

# MapuSoft Technologies 1.3.5 Release Notes

**Release 1.3.5**  
**May, 2009**  
**Revision 1**

These release notes accompany Release 1.3.5 of MapuSoft Technologies. They briefly describe new hardware and software features and provide a summary of the current software limitations and known defects, if any, that exist in this release.

**Contents**

**Updated Features of MapuSoft Products..... 3**

**OS Abtractor ..... 4**

    OS Abtractor POSIX ..... 4

    micro-ITRON OS Abtractor ..... 7

**OS Changer..... 11**

**OS PAL ..... 11**

**Known Limitations ..... 12**

**Host and Target Feature OS Support..... 12**

**OS PAL Release 1.3.5 Host System Requirements ..... 13**

**Technical Support ..... 13**

**Revision History ..... 14**

## Updated Features of MapuSoft Products

Release 1.3.5, build 7667 of MapuSoft Technologies encompasses all upgrades to MapuSoft's OS Abstractor, OS Changer, and OS PAL Solutions. These package components are as follows:

No	Product Name	Version
1	Demo_osabstractor	3.5.3.6180
2	Demo_osabstractor_posix	3.5.3.6181
3	Demo_oschanger_nucleus	3.5.3.6182
4	Demo_oschanger_psos	3.5.3.6183
5	Demo_oschanger_vxworks	3.5.3.6184
6	Include	3.5.3.6185
7	osabstractor_linux	3.5.3.6355
8	Osabstractor_lynxos	3.5.3.6335
9	osabstractor_mqx	3.5.3.6336
10	osabstractor_nucleus	3.5.3.6403
11	osabstractor_posix	3.5.3.6404
12	osabstractor_qnx	3.5.3.6339
13	osabstractor_solaris	3.5.3.6340
14	osabstractor_threadx	3.5.3.6341
15	osabstractor_uitron	3.5.3.6342
	osabstractor_vxworks	3.5.3.6343
16	osabstractor_windows	3.5.3.6344
17	oschanger_nucleus	3.5.3.6276
18	oschanger_psos	3.5.3.6254
19	oschanger_vxworks	3.5.3.6252
20	OS_PAL	1.3.5.7667

## OS Abstractor

**New APIs**— The following new APIs have been added:

- OS\_Process\_Unregister\_Exit\_Hook – Removes a process exit hook which was previously registered using OS\_Process\_Register\_Exit\_Hook

**Modified APIs** — The following APIs have been modified:

- OS\_Get\_Pipe\_Message\_Count – New parameter added which will be populated with the pipe count. Function now returns a status.
- OS\_Get\_Queue\_Message\_Count – New parameter added which will be populated with the queue count. Function now returns a status.
- OS\_Process\_Register\_Exit\_Hook – hook function now takes a parameter. A new parameter was added which is the value that will be passed to the hook function when it is invoked.

## OS Abstractor POSIX

OS Abstractor POSIX has added the following APIs for the release 1.3.5.

**New APIs**— The following new APIs have been added:

- \_exit – This function terminates the calling process.
- abort – This function causes abnormal process termination to occur, unless the signal SIGABRT is being caught and the signal handler does not return.
- alarm – This function causes the system to generate a SIGALRM signal for the process after the number of realtime seconds specified by seconds has elapsed.
- atexit – This function registers a function, to be called without arguments at normal program termination.
- confstr – This function return configuration-defined string values.
- execl – This function replaces the current process image with a new process image.
- execlp – This function replaces the current process image with a new process image.
- execlp – This function replaces the current process image with a new process image.
- execvp – This function replaces the current process image with a new process image.
- execve – This function replaces the current process image with a new process image.

