

---

# MVME2500 VxWorks 6.8

User Guide

P/N: 6806800L66D

September 2019

---



**SMART**<sup>™</sup>  
Embedded Computing

© 2019 SMART Embedded Computing™, Inc.

All Rights Reserved.

## Trademarks

The stylized "S" and "SMART" is a registered trademark of SMART Modular Technologies, Inc. and "SMART Embedded Computing" and the SMART Embedded Computing logo are trademarks of SMART Modular Technologies, Inc. All other names and logos referred to are trade names, trademarks, or registered trademarks of their respective owners. These materials are provided by SMART Embedded Computing as a service to its customers and may be used for informational purposes only.

## Disclaimer\*

SMART Embedded Computing (SMART EC) assumes no responsibility for errors or omissions in these materials. **These materials are provided "AS IS" without warranty of any kind, either expressed or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.** SMART EC further does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SMART EC shall not be liable for any special, indirect, incidental, or consequential damages, including without limitation, lost revenues or lost profits, which may result from the use of these materials. SMART EC may make changes to these materials, or to the products described therein, at any time without notice. SMART EC makes no commitment to update the information contained within these materials.

Electronic versions of this material may be read online, downloaded for personal use, or referenced in another document as a URL to a SMART EC website. The text itself may not be published commercially in print or electronic form, edited, translated, or otherwise altered without the permission of SMART EC.

It is possible that this publication may contain reference to or information about SMART EC products, programming, or services that are not available in your country. Such references or information must not be construed to mean that SMART EC intends to announce such SMART EC products, programming, or services in your country.

## Limited and Restricted Rights Legend

If the documentation contained herein is supplied, directly or indirectly, to the U.S. Government, the following notice shall apply unless otherwise agreed to in writing by SMART Embedded Computing.

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data clause at DFARS 252.227-7013 (Nov. 1995) and of the Rights in Noncommercial Computer Software and Documentation clause at DFARS 252.227-7014 (Jun. 1995).

**SMART Embedded Computing, Inc.**

2900 S. Diablo Way, Suite 190

Tempe, Arizona 85282

USA

\*For full legal terms and conditions, visit [www.smartembedded.com/lecl/legal](http://www.smartembedded.com/lecl/legal)

# Table of Contents

---

- About this Manual** ..... **5**
- 1.1 Overview ..... 9
- 1.2 Deliverables ..... 9
- 2.1 Introduction ..... 11
- 2.2 Network Boot ..... 11
  - 2.2.1 Prerequisites ..... 11
  - 2.2.2 Booting Procedure ..... 11
- 2.3 Disk Boot ..... 12
  - 2.3.1 Prerequisites ..... 12
  - 2.3.2 Booting Procedure ..... 13
- 2.4 USB Boot ..... 14
  - 2.4.1 Prerequisites ..... 14
  - 2.4.2 Booting Procedure ..... 14
- 2.5 SPI Flash Boot ..... 15
  - 2.5.1 Prerequisites ..... 15
  - 2.5.2 Booting Procedure ..... 15
- 3.1 Building Procedure ..... 19
  
- A Sample Output** ..... **27**
- A.1 Network Boot ..... 27
- A.2 Disk Boot ..... 28
- A.3 USB Boot ..... 29
- A.4 SPI Flash Boot ..... 31

## Table of Contents

---

# About this Manual

---

## Overview of Contents

This manual is divided into the following chapters and appendix:

*Chapter 1, Introduction on page 9* provides an overview of this manual.

*Chapter 2, Booting VxWorks on page 11* describes the procedure to boot VxWorks on the MVME2500.

*Chapter 3, Building Board Support Package on page 19* describes the procedure to build Board Support Package (BSP).

*Appendix A, Sample Output on page 27* provides the sample output of VxWorks booting through network, disk and USB.



## Abbreviations






This document uses the following abbreviations:

| Abbreviation | Definition                            |
|--------------|---------------------------------------|
| BSP          | Board Support Package                 |
| FTP          | File Transfer Protocol                |
| IP           | Internet Protocol                     |
| SATA         | Serial Advanced Technology Attachment |
| SBC          | Single Board Computer                 |
| TFTP         | Trivial File Transfer Protocol        |

# Conventions

The following table describes the conventions used throughout this manual. .

| Notation  | Description   |
|---|---|
| 0x00000000  | Typical notation for hexadecimal numbers (digits are 0 through F), for example used for addresses and offsets   |
| 0b0000  | Same for binary numbers (digits are 0 and 1)  |
| <b>bold</b>   | Used to emphasize a word  |
| Screen  | Used for on-screen output and code related elements or commands.<br>Sample of Programming used in a table (9pt) |
| <b>Courier + Bold</b>   | Used to characterize user input and to separate it from system output   |
| <i>Reference</i>  | Used for references and for table and figure descriptions   |
| File > Exit   | Notation for selecting a submenu  |
| <text>  | Notation for variables and keys   |
| [text]  | Notation for software buttons to click on the screen and parameter description                                  |
| ...   | Repeated item for example node 1, node 2, ..., node 12  |
| .   | Omission of information from example/command that is not necessary at the time                                  |
| ..  | Ranges, for example: 0..4 means one of the integers 0,1,2,3, and 4 (used in registers)                          |
|   | Logical OR  |
|  | Indicates a hazardous situation which, if not avoided, could result in death or serious injury                  |
|  | Indicates a hazardous situation which, if not avoided, may result in minor or moderate injury                   |

| Notation  | Description  |
|---|--|
|                            | Indicates a property damage message  |
|                            | Indicates a hot surface that could result in moderate or serious injury                            |
|                            | Indicates an electrical situation that could result in moderate injury or death                    |
| <p>Use ESD protection</p>  | Indicates that when working in an ESD environment care should be taken to use proper ESD practices |
|                           | No danger encountered, pay attention to important information                                      |

## Summary of Changes

This manual has been revised and replaces all prior editions.

| Part Number | Publication Date | Description  |
|-------------|------------------|--|
| 6806800L66D | September 2019   | Rebranded to SMART Embedded Computing template.  |
| 6806800L66C | February 2014    | Re-branded to Artesyn template. Added SPI Flash Boot on page 16, Appendix A, SPI Flash Boot, on page 31. |
| 6806800L66B | February 2011    | Final Version  |
| 6806800L66A | November 2010    | Preliminary Version  |





# Introduction

---

## 1.1 Overview

The MVME2500 Single Board Computer (SBC) is a VMEbus board, which features a single-core P2010 or the dual-core P2020 NXP® QorIQ® processors.

