

Outline

- Software Requirements
- Software Setup
- Generate VxWorks BSP
- Create VxWorks Project
- Create VxWorks System Image
- Generate and Run an ACE File

Additional Setup Details

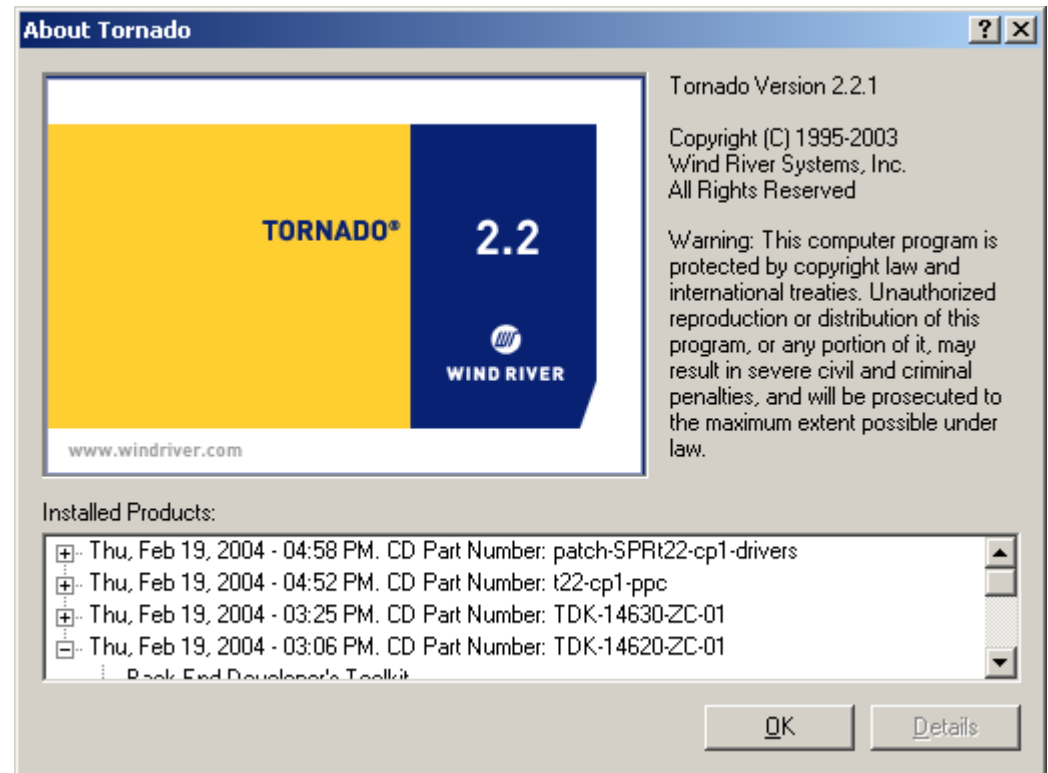
- Refer to *ml310_overview_setup.ppt* for details on:
 - Software Requirements
 - ML310 Board Setup
 - Equipment and Cables
 - Software
 - Terminal Programs
 - This presentation requires the 9600-8-N-1 Baud terminal setup



Software Setup

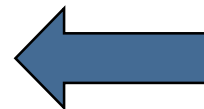
- Install Wind River Systems Tornado 2.2

- CD part number TDK-14620-ZC-01
- CD part number TDK-14630-ZC-01
- Download and install Patch T22CP1: Tornado 2.2 Cumulative Patch 1 "CP-1"
- The patch files:
 - T22CP1_install.html
 - t22-cp1-ppc.tar
 - t22-cp1-drivers.tar
- Details on the installed products can be found by clicking on **Help** → **About Tornado** → **Details**



Software Setup

- Install Xilinx ISE 6.2i SP3

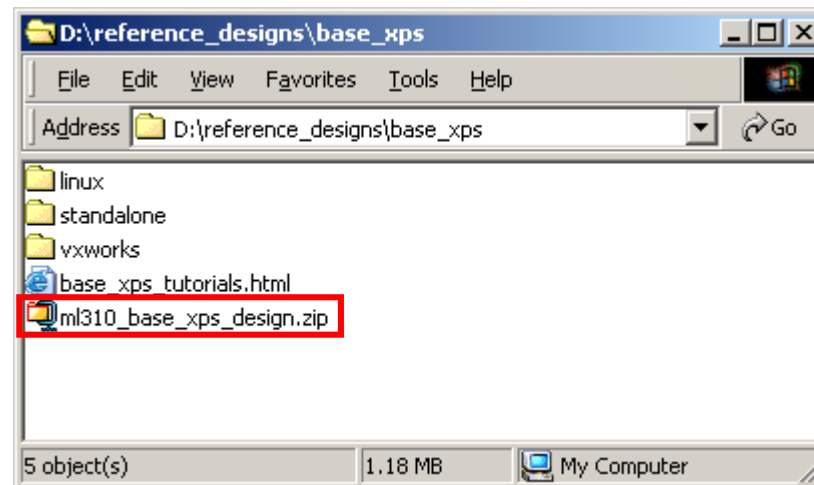


- Install EDK 6.2i SP2

File Setup

- This presentation shows how to re-create the pre-built BSP and project by starting with the EDK Base XPS design
- Files needed from ML310 CD:
 - **EDK Project:**

<cd drive>:\reference_designs\base_xps\ml310_base_xps_design.zip



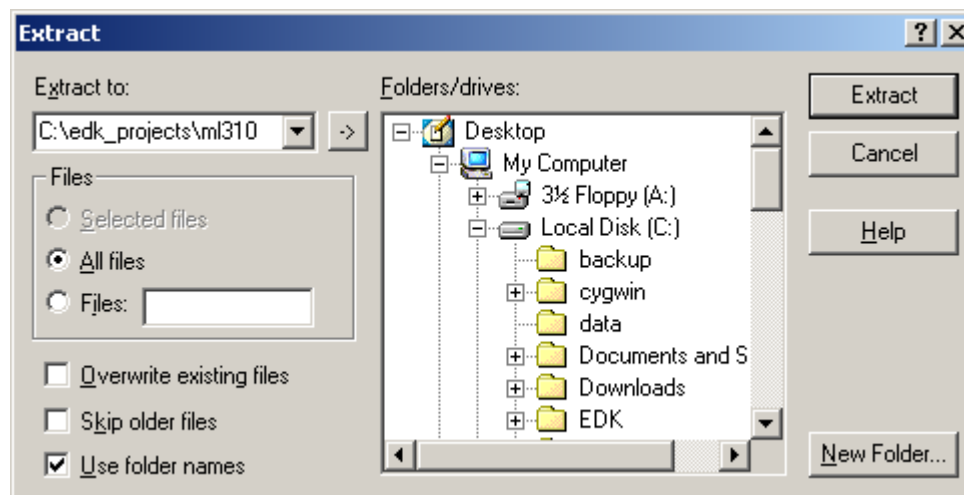
File Setup

- Unzip the EDK design project zip file:
ml310_base_xps_design.zip

to:

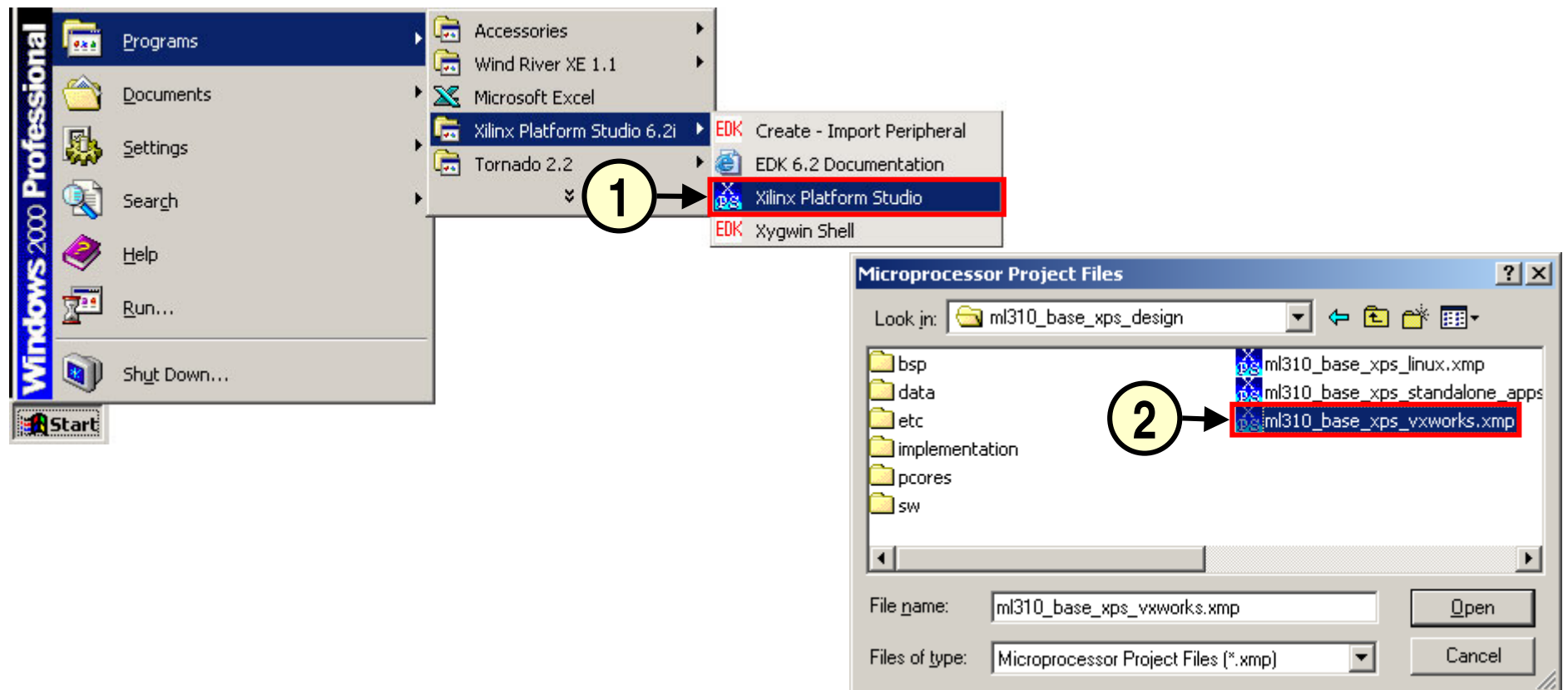
C:\edk_projects\ml310

- Note: EDK projects can be located anywhere; this is just a suggested location



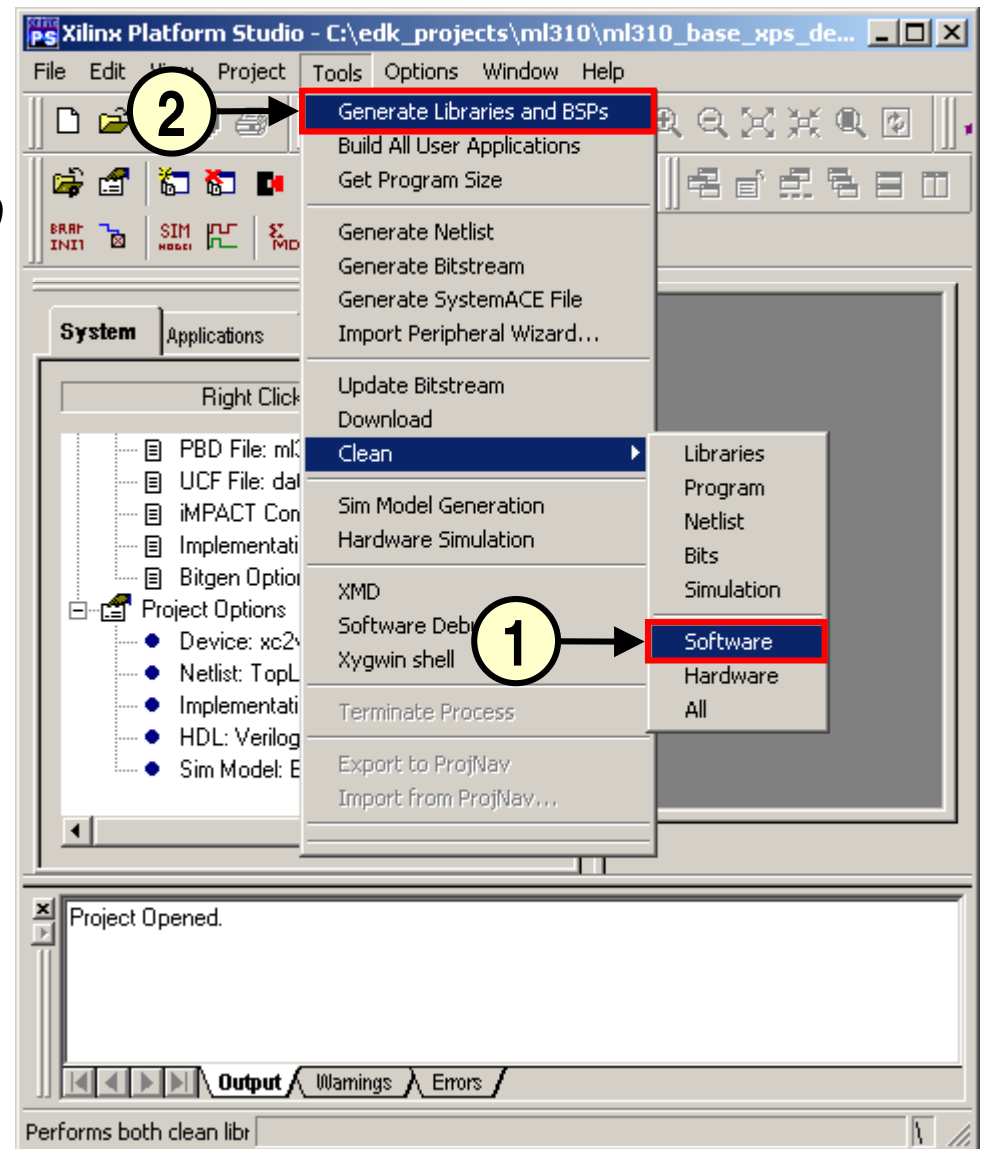
Run Xilinx Platform Studio

- Launch Xilinx Platform Studio (XPS) (1)
- Within XPS select **File** → **Open Project** and choose (2):
`<edk_proj>\ml310\ml310_base_xps_design\ml310_base_xps_vxworks.xmp`



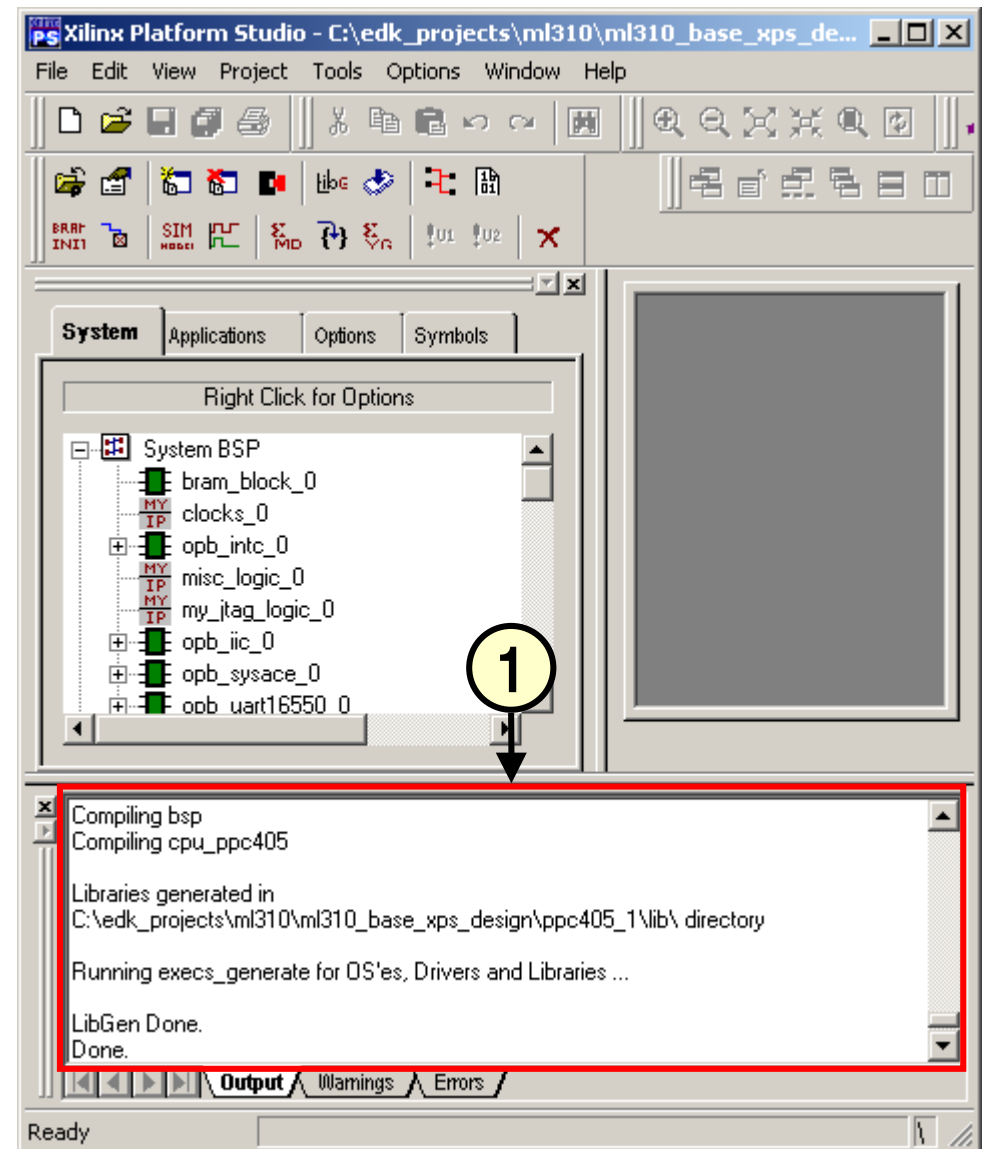
Generate BSP in EDK

- After loading *ml310_base_xps_vxworks.xmp*
Select :
 - Tools → Clean → Software (1)
 - Tools → Generate Libraries and BSPs (2)



Generate BSP in EDK

- As the VxWorks BSP is being generated, informational messages will be sent to the output window (1)