## MapuSoft Technologies 1.3.5 Release Notes

- `execvp` – This function replaces the current process image with a new process image.
- `exit` – This function terminates the calling process.
- `fork` – This function creates a new process.
- `getpgrp` – This function returns the process group ID of the calling process.
- `getppid` – This function returns the parent process ID of the calling process.
- `mmap` – This function establishes a mapping between a process' address space and a file, shared memory object, or typed memory object.
- `mprotect` – This function sets protection of memory mapping.
- `msync` – This function synchronizes memory with physical storage.
- `munmap` – This function removes any mappings for those entire pages containing any part of the address space of the process starting at `addr` and continuing for `len` bytes.
- `pause` – This function suspends the calling thread until delivery of a signal whose action is either to execute a signal-catching function or to terminate the process.
- `pipe` – This function creates an inter-process channel.
- `posix_spawn` – This function creates a new process. (Child process) from the specified process image.
- `posix_spawn_file_actions_addclose` – This function adds or deletes a close or open action to a spawn file actions object.
- `posix_spawn_file_actions_addopen` – This function adds or deletes a close or open action to a spawn file actions object.
- `posix_spawn_file_actions_addup2` – This function adds a `dup2()` action to the object referenced by `file_actions` that causes the file descriptor `fdes` to be duplicated as `newfdes` when a new process is spawned using this file actions object.
- `posix_spawn_file_actions_destroy` – This function destroys the object referenced by `file_actions`; the object becomes, in effect, uninitialized.
- `posix_spawn_file_actions_init` – This function destroys the object referenced by `file_actions`; the object becomes, in effect, uninitialized.
- `posix_spawnattr_destroy` – This function destroys a spawn attributes object.
- `posix_spawnattr_getflags` – This function obtains the value of the spawn-flags attribute from the attributes object referenced by `attr`.
- `posix_spawnattr_getschedpolicy` – This function obtains the value of the spawn-schedpolicy attribute from the attributes object referenced by `attr`.
- `posix_spawnattr_getsigdefault` – This function obtains the value of the spawn-sigdefault attribute from the attributes object referenced by `attr`.
- `posix_spawnattr_init` – This function destroys a spawn attributes object.
- `posix_spawnattr_setflags` – This function sets the spawn-flags attribute in an initialized attributes object referenced by `attr`.

## MapuSoft Technologies 1.3.5 Release Notes

- `posix_spawnattr_setpgroup` – This function sets the `spawn-pgroup` attribute in an initialized attributes object referenced by `attr`.
- `posix_spawnattr_setschedparam` – This function sets the `spawn-schedparam` attribute in an initialized attributes object referenced by `attr`.
- `posix_spawnattr_setschedpolicy` – This function sets the `spawn-schedpolicy` attribute in an initialized attributes object referenced by `attr`.
- `posix_spawnattr_setsigdefault` – This function sets the `spawn-sigdefault` attribute in an initialized attributes object referenced by `attr`.
- `posix_spawnattr_setsigmask` – This function sets the `spawn-sigmask` attribute in an initialized attributes object referenced by `attr`.
- `posix_spawnnp` – This function creates a new process (child process) from the specified process image.
- `pthread_atfork` – This function declare fork handlers to be called before and after `fork()`, in the context of the thread that called `fork()`.
- `pthread_getconcurrency` – This function gets and sets the level of concurrency.
- `pthread_getcpuclockid` – This function accesses a thread CPU-time clock.
- `pthread_setconcurrency` – This function allows an application to inform the threads implementation of its desired concurrency level.
- `pthread_setschedparam` – This function gets and sets the scheduling policy and parameters of individual threads within a multi-threaded process to be retrieved and set.
- `pthread_setschedprio`–This function sets the scheduling priority of the thread.
- `raise` – This function sends a signal to the executing process.
- `regcomp` – This function compiles the regular expression contained in the string pointed to by the pattern argument and place the results in the structure pointed to by `preg`.
- `regerror` – This function compile the regular expression contained in the string pointed to by the pattern argument and place the results in the structure pointed to by `preg`.
- `regexexec` – This function compile the regular expression contained in the string pointed to by the pattern argument and place the results in the structure pointed to by `preg`.
- `regfree` – This function compile the regular expression contained in the string pointed to by the pattern argument and place the results in the structure pointed to by `preg`.
- `sched_setscheduler` – This function sets scheduling policy and parameters (real time).
- `sem_timedwait` – This function locks the semaphore referenced by `sem` as in the `sem_wait()` function.
- `setsid` – This function creates session and set process group ID.

- shm\_open – This function opens a shared memory object (real time).
- shm\_unlink – This function removes a shared memory object (real time).
- signal – This function handles signal management.
- sigpending – This function examines pending signals.
- sigprocmask – This function examines and changes blocked signals.
- sigqueue – This function queues a signal to a process.
- sigsuspend–This function waits for a signal.
- sigtimedwait – This function waits for queued signals (real time).
- sigwait – This function waits for queued signals.
- sigwaitinfo – This function waits for queued signals (real time).
- sysconf – This function gets configurable system variables.
- times – This function gets process and waited-for child process times.
- uname – This function gets the name of the current system.
- unsetenv – This function removes an environment variable.
- wait – This function waits for a child process to stop or terminate.
- waitpid – This function waits for a child process to stop or terminate.