This document describes the procedure to boot VxWorks 6.8 on the MVME2500 board.

## 1.2 Deliverables

The following table lists the MVME2500 deliverables.

*Table 1-1 List of Deliverables*

| <b>Name</b>         | <b>Description</b>                                   |
|---------------------|--|
| vxWorks-2020.st     | VxWorks boot image file for P2020 blades             |
| vxWorks-2010.st     | VxWorks boot image file for P2010 blades             |
| mvme2500_sp1.tar.gz | VxWorks 6.8 Board Support Package (BSP) for MVME2500 |



# Booting VxWorks

---

## 2.1 Introduction

You can boot VxWorks on the MVME2500 board using any of the following methods:

- Network Boot
- Disk Boot
- USB Boot
- SPI Flash Boot

## 2.2 Network Boot

### 2.2.1 Prerequisites

You should have connectivity to the TFTP server.

### 2.2.2 Booting Procedure

The TFTP server should be configured and started in the connected PC. The VxWorks boot Image file, `vxWorks-2020.st` or `vxWorks-2010.st`, should be made available at the standard TFTP boot image path `/tftpboot`.

To boot VxWorks through network, perform the following steps:

1. Power up the MVME2500 board.  
By default, it provides the U-Boot prompt.
2. Set the environmental variables at the U-Boot prompt.

```
setenv ipaddr <Board ip address>
setenv serverip <TFTP server ip address>
setenv gatewayip <Gateway ip address>
setenv netmask <Netmask>
```

Example:

```
setenv ipaddr 10.130.101.206
setenv serverip 10.130.101.216
setenv gatewayip 10.130.101.254
setenv netmask 255.255.255.0
```

- 
3. Set the VxWorks boot image file name.

```
setenv vxbootfile vxWorks-2020.st
```

```
setenv vxbootfile vxWorks-2010.st
```

4. Set the VxWorks bootline arguments.

```
setenv vxbootargs 'motetsec(0,0)10.130.101.216:vxWorks
```

```
h=10.130.101.216 e=10.130.101.206:ffffff00 u=vxworks pw=vxworks
```

```
f=0x80'
```

Parameters description:

motetsec(0,0) : Ethernet interface 0 on cpu 0

10.130.101.216 : Host Machine IP

10.130.101.206 : Board IP

ffffff00 : Netmask

u=vxworks : Username on host machine

p=vxworks : Password for the above user in host machine

f=0x80 : File Transfer Protocol (FTP)

5. Set the VxWorks network boot command.

```
setenv vxboot 'tftpboot $vxbootfile && setenv bootargs $vxbootargs  
&& bootvx'
```

6. Save your current environmental variables.

```
saveenv
```

7. To boot VxWorks through network, execute the following command:

```
run vxboot
```

For sample output, refer [Section A.1, Network Boot on page 27](#).

## 2.3 Disk Boot

### 2.3.1 Prerequisites

You should have:

- Serial Advanced Technology Attachment (SATA) hard disk with ext2 file system loaded, and
- VxWorks image loaded to the ext2 file system

---

## 2.3.2 Booting Procedure

To boot VxWorks using disk, perform the following steps:

1. Power up the MVME2500 board.  
By default, it provides the U-Boot prompt.
2. Set the environmental variables at the U-Boot prompt.

```
setenv ipaddr <Board ip address>
setenv serverip <TFTP server ip address>
setenv gatewayip <Gateway ip address>
setenv netmask <Netmask>
```

Example:

```
setenv ipaddr 10.130.101.206
setenv serverip 10.130.101.216
setenv gatewayip 10.130.101.254
setenv netmask 255.255.255.0
```

3. Set the VxWorks boot image file name.

```
setenv vxbootfile vxWorks-2020.st
setenv vxbootfile vxWorks-2010.st
```

4. Set the VxWorks bootline arguments.

```
setenv vxbootargs 'motetsec(0,0)10.130.101.216:vxWorks
h=10.130.101.216 e=10.130.101.206:ffffff00 u=vxworks pw=vxworks
f=0x80'
```

Parameters description:

motetsec(0,0) : Ethernet interface 0 on cpu 0  
10.130.101.216 : Host Machine IP  
10.130.101.206 : Board IP  
ffffff00 : Netmask  
u=vxworks : Username on host machine  
p=vxworks : Password for the above user in host machine  
f=0x80 : File Transfer Protocol (FTP)

5. Set the VxWorks disk boot command.

```
setenv vxdiskboot 'ext2load scsi 0:1 0x1000000 $vxbootfile &&
setenv bootargs $vxbootargs && bootvx'
```

6. Save your current environmental variables.

```
saveenv
```

- 
7. To boot VxWorks through hard disk, execute the following command:

```
run vxdiskboot
```

For sample output, refer [Section A.2, Disk Boot on page 28](#).

## 2.4 USB Boot

### 2.4.1 Prerequisites

You should have:

- USB pen drive with VxWorks image, and vfat or ext2fs file system

### 2.4.2 Booting Procedure

To boot VxWorks using USB, perform the following steps:

1. Power up the MVME2500 board.  
By default, it provides the U-Boot prompt.

2. Set the environmental variables.

```
setenv ipaddr <Board IP address>
setenv serverip <TFTP server IP address>
setenv gatewayip <Gateway IP address>
setenv netmask <Netmask>
```

Example:

```
setenv ipaddr 10.130.101.206
setenv serverip 10.130.101.216
setenv gatewayip 10.130.101.254
setenv netmask 255.255.255.0
```

3. Set the VxWorks boot image file name.

```
setenv vxbootfile vxWorks-2020.st
setenv vxbootfile vxWorks-2010.st
```

4. Set the VxWorks bootline arguments.

```
setenv vxbootargs 'motetsec(0,0)10.130.101.216:vxWorks
h=10.130.101.216 e=10.130.101.206:ffffff00 u=vxworks pw=vxworks
f=0x80'
```

---

Parameters description:

`motetsec(0,0)` : Ethernet interface 0 on cpu 0

`10.130.101.216` : Host Machine IP

`10.130.101.206` : Board IP

`ffffff00` : Netmask

`u=vxworks` : Username on host machine

`p=vxworks` : Password for the above user in host machine

`f=0x80` : File Transfer Protocol (FTP)

5. Set the VxWorks USB boot command.

```
setenv vxusbboot 'usb reset && fatload usb 0:1 0x1000000
$vxbootfile && setenv bootargs $vxbootargs && bootvx'
```

6. Save your current environmental variables.

```
saveenv
```

7. To boot VxWorks through USB, execute the following command:

```
run vxusbboot
```

For sample output, refer [Section A.3, USB Boot on page 29](#).

