BSP Validation Checklist

This document contains the BSP Validation Checklist that Wind River uses when testing and validating a new BSP. Use the BSP Validation Checklist as a guide through the verification of all aspects of the BSP product: installation, testing, and package verification.

BSP Validation Checklist		
Architecture Product ID:	**BSP Product ID:**	
Architecture Patches:		
Optional Product 1:		
Optional Product 2:		
Optional Product 3:		
Functional Test		
Item	**Date**	**Status**
1. Install all relevant patches to a fresh copy of the architecture product.		
2. Install the BSP product.		
3. Make a copy of the installed BSP directory. Make a copy of the installed project directory for this BSP. The project for each BSP is named *BSP_toolchain* and can be found in the **target/proj** directory.		
4. Test all installed images (a simple test of basic functionality). This includes all images installed in the project directory.		
5. Rebuild the BSP components. Make sure all images build correctly. Compare the new and old images for any size differences.		
vxWorks		
vxWorks.st		
vxWorks_rom		
vxWorks.st_rom		
vxWorks.res_rom		
vxWorks.res_rom_res_low		
bootrom		
bootrom_uncmp		
bootrom_res		
6. Rebuild all the project facility images for this BSP. Make sure all images build correctly for both GNU and Diab toolchains. Compare the new and old images for any size differences.		
GNU		
vxWorks		
vxWorks_rom		
vxWorks_romCopy		
vxWorks_romResident		

BSP Validation Checklist		
Diab		
vxWorks..	....	....
vxWorks_romCopy	....	....
vxWorks_romCopy	....	....
vxWorks_romResident	....	....
Build Test		
Item	**Date**	**Status**
1. Build and test any standard images not previously built and tested. Verify that any images that fail to build or run are noted in **target.nr** under the **SPECIAL CONSIDERATIONS** heading.		
vxWorks	....	....
vxWorks.st	....	....
vxWorks_rom	....	....
vxWorks.st_rom	....	....
vxWorks.res_rom	....	....
vxWorks.res_rom_nosym	....	....
vxWorks.res_rom_res_low	....	....
vxWorks.res_rom_nosym_res_low	....	....
bootrom	....	....
bootrom_uncmp	....	....
bootrom_res	....	....
bootrom_res_high	....	....
2. Build and test the following images for the GNU toolchain from the project facility. Verify that any images that fail to build or run are noted in **target.nr** under the **SPECIAL CONSIDERATIONS** heading.		
vxWorks	....	....
vxWorks_rom	....	....
vxWorks_romCopy	....	....
vxWorks_romResident	....	....
3. Build and test the following images for the Diab toolchain from the project facility. Verify that any images that fail to build or run are noted in **target.nr** under the **SPECIAL CONSIDERATIONS** heading.		
vxWorks	....	....
vxWorks_rom	....	....
vxWorks_romCopy	....	....
vxWorks_romResident	....	....
4. Check for the inline assembly macro **__asm__** in all C files of the BSP. All files that use this macro should use the new macro, **_WRS_ASM**. Please see the *BSP Developer's Guide: Portable C Coding Standard* for more details.		

BSP Validation Checklist

Product Packaging Test

Item	Date	Status
1. Generate a list of all files installed by the BSP product.		
2. Verify that the BSP does not deliver any WRS generic driver header files, source files, or object modules.		
3. Verify that the dependency file **depend.***bspname* is not part of the list of files installed by the BSP product.		
4. (Optional) Verify that the BSP does not deliver any unnecessary deliverables such as object files (all **.o** files), **ctdt.c**, or **symTbl.c** in either the BSP or project directories.		
5. Check for the correct version of the BSP. The valid BSP version numbers are VxWorks 5.2 BSP or earlier version - **1.0/***x* Tornado 1.0 and 1.0.1 BSP version - **1.1/***x* Tornado 2.X BSP version - **1.2/***x* where *x* is the BSP revision number. Verify that the **README** file in the BSP directory has the correct version number. Verify that the macros **BSP_VERSION** and **BSP_REV** in **config.h** also reflect the same BSP version.		
6. Verify that there are no third-party files in **target/h**, **target/src/drv/**, **target/config/all**, or **target/src/config**.		
7. Verify that any special drivers unique to the BSP are part of **MACH_EXTRA** or **sysLib.c** (either literally or pulled in by a **#include** statement).		
8. Verify that the BSP contains no copyrighted files or any files derived from restricted Wind River sources. Wind River does not release the source code for Ethernet or SCSI drivers to users.		

