



APPCOE Quick Start Guide

Release 1.8.1

Copyright (c) 2021

MapuSoft Technologies, Inc,
Unit 50197
Mobile, AL 36605
www.mapusoft.com

Table of Contents

Chapter 1 - About this Guide	4
Objectives	5
Audience	5
How to Use This Manual	5
Contents in this Guide	5
System Information	5
Document Conventions	6
MapuSoft Technologies and Related Documentation	7
Requesting Support	9
Registering a New Account	9
Submitting a Ticket	9
Live Support	10
Documentation Feedback	10
Chapter 2 – Introduction to AppCOE	11
About AppCOE	12
AppCOE IDE	13
Chapter 3 – Installation of Guest Additions on Windows..	16
Step-1: VirtualBox Set-up on Windows Operating System	17
Step-2: Importing Guest (Linux) Operating System	17
Step-3: Share Folders between Host & Guest Operating System	18
Step-4: Mounting of shared folder on Guest Operating System	20
Chapter 4 – Installation of Guest Additions on Linux.....	21
Step-1: VirtualBox Initial Set-up on Host Operating System	22
Step-2: Creating Guest (Windows) Operating System	22
Step-3: Running your virtual machine on Host Operating System	25
Step-4: Mounting of shared folder on Guest Operating System	25
Chapter 5 – Porting a Legacy VxWorks App to Linux.....	26
Step 1: Import Legacy Application and Project File into AppCOE	27
Step 2: Link-in MT's Products with the Application	27
Step 3: Build the Application to Include MT's Products	28
Step 4: Run the Application on the Host in AppCOE	28
Step 5: Generate Code to Run the Application on the New Target OS	28
Step 6: Run the Application on the Target OS	30
Chapter 6 – Profiler Demo	31
Step 1: Open the Profiler	32
Step 2: Load the Profiler Data File	32
Step 3: Show the Graphs for the Profiler Data	32
Chapter 7 – Full Library Package Demo	33
Step 1: Create the Full Library Package	34
Step 2: Show the Full Library Package Created by AppCOE	34
Chapter 8 – Ada-C/C++ Changer Demo	35
Step 1: Use Ada-C/C++ Changer	36
Step 2: Build the new C/C++ code	37
Step 3: Run the new C/C++ code	37
Revision History	38

List of Figures

Figure 1_1: Create a Support Ticket from AppCOE.....	9
Figure 2_1: AppCOE Overview Diagram.....	12
Figure 2_2: AppCOE IDE.....	13
Figure 3_2: Shared Folder Settings window.....	19
Figure 3_3: After set-up completion.....	19
Figure 4_1: Oracle VM VirtualBox Manager	22
Figure 4_2: Virtual Hard Disk	23

List of Tables

Table-1: System Information.....	5
Table-2: Notice Icons	6
Table-3: Text and Syntax Conventions	6
Table-4: Document Description Table	7

Chapter 1 - About this Guide

This chapter contains the following topics:

Objectives

Audience

How to Use This Manual

Contents in this Guide

System Information

Document Conventions

MapuSoft Technologies and Related Documentation

Requesting Support

Documentation Feedback

Objectives

This manual contains instructions on how to become familiar with AppCOE product and use it with ease.

Audience

This manual is designed for Mapusoft Customers/Distributors who already have valid licenses. This manual requires technical knowledge and experience with the Embedded Systems, access to Mapusoft FTP, distributor links

How to Use This Manual

This manual explains how to quickly set-up AppCOE on Windows/Linux Host and run the demos that came along AppCOE.

Please contact MapuSoft if you need help in getting the components mentioned below.

- Application Common Operating Environment (AppCOE) – [Download Here](#)
- AppCOE License Key – [Request One Here](#)
- VxWorks Sample Application & Project Files – [Download Here](#)
- Ada Sample Application – [Download Here](#)

Contents in this Guide

Introduction to AppCOE
 Setting up Guest Additions to Host OS
 Sharing AppCOE Projects & Profiler files between Windows/Linux
 Porting a VxWorks Sample Application on Host

System Information

Table-1 defines the system information used in this manual.

Table-1: System Information

System Details	System Version	More Information
Host System Configuration	Windows 7 Ultimate,	Processor: Intel(R) Pentium(R) D CPU 2.80 GHz Installed Memory (RAM): 3.19 GB System Type: 32-bit operating system
	Windows 8.1	Processor: Intel(R) Pentium(R) D CPU 2.80 GHz Installed Memory (RAM): 3.19 GB System Type: 64-bit operating system
Target System Configuration	Linux - CentOS release 5.5 (Final)	Kernel: 2.6.18-308.11.1.el5 #1 Installed Memory (RAM): 3.19 GB System Type: i686 i386 & X86_64 GNU/Linux
Virtual Box	VirtualBox- 4.1.18-r78361-Win	https://www.virtualbox.org

AppCOE	1.8.1	http://www.mapusoft.com/
--------	-------	---

Document Conventions

Table-2 defines the notice icons used in this manual.

Table-2: Notice Icons


Icon	Meaning	Description
NOTE	Informational note	Indicates important features or notes.
	Caution	Indicates a situation that might result in loss of data or software damage.

Table-3 defines the text and syntax conventions used in this manual.

Table-3: Text and Syntax Conventions

Convention	Description
Courier New	Identifies Program listings and Program examples.
<i>Italic text like this</i>	<ul style="list-style-type: none"> Introduces important new terms. Identifies book names
COURIER NEW, ALL CAPS	Identifies File names.
Courier New, Bold	Identifies Interactive Command lines

MapuSoft Technologies and Related Documentation

Note: The below mentioned documentation can be found under the Help menu in AppCOE.

Table-4 define the list of manual available for AppCOE

Table-4: Document Description Table

User Guides	Description
System Configuration Guide	Provides detailed description on the system configuration to work with MapuSoft products. This guide: <ul style="list-style-type: none"> Describes the system requirements and configurations to get started with MapuSoft Technologies products
Application Common Operating Environment Guide	Provides detailed description of how to do porting and abstraction using AppCOE. This guide: <ul style="list-style-type: none"> Explains how to port applications Explains how to import legacy applications Explains how to do code optimization Explains how to generate library packages Explains on Application profiling and platform profiling
OS Abtractor Interface Reference Manual	Provides detailed description of how to use OS Abstraction. This guide: <ul style="list-style-type: none"> Explains how to develop code independent of the underlying OS Explains how to make your software easily support multiple OS platforms
VxWorks Interface Reference Manual	Provides detailed description of how to get started with VxWorks interface support that MapuSoft provides. This guide: <ul style="list-style-type: none"> Explains how to use VxWorks interface, port applications
micro-ITRON Interface Reference Manual	Provides detailed description of how to get started with micro-ITRON interface support that MapuSoft provides. This guide: <ul style="list-style-type: none"> Explains how to use micro-ITRON interface, port applications
pSOS Interface Reference Manual	Provides detailed description of how to get started with pSOS interface support that MapuSoft provides. This guide: <ul style="list-style-type: none"> Explains how to use pSOS interface, port applications
pSOS Classic Interface Reference Manual	Provides detailed description of how to get started with pSOS Classic interface support that MapuSoft provides. This guide: <ul style="list-style-type: none"> Explains how to use pSOS Classic interface, port applications
Nucleus Interface Reference Manual	Provides detailed description of how to get started with Nucleus interface support that MapuSoft provides. This guide: <ul style="list-style-type: none"> Explains how to use Nucleus interface, port applications
ThreadX Interface Reference Manual	Provides detailed description of how to get started with ThreadX interface support that MapuSoft provides. This guide: <ul style="list-style-type: none"> Explains how to use ThreadX interface, port applications