## 2.5 SPI Flash Boot

### 2.5.1 Prerequisites

You should have:

- TFTP server connectivity is required only when you want to copy new VxWorks image to SPI Flash.

### 2.5.2 Booting Procedure

The TFTP server should be configured and started in the connected PC. The VxWorks boot Image file, `vxWorks-2020.st` or `vxWorks-2010.st`, should be made available at the standard TFTP boot image path `/tftpboot`.

Follow 1 to 8 steps to copy VxWorks image from network to SPI Flash.

1. Power up the MVME2500 board.  
By default, it provides the U-Boot prompt.
2. Set the environmental variables at the U-Boot prompt.

```
setenv ipaddr <Board ip address>
```

```
setenv serverip <TFTP server ip address>
```

---

```
setenv gatewayip <Gateway ip address>
setenv netmask <Netmask>
```

Example:

```
setenv ipaddr 10.130.101.206
setenv serverip 10.130.101.216
setenv gatewayip 10.130.101.254
setenv netmask 255.255.255.0
```

3. Set the VxWorks boot image file name.

```
setenv vxbootfile vxWorks-2020.st
setenv vxbootfile vxWorks-2010.st
```

4. Set the VxWorks bootline arguments.

```
setenv vxbootargs 'motetsec(0,0)10.130.101.216:vxWorks
h=10.130.101.216 e=10.130.101.206:ffffff00 u=vxworks
pw=vxworks f=0x80'
```

Parameters description:

```
motetsec(0,0) : Ethernet interface 0 on cpu 0
10.130.101.216 : Host Machine IP
10.130.101.206 : Board IP
ffffff00 : Netmask
u=vxworks : Username on host machine
p=vxworks : Password for the above user in host machine
f=0x80 : File Transfer Protocol (FTP)
```

5. Initialize SPI Flash0 device:

```
sf probe 0
```

**Note:** To initialize SPI Flash 1, set the command as below:

```
sf probe 1
```

6. To erase SPI Flash memory region:

```
sf erase 0x200000 0x300000
```

**Note:** Maximum space available in SPI Flash for VxWorks image is 5MB (0x500000). Here it is assumed that the VxWorks image size is less than 3MB(0x300000). If the VxWorks image size is more than 3MB, then you can change the image size here accordingly.

7. Load the VxWorks image from network to the memory location (0x1000000):

```
tftpboot VxWorks.st
```



- 
8. Write to SPI Flash 0 device at location (0x200000) from memory location(0x1000000):

```
sf write 0x1000000 0x200000 0x300000
```

**Note:** VxWorks image is loaded permanently to SPI flash 0. To load Vxworks image use SPI Flash device from next time.

To boot directly from SPI Flash follow below steps:

1. Initialize SPI Flash0 device

```
sf probe 0
```

**Note:** To initialize SPI Flash 1, set the command as below:

```
sf probe 1
```

2. Copy VxWorks image from SPI Flash 0 location (0x200000) to memory location (0x1000000) with 3MB size.

```
read 0x1000000 0x200000 0x300000
```

3. Set the VxWorks SPI Flash boot command.

```
setenv vxboot 'setenv bootargs $vxbootargs && bootvx'
```

4. Save your current environmental variables.

```
saveenv
```

5. To boot VxWorks through SPI Flash, execute the following command:

```
run vxboot
```



**Before executing SPI erase/write commands, ensure that the SPI Flash offset location and length are correct. Incorrect address can corrupt the U-Boot image.**

For sample output, refer to [Section A.4, SPI Flash Boot on page 31](#).



