

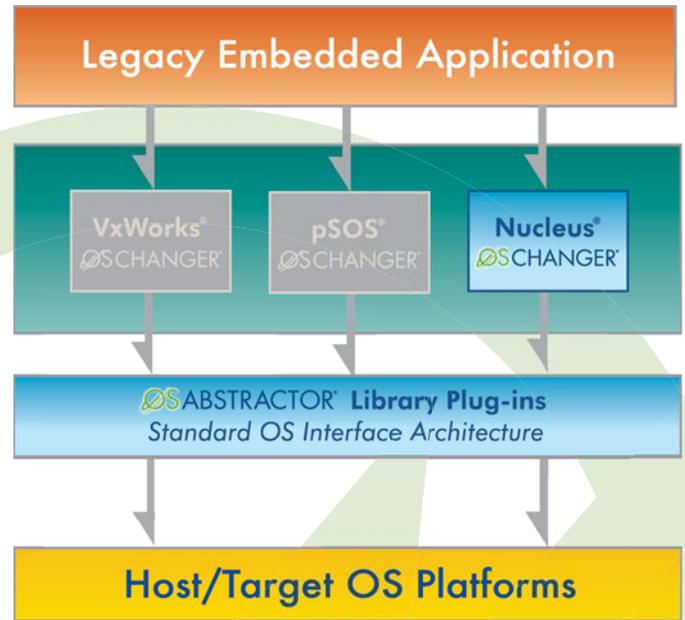
## Change Your OS - Keep Your Code

The OS Changer family of products gives developers the freedom to switch operating systems while leveraging on their existing code and knowledge base to protect their software investment. Nucleus Plus OS Changer gives developers the ability to reuse Nucleus Plus applications on different operating systems. A Nucleus NET OS Changer is also available to reuse Nucleus NET applications on different operating systems.



## Nucleus OS Changer Highlights

- Protect your software investment by re-using your Nucleus code
- Reduce time to market by migrating Nucleus code to a standard OS interface architecture
- Protect your knowledge-base by using familiar APIs and eliminate the learning curve on the new OS platform
- Eliminate dependency on a single OS vendor and switch to
  - An OS that meets your performance and memory footprint needs
  - An OS that offers better tools, middleware/drivers and support
  - An OS that supports your next generation silicon
- Reduce ongoing development and maintenance cost
  - Develop target specific code on a host platform
  - Re-use one set of code across multiple host & target OS platforms
  - Break down Nucleus applications into manageable pieces to reduce complexity and add module protection
  - Use same APIs for inter-task and inter-process communications
- OS Changer is highly optimized for each specific OS platform
- Eclipse-based host environment is available to port Nucleus applications using OS Changer in OS PAL (refer to the OS PAL datasheet)
- OS Changer includes access to the BASE OS Abstractor API features to allow development of highly portable applications (refer to the OS Abstractor datasheet)
- Additionally, POSIX or open source Linux code can be reused on a new OS platform with POSIX OS Abstractor (refer to the POSIX OS Abstractor datasheet)
- Nucleus OS Changer is offered royalty-free with source code



## Using Nucleus OS Changer

OS Changer is designed for use as a C library. Services used inside your application software are extracted from the OS Abstractor libraries and are combined with the other application objects to produce the complete image. The following are the porting steps:

- Indicate the target OS and other configurations in the usr.h file provided by OS Changer
- Include the OS Changer header files instead of the Nucleus header files in your source code
- Configure the BSP code provided by the target OS vendor to make sure the OS will run properly on your target platform
- Compile and link your application with the OS Changer libraries and header files. Refer to the standard demo application for additional information about the project settings.
- Resolve all compiler and linker errors if any
- Port low-level drivers to target OS (or use a native one if available)
- Download the application image to the target system and let it run
- Please refer to the programming manual to fix any run time errors

## Nucleus OS Changer API Support

The table below lists MapuSoft's Nucleus API support. MapuSoft also offers NET API compliance. Nucleus OS Changer is currently supported on the following target operating systems: VxWorks® 5x/6x, Windows® CE/Mobile/XP/Vista, Linux®/RT Linux®, LynxOS®/LynxOS-SE®, MQX®, Solaris®, Unix®, µITRON®, ThreadX®, T-Kernel®, QNX® and eCOS®. Please note that MapuSoft may provide further support to include additional APIs or operating systems not listed. For a current listing visit <http://mapusoft.com/products/offerings> or email: [info@mapusoft.com](mailto:info@mapusoft.com)