POSIX Interface Reference Manual	<p>Provides detailed description of how to get started with POSIX interface support that MapuSoft provides. This guide:</p> <ul style="list-style-type: none"> • Explains how to use POSIX interface, port applications
Windows Interface Reference Manual	<p>Provides detailed description of how to get started with Windows interface support that MapuSoft provides. This guide:</p> <ul style="list-style-type: none"> • Explains how to use Windows interface, port applications
μC/OS Interface Reference Manual	<p>Provides detailed description of how to get started with μC/OS interface support that MapuSoft provides. This guide:</p> <ul style="list-style-type: none"> • Explains how to use μC/OS interface, port applications
FreeRTOS Interface Reference Manual	<p>Provides detailed description of how to get started with FreeRTOS interface support that MapuSoft provides. This guide:</p> <ul style="list-style-type: none"> • Explains how to use FreeRTOS interface, port applications
RTLinux Interface Reference Manual	<p>Provides detailed description of how to get started with RTLinux interface support that MapuSoft provides. This guide:</p> <ul style="list-style-type: none"> • Explains how to use RTLinux interface, port applications
VRTX Interface Reference Manual	<p>Provides detailed description of how to get started with VRTX interface support that MapuSoft provides. This guide:</p> <p>Explains how to use VRTX interface, port applications</p>
QNX Interface Reference Manual	<p>Provides detailed description of how to get started with QNX interface support that MapuSoft provides. This guide:</p> <p>Explains how to use QNX interface, port applications</p>
Release Notes	<p>Provides the updated release information about MapuSoft Technologies new products and features for the latest release.</p> <p>This document:</p> <ul style="list-style-type: none"> • Gives detailed information of the new products • Gives detailed information of the new features added into this release and their limitations, if required

Requesting 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 conversation/ticket at <http://www.mapusoft.com/support>

Anyone can initially contact sales/admin/tech via the above mechanism however tech support is offered to only registered users or evaluation customers.

Registering a New Account

If you are a customer with valid tech support contract or a trial user, please request an account be created by providing your email address, company address, telephone number etc. by contacting sales@mapusoft.com. You will be provided via account name (your email) and also password to sign-in

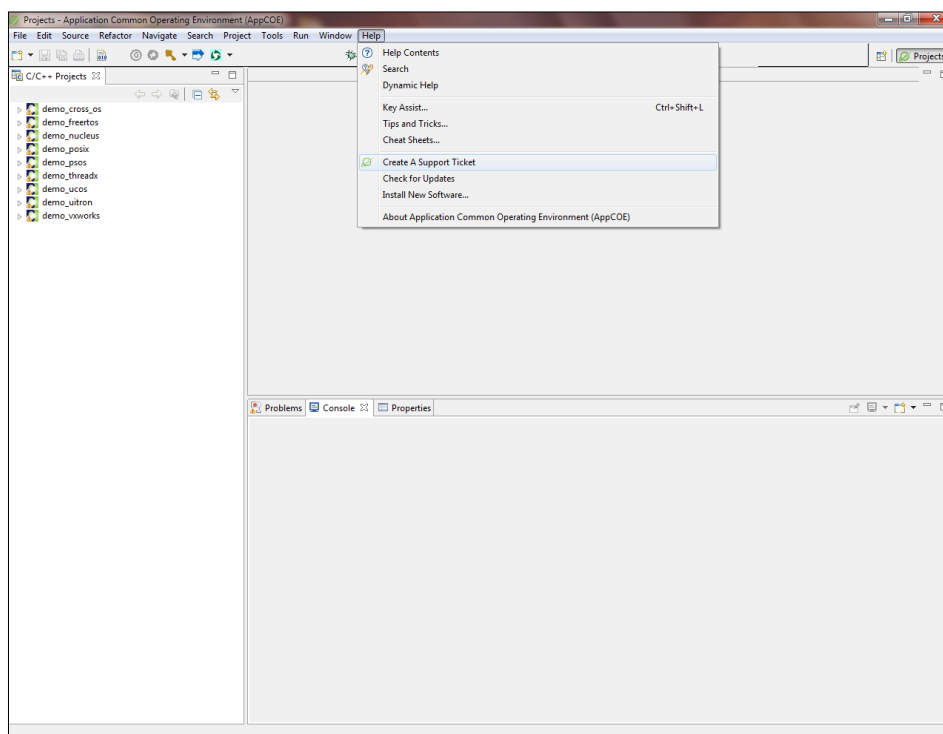
Submitting a Ticket

1. To submit a ticket, simple sign-in into your account <http://www.mapusoft.com/support> and open a conversation.
MapuSoft Support personnel will get back to you within 48 hours with a valid response.

2. To submit a ticket from within AppCOE IDE

From AppCOE main menu, Select Help > Create a Support Ticket as shown in below Figure 1_1Figure 1_1.

Figure 1_1: Create a Support Ticket from AppCOE



To submit a ticket, simply sign-in into your account <http://www.mapusoft.com/support> and open a conversation.

MapuSoft Support personnel will get back to you within 48 hours with a valid response.

Live Support

Chat: MapuSoft Technologies also provides technical support through Live Chat from www.mapusoft.com website. If Chat is offline, please leave a detailed message including your email address, telephone number and company name so that MapuSoft personnel's can quickly respond to either responding to your chat by calling you on the number that you have provided

Telephone: You can also reach us at our toll-free number: **1-877-627-8763** and press the tech support option to contact MapuSoft tech support team for any urgent assistance.