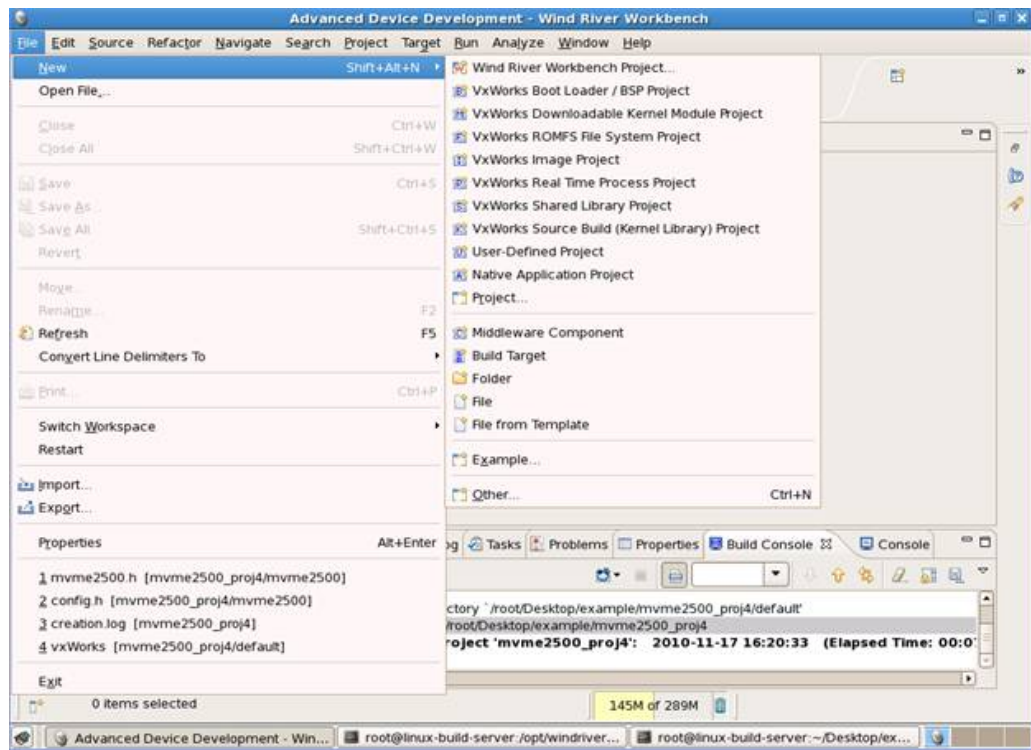
# Building Board Support Package

## 3.1 Building Procedure

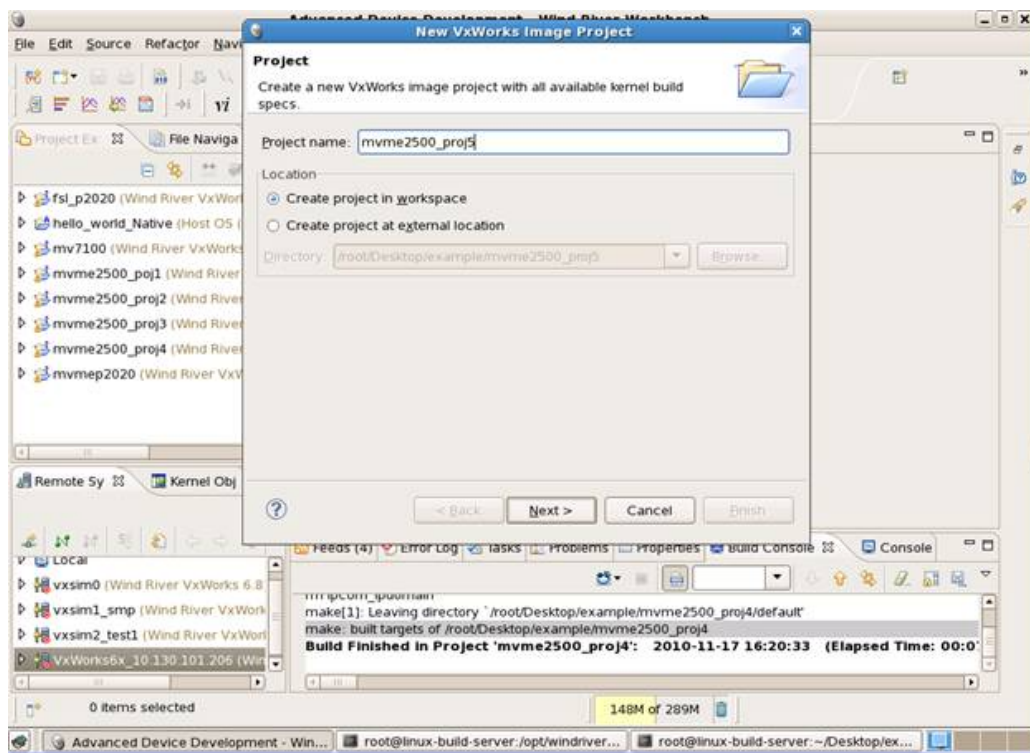
The `mvme2500_sp1.tar.gz` contains VxWorks 6.8 BSP source files for the MVME2500 board.

Perform the following steps to build the BSP:

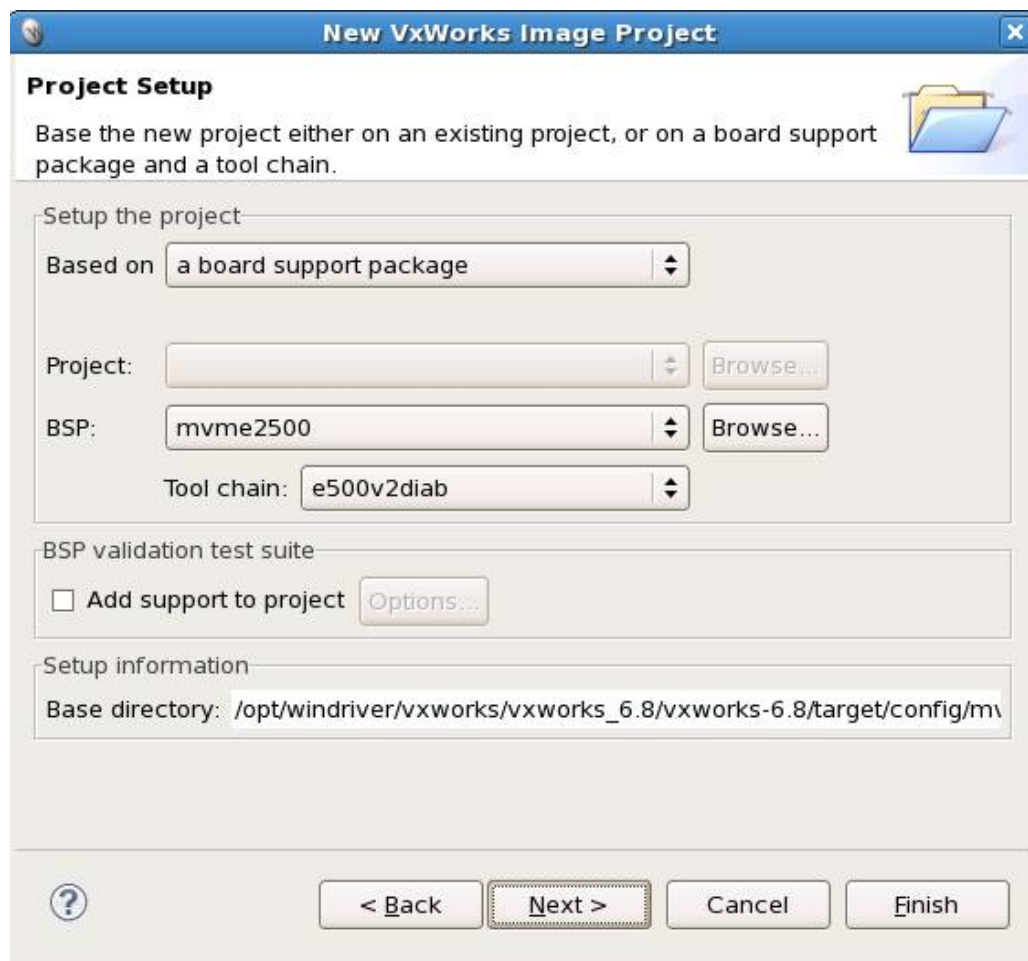
1. Extract the `mvme2500_sp1.tar.gz` to any working directory.
2. Start the Wind River VxWorks Workbench by executing the following command:  
`<vxWorks Installation Directory>/startWorkbench.sh`
3. Open the Wind River VxWorks Workbench. Select **File -> New -> VxWorks Image Project**