**Modified APIs** — modified the following APIs.

- None

### **micro-ITRON OS Abstractor**

micro-ITRON OS Abstractor has added the following APIs for the release 1.3.5.

**New APIs**—The following new APIs have been added:

- cre\_tsk– This service call creates a task with an ID number specified by tskid.
- acre\_tsk–This service call creates a task and assigns a task ID from the pool of unassigned task IDs.
- del\_tsk–This service call deletes the task specified by tskid.
- act\_tsk–This service call activates the task specified by tskid.
- exd\_tsk–This service call terminates and deletes the invoking task.
- ter\_tsk– This service call terminates the task specified by tskid.
- chg\_pri–This service call changes the base priority of the task specified by tskid to the priority value specified by tskpri.
- get\_pri–This service call returns the current priority of the task specified by tskid through tskpri.
- slp\_tsk–This service call moves the invoking task to the sleeping state.

## MapuSoft Technologies 1.3.5 Release Notes

- `tslp_tsk`–This service call has the same functionality as `slp_tsk` with an additional timeout feature.
- `wup_tsk`–This service call wakes up the task specified by `tskid` from sleeping.
- `can_wup`–This service call cancels all queued wakeup requests for the task specified by `tskid` and returns the cancelled request count for the task.
- `sus_tsk`–This service call suspends the task specified by `tskid`.
- `rsm_tsk`–This service call releases the task specified by `tskid` from the `SUSPENDED` state and allows the task to continue its normal processing.
- `frsm_tsk`–This service call releases the task specified by `tskid` from the `SUSPENDED` state and allows the task to continue its normal processing.
- `dly_tsk`–This service call delays the execution of the invoking task for the amount of time specified in `dlytim`.
- `cre_sem`–This service call creates a semaphore with an ID number specified by `semid` based on the information contained in the packet pointed to by `pk_csem`.
- `acre_sem`–This service call creates a semaphore with an ID number specified by `semid` based on the information contained in the packet pointed to by `pk_csem`.
- `del_sem`–This service call deletes the semaphore specified by `semid`.
- `sig_sem`–This service call releases one resource to the semaphore specified by `semid`.
- `wai_sem`–This service call acquires one resource from the semaphore specified by `semid`.
- `pol_sem`–This service call acquires one resource from the semaphore specified by `semid`.
- `twai_sem`–This service call acquires one resource from the semaphore specified by `semid`.
- `cre_flg`–This service call creates an eventflag with an ID number specified by `flgid` based on the information contained in the packet pointed to by `pk_cflg`.
- `acre_flg`–This service call creates an eventflag with an ID number specified by `flgid` based on the information contained in the packet pointed to by `pk_cflg`.
- `del_flg`–This service call deletes the eventflag specified by `flgid`.
- `set_flg`–This service call sets the bits specified by `setptn` in the eventflag specified by `flgid`.
- `clr_flg`–This service call clears the bits in the eventflag specified by `flgid` that correspond to 0 bit in `clrptn`.
- `wai_flg`–This service call causes invoking task to wait until the eventflag specified by `flgid` satisfies the release condition.
- `pol_flg`–This service call causes invoking task to wait until the eventflag specified by `flgid` satisfies the release condition.
- `twai_flg`–This service call causes invoking task to wait until the eventflag specified by `flgid` satisfies the release condition.