Optional Installation

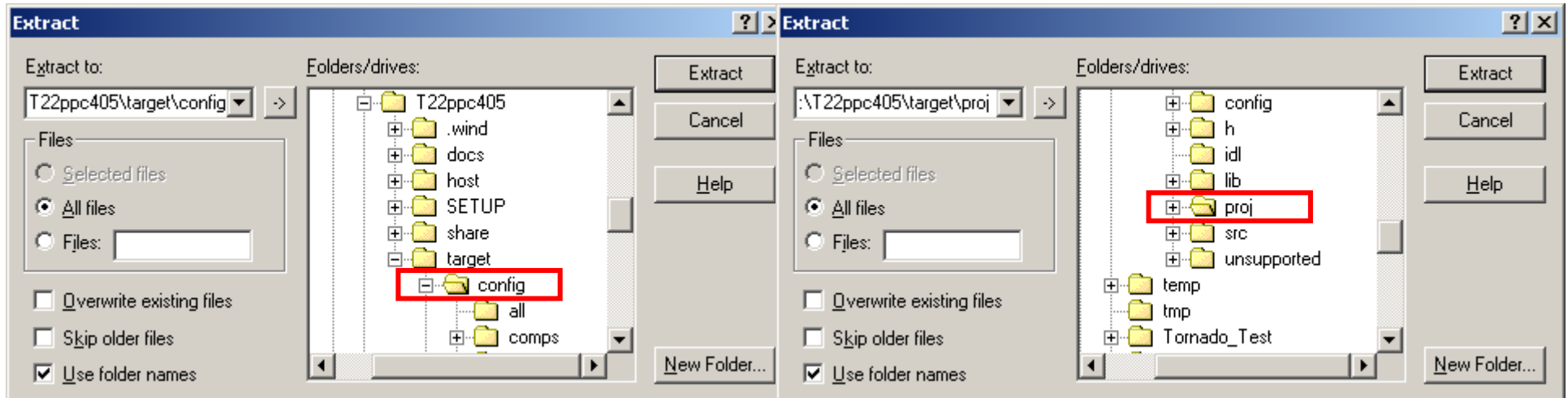
- Optional: Bypass the the following VxWorks BSP and Project creation steps by using the pre-built BSP and Project:

BSP: `<cd drive>:\reference_designs\base_xps\vxworks\ml310_base_xps_vxworks_bsp.zip`

Project: `<cd drive>:\reference_designs\base_xps\vxworks\ml310_base_xps_vxworks_proj.zip`

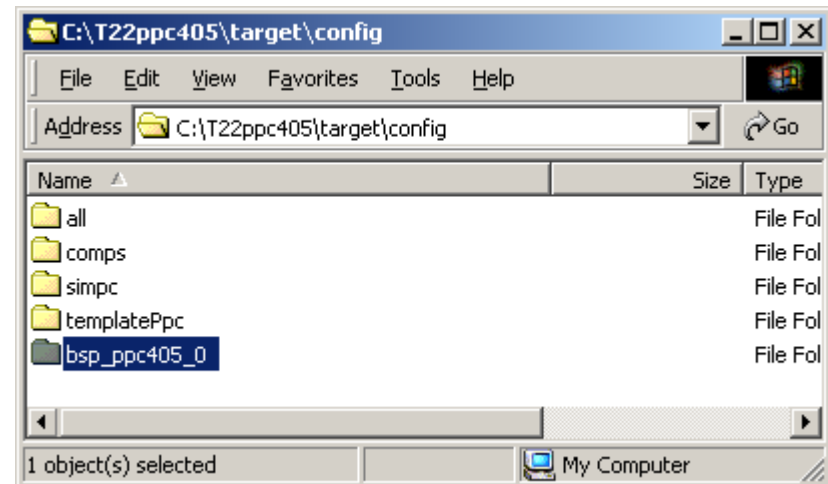
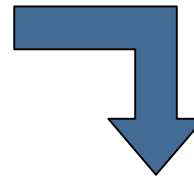
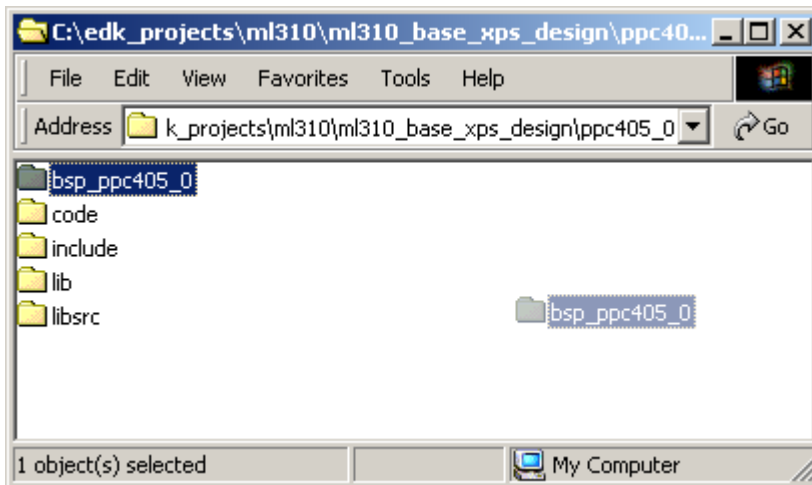
- Unzip the file *ml310_base_xps_vxworks_bsp.zip* to the *<Tornado Dir>\target\config* directory
- Unzip the *ml310_base_xps_vxworks_proj.zip* file to the *<Tornado Dir>\target\proj* directory

– Skip slides 12 through 36 if doing the Optional Installation



Copy Updated BSP to Tornado Tree

- Copy
`<edk_proj>\ml310\ml310_base_xps_design\ppc405_0\bsp_ppc405_0`
to:
`<Tornado Dir>\target\config\`
 - Hold down **<Ctrl>** while dragging to copy instead of move



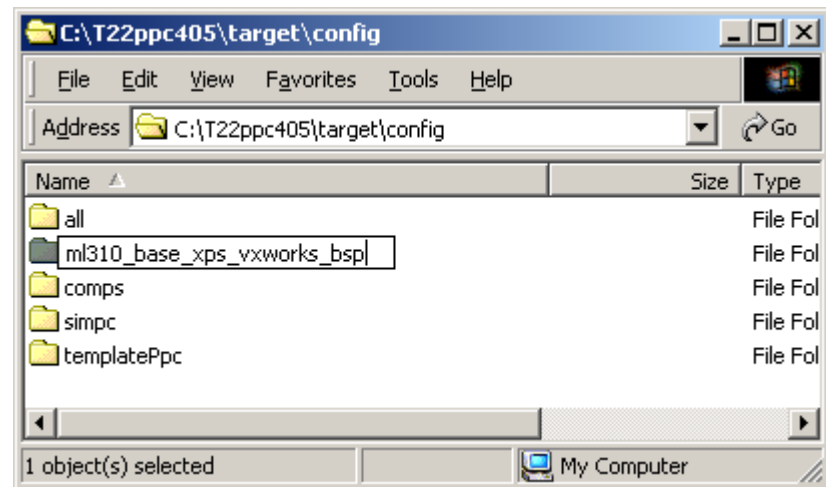
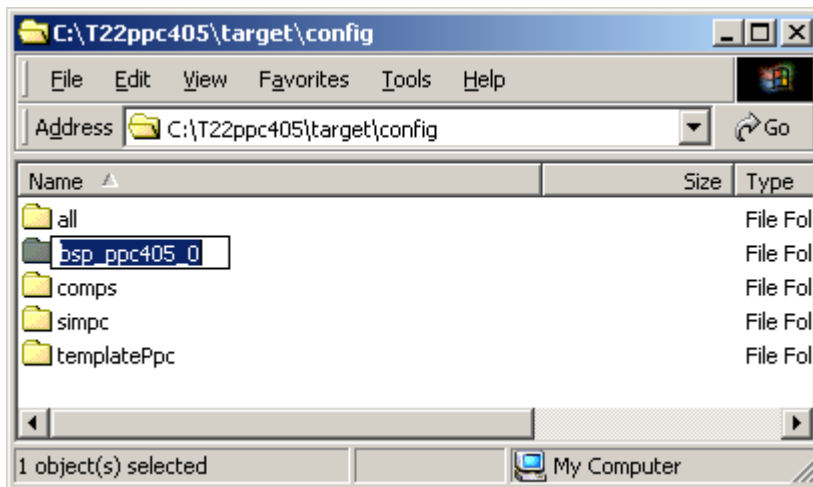
Rename BSP in Tornado Tree

- In *<Tornado Dir>\target\config* rename the bsp directory:

bsp_ppc405_0

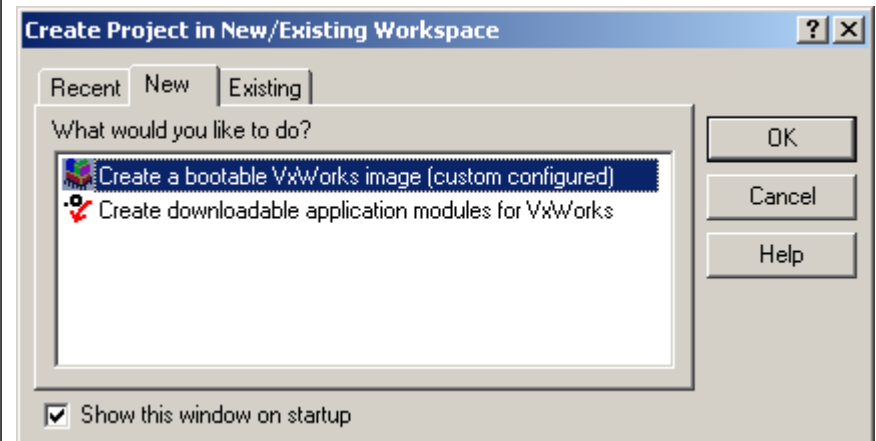
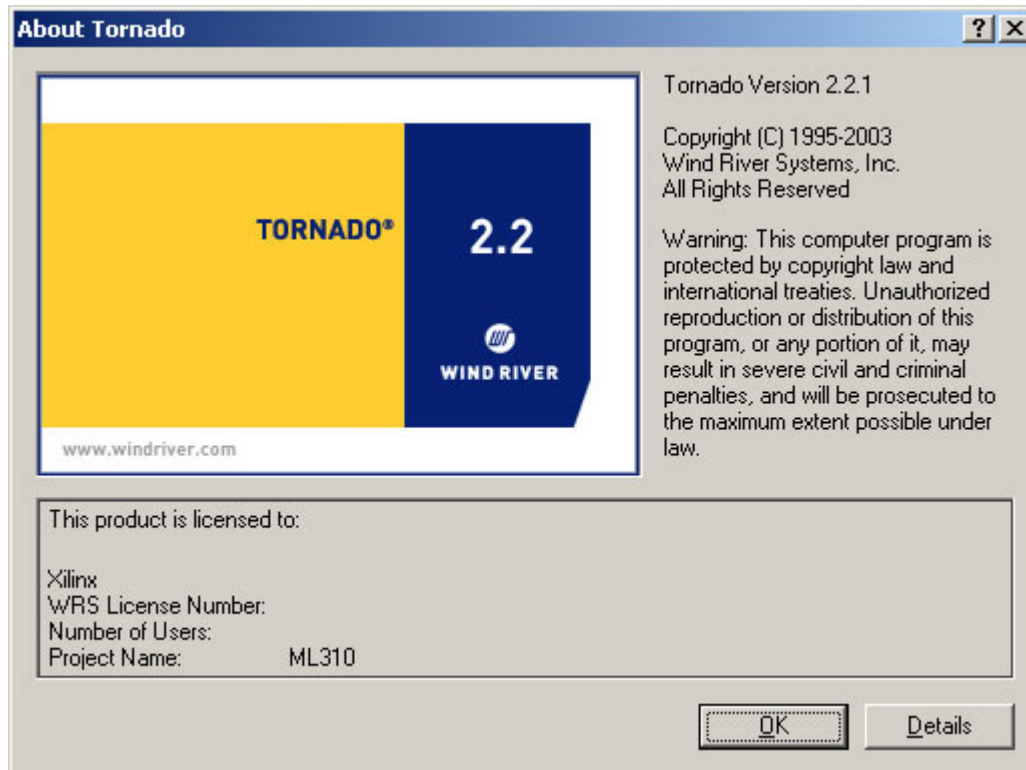
to:

ml310_base_xps_vxworks_bsp



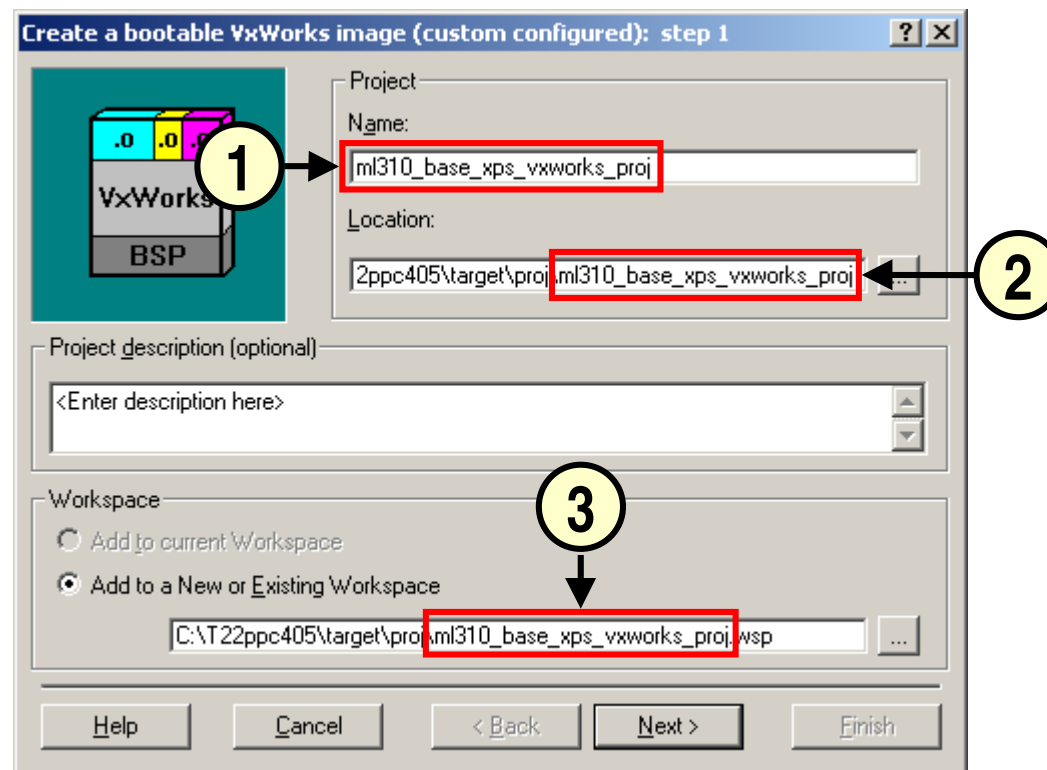
Create VxWorks Project

- Launch Tornado and select **File** → **New Project**
- Select “Create a bootable VxWorks image...”



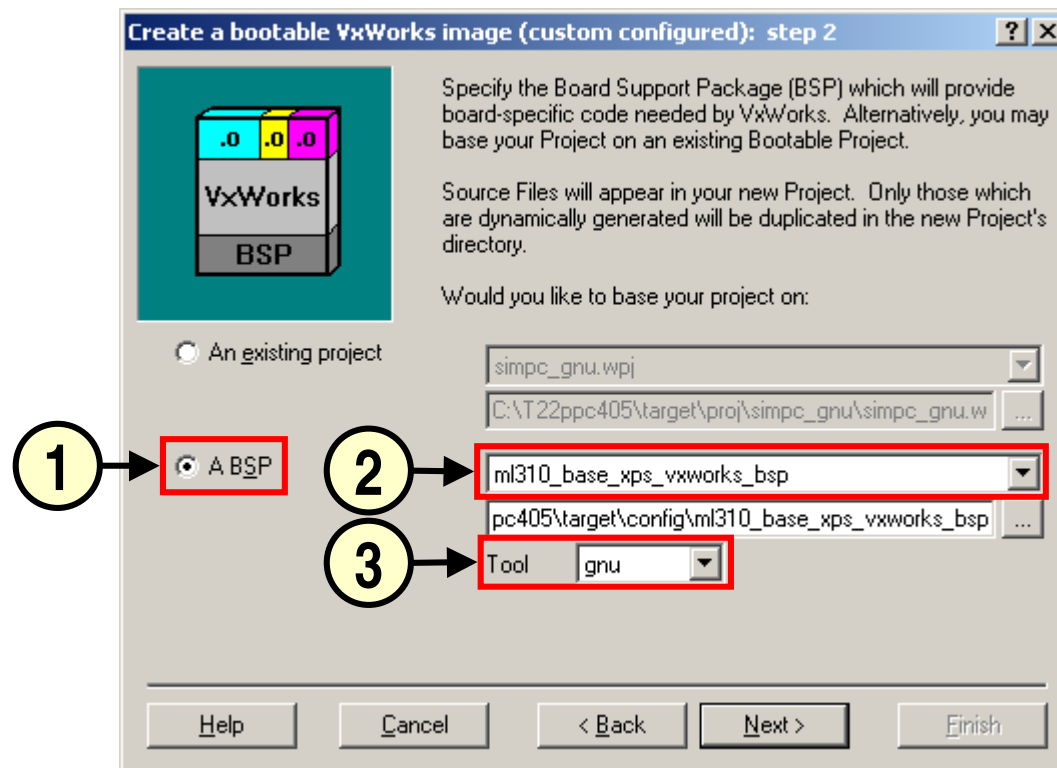
Create VxWorks Project

- Set the project name to **ml310_base_xps_vxworks_proj** (1)
- Update the Location and Workspace entries with this project name (2) , (3)



