Home

Microware OS-9®
Release Notes

Version 4.5

RadiSys
THE POWER OF WE

www.radisys.com
Revision A • September 2005

Copyright and publication information

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.

Contents

Chapter 1: Introduction

Chapter 2: Operating System

Chapter 3: Host Applications

Chapter 4: Components

1 Introduction

Microware OS-9 version 4.5 represents a maintenance and update release to incorporate all of the improvements that have been introduced into the component parts. This release includes support for PowerPC processors.

2 Operating System

This chapter provides an overview of the changes and improvements made to OS-9 for version 4.5.

Enhancements

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

- CF17102: Support for the MPC5200 was added to OS-9.

  OS-9 now support the Motorola MPC5200 processor. Both the TQ Systems TQM5200 and MicroSys PM520 boards are supported.

- CF17324: Event creation is slow when many events exist

  The kernel was fixed to more quickly create events when many exist. Events are now created up to 125 times faster than previous kernels.

- CF17324: Opening a path spends too much time in the kernel and ioman.

  `kernel` and `ioman` were modified to reduce the overhead of opening a path to an already attached device. The `kernel`/`ioman` overhead was cut in half. As a side-effect of this fix, `_os_unlink()` is several times faster than previous kernels.

- CF17354: OS-9 for PowerPC should have reduced power idle handling

  The operating system was enhanced to include sample extension modules that can put certain PowerPC processors (603, 603e, and e500 cores) into "doze" or "nap" mode. This extension module is in the PORTS directories for appropriate processors. It can be utilized by adding the "idle" module to an extension list (see the `init` module dialog in the configuration wizard) and adding the `idle_nap` or `idle_doze` file (both files have the module name `idle`) to the bootfile via `user.ml`.

- OS-9 should support `dup2()`

  `ioman` and the Ultra C libraries (`os_lib.l`) were modified to add support for a new system call: `_os_dup2()`. Refer to the Ultra C Library Reference or OS-9 Technical Manual for more information about this call. It is utilized by the `dup2()` Unix compatibility library.

- CF16585: The Windows file system manager (`pcf`) creates many, similarly named files very slowly.

  `pcf` was enhanced to more efficiently create MS-DOS aliases for similarly named long file name files. In one case, the enhanced `pcf` created the 200th such file 35 times faster than the prior `pcf`.

- CF16946: RBF should be enhanced to allow a single trailing slash (/) when opening directories.

  The RBF file manager (`rbf`) was enhanced to allow a single trailing slash when opening directories. `chdir()` is considered a directory open. Thus, given a RAMdisk (`/r0`) with a directory `X/Y/Z`, the following sequence of commands is now allowed:

  ```
  $ chd /r0/X/
  $ chd Y/./
  $ chd Z/../Z/
  $ chd ../...../
  ```

```
$ pd
/r0
```

Resolved Problems

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

- CF17387: The FPU module for MIPS64 doesn't handle unimplemented operation exceptions

  The FPU module (`fpu64`) for MIPS64 was modified to include support for handling unimplemented operation exceptions on MIPS64. This allows it to handle operations on denormalized numbers and NaNs.

- CF17390: The conversion of some 32-bit floating-point values to integer fails on MIPS3000

  The FPU module for MIPS3000 processors was fixed to correctly convert all 32-bit floating-point values to integers.

- CF17396: The MVME5500 and MVME6100 do not have their level 2 nor level 3 caches enabled.

  The L2 and L3 caches are now turned on by `initext` code for MVME5500 and MVME6100.

- 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.

- CF17090: SSM and cache modules for PowerPC do not behave correctly when a user cache mode region ends at address 0xffffffff.

  The SSM and cache modules for PowerPC were updated to correctly handle a user cache mode region that ends at address 0xffffffff.

- CF17125: non-68K `pipeman` returns #203 instead of #216

  When doing a `dir` on a non-existent pipe, `pipeman` was doing a mode check first and returning #203 `EOS_BMODE` rather than the expected #216 `EOS_PNNF`. `pipeman` has been fixed to check for pipe existence first, before proceeding with mode checks.

- 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.

- CF17404: Corrupt modules with 0-length in header would cause a boot loop.

  Low-level boot code would check for module sync code, then, if found, use the module length field to know how for to skip for the next module. If a corrupt module header was found with a 0 length, the boot code would loop forever. A fix was made so 0-length modules would be ignored.

- CF17408: Killing processes with multiple threads can lead to system corruption

  The kernel was fixed to not corrupt the system memory list when releasing the memory resources related to a process and its threads when it's being killed. A new process state bit was created for this purpose and is reflected in a new `procs` utility.

- PC file manager doesn't respect `CTRL_NOWRITE`

  The PC file manager (`pcf`) for OS-9 was fixed to respect the `CTRL_NOWRITE` flag which indicates that file descriptor last-access time should not be updated on every file access.

- RBF can deadlock if multiple paths are extending the same file

  RBF was fixed to not deadlock if multiple paths are expanding the same file.

- RBF needs to work better in power loss situations

  RBF was fixed to write directory entries and file descriptors in a better way in the face of power loss. In addition, support for the AC fail (`SS_ACFAIL`) setstat was added to prevent any writing after the loss of power is recognized.

- CF16783: SCF drivers can send unsolicited `SIGWAKE` to processes.

  The SCF file manager (`scf`) was fixed to clear any pending `SIGWAKE` information out of the logical unit static storage, regardless of how the prior I/O operation terminated.

- CF14103: setstat `SS_LUOPT` is also writing the `v_err` flag

  SCF was fixed to not copy the `v_err` flag specified by the user in an `_os_ss_luopt()` call. Copying `v_err` could result in unmotivated errors being returned by SCF on subsequent I/O operations.

