

WIND RIVER® LINUX

RELEASE NOTES

7.0

RCPL 6



Copyright Notice

Copyright © 2015 Wind River Systems, Inc.

All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means without the prior written permission of Wind River Systems, Inc.

Wind River, Tornado, and VxWorks are registered trademarks of Wind River Systems, Inc. The Wind River logo is a trademark of Wind River Systems, Inc. Any third-party trademarks referenced are the property of their respective owners. For further information regarding Wind River trademarks, please see:

www.windriver.com/company/terms/trademark.html

This product may include software licensed to Wind River by third parties. Relevant notices (if any) are provided in your product installation at one of the following locations:

`installDir/product_name/3rd_party_licensor_notice.pdf`
`installDir/legal-notices/`

Wind River may refer to third-party documentation by listing publications or providing links to third-party Web sites for informational purposes. Wind River accepts no responsibility for the information provided in such third-party documentation.

Corporate Headquarters

Wind River
500 Wind River Way
Alameda, CA 94501-1153
U.S.A.
Toll free (U.S.A.): 800-545-WIND
Telephone: 510-748-4100
Facsimile: 510-749-2010

For additional contact information, see the Wind River Web site:

www.windriver.com

For information on how to contact Customer Support, see:

www.windriver.com/support

22 May 2015

Contents

1 About this Release	5
About Wind River Linux 7.0	5
Latest Release Information	6
Installation and Licensing	6
Minimal Install	7
Performing a Minimal Install	7
Adding Workbench to a Minimal Install	8
Migration and Backward Compatibility	9
Workbench Support	9
Product Updates as Patches	10
2 System Requirements	11
Host System Requirements	11
Development Host Requirements	12
Necessary Development Host Libraries and Executables	13
Target System Requirements	13
Supported Target Boards	14
Targets Simulated by QEMU	14
Targets Simulated by Simics	14
Installation Requirements and Issues	14
Installing Multiple Products Into the Same Location	14
3 Changes in This Release	17
About Changes to Wind River Linux 7.0	17
New or Enhanced Features	17
New Use Cases and Enhanced Procedures	18
Supported BSPs	18
4 Known Problems	21
Defects Fixed in Wind River Linux 7.0	21
Usage Caveats	21
Known Problems	24
Documentation Updates and Errata	28
5 Wind River Workbench 4	29
Changes in This Workbench Release	29

Wind River Workbench 4	29
Wind River Run-Time Analysis Tools	31
6 The GNU Toolchain	33
GNU Toolchain Introduction	33
Toolchain Changes in This Release	34
Enhancements	34
Fixed Problems	34
Rebuilding the Toolchain from Source	34
Migration and Backward Compatibility	34
Known Problems	35
7 Package Lists	37
About Available Software Packages	37
Obtaining Package Lists and SPDX Licensing Data and Reports	37
8 Customer Services	39
Customer Services	39

1

About this Release

About Wind River Linux 7.0	5
Latest Release Information	6
Installation and Licensing	6
Migration and Backward Compatibility	9
Workbench Support	9
Product Updates as Patches	10

About Wind River Linux 7.0

These release notes cover Wind River Linux and related products.

Product Information

- Wind River Linux 7.0

Wind River Linux is the base product. Read the information beginning with this overview and including [Changes in this Release](#) and *Known Problems* for more information.



NOTE: Information pertaining to the Wind River Linux base product generally applies to add on products as well. Read this information in combination with information about any add-on products you are using.

- *Workbench for Linux 4*

Workbench is an add-on IDE for developing Wind River Linux projects.

Release information for additional products, such as Wind River Linux profiles, including the Wind River Linux Carrier Grade Profile, Wind River Linux Open Virtualization Profile, and Wind River Compute Node, is provided in product-specific release notes available at <https://knowledge.windriver.com>.

Reference Material

These release notes also include the following reference information.

- [Target Packages](#).

Lists of target packages available for Wind River Linux.

Latest Release Information

The latest information on this release can be found in the Wind River Linux area of the Knowledge Library on the Wind River Support Network site.