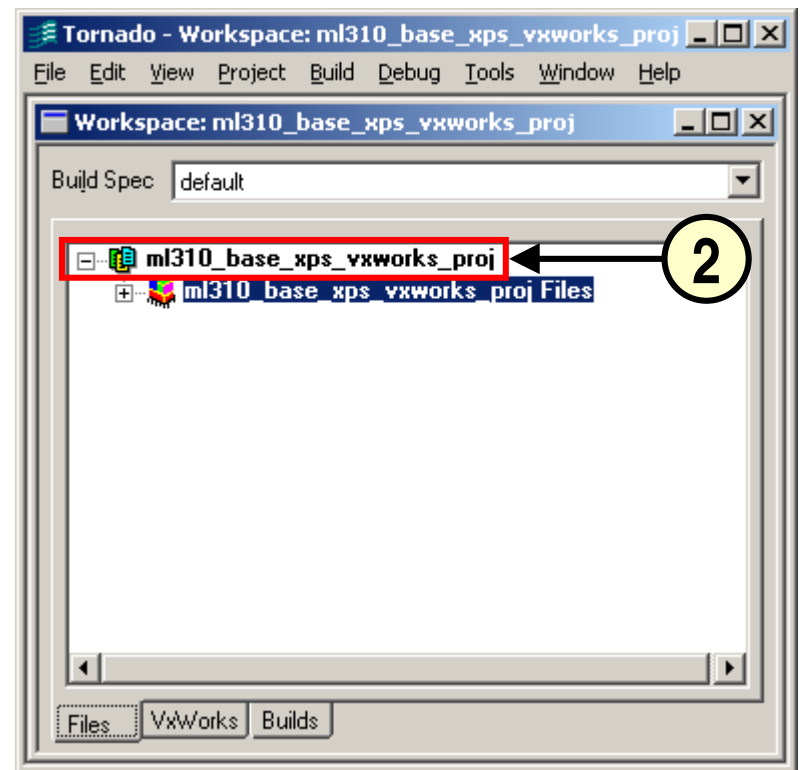
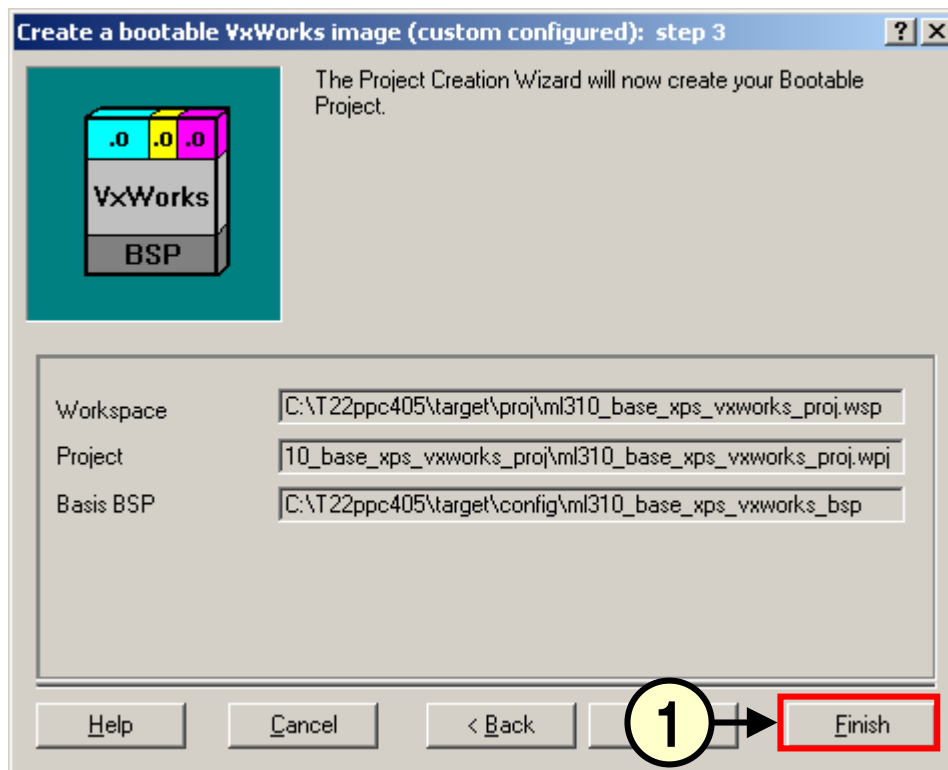
Create VxWorks Project

- Base the project on a BSP (1), (2)
 - Select the **ml310_base_xps_vxworks_bsp** we just added to the Tornado tree
- Select **gnu** from the Tool drop-down menu (3)



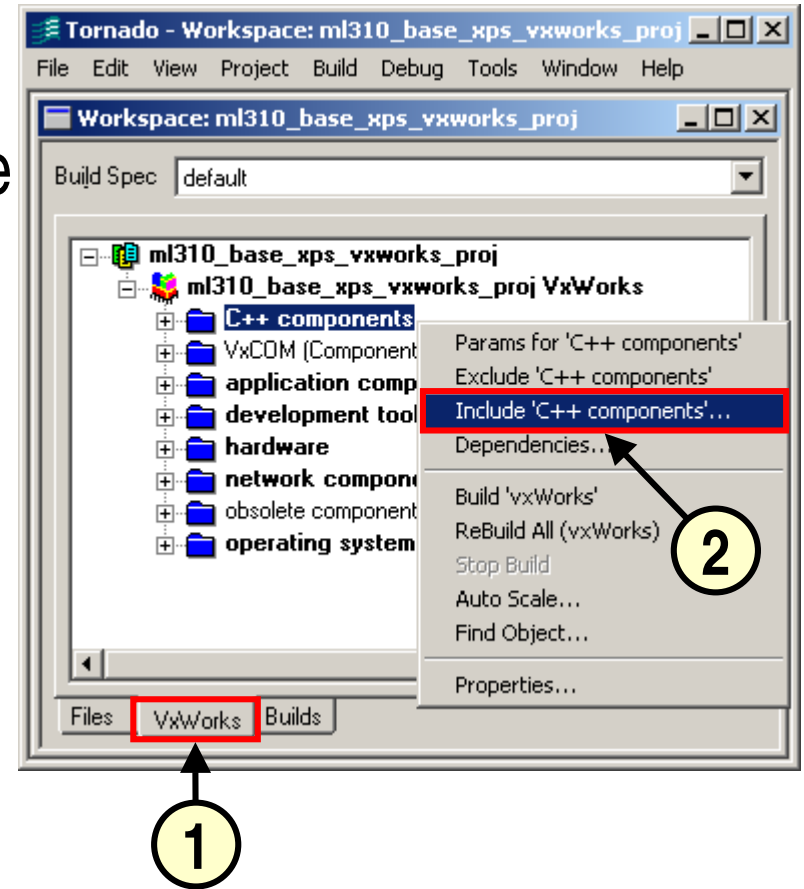
Create VxWorks Project

- Click **Finish** to create the Bootable project (1)
- New Tornado Project appears after BSP processed (2)



Configure System Image

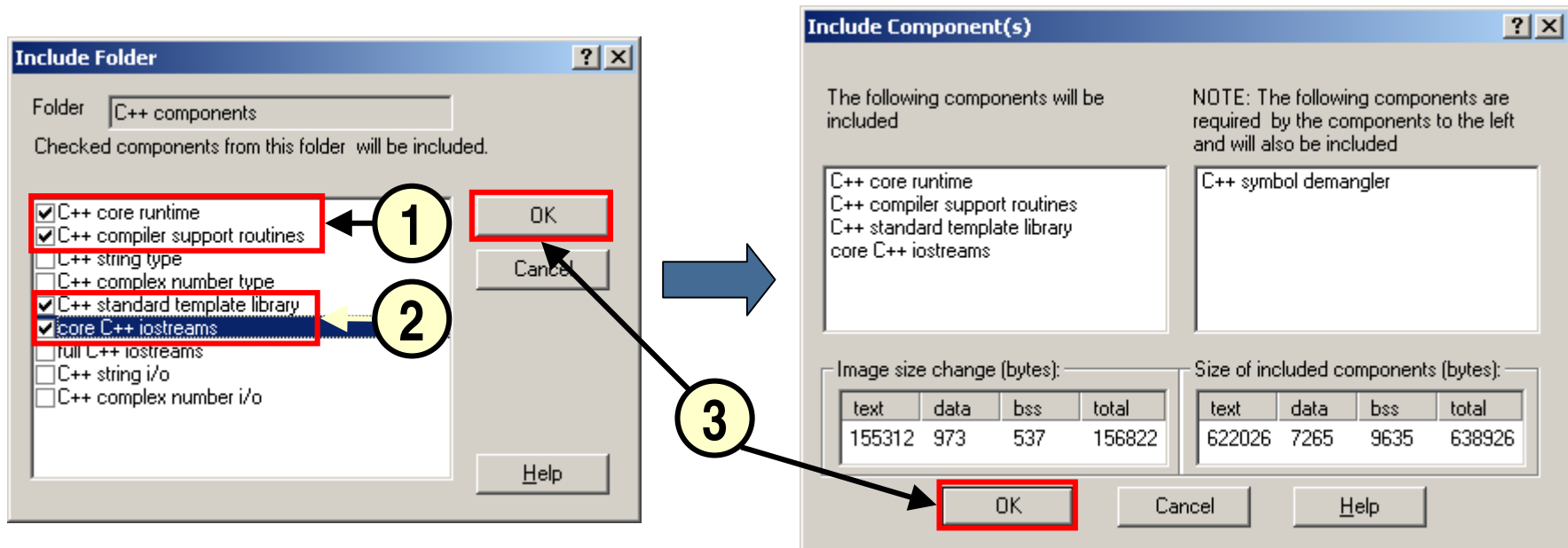
- To reproduce the pre-compiled Base XPS VxWorks system image
 - Components must be Included or Excluded
- Select the “VxWorks” tab (1)
- Right-click on the desired components and select “Include <component>” (2)
 - In some cases you will select “Exclude <component>”



Configure System Image

- Include: **C++ Components**

- Some components are “pre-checked” - leave these checked (1)
- Check “C++ standard template library” (2)
- Check “core C++ iostreams” (2)
- Click **OK** in the respective dialog boxes (3)

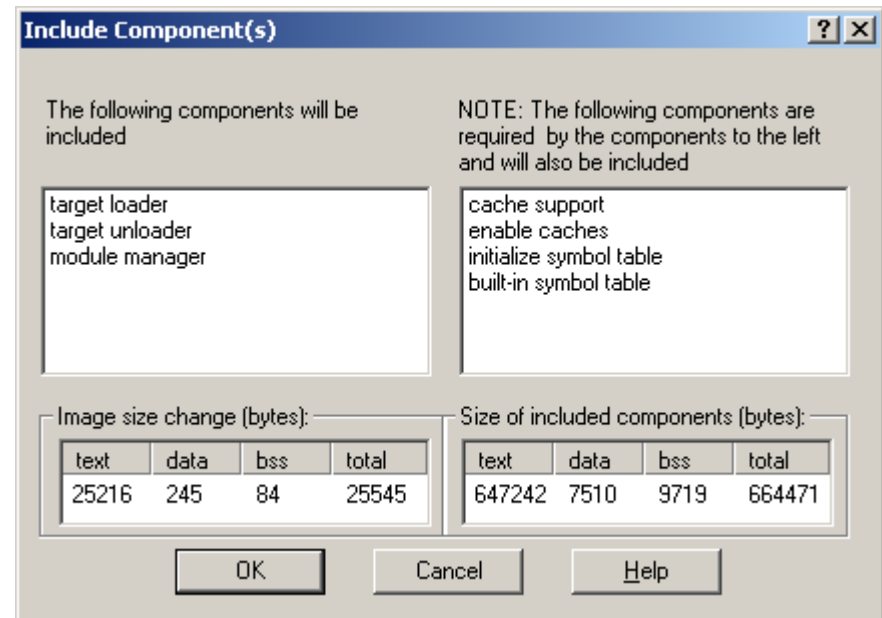
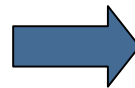
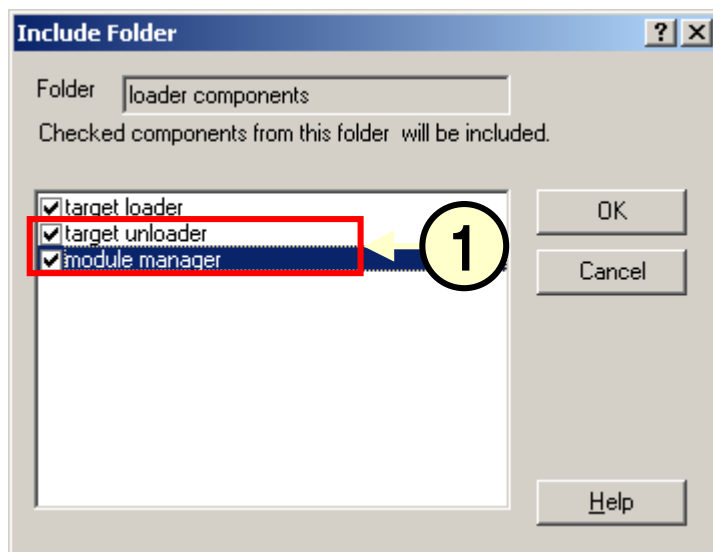


Configure System Image

- Include:

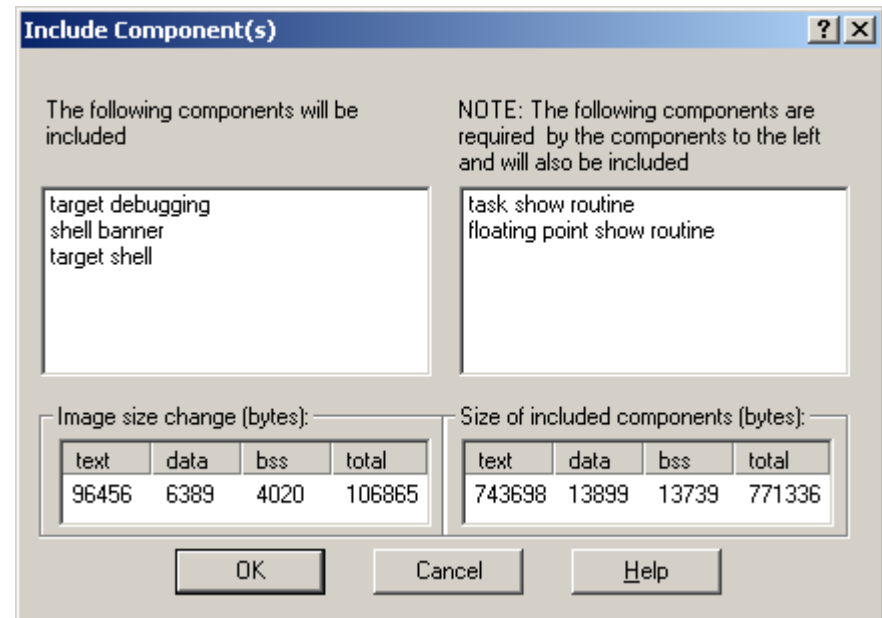
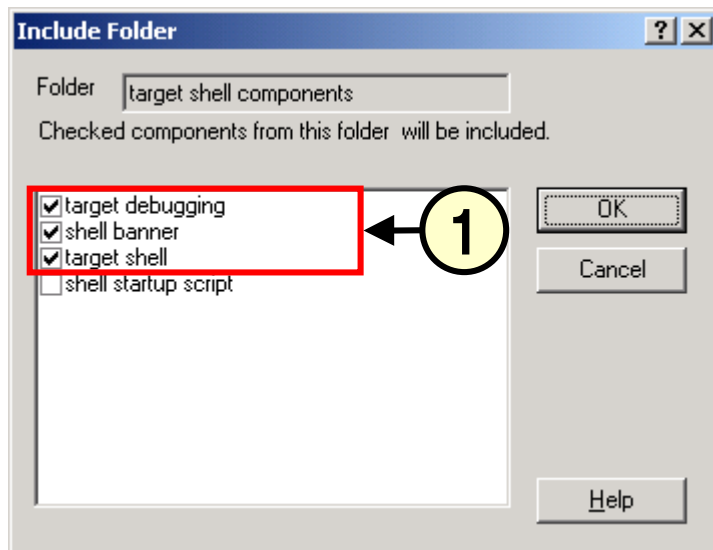
development tool components > loader components

- Check “target unloader” (1)
- Check “module manager” (1)
- Click OK in the respective dialog boxes



Configure System Image

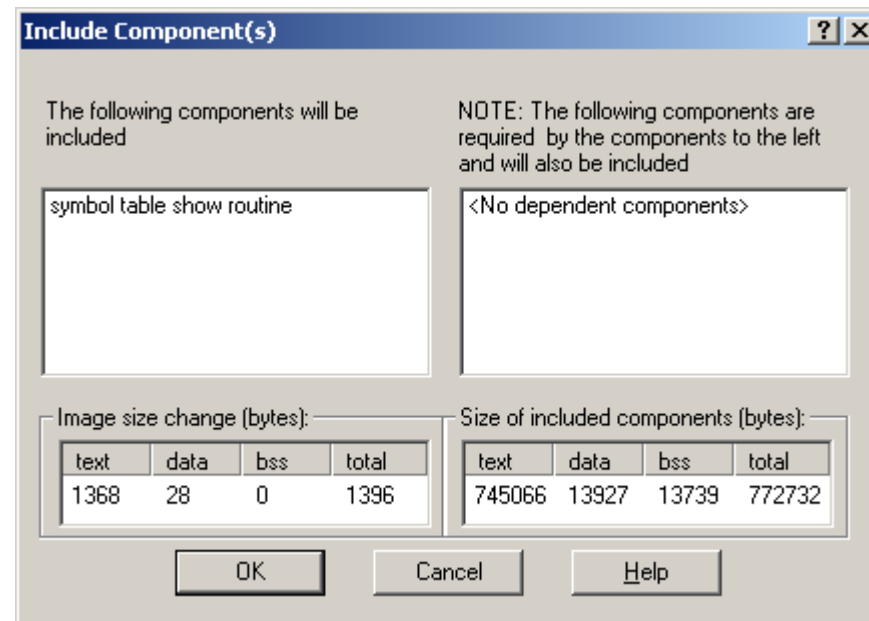
- Include:
 - development tool components > target shell components
 - Use default includes of the first 3 check boxes (1)



Configure System Image

- Include:

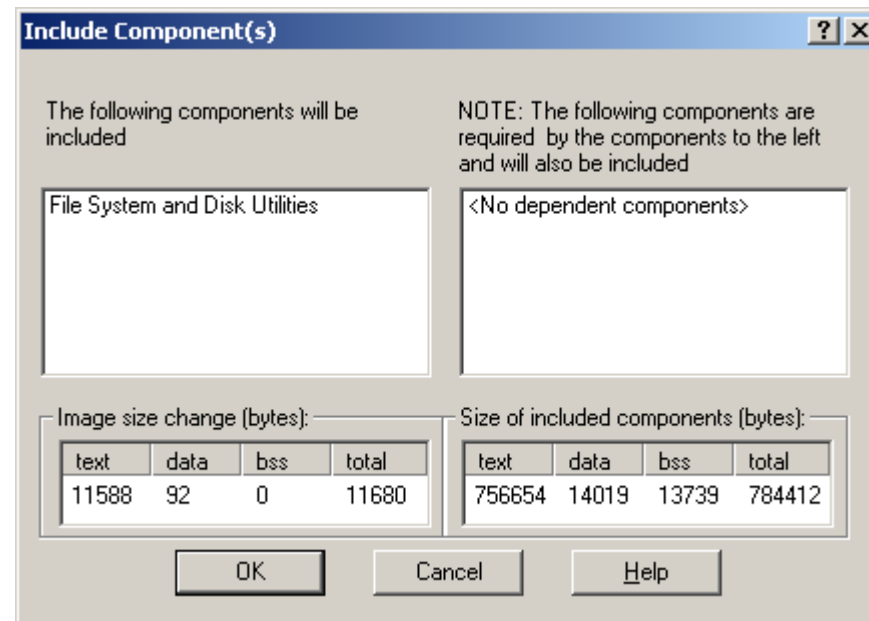
development tool components > show routines
> symbol table show routine