Documentation Feedback

We greatly appreciate your feedback. Simple sign-in or just start a conversation and let us know via: <http://www.mapusoft.com/support>

Chapter 2 – Introduction to AppCOE

About AppCOE

AppCOE IDE

About AppCOE

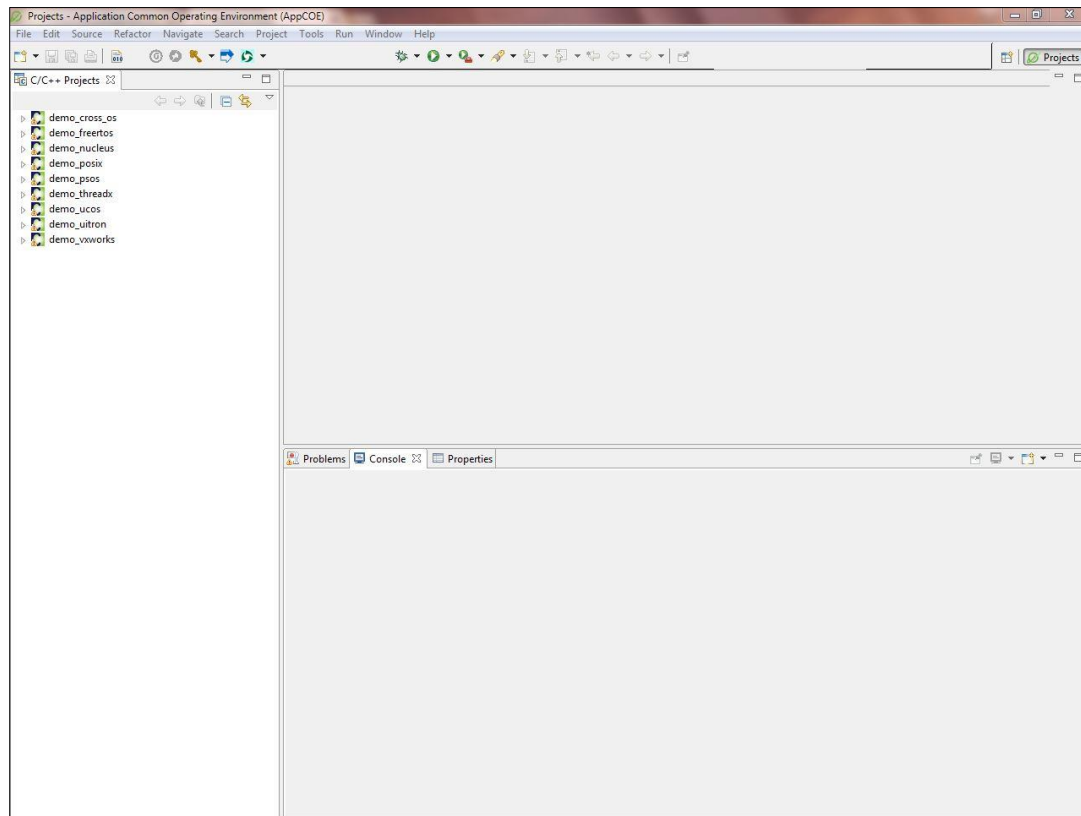
AppCOE is an Eclipse based IDE. AppCOE integrates software interoperability & reuse tools like OS Changer and OS Abstractor with Eclipse's CDT to offer an IDE for developing and porting embedded applications on many host/target platforms.

With AppCOE you can perform the following actions:

- Creation of C and C++ AppCOE projects
- Porting of legacy applications
- Host development with simulation for many OS applications
- Converting Ada source code to C/C++ code
- Platform and Application profiling
- Automatic configuration of any OS Changer and OS Abstractor APIs needed by the application
- Custom configuration of OS & OS Abstractor resources needed by the application
- Custom configuration of OS Abstractor for single or multi-application development (Process Feature support)
- Optimized source code generation
- Full Source Library Package generation

Contact MapuSoft to receive the components (license) needed for using AppCOE. The steps for using AppCOE are comprehensively described in AppCOE User Manual from **AppCOE > Help > Help Contents > Application Common Operating Environment User Manual (AppCOE)**. Below Figure 2_1 represents AppCOE Overview.

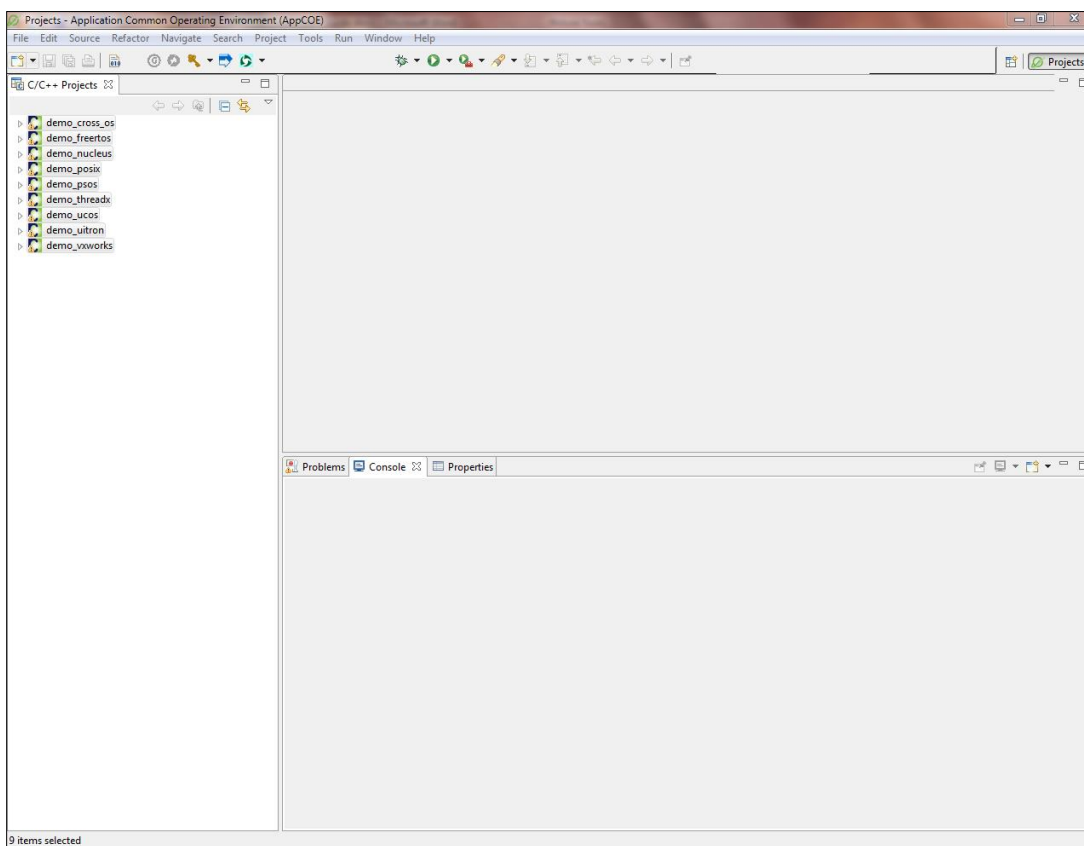
Figure 2_1: AppCOE Overview Diagram



AppCOE IDE

Note: Some features require a license. Click <http://mapusoft.com/downloads/> to request an evaluation license. Below Figure 2_2 represents AppCOE IDE.