## MapuSoft Technologies 1.3.5 Release Notes

- `cre_dtq`–This service call creates a data queue with an ID number specified by `dtqid` based on the information contained in the packet pointed to by `pk_cdtq`.
- `acre_dtq`–This service call a data queue ID from the pool of unassigned data queue IDs and returns the assigned data queue ID.
- `del_dtq`–This service call deletes the data queue specified by `dtqid`.
- `snd_dtq`–This service call sends the data element specified by `data` to the data queue specified by `dtqid`.
- `psnd_dtq`–This service call sends the data element specified by `data` to the data queue specified by `dtqid`.
- `tsnd_dtq`–This service call sends the data element specified by `data` to the data queue specified by `dtqid`.
- `fsnd_dtq`–This service call forcibly sends the data element specified by `data` to the data queue specified by `dtqid`.
- `rcv_dtq`–This service call a data element from the data queue specified by `dtqid` and returns the data element through `data`.
- `prcv_dtq`–This service call receives a data element from the data queue specified by `dtqid` and returns the data element through `data`. `prcv_dtq` is a polling service call with the same functionality as `rcv_dtq`.
- `trcv_dtq`–This service call receives a data element from the data queue specified by `dtqid` and returns the data element through `data`.
- `cre_mtx`–This service call creates a mutex with an ID number specified by `mtxid` based on the information contained in the packet pointed to by `pk_cmtx`.
- `acre_mtx`–This service call assigns a mutex ID from the pool of unassigned mutex IDs and returns the assigned mutex ID.
- `del_mtx`–This service call deletes the mutex specified by `mtxid`.
- `loc_mtx`–This service call locks the mutex specified by `mtxid`.
- `ploc_mtx`–This service call a polling service call with the same functionality as `loc_mtx`.
- `tloc_mtx`–This service call has the same functionality as `loc_mtx` with an additional timeout feature.
- `unl_mtx`–This service call unlocks the mutex specified by `mtxid`.
- `cre_mbf`–This service call creates a message buffer with an ID number specified by `mbfid` based on the information contained in the packet pointed to by `pk_cmbf`.
- `acre_mbf`–This service call assigns a message buffer ID from the pool of unassigned message buffer IDs and returns the assigned message buffer ID.
- `del_mbf`–This service call deletes the message buffer specified by `mbfid`.
- `snd_mbf`–This service call sends a message to the message buffer specified by `mbfid`.

## MapuSoft Technologies 1.3.5 Release Notes

- `psnd_mbf`–This service call is a polling service call with the same functionality as `snd_mbf`.
- `tsnd_mbf`–This service call has the same functionality as `snd_mbf` with an additional timeout feature.
- `rcv_mbf`–This service call receives a message from the message buffer specified by `mbfid` and stores it in the memory area starting from the address specified by `msg`.
- `prcv_mbf`–This service call a polling service call with the same functionality as `rcv_mbf`.
- `trcv_mbf`– This service call has the same functionality as `rcv_mbf` with an additional timeout feature.
- `cre_mpf`–This service call creates a fixed-sized memory pool with an ID number specified by `mpfid` based on the information contained in the packet pointed to by `pk_cmpf`.
- `acre_mpf`–This service call assigns a fixed-sized memory pool ID from the pool of unassigned fixed-sized memory pool IDs and returns the assigned fixed-sized memory pool ID.
- `del_mpf`–This service call deletes the fixed-sized memory pool specified by `mpfid`.
- `get_mpf`–This service call acquires a memory block from the fixed-sized memory pool specified by `mpfid`.
- `pget_mpf`–This service call is a polling service call with the same functionality as `get_mpf`.
- `tget_mpf`–This service call has the same functionality as `get_mpf` with an additional timeout feature.
- `rel_mpf`–This service call releases the memory block starting from the address specified by `blk` to the variable-sized memory pool specified by `mplid`.
- `cre_mpl`–This service call creates a variable-sized memory pool with an ID number specified by `mplid` based on the information contained in the packet pointed to by `pk_cmpl`.
- `acre_mpl`–This service call assigns a variable-sized memory pool ID from the pool of unassigned variable-sized memory pool IDs and returns the assigned variable-sized memory pool ID.
- `del_mpl`–This service call deletes the variable-sized memory pool specified by `mplid`.
- `get_mpl`–This service call acquires a memory block whose size is specified by `blksz` from the variable-sized memory pool specified by `mplid`.
- `pget_mpl`–This service call is a polling service call with the same functionality as `get_mpl`.
- `tget_mpl`–This service call has the same functionality as `get_mpl` with an additional timeout feature.