Configure System Image

- Include:

operating system components > IO system components
> File System and Disk Utilities



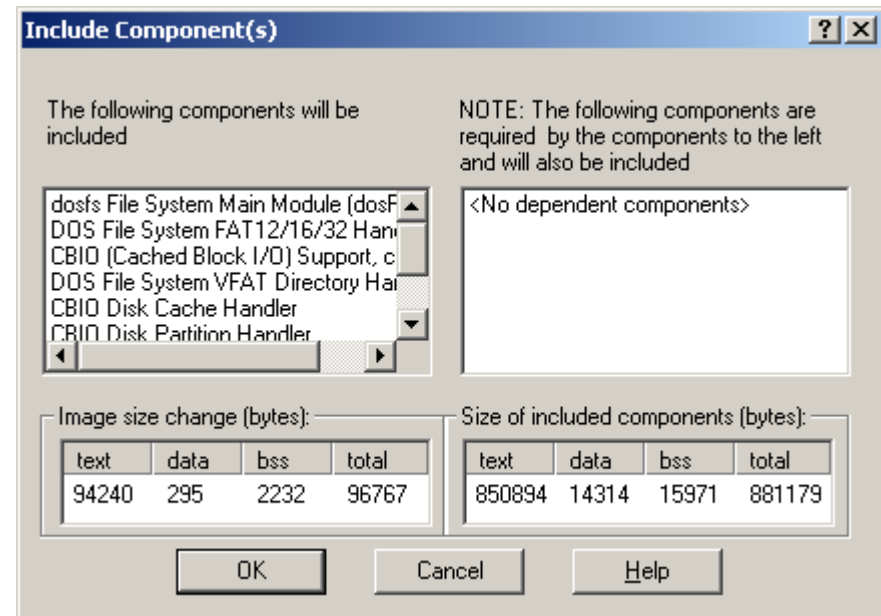
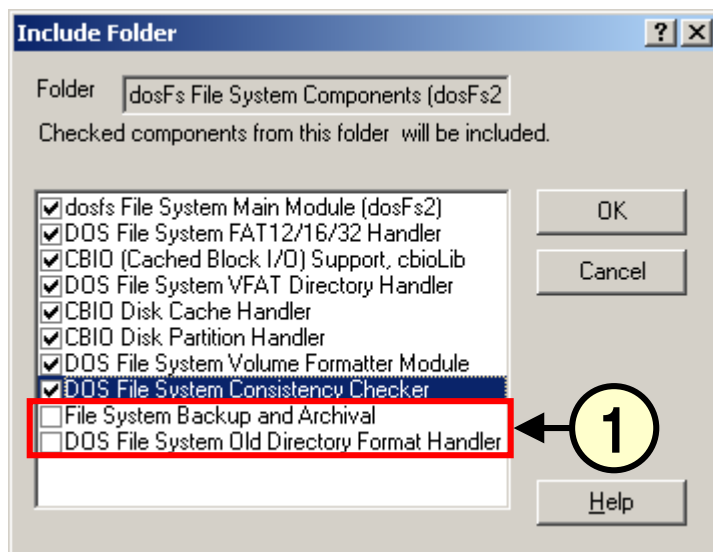
Configure System Image

- Include:

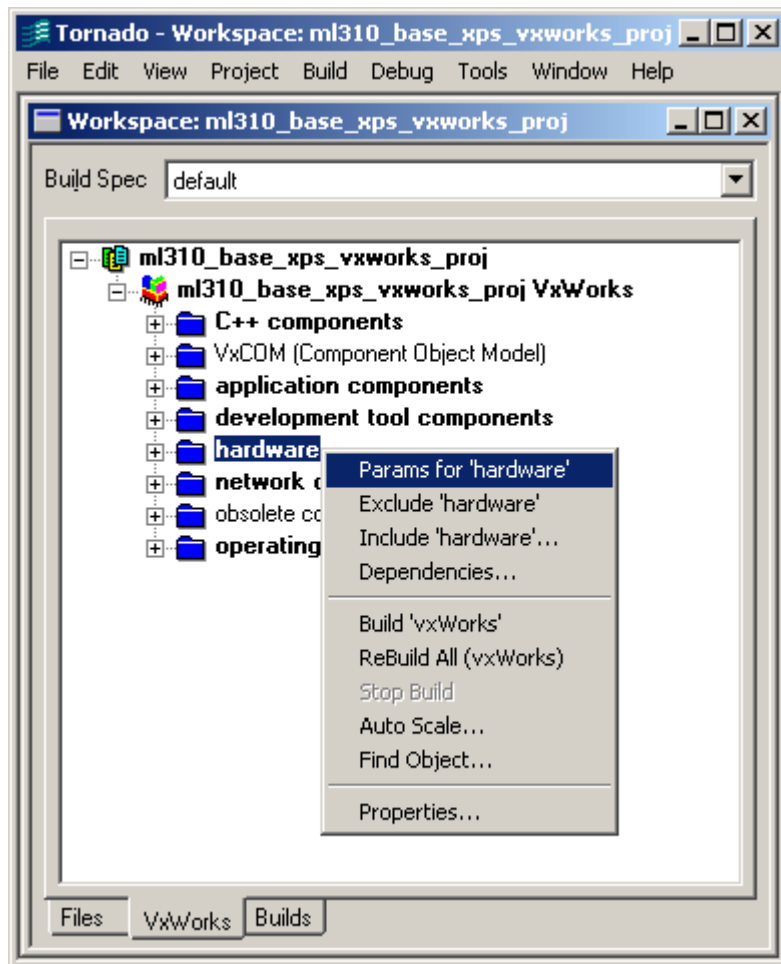
operating system components > IO system components
> dosFs File System Components

– Check **all** boxes, except (1):

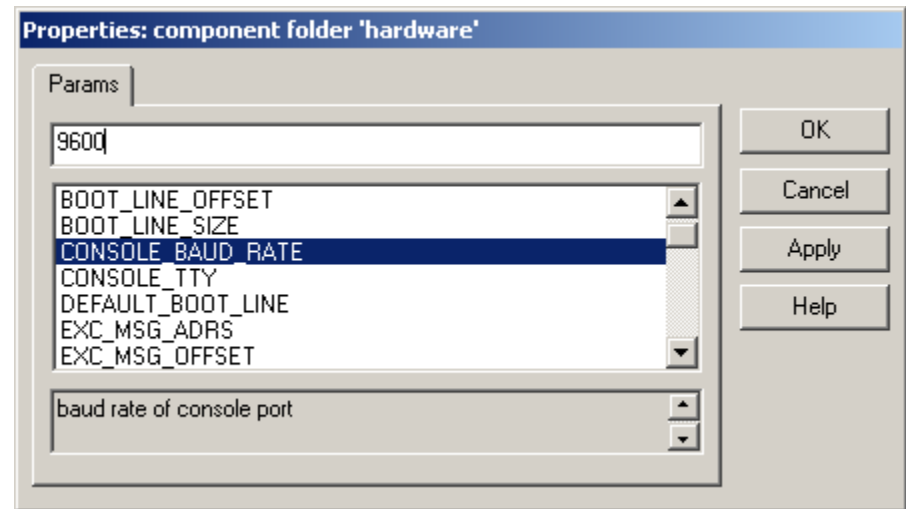
- File System Backup and Archival
- DOS File System Old Directory Format Handler



Configure System Image

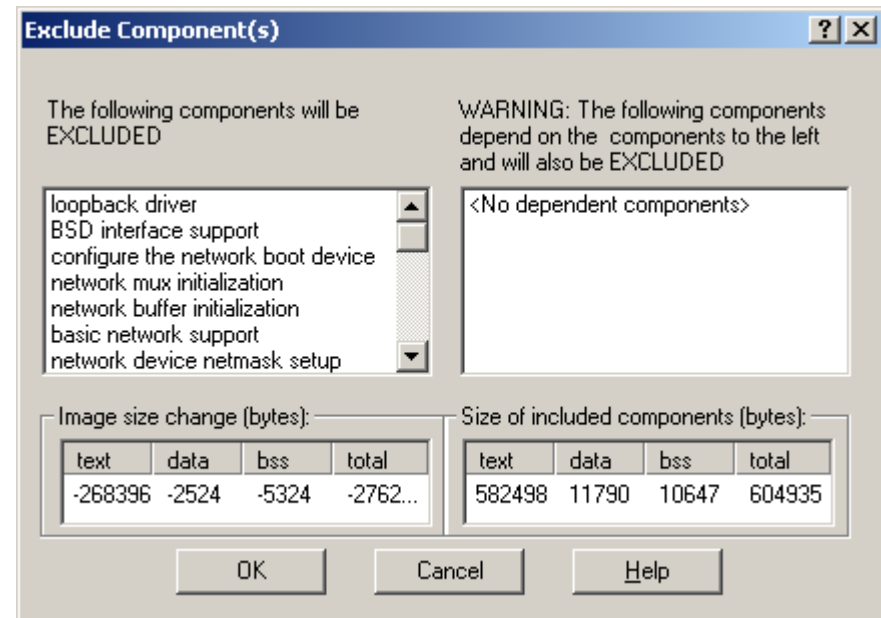
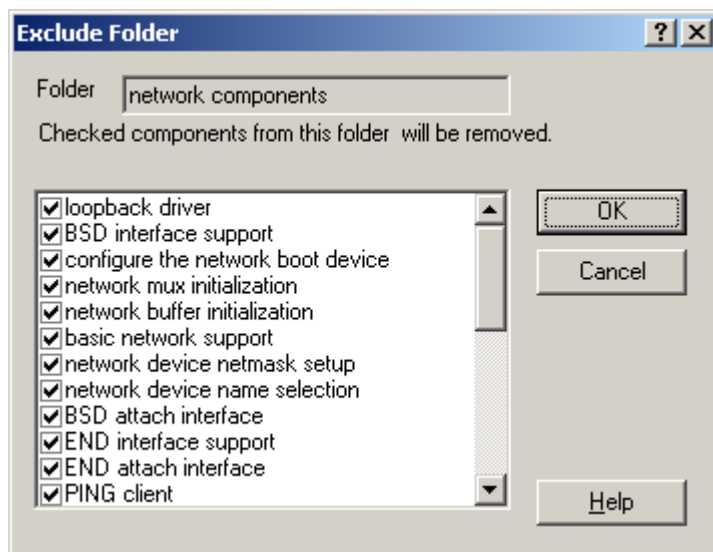


- Set Parameter:
 - Right-click on hardware
- Click on Params for 'hardware'
- Click on `CONSOLE_BAUD_RATE` then enter 9600 to set the baud rate
- Click **OK**



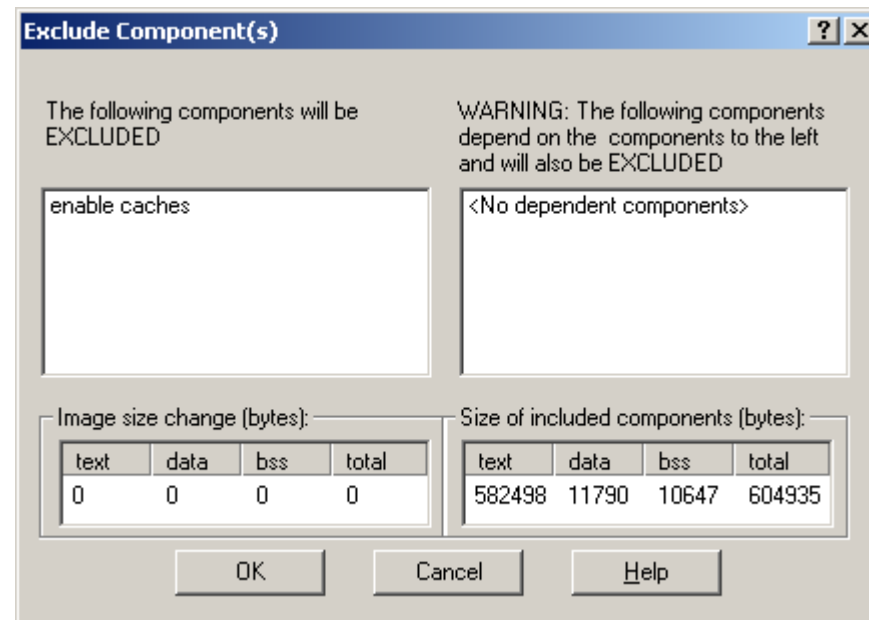
Configure System Image

- Do this step after adding other components
 - Exclude:
 - network components
 - Click **OK**



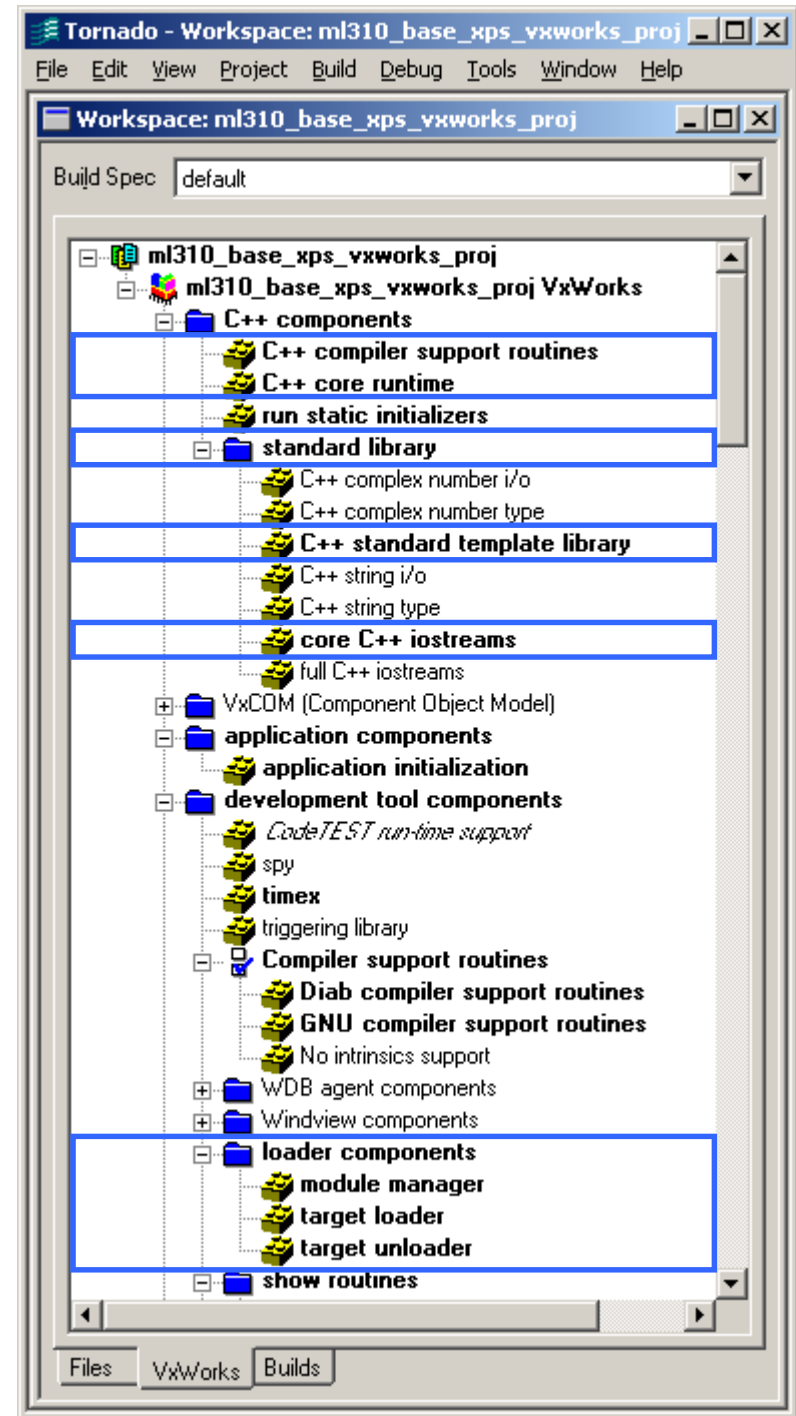
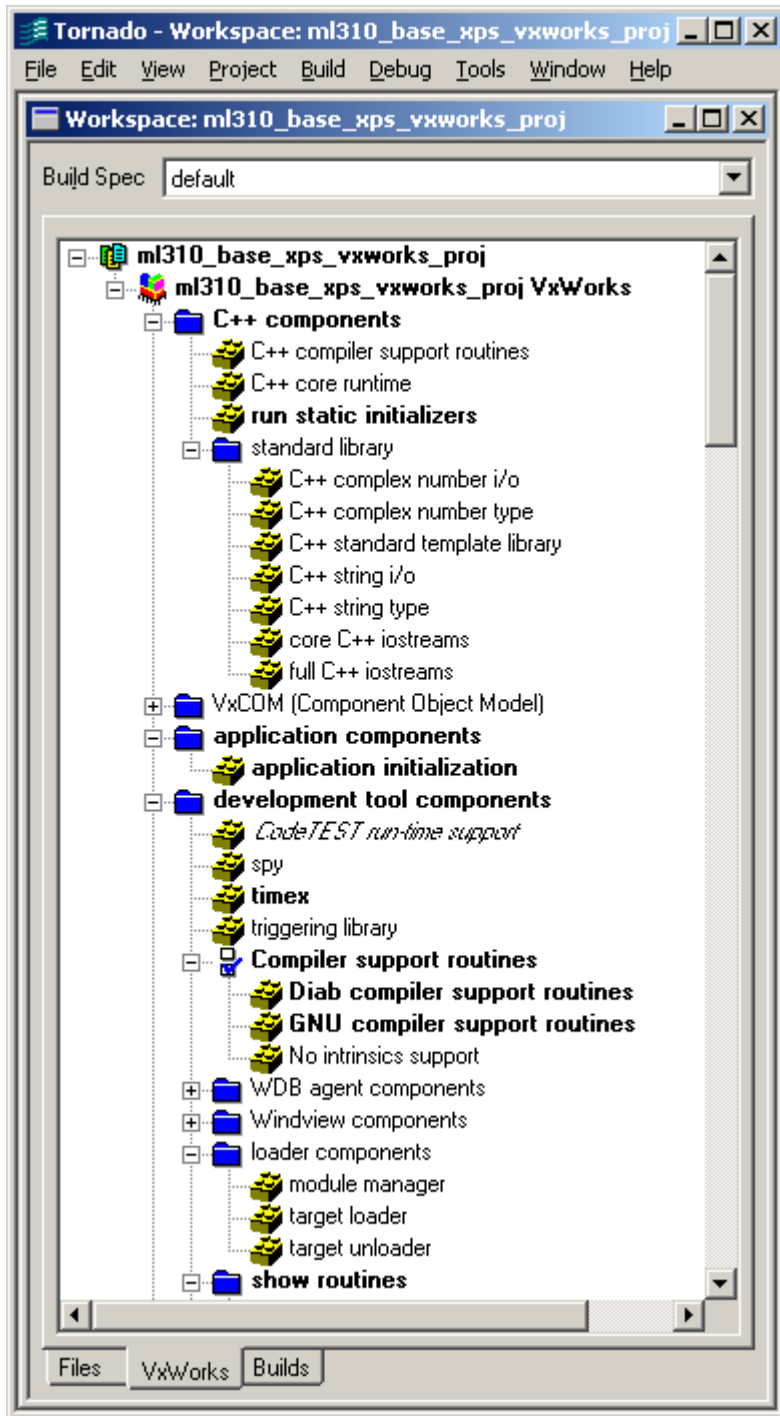
Configure System Image