Figure 2_2: AppCOE IDE



You can view the following project files on the left pane of the AppCOE IDE window

- **demo_ada_to_c** – Sample project file containing sample ADA application & converted C code
- **demo_cross_os** – Sample project file containing OS_Abstractor application using OS Abstractor API's
- **demo_freertos** -- Sample project file containing FreeRTOS application
- **demo_ucos** -- Sample project file containing μ C/OS application
- **demo_posix** – Sample project file containing POSIX application
- **demo_nucleus** – Sample project file containing Nucleus application
- **demo_threadx** – Sample project file containing ThreadX application
- **demo_psos** – Sample project file containing pSOS application
- **demo_uitron** – Sample project file containing uITRON application
- **demo_vxworks** – Sample project file containing sample VxWorks application
- **demo_windows** – Sample project file containing windows application

Note: If you don't have the valid license, you can only run the executables. To build, debug or generate code for different targets, you will need a valid license. Please [contact Mapusoft](#) for valid Licenses.

To share AppCOE Projects & Profiler files between Windows/Linux (Host-Guest OS), Please follow [Chapter 3](#) & [Chapter 4](#), if you know the process, you can skip these chapters

Chapter 3 – Installation of Guest Additions on Windows

This chapter contains the following topics:

- VirtualBox Set-up on Windows Operating System

- Importing Guest (Linux) Operating System

- Share Folders between Host & Guest Operating System

- Mounting of shared folder on Guest Operating System

Step-1: VirtualBox Set-up on Windows Operating System

VirtualBox is a powerful [virtualization](#) product for enterprise as well as home use. VirtualBox is also one of the professional solutions that is freely available as Open Source Software under the terms of the GNU General Public License (GPL) version 2. See "[About Virtual Box](#)" for an introduction.

1. VirtualBox comes in many different packages, and installation depends on your host operating system. If you have installed the software before, then jump to **Step-2**, otherwise continue forward
2. Download Pre-Configured **VirtualBox.zip** for Windows Host from MT's FTP server along the path <https://www.virtualbox.org/wiki/Downloads>
3. After downloading the files for Windows Host from the server, extract it & launch the file **VirtualBox-4.1.18-78361-Win.exe** and go through the installation wizard, as installation should be straightforward
4. If you run into trouble or have special requirements, please refer to [Chapter 2, Installation details](#) [VirtualBox Website] for details about the various installation methods

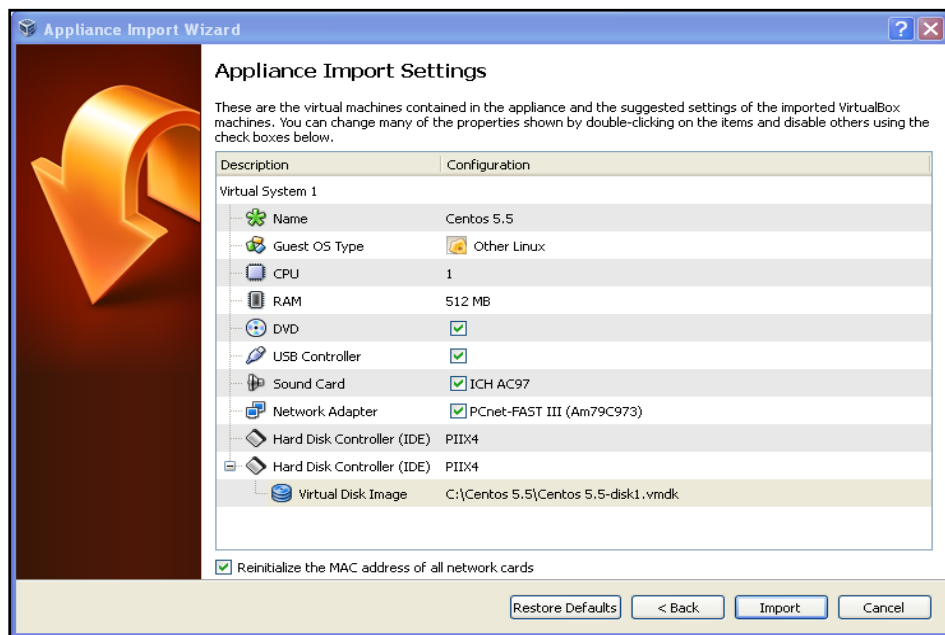
Note: You will need password for accessing the MT's FTP site. [Contact Mapusoft](#) if you don't have the access permissions.

Step-2: Importing Guest (Linux) Operating System


Note: Before starting this step, make sure you have a minimum of 20GB of disk space in your system.

1. After installation, you can start VirtualBox as follows: On a Windows host, in the standard "**Programs**" menu, click on the item in the "**Virtual Box**" group. On Vista or Windows 7, you can also type "**Virtual Box**" in the search box of the "**Start**" menu
2. To **import** an appliance, simply double-click on the OVF/OVA file
3. Alternatively, select "**File**" -> "**Import appliance**" from the Manager window
4. In the file dialog that comes up, navigate to the file with **CentOS5.5.ovf** file extension
5. If VirtualBox can handle the file, a dialog similar to the following Figure 3_1 as shown below

Figure 3_1: Appliance Import Settings



Step-3: Share Folders between Host & Guest Operating System

1. To log-into Centos 5.5, use the following credentials, User – **root**, Password – **mapusoft**
2. In the window of a running VM, you can select "**Shared Folders**" from the "**Devices**" menu, or click on the folder icon  on the status bar in the bottom right corner.

If your VM is not currently running, you can configure shared folders in your virtual machine's "**Settings**" dialog as shown in Figure 3_2 & Figure 3_3 .

Note: Make sure read/write privileges are provided to the shared folder.