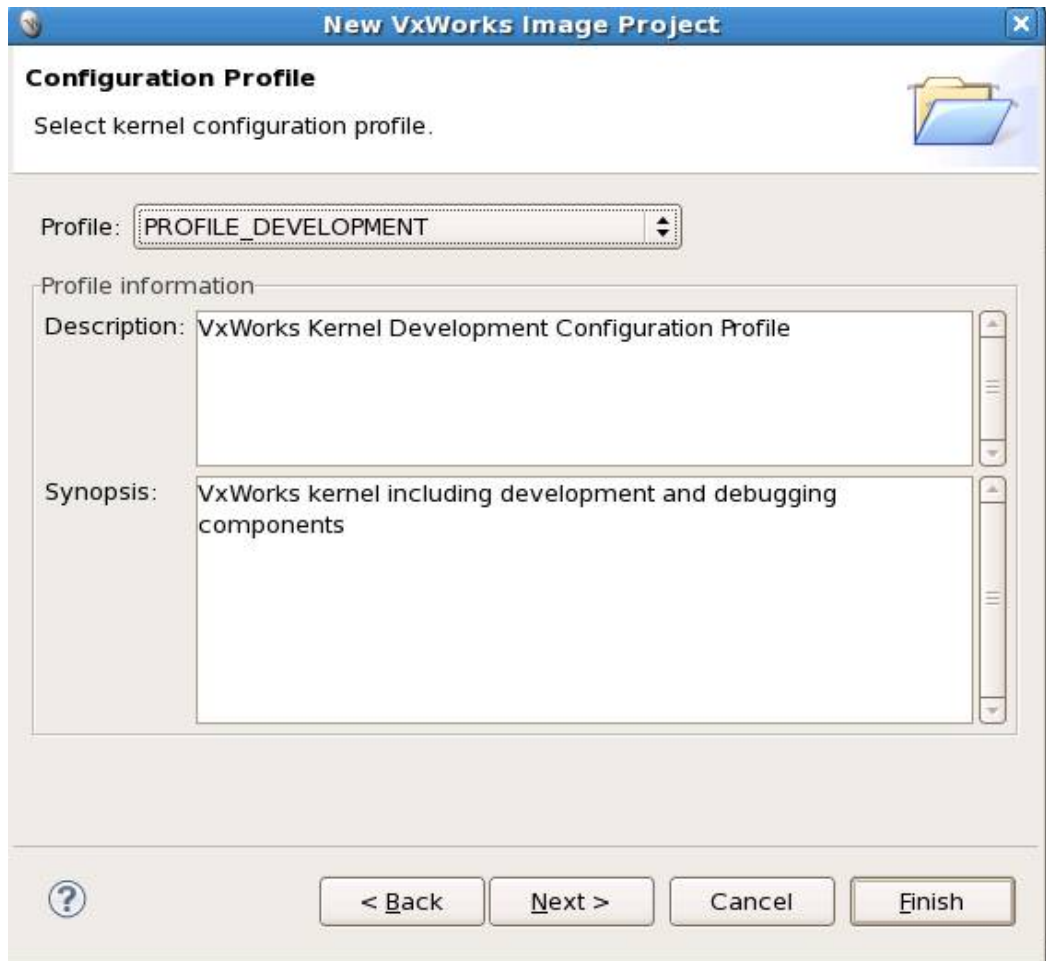
4. Give a project name and then click **Next**.



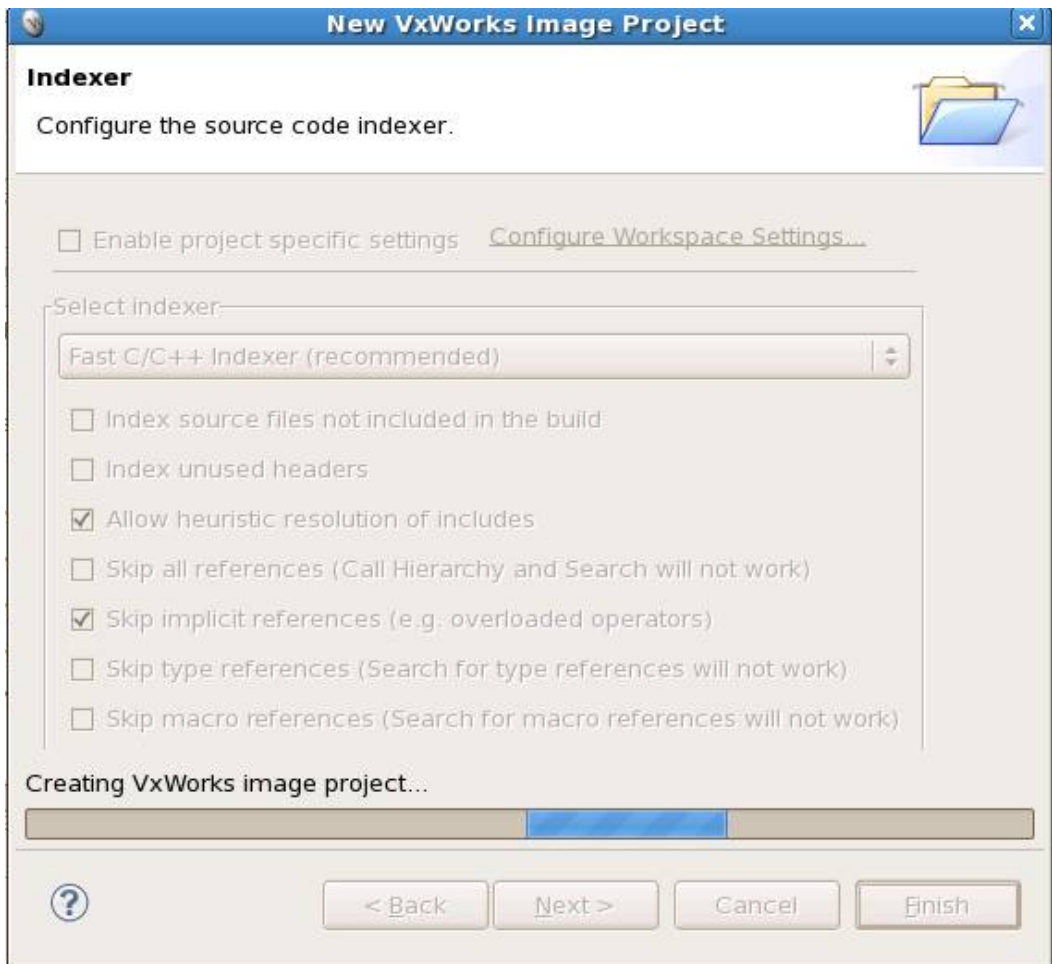
5. Go to BSP and select **mvme2500** from the dropdown list. Click **Browse** and point to the location where you have extracted the BSP. Click **Next**.