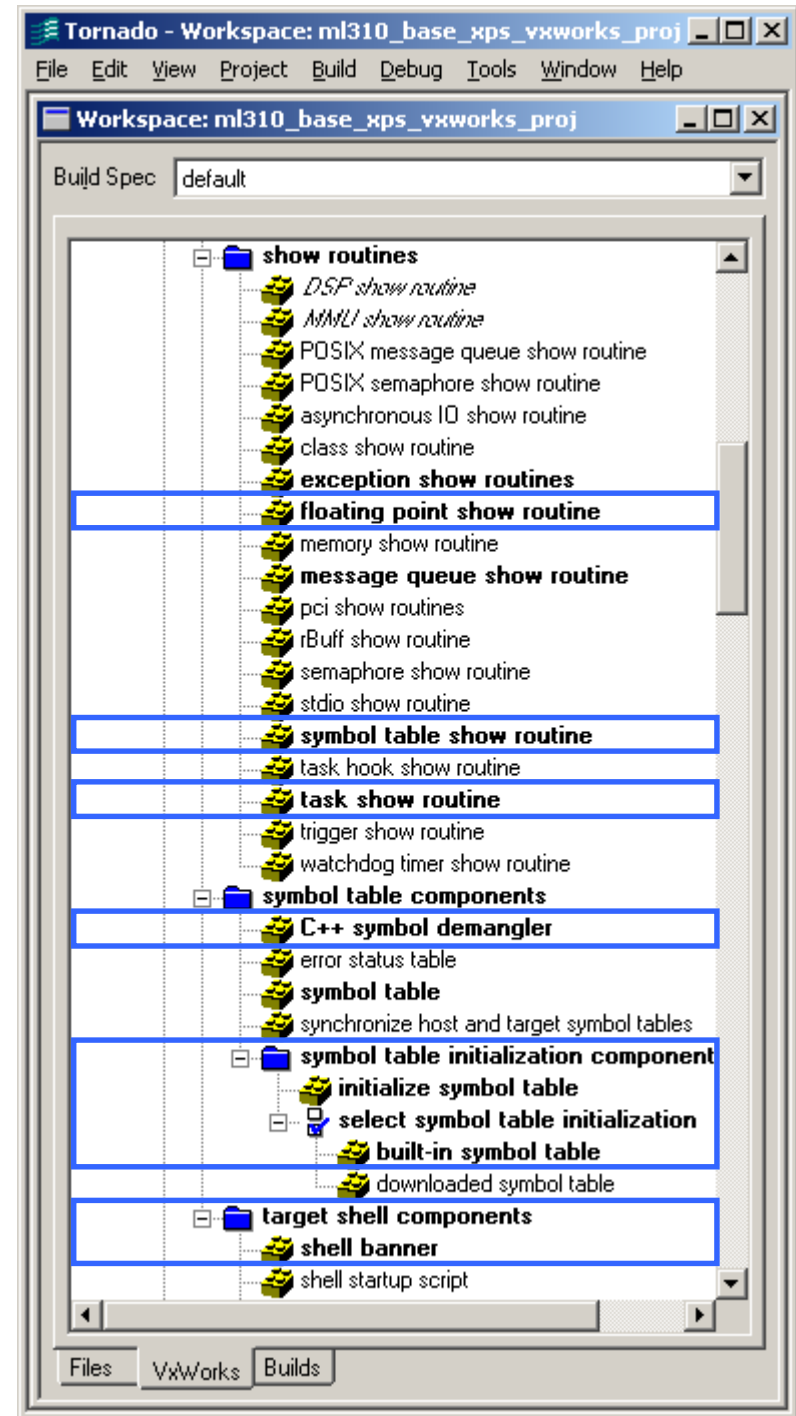
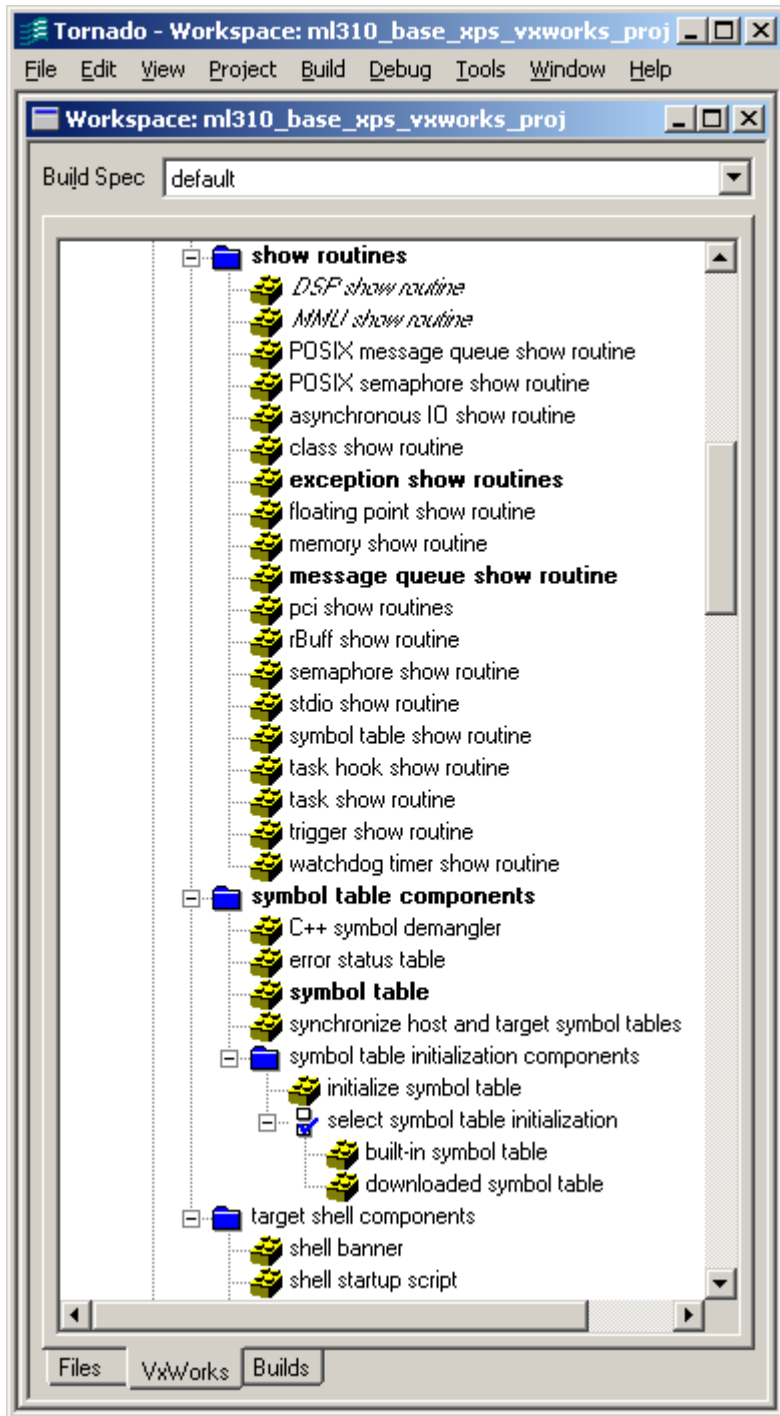
- Do this step last, after adding other components
 - Exclude:
 - hardware > memory > enable caches
 - Click **OK**

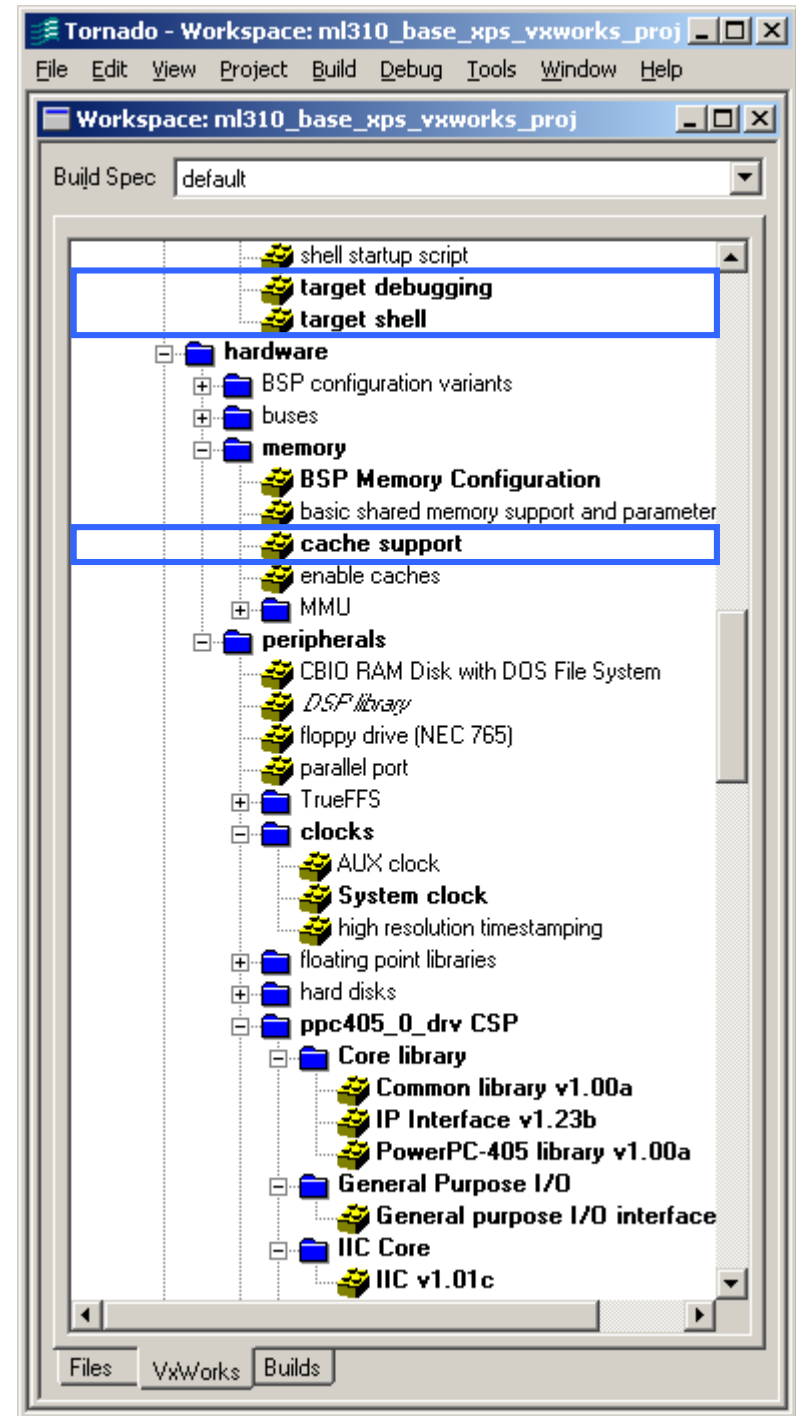
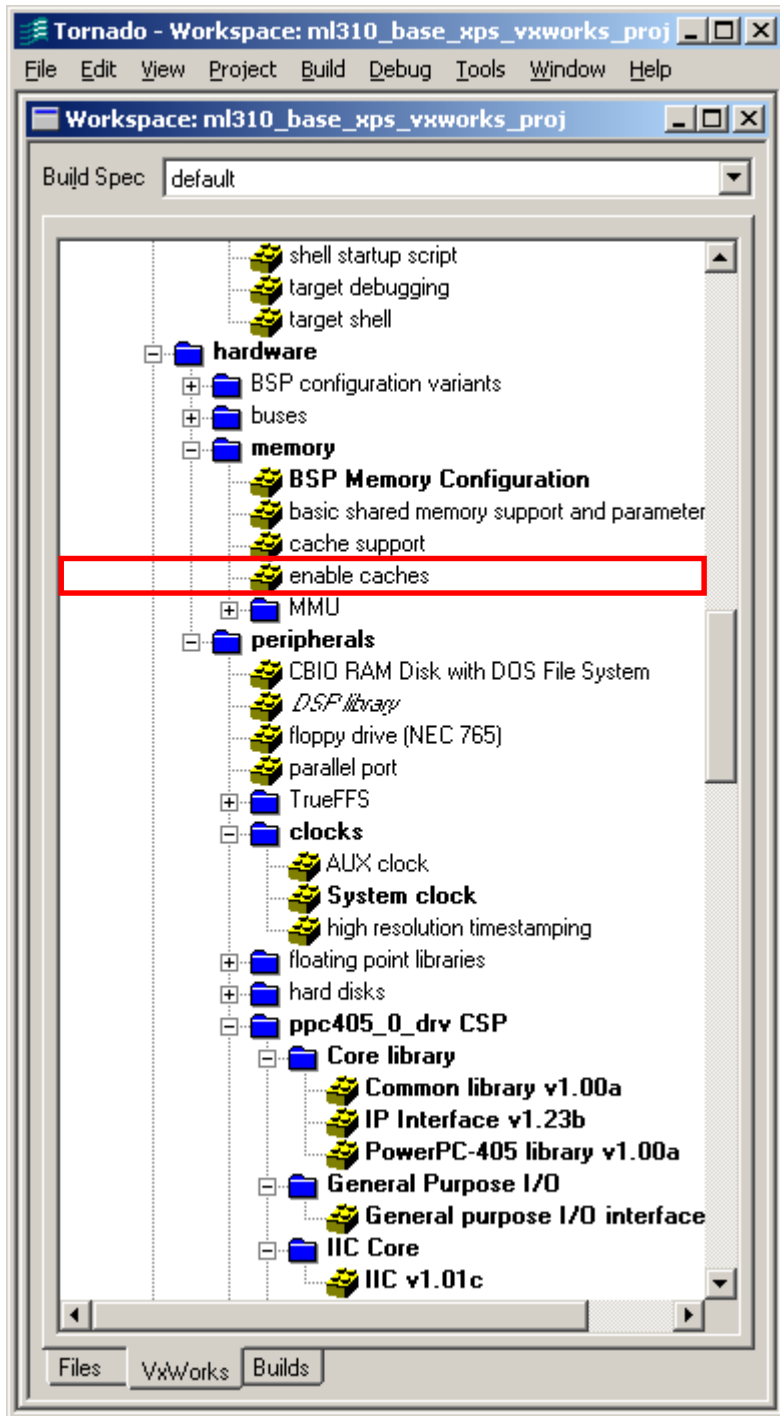