Configuration Test

Item	Date	Status
1. Test each boot method supported by the default BSP configuration. Install and boot the boot ROM and check that it can download an image using each boot method. Also for each boot method, attempt to download an invalid VxWorks image (a nonexistent image or one in an improper format) and then verify that the boot ROM can still download and execute a valid image after a failed attempt. Attach a separate sheet with details on the boot methods tested and the results.		
2. Based on the target.nr information, build and boot different configurations of the BSP that include support for features not included in the delivered BSP configuration. Attach a separate sheet with details on the different configurations tested, what features were covered, what features were not covered, and the results of testing. Test each boot method as in step 1.		

BSP Validation Checklist

BSP Validation Test Suite

Item	Date	Status
1. Verify that all applicable tests are successful. Attach a printout of the VTS checklist.		
2. If a test does not run successfully for this BSP, verify that the fact is noted in **target.nr**.		

Optional Product Test

Item	Date	Status
1. If the BSP has support for USB, test USB support for this BSP. To do this, follow the steps below, using the project facility to configure USB. For further information on USB, please see the USB documentation. **Project Facility Configuration:** - Select the USB host stack component. - Select at least one host controller, OHCI or UHCI depending on the BSP support. - Include either mouse or keyboard peripheral components if the hardware is available. **NOTE**: Do not include any initialization components. If you intend to use the target shell for testing, make sure a symbol table is available on the target. **USB Testing:** - Boot the target with USB support. - From either the target or the host shell, invoke the USB tool by calling the **usbTool( )** command. This produces the **usb** tool prompt. Run the following commands from the **usb** tool prompt: **usb> usbi** (initialize stack) **usb> attach ohci** (or **uhci** depending on your configuration) This ensures that USB is configured properly on the target. If a keyboard or mouse USB peripheral is available, test them as follows: **Mouse Testing:** - Attach the mouse device at one of the ports. - Ensure that the device has been detected by enumerating the bus. Call the **usbenum** command from the **usb** tool. - Run the mouse test by calling the **mousete** command from the **usb** tool. The coordinates of the mouse should be displayed as the mouse is moved. **Keyboard Testing:** - Attach the keyboard device and initialize the keyboard by invoking **kbdi** from the **usb** tool. - Run the keyboard test by calling the **kbdpo** command from the **usb** tool. Keys pressed on the keyboard are displayed at the shell.		

BSP Validation Checklist		
2. If the BSP has support for TrueFFS, test TrueFFS support for this BSP. To do this, follow the steps below. For further information on TrueFFS, please see the TrueFFS documentation. You can use either a command-line build or the project facility to test TrueFFS. **Command-Line Configuration:** - Define **INCLUDE_TFFS** in **config.h**. - Define appropriate media and translation layers by defining the **MTD** and **TL** macros in **sysTffs.c**. - Define **INCLUDE_SHOW_ROUTINES** if you need TrueFFS show routines. **Project Facility Configuration:** - Configure the **INCLUDE_TFFS** component. - Select appropriate media and translation layers with the **MTD** and **TL** components. - Select the **TFFS_SHOW** component if you need TrueFFS show routines. **Testing TrueFFS:** - Boot the target with the TrueFFS configuration. - Run **sysTffsFormat( )** (if one exists in **sysTffs.c**) or **tffsDevFormat( )** from either the target shell or the host shell. - Run **usrTffsConfig( )**. - Copy files from the host to the Flash device you have created, and check that the copy operation was successful.		
3. Configure the WDB agent with serial communication and a target server to use the serial backend. Set up the target appropriately with correct serial connections and test the WDB connection by opening a windsh shell and trying a few shell commands such as **i** and **version**.		

Target Information Worksheet

Item	Date	Status
1. Print and attach the BSP Target Information Worksheet to this checklist. Typically, this worksheet is provided in **target.nr**, although some developers might supply it in a different file. See *BSP Developer's Guide: Target Information Reference Page: target.nr*.		
2. Generate the documentation for the BSP. Verify that your documents are complete and conform to the Wind River documentation standards. Attach a printout of the documentation.		
3. Verify that the attached documentation for this BSP contains marketing, engineering, and support contact points (both e-mail and voice).		
4. Add any comments that help to explain the nature and severity of any deviations noted during the testing.		

BSP Validation: PASS: FAIL:	**Checklist Completed:**
Engineer:	**Date:**
Signature:	