Site Configuration

SETUP GUIDE
Linux Hosts
Single Workstation Installation

May 08
4 Generating Node-Locked Workstation Licenses ......................... 11
  4.1 Generating a Node-Locked License File .................................. 11
5 Installing Wind River Products on a Development Workstation ...... 15
  5.1 Before You Begin ...................................................................... 15
  5.2 Installing Wind River Products ................................................. 16
    5.2.1 Standard Installation ....................................................... 16
    5.2.2 Installing a Service Pack ................................................ 18
  5.3 Non-Typical Installations ......................................................... 19
    5.3.1 Installing Workbench into an Existing Eclipse Environment .... 19
    5.3.2 Incremental Installation .................................................... 19
    5.3.3 Parallel Installation .......................................................... 19
6 Post-Installation Tasks ............................................................... 21
  6.1 Installing Patches ................................................................. 21
  6.2 Permanently Activating a Temporary Installation ..................... 22
    6.2.1 Activating Temporary Installations ..................................... 22
    Distributing License Files ....................................................... 23
    Setting an Environment Variable to Access a License Server .......... 23
    6.2.2 Uninstalling a Development Computer License File ............. 24
  6.3 Downloading and Installing Board Support Packages .................. 24
    6.3.1 When Should You Download a BSP? .................................... 24
    6.3.2 Navigating the BSP Web Site ............................................ 25
    Locating a Specific BSP ......................................................... 25
    6.3.3 Downloading and Installing a BSP ..................................... 26
    The BSP Download Page ....................................................... 26
    Installing a BSP with the Maintenance Tool ............................. 26

Distributing License Files .......................................................... 23
Setting an Environment Variable to Access a License Server ............ 23
6.2.2 Uninstalling a Development Computer License File ............. 24
6.3 Downloading and Installing Board Support Packages .................. 24
  6.3.1 When Should You Download a BSP? .................................... 24
  6.3.2 Navigating the BSP Web Site ............................................ 25
  Locating a Specific BSP ......................................................... 25
  6.3.3 Downloading and Installing a BSP ..................................... 26
  The BSP Download Page ....................................................... 26
  Installing a BSP with the Maintenance Tool ............................. 26
6.4 Building the VxWorks Run-Time Source .......................................................... 27
6.4.1 The Kernel and Platform Build Processes ............................................. 28
  Building VxWorks Kernel Source Code .................................................... 28
  Building Wind River VxWorks Platforms Source Code ........................ 29
6.4.2 Where to Find Additional Information ................................................. 29

6.5 Building the Linux Target Run-Time Source .................................................... 30
6.5.1 When Is It Necessary to Build Linux Source Code? ............................ 30
6.5.2 Choosing a Build Method .................................................................... 30
6.5.3 Where to Find Additional Information ................................................. 31

6.6 Uninstalling Wind River Products .............................................................. 31
  Archiving Downloaded Plug-Ins ............................................................. 31
  Preserving Workbench Project Files ....................................................... 31
6.6.1 Uninstalling Through the Maintenance Tool ..................................... 32
6.6.2 Uninstalling a Service Pack ............................................................. 32

7 Wind River Environment Utility (wrenv) ..................................................... 33
7.1 What Is wrenv? ............................................................................................ 33
7.2 When Are You Required to Invoke the wrenv Utility? .............................. 34
  7.2.1 Invoking the wrenv Utility ................................................................. 34
7.3 wrenv Command Options ........................................................................ 35
7.4 wrenv Usage Examples ............................................................................ 35
  Creating a Development Shell ................................................................. 35
  Setting the Development Environment Without Spawning a Shell .... 35
7.5 How Does wrenv Create a Unified Environment Setting? ..................... 36
7.6 What Is the Installation Properties File? ................................................... 37
  7.6.1 Package Descriptor ........................................................................... 39
7.7 Advanced wrenv Topics .......................................................................... 39
7.7.1 Using Packages from Multiple Installation Locations ..................... 39
7.7.2 Special Packages ............................................................................. 41
   Initializer Packages ............................................................................ 41
   Extension Packages ............................................................................ 41

8 Configuring a Remote Build Server .................................................... 43
8.1 Do You Need to Set Up a Remote Build Server? ............................... 43
8.2 Performing Remote Builds with Wind River Workbench ..................... 43
8.3 General Requirements for Remote Builds ......................................... 44
1 Overview

1.1 Introduction

This document guides you through the process of installing and configuring Wind River products on a single Linux developer workstation.

You, or someone in your organization, may need to complete the following site configuration tasks:

▪ Installing the product.
▪ Permanently activating your product installation.
▪ Obtaining node-locked licenses.
▪ Compiling run-time source code.

What This Guide Does Not Cover

The workflow outlined in this document does not describe:

▪ batch (automated) workstation installations
▪ shared (file server) installations
▪ installations for Windows and Solaris workstations

You can find documentation for these topics, as well as the most current version of this document, at

www.windriver.com/licensing/
2.1 Supported Host Types for Wind River Products

Wind River products support development on either Windows, Solaris, or Linux hosts (in many cases, all three host types are supported). This chapter outlines a recommended host system configuration for your chosen host type (both hardware and software) that is applicable to most Wind River products. Product-specific information on supported host types, as well as minimum and recommended host system hardware and software requirements, are provided in your product release notes.

NOTE: The guidelines in this chapter are generalized for all Wind River products. Your host system may require more resources or less, depending on what product you have purchased and your performance requirements. For more specific information, see your product release notes.
2.2 Host System Hardware Requirements

This section provides a set of Wind River-recommended host system hardware requirements for development using a typical Wind River product. The recommended configuration is expected to provide good performance for development on a typical user system. Primarily, this section addresses development using Wind River Eclipse-based products. It also briefly addresses command-line user requirements and optional hardware recommendations. For more information on specific host system requirements for your product, see your product release notes.

For information on host software requirements, see 2.3 Host System Software Requirements, p.5.

2.2.1 Requirements for Eclipse-Based Products

This section lists the recommended host hardware configuration for development using a Wind River Eclipse-based product (such as Workbench).