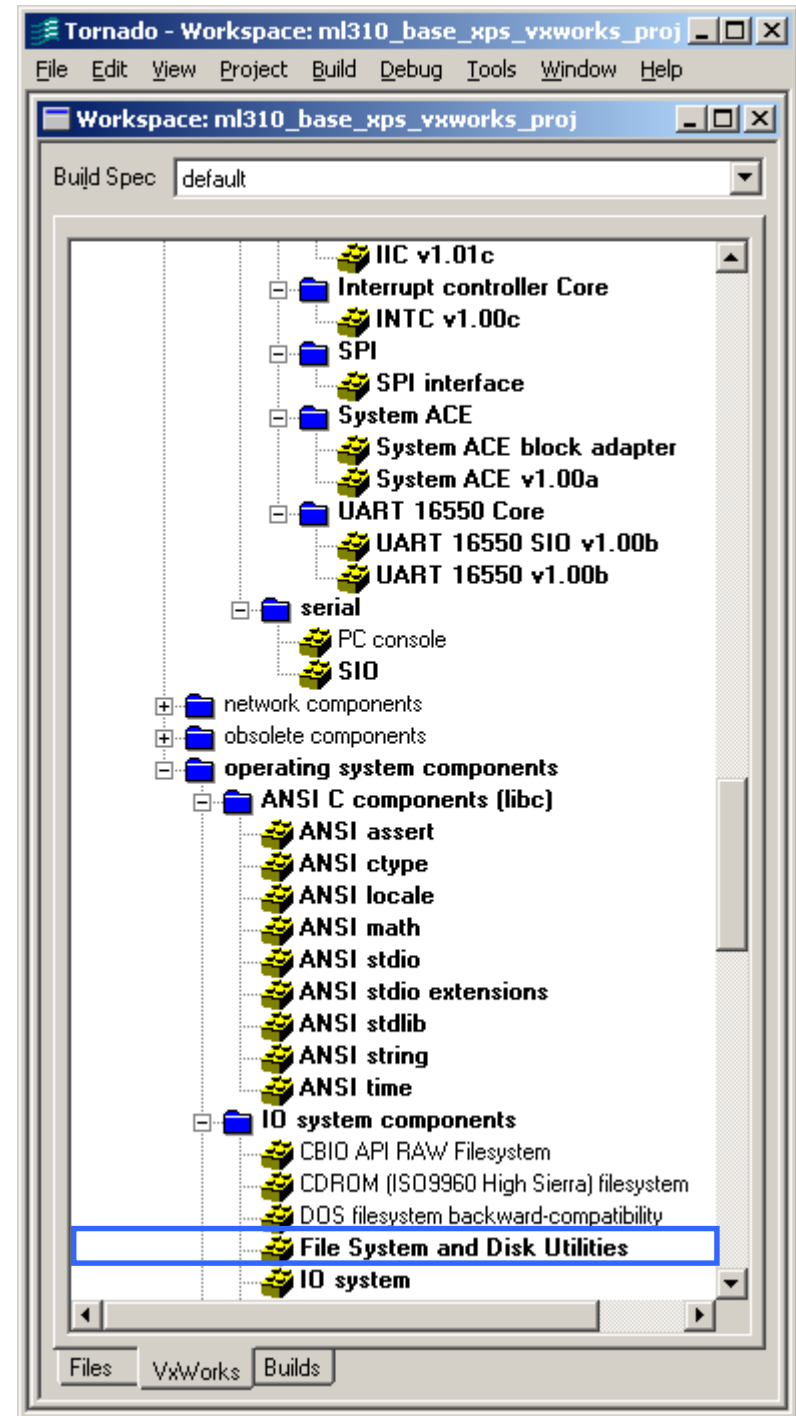
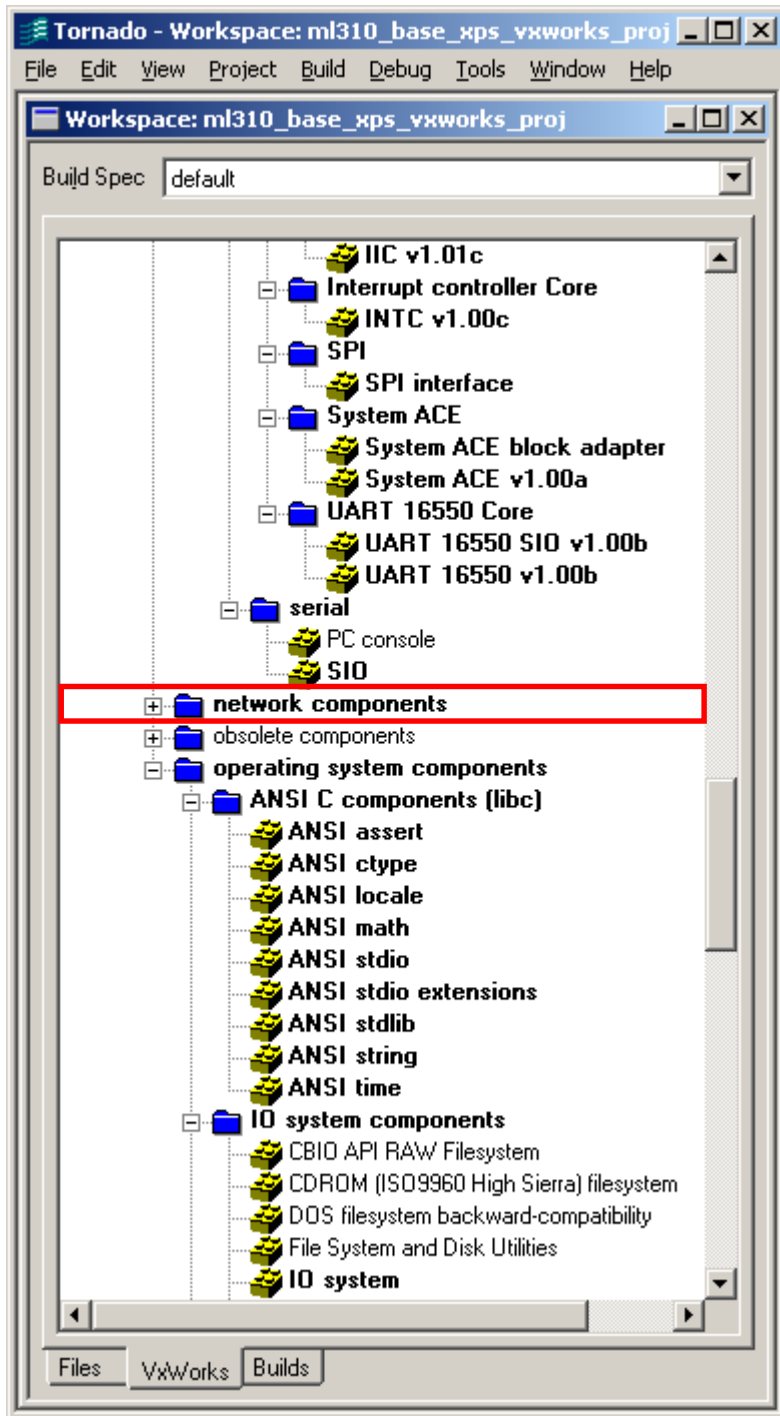
Configure System Image

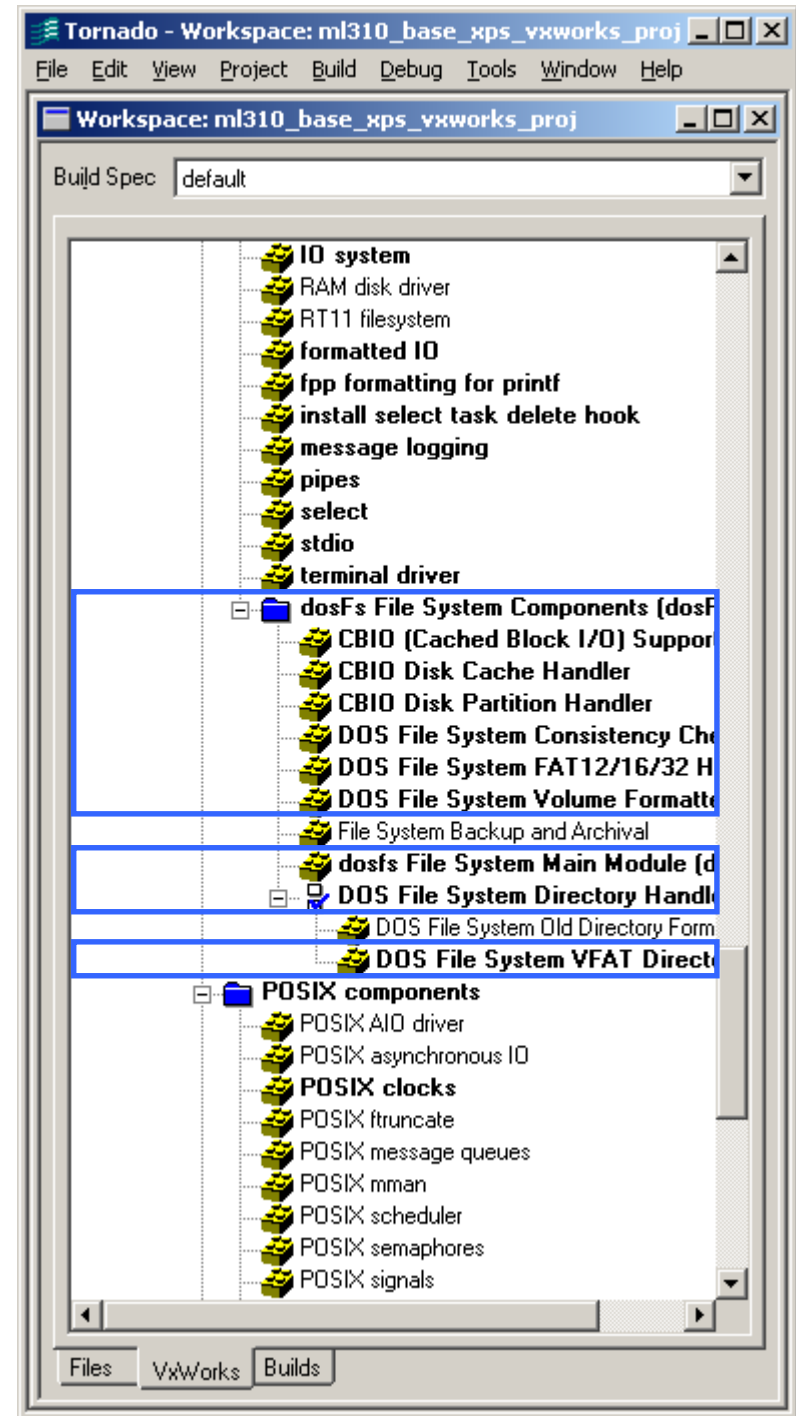
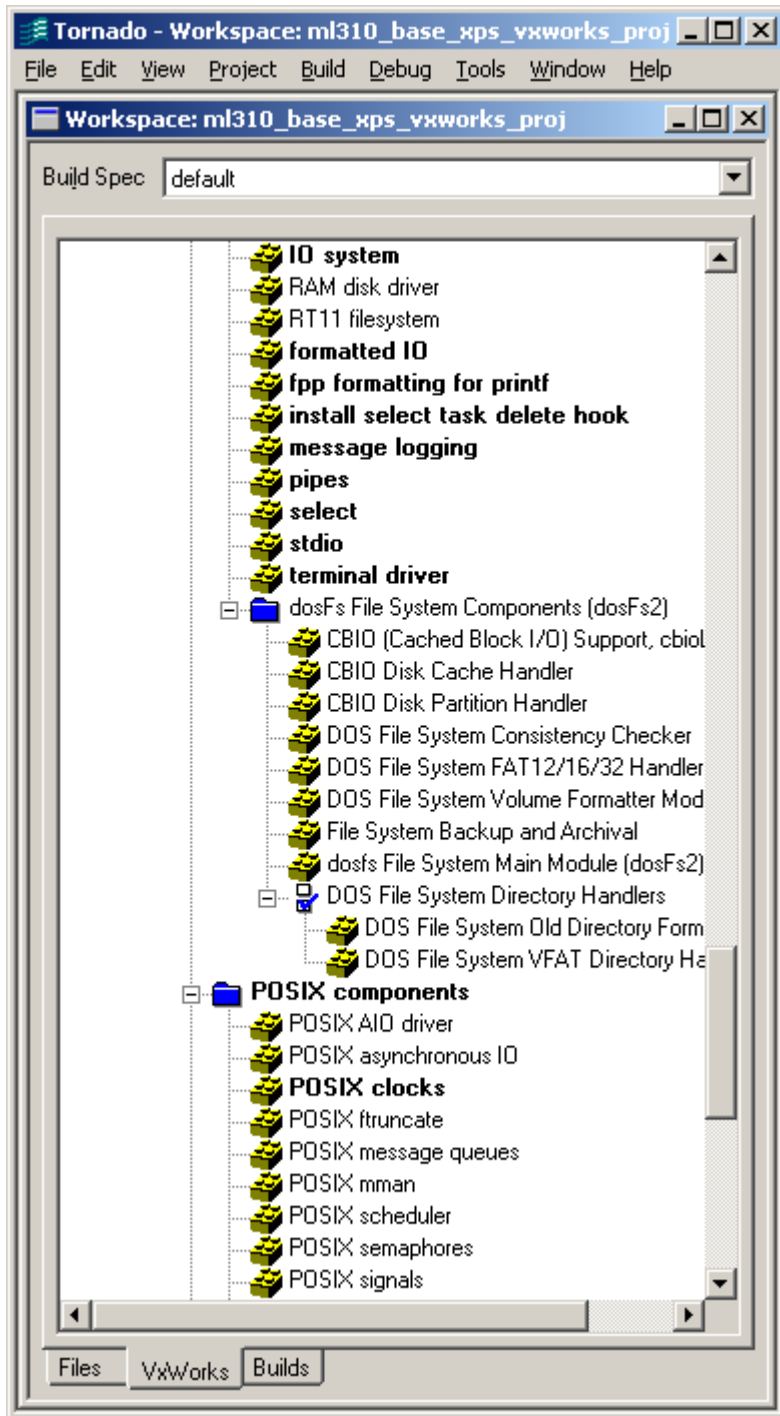
- The next several slides graphically compare the differences between:
 - A default Workspace, based on the BSP (left side)
 - A modified Workspace after system configuration (right side)
- Your included components should match the screenshots on the right
- **Blue** boxes on the right side show components included
- **Red** boxes on the left side show components excluded
 - Either because it was initially included or was added as a result of a dependency
- **Note:** If you have made any errors in configuring the system, the items may be corrected (included or excluded) individually at this time

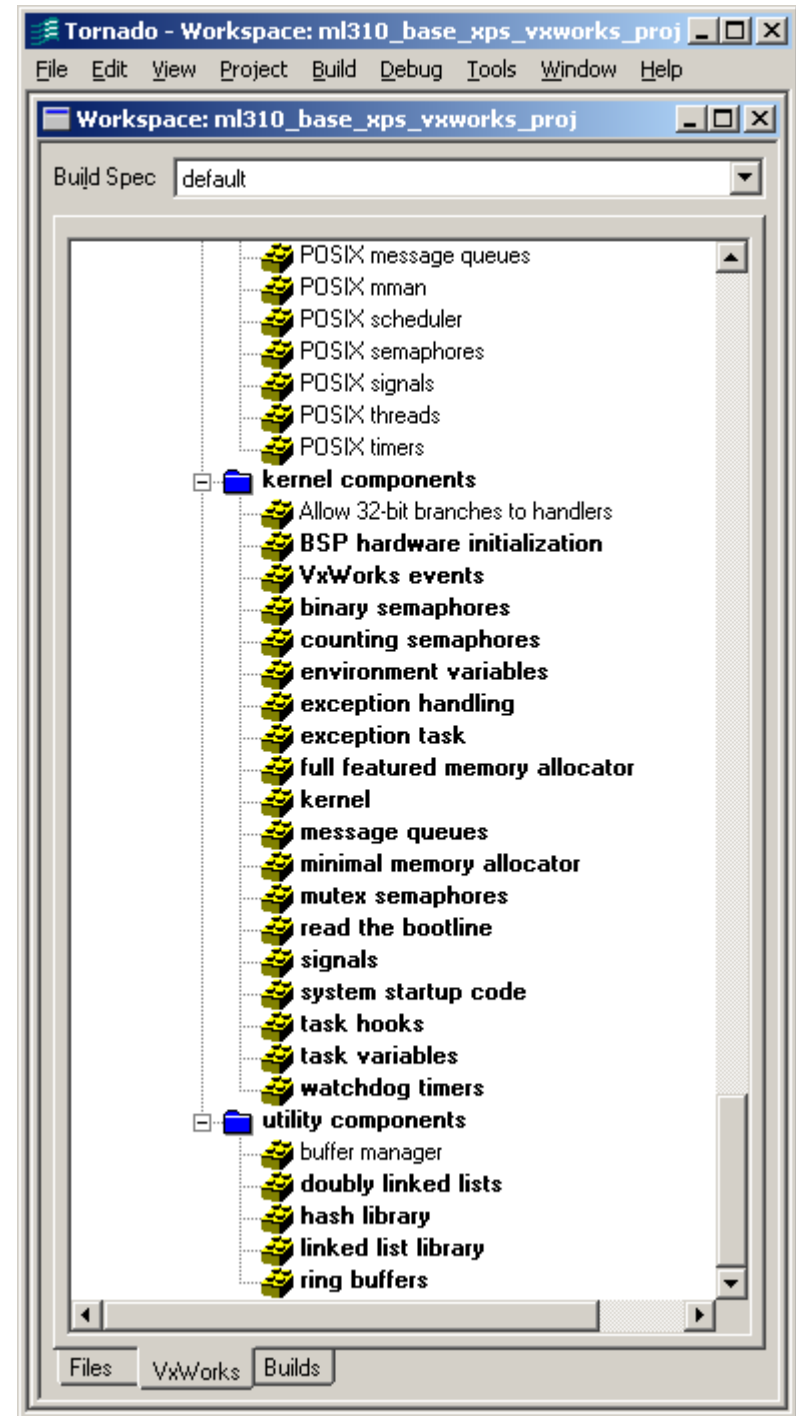
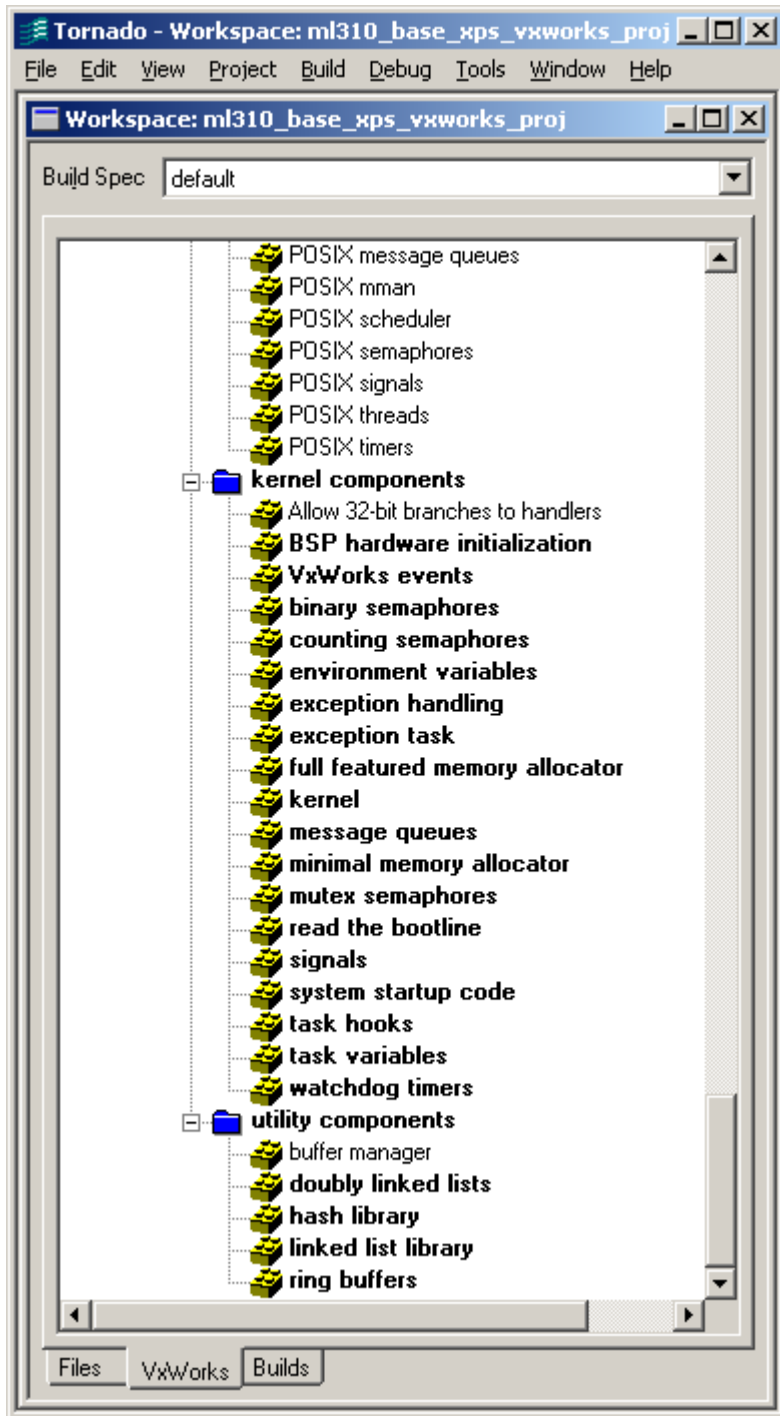