The Wind River Support Network site is here:

<http://www.windriver.com/support/>

You can access the Knowledge Library here: <https://knowledge.windriver.com>.

This site includes links to topics such as known problems, fixed problems, documentation, and patches.

For complete details of test results, please contact your Wind River representative.



NOTE: Wind River strongly recommends that you visit the Wind River Support Network site before installing or using this product. The Wind River Support Network may include some mandatory software patches or other critical information regarding this release.

Installation and Licensing

Installing Wind River Linux requires that host system prerequisites be met, followed by installation.

For information on installing your platform and configuring your product licenses, see the Wind River product installation and licensing guides.

These are accessible from the following URL:

<http://www.windriver.com/licensing/documents>



NOTE: Before installing the product be sure you have installed the host software required for proper product installation as described in [Host System Requirements](#) on page 11.

You can install Wind River Linux as any user, including the root user, and make your installation available to other users, but you should not create or build projects as the root user.

The location you choose to install your software in is referred to as *installDir* in the documentation.

Refer to [Host System Requirements](#) on page 11 for information on supported development hosts.

Refer to [Target System Requirements](#) on page 13 and to the Wind River Knowledge Library for information on supported targets.

Minimal Install

You can opt for a minimal installation of Wind River Linux instead of a full installation by running the git-based installer, **install-WRL7.pl**.

During install time, only the configure system and a minimal number of recipes are downloaded.

During the build process of a platform project, additional downloads are performed when working with a minimal install. The default directory for these downloads is **installDir/cached_downloads**.

This enables multiple platform projects to reuse the already-downloaded packages to save disk space. However, if write permissions are not set up properly in the *installDir*, then the default directory is **projectDir/bitbake_build/downloads**. If the downloaded packages are installed to this directory, they will not be reused.

Alternatively, you can specify the location of the download directory by using the **configure** option **--with-dl-dir**. For details, refer to *Wind River Linux User's Guide: Platform Project Image Development*.

When performing a product update, your minimal installation is updated. A product reinstall is required to switch to a full installation.

Performing a Minimal Install

Perform a minimal installation of Wind River Linux by downloading and running the git-based installer script, **install-WRL7.pl**.

Step 1 Log in to Wind Share at the following Web address:

<https://windshare.windriver.com/>

Step 2 Select **File Sharing** in the navigation panel on the left-hand side.

Step 3 Navigate to the home page of **File Sharing** as illustrated in the following figure:



Step 4 Select the directory **WRLinux-7-wrgit-installer**.

Step 5 Look for the **install-WRL7.pl** script and click **Download**.

Hover over the script file name to view the **Download** option.

Step 6 Save the script to a location on the host system.

Step 7 Run the script as follows:

```
$ ./install-WRL7.pl --minimal \
--url https://windshare.windriver.com/remote.php/webdav
```

Adding Workbench to a Minimal Install

By default, Workbench is not included in a minimal Wind River Linux installation, but you can add it.

- To add Workbench, perform one of the following procedures:

Options	Description
Install Workbench when a minimal install already exists	<ol style="list-style-type: none">1. Install Workbench by running the Installer program.2. Choose the same <i>installDir</i> as the minimal install.3. Select the Custom installation option and deselect the <i>wrlinux</i> contents.4. Finish the installation. The Installer program automatically registers the minimal install with Workbench.
Install a minimal installation of Wind River Linux in the same location as your existing Workbench installation	<p>Perform a minimal install by running the install-WRL7.pl script (the git-based installer) from within the <i>installDir</i>. This directory is the parent directory of workbench-4.</p> <p>For information on how to run the script, refer to Performing a Minimal Install on page 7.</p> <p>The script automatically registers the minimal install with Workbench.</p>

Options	Description
Install a minimal installation of Wind River Linux in a different location than your existing Workbench installation	<ol style="list-style-type: none">1. Perform a minimal install by running the install-WRL7.pl script. For information on how to run the script, refer to Performing a Minimal Install on page 7.2. In the <i>installDir</i> of Workbench, create a symbolic link that points to the wrlinux-7 directory of your minimal install. For example: <pre>\$ ln -s /home/wruser/wrlinux-7 wrlinux-7</pre> Workbench requires the installation's symbolic link to be named wrlinux-7 after the default installation directory name. If you name it something different, Workbench will not identify the installation properly.3. Navigate to the <i>installDir</i> of your minimal install.4. Register the minimal install with Workbench by running the following command: <pre>\$./setup/postinstall.sh</pre>

Migration and Backward Compatibility

Custom packages from previous releases of Wind River Linux may be imported to Wind River Linux 7.0 using the instructions in the Wind River Linux Migration Guide, 7.0.