NOTE: The following requirements do not represent a minimum configuration. Depending on your development requirements and any other third-party software that you are running, you may require more or less than this configuration.

Linux Host Hardware Requirements

Wind River recommends the following hardware configuration for typical development with an Eclipse-based product on a Linux host:

- Intel Pentium 4 class processor, 1.7 GHz or higher recommended.
- 1 GB of RAM.
- Disk space:
  - The amount of disk space required for a typical installation varies from product to product. Depending on your product, the size of your own applications, and your development environment, your host may require several gigabytes of disk space.
  - A DVD-ROM drive or networked DVD-ROM for installation.
2.2.2 Command-Line Users

If you are a command-line user, you may achieve acceptable performance from a host system with fewer resources than the recommended configuration provided in 2.2.1 Requirements for Eclipse-Based Products, p. 4. However, note that even a small amount of development in the Eclipse-based environment requires that you meet the minimum configuration for your product. For product-specific information on minimum host system requirements, see your product release notes.

2.2.3 Optional Hardware

The following hardware is optional but may be useful to you during your development:

- A network interface card for debugging over a network is recommended.

2.3 Host System Software Requirements

This section provides a set of Wind River-recommended host system software requirements for development using a typical Wind River product. The recommended configuration is expected to provide good performance for development on a typical user system. Primarily, this section addresses development using Wind River Eclipse-based products. For more information on specific host system requirements for your product, see your product release notes.

For information on host hardware requirements, see 2.2 Host System Hardware Requirements, p. 4.

2.3.1 Requirements for Eclipse-Based Products

This section lists the recommended host software configuration for development using a Wind River Eclipse-based product (such as Workbench).
NOTE: Software requirements for Wind River products vary considerably from one product to another. Be sure to check your product release notes for additional information before beginning development.

Linux Host Software Requirements

Wind River recommends the following OS version and software for typical development with an Eclipse-based product on a Linux host:

- One of the following host operating systems:
  - Red Hat Enterprise Linux Workstation 4.0 (Update 5)
  - Red Hat Enterprise Linux 5 Desktop with Workstation option (32- or 64-bit)
  - SUSE Linux/openSUSE 10.2
  - Novell SUSE Linux Enterprise Desktop 10 (Service Pack 1)
  - Fedora 7

- Additional required packages for the development host, as listed in the installDir/wrlinux-2.0/wrlinux/required-host.txt files.

- The X Window System, with the GNOME, KDE, or other desktop of your choice.

- A current version of a standards-compliant browser.

- TCP/IP must be installed on the host system, even if it is being used as a standalone computer with a serial connection to the target.

2.3.2 Optional Software and Resources

The following software and resources are optional but may be useful to you during your development:

- An active Internet connection is recommended during initial installation to access patches, documentation, and other important information from the Wind River Online Support Web site.
3 Licensing and Installation Workflow

3.1 How Licensing Works

Many Wind River products are license-managed. This means that your organization buys a license that allows your developers to run a certain number of copies (also known as seats) of a Wind River product simultaneously.

To run a license-managed product, each development host must have its own license file, generated from the Wind River licensing Web site. Licensing tasks can be completed either before or after installation tasks.

3.2 Overview of Licensing Tasks

You or someone in your organization must complete the following tasks before you can run license-managed Wind River products.
For detailed information on the steps below, see 4. Generating Node-Locked Workstation Licenses.

Step 1: Log in to the Wind River licensing Web site.
Before you can activate your Wind River product and your developers can use it, you must log in to and create an account on the Wind River licensing Web site: http://www.windriver.com/licensing:

Step 2: Add the development computer to your account.
You must add information about each development host computer in your organization that will run license-managed products.

Step 3: Generate Product Activation files.
You must generate a Product Activation file for each development computer.

3.3 Overview of Installation Tasks

You can install Wind River products before or after generating a Product Activation file.

Step 1: Install Wind River products on a workstation.
Your developers can wait for you to give them a Product Activation file before installing, or they can install using installation keys from the Developer Essentials sheet and receive a temporary license.

For more information on this step, see 5. Installing Wind River Products on a Development Workstation.

Step 2: Activate workstation licenses.
If your developers installed their license-managed products using a temporary license (also known as temporary activation), you or they must permanently activate those products.

For more information, see 6.2 Permanently Activating a Temporary Installation, p.22.
Step 3: Uninstall Wind River products from a workstation.
If you no longer need to use a particular Wind River product, you can uninstall it from your development computer using the Wind River Maintenance Tool.
For more information, see 6.6 Uninstalling Wind River Products, p.31.

Step 4: Rehost product seats on a different development computer (optional).
After you uninstall a product from a workstation, you can reassign its license to another development computer.
4.1 Generating a Node-Locked License File

Complete the following steps to generate a node-locked license file for a development computer.

1. Log in to the Wind River licensing Web site and create your account.
2. Activate your license.
3. Add your development computer as a new host.
4. Activate your products and generate a license file for the development computer.

These tasks are described in greater detail in the following sections.

Step 1: Log in to the Wind River licensing Web site and create your account.

1. Locate your License Administrator Essentials sheet, as in Figure 4-1. You will need information from it to complete this task.

3. Check the list of products that can be activated from this site to be sure your product appears here. If it appears in the list, click **Login**.

   **NOTE:** Some older products cannot be activated from the licensing Web site; to activate those products, see the installation information that was included with the products. If you do not have access to installation instructions for an older product, contact Wind River Customer Support or e-mail to license@windriver.com.

4. Log in to the site:
4 Generating Node-Locked Workstation Licenses
4.1 Generating a Node-Locked License File

- If you already have a Wind River User ID and Password (for example because you previously logged in to this site or the Wind River support site), type them and click Login.

  or

- If you are not yet a registered user on any Wind River site, click Register for a User ID and Password. Type your user profile information, including your license number from the License Administrator Essentials or the Developer Essentials sheet. Type a password, confirm it, then click Submit.

Step 2: Activate your license.

1. Click Activate your products to open the Add Licenses screen. Type in your License Number and License Administrator Token from the License Administrator Essentials sheet.

