APPLICATION NOTE
Workload Consolidation

CONCURRENT TECHNOLOGIES

Step by Step Instructions for creating Signed Secure Bootable VxWorks 7 UEFI Boot Loader and Signed Final Images

Contents

Summary

To understand why the associated application note is of value, we must first establish why VxWorks is being used, and the relevance it has within the current embedded systems market. This short publication aims to give a brief overview of VxWorks, UEFI Secure Boot and why they are used in Concurrent Technologies' Embedded Solutions.

Definitions

According to Search Networking, Tech Targets (2007)[i] "VxWorks is a real-time operating system (RTOS) that can be used in embedded systems." Use of the system allows the user to "control network and communication devices, test and measurement equipment, computer peripherals, automotive systems, avionics (aeronautics and astronautics) equipment and diverse consumer products."

As explained by the Microsoft (2019)[ii] "Secure boot is a security standard developed by members of the PC industry to help make sure that a device boots using only software that is trusted by the Original Equipment Manufacturer (OEM)."

How they Work? And why Concurrent Technologies use them?

In an era where security is paramount in many systems, it is intrinsic for users to be able to boot and load software that is guaranteed to be as secure by using a trust-based mechanism. Using UEFI Secure Boot allows the user to have confidence in their systems functioning as intended, without the possibility of any tampering or altering of the operating system and associated drivers. UEFI Secure Boot ensures successful boot up by verifying each stage – this ensures that only the correct Software is loaded and run so that malicious code cannot be executed at this stage of the boot process. This is one of the key characteristics about Secure Boot, and the main reason Concurrent Technologies choose to incorporate it into their embedded systems.

VxWorks, as previously mentioned is a real-time operating system, this means users can control their application in the correct time frame. Operating systems are categorized based on their latency; the lower the latency a system allows, the better it is for real-time operation and the more effective and efficient it is for typical applications within the defense industry. Latency, in layman's term, is how responsive a system is and for VxWorks the interrupt latency is typically around 100µs.

Other operating systems, such as Windows, may appear real-time due to their perceived speedy response times to a user request, however the latency is likely to be in the order of ms rather than µs. In addition, due to the fact the user is not in control of what is happening/processing when using Windows, there are inconsistencies in response time that are a barrier to it being used for real-time solutions. Applications within the defense industry require a deterministic response, so that processes complete within a consistent time frame. Using VxWorks ensures applications and products are working within tight latency in a deterministic fashion to enable complete control of the operating system for the user.

Advantages and Limitations

As with all applications, there are advantages and limitations that must be considered when using RTOS and UEFI BIOS'.

VxWorks:

- ✓ The use of RTOS allows maximum consumption of the system – The user can keep all devices active, with a large resource output
- ✓ RTOS are mostly, if not always error free – This means the user can have confidence of fewer system crashes or other foreseeable issues that could affect their application
- ✓ Finally, the use of an RTOS allows the user to focus on the application. This is because no 'unknown' background tasks are executed and so complete focus falls to the current task. Less action or management is required and exact results can be given on the current work execution
- X Ease of application – RTOS are known for their ease of access when it comes to installation. Due to the desire for proficiency in programming there is an often-difficult process involved during set up
- X Following in a similar path to the previous disadvantage is the complex algorithms and codes required for use. A desired outcome is a result of complexity in designing algorithms and precise codes. This is an obvious disadvantage for a user with less experience or technical knowledge
- X Finally, the systems, due to their expert application are more expensive. Expense in resources required to run, and even more expensive with regards to the knowledge required to set up and use them.[iii]

UEFI Secure Boot:

- ✓ One obvious advantage is the use of Secure Boot itself. This ensures no tampering or external corruption of the operating system due to the advance security the system holds
- ✓ A simplified boot process allows shorter OS load times. An increased load time is often crucial in many situations and an obvious advantage of Secure Boot
- ✓ Finally, a secondary partition is stored in another location. Meaning in the event of an unforeseeable crash, you are ensured that your partition table can be retrieved and recovered without any secondary corruption
- X One key disadvantage is there is an overhead to a system that doesn't really need it. One example would be 32-bit pointer-sectors for partitions that only need to load an operating system
- X Finally UEFI still doesn't fix one of the problems of our old BIOS/MBR setups. We still have to re-probe for devices once the operating system loads[iv]

1 - The Application Note

The application note, entitled: Step by Step Instructions for creating signed Secure Bootable VxWorks 7 UEFI Boot Loader and signed final images, provides the reader with an overview of how a user can create a signed Secure Bootable VxWorks 7 UEFI Boot Loader and signed VxWorks final images. It is important to note, that the application note has been written under the pretenses that the user is familiar with configuring, building and creating VxWorks images and they are doing so on a device that already has the Concurrent Technologies supplied Board Support Package. The application note does however touch upon how a user can include the Security package, if they have not already installed it. A final thing to remember is that the signature created in this example are those created by the WindRiver Workstation Software. It is possible to use one's own signature list, however this is beyond the scope of the application note.

2 - Initial Requirements

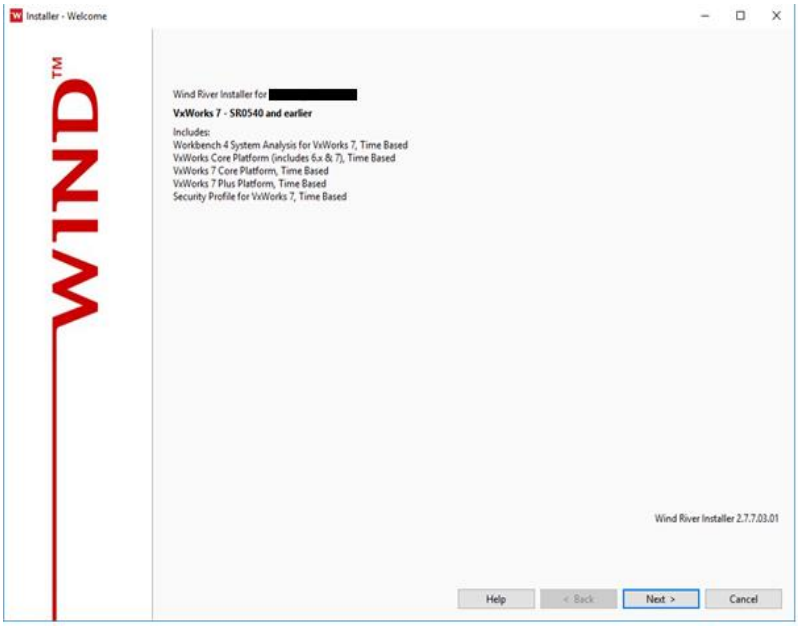
This procedure was performed using the following equipment and software:

- Concurrent Technologies target processor board – TRE5x/msd running BIOS version 4.10.01 used in this example
- Suitable 3U VPX Development Chassis
- USB hub and USB keyboard and USB key drive FAT32 formatted – Kingston Traveler 32GB G4 used in this example
- Display Port compatible display
- A development PC running Windows 10 for running WindRiver Workbench 4
- VxWorks 7 SR0510 base software installed on the above development PC
- Relevant Concurrent Technologies VxWorks 7 BSP installed on development PC – vxw7_tre5_x64_bsp_2_01_07 used in this example

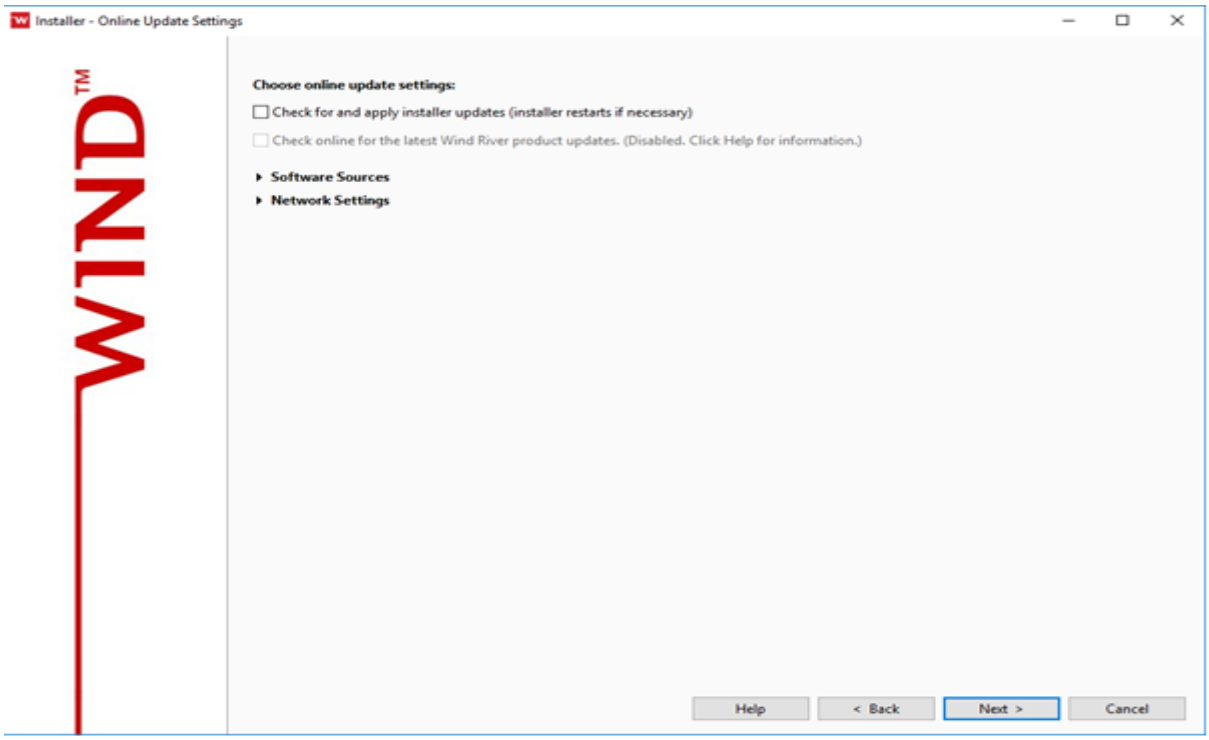
3 - VxWorks 7 Network Security and Secure Loader Package install and configuration

If the user has already installed the Network Security and Secure Loader packages (part of the Security Profile) then they can skip this section and move to Chapter 4.