The Wind River Linux Migration Guide provides instructions and links to specific migration tools depending on what version of Wind River Linux you are migrating from.

Wind River Support may also help you with any questions you have regarding migrating to Wind River Linux 7.0.

Workbench Support

Wind River Linux Platforms support user-mode debugging with the ptrace agent, and run-time analysis tools in Wind River Workbench 4, Linux Version.

If you intend to use Wind River Workbench with Wind River Linux 7.0, you must install Wind River Workbench 4. Wind River Workbench 3.3.4 or earlier is not supported with Wind River Linux 7.0.

Product Updates as Patches

There are many Wind River patches applied to the architectures, boards, and packages of Wind River Linux.

Wind River Linux releases product updates also called RCPLs, or rolling cumulative patch layers. These new updates contain many fixes and enhanced functionality for Wind River Linux, and are considered a mandatory update. For information on updating your system to the latest RCPL, see the *Wind River Linux User's Guide*.

For information on specific patches, refer to the documentation where the patch is located.

To discover and examine BSP kernel specific patches please follow the instructions provided in the BSP **README** file.

For further information, contact your Wind River representative.

2

System Requirements

Host System Requirements	11
Target System Requirements	13
Installation Requirements and Issues	14

Host System Requirements

The host is the computer on which you do your development work. Wind River Linux 7.0 is designed to run on any host that meets the requirements in this section. These requirements assume that the target is a separate computer.

While many Linux distributions are capable of running Wind River Linux, Wind River provides a list of recommended host distributions and their required packages at:

<http://www.windriver.com/support>

Once you log in, select **Knowledge Library > Products > Operating Systems > Linux > Linux 7 > Getting Started** to access the *System Requirements - Recommended hosts list* link with information on the recommended host distributions.

For a list of supported hosts for Wind River Workbench, go to <http://windriver.com/products/workbench/> and select the **Download** link for **Wind River Workbench Product Note**.

Wind River Linux is tested only against recommended hosts to help us identify and correct problems early on through our own Q&A process. Using a recommended host helps speed up your initial development effort while decreasing the possibility of running into unknown host-related problems.

However, Wind River Linux is supported on any hosts that meet the requirements listed in the following paragraphs. If your chosen host meets the requirements and you still have issues, Wind River will help you resolve them.

Development Host Requirements

The build system is largely self-hosted, but relies on the host environment for basic functionality.

About Host System Requirements

Host system requirements include:

- The host system must have standard Unix file system semantics, and compatible versions of some basic tools and utilities. See [Host System Requirements](#) on page 11 for additional information on accessing the *System Requirements - Recommended hosts list*.
- At least 100 GB of disk space for the host system distribution, Wind River Linux and required packages, and basic platform development. Up to 500 GB of additional disk space may be required depending on the number of platform projects you plan to develop and their requirements.
- The build system does not require root privileges, though you may need admin access to configure the host to meet host requirements.
- The exact host tool requirements vary, but the following are the most likely to require additional effort:
 - You must have a working C compiler to build host tools.
- Builds must take place on file systems which provide stable and consistent file system behavior. The ext2, ext3, and ext4 file system family are known to work with default options. Specific requirements include:
 - Timestamps which consistently reflect the time at which operations occur, and which are monotonically increasing. Networked file systems (NFS) can produce very strange results if clocks are not synchronized.



NOTE: Wind River provides limited support for building across NFS as described in the following section.

- Reliable locking using flock(2), with persistence into a child process is mandatory.
- Standard permissions behavior. In particular, non-root users must not be able to give away files.

