OS CHANGER®

TECHNICAL DATASHEET

µC/OS PORTING KIT

OS Changer µC/OS™ Porting Kit is a C/C++ source-level virtualization technology that allows you to easily re-use your software developed using µC/OS APIs on another OS, while providing real-time performance. It eliminates the manual porting effort, saves money and shortens the time to market. OS Changer can also be used to simulate the µC/OS Interface on a host machine. OS Changer Interface connects to your existing application that was developed on µC/OS while the OS Abstractor Target Specific Module (specific to your target OS) provides the connection to the OS you are moving to.

OPTIMIZED CODE GENERATION: OPTION ONE

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

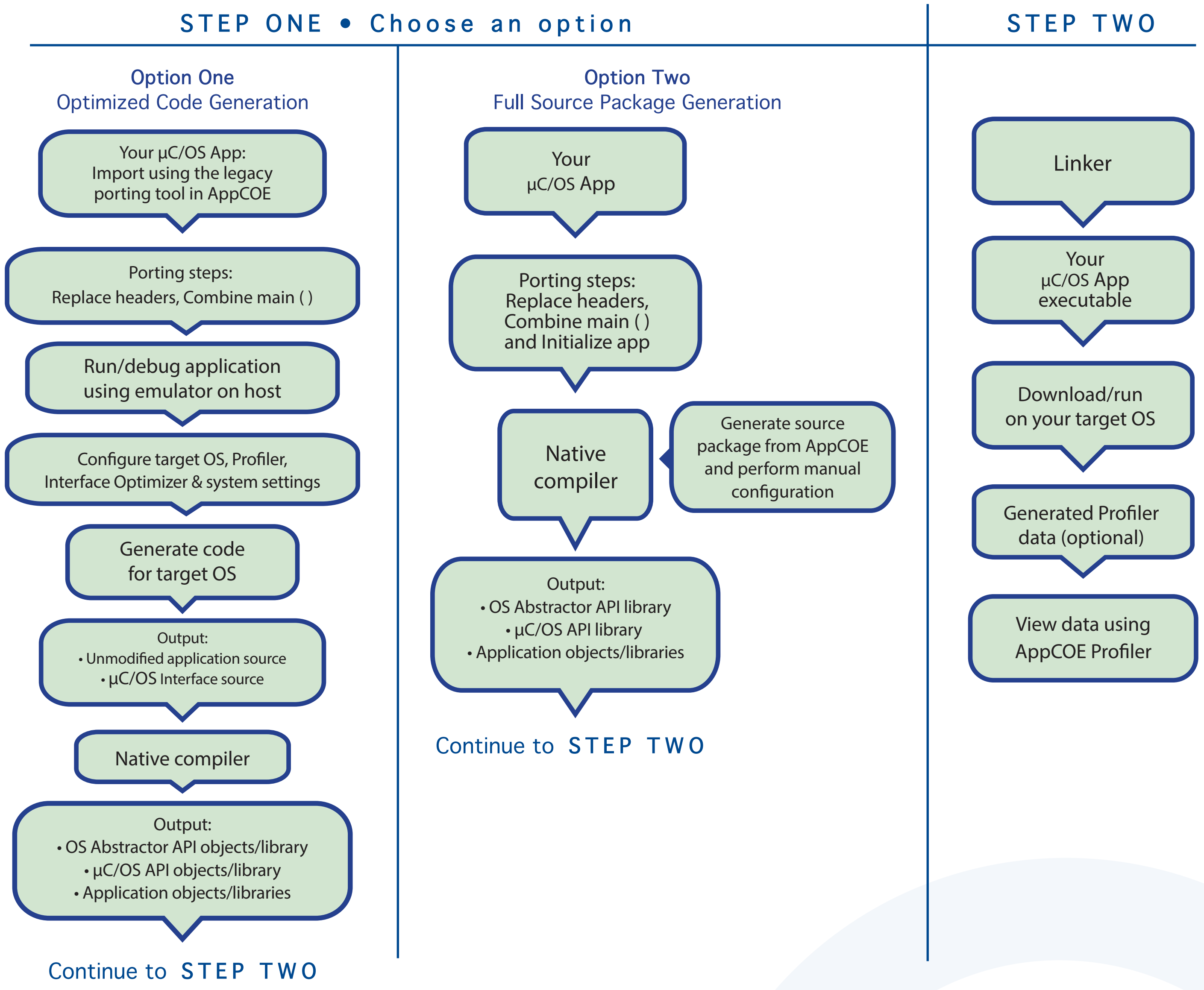
FULL SOURCE PACKAGE GENERATION: OPTION TWO

- Use with your preferred IDE/tools instead of the AppCOE development environment
- Provides a Porting Kit in a source code format which contains all the µC/OS Interface functions for a specific target OS
- Requires manual configuration and initialization instead of using the AppCOE GUI-based wizard

OSCHANGER®

TECHNICAL DATASHEET

µC/OS PORTING KIT


STEP ONE • Choose an option
STEP TWO
Option One
Optimized Code Generation
Your µC/OS App: Import using the legacy porting tool in AppCOE
Porting steps: Replace headers, Combine main ( )
Run/debug application using emulator on host
Configure target OS, Profiler, Interface Optimizer & system settings
Generate code for target OS
Output: • Unmodified application source • µC/OS Interface source
Native compiler
Output: • OS Abstractor API objects/library • µC/OS API objects/library • Application objects/libraries
Continue to STEP TWO
Option Two
Full Source Package Generation
Your µC/OS App
Porting steps: Replace headers, Combine main ( ) and Initialize app
Native compiler
Generate source package from AppCOE and perform manual configuration
Output: • OS Abstractor API library • µC/OS API library • Application objects/libraries
Continue to STEP TWO
Linker
Your µC/OS App executable
Download/run on your target OS
Generated Profiler data (optional)
View data using AppCOE Profiler

OS CHANGER® µC/OS PORTING KIT

TECHNICAL DATASHEET

Technical Highlights

Includes a Process Feature

> Port your application to a single or 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
  * 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

> OS Abstractor APIs also available for use in your µC/OS application
> OS Changer µC/OS Interface can be used within a single or across multiple applications

Thread Pooling

> Applications can pool threads to increase platform robustness and performance by eliminating the overhead associated with actual task creation and 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 Scalable

> The AppCOE GUI-based wizard reads your application to custom generate optimized µC/OS Interface code that is specific to your application resulting in increased performance and reduction of memory footprint

Target Hardware Independence

> Products support any target hardware supported by your target OS architecture, including 32/64 bit & SMP/UP architectures

In-house OS Support

> Can easily be extended to support your in-house OS

µC/OS Interface API Coverage & Target OS Support

You can find the supported µC/OS APIs here:
https://www.mapusoft.com/wp-content/uploads/documents/Release_Notes-ucos-APIs.pdf

Below are the target operating systems supported by the OS Changer µC/OS Porting Kit:

Android®	LynxOS-178®	Nucleus®	ThreadX®
eCOS®	micro-ITRON®	QNX Neutrino RTOS®	Unix®
Linux/POSIX	Freescale MQX®	RT Linux®	VxWorks®
LynxOS®	NetBSD®	Solaris®	In-House
LynxOS-SE®	FreeRTOS™		

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

MAPUSOFT