` source code to exit normally; `_os_exit` does not flush the `stdio` buffers. This corrected the problem.

- CF 13764: `os9make` crashes when calling to `deeply` with sub-makes.

The array that stores the number of include files when using a for loop was allocated with storage for just 33 files. The size of the include file array for this instance has been increased substantially.

- CF 14134: `mdir -e` does not display the type `cdb` correctly.

The type for a `cdb` module formerly displayed as “5” and is now displayed as “`cdb`”

- CF 14147: `mdir` always show module directories
`mdir` was fixed to properly exclude module directories from listing that should not include module directories.
- CF 13942: `dir` command causes an exception on long filename
The `dir` utility was fixed to handle file names up to 256 characters long. The old limit was 44, which was based on RBF's limit for file names.
- CF13943: The `dir` command shows invalid entries on PCF/FAT files.
The `dir` utility now prints a warning if the checksum of the 8.3 name does not match the LFN record. This is an indication that the file system is inconsistent due to media error or is being modified by an operating system capable of supporting non-long file names.
- CF13944: The `dir` command continues the directory search after a NULL entry.
Previously, the `stdio` buffer was flushed between every entry; this caused excessive I/O occurrences. This has been corrected in edition 66; the `dir` command is now faster for PCF directory listings.
- CF14340: Utilities sometimes ignore last line of `-z` input file if no EOL.
Utilities with the `-z` options have been updated to correct this problem.
- CF14147: `mdir` shows module directories when it is only supposed to show modules of a particular type.
This problem has been fixed.
- CF14234: `format` does not handle the <Esc> key well at the prompt.
The `ynans` function has been fixed so that input of escape does not cause an endless loop.
- CF14839: `fsave` crashes on long and deep directory trees.
Previously, `fsave` was only saving 80 characters or less to the backup-records file (in this example, `backup_dates`) and then only reading 80 characters or less the next time `fsave` was run. This was causing the second iteration of `fsave` to look for a directory that did not exist.
Edition 34 of `fsave` has been updated. The size of pathnames has been increased to 800 characters for reading to the record, writing to the record, and printing to the screen.
- CF15120: the `popen()` utility does not work correctly in all situations.
The `popen()` utility in `unix.1` has been rewritten to operate more similarly to the `popen()` found in mainstream operating systems.
- CF15608: Problems using `fdisk` on large drives.

`fdisk` did not support LBA mode for large IDE drives such as 20 GB, 40 GB, or 60GB. `fdisk` edition 28 now properly reads the correct size of a hard disk greater than 8 GB and properly converts to CHS (cylinders, heads, sectors).

- CF16057: `procs` for 68k did not always show process 65
When running `procs`, the last process in the system process table (typically 65) would not always be shown. The code has been modified to display all processes.
- CF16080: `frestore` can not restore files saved by `fsave` edition 34.
When trying to do a `frestore` of files saved with `fsave` edition 34, an error "frestore: can't identify volume, ident block is corrupt." would be returned. This was due to differences in the identification block structure made by `fsave` edition 34. `frestore` has been updated to restore either style of archive.
- CF16230: `mv` command did not handle moving busy files gracefully.
When `mv` tries to move a file, it creates the destination link then tries to delete the source link. If the source file was busy (in use), the delete would fail, leaving both the source and destination link. This fix sets `FAM_NONSHARE` when opening the source entry. This will cause the open to fail if the file is in use/busy/nonshareable. That way, if you try to `mv` a busy file, it will fail immediately rather than copying/linking the file, then failing, leaving you with both the source and destination files.
- CF16332: `mshell` redirection with `-x` problems
If the `-x` option was specified for `mshell` to exit on error, and there was a redirection error, `mshell` would print out the error, but not exit. `mshell` now properly prints the errors and exits. `mshell` was also fixed to properly report when it could not redirect input to a procedure file.
- CF16337: `mshell` can't handle an `assign` with a `-?` in it
When running a script or launching a new `mshell` with an alias containing "`?`", `mshell` would print the `mshell` options. `mshell` has been fixed to properly parse options and environment variables when starting a new `mshell`.
- CF16612: `mshell` does not exit if telnet connection is killed.
If a telnet connection was not closed cleanly, `mshell` would remain on the system. `mshell` now verifies the input path is still active when waiting on input every 30 seconds. Power management users can use the `-ny` option to disable this functionality.
- CF16689: `editmod` can give errors if `#undef` is used
When `#undef` was used to remove known symbols, `editmod` would remove additional identifiers. This problem has been resolved.
- CF16797: `os9make`'s `ifexists` can be true when it should be false
`os9make` was fixed to correctly handle conditionals that don't expect boolean expressions (`ifexists`, `ifmake`, etc.)
- `fixmod` uses the `dump` path twice when dumping "junk" (non-module data)

The path passed to `-d` is no longer added twice to the pathlist for dumped “junk”.

- `os9make` translates the slash character commonly used on MS-DOS command line options

`os9make` was fixed to avoid translation of the pathlist character from forward slash (/) to backward slash (\) when the slash appears to be the start of a MS-DOS command line option.

- CF17099: `os9make` is confused by a dependency on `<reserved_word>.r` (e.g. `if.r`)

`os9make` was fixed to no longer mistake the start of file names with its own reserved words.

MAUI Notes

The following sections contain release notes for MAUI.

Enhancements

- **CF15380:** The MAUI Graphics driver `gx_c1543` does not support extended device capabilities.

Support has been added for extended device capabilities to the MAUI Graphics driver `gx_c1543`. This data is retrieved via the `gfx_get_dev_capexten()` call.

Resolved Problems

This section gives a description of CustomerFirst (CF) and ClearQuest (RSYS) incidents related to MAUI and how they were resolved for the current release.

- **CF 14560:** `jview` gets Non-Fatal MAUI error

A MAUI non-fatal error would occur every time the `jview` demo was run. This was caused by a call to an unnecessary call to `gfx_term`. The call to the `gfx_term` function has been removed and the error no longer occurs.

- **CF 14581:** `winink` MAUI demo problems.

The `winink` demo has problems coinciding its window area with its drawing area. This problem manifests itself in various ways such as being able to ink onto the window manager or other windows.

- **CF16357:** JAVA SWING text displayed with a white background with an MAUI IOBLT driver.

JAVA composes its display off screen and then copies it on screen. There was a bug in `$MWOS/SRC/DPIO/MFM/DRVR/GX_COMM/gdv_blt.c` (MAUI graphics driver IOBLT common code) that caused the MAUI high level BLT code's mixing mode to change from `BLT_MIX_RWT` or `BLT_MIX_RWM` to `BLT_MIX_REPLACE`, even if no graphics RAM was involved in the operation. This fix only applies to graphics drivers that define `GDV_INCLUDE_IOBLT` and BLTs from non-graphics memory to non-graphics memory.

- **CF17364:** MAUI graphics drivers terminate a semaphore twice and uses it before the second `_os_sema_term()`.

The common source code for all MAUI graphics drivers was modified to not terminate the semaphore until it is done being used.

Add-Ons

This section describes the enhancements, resolved problems, and known issues for the add-on products in this release. CustomerFirst (CF) and ClearQuest (RSYS) incidents and workarounds are provided, where applicable.

Enhancements

- CF16238: Remove 16 MB barrier in RBFTL.
RBFTL was updated to be able to access drives larger than 16 MB. This enhancement will break backwards compatibility. RBFTL has also been modified to store files in their native endianness.