About Development on Microsoft Windows

Microsoft Windows 7 is supported for application development only. Wind River provides cross-compile toolchains, and the ability to create an SDK for use on a Windows host. The Windows host must meet the requirements for running Wind River Workbench to successfully develop applications using the Wind River Linux SDK.

About Support for NFS Installation and Builds

Wind River discourages NFS builds. The following NFS-related features are supported:

- Installation on an NFS server
- Using a SDK generated as part of a platform project build with NFS
- Configuring a platform project with parts of the file system located on NFS.

To make this work, you must specify the location of the temporary directory where BitBake maintains build-related files on a local disk or ramdisk. To do this, use the **--enable-tmpdir=***path_to_Dir* **configure** option when you configure your platform project.

The following is not supported:

- Building on a NFS file system

When doing builds on NFS mounts, the file server and build machine must have synchronized clocks. If not, some package builds may fail. A possible solution is to use a Network Time Protocol (NTP) server. Red Hat Linux Enterprise WS includes NTPD. In addition to NTP issues, some operations require static inodes to be assigned to files in order to emulate special files, as well as owners/groups/permissions. [WIND00419137]

Necessary Development Host Libraries and Executables

Wind River Linux 7.0 requires a number of host libraries and executables to be installed into the host Linux distribution (Ubuntu 12.04 for example).

As part of the installation process, you will receive a command-line message that lists the required packages, with instructions to run the following script in a console to install packages automatically, after installation completes:

```
$ installDir/wrlinux-7/scripts/host_package_install.sh --install
```



NOTE: The example above includes the **--install** option, for installing required packages. For additional options, run the script with the **--help** option.

In addition, the **configure** script automatically checks for the required executables and libraries on your host. If you do not have the necessary host executables and binaries installed, you will receive a corresponding message from **configure** when you try to configure a platform project, which is saved in the **config.log** file.

When this happens, it will be necessary for you to install the missing packages identified during the build process manually.

Note that the Git version control system is required on all platforms.

Wind River provides a list of required packages for our recommended hosts. See [Host System Requirements](#) on page 11.

Target System Requirements

The target is the computer for which you are developing. Typical minimal requirements must be met as a prerequisite to running host and target on separate computers. These requirements assume that the host is a separate computer.

This section lists typical requirements.

- Wind River supported target board.
- 64MB RAM is a reasonable minimum but requirements vary. More, for example, if a ramdisk is used.
- Space for file system on hard disk, ramdisk, flash memory, floppy, or hard disk.
- Ethernet or serial connection.
- Keyboard and monitor where supported by target (recommended for configuration if using network booting).

Refer to the board-specific **README** files for details on specific target requirements.

Supported Target Boards

Explore links to information on target boards supported by Wind River Linux.

Refer to [Supported BSPs](#) on page 18 for a list of board support packages (BSPs) shipped with Wind River Linux 7.0.

Targets Simulated by QEMU

QEMU, an open source machine simulator, allows you to configure the kernel and root file system for particular boards for which Wind River has created board support packages (BSPs).

QEMU currently supports the following BSPs:

- qemuarm
- qemuarma9
- qemumips
- qemumips64
- qemuppc
- qemux86
- qemux86-64

Refer to the BSP **README** file for specific information on the BSP you wish to simulate.

For a list of QEMU BSP names with cross-references to QEMU BSPs from previous releases, see the *Wind River Linux User's Guide: Introduction*.

Targets Simulated by Simics

Wind River Simics is a fast, functionally-accurate, full system simulator.

Simics creates a high-performance virtual environment in which any electronic system – from a single board to complex, heterogeneous, multi-board, multi-processor, multi-core systems – can be defined, developed and deployed.

Wind River Simics, an optional product offered by Wind River, supports virtually any target platform, and is a fully developed integration layer for Wind River Linux 7.0. For your convenience, an evaluation copy is provided for all Wind River Linux 7.0 customers.

Installation Requirements and Issues

Installing Multiple Products Into the Same Location

If you install an additional product into the same location (directory or folder) as an existing product, you must first run the existing product installer and provide it the new product license

key to add necessary base features and then run the installer on the new media and install the desired products.