Figure 3_2: Shared Folder Settings window

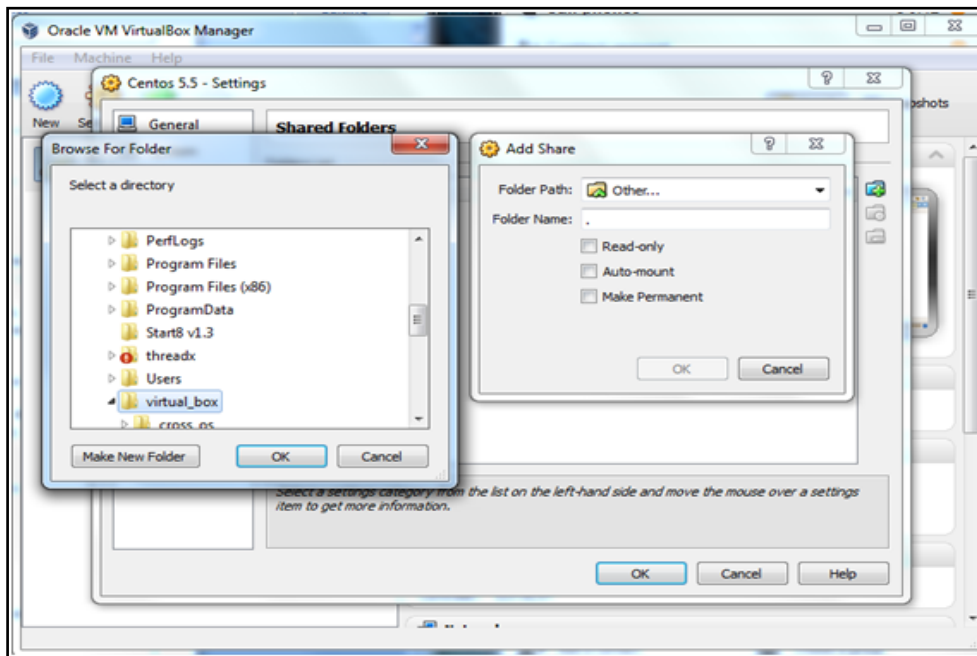
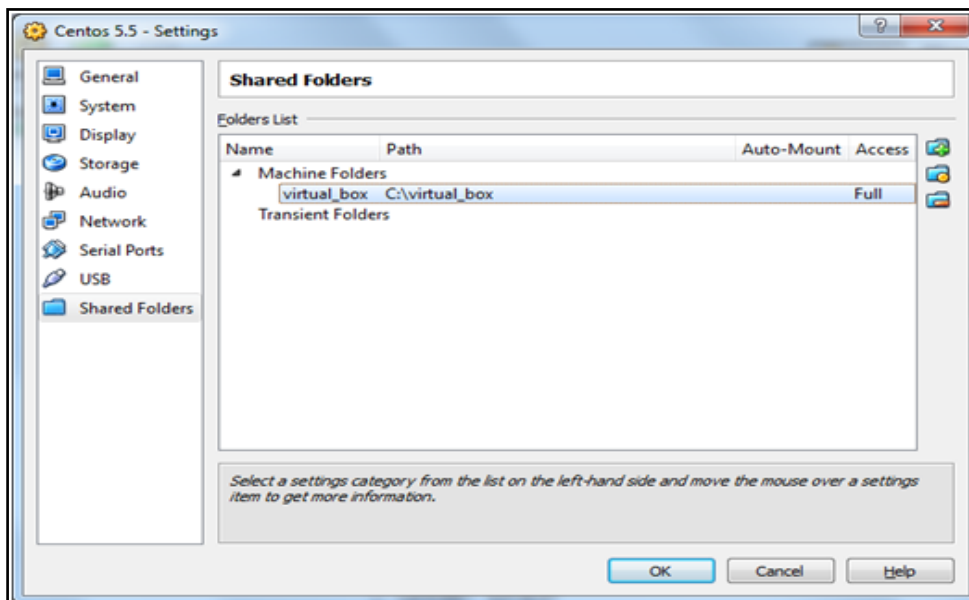


Figure 3_3: After set-up completion



Step-4: Mounting of shared folder on Guest Operating System

Note: If you cannot see the share folder, you have to disable the firewall (anti-virus and Windows) on your host system.

1. **[Manual Mounting]** Open Terminal and mount the virtual_box using the following command
sudo mount -t vboxsf virtual_box /media/share

Note: “**virtual_box**” is the name of the shared folder (use lowercase) i.e. the share name specified with VBoxManage or the GUI, and “**/media/share**” is the mount-point with the path where you want the share to be mounted on the guest.

2. **[Automatic Mounting]** Open Terminal and execute the following command
sudo gedit /etc/init.d/rc.local
3. Add the following line to the file and save it
sudo mount -t vboxsf virtual_box /media/share

Note: Refer <http://www.virtualbox.org/manual/ch04.html> for more clarifications on setting up shared folders on host operating system.

Chapter 4 – Installation of Guest Additions on Linux

This chapter contains the following topics:

VirtualBox Initial Set-up on Host Operating System

Creating Guest (Windows) Operating System

Running your virtual machine on Host Operating System

Mounting of shared folder on Guest Operating System

Step-1: VirtualBox Initial Set-up on Host Operating System

VirtualBox is a powerful [virtualization](#) product for enterprise as well as home use. VirtualBox is also one of the professional solutions that are freely available as Open Source Software under the terms of the GNU General Public License (GPL) version 2. See "[About VirtualBox](#)" for an introduction.

1. VirtualBox comes in many different packages, and installation depends on your host operating system. If you have installed the software before, then jump to **Step-2**, otherwise continue forward
2. Download the **VirtualBox** related files for Linux Host from https://www.virtualbox.org/wiki/Linux_Downloads
3. After downloading the files for Linux Host, launch the file **executable** and go through the installation wizard, as installation should be straightforward
4. If you run into trouble or have special requirements, please refer to [Chapter 2, Installation details](#) [VirtualBox Website] for details about the various installation methods

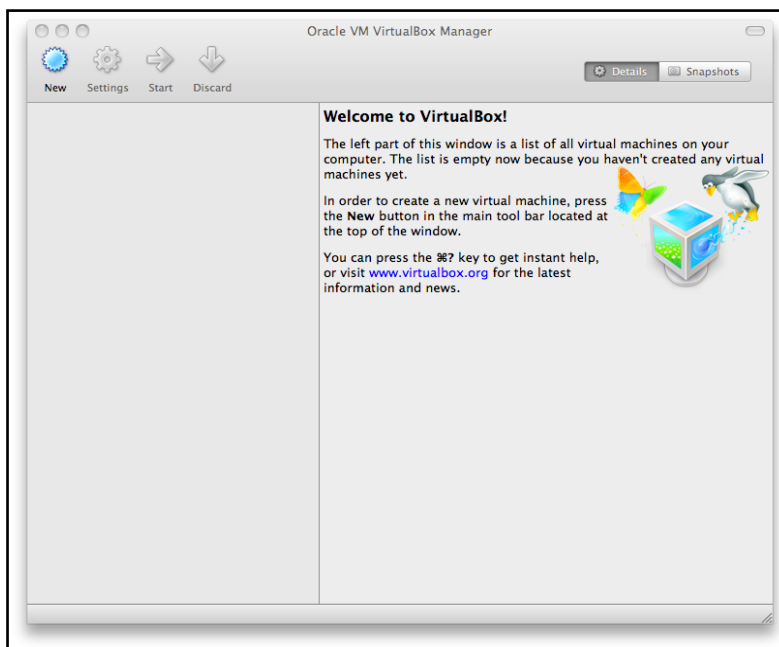
Note: You can also install VirtualBox through command line.