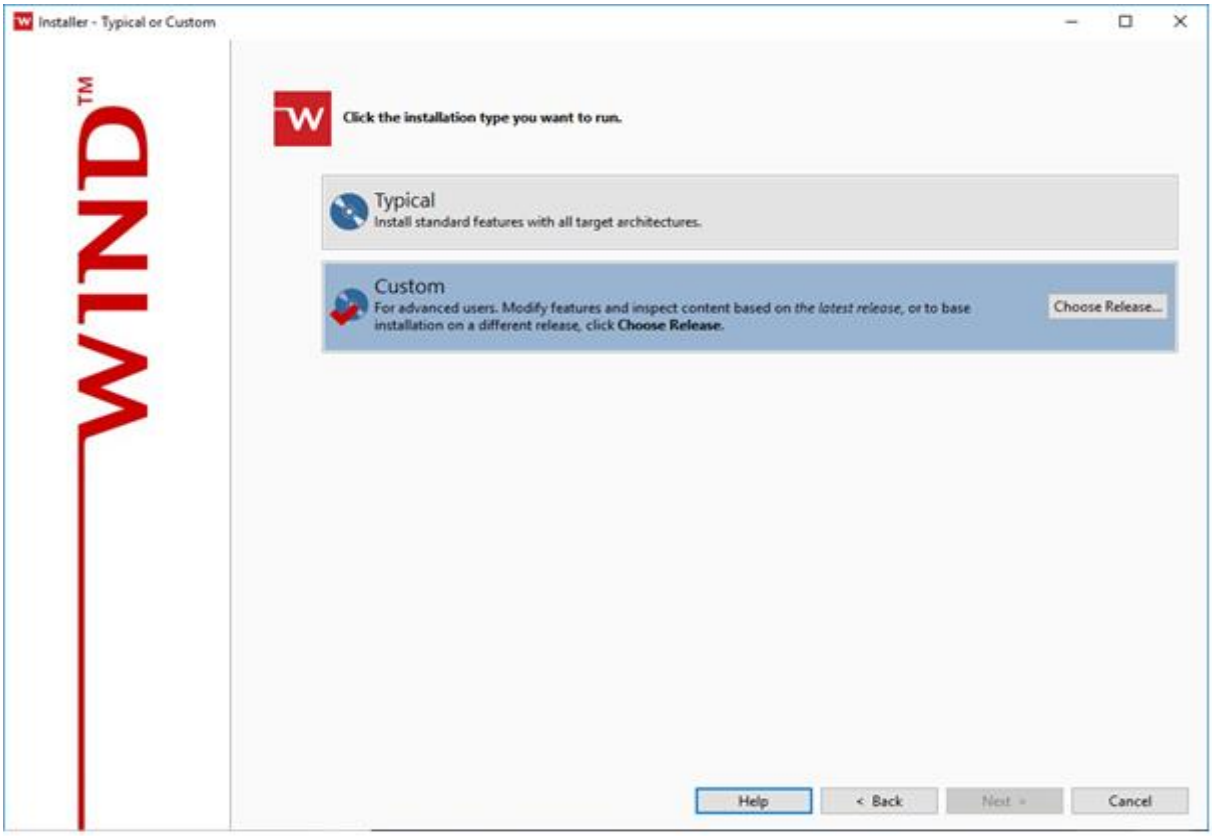
Otherwise, run the VxWorks 7 installer and select 'Next': -



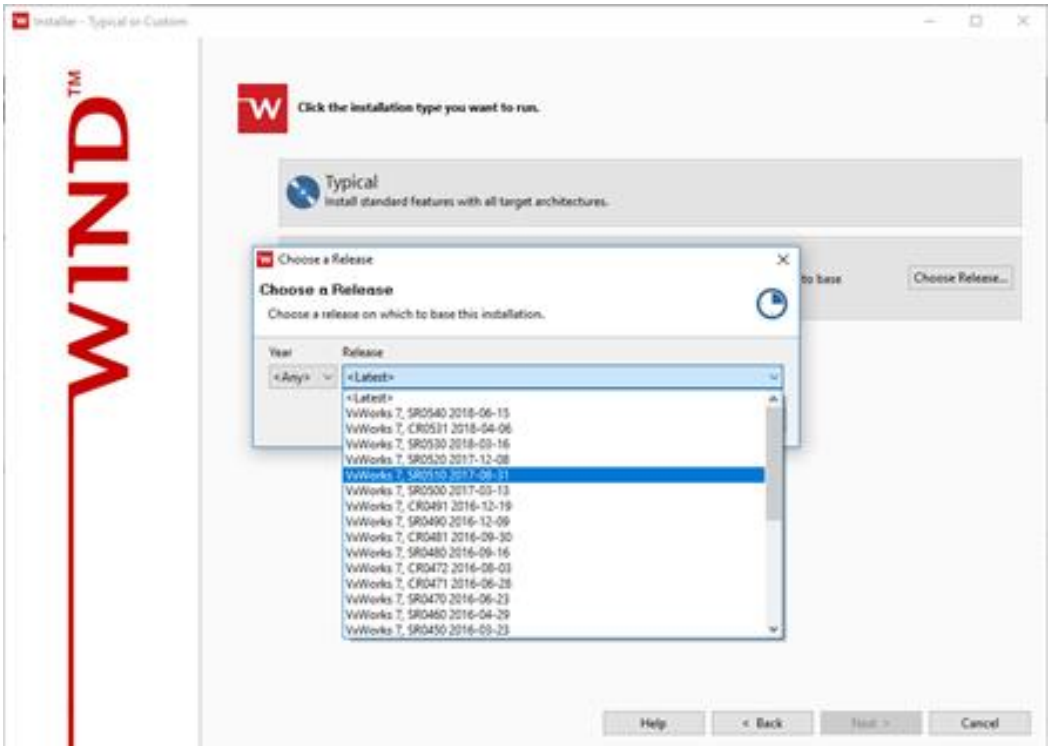
Un-tick the ‘Check for and apply...’ box and select ‘Next’: -



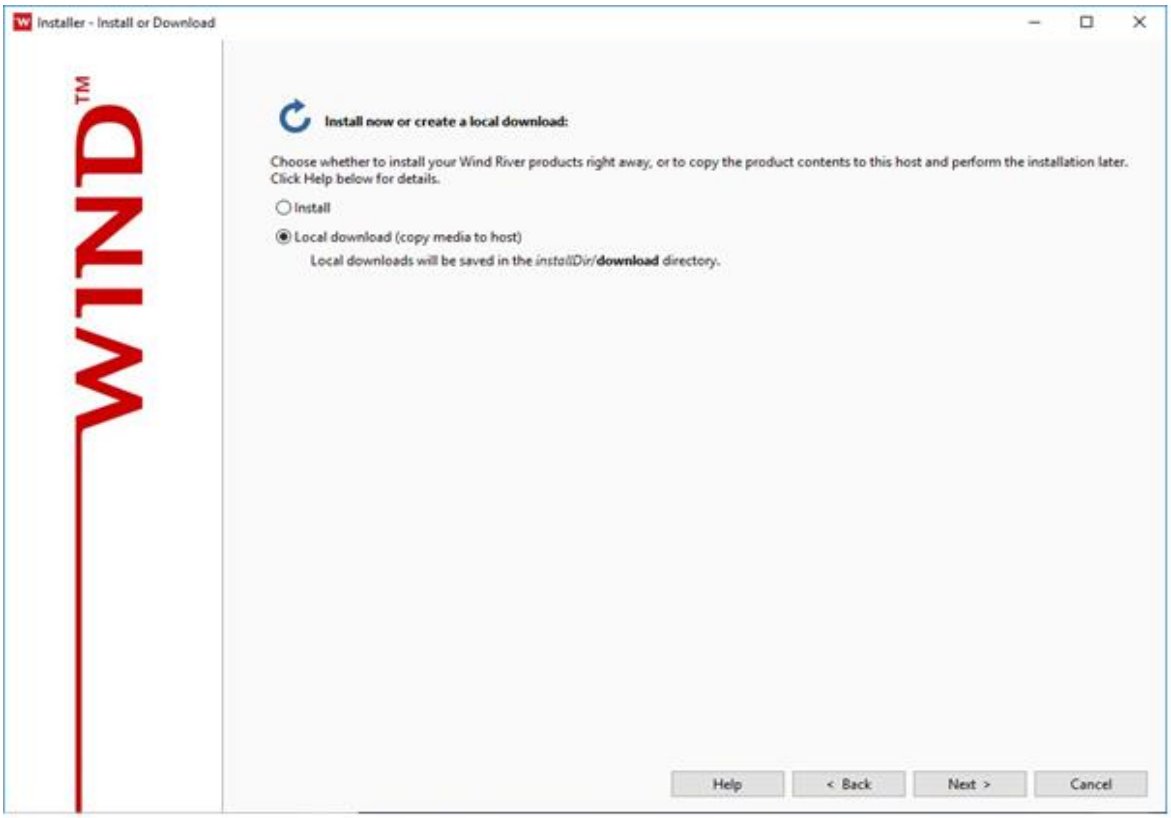
Select Custom and Choose Release, which will display a list: -



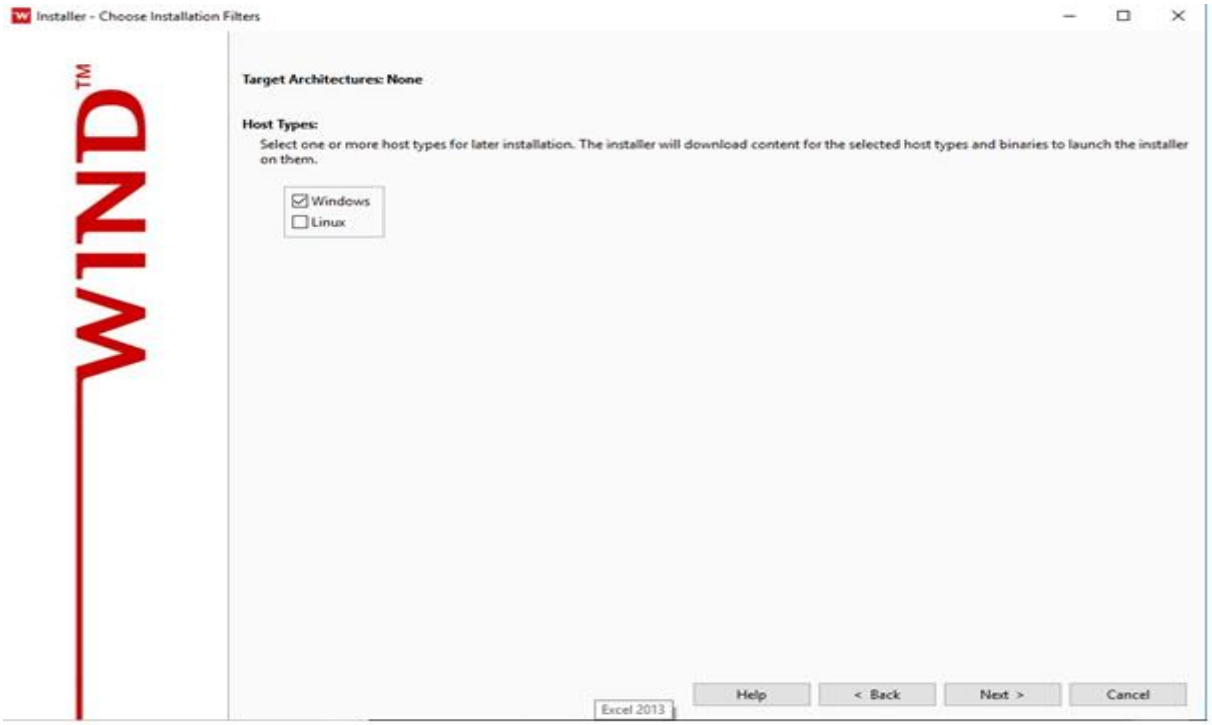
Select the required version, in this case VxWorks 7 SR0510 followed by 'Next': -