   If you expect to eventually manage several product licenses, also type in a label (for example, VxWorks GPP 3.4 or Alameda Campus) to make it easier to identify this particular license in the future.

2. Click Submit. On the Add Licenses Confirmation screen, check to be sure all the information you typed in is correct.

   Your license has now been added to your account. If you order additional Wind River products in the future, click Manage Licenses and then click Add New License to add the new products to your account.

Step 3: Add your development computer as a new host.

Once your license has been activated, use the Manage Hosts screen to add information about the development computer. If you have any hosts already defined, they appear on this screen.

1. To add a host, click Add New Host.

2. On the Create Host screen, enter the host computer’s name, host ID, and other requested information.

   NOTE: For descriptions of the fields, including instructions for finding your host ID, click More information.

3. When you have filled in all required fields, click Create.

   Your new host appears on the Create Host Confirmation screen.
Step 4: Activate your products and generate a license file for the development computer.

1. After you have added the development computer to your account, click Manage Licenses to activate the products you have purchased.

2. From the drop-down list next to the license on the Manage Licenses screen, select Activate Products.

3. Select the product you want to allocate to this development computer (look for NL in the License Type column). Click Next.

4. From the Host Label drop-down list, select this development computer, then click Next.

5. Confirm that you have selected the correct host, then click Next.

6. Click Download to download the license file immediately, or enter an e-mail address if you want the license file to be sent to the person responsible for this host computer. If you entered an e-mail address, click Send.

   **NOTE:** You can both download the license file and receive it in e-mail.

   If you download the file, you must be sure to preserve the .lic ending on the license file or the file will not work properly.

   If you chose to send licenses by e-mail, the E-mail Licenses Confirmation screen displays to whom they were sent.

Once you have the node-locked license file (WRSLicense.lic), see 5. Installing Wind River Products on a Development Workstation for instructions on how to use it.
5.1 Before You Begin

Before you can install your Wind River products, make sure you have the following items available:

- Your product box, which includes:
  - installation media (discs or image files).
  - the Developer Essentials sheet
  - the Getting Started book
- Several gigabytes (GB) of disk space on your development computer for Standard (recommended) installations.

The amount of disk space required varies for each product. For a given installation disk, the installer calculates the amount of space required for all features you selected; this is less than the actual space required if your installation contains multiple disks. Verify that you have adequate free space before you begin installation.
Site Configuration Setup Guide
Linux Hosts, Single Workstation Installation

- Write permissions to your installation directory.
- For typical installations, either:
  - A Product Activation file to permanently activate your products during installation.
  or
  - A license authorization code and Internet access to temporarily activate your products.
- For non-typical installations, installation keys (printed on your Developer Essentials sheet).

5.2 Installing Wind River Products

You can install a Wind River product from the physical media, or from DVD images that have been copied onto a server.

5.2.1 Standard Installation

To install a Wind River product on a development workstation, complete the following steps.

1. Launch the installer by putting the disc labelled Install First into your drive, or by navigating to the product image, and running setup_linux.

   NOTE: If you are installing the product from a shared network directory, ensure that the directory name and path do not include spaces. If there is a space character anywhere in the absolute path, some elements of the installation will not be available.
2. Read each installer screen and provide the requested information, then click Next.

3. From the Choose Activation Type screen, select the type of installation you want:

   **Temporary activation**
   Select this method if you wish to install and use the products for a limited time. However, to continue using them, you must later permanently activate this installation (using a license file).

   When you select this option, the installer program automatically locates your License Authorization Code (LAC). To use this method, you must have Internet access.

   **Permanent activation**
   Select this option if you have a Product Activation file. You do not need Internet access to install and activate Wind River products using this option.

   When you select this option, you can browse to the directory location of your Product Activation file.

   **Advanced**
   When you click the Advanced button, a third option, Manual activation (not typical), appears. Under this method, you install products by entering installation keys without activating your products. This is only performed when you do not have a Product Activation file, do not have an Internet connection, and prefer to set up your licensing manually.

4. Select an installation type, then enter the required information.

4. On the Choose Installation Filters screen, select Standard Installation or Custom Installation.
NOTE: Wind River recommends a standard installation, but if disk space is a problem, a customized installation allows you to install just those products and tools that you need.

You can also click the **Advanced** button on this screen to select host types. For single-workstation installations, your host type is pre-selected.

5. If you selected **Standard Installation**, the next screen shows you the products and features that will be installed; click **Install** to start the installation process. If you selected **Custom Installation**, click **Next**. Select the products and tools you want to install, then click **Install**.

**CAUTION:** Use caution when unselecting features, as unselecting critical features could prevent the products from functioning properly.

If you installed using a Product Activation file, your Wind River products are now installed, activated, and ready for you to begin working with them.

If you installed using a license authorization code or installation keys, you may use your products now but you must permanently activate them to continue working with them. For information on how to do this, see **6.2 Permanently Activating a Temporary Installation**, p.22.

If you were not able to install, see your license administrator.

### 5.2.2 Installing a Service Pack

Each Wind River product service pack is delivered with its own installer program, similar to the one you used to install your products in **5.2.1 Standard Installation**, p.16. The installer program automatically uses the installation keys that are included with the service pack.

Service packs are installed as additions to a previous installation; they do not overwrite existing product installations.
5.3 Non-Typical Installations

You may not need the instructions in this section, depending on your organization’s setup.

5.3.1 Installing Workbench into an Existing Eclipse Environment

Wind River Workbench provides a complete Eclipse framework. You do not need to install Eclipse separately. However, if you have your own customized Eclipse installation, and you prefer to integrate Workbench into it, you must:

1. Install Wind River Workbench (as described in this document).
2. Follow the instructions in Wind River Workbench User’s Guide: Using Workbench in an Existing Eclipse Environment to integrate Wind River Workbench into your existing Eclipse installation.

5.3.2 Incremental Installation

An incremental product installation can be performed in situations such as the following:

- You installed only certain features of a Wind River product (by selecting Custom rather than Standard installation) and now wish to add the excluded features.
- You are installing complementary Wind River products (for example, adding On-Chip Debugging to your existing installation of Wind River Workbench).