Step-2: Creating Guest (Windows) Operating System

Note: Before starting this step, make sure you have a minimum of 20GB of disk space in your system.

1. After installation, you can start VirtualBox as follows: Depending on your desktop environment, a "**VirtualBox**" item may have been placed in either the "**System**" or "**System Tools**" group of your "**Applications**" menu. Alternatively, you can type **VirtualBox** in your terminal
2. When you start **VirtualBox** for the first time, a window like the following Figure 4_1 should come up

Figure 4_1: Oracle VM VirtualBox Manager

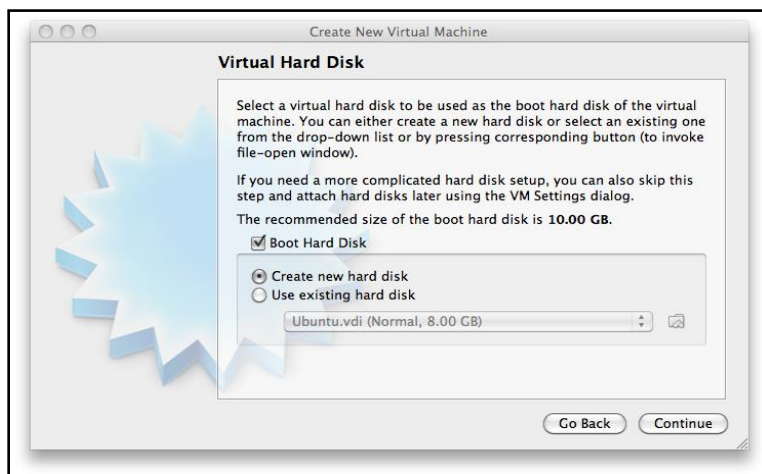


3. Click on the **"New"** button at the top of the VirtualBox Manager window. A wizard will pop up to guide you through setting up a new virtual machine (VM)
4. On the following pages, the wizard will ask you for the bare minimum of information that is needed to create a VM, in particular
5. The **VM name** will later be shown in the VM list of the VirtualBox Manager window, and it will be used for the VM's files on disk
6. For **"Operating System Type"**, select the operating system that you want to install
7. On the next page, select the **memory (RAM)** that VirtualBox should allocate every time the virtual machine is started [1GB RAM Advised]

Note: Choose this setting carefully! The memory you give to the VM will not be available to your host OS while the VM is running, so do not specify more than you can spare. For example, if your host machine has 1 GB of RAM and you enter 512 MB as the amount of RAM for a particular virtual machine, while that VM is running, you will only have 512 MB left for all the other software on your host. If you run two VMs at the same time, even more memory will be allocated for the second VM (which may not even be able to start if that memory is not available). On the other hand, you should specify as much as your guest OS (and your applications) will require to run properly.

8. Next, you must specify a **virtual hard disk** for your VM and the wizard is shown in Figure 4_2 .


Figure 4_2: Virtual Hard Disk



9. To create a new, empty virtual hard disk, press the **"New"** button
10. This brings up another window, the **"Create New Virtual Disk Wizard"**, which helps you create a new disk image file in the new virtual machine's folder and select **VMDK** (Virtual Machine Disk) file type
11. Select **Dynamically Allocated storage type** and then proceed further
12. Provide the **Virtual disk file** location & size and **create new virtual disk**

Note: Refer <http://www.virtualbox.org/manual/ch01.html> for more clarifications on setting up guest operating system.

Step-3: Running your virtual machine on Host Operating System

1. To start your virtual machine, Double-click on its entry in the list within the Manager window
2. When a VM gets started for the first time, another wizard, the "**First Start Wizard**", will pop up to help you select an **installation medium**. Since the VM is created empty, it would otherwise behave just like a real computer with no operating system installed. It will do nothing and display an **error** message that no bootable operating system was found
3. If you have physical CD or DVD media from which you want to install your guest operating system (e.g. in the case of a **Windows installation** CD or DVD), put the media into your host's CD or DVD drive.
4. Then, in the wizard's drop-down list of installation media, select "**Host drive**" with the correct drive letter. This will allow your VM to access the media in your host drive, and you can proceed to install from there.
5. If you have downloaded installation media from the **Internet** in the form of an ISO image file, **mount** the ISO file directly. VirtualBox will then present this file as a CD or DVD-ROM drive to the virtual machine
6. Follow the installation procedure of Windows and login to your Guest Operating System
7. In the window of a running VM, you can select "**Shared Folders**" from the "**Devices**" menu, or click on the folder icon  on the status bar in the bottom right corner.
8. If your VM is not currently running, you can configure shared folders in your virtual machine's "**Settings**" dialog

Note: Make sure read/write privileges are provided to the shared folder.

Step-4: Mounting of shared folder on Guest Operating System

Note: If you cannot see the share folder, you have to disable the firewall (anti-virus and Windows) on your host system.

1. Open Windows Explorer and look for "**My Networking Places**" -> "**Entire Network**" -> "**VirtualBox Shared Folders**". By right-clicking on a shared folder and selecting "**Map network drive**" from the menu that pops up, you can assign a drive letter to that shared folder
2. Alternatively, on the Windows command line, use the following command

net use x: \\vboxsvr\sharename

While **vboxsvr** is a fixed name (note that vboxsrv would also work), replace "**x:**" with the drive letter that you want to use for the share, and **sharename** with the share name specified with **VBoxManage**

Note: Refer <http://www.virtualbox.org/manual/ch04.html> for more clarifications on setting up shared folders on host operating system.

Chapter 5 – Porting a Legacy VxWorks App to Linux

This chapter contains the following topics:

- Import Legacy Application and Project File into AppCOE

- Link-in MT's Products with the Application

- Build the Application to Include MT's Products


- Run the Application on the Host in AppCOE

- Generate Code to Run the Application on the New Target OS

- Run the Application on the Target OS

Step 1: Import Legacy Application and Project File into AppCOE

Talking-Point: First I'm going to import my legacy VxWorks application and Workbench project file into AppCOE.