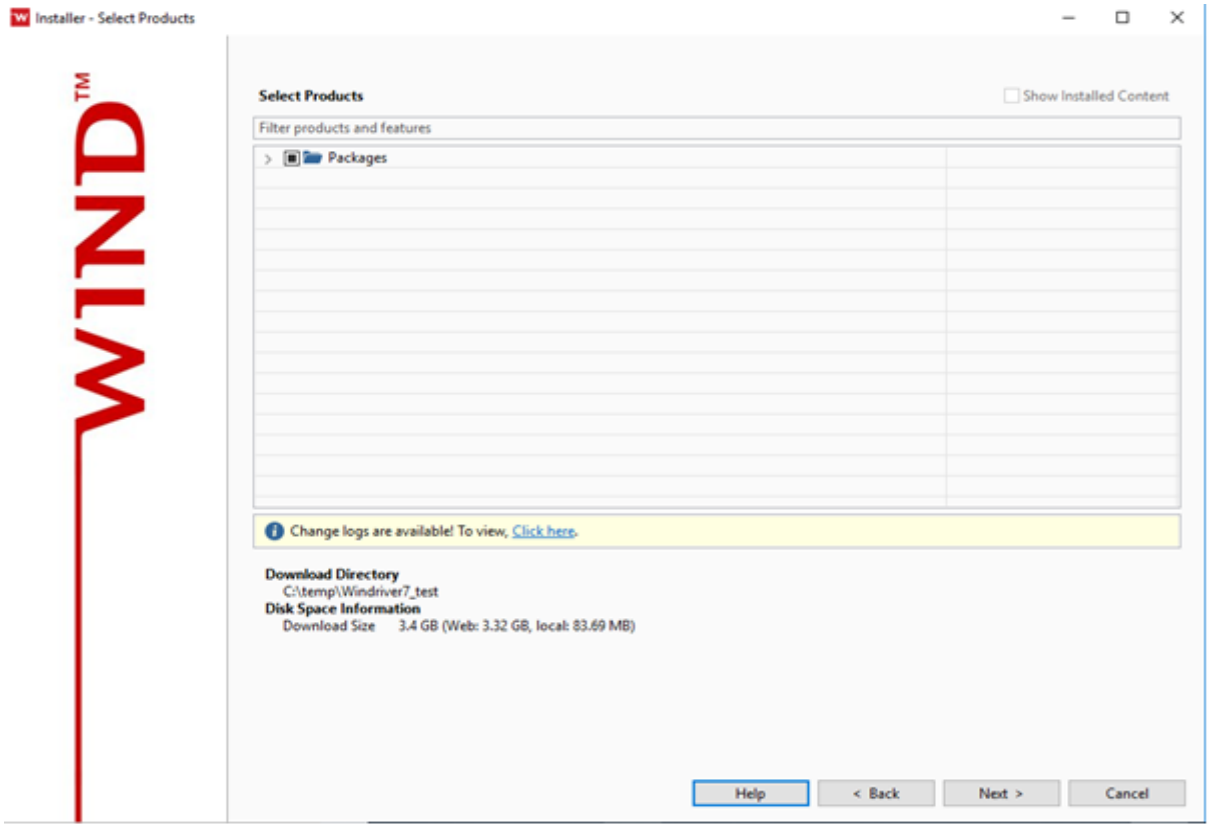
Then click on the install method – Local download in this case, followed by 'Next': -



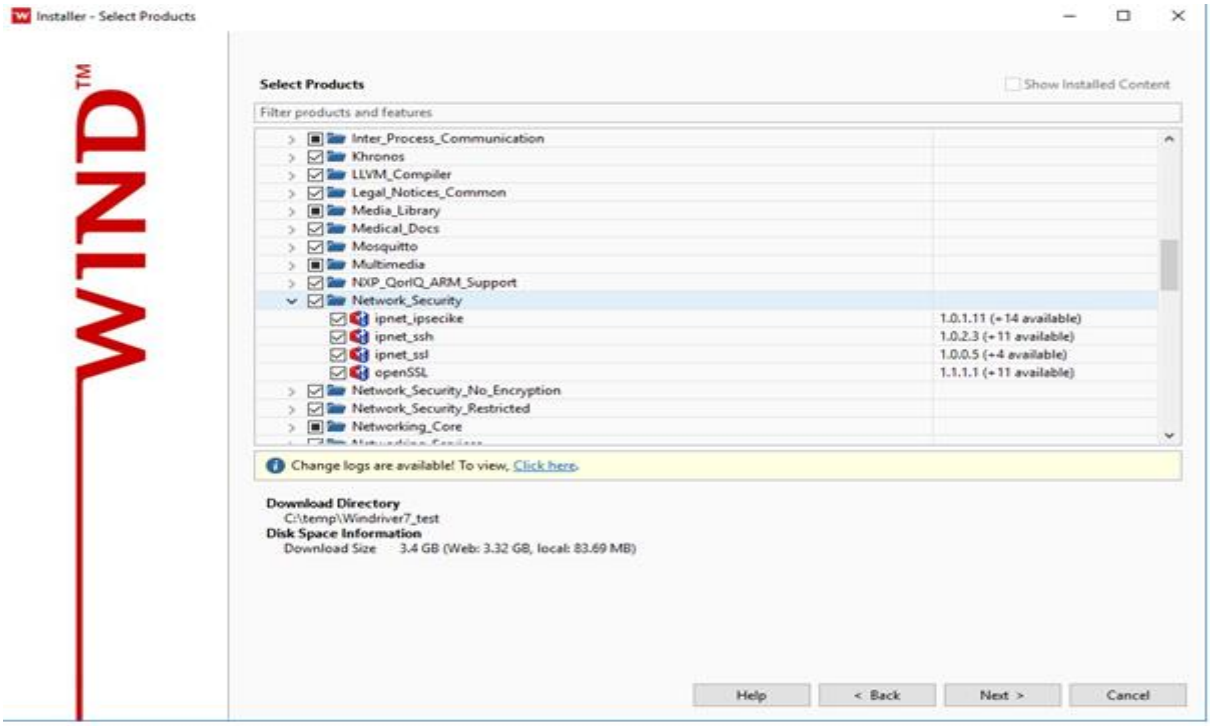
Then select the host type, Windows in this case followed by 'Next': -



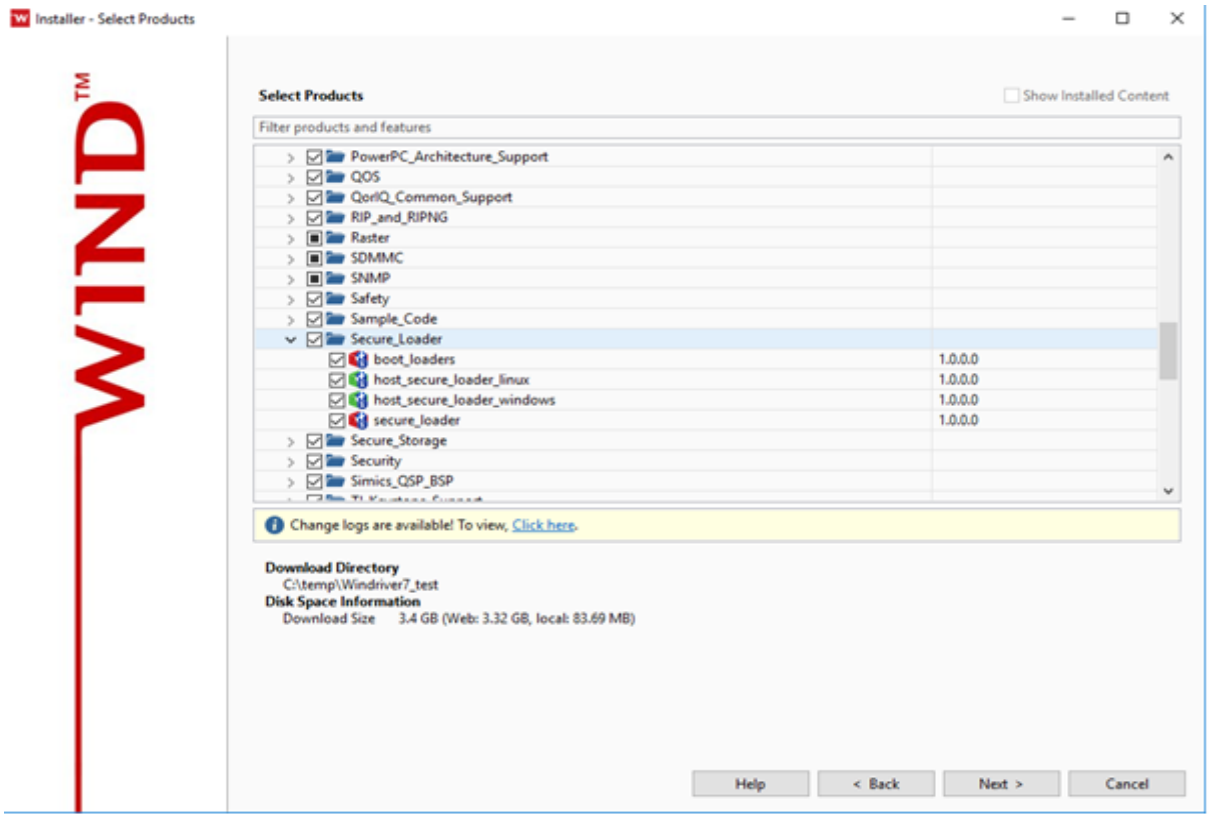
Then click on Packages to bring up the list of installed/available Packages: -