- `rel_mpl`—This service call releases the memory block starting from the address specified by `blk` to the variable-sized memory pool specified by `mplid`.
- `get_tim`—This service call returns the current system time through `system`.
- `set_tim`—This service call sets the system time to the value specified by `system`.
- `cre_cyc`—This service call creates a cyclic handler with an ID number specified by `cycid` based on the information contained in the packet pointed to by `pk_ccyc`.
- `acre_cyc`—This service call assigns a cyclic handler ID from the pool of unassigned cyclic handler IDs and returns the assigned cyclic handler ID.
- `del_cyc`—This service call deletes the cyclic handler specified by `cycid`.
- `sta_cyc`—This service call moves the cyclic handler specified by `cycid` to an operational state.
- `stp_cyc`—This service call moves the cyclic handler specified by `cycid` to a non-operational state.
- `cre_alm`—This service call creates an alarm handler with an ID number specified by `almid` based on the information contained in the packet pointed to by `pk_calm`.
- `acre_alm`—This service call assigns an alarm handler ID from the pool of unassigned alarm handler IDs and returns the assigned alarm handler ID.
- `del_alm`—This service call deletes the alarm handler specified by `almid`.
- `sta_alm`—This service call sets the activation time of the alarm handler specified by `almid`.
- `stp_alm`—This service call releases the activation time of the alarm handler specified by `almid` and moves the alarm handler to a non-operational state.
- `get_tid`—This service call references the ID number of the task in the RUNNING state and return the task ID through `tskid`.

## OS Changer

No new features are added to the OS Changer product for release 1.3.5 apart from adding fixes to known bugs.

## OS PAL

- **Time Units in OS PAL Profiler**—Profiler Y-axis time units is added (Time can be captured in nano seconds, micro seconds, milli seconds or seconds).
- **Adding new templates to OS PAL C project**—OS PAL now provides the users the ability to create and add new templates to OS PAL C project.
- **Importing WRS projects/Legacy code**—OS PAL now provides the users to import Wind River Workbench projects to OS PAL.

## Known Limitations

- Profiler Feature is not supported in Nucleus and ThreadX targets in this release.
- Task pooling feature is not supported in Nucleus and ThreadX targets in this release.
- Creating API Profiling functions while creating a C Project, we do not support overloaded functions.
- Self deletion of POSIX thread is not supported in Nucleus target.
- LynxOS 5.0 and RT Linux are not yet validated in this release.
- Application for vxWorks 6.7 should avoid defining XOPEN\_SOURCE to 600.

## Host and Target Feature OS Support

Target OS	OS Abstractor®		OS Changer®			OS PAL®		
	BASE	POSIX	VxWorks®	pSOS®	Nucleus®	Host	Target	Profiler
VxWorks® 6x	√	√		√	√		√	√
VxWorks® 5x	√	√		√	√		√	√
Linux® 2.4	√	√	√	√	√		√	√
Linux® 2.6	√	√	√	√	√	√	√	√
LynxOS®	√	√	√	√	√		√	√
LynxOS-SE®	√	√	√	√	√		√	√
Solaris®	√	√	√	√	√		√	√
Unix®	√	√	√	√	√		√	√
eCOS®	√	√	√	√	√			
Windows® XP/Vista	√	√	√	√	√	√	√	√
WindowsCE®	√	√	√	√	√		√	√
Nucleus®	√	√	√	√			√	
ThreadX®	√	√	√	√	√		√	
MQX®	√	√	√	√	√		√	
QNX®	√	√	√	√	√		√	√
T-Kernel®	√	√	√	√	√			
micro-ITRON®	√	√	√	√	√		√	

## OS PAL Release 1.3.5 Host System Requirements

No	Supported Host Platforms	System Requirements
1	Windows XP	Minimum 128 MB RAM
2	Windows Vista	Minimum 128 MB RAM
4	Linux 2.6	Minimum 128 MB RAM

### Technical Support

Technical support is available through the MapuSoft Technologies Support Centre. If you are a customer with an active MapuSoft support contract, or covered under warranty, and need post sales technical support, you can access our tools and resources online or open a ticket at <https://www.mapusoft.com/support>.

## Revision History

May 2009–Revision 1, Release 1.3.5 of MapuSoft Technologies.

---

© Copyright 2009 MapuSoft Technologies, Inc. - All Rights Reserved

MapuSoft retains all copyrights and other property rights in all text, graphic images, and software owned by MapuSoft and hereby authorizes you to electronically copy documents published herein solely for the purpose of reviewing the information.

You may not alter any files in this web site for rebroadcast, or print the information contained therein, without prior written permission from MapuSoft.

MapuSoft assumes no responsibility for errors or omissions in this publication or other documents which are referenced by or linked to this publication. This publication could include technical or other inaccuracies, and not all products or services referenced herein are available in all areas. MapuSoft assumes no responsibility to you or any third party for the consequences of an error or omissions. The information on this web site, is periodically updated and may change without notice.