1. Click on the **Arrow** pointing down next to the green Porting button  on the toolbar in AppCOE
2. Chose **VxWorks**
3. Chose **Import Workbench Project**
4. Browse to the location of the sample application and project files on your computer
5. Click **OK**
6. Click **Next**
7. Click the check-box next to **simple_vxworks_app**
8. Click **Finish**
9. In the field for Root Task Prototype, type "**int windDemo(int argc)**"
10. In the field for Root Stack Task Size, type "**OS_MIN_STACK_SIZE**"
11. Click **OK**
12. Expand your new project folder in the APPCOE C/C++ Projects pane
13. Delete **main.c**

Talking-Point: I'm deleting main.c because it was automatically imported from my legacy application; however in this particular project there was no code to be ported in that file. If there had been, I would move that code into MT's main file and delete the old main.c

Step 2: Link-in MT's Products with the Application

Talking-Point: Now that my application is in AppCOE , I'm ready to link-in MapuSoft's products.

1. Double click on the **winddemo.c** file in the **Source** folder in your project to open it
2. Comment out the #include directives by adding **/*** at the beginning and ***/** at the end as shown below. The text should turn green once the comment is active

Talking-Point: I'm going to comment out the native VxWorks header files, since the application won't need them anymore.

```

/*
#include "vxWorks.h"
#include "semLib.h"
#include "taskLib.h"
#include "msgQLib.h"
#include "wdLib.h"
#include "logLib.h"
#include "tickLib.h"
#include "sysLib.h"
#include "stdio.h"
*/

```

Talking-Point: I'm going to add MapuSoft's products to replace the VxWorks header files, which I commented out in the previous step.

1. Link-in MT's header files with the application by removing the comment markers in front of MapuSoft's header files

```
#include "os_target.h"
#include "vxworks_interface.h"
```

2. **Save** the file


Step 3: Build the Application to Include MT's Products

Talking-Point: Now I'm going to rebuild my application to include MapuSoft's products.

1. Select the top level (the project name) of the project `simple_vxworks_app`
2. Right click
3. Select **Build Project**

Step 4: Run the Application on the Host in AppCOE

Talking-Point: Now that my application is using MapuSoft's products, I'm able to run this real-time VxWorks application on a host for simulation and debugging. MapuSoft provides the best possible simulation because we do not add a scheduler which would cause a performance strain. The only constraint for this application is the non real-time OS, Windows, being used as a host. Also, debugging on a readily available host machine, such as this Windows computer is much easier than debugging directly in the target environment.

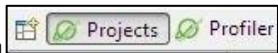

1. Right click the project that you created
2. Select **Debug As -> AppCOE Local C/C++ Application**
3. Once the Debug prospective is open, click on the **resume button** 
4. Bring up the debugger console (black box) to show the application's execution

Talking-Point: Here is my VxWorks application running on the host.

5. When finished, close the console to stop it from running

Step 5: Generate Code to Run the Application on the New Target OS

Talking-Point: Now I'm ready to move my VxWorks application Linux.

1. Click on the **AppCOE Projects Perspective** button  to get back to your project
2. Select the project that you created
3. Click on the **Optimizer** button  from the toolbar in AppCOE
4. Choose **Linux** and **2.6.x** as the target OS from the drop down boxes
5. Select the box next to **Generate Project File**

6. In the field marked **Destination Path**, browse to the shared folder you set up
C:\virtual_box
7. Click **Next**
8. In the field marked **File Path to Store Profiled Data**, type
C:\virtual_box\simple_vxworks_app
12. In the field **Number of Data in Memory Before Each Write**, type “500” (replace default)
13. In the field **Maximum Profiler Data to Collect**, type “500” (replace default)
14. Click on the **Platform API Profiling** tab
15. Select the boxes next to **Enable OS Abstractor interface API Profiling** and
Enable VxWorks OS Changer interface API Profiling

Talking-Point: This Platform Profiling feature will provide me with data concerning utilization of MapuSoft’s APIs in my application. I will be able to view graphs and charts that detail performance data such as API execution time

16. Click on the **Application Functions Profiling** tab

Talking-Point: This Application Functions Profiling tab will provide me with data concerning the functions in my application. This data will be presented in charts and graphs to analyze and identify bottlenecks which are slowing down my application. I’m going to input certain functions that I want to track.

17. Select **Enable Application Function(s) Profiling**
18. Enter the name “taskHighPri” in the **Application Function** box
19. Click **Add**
20. Enter the name “taskLowPri” in the **Application Function** box
21. Click **Add**
22. Click **Next**
23. Show the Inline Feature, but keep it as default

Talking-Point: AppCOE reads my application and shows a list of all the APIs used here. When the code is generated, MapuSoft’s products will be scaled specifically for my application, rather than a one-size-fits-all abstraction. This feature allows me to move certain APIs inline to increase the performance of my application.

22. Click **Next**
23. Show each configuration tab (leave all options as default with Task Pooling and Process Features turned off – they won’t work with this sample application)

Talking-Point: AppCOE provides GUIs to configure MapuSoft’s products for my application. The changes are made automatically included when I generate the code. MapuSoft’s platform includes many advanced development features. For example, Task Pooling gives the ability to reuse task envelopes again and again, rather than recreating them each time. The process features enables independent development of complex and multiple applications. This provides protection for my application since it is broken down in to individually protected processes.

24. Click **Finish**

Step 6: Run the Application on the Target OS

Talking-Point: Now that MapuSoft's products have been generated for my application, I'm ready to run my legacy VxWorks application on Linux.

1. Type "**cd /media/share/simple_vxworks_app**"
2. Hit **enter**
3. Type "**make clean all ROOT_DIR=\$PWD**"
4. Hit **enter**
5. Type "**./simple_vxworks_app_out**"
6. Hit **enter**
7. Press **Ctrl+ C** to stop the application

Talking-Point: Now my VxWorks application is running on Linux. If I wished to port this application to a different OS, I would only need to repeat the code generation steps and choose a different OS. This provides true cross-OS development.

Chapter 6 – Profiler Demo

This chapter contains the following topics:

- Open the Profiler

- Load the Profiler Data File

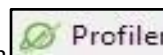
- Show the Graphs for the Profiler Data

Talking-Point: Now that the Profiler data is generated, I can view it in the AppCOE Profiler perspective.

Note: Before showing the profiler data, you need to browse to the file in the root directory in the Linux virtual machine, and copy it over to your shared folder.

Step 1: Open the Profiler

1. Click on the AppCOE Profiler perspective button



Step 2: Load the Profiler Data File

1. Click **Tools**
2. Click **Load Profiler Data File**
3. Browse to the .pal file in **C:\virtual_box**
4. Click **Open**

Step 3: Show the Graphs for the Profiler Data

1. Expand the Profiler data all the way down to the API level by clicking the plus button next to the levels
2. Select an API under **Platform APIs**

Talking-Point: The Profiler provides me with data concerning utilization of MapuSoft's APIs in my application for performance impact analysis.