Then select the Network Security Package: -

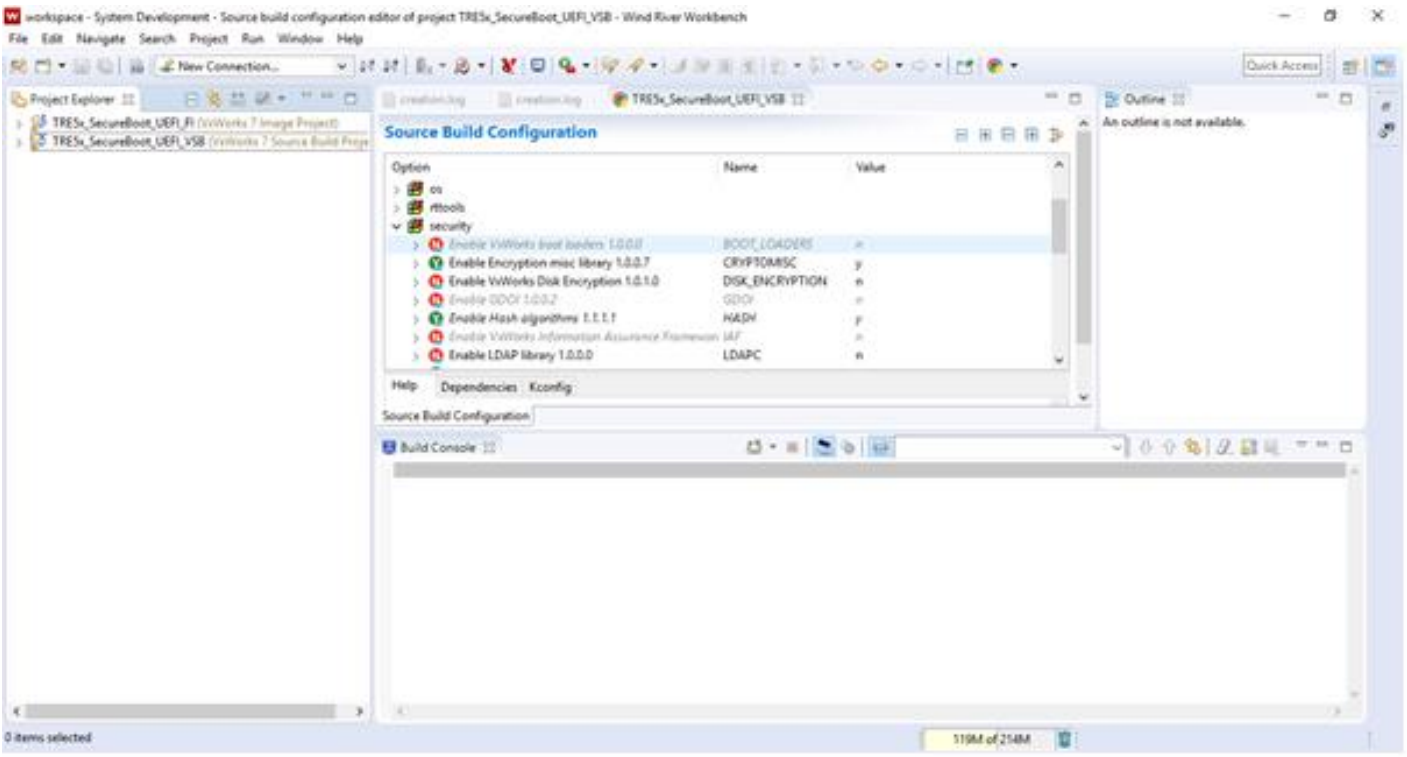


Then select the Secure Loader Package followed by Next. This will then install the required two required packages: -

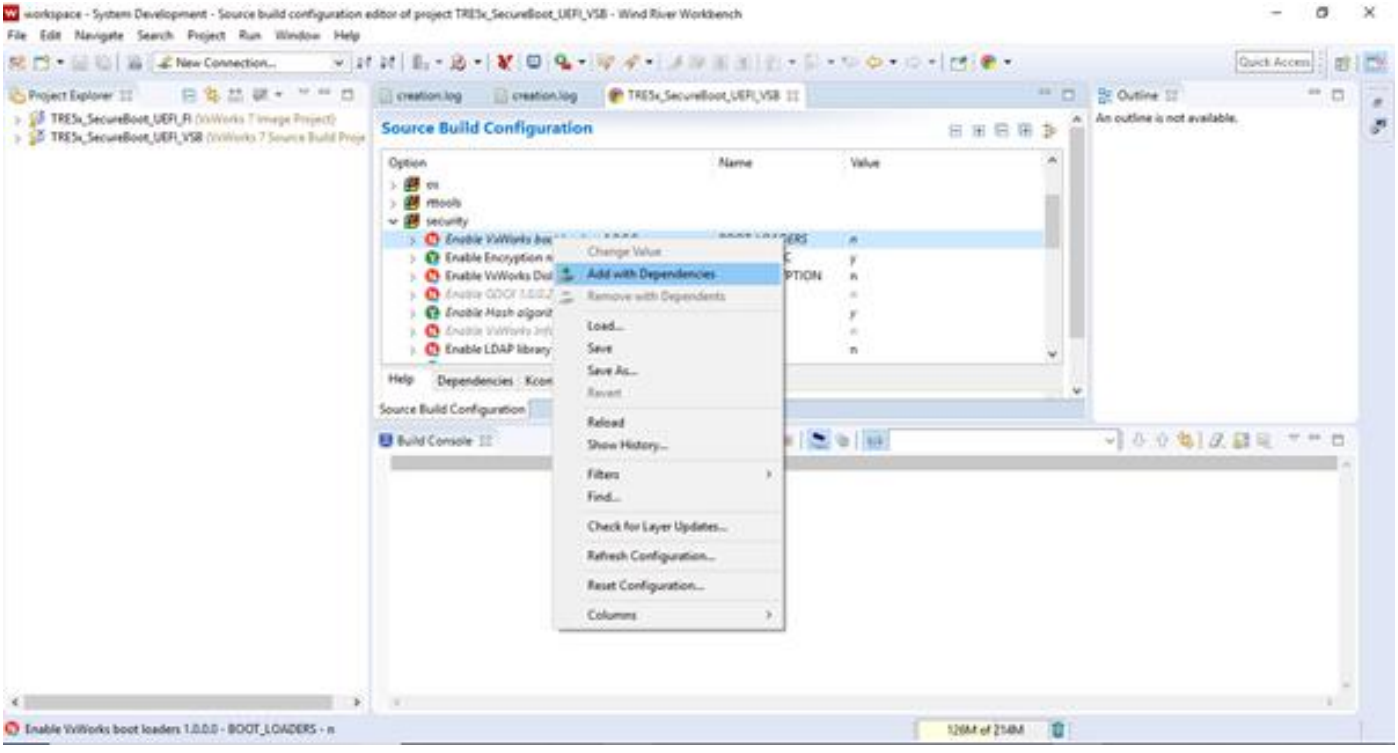


4 - VxWorks 7 VSB and Final Image configuration and build

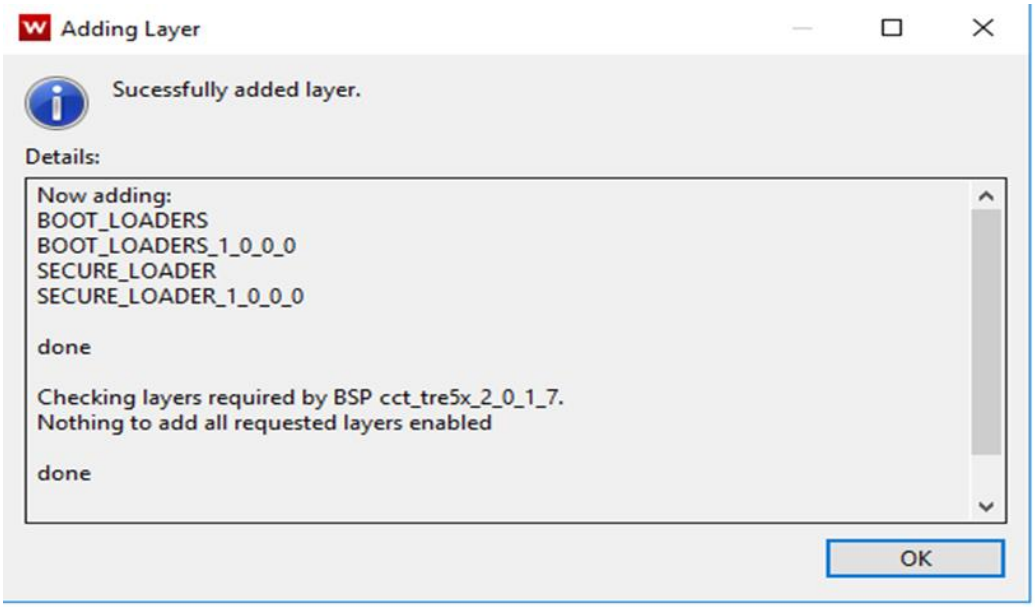
Start up the WindRiver Workbench 4.0. Select the correct project VSB and double click on the Source Build Configuration. Click on the Security Option: -



Then right click on the Enable VxWorks Boot Loaders 1.0.0.0 line and select Add with Dependencies: -



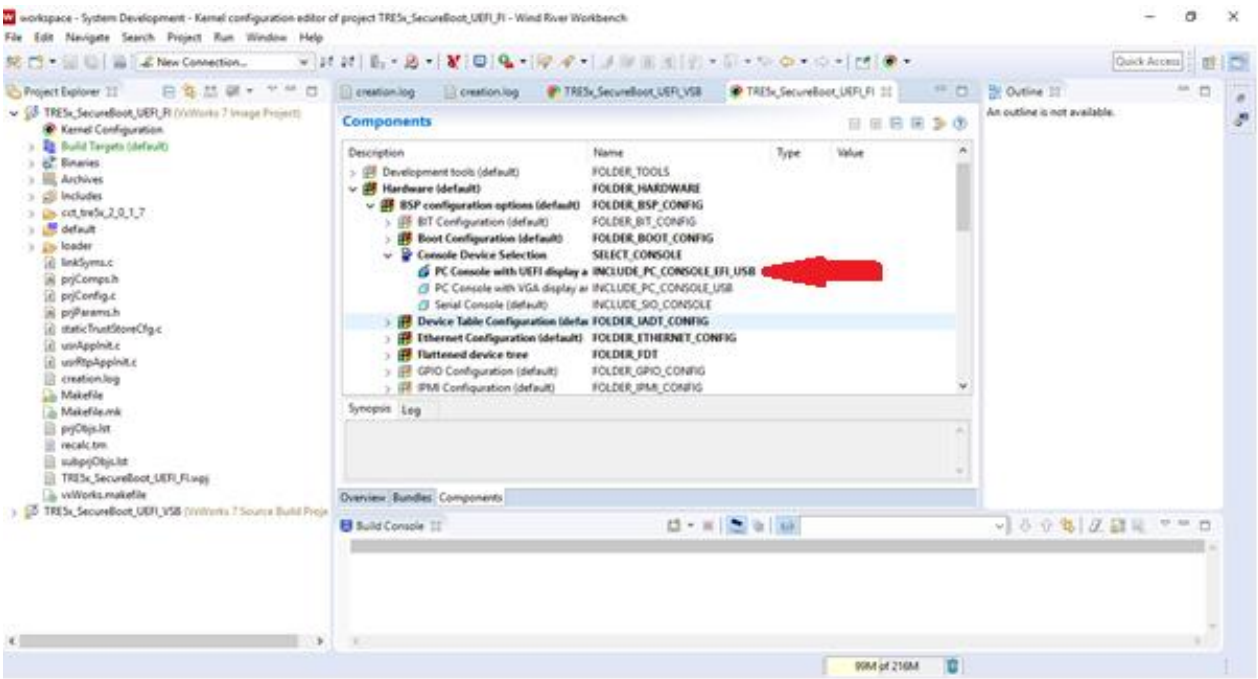
This will bring up an Adding Layer window which will show the layers being added. Click OK to add: -



Repeat for Enable openSSL library 1.1.1.1 if not already added.