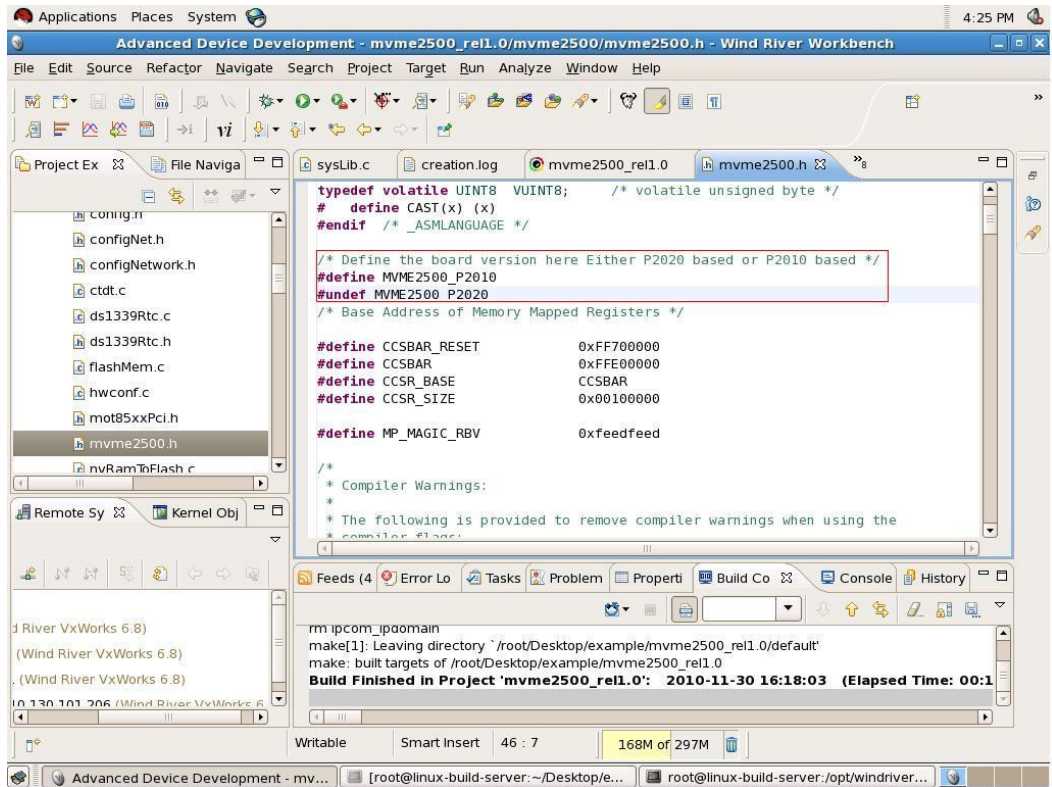
6. Select any configuration profile.