3. Right click on the **Profiler Data Explorer** tab to view graph options

Talking-Point: The Profiler provides many different graph and chart options.

4. Select an API listed under **Application Functions**

Talking-Point: It also shows data concerning the functions in my application to analyze and identify bottlenecks which are slowing down my application.

5. Right click on the **Profiler Data Explorer** tab to view graph options

Talking-Point: The Profiler provides many different graph and chart options for my application functions as well. I can also generate a timing report.

6. Click on the Timing Report icon (sand timer graphic then type in a file name of your choosing and save it to your **virtual_box** folder
7. Browse to the file on your machine and open to present

Chapter 7 – Full Library Package Demo


This chapter contains the following topics:

- Create the Full Library Package

- Show the Full Library Package Created by AppCOE

Talking-Point: AppCOE can also create Full Library Packages to complete the porting and development outside of AppCOE with your own tools and environment.

Step 1: Create the Full Library Package

1. Click on the **Create Full Library Package** button  (double white and blue arrows) on the tool bar
2. Choose the Target OS
3. Click **Next**
4. Choose the OS Changer or OS Abstractor products needed
5. Click **Next**
6. Choose the place you will save the generated package
7. Click **Finish**

Step 2: Show the Full Library Package Created by AppCOE

1. Open the folder created by AppCOE
2. Show the files included in the package
 - a. A sample application
 - b. Libraries containing the Mapusoft products needed to run their application on the new OS

Talking-Point: The Full Library Package includes libraries with full source code to manually link into the applications. Once the application is recompiled with MapuSoft's products it will run on the new OS.

Chapter 8 – Ada-C/C++ Changer Demo

This chapter contains the following topics:

- Use Ada-C/C++ Changer

- Build the new C/C++ code

- Run the new C/C++ code

Talking-Point: I will demonstrate converting Ada code to C/C++ using the Ada-C/C++ Changer and then run the C/C++ code in AppCOE.

Step 1: Use Ada-C/C++ Changer

1. Right Click the AppCOE & Select **New > Ada-C/C++ Changer Project**
2. Set a project name and select the project Name : Ada-C/C++ Scheduling (or) Real-time OS Abstractor Scheduling

Note:

- Ada C/C++ Changer Projects created with Ada-C/C++ Scheduling will not include OS Abstractor Features
 - Ada C/C++ Changer Projects with the Real-time OS Abstractor Scheduling will include OS Abstractor Features.
 - If **Real-time OS Abstractor Scheduling** option is selected then proceed to **Next Step**
 - If **Ada-C/C++ Scheduling option** is selected then **Skip Step 3.**
3. On Basic Settings window, define the basic properties of your Ada C/C++ Changer project and click **Next**
 4. On Select Configurations window, select the platforms and configurations for deployment and click **Next**
 5. Browse the Ada Source Directory folder and select the Ada Sample project
 6. In the **Main Ada procedure** box, type Sudoku_Test
 7. Click **Next**
 8. Show the different configuration tabs
 9. Click **Next**

Note:

- If **Real-time OS Abstractor Scheduling** option is selected then proceed to **Next Step**
- If **Ada-C/C++ Scheduling option** is selected then **Skip Step10.**

10. If desired, set the resultant C/C++ to integrate with one of the MapuSoft interfaces

Talking-Point: By integrating the new C/C++ code with OS Abstractor, the application will have better performance, stability and portability.

11. Click **Finish**

Step 2: Build the new C/C++ code

1. Select the new project that was created in AppCOE
2. Right click, choose **Build**
3. Click **Continue**
4. Click **OK**

Step 3: Run the new C/C++ code

1. Right click on the project
2. Select **Run As**
3. Select **the Local C/C++ Application**
4. Show the code running in the console

Talking-Point: Here is my newly converted C/C++ application running in the AppCOE simulator.

Revision History

Document Title: AppCOE Quick Start Guide

Release Number: 1.8.1

Release	Revision	Orig. of Change	Description of Change
1.3.5	0.1	VV	<ul style="list-style-type: none"> Added revision history Changed the version numbers for the document
1.3.6	0.1	VV	<ul style="list-style-type: none"> Changed the Release Number
1.3.7	0.1	VV	<ul style="list-style-type: none"> Changed the Release Number
1.3.8	0.1	VV	<ul style="list-style-type: none"> Changed the Release Number
1.3.9	0.1	VV	<ul style="list-style-type: none"> Changes to the Ada-C/C++ Changer project creation
1.3.9.1	0.1	VV	<ul style="list-style-type: none"> Changed the Release Number
1.3.9.2	0.1	VV	<ul style="list-style-type: none"> Changed the Release Number
1.4	0.1	VV	<ul style="list-style-type: none"> Changed the Release Number
1.5	0.1	VV	<ul style="list-style-type: none"> Changed the Release Number
1.6	0.1	VV	<ul style="list-style-type: none"> Integrated FreeRTOS Interface Integrated μC/OS Interface Ada new release (Adacgen 4.038)
1.7	0.1	VV	<ul style="list-style-type: none"> Integrated VRTX Interface
1.8.0	0.1	VV	<ul style="list-style-type: none"> Bug fixes done on the previous release. AppCOE has been updated to Eclipse IDE and Installed Features version Mars.2 (4.5.2) for all operating systems. The Java Runtime Environment (JRE) has been updated to version 1.8 for all operating systems.
1.8.1	0.1	VV	<ul style="list-style-type: none"> Bugs from the previous release resolved. A new project type of "AppCOE" is added for code

			<p>generation on Windows and Linux targets.</p> <ul style="list-style-type: none"> • Installer replaced with a new, more advanced, installer which provides better usability, performance, and features.
--	--	--	---

© Copyright 2021 MapuSoft Technologies, Inc. - All Rights Reserved

The information contained herein is subject to change without notice. The materials located on the Mapusoft. ("MapuSoft") web site are protected by copyright, trademark and other forms of proprietary rights and are owned or controlled by MapuSoft or the party credited as the provider of the information.

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 document for advertisement, or print the information contained herein, 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.

This product includes the software with the following trademarks:

MS-DOS is a trademark of Microsoft Corporation.

UNIX is a trademark of X/Open.

IBM PC is a trademark of International Business Machines, Inc.

Nucleus PLUS and Nucleus NET are registered trademarks of Mentor Graphics Corporation.

Linux is a registered trademark of Linus Torvald.

VxWorks and pSOS are registered trademarks of Wind River Systems.

µC/OS is a registered trademark of Micrium Inc.

FreeRTOS is a registered trademark of Real Time Engineers Ltd.