Then build the VSB project.

When the build is complete, open the Final Image project and open the Kernel Configuration file. Open the Hardware → Console Device Selection tab in the Components window and right click on PC Console with UEFI display: -



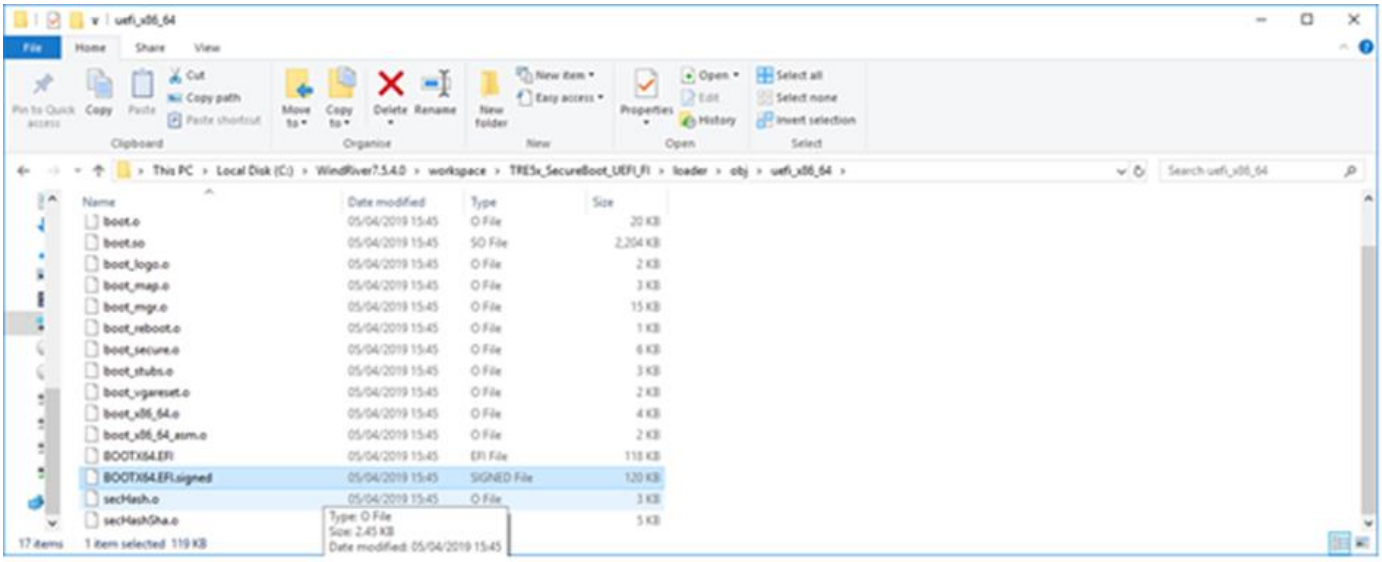
Build the Final Image project. This will produce a signed VxWorks image which can be copied on to the USB drive.

5 - Bootable USB drive creation

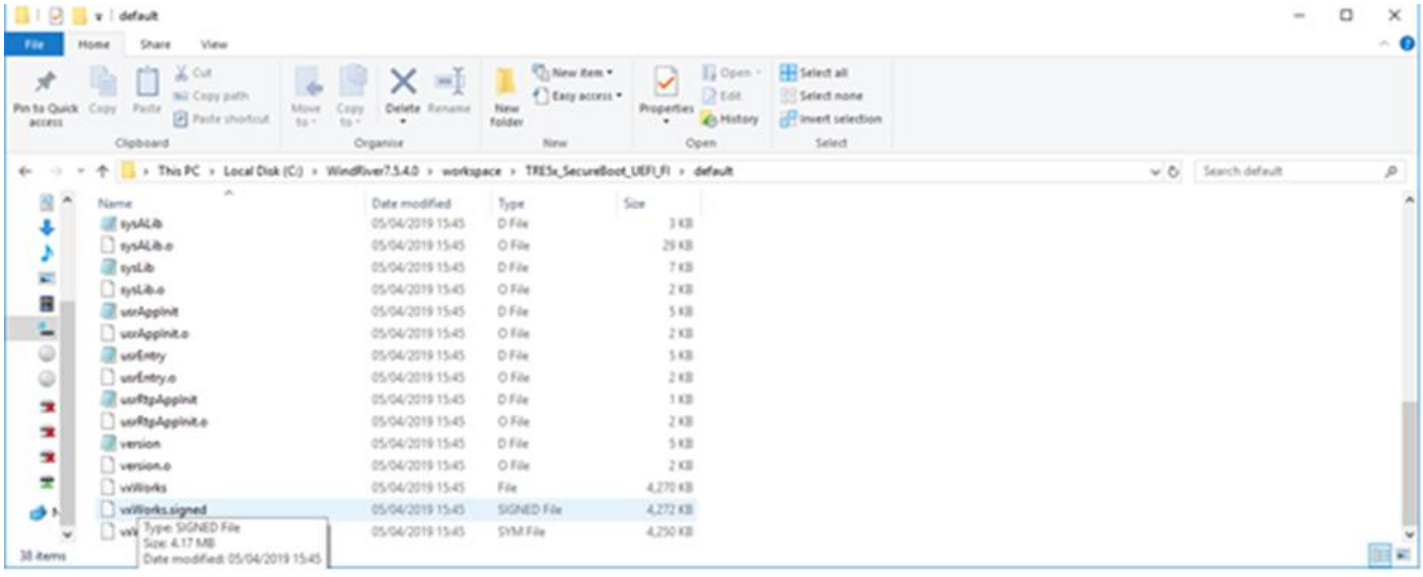
Firstly, create the following directory structure on the FAT32 formatted USB drive: -

/EFI/BOOT

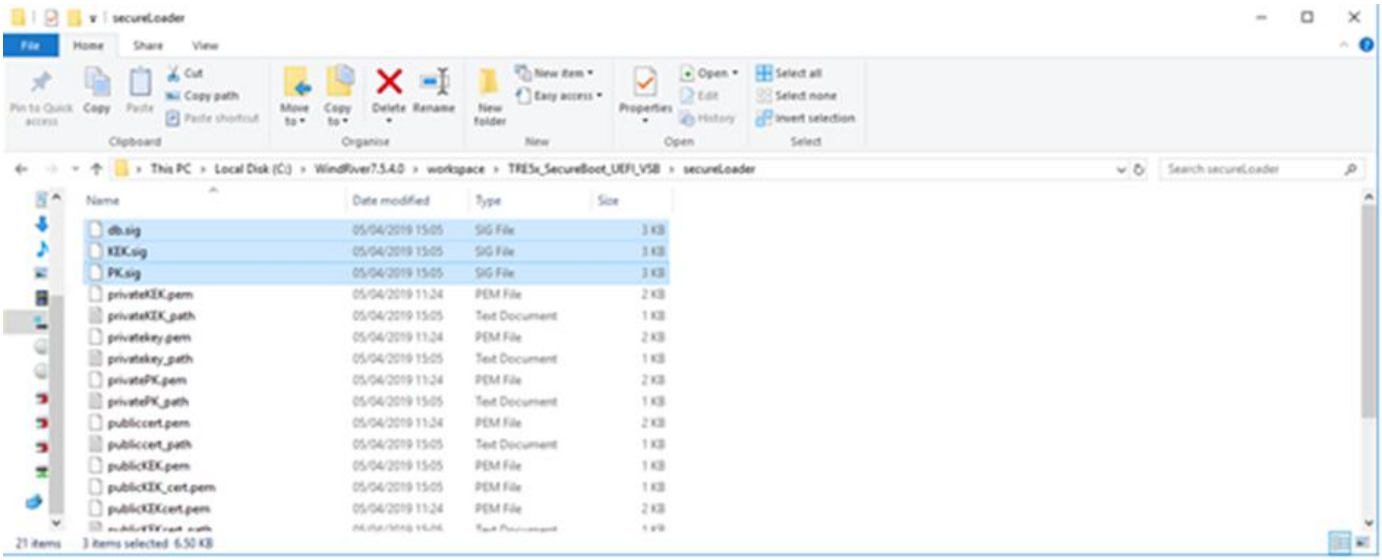
Copy the VxWorks UEFI Boot Loader into the /EFI/BOOT directory on the USB drive. The file BOOTX64.EFI.signed needs to be renamed as BOOTX64.EFI. Here is the location under the workspace directory of the Final Image project (…/loader/obj/uefi_x86_64): -



Copy the signed VxWorks Final Image onto the USB drive in the /EFI/BOOT directory. The file 'vxWorks.signed' needs to be renamed as BOOTAPP.SYS. Also copy the unsigned VxWorks image into the same directory. Both files reside under the workspace directory of the Final Image default directory: -



Then copy the 3 signature files (PK.sig, KEK.sig and DB.sig) into the root directory (/) of the USB drive. These files were created when the VSB project was built and reside in that projects work space directory under secureLoader: -