7. Click **Finish**.

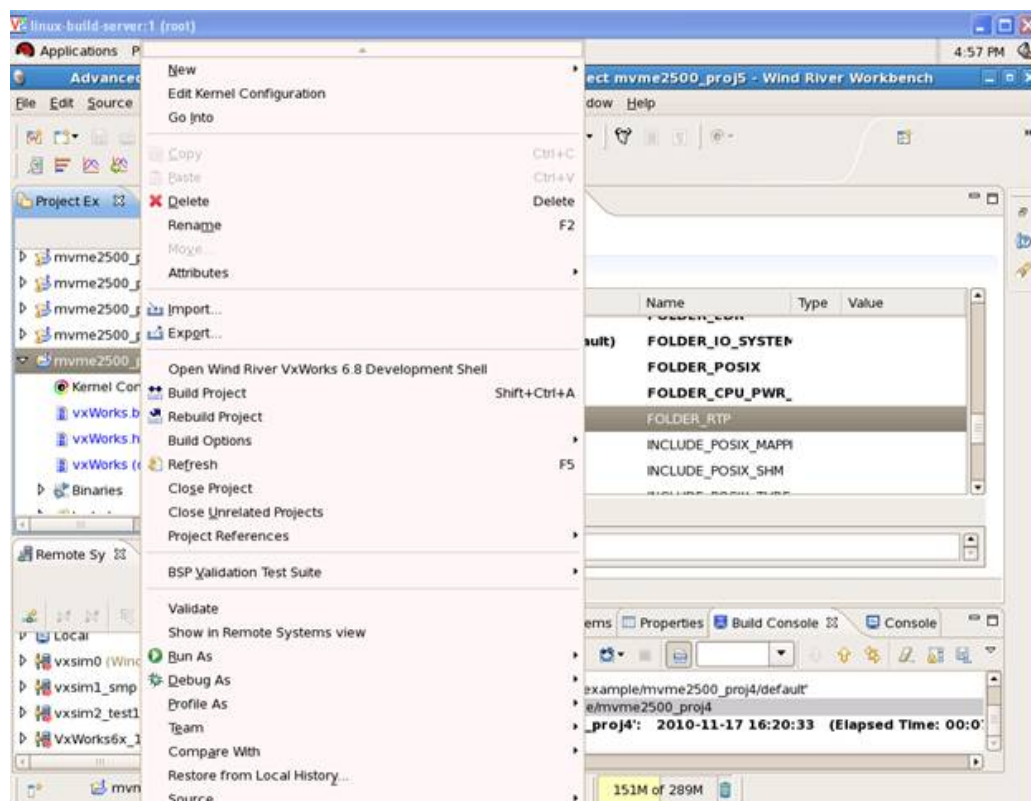


8. Double-click `mvme2500.h` file. Enable `MVME2500_P2020` or `MVME2500_P2010` based on the `MVME2500` board version.





9. Right-click the project name which you have created, and select **Build Project**.





# Sample Output

---

## A.1 Network Boot

```

=> run vxboot
Speed: 100, full duplex
Using eTSEC1 device
TFTP from server 10.130.101.216; our IP address is 10.130.101.217
Filename 'vxWorks-2020.st'.
Load address: 0x1000000
Loading:
#####
#####
#####
#####
done
Bytes transferred = 3460440 (34cd58 hex)
## Ethernet MAC address not copied to NV RAM
## Using bootline (@ 0x4200): motetsec(0,0)10.130.101.216:vxWorks
h=10.130.101.216 e=10.130.101.154:ffffff00 u=vxworks pw=vxworks f=0x80
## Starting vxWorks at 0x00100000 ...
Target Name: vxTarget
Bulk Class Driver Successfully Initialized
usbCbiUfiDevInit() returned OK
CBI Class Driver Successfully Initialized

Adding 8266 symbols for standalone.

```



---

```
h=10.130.101.216 e=10.130.101.250:ffffff00 u=vxworks pw=vxworks f=0x80
## Starting vxWorks at 0x00100000 ...
Bulk Class Driver Successfully Initialized
usbCbiUfiDevInit() returned OK
CBI Class Driver Successfully Initialized
```

```
Adding 5373 symbols for standalone.
```

```
))))))))))))))))))))))))))))))))))))))))))
))))))))))))))))))))))))))))))))))))))))))
))))))))))))))))))))))))))))))))))))))))))
      )))))))) )))) )))))))) )))) )))) )))) (R)
) )))))))) )))) )))))))) )))) )))) ))))
)) )))))) )))))))) )))))) )))) )))) ))))
)) )))) )) )))) )) )) )) )))))))) )))) )) )))) )) )))) ))))
)))) )))) )) )))) )) )) )) )) )))))))) )))) )) )))) )))) ))))
)))) )))) )) )))) )) )) )) )) )))))))) )))) )) )))) )))) ))))
))))) )))) )))) )))) )) )) )) )) )))))))) )))) )) )))) )))) ))))
))))) )))) )))) )))) )) )) )) )) )))))))) )))) )) )))) )))) ))))
))))) )))) )))) )))) )) )) )) )) )))))))) )))) )) )))) )))) ))))
))))) )))) )))) )))) )) )) )) )) )))))))) )))) )) )))) )))) ))))
))))))))))))))))))))))))))))))))))))))))))
)))))))))))))))))))))))))))))))))) Development System
))))))))))))))))))))))))))))))))))
)))))))))))))))))))))))))))))))))) VxWorks 6.8
)))))))))))))))))))))))))))))))))) KERNEL: WIND version 2.13
)))))))))))))))))))))))))))))))))) Copyright Wind River Systems, Inc., 1984-2009
CPU: Freescale P2010E - Security Engine. Processor #0.
Memory Size: 0x40000000. BSP version 2.0/2.
Created: Feb 23 2011, 15:31:29
ED&R Policy Mode: Deployed
```

```
-> 0xc92b50 (devConnect): vxbIntelAhciInstConnect pDev 0x2be358
0xc92b50 (devConnect): ahciDrv called 0xc8fa20 0xa4100000
Found Bulk Device with 1 Logical Units at node 2
Instantiating /ahci00:1 as rawFs, device = 0x50001
->
```

## A.3 USB Boot

```
=> run vxusbboot
```



---

```
))))))))))))))))))))))))))
))))))))))))))))))))))))))      VxWorks 6.8
))))))))))))))))))))))))))      KERNEL: WIND version 2.13
))))))))))))))))))))))))))      Copyright Wind River Systems, Inc., 1984-2009
CPU: Freescale P2010E - Security Engine.  Processor #0.
Memory Size: 0x40000000.  BSP version 2.0/2.
Created: Feb 23 2011, 15:31:29
ED&R Policy Mode: Deployed
WDB Comm Type: WDB_COMM_END
WDB: Ready.
->
```

## A.4 SPI Flash Boot

```
U-Boot 2010.06 (Aug 20 2012 - 13:08:23)
CPU0: P2020E, Version: 2.1, (0x80ea0021)
Core: E500, Version: 5.1, (0x80211051)
Clock Configuration:
    CPU0:1000 MHz, CPU1:1000 MHz,
    CCB:400 MHz,
    DDR:400 MHz (800 MT/s data rate) (Asynchronous), LBC:25 MHz
L1: D-cache 32 kB enabled
    I-cache 32 kB enabled
Board: MVME2500
    Emerson Network Power, Embedded Computing
    Monitor Version: 1.5
    FPGA Seq.Ver: 2.5
    Is not VME system controller
I2C: ready
SPI: ready
DRAM: Initializing.... DDR: 2 GiB (DDR3, 64-bit, CL=6, ECC on)
L2: 512 KB enabled
MMC: FSL_ESDHC: 0
EEPROM: Read MAC Address
PCIE2 connected as Root Complex (base addr ffe09000)
PCIE2 on bus 00 - 00
    PCIE3 connected as Root Complex (base addr ffe08000)
```

---

```
PCIE3 on bus 01 - 01
PCIE1 connected as Root Complex (base addr ffe0a000)
    Scanning PCI bus 03
        04 01 10e3 0148 0680 00
        03 00 10e3 8114 0604 ff
PCIE1 on bus 02 - 04
```

```
In:    serial
Out:   serial
Err:   serial
```

```
Ser#: 9228697
I-cache enabled. (L1CSR1: 0x00000001)
D-cache enabled. (L1CSR0: 0x00000001) (write-through)
SCSI:  Error SCSI Controller (11AB,6121) not found
Net:   eTSEC1, eTSEC2, eTSEC3
Bootreg = a5 BootDev: SPI1  Switch: SPI1
Autoboot in 3 seconds (hit 'h' to stop)
MVME2500=>
MVME2500=> sf
sf - SPI flash sub-system
```

```
Usage:
sf probe [bus:]cs [hz] [mode] - init flash device on given SPI bus
                                and chip select
sf read addr offset len      - read `len' bytes starting at
`offset' to memory at `addr'
sf write addr offset len     - write `len' bytes from memory
                                at `addr' to flash at `offset'
sf erase offset len          - erase `len' bytes from `offset'
MVME2500=>
MVME2500=> sf probe 0
8192 KiB AT25DF641 at 0:0 is now current device
MVME2500=> sf erase 0x200000 0x300000
MVME2500=>
MVME2500=> tftpboot vxWorks.st
```