When you do an incremental installation, you can again choose Standard or Custom mode. In a standard installation, the installer program installs only those products or features that are not yet present in your installation directory. If you select Custom installation, the installer program shows you a full list of the available products and features, indicating which ones are already installed. You can choose whether to re-install what is already present. (By default, existing features are not re-installed.)

5.3.3 Parallel Installation

If you have a previous version of Workbench installed, you can install a later version alongside it as long as the two are in different directories.
6 Post-Installation Tasks

6.1 Installing Patches 21
6.2 Permanently Activating a Temporary Installation 22
6.3 Downloading and Installing Board Support Packages 24
6.4 Building the VxWorks Run-Time Source 27
6.5 Building the Linux Target Run-Time Source 30
6.6 Uninstalling Wind River Products 31

6.1 Installing Patches

To install patches for your Wind River products, use the Wind River Product Maintenance Tool, provided with products from Wind River Workbench 3.0 forward and available from your installation directory. The Maintenance Tool handles all updates to Wind River software after initial installation, as well as product uninstallation.

To launch the Maintenance Tool, do the following:

- From the GUI
  Select Applications (the main menu on the panel) > Wind River > Uninstall and Maintenance > Maintenance Tool.
### 6.2 Permanently Activating a Temporary Installation

If you used a Product Activation file (install.txt) when installing Wind River products, those products are permanently activated and require no further action.

If you used a license authorization code (LAC) or installation key to temporarily activate your products, you can use those products for a limited time but you must obtain a license file to permanently activate your products.

**NOTE:** The length of a temporary activation varies by product, but is typically no less than 30 days. Wind River license-managed tools display a warning as they approach expiration.

*Permanent activation* means that a product may be used to the fullest extent of its license.

#### 6.2.1 Activating Temporary Installations

There are two ways to permanently activate Wind River products that were issued temporary licenses:

- You can create and distribute license files for each development workstation.
- Or
- On each development workstation, you can set an environment variable to access a license server.

---

**From the command line**

Execute the following commands from the command shell:

```
cd installDir/maintenance/mtool
./mtool_linux
```

**NOTE:** Patches cannot be installed with a silent or unattended installation method.
Distributing License Files

The easiest (and recommended) way to activate temporary licenses is to make a license file available to each member of your team.

1. Follow the instructions in the Generating Workstation Licenses chapter to create the appropriate license file(s).

2. Name the new file WRSLicense.lic.

3. Copy the file into the installDir/license directory of each development system.
   or

   3. Place the license file on a server and make it available for your development team to download onto their computers. Be sure to instruct them to copy the file to their installDir/license directory.

   **NOTE:** Each node-locked development computer requires a separate license file, generated using that computer’s host ID.

Setting an Environment Variable to Access a License Server

The second way to permanently activate temporary licenses is to set an environment variable to point to a license server.

   **NOTE:** In order to use this method, you (or someone in your organization) must first generate a server license file and install a license server.

1. Create, or instruct each member of your team to create, an environment variable on the development computer called WRSD_LICENSE_FILE.

2. Set the value of this variable to the port@servername indicated in the SERVER line of the Product Activation file you downloaded from the licensing Web site.

   For example, the SERVER line may look like the following:

   ```
   SERVER jupiter hostID 27000
   ```

   In this example, the server name is jupiter and the port number is 27000, so you would set the variable to 27000@jupiter.

   Once the variable is set, this permanently activates the temporary licenses.
6.2.2 Uninstalling a Development Computer License File

If you are uninstalling a Wind River product from a development computer, the uninstallation process described in 6.6 Uninstalling Wind River Products, p.31, does not remove the license file.

You must manually remove it by navigating to the `installDir/license` directory and deleting the file, usually named `WRSLicense.lic`.

NOTE: If the product you are uninstalling was listed in a merged license file that resides in the product’s installation tree, removing the license file affects all products whose licenses are also under the file’s control.

Wind River recommends that you back up the file, delete the entry for the product(s) you are uninstalling, then save the file. The remaining products on the development computer (and listed in the license file) should work as before.

6.3 Downloading and Installing Board Support Packages

Wind River provides board support packages (BSPs) for both the VxWorks and Wind River Linux operating systems. Although the specific details of a BSP vary considerably between VxWorks and Linux, BSPs serve the same purpose on each operating system. That is, BSPs provide the necessary code to run your customized VxWorks or Wind River Linux operating system instance on specific target hardware. BSPs provided by Wind River can be used for development on available hardware or can be used as a base for BSP development for your custom hardware.

NOTE: Wind River recommends that, when developing for custom hardware, you try to customize an existing BSP instead of trying to write your BSP from scratch.

For more information on BSP development, see the VxWorks BSP Developer’s Guide or the Wind River Linux Platforms Users Guide.

6.3.1 When Should You Download a BSP?

Many VxWorks and Wind River Linux distributions provide board support packages directly on the product installation media. However, new BSPs and
updates are continuously added to the BSP Web page. Therefore, to have access to the latest board support packages and updates, you must visit the BSP Web page.

If the BSP you wish to use for development is included on your installation media, check the BSP Web site to be sure that it is the latest version. If the BSPs included on your installation media are not suitable for your development, see the BSP Web site for a full list of available BSPs.

**Compatibility**

The Wind River installation program does not verify version compatibility between your Wind River products and BSPs. For compatibility information, see the documentation that accompanies the BSP.

### 6.3.2 Navigating the BSP Web Site

The Wind River public Web site provides you with access to a listing of all available Wind River BSPs. The BSP main page is:

http://www.windriver.com/products/bsp_web/

You can also navigate to this page from the Wind River Web site home page by selecting **Products > Board Support Packages**.

**NOTE:** This Web page is publicly accessible, but you must have a valid Online Support login in order to download a BSP .zip file.

### Locating a Specific BSP

From the BSP main page, you can choose to locate a specific BSP by:

**Architecture**

This is the generic processor family to which the desired CPU belongs. For example, ARM or PowerPC. If you know the architecture family for your development processor, use this category.