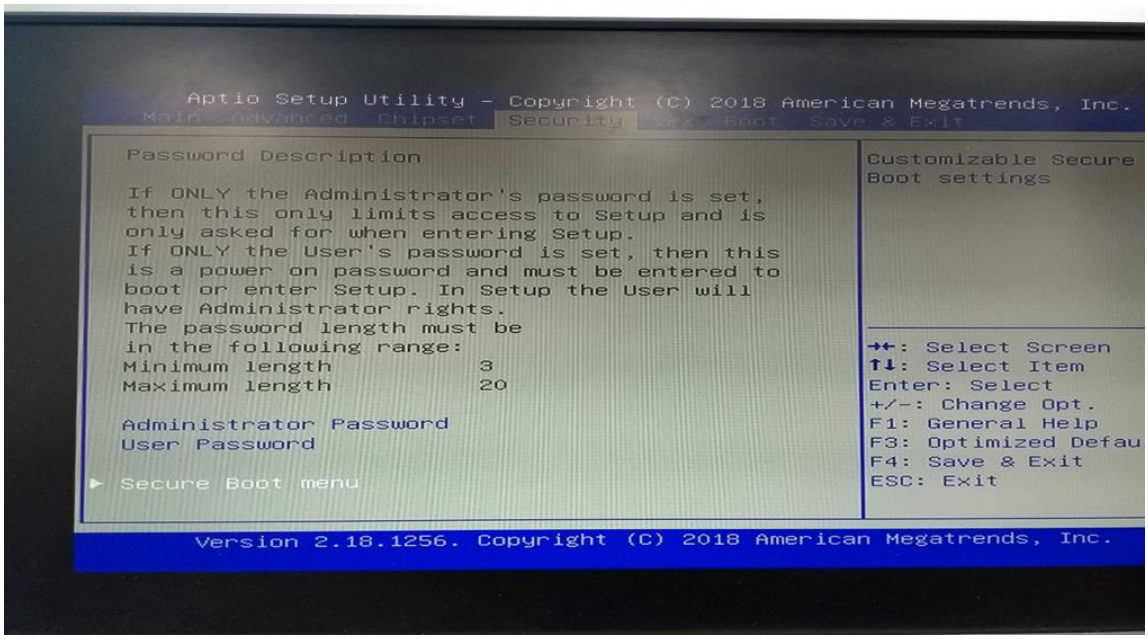


Eject the USB drive from the development system and insert it into the target USB hub.

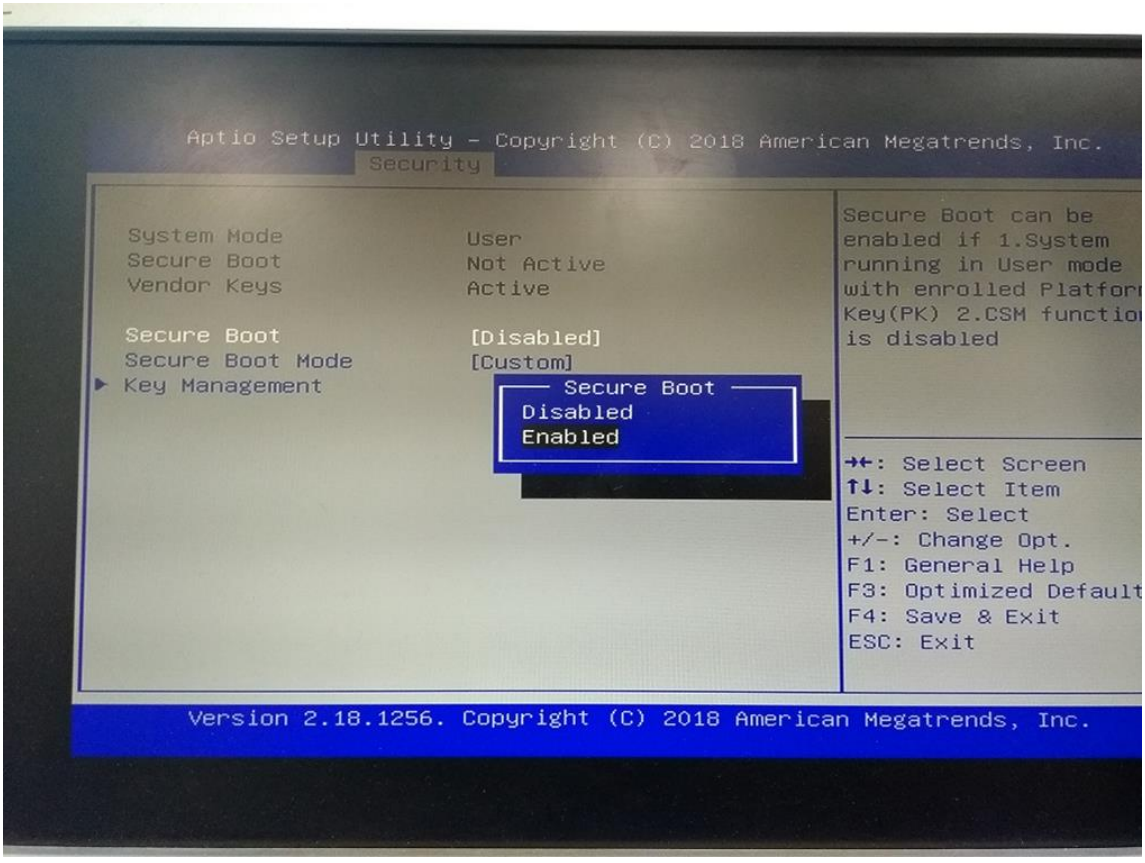
6 - UEFI Firmware configuration and signature loading

Power on the target system. Press F2 to enter the BIOS setup screen.

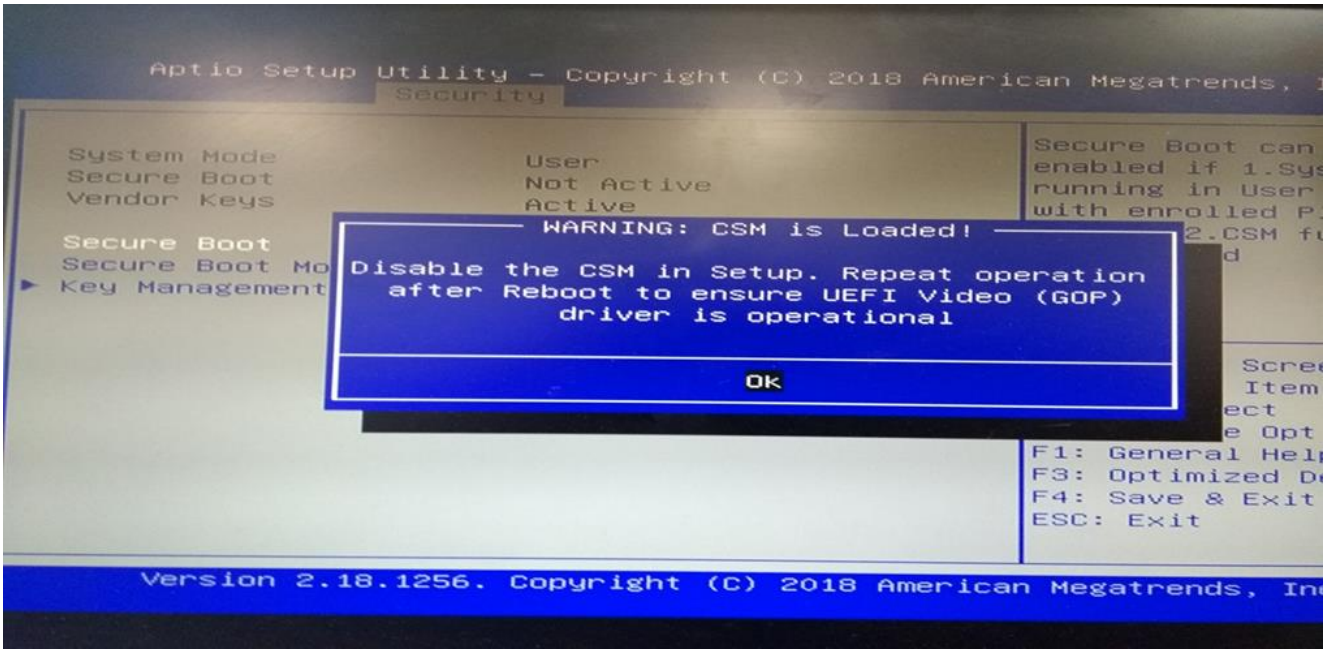
Select the Security menu, followed by Secure Boot menu: -



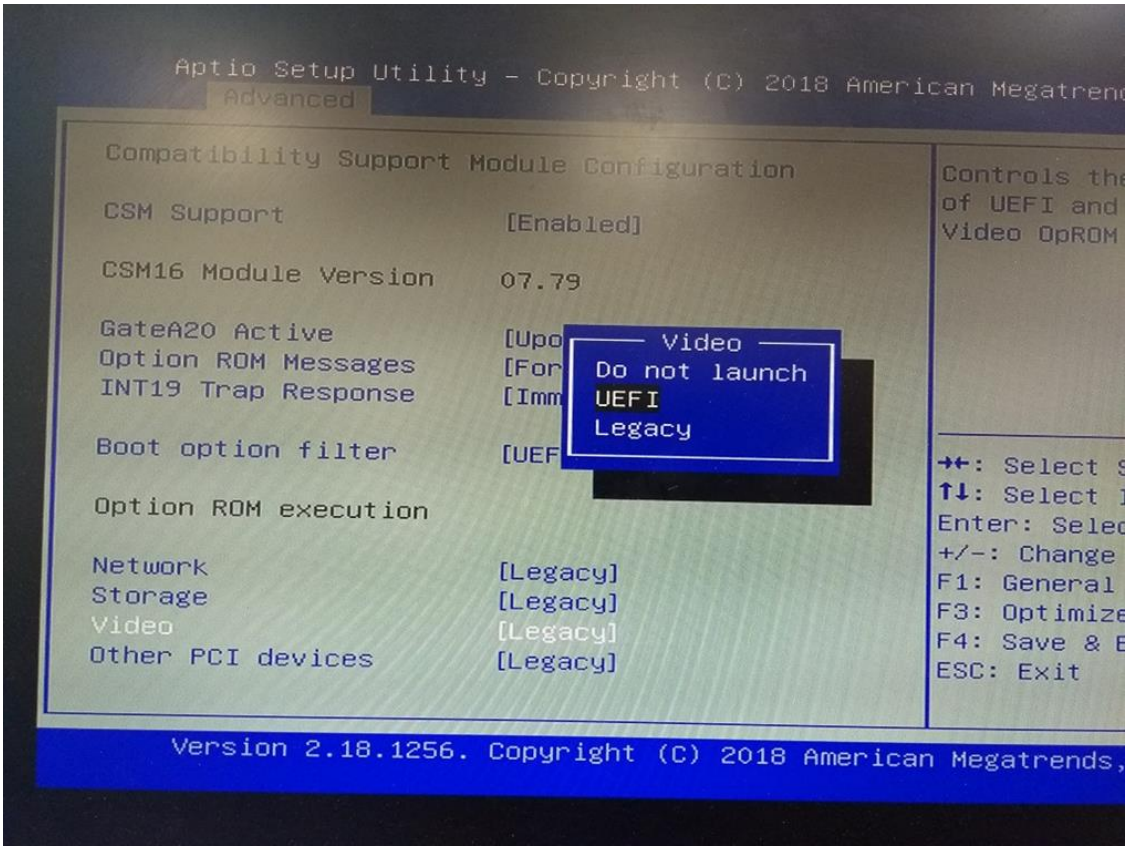
Set Secure Boot to Enabled: -



This will bring up a warning dialogue box requesting that CSM be disabled: -



Press return, followed by Escape to return to the top-level menu. Then select Advanced, followed by CSM Configuration. Select Video under Option ROM execution sub-menu and select UEFI: -