For example, if you are adding a new add-on profile, such as Wind River Linux Carrier Grade Profile, to an existing Wind River Linux 7.0 installation, do the following:

- Step 1** Run the existing product installer from the Wind River Linux 7.0 media
- Step 2** Follow through all the prompts, supplying the new license key when the license key is requested.
The new license key is required for the ESD tool to identify the product(s) you are licensed to install.
- Step 3** Verify that the installer retrieves product information specific to your license.
Ensure that the product is selected to install it.
- Step 4** Follow through all the prompts to complete the installation.
You can re-use the existing key when the license key is requested.

3

Changes in This Release

[About Changes to Wind River Linux 7.0](#) 17

[New or Enhanced Features](#) 17

About Changes to Wind River Linux 7.0

Wind River Linux 7.0 leverages the Yocto Project build system components.

The details are highlighted in this section. For changes related to other products included with your Wind River Linux purchase, see the following sections:

- [Wind River Workbench 4](#) on page 29
- [GNU Toolchain Introduction](#) on page 33

New or Enhanced Features

Learn about major new and enhanced features available with Wind River Linux 7.0.

For details on these and all Wind River Linux features, refer to the online documentation.

Quick Start SDK for Windows Application Developers

For Windows-based Wind River Linux installations, a 32-bit SDK is provided to help application developers get started for the following architectures:

- ARM
- ia32
- MIPS
- POWER

The SDK is available in the **`installDir/wrlinux-7/SDK/wr-sdk-version/arch-quickstart-686-mingw32/arch`** directory. You may use the SDK directly from the command line or imported into the Workbench environment. For additional information, see the *Wind River Linux User's Guide: Using the SDK*

Run the Linux Standard Base (LSB) Tests

Using the **lsbtesting** feature template, you can configure a platform project to include the necessary components to run the LSB tests to verify that your target platform image is LSB-compliant.

New Use Cases and Enhanced Procedures

New use cases and procedures are supported in Wind River Linux 7.0.

The following new use cases and procedures are included:

- Enhanced procedures for working with and using the Wind River Linux SDK
- Git-based installer/update tool
- Creating Subset Repositories of meta-openembedded Layers
- New Platform Project Configure tool works with Workbench and from the command-line
- Securing sstate Cache – read-only and verification/signing options
- Verifying Builds with the Image Manifest (feature/image-manifest)
- Managing Builds with the Yocto Project Toaster
- Maintaining Open Source License Compliance (feature/archiver)
- New section on Creating Optimized Custom Kernel Builds, includes:
 - Creating a Platform Project with a Dummy Kernel
 - Building the Kernel Using the Custom Kernel Recipe
 - Building the Kernel from External Source
 - Extracting the Kernel Build Output
- Deploying an Image with a Virtual Machine Manager (VMWare vmdk support)
- Application project creation and debugging via TCF/GDB
- New System Analysis
- New Memory Analysis

Supported BSPs

Wind River Linux provides some BSPs as part of the installation, and additional BSPs are available through ESD.

A number of BSPs are shipped with Wind River Linux 7.0:

- qemuarm
- qemuarm9
- qemuarm64
- qemux86
- qemux86-64
- qemumips
- qemumips64
- qemuppc

Additional BSPs are available through electronic software delivery (ESD) once you install Wind River Linux 7.0. These include:

- cav-octeon3
- cav-thunderx
- fsl-t2xxx
- fsl-ls10xx
- intel-x86

Contact your Wind River representative to learn about newly available architecture support.

4

Known Problems

[Defects Fixed in Wind River Linux 7.0](#) 21

[Usage Caveats](#) 21

[Known Problems](#) 24

[Documentation Updates and Errata](#) 28

Defects Fixed in Wind River Linux 7.0

Refer to the Knowledge Library on the Wind River Support Network for the current list of defects.

The Knowledge Library provides the latest list of defects and additional information such as workarounds to current issues. You can access it here:

[Defects for Linux 7](#)

You can also access the Wind River Linux defects list using the following link or by logging in to the Knowledge Library. Locate the defects list by selecting **Products Operating Systems Linux Linux 7** and then clicking **-Choose-Defects for Linux 7**.