**Hardware Vendor**

This is a list of hardware vendors for the development boards that are supported by the Wind River BSPs. If you know the manufacturer of the development board for the BSP you wish to use, use this category.
NOTE: This category is organized by hardware development board vendors, not by processor manufacturer. In many cases, the CPU manufacturer is not the board vendor.

Market
This is the general market targeted by the development board or processor. For example, processors that are targeted for use in medical equipment are grouped into one category while processors targeted for the automotive market are grouped into that category.

Platform
This is a list of available Wind River products that include BSP support. You can use this category to see a list of all BSPs available for your specific Wind River product and version.

6.3.3 Downloading and Installing a BSP

This section provides detailed instructions for downloading and installing BSPs for products from Wind River Workbench 3.0 forward.

The BSP Download Page

Once you have located your desired BSP, you must download and install the associated .zip file. To get to the download page, find the correct entry in the BSP list and click More (this button appears at the far right end of the specific BSP entry line). This link brings you to the BSP technical details page.

If a version of the BSP is available for download, a link to a downloadable .zip file appears at the top of the page under the BSP Sales Contact field. Certain BSPs are only available as part of a product distribution. If this is the case, the Product Availability field indicates that the BSP is available on CD-ROM (or DVD-ROM). If you do not have the product CD (or DVD) for the BSP, contact Wind River Customer Support for assistance.

Installing a BSP with the Maintenance Tool

Downloaded BSPs are placed in your installDir/updates directory. To install a new BSP, launch the product Maintenance Tool:
6 Post-Installation Tasks
6.4 Building the VxWorks Run-Time Source

- From the GUI
  Select Applications (the main menu on the panel) > Wind River > Uninstall and Maintenance > Maintenance Tool.

- From the command line
  Execute the following commands from the command shell:

```
% cd installDir/maintenance/mtool
%/mtool_linux
```

6.4 Building the VxWorks Run-Time Source

The information in this section applies to one or more of the following product bundles:

- Wind River General Purpose Platform, VxWorks Edition
- Wind River Workbench for VxWorks
- Wind River VxWorks Platforms

**NOTE:** Wind River VxWorks Platforms includes:
- Wind River Platform for Automotive Devices, VxWorks Edition (Platform AD)
- Wind River Platform for Consumer Devices, VxWorks Edition (Platform CD)
- Wind River Platform for Industrial Devices, VxWorks Edition (Platform ID)

Any of the above products can be optionally purchased as a source code product. However, Wind River VxWorks Platforms products (Platform AD, Platform CD, Platform ID, and Platform NE) always include source code for technologies such as the Wind River Network Stack. The General Purpose Platform and the Wind River Workbench for VxWorks products are pre-compiled and do not require a source code build (even if you have purchased the source code option). The Platform AD, Platform CD, Platform ID, and Platform NE source code must be compiled prior to using the Platform product.

**NOTE:** Platform AD, Platform CD, Platform ID, and Platform NE include a pre-built kernel; but all other Platform technologies, including the network stack, must be built from source prior to first use.
6.4.1 The Kernel and Platform Build Processes

Depending on what product and options you have purchased as well as your development requirements, you may wish to perform a VxWorks kernel build or a Platform technology build, or both. (Note that for the General Purpose Platform and the Wind River Workbench for VxWorks products, neither build is required).

Additional information on each build process (and when it must be performed) is provided in the remaining sections. Complete build instructions and additional information on each build type is available from one of the following manuals (depending on which product you have purchased):

- Wind River General Purpose Platform, VxWorks Edition Getting Started: Compiling Source
- Wind River Workbench for VxWorks Getting Started: Compiling Source
- Wind River VxWorks Platforms Getting Started: Compiling Platform Component Source

Building VxWorks Kernel Source Code

In general, you are not required to compile the VxWorks operating system (kernel) source code in order to begin development with any VxWorks-based product, even if you purchase a source code product. The General Purpose Platform is shipped fully pre-compiled and users are not required to compile any source code before using this product. Wind River VxWorks Platforms users must compile their product before use, but that requirement is limited to the network stack and other technologies that sit above the underlying operating system. The kernel itself is pre-compiled and does not need to be built prior to use. (For more information on building Wind River VxWorks Platforms products, see Building Wind River VxWorks Platforms Source Code, p.29.)

Although it is not required, there are many reasons for which you might wish to build or, at a minimum, reference the VxWorks source code. One of the most common reasons is to ease the application debugging process (the pre-built VxWorks kernel images are not built with debug symbols).
Building Wind River VxWorks Platforms Source Code

Wind River VxWorks Platforms products (Platform AD, Platform CD, Platform ID, and Platform NE) are always shipped as source code and must be compiled prior to using the product.

**NOTE:** In all Wind River VxWorks Platforms products, the VxWorks kernel and system libraries are shipped in a pre-built form as well as source form. You can use these pre-built binaries to develop a custom VxWorks image. However, the pre-built code does not include networking facilities or any other Platform technologies. Therefore, you will not have access to any of those features until you have built the Platform source.

The following section provides a brief overview of the build process.

### Wind River VxWorks Platforms Build Process Overview

Building the Wind River VxWorks Platforms source is a two step process.

1. First, you can customize the features in your Platform using the Platform Makefile. This makefile allows you to specify what features you want to have available in your Platform during development. You can also choose to skip this customization and compile your Platform without modification using the default makefile options.

   For complete information on customizing your Platform using the Platform Makefile, see *Wind River VxWorks Platforms Getting Started: Customizing the Default Configuration*.

2. Next, you must execute the source code compile using Wind River Workbench or the vxprj command-line utility. (For more information on the vxprj command-line utility, see the *VxWorks Command-Line Tools User’s Guide.*) This step also requires you to choose the compiler and target architecture you wish to compile for.

### 6.4.2 Where to Find Additional Information

Complete build instructions and additional information on each build type is available from one of the following manuals (depending on which product you have purchased):

- *Wind River General Purpose Platform, VxWorks Edition Getting Started*
- *Wind River Workbench for VxWorks Getting Started*
6.5 Building the Linux Target Run-Time Source

6.5.1 When Is It Necessary to Build Linux Source Code?

All Wind River Linux Platforms require a source code build prior to using the product for development. You must build the source using one of the two build methods that are available for building the product. The available methods are the Red Hat package manager (RPM) build method and the source build method.