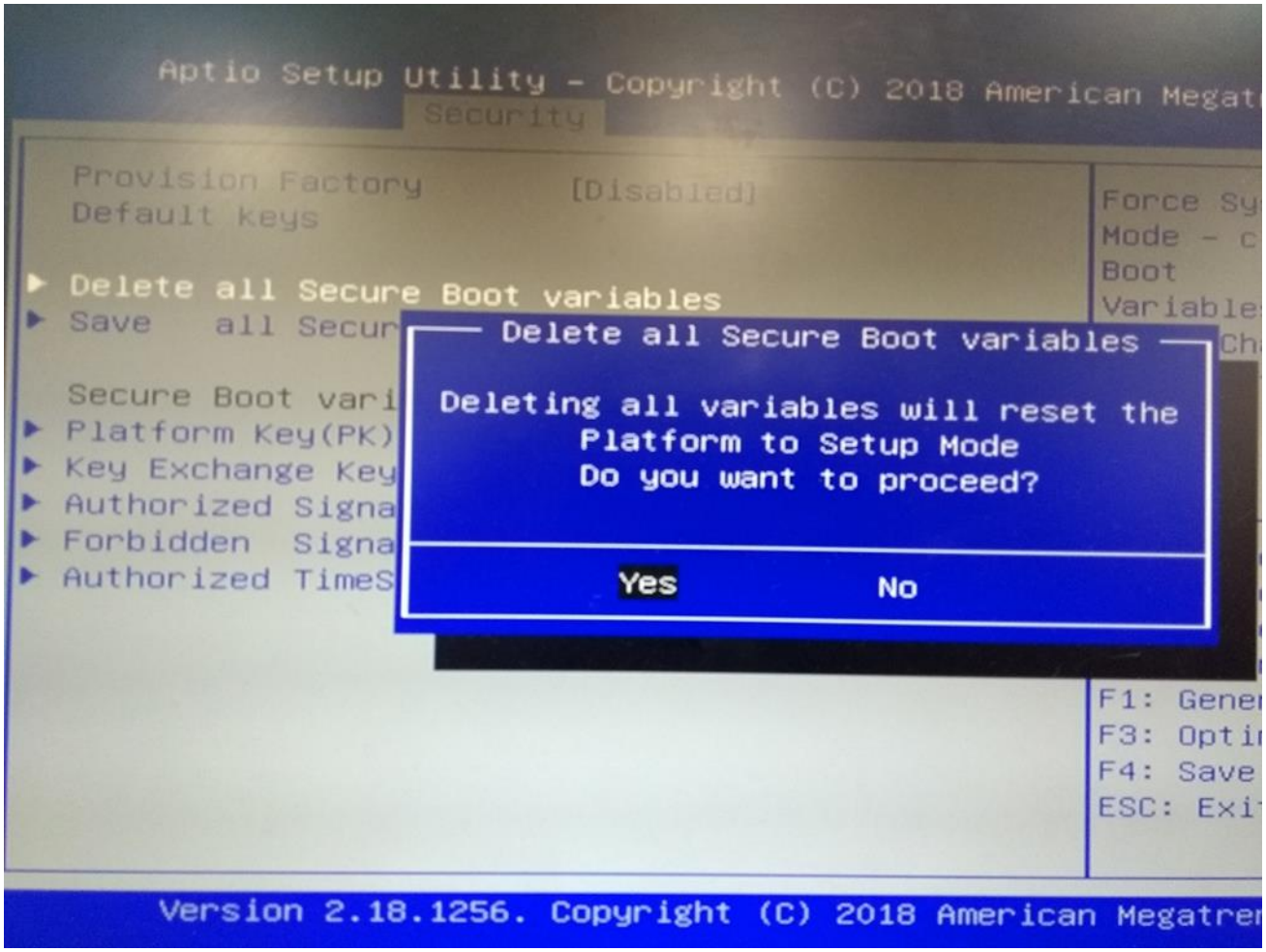


Then select CSM Support and set to Disabled.

Press F4 to save configuration and reset board.

Press F2 to re-enter setup menu. Select Security followed by Secure Boot, followed by Key Management.

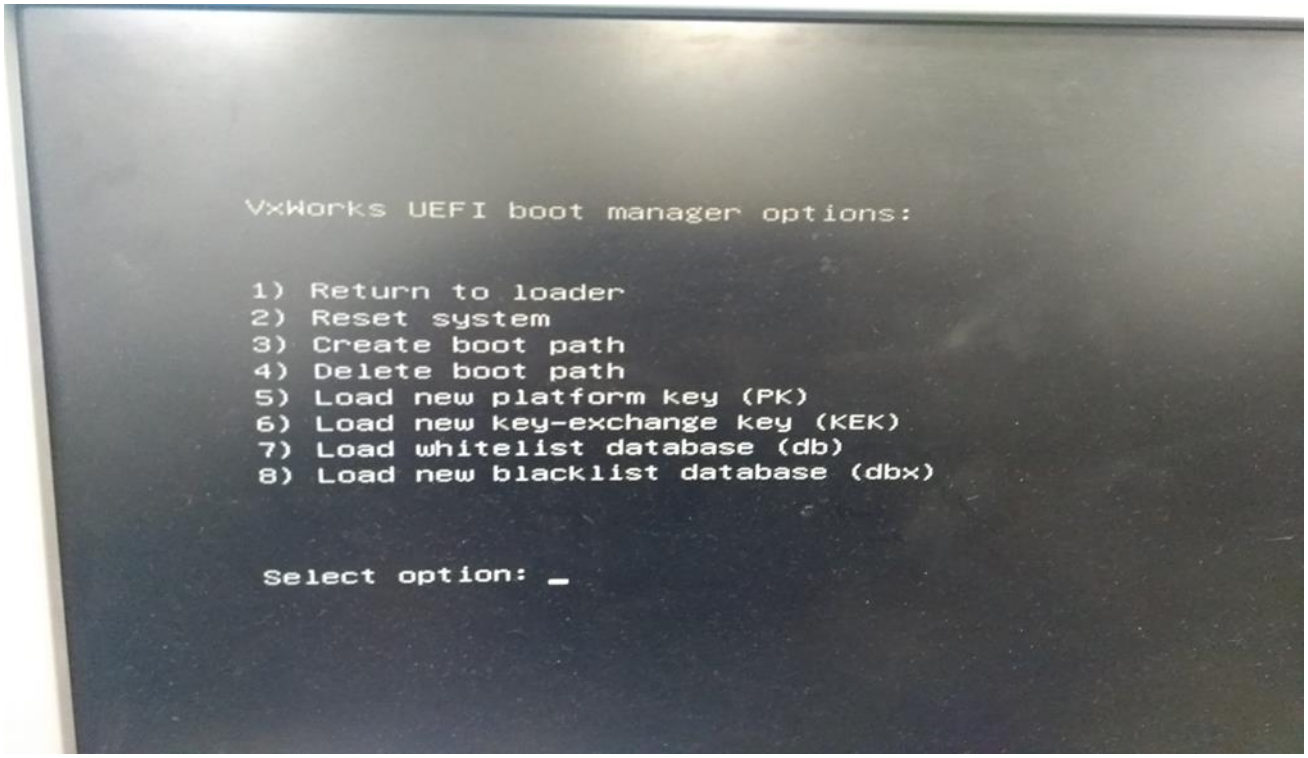
Under Key Management menu select Delete all Secure Boot variables and select Yes. This clears the factory default keys PK, KEK and DB variables: -



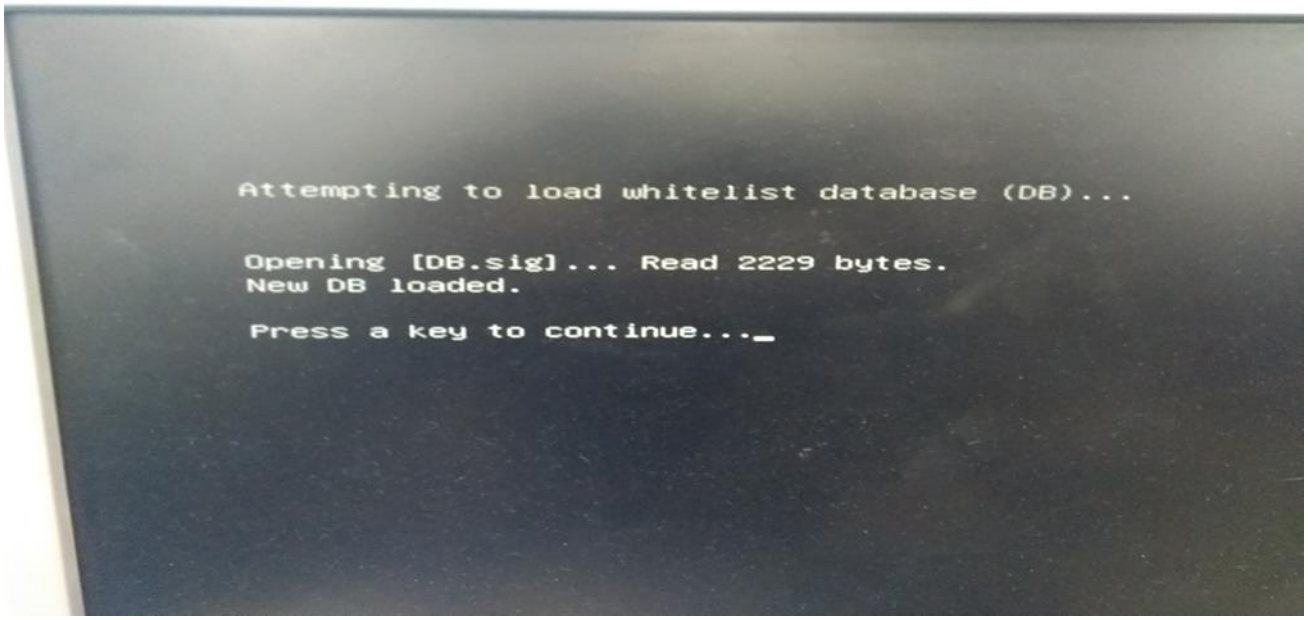
Press Escape to return to the Secure Boot menu. Now need to disable Secure Boot so that the UEFI Boot Manager can be started and the user signatures added to the firmware.