For additional information, see [Known Problems](#) on page 24 and [Usage Caveats](#) on page 21.

Usage Caveats

Some caveats may apply to using certain features of Wind River Linux under specific conditions.

The following list describes caveats in detail.



NOTE: Where applicable, Jira defect numbers are provided (for example [LIN-1234]) for help in tracking the issue.

Local product installation fails for missing repository [LIN7-3049]

If installing from an incomplete download, a required component may be skipped with the message:

```
WARNING: skipped due to missing: directory_path
```

In this example, *directory_path* refers to the location of the missing repository, for example:

```
WARNING: skipped due to missing: /data/wruser/Build/03011747-  
windshare_product_download  
GIT_20150301/NA-NA-NA/LB21_7.0_RCPL0003/WRLinux-7-Core//LB21_7.0_RCPL0003-Core/  
anongit.freedesktop.org.git.xorg.proto.dri2proto
```

If you are receiving such warnings, your installation is not complete. To complete the installation, perform another download of the release to recover the missing content, and re-install the product.

qemu-native cannot start graphics on remote target [LIN7-2009]

To use the QEMU graphics console (**make start-target TOPTS=-gc**) you need to add the following to **local.conf** before building:

```
PACKAGECONFIG_append_pn-qemu-native = " sdl"  
PACKAGECONFIG_append_pn-nativesdk-qemu = " sdl"  
ASSUME_PROVIDED += "libsdl-native"
```

This is required for both **qemux86** and **qemux86-64** targets.

Building Kernel Modules from the SDK

When building kernel modules using an SDK (from the output of **make export-sdk**), the environment configuration assumes that **gcc** will be used for linking and sets the **LDFLAGS** as arguments to **gcc**. The kernel, however, uses **ld** directly and the **LDFLAGS** specified by the environment may not be valid for **ld**.

It is advised that you use the following command when building kernel modules:

```
LDFLAGS="" make -C $KERNELDIR M=/path/to/module
```

Enabling commercial Wind River-supplied packages with commercial licenses

Wind River Linux 7.0 oe-core contains eight packages with commercial licenses: **gst-fluendo-mp3**, **gst-openmax**, **gst-plugins-ugly**, **lame**, **libmad**, **libomxil**, **mpeg2dec**, **qmmmp**.

To use these packages in your platform project image, you must add them to your platform project's **projectDir/local.conf** file, to the **LICENSE_FLAGS_WHITELIST** section. The following example adds all of these packages to the whitelist, enabling all packages for your platform project image:

```
LICENSE_FLAGS_WHITELIST = "commercial_gst-fluendo-mp3 \  
                           commercial_gst-openmax \  
                           commercial_gst-plugins-ugly \  
                           commercial_lame \  
                           commercial_libmad \  
                           commercial_mpeg2dec \  
                           commercial_qmmmp"
```

```
commercial_libomxil \  
commercial_mpeg2dec \  
commercial_qmmp"
```

Add or remove the files to include or exclude them for development.

Limitations with BusyBox getty for glib_small rootfs

When using telnet on ttyS0 to connect to a target file system using SSH via dropbear, the expected startup behavior is to first view an introductory screen, which returns to a command prompt when you press Enter. If the terminal becomes garbled and unresponsive, it may be due to the limited implementation of getty that BusyBox provides. To overcome this limitation, install the **util-linux-agetty** package on the target. [LIN5-8699, LIN5-1052]

Video encoding and decoding support

Due to licensing issues, Wind River Linux 7.0 does not contain support for video decoding or encoding. Talk to your Wind River account manager if you need help with this feature. [LIN4-702]

ustar format limitations

Due to limitations of the ustar format which is forced for compatibility by **automake**, users performing platform builds must use a **uid** and **gid** smaller than 32 bits (less than 2097151).

Gnome menu icons

In some new Gnome distributions, menus do not show icons by default, unlike previous releases. If you want to enable menu icons, go to the configuration center, locate **gconf-editor**, and modify the setting for **desktop > gnome > interface > Menus have icons**.