Additional information on each build type is provided in the following section. Complete instructions for building your source code with each build type are provided in the Wind River Linux Platforms Users Guide.

6.5.2 Choosing a Build Method

As stated previously, you must build your Platforms source before using the product, and there are two methods available for building your Platform source code. The first method is the RPM build method. This method is the fastest way to build a complete run-time system. The RPM build method uses a pre-built, target hardware and file system-specific kernel as well other pre-built elements to assemble a complete run-time system. Because this method uses largely pre-built source code to assemble the target run-time, the build completes in just minutes.

The second build option is a full source code build. This method allows you to build the entire run-time system, including the kernel, from source. However, when using this method, it can take several hours for a complete system to build the first time it is run. If necessary, you can shorten the build time while still maintaining control over what source code is built, by using the source build method for your first build and then using the RPM build method for subsequent builds. You can choose the source code build method to build only certain aspects of the system (such as a kernel-only build) and then assemble the final run-time system using the RPM method.
6.5.3 Where to Find Additional Information

Complete information on the available build types and step-by-step tutorials that walk you through the Wind River Linux Platforms target run-time build process are provided in the *Wind River Linux Platforms User’s Guide*.

6.6 Uninstalling Wind River Products

This section describes the process for removing Wind River products from a system. If you are uninstalling a Wind River product in preparation for reinstalling a newer version, or you are moving the product to a new development computer and you want to continue working with it, you may first wish to archive some files before uninstalling.

**Archiving Downloaded Plug-Ins**

If you installed any third-party plug-ins in the installation directory of your Wind River product and you want to preserve them, you must move them outside the installation directory. For more information about working with plug-ins, see *Wind River Workbench User's Guide: Integrating Plug-Ins*.

**Preserving Workbench Project Files**

To preserve the state of Workbench projects and be able to recreate them after reinstalling the product, copy the following files to a location outside the Workbench installation directory:

<table>
<thead>
<tr>
<th>Project File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*.project</td>
<td>Eclipse platform project file containing general information about the project.</td>
</tr>
<tr>
<td>*.wrproject</td>
<td>Workbench project file containing mostly general build properties.</td>
</tr>
</tbody>
</table>
For user-defined projects, all makefiles must also be version-controlled.

6.6.1 Uninstalling Through the Maintenance Tool

To remove Wind River products, it is recommended that you use the Maintenance Tool provided with your product installation. The Maintenance Tool can uninstall many products at the same time.

**NOTE:** The Maintenance Tool removes all Wind River products from Wind River Workbench 3.0 forward; it does not uninstall products from previous releases.

To launch the uninstallation tool, do the following:

- **From the GUI**
  
  Select *Applications* (the main menu on the panel) > *Wind River* > *Uninstall and Maintenance* > *Maintenance Tool*.

- **From the command line**
  
  Execute the following commands from the command shell:

  ```
  % cd installDir/maintenance/mtool
  % ./mtool_linux
  ```

6.6.2 Uninstalling a Service Pack

Service packs are uninstalled the same way a standard release is uninstalled. Follow the instructions in 6.6.1 *Uninstalling Through the Maintenance Tool*, p.32, to uninstall the product(s) you no longer want to use.
7
Wind River
Environment Utility
(wrenv)

7.1 What Is wrenv?

The Wind River environment utility, wrenv, is primarily used to create a command shell with a pre-loaded environment. The utility guarantees a consistent, portable execution environment for tools (such as make) that are launched from Wind River Workbench or from a command-line automation environment such as the VxWorks development shell. The wrenv utility also provides services that other tools can use to query environment settings that are set by wrenv. The wrenv utility replaces the need for the multiple host and shell-specific scripts (Vars scripts) that were used to set environment variables and paths in earlier Wind River products.
7.2 When Are You Required to Invoke the wrenv Utility?

In many cases, the Wind River environment utility (wrenv) is invoked automatically by the Wind River tools when your product is launched. Therefore, when you are using a standard tools product (such as Wind River Workbench) to do standard development, you are not required to launch wrenv manually. However, there are certain cases where this is required. For example, when using the Workbench tools for VxWorks from the command line (using the vxprj command-line facility), you must begin by invoking wrenv.

For more information on using the wrenv utility, see 7.2.1 Invoking the wrenv Utility, p.34, and 7.5 How Does wrenv Create a Unified Environment Setting?, p.36, as well as the VxWorks Command-Line Tools User’s Guide: Creating a Development Shell with wrenv.

7.2.1 Invoking the wrenv Utility

To invoke the wrenv utility, open a host shell and type the following:

```
% installDir/wrenv.sh -p package
```

where package is the name of the product package you wish to use.

For example, to invoke the proper environment for VxWorks 6.4 development, type the following:

```
% installDir/wrenv.sh -p vxworks-6.4
```

**NOTE:** If your shell configuration file overwrites the environment each time a new shell is created, the above command may not work. If you have difficulty invoking the Workbench tools after executing the above command, invoke wrenv as follows:

```
% eval `installDir/wrenv.sh -p vxworks-6.4 -o print_env -f shell`
```

where shell is sh or csh, depending on your current shell program.
7.3 wrenv Command Options

The general syntax for a wrenv command is as follows:

```
% wrenv options env=value command [args]
```

The -p option (-p package) selects the package to use for the environment initialization and is used for all invocations. The only exception is when an initializer package is defined in install.properties. (For more information on initializer packages, see Initializer Packages, p.41.)

In addition to the -p option, wrenv accepts a number of additional command-line options that allow you to further customize your development environment. For information on these additional wrenv command-line options, see the VxWorks Command-Line Tools User’s Guide: Creating a Development Shell with wrenv.

7.4 wrenv Usage Examples

The following are some common use case scenarios for the wrenv utility. An advanced use case regarding installing packages from multiple installation locations is also provided later in this text (see 7.7.1 Using Packages from Multiple Installation Locations, p.39).