Press F4 to save configuration and reboot board.

Allow the system to reboot and start the VxWorks UEFI Boot Manager. Press any key to stop the Boot Loader process. Then press 'M' to enter the Boot Manager menu: -



To load the whitelist DB UEFI signature list into the firmware select option 7:-



Press any key to return to the menu. Repeat the process for KEK UEFI signature list by selecting option 6. Then repeat the process again for PK UEFI signature list by selecting option 5.

Press 2 to reset the system.

Press F2 to re-enter the Setup menu.

Select Security and Secure Boot menu.

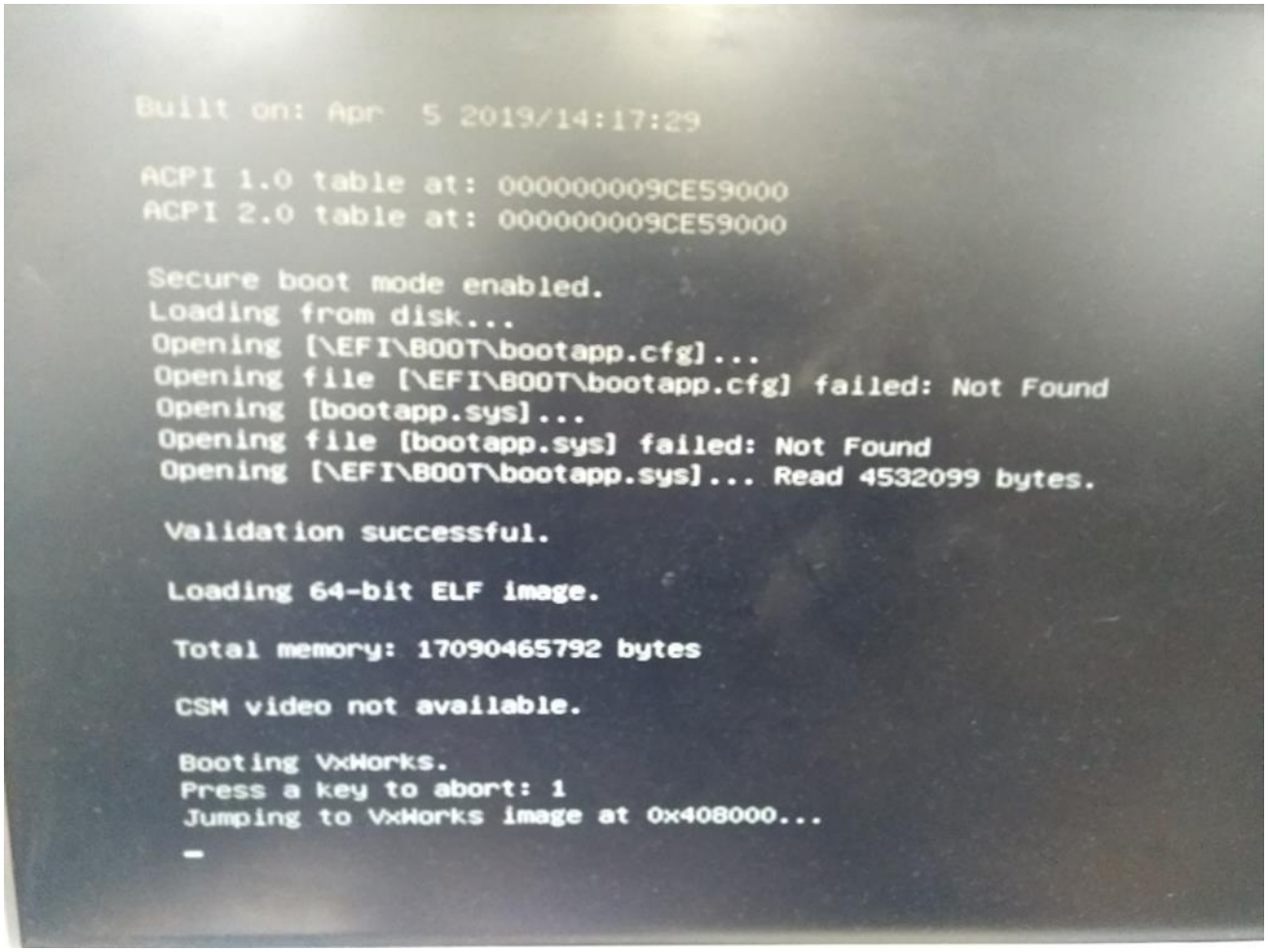
Enable Secure Boot.

7 - Signed and unsigned VxWorks image testing

This chapter tests the booting of signed and unsigned VxWorks images.

Press F4 to save the configuration and reboot the system.

The UEFI firmware loads and starts the signed VxWorks UEFI boot loader. Then the VxWorks UEFI boot loader loads and starts the signed VxWorks image: -



To test that an unsigned image is not loaded by the VxWorks UEFI loader rename the required files.

Press Control-X to reboot the system. Press F2 to enter setup menu. Select Security followed by Secure

Boot. Disable Secure Boot to re-enable the UEFI Shell. Press F4 to save and reboot.

Press F12 to enter the Boot Manager. Select the UEFI Shell.

At the UEFI Shell prompt type: -

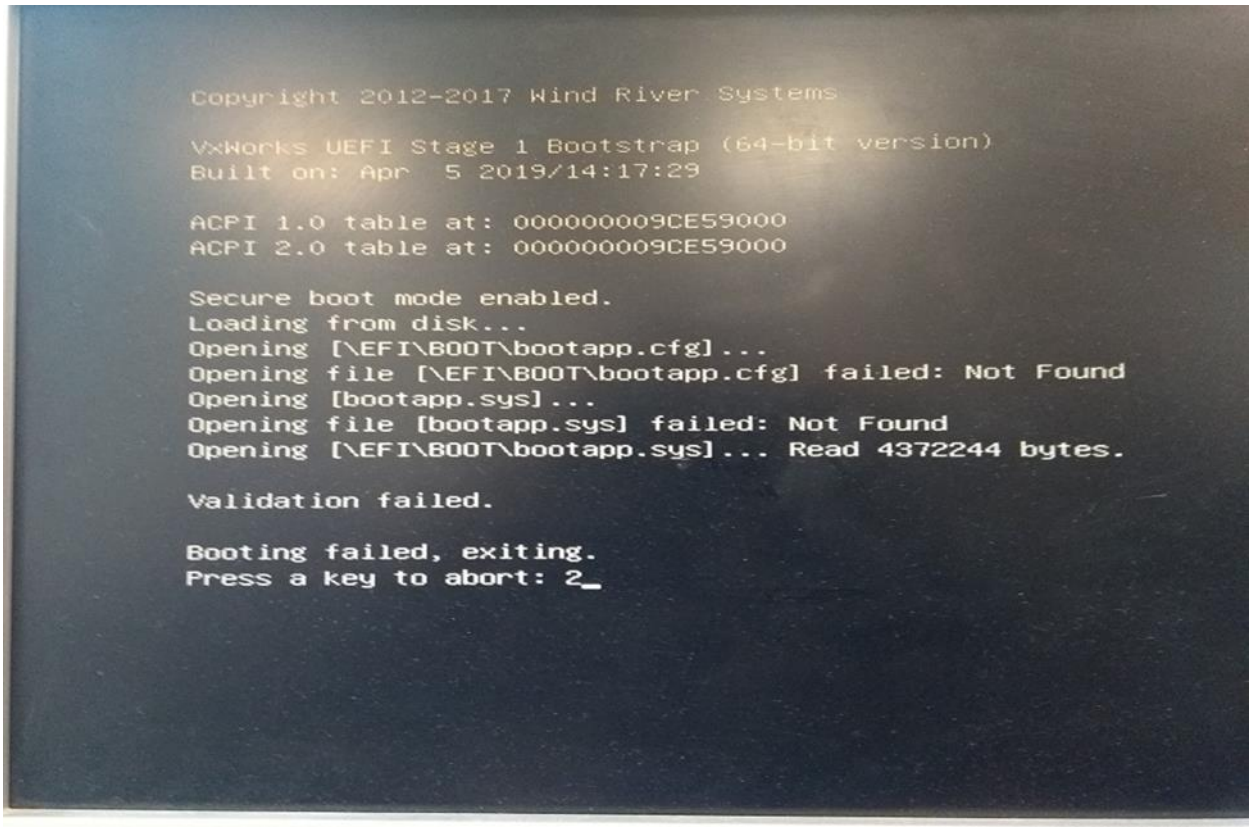
FS0:

mv \EFI\BOOT\BOOTAPP.SYS \EFI\BOOT\vxWorks.signed

mv \EFI\BOOT\vxWorks \EFI\BOOT\BOOTAPP.SYS

Then reset the system by typing reset.

Press F2 to enter setup menu. Select Security followed by Secure Boot. Enable Secure Boot to disable the UEFI Shell. The system reboots and attempts to load the unsigned image: -



The message 'Validation failed' and 'Booting failed, exiting' are output. The boot process is repeated, and on the second failure returns to the UEFI setup menu, which would normally be password protected.

For additional information, please visit http://www.gocct.com

Concurrent Technologies Plc.
4 Gilberd Court, Newcomen Way
Colchester, Essex CO4 9WN
U.K.
Tel: (+44) 1206 752626
support@cct.co.uk

NOTE:

Information furnished by Concurrent Technologies is believed to be accurate and reliable. However, Concurrent Technologies assumes no responsibility for any errors contained in this document and makes no commitment to update or to keep current the information contained in this document. Concurrent Technologies reserves the right to change specifications at any time without notice.

Concurrent Technologies assumes no responsibility either for the use of this document or for any infringements of the patent or other rights of third parties which may result from its use. In particular, no license is either granted or implied under any patent or patent rights belonging to Concurrent Technologies.

[i] *The definition of VxWorks can be found here: https://searchnetworking.techtarget.com/definition/VxWorks*

[ii] *The definition of Secure boot can be found here: https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/oem-secure-boot*

[iii] *Advantages and Disadvantages of using RTOS as explained by ITRelease (2014): http://www.itrelease.com/2014/07/advantages-disadvantages-real-time-operating-systems/*

[iv] *Advantages and Disadvantages of Secure Boot as explained by Phoenix TS (2016): https://phoenixts.com/blog/booting-uefi-mode/*