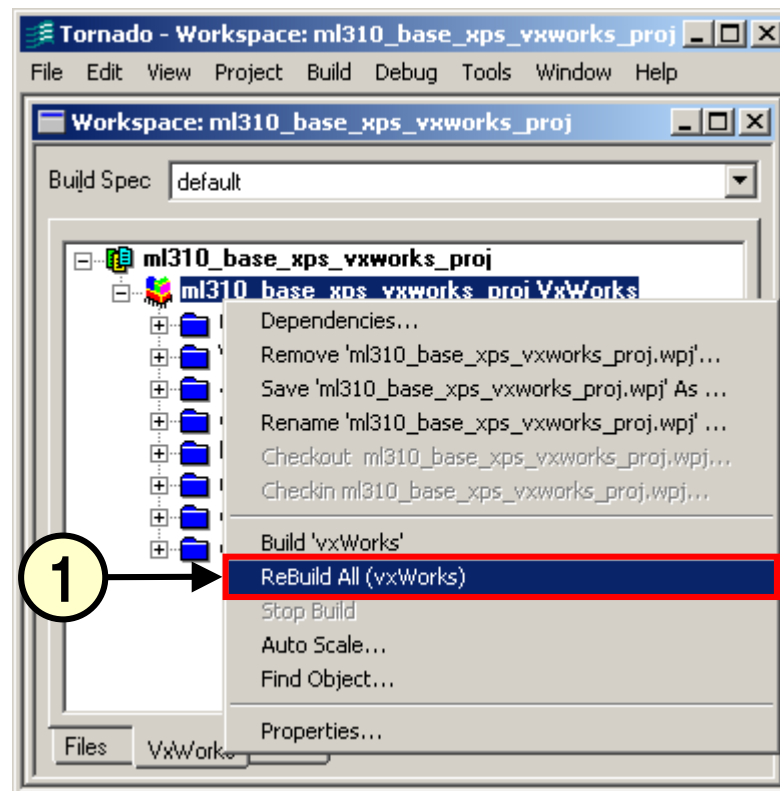






Create VxWorks System Image

- Create a VxWorks system image
 - Perform a “Rebuild All (vxWorks)” (1)



Create VxWorks System Image

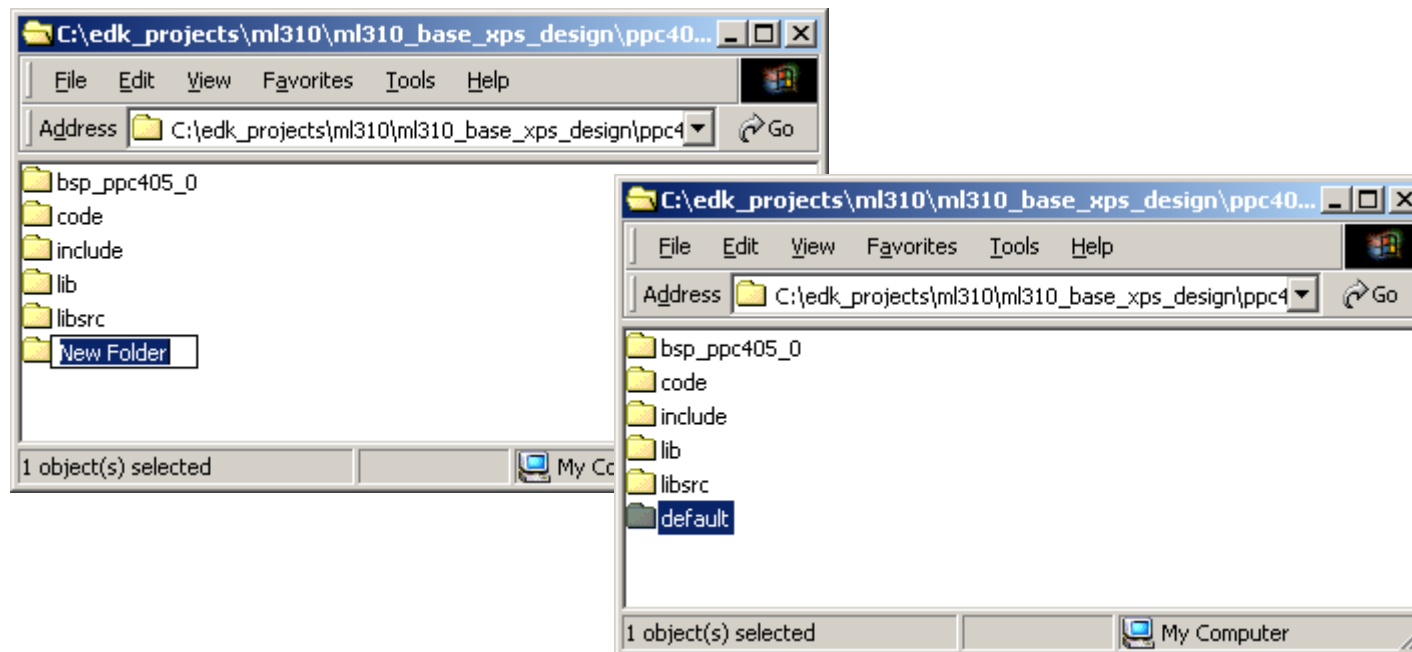
- A successful compile creates a VxWorks ELF file system image

<Tornado Dir>\target\proj\ml310_base_xps_vxworks_proj\default\vxWorks

```
Tornado - [Build Output]
File Edit View Project Build Debug Tools Window Help
ccppc -g -mcpu=405 -mstrict-align -ansi -fno-builtin -I. -IC:\T22ppc405\target\config\ml3
10_base_xps_vxworks_bsp -IC:\T22ppc405\target\h -IC:\T22ppc405\target\config\comps\src -IC
:\T22ppc405\target\src\drv -DCPU=PPC405 -DTOOL_FAMILY=gnu -DTOOL=gnu -DPRJ_BUILD -IC:\T22p
pc405\target\config\ml310_base_xps_vxworks_bsp\ppc405_0_drv_csp\xsrc -DXENV_VXWORKS -g -c
C:\T22ppc405\target\config\comps\src\dataSegPad.c
ldppc -X -N -e _sysInit -Ttext 00010000 \
    dataSegPad.o partialImage.o ctdt.o symTbl.o \
    -defsym wrs_kernel_rom_size=0x0 -T C:\T22ppc405\target\h\tool\gnu\ldscripts\link.DOTBOOTR
AM -o vxWorks
C:\T22ppc405\host\x86-win32\bin\vxsize ppc -v 00200000 00010000 vxWorks
vxWorks: 734160(t) + 87616(d) + 15136(b) = 836912 (1194704 unused)
Done.
```

Create Default Directory

- Create directory
`<edk_proj>\ml310\ml310_base_xps_design\ppc405_0\default`

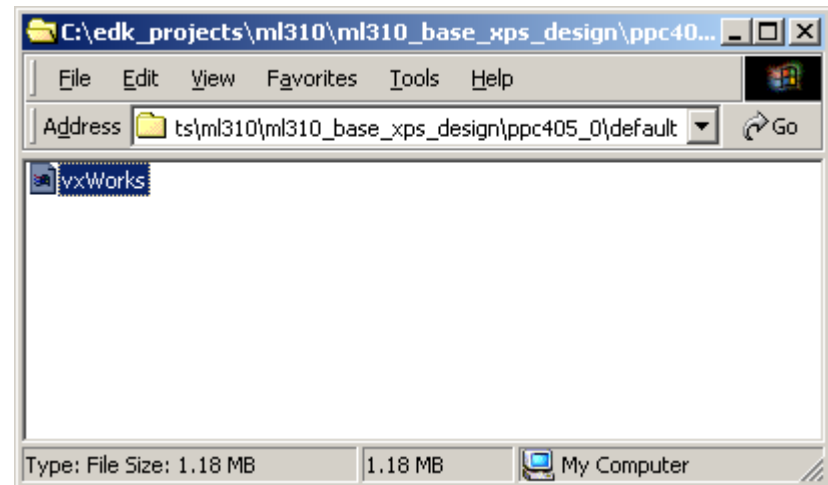
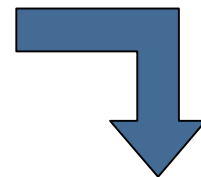
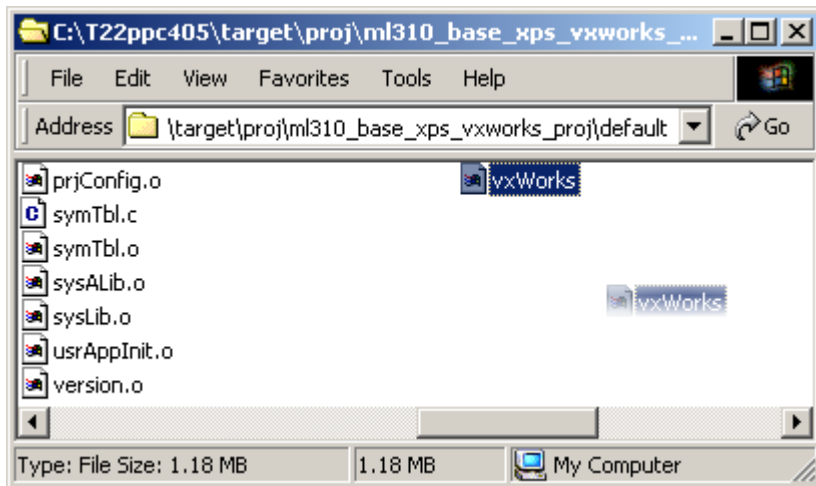


Create VxWorks Image

- Copy *<Tornado Dir>\target\proj\ml310_base_xps_vxworks_proj\default\vxWorks* to:

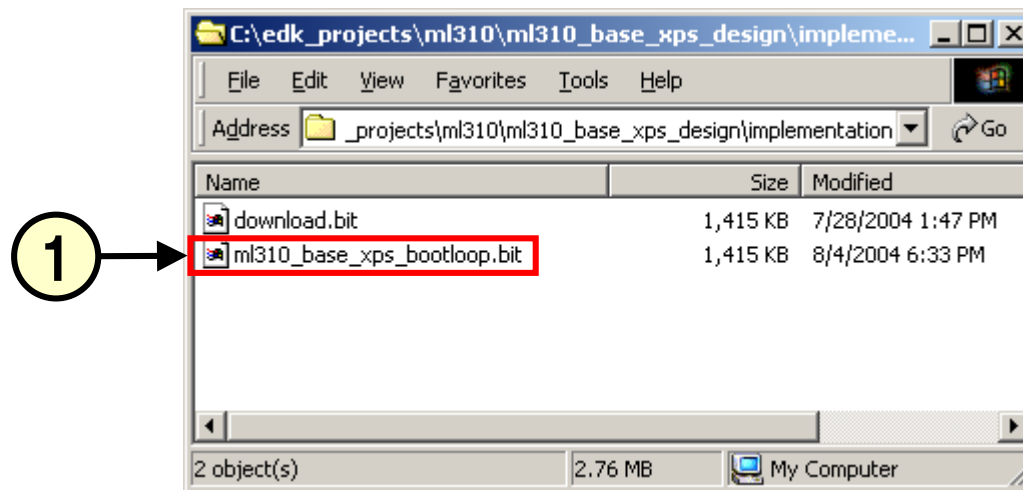
<edk_proj>\ml310\ml310_base_xps_design\ppc405_0\default

- Use **<Ctrl>** key while dragging to make a copy



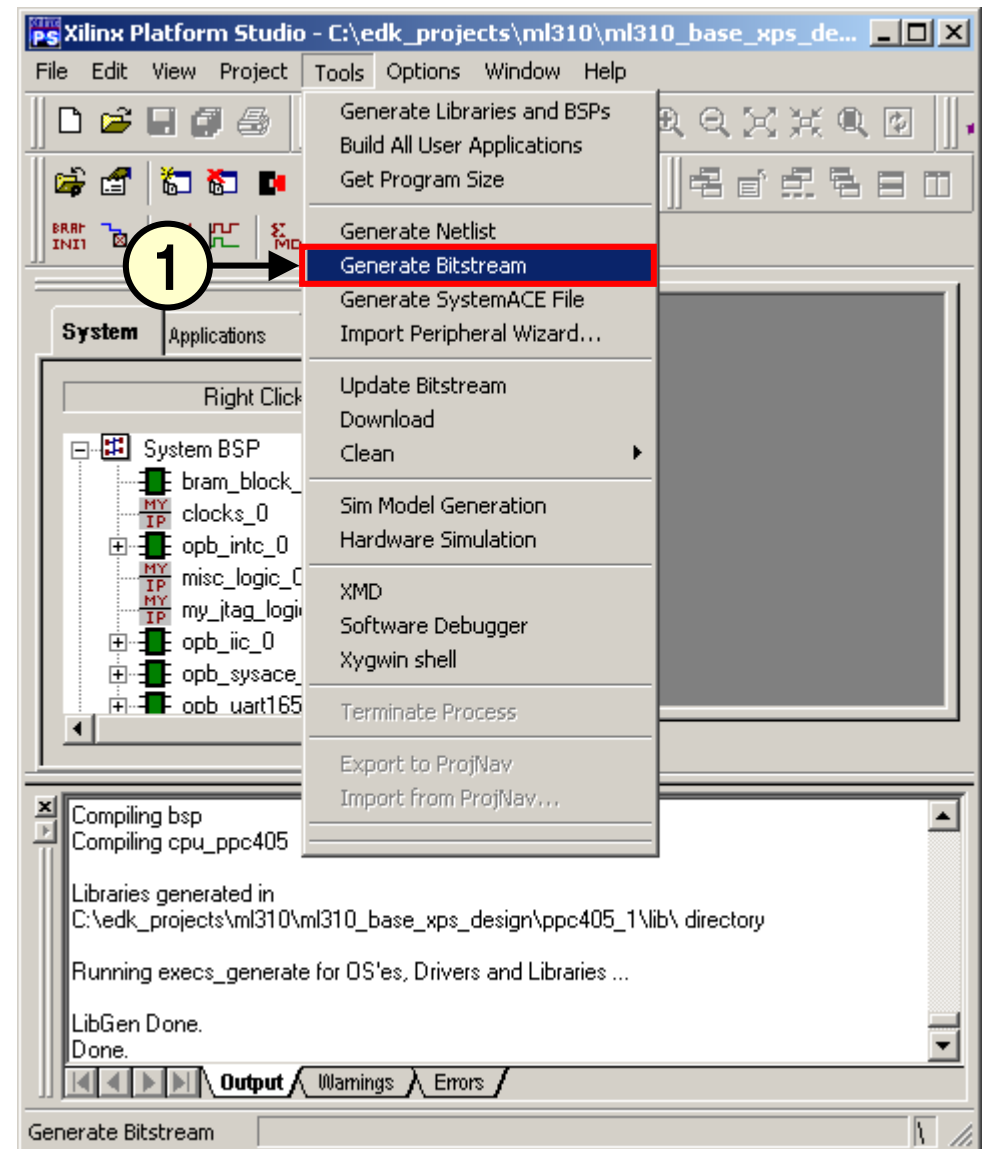
Create an ACE File

- ACE file generation requires a bitstream; the *ml310_base_xps_design.zip* file comes with a precompiled bitstream
 - You can use the existing bitstream (1), and skip the next three slides, **or**
 - You can recompile it as shown on the next three slides



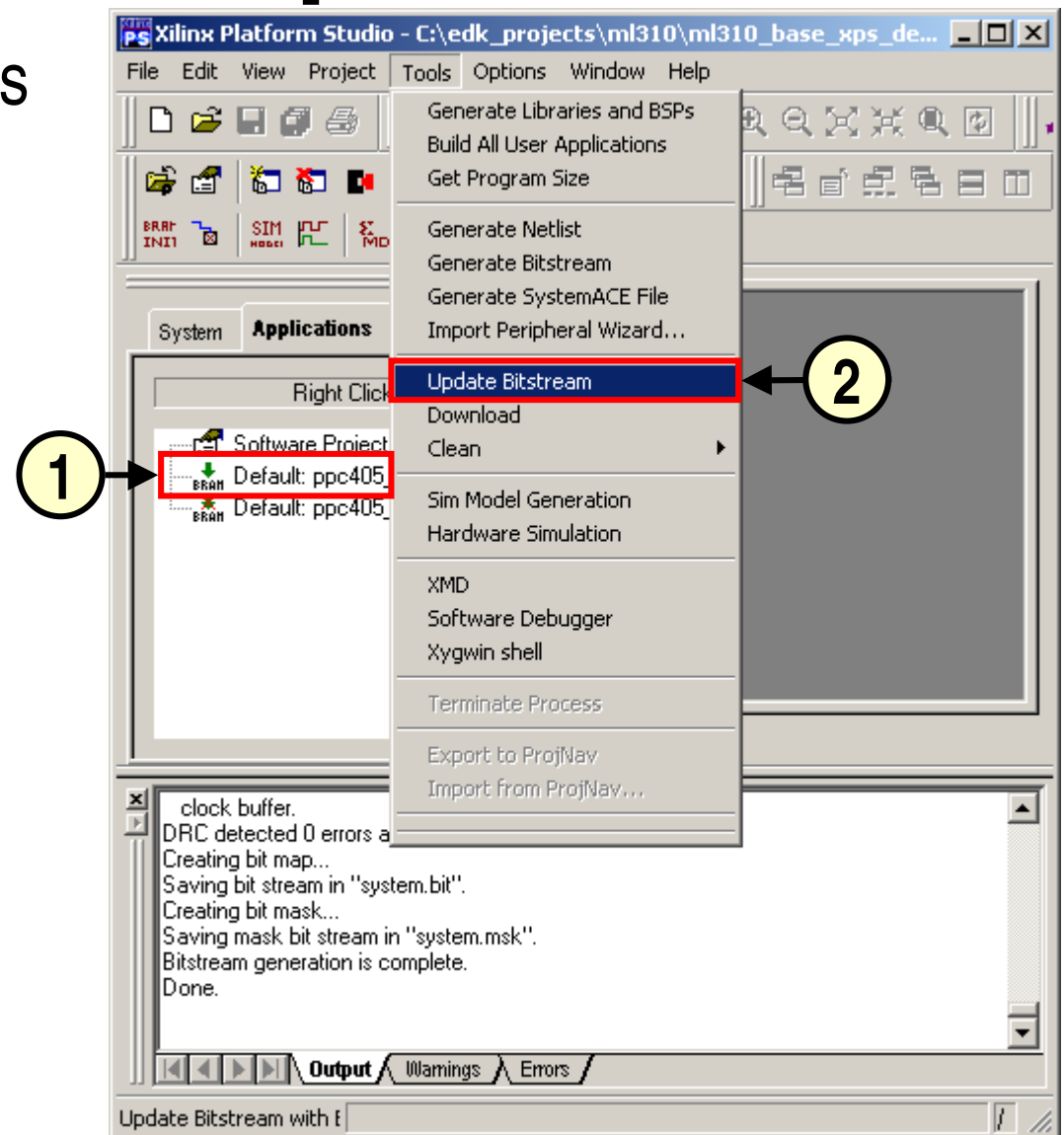
Generate Bitstream

- Invoke the hardware implementation tools
 - From XPS, select **Tools** → **Generate Bitstream** (1)
- Once a bitstream is generated, an ACE file can be created
 - Generate Bitstream takes ~50 min on a Win2K 700MHz PC