Creating a Development Shell

As mentioned previously, the most common usage of wrenv is to set up a specific development environment prior to using your Wind River product. The preferred syntax is as follows:

```
% wrenv.sh –p package
```

This command spawns a development shell with the environment for the selected package (package).

Setting the Development Environment Without Spawning a Shell

In some cases, it is not desirable to start a development shell—for example, when you are using scripting to set the environment on a host, or when you cannot start
a development shell because your .cshrc or .profile file overrides PATH or other variables whenever a shell is started.

In these cases, the following command can be used to output the environment variables to standard out:

```
% wrenv.sh -p package -o print_env -f format
```

In this example, `format` specifies the output format or shell type, which can be set to plain, sh, csh, bat, or tcl.

For example, to set the environment in csh without starting a sub-shell, use the following command:

```
% eval 'wrenv.sh -p package -o print_env -f csh'
```

### 7.5 How Does wrenv Create a Unified Environment Setting?

The `wrenv` utility takes advantage of an installation properties file (`install.properties`). The installation properties file, which can be considered a type of installation registry, consolidates various pieces of information in a single location, including the information needed for environment settings. The installation properties file is independent of host platform and provides information to help enforce correct environment setting dependencies across multiple installed component packages (such as Wind River Workbench, Wind River VxWorks Platforms, documentation, and test packages). At product installation time, the installation properties file is created by aggregating the package property files (`package.properties`) that accompany each installed package. The installer invokes a post-installation script (`installDir/setup/postinstall.sh`) to aggregate the necessary `package.properties` files. The `wrenv` utility is the primary processing engine for information stored in the installation properties file. Typically, `wrenv` processes any environment setting information related directly or indirectly to the selected package (-p option) in the installation properties file and then creates a new command shell containing the specified environment.
7.6 What Is the Installation Properties File?

**NOTE:** Do not edit the install.properties file. The information provided in this section is for reference purposes only.

The following is a typical example of the contents of an installation properties file. Comment lines in a properties file begin with a # symbol as the first non-white-space character.

```properties
#Sample install.properties file
vxworks66.name=vxworks-6.6
vxworks66.version=6.6
vxworks66.type=platform
vxworks66.subtype=vxworks
vxworks66.label=Wind River VxWorks 6.6
vxworks66.bdp_type=preference
vxworks66.bdp_creationcmd=cat
$WIND_BASE/host/resource/bdgen/%projecttype%.cache
vxworks66.eval.01=export WIND_HOME=$(builtin:InstallHome)
vxworks66.eval.02=export WIND_BASE=$(WIND_HOME)/vxworks-6.6
vxworks66.eval.03=export WIND_USR=$(WIND_BASE)/target$/usr
vxworks66.eval.04=require [compiler,diab,5.6.0]. [compiler,gnu-vxworks-6.6,4.1.2]
vxworks66.eval.05=require [components,,1.0]
vxworks66.eval.06=addpath PATH $(WIND_BASE)$/vxtest$/src$/scripts
vxworks66.eval.07=addpath PATH $(WIND_BASE)$/host$/$(WIND_HOST_TYPE)$/bin
vxworks66.eval.08=addpath LD_LIBRARY_PATH
$ (WIND_BASE)$/host$/$(WIND_HOST_TYPE)$/lib
vxworks66.eval.09=addpath WIND_SAMPLES $(WIND_BASE)$/target$/src$/demo
vxworks66.eval.10=addpath WIND_SAMPLES $(WIND_USR)$/apps$/samples
vxworks66.eval.11=addpath MANPATH $(WIND_BASE)/man
vxworks66.eval.12=export TCLLIBPATH=$(WIND_BASE)/host/resource/tcl
vxworks66.eval.13=export WIND_PLATFORM=vxworks-6.6
vxworks66.eval.14=optional [components,,1.0]
vxworks66.eval.15=optional wrmscomponents
workbench30.name=workbench-3.0
workbench30.version=3.0
workbench30.type=workbench
workbench30.label=Wind River Workbench 3.0
workbench30.eval.01=export WIND_HOME=$(builtin:InstallHome)
workbench30.eval.02=export WIND_DOCS=$ (WIND_HOME)$/docs
workbench30.eval.03=export WIND_HOST_TYPE=$(builtin:HostType)
workbench30.eval.04=export WIND_TOOLS=$(WIND_HOME)/workbench-3.0
workbench30.eval.05=define WIND_LICENSE=$(WIND_HOME)$/license
workbench30.eval.06=export WIND_DFW_PATH=$(WIND_TOOLS)$/dfw$/0199_1
workbench30.eval.07=define WIND_JRE_PATH=$(WIND_HOME)$/jre$/1.5.0_11
workbench30.eval.08=export WIND_JRE_HOME=$(WIND_JRE_PATH) $/ (WIND_HOST_TYPE)
workbench30.eval.09=export
WIND_WRWB_PATH=$(WIND_TOOLS)$/wrwb$/platform$/eclipse
workbench30.eval.10=export
WIND_FOUNDATION_PATH=$(WIND_TOOLS)$/foundation$/4.1.1
```

37
The `install.properties` file is a hierarchical registry of package components. It aggregates information from the `package.properties` files that accompany each installed package. An entry in a properties file has the following form:

```
rootkey.subkey=value .subkey=value ...
```

The `install.properties` file is dynamically created at installation time by concatenating all `package.properties` files found in the first- and second-level directories under the installation home directory (`installDir`). This is done by a post-installation program without any involvement from the `wrenv` program.

Each entry in the installation properties file can be considered as a hierarchical registry entry. The `root key` of each entry is a unique key determined by the name of an installed package component and is used by `wrenv` to group related entries together into packages. A hierarchical system of `subkeys` exists under each root key, with each subkey having its own unique interpretation. Subkeys define items such as the package name or version numbers and compatibility with other packages as well as more complex environment settings.
7.6.1 Package Descriptor

Package descriptors are used to reference other packages. The format of a package descriptor is as follows:

\[\text{type-name}, \text{subtype-name}, \text{low-version}, \text{high-version}\]

The `type-name` and `subtype-name` values select the package class of the referenced package. The `low-version` and `high-version` values are used to specify a range of version numbers from the installed packages of the specified class that `wrenv` should reference. By default, `wrenv` selects the highest matching version. You can control this by specifying preferred packages using the `–P` command-line option. The `high-version` field is optional and, when not specified, `wrenv` matches on all versions with the same major and minor version number that are greater or equal to the `low-version`.