NUCLEUS APIs			
<i>Task Functions</i>		<i>Partitions</i>	<i>Interrupts</i>
NU_Create_Task	NU_Receive_From_Pipe	NU_Allocate_Partition	NU_Setup_Vector
NU_Delete_Task	NU_Established_Pipes	NU_Deallocate_Partition	NU_Control_Interrupts
NU_Reset_Task	NU_Pipe_Information	NU_Create_Partition_Pool	NU_Local_Control_Interrupts
NU_Restart_Task	NU_Pipe_Pointers	NU_Delete_Partition_Pool	NU_Restore_Interrupts
NU_Terminate_Task		NU_Partition_Pool_Information	
NU_Resume_Task	<b>MailBox</b>	NU_Partition_Pool_Pointers	<b>Memory Pools</b>
NU_Suspend_Task	NU_Create_Mailbox		NU_Protect
NU_Relinquish	NU_Delete_Mailbox	<b>Timers</b>	NU_Unprotect
NU_Sleep	NU_Reset_Mailbox	NU_Create_Timer	
NU_Change_Priority	NU_Send_To_Mailbox	NU_Delete_Timer	<b>Events</b>
NU_Change_Preemption	NU_Broadcast_To_Mailbox	NU_Control_Timer	NU_Create_Event_Group
NU_Change_Time_Slice	NU_Receive_From_Mailbox	NU_Get_Remaining_Time	NU_Delete_Event_Group
NU_Check_Stack	NU_Establish_Mailboxes	NU_Established_Timers	NU_Set_Events
NU_Current_Task_Pointer	NU_Mailbox_Formation	NU_Reset_Timer	NU_Retrieve_Events
NU_Established_Tasks	NU_Mailbox_Pointers	NU_Timer_Information	
NU_Task_Information		NU_Timer_Pointers	<b>Clocks</b>
NU_Task_Pointers	<b>Semaphores</b>		NU_Set_Clock
	NU_Create_Semaphore	<b>Device Drivers</b>	NU_Retrieve_Clock
<b>Queues</b>	NU_Delete_Semaphore	NU_Create_Driver	
NU_Create_Queue	NU_Reset_Semaphore	NU_Request_Driver	<b>Miscellaneous</b>
NU_Delete_Queue	NU_Obtain_Semaphore	NU_Delete_Driver	NU_Release_Information
NU_Reset_Queue	NU_Release_Semaphore	NU_Resume_Driver	NU_License_Information
NU_Send_To_Front_Of_Queue	NU_Established_Semaphores	NU_Suspend_Driver	NU_Disable_History_Saving
NU_Send_To_Queue	NU_Semaphore_Information	NU_Established_Drivers	NU_Enable_History_Saving
NU_Broadcast_To_Queue	NU_Semaphore_Pointers	NU_Driver_Pointers	NU_Make_History_Entry
NU_Receive_From_Queue			NU_Retrieve_History_Entry
NU_Established_Queues	<b>Memory Pools</b>	<b>HISR</b>	
NU_Queue_Information	NU_Create_Memory_Pool	NU_Activate_HISR	
NU_Queue_Pointers	NU_Delete_Memory_Pool	NU_Create_HISR	
	NU_Allocate_Memory	NU_Current_HISR	
<b>Pipes</b>	NU_Deallocate_Memory	NU_Register_LISR	
NU_Create_Pipe	NU_Established_Memory_Pool	NU_Delete_HISR	
NU_Delete_Pipe	NU_Memory_Pool_Information	NU_Established_HISRs	
NU_Reset_Pipe	NU_Memory_Pool_Pointers	NU_HISR_Pointers	
NU_Send_To_Front_Of_Pipe		NU_HISR_Information	
NU_Send_To_Pipe			
NU_Broadcast_To_Pipe			

## MapuSoft Technologies, Inc.

Porting embedded applications from one OS to another OS is often an underestimated, tedious and time-consuming task. It also requires expensive and skillful resources that take away the focus on building your product. Embedded applications demand more and more performance, scalability and development flexibility from the underlying OS. Developers are forced to change their OS or extend support for more than one OS quickly as the market demands. Developers find that they need to leverage the existing software and knowledge base when migrating to next generation platforms. This has brought a need for the development of highly re-usable software that can run across proprietary and multiple commercial operating systems as well as utilizes open source components or other low cost alternatives.

It's not easy for developers to adapt existing software to a new OS or enable it to support multiple operating systems without incurring high costs and increasing time to market entry. MapuSoft offers OS PAL, OS Abstractor and OS Changer products to help developers streamline development processes and re-use their embedded software on one or more operating systems. MT offers porting, integration, support and training services to help developers easily migrate from legacy platforms to the next generation.

## MapuSoft Custom Services

- Provide full porting, integration and validation services
- Extend OS Changer APIs
- Migrate in-house abstraction to OS Abstractor framework
- Add OS Abstractor support to your proprietary operating system
- Offer on-site and off-site training on operating systems and advanced porting techniques

# FREE OS Changer Trial Software

Go to: [www.mapusoft.com/downloads](http://www.mapusoft.com/downloads)

- Receive 30 days of FREE technical support!

MapuSoft Technologies, Inc.

1301 Azalea Road, Mobile, AL 36693 USA

Toll Free: 1-877-MAPUSOFT (1-877-627-8763)

Tel: 251-665-0280, Fax: 251-665-0288

[www.MapuSoft.com](http://www.MapuSoft.com)

**MAPUSOFT**  
Porting Made Easy<sup>SM</sup>