Git checkpoint_end not supported

Using **checkpoint_end** with **git** is no longer supported. Use **wrs_meta** instead. For example, do *not* use:

```
$ git show checkpoint_end \  
| filterdiff -i '*common_pc*.scc' | patch -s -p2 \ -d /tmp
```

Instead, use:

```
$ git show wrs_meta^^^^..wrs_meta \  
| filterdiff -i '*/common_pc-*.scc' | patch -f -s -p2 -d /tmp/foo
```

Distribution support for time command

While all supported distributions include the **time** command, it may be missing on others. If this is the case for your environment, installing **time** is a prerequisite for using Wind River Linux.

Target out of memory conditions

When running a high volume or complex tests on a remote target, Out of Memory conditions may occur. This is normal on systems with limited memory.

License File Checksum Errors

It is possible for a recipe to process and fail for a license file checksum error, even though no real error is present [LIN5-5195].

This issue has been discovered on the **perf_3.4.bb** and the **virtio-trace.bb** license files. What happens is that the recipe is processed before the sysroot is present during the build process. This action causes the checksum to be invalid, and return an error.

To correct this,

- open the license file and add or edit the following value to read:

```
do_populate_lic[depends] += "virtual/kernel:do_populate_sysroot"
```

Using \$LD and \$LDFLAGS for linking applications and building kernel modules

Applications

When linking applications as part of using the SDK for development, you should only use **\$LDFLAGS** and **\$CC**, and not use **\$LD**. This is required because the compiler has built-in references to the support libraries for building the application(s).

Kernel modules

When building kernel modules, do not use **\$LDFLAGS** at all. The kernel sources know the explicit set of variables required when linking kernel modules. If you need to compile kernel modules in the directory created from extracting the SDK, you must first unset **\$LDFLAGS** using the following command:

```
$ unset LDFLAGS
```

[LIN6-59]

IPC pthread_mutex deadlock with installed signal handler

If a signal handler is installed in a multi-threaded application that uses shared resources, you will have "undefined behavior" as described in the Linux man pages. This may result in segmentation fault error messages, and may cause the mutex, or other shared resource(s), to not be freed or unlocked when the process dies.

This behavior is consistent in all known kernel versions. [LIN5-18807]

Known Problems

These are problems known at release.

For the most current list of known problems for this release, see the following Web site:

<http://www.windriver.com/support>

The current status of these issues is available online.

Note that you are strongly encouraged to read your board's **README** file, available in:

projectDir/layers/wr-bsps/boardName

These and other **README** files are automatically copied into your build directory at configure time.



NOTE: Where applicable, Jira defect numbers are provided (for example [LIN-1234]) for help in tracking the issue.

cleanall removes downloaded packages

If working with a minimal installation of Wind River Linux, performing a cleanall will remove all downloaded packages. The packages will be downloaded again during build time.

Heapscope not supported on MIPS and aarch64

Heapscope does not work on MIPS and aarch64 targets because these do not currently support uprobes, which are required for Heapscope.

Heapscope not supported with preemp-rt kernel

Heapscope does not work with the preempt-rt kernel because the preempt-rt kernel does not currently support uprobes, which are required for Heapscope.

Enabling Uprobe tracing on the ARM BSP fsl-ls10xx causes target to hang

The pre-built glibc is built using the Thumb-2 instruction set on ARMv7 machines. However, the ARM uprobe does not currently support Thumb-2. If you want to use ARM uprobe on an ARMv7 machine, the glibc should be rebuilt using the option **--with-template=feature/build_libc**. Also, the probed application should be built without Thumb-2 instructions, as well.

Internal compiler errors

Receiving internal compile errors that read:

```
arch-wrs-linux-gnu-gcc:
Internal compiler error: Terminated program as
```

The variable *program as* in this example may be **cc1**, or for the arm compiler, it may be **wrs-linux-gnueabi-gcc**, or even **-g++** [LIN5-793].

In rare instances, you may receive this warning. If you do, simply rebuild your platform project image to