For example, `[platform,vxworks,6.3,6.6]` refers to any version of VxWorks version 6.3 up to—but not including—6.6. It might seem awkward at first that the `high-version` field is non-inclusive; however, this was done by design because one would otherwise have to use something like 6.5.999.999 to specify the same behavior.

7.7 Advanced wrenv Topics

This section includes information on advanced topics. These features may not be supported for your installation. If you have questions on these topics and their support, contact Wind River Customer Support.

7.7.1 Using Packages from Multiple Installation Locations

In some cases, you may wish to install each package in a separate installation root directory. Two different ways to accomplish this scenario are described in the following sections.

Note that the following examples use three sample packages as follows:

- IDE
  - Installation location: `/inst/ide`
  - Root path variable: `IDE_PATH`
Run-time
- Installation location \texttt{/inst/runtime}
- Root path variable: \texttt{RUNTIME\_PATH}

Compiler
- Installation location: \texttt{/inst/compiler}
- Root path variable: \texttt{COMPILER\_PATH}

Each package contains a \texttt{package.properties} file in the top-level directory.

Using Multiple -i Options

With multiple -i options, \texttt{wrenv} concatenates the contents of each file specified, dynamically creating an \texttt{install.properties} file with all the relevant packages. (For more information on the \texttt{install.properties} file, see \textit{7.6 What Is the Installation Properties File?}, p.37.) The package root environment variable for each package is specified as follows:

\begin{verbatim}
% wrenv.sh -p runtime -i /inst/ide/package.properties
   -i /inst/runtime/package.properties -i /inst/compiler/package.properties
INSTALL\_PATH=/inst/root IDE\_PATH=/inst/ide RUNTIME\_PATH=/inst/runtime
COMPILER\_PATH=/inst/compiler
\end{verbatim}

Using the Initializer Package

The initializer package approach removes the need to specify numerous command-line options. For this example, the initializer package is as follows:

\begin{verbatim}
initializer.name=initializer
initializer.type=initializer
initializer.eval.01=export INSTALL\_PATH=/inst/root
initializer.eval.02=export IDE\_PATH=/inst/ide
initializer.eval.03=export RUNTIME\_PATH=/inst/runtime
initializer.eval.04=export COMPILER\_PATH=/inst/compiler
initializer.eval.05=include $(IDE\_PATH)$/package.properties
initializer.eval.06=include $(RUNTIME\_PATH)$/package.properties
initializer.eval.07=include $(COMPILER\_PATH)$/package.properties
initializer.eval.08=default runtime
\end{verbatim}

Assuming the above is in the file \texttt{myplatform}, the following command is used to set up the environment:

\begin{verbatim}
% wrenv.sh -i ~/myplatform
\end{verbatim}

For more information on using the initializer package, see the next section.
7.7.2 Special Packages

This section describes the special package types that are available for \texttt{wrenv}.

Initializer Packages

An initializer package (if provided, see \texttt{7.7.1 Using Packages from Multiple Installation Locations}, p.39) is evaluated prior to evaluating the package specified by the \texttt{-p} option. This package is intended to include other properties files (see \texttt{7.6 What Is the Installation Properties File?}, p.37), define variables, and set the default package name.

An example initializer package file might include the following lines:

\begin{verbatim}
initializer.name=initializer
initializer.type=initializer
initializer.eval.01=export TEST_HOME=$(builtin:InstallHome)
initializer.eval.02=include $(TEST_HOME)$/install.properties1
initializer.eval.03=include $(TEST_HOME)$/install.properties2
initializer.eval.04=include $(TEST_HOME)$/install.properties3
initializer.eval.05=default vxworks-6.1
\end{verbatim}

Extension Packages

Packages of the type \texttt{extension} are used to dynamically extend other packages. These packages are evaluated automatically when the package they extend is evaluated. This is very useful to add properties to packages that cannot be modified directly. For example, this can happen in the following situations:

- When the package to be extended is not owned by your company or your group.
- For packages that are optionally installed but still must be integrated with some package.

An extension package has the same format as any other package with the following key points to remember:

- The \texttt{type} value must be set to \texttt{extension}.
- Extension packages are matched with the extended package using one or more \texttt{compatibility} properties.

For example, the add-on package to Workbench 2.5 could look as follows:

\begin{verbatim}
testext.name=testext
\end{verbatim}
testext.type=extension
testext.subtype=testext

testext.version=1.0
testext.compatible=[workbench,,2.5]
testext.eval.01=export TESTEXT_BASE=$(WIND_HOME)$/textext
testext.eval.02=addpath PATH $(TESTEXT_BASE)$/bin$/$(WIND_HOST_TYPE)
testext.eval.04=addpath SAMPLES $(TESTEXT_BASE)$/samples
testext.eval.06=addpath WIND_EXTENSIONS $(TESTEXT_BASE)$/extensions
8.1 Do You Need to Set Up a Remote Build Server?

A remote build server is desirable if your developers want to offload build cycles to a remote system rather than doing builds on their development computers.

8.2 Performing Remote Builds with Wind River Workbench

The Workbench remote build feature allows you to develop, build, and run your applications on a local host that is running Workbench, using a workspace that is located on a remote host as if it were on a local disk.
When you launch the build, a network connection (rlogin or SSH) is established to the build host, and the actual build command is executed there, using an intermediate script that allows you to set up the needed environment for the build process.

### 8.3 General Requirements for Remote Builds

- The workspace root directory must be accessible from both computers.
- Only Eclipse projects located underneath the workspace root can be remotely built. In other words, linked resources are not supported for files outside the workspace.
- A rlogin or SSH remote connection to the build host must be possible.

For more information about the Workbench remote build feature, see the Developing on Remote Hosts section of Wind River Workbench User’s Guide: Building: Use Cases.