- CF17105: Sometimes, pressing `^C` or `^E` does not kill the last process to do I/O on the terminal

  `ioman` was enhanced to send additional information (a parameter block for `SS_RELEASE`) to file managers when threads exit. This information lets SCF know that only a thread is shutting down, not necessarily the process containing the thread.

- CF17437: The floating-point context can be corrupted by signal handling.

  The kernel was fixed to properly maintain FPU contexts when a process enters its timeslice with a signal pending. This was only a problem for hardware floating-point processors.

3 Host Applications

This chapter contains release notes for host applications used with OS-9 v4.5.

Configuration Wizard Notes

The following sections represent changes and updates for the Configuration Wizard.

Enhancements

The following changes were made to the Wizard to increase the functionality:

- CF16986: The Wizard should allow the user to specify the block size for the `mbuf` pool.

  The Configuration Wizard was enhanced to have a drop-down selection box for choosing the block size of allocations from the `mbuf` pool. The default remains 64 bytes, but the user can choose either 32, 64, 128, 256, 512, or 1024 bytes.

- CF16986: The Wizard should allow the `spfndpd` and/or `spfnppd` daemons to be started automatically.

  The Configuration Wizard was enhanced to have check-boxes for starting the user-state debugging daemon (`spfndpd`) and/or the system profiler (`spfnppd`) daemon. These check-boxes appear on the SPF options pane. A new `.ml` key (`SPFNPPD`) is available to conditionalize `bootfile.ml`. It is `TRUE` when the "Start spfnppd" box is checked and high-level networking is enabled.

- CF17354: The Wizard should have a way to specify additional extension modules for the `init` module

  The Configuration Wizard was enhanced to allow additional extension modules to be specified. On the "Init Module" dialog there are two additional edit fields: Pre-I/O and Post-I/O. Any number of extension modules can be specified here. Module names that are prefixed with a plus ('+') or have no prefix are added to the end of the pre or post I/O extension module list. Module names that are prefixed with a minus ('-') are added to the beginning of the pre or post I/O extension module list.

Resolved Problems

This section provides a list of CustomerFirst incidents related to the Configuration Wizard and how they were resolved for the current release.

- CF16986: The Wizard should allow more than 80 characters for advanced interface configuration strings.

  The Configuration Wizard was enhanced to allow 2048 characters for advanced interface configuration strings. Note: The `init` module parameter string is limited to about 1000 characters and the advanced interface configuration strings are part of the `init` module parameter string.

- CF16953: The Configuration Wizard has an arbitrary 15-character limit on a configuration name and fails to check for this limit when the configuration name is changed.

  The Configuration Wizard (`os9p.exe`) was enhanced to more clearly describe the limit and the limit was raised considerably. The "Change Configuration

Name" menu item was changed to "Save Configuration As..." for more clarity and error checking was added to ensure the higher configuration name length limit was not exceeded.

Hawk Notes

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

Enhancements

- 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 `<del>` key or the right-click menu can be used to remove Watch expressions from the window.

- 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.

- 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

Resolved Problems

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

- 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.

- CF 17345: 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.

- CF17053: Hawk cannot debug on a Pentium 4 target

  The Hawk debuggers (`mwsrcdbg.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.

4 Components

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

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 incidents are included.

- 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 csl and mt_csl to aid in debugging

  The shipping images of OS-9 have been modified to include the symbol files (.stb) for csl and mt_csl. 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

- 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 0x20000000000000 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.l 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.l 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 incidents are included.

- 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.

- CF17108: sppro100's buffer handling has several efficiency issues.

  The buffer handling for the Intel Ethernet Pro 10/100 PCI board was rewritten to improve efficiency.

Resolved Problems

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

- 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: DEC 21140 driver needs to work for all MOTRAVEN MVME boards

  The DEC 21140 driver (`sp21140`) was updated to require the user to specify the speed and duplex of the Ethernet connection in the Wizard. This allows it to correctly identify the network for PHYs that cannot correctly identify the network.

- 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: The 91c94 Ethernet driver doesn't correctly get the MAC address from the hardware

  The 91c94 Ethernet driver (`sp91c94`) was fixed to correctly get the MAC address from the hardware.

- 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()`.

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 incidents are included.

- `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.

- `dcheck` enhanced to better check directory entries

  `dcheck` was modified to do more checks on directory entries before assuming that the FD sector is correct. The user will be prompted to correct any of the new problems encountered.

Resolved Problems

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

- 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 reservered words.

MAUI Notes

The following sections contain release notes for MAUI.

Resolved Problems

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

- CF16932: MAUI usage from threads can result in `EOS_MAUI_NOTOWNER` errors

  The MAUI common driver source was fixed to record the correct process ID in the data structures such that any thread within a process will be considered the object's owner.

- 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 incidents and workarounds are provided, where applicable.

Resolved Problems

- CF14399: HawkEye should warn you if you leave the trigger dialog without adding your trigger.

  HawkEye was changed to pop-up a message box asking if you really want to leave the trigger composition dialog box if it appears that you have composed a trigger and have not clicked `Add` to add it to the trigger list.

- CF16674: `slmlib.l` should be linkable to threaded applications.

  The `slmlib.l` library was changed to indicate that it is inherently thread-safe. Thus, the library can be linked safely to both threaded and non-threaded applications.

- CF17356: HawkEye user events are not handled very well

  HawkEye was fixed to properly load the supporting values used with `hawk_control_log2()` and to correctly display and filter for user event IDs greater than 255.

- HawkEye doesn’t show the name of events when they are deleted.

  HawkEye was fixed to correctly show the name of the event being deleted by a `F_EVENT/EV_DELETE` system call.