---

```

Speed: 100, full duplex
Using eTSEC1 device
TFTP from server 10.130.101.113; our IP address is 10.130.101.246
Filename 'vxWorks.st'.
Load address: 0x1000000
Loading:
#####
#####
#####
done
Bytes transferred = 2803150 (2ac5ce hex)
MVME2500=> md 0x1000000
01000000: 7f454c46 01020100 00000000 00000000      .ELF.....
01000010: 00020014 00000001 00100000 00000034      .....4
01000020: 00245274 80000000 00340020 00020028      .$Rt....4. ...(
01000030: 00150012 00000001 00000080 00100000      .....
01000040: 00100000 001e80b0 001e80b0 00000007      .....
01000050: 00000040 00000001 001e8140 00300000      ...@.....@.0..
01000060: 00300000 0002f9f0 0005c290 00000006      .0.....
01000070: 00000020 00000000 00000000 00000000      ... ..
01000080: 7c681b78 7c6000a6 5464045e 548403da      |h.x|\`..Td.^T...
01000090: 54840524 7c800124 4c00012c 7c631a78      T..$|..$L.,|c.x
010000a0: 7c7453a6 7cc63278 7c0004ac 4c00012c      |tS.|.2x|...L.,
010000b0: 7cd2fba6 38c00000 7c0004ac 4c00012c      |...8...|...L.,
010000c0: 7cd2fba6 38c00002 7c0004ac 4c00012c      |...8...|...L.,
010000d0: 7cd3fba6 38c00002 7c0004ac 4c00012c      |...8...|...L.,
010000e0: 7cd3fba6 4c00012c 38c00000 7c0004ac      |...L...,8...|...
010000f0: 4c00012c 7cd3fba6 7c0004ac 4c00012c      L.,|...|...L.,
MVME2500=>
MVME2500=> sf write 0x1000000 0x200000 0x300000
MVME2500=>
MVME2500=> reset

U-Boot 2010.06 (Aug 20 2012 - 13:08:23)CPU0: P2020E, Version: 2.1,
(0x80ea0021)
Core: E500, Version: 5.1, (0x80211051)

```

---

Clock Configuration:

CPU0:1000 MHz, CPU1:1000 MHz,

CCB:400 MHz,

DDR:400 MHz (800 MT/s data rate) (Asynchronous), LBC:25 MHz

L1: D-cache 32 kB enabled

I-cache 32 kB enabled

Board: MVME2500

Emerson Network Power, Embedded Computing

Monitor Version: 1.5

FPGA Seq.Ver: 2.5

Is not VME system controller

I2C: ready

SPI: ready

DRAM: Initializing.... DDR: 2 GiB (DDR3, 64-bit, CL=6, ECC on)

L2: 512 KB enabled

MMC: FSL\_ESDHC: 0

EEPROM: Read MAC Address

PCIE2 connected as Root Complex (base addr ffe09000)

PCIE2 on bus 00 - 00

PCIE3 connected as Root Complex (base addr ffe08000)

PCIE3 on bus 01 - 01

PCIE1 connected as Root Complex (base addr ffe0a000)

Scanning PCI bus 03

04 01 10e3 0148 0680 00

03 00 10e3 8114 0604 ff

PCIE1 on bus 02 - 04

In: serial

Out: serial

Err: serial

Ser#: 9228697

I-cache enabled. (L1CSR1: 0x00000001)

D-cache enabled. (L1CSR0: 0x00000001) (write-through)

SCSI: Error SCSI Controller (11AB,6121) not found

Net: eTSEC1, eTSEC2, eTSEC3

```

Bootreg = a5 BootDev: SPI1   Switch: SPI1
Autoboot in 3 seconds (hit 'h' to stop)
MVME2500=>
MVME2500=> sf probe 0
8192 KiB AT25DF641 at 0:0 is now current device
MVME2500=> sf read 0x1000000 0x200000 0x300000
MVME2500=> md 0x1000000
01000000: 7f454c46 01020100 00000000 00000000   .ELF.....
01000010: 00020014 00000001 00100000 00000034   .....4
01000020: 00245274 80000000 00340020 00020028   .$Rt....4. ...(
01000030: 00150012 00000001 00000080 00100000   .....
01000040: 00100000 001e80b0 001e80b0 00000007   .....
01000050: 00000040 00000001 001e8140 00300000   ...@.....@.0..
01000060: 00300000 0002f9f0 0005c290 00000006   .0.....
01000070: 00000020 00000000 00000000 00000000   ... ..
01000080: 7c681b78 7c6000a6 5464045e 548403da   |h.x|\`..Td.^T...
01000090: 54840524 7c800124 4c00012c 7c631a78   T..$|..$L..,|c.x
010000a0: 7c7453a6 7cc63278 7c0004ac 4c00012c   |tS.|.2x|...L..,
010000b0: 7cd2fba6 38c00000 7c0004ac 4c00012c   |...8...|...L..,
010000c0: 7cd2fba6 38c00002 7c0004ac 4c00012c   |...8...|...L..,
010000d0: 7cd3fba6 38c00002 7c0004ac 4c00012c   |...8...|...L..,
010000e0: 7cd3fba6 4c00012c 38c00000 7c0004ac   |...L..,8...|...
010000f0: 4c00012c 7cd3fba6 7c0004ac 4c00012c   L..,|...|...L..,
MVME2500=>
MVME2500=> setenv bootargs $vxbootargs && bootvx
## Ethernet MAC address not copied to NV RAM
## Using bootline (@ 0x4200): motetsec(0,0)10.130.101.113:vxWorks
h=10.130.101.113 e=10.130.101.246:ffffff00 u=vxworks pw=vxworks f=0x80
## Starting vxWorks at 0x00100000 ...
Target Name: vxTarget

```

Adding 6898 symbols for standalone.





