

## STEP ONE • Choose an option

## STEP TWO

### Option One Optimized Code Generation

Your  $\mu$ ITRON App:  
Import using the  
legacy porting tool

Porting steps:  
Replace headers, Combine main ( )

Run/debug application  
using emulator on host

Configure target OS, Profiler,  
Interface Optimizer & system settings

Generate Interface  
code for target OS

Output:  
Unmodified application code  
Optimized interface files

Native compiler

Output:  
Cross-OS Interface object  
Existing OS Interface object  
Application objects

Continue to STEP TWO

### Option Two Full Source Package Generation

Your  $\mu$ ITRON App

Porting steps:  
Replace headers,  
Combine main ( )  
and Initialize app

Native  
compiler

Generate source  
package from OS PAL  
and perform manual  
configuration

Output:  
Cross-OS Interface library  
Existing OS Interface library  
Application object code

Continue to STEP TWO

Linker

Your  $\mu$ ITRON App  
executable

Download/run  
on your target OS

Generated Profiler  
data (optional)

View data using  
OS PAL Profiler

### OPTIMIZED CODE GENERATION: OPTION ONE

- Legacy Porting Tool to easily import your application into OS PAL
- Perform your porting work on an Eclipse-based Windows or Linux host machine with provided GNU tools for x86
- Generate optimized  $\mu$ ITRON Interface code for your target, specific to your  $\mu$ ITRON application
- Generate project files for your target IDE/tools environment
- Enable target profiling of the  $\mu$ ITRON Interface and of the application functions to collect valuable performance data and generate comparative performance reports
- Selectively optimize each  $\mu$ ITRON Interface function for performance based on its usage in your application
- Automatically generate initialization & configuration code based on the settings you chose in the GUI-based wizard

### FULL SOURCE PACKAGE GENERATION: OPTION TWO

- Suitable for applications that link with other libraries which also needed to be ported
- Use with your preferred IDE/tools instead of the provided OS PAL Eclipse-based environment
- Provides a Porting Kit in a source code library format which contains all the  $\mu$ ITRON Interface functions for a specific target OS
- Requires manual configuration and initialization instead of using the OS PAL GUI-based wizard

## Technical Highlights

### It's Not Your Typical Wrapper

- > Provides most of the OS features by itself and does not depend on the OS, except for a few features such as priority scheduling, semaphore, messaging and thread suspend/resume

### Includes a Process Feature

- > Port your application to a single or to multiple processes utilizing the user shared region provided for your global variables
- > Create a new process by compiling the application separately or by launching it from your main application
- > Provides software-based process features, even if the underlying target OS does not offer support
- > Applications can pre-allocate heap memory during process creation
  - \* Also set maximum limits regarding the amount of heap memory each application can use to prevent applications from using up all of the system memory and impacting other applications

### API Flexibility

- > Use the same OS Changer Interface APIs within a single or across multiple processes

### Thread Pooling

- > Applications can pool threads to increase platform robustness & performance by eliminating the overhead associated with actual task creation & task deletion at run-time

### Mission Critical Features

- > Applications have the ability to asynchronously recover from fatal software errors through a soft reset by rolling the stack back to the start of the application

### Highly Scalability

- > The OS PAL GUI-based wizard reads your application to custom generate optimized  $\mu$ ITRON Interface code that is specific to your application resulting in increased performance and reduction of memory footprint

### Target Independence

- > Products support any target hardware supported by your target OS Architecture
- > Support for 16/32/64 bit architectures

### Processor

- > SMP and UP modes are supported

### $\mu$ ITRON

- > Supports version 4.0 standard

## $\mu$ ITRON Interface API Coverage & Target OS Support

Please refer to the latest release notes for the  $\mu$ ITRON API coverage offered by the OS Changer Interface, found at the link below:

[http://mapusoft.com/admin/wp-content/uploads/release\\_notes\\_138.pdf](http://mapusoft.com/admin/wp-content/uploads/release_notes_138.pdf)

Below are the target operating systems supported by the OS Changer -  $\mu$ ITRON Porting Kit:

Android®	MQX®	Solaris®	Unix®
Linux® 2.4/2.6	NetBSD®	ThreadX®	VxWorks® 6x/5x
LynxOS®	QNX® Neutrino® RTOS	T-Kernel®	Windows® XP/Vista/7/CE
LynxOS-SE®	RT Linux®	$\mu$ C/OS III®	

A free evaluation can be downloaded here: <http://mapusoft.com/downloads/>

You can contact MapuSoft to request a license key for evaluation here: <http://mapusoft.com/contact>

User manuals & technical documentation can be found here: <http://mapusoft.com/products/techdata/>

For any technical or sales questions please submit a ticket at the MapuSoft support site here: <http://mapusoft.com/support/>