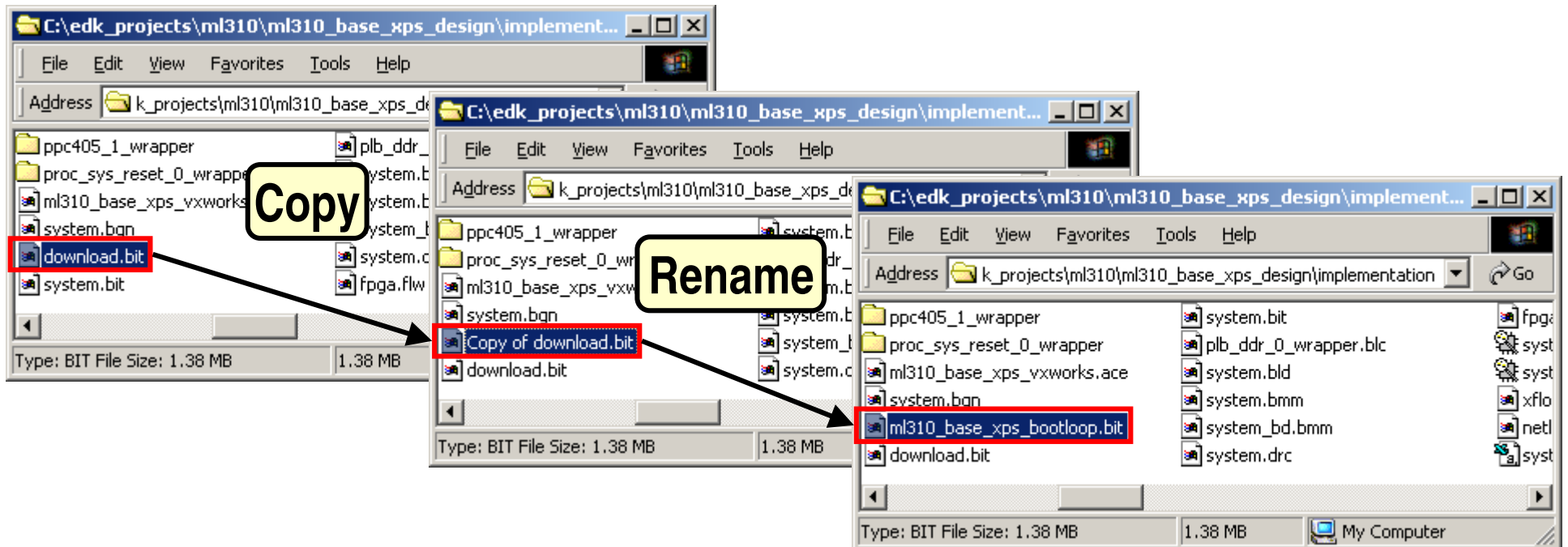
Update Bitstream with Bootloop

- The green arrow indicates the BRAMs are “marked for initialization” with the bootloop code (1)
- From XPS, select **Tools** → **Update Bitstream** (2)
- The updated bitstream *download.bit* will be generated



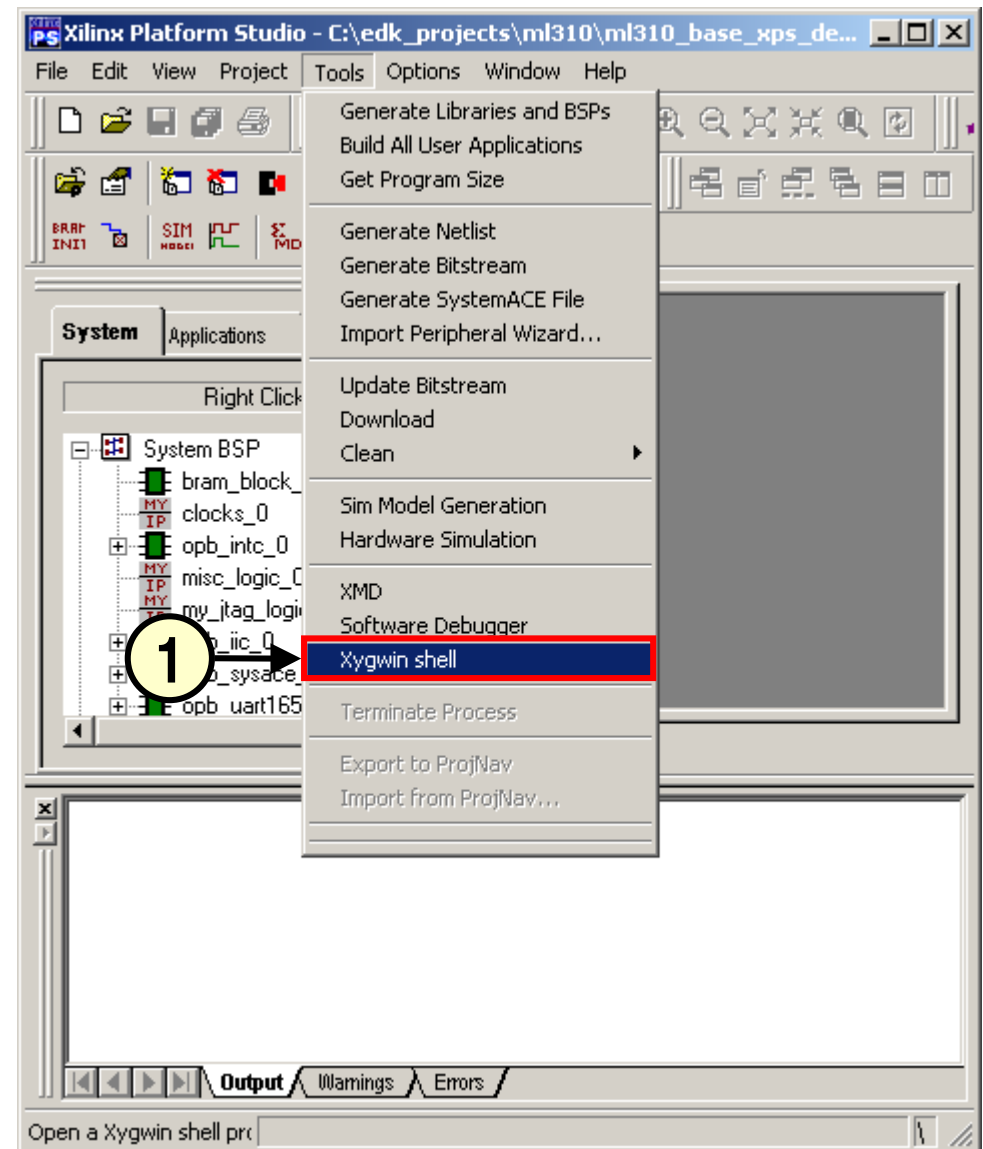
Rename Bitstream

- Make a copy of
`<edk_proj>\ml310\ml310_base_xps_design\implementation\download.bit`
- Rename *Copy of download.bit* to *ml310_base_xps_bootloop.bit*
 - When creating concatenated ACE files (i.e. Bitstream plus separate ELF), we are required to use the bootloop application
 - The new file name indicates the contents of the BRAM and will be used later



Create an ACE File

- Open an Xygwin shell
 - The xygwin shell is used for entering and executing the commands to create a concatenated (HW+SW) ACE file
 - Base XPS VxWorks ACE file creation takes ~10 min on a Win2K 700MHz PC



Create an ACE File

- At the xygwin prompt, type (1):
cd data
./genace_base_xps_vxworks.bat
- This creates a concatenated (HW+SW) ACE file
 - Input: vxworks ELF file, ml310_base_xps_bootloop.bit (2)
 - Output: ml310_base_xps_vxworks.ace (3)

```
C:\D:\EDK_6_2i_Gm_13_6_1\xygwin\bin\bash.exe
bash-2.05$ cd data
bash-2.05$ ./genace_base_xps_vxworks.bat

c:\edk_projects\ml310\ml310_base_xps_design\data>xmd genace.tcl -jprog -hw ../im
plementation/ml310_base_xps_bootloop.bit -elf ../ppc405_0/default/vxworks -ace .
../implementation/ml310_base_xps_vxworks.ace -board ml310
Xilinx Microprocessor Debug (XMD) Engine
Xilinx EDK 6.2.2 Build EDK_Gm.13.6
Copyright (c) 1995-2002 Xilinx, Inc. All rights reserved.
Executing user script : genace.tcl
user sw is ../ppc405_0/default/vxworks
user hw is ../implementation/ml310_base_xps_bootloop.bit
user ace is ../implementation/ml310_base_xps_vxworks.ace
jprog is true
board is ml310

#####
Converting Bitstream '../implementation/ml310_base_xps_bootloop.bit' to SVF file
'../implementation/ml310_base_xps_bootloop_hw.svf'
Executing 'impact -batch bit2svf.scr'
```

Create an ACE File

- *genace_base_xps_vxworks.bat* uses XMD and a *genace.tcl* script with ML310 appropriate options to generate an ACE file (1)

```
C:\D:\EDK_6_2i_Gm_13_6_1\xygwin\bin\bash.exe

JTAG chain configuration
-----
Device   ID Code      IR Length  Part Name
  1      0127e093      14         xc2vp30

XMD: Connected to PowerPC target. Processor Version No : 0x00000000
Address mapping for accessing special PowerPC features from XMD/GDB:
I-Cache <Data> : Start Address - 0x70000000
I-Cache <Tag>  : Disabled
D-Cache <Data> : Start Address - 0x78000000
D-Cache <Tag>  : Disabled
ISOCM        : Start Address - 0x50000000
TLB          : Disabled
DCR          : Disabled
Processor started. Type "stop" to stop processor

#####
Converting SVF file '../implementation/ml310_base_xps_vxworks.svf' to SystemACE
file '../implementation/ml310_base_xps_vxworks.ace'
Executing 'impact -batch svf2ace.scr'

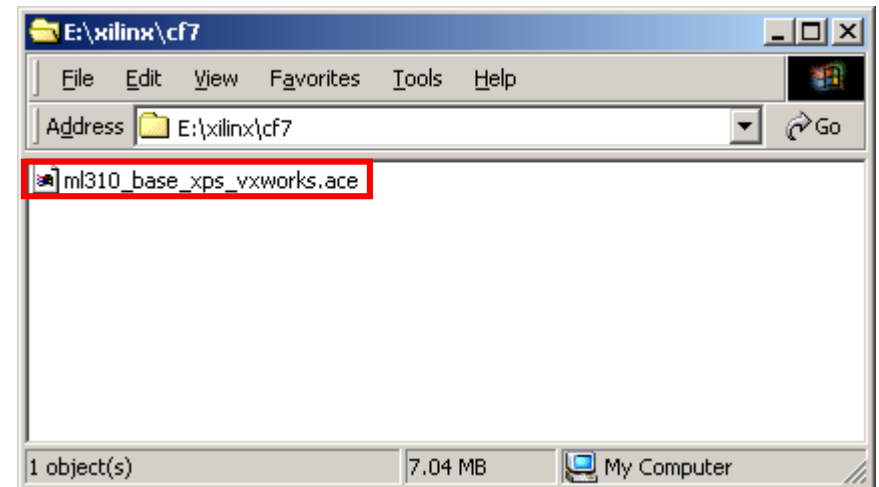
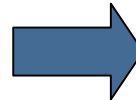
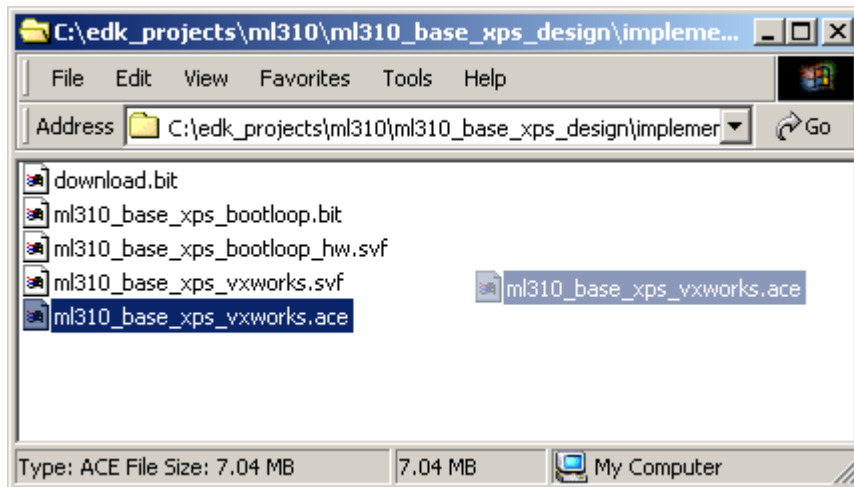
SystemACE file '../implementation/ml310_base_xps_vxworks.ace' created successfully
bash-2.05$
```

1



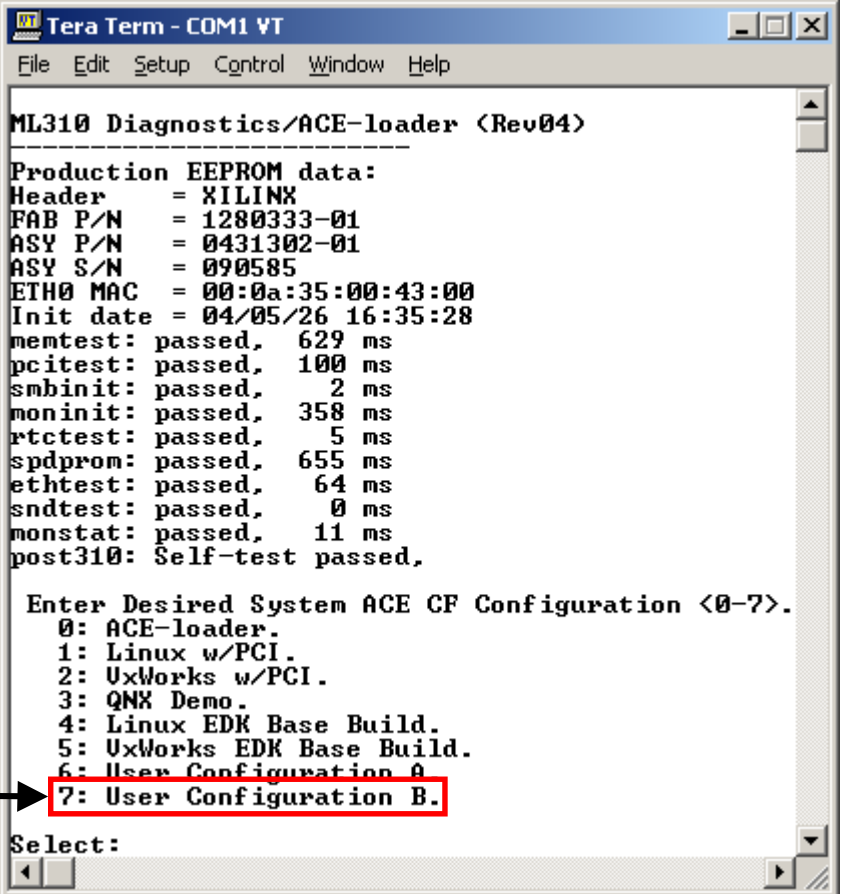
Run ACE File

- Copy *ml310_base_xps_vxworks.ace* to *xilinx\cf7* directory on your CompactFlash card
 - **Important:** Delete any existing ace files in this cf7 directory
 - **Note:** Use a CompactFlash reader to mount the CompactFlash as a disk drive



Run ACE File

- Eject the CompactFlash from your PC and insert it back into the ML310, open a new terminal window, and power-up the board
- Type 7, for User Configuration B (1) to run the newly created ACE file and load vxWorks



```
Tera Term - COM1 VT
File Edit Setup Control Window Help
ML310 Diagnostics/ACE-loader <Rev04>
-----
Production EEPROM data:
Header = XILINX
FAB P/N = 1280333-01
ASY P/N = 0431302-01
ASY S/N = 090585
ETH0 MAC = 00:0a:35:00:43:00
Init date = 04/05/26 16:35:28
mentest: passed, 629 ms
pcitest: passed, 100 ms
smbinit: passed, 2 ms
moninit: passed, 358 ms
rtctest: passed, 5 ms
spdprom: passed, 655 ms
ethctest: passed, 64 ms
sndtest: passed, 0 ms
monstat: passed, 11 ms
post310: Self-test passed.

Enter Desired System ACE CF Configuration <0-7>.
0: ACE-loader.
1: Linux w/PCI.
2: UxWorks w/PCI.
3: QNX Demo.
4: Linux EDK Base Build.
5: UxWorks EDK Base Build.
6: User Configuration A.
7: User Configuration B.
Select:
```


Target Shell Task Listing

- After the VxWorks banner screen, type `i` (1)
- View a list of tasks running on the ML310 (2)

1 → `-> i`

```
Development System
UxWorks version 5.5.1
KERNEL: WIND version 2.6
Copyright Wind River Systems, Inc., 1984-2003

CPU: ppc405_0 UirtexII Pro PPC405. Processor #0.
Memory Size: 0x200000. BSP version 1.2/0.
```

NAME	ENTRY	TID	PRI	STATUS	PC	SP	ERRNO	DELAY
tExcTask	excTask	1f8ec0	0	PEND	a7528	1f8da0	0	0
tLogTask	logTask	1f64d8	0	PEND	a7528	1f63c8	0	0
tShell	shell	1f45f0	1	READY	a001c	1f41d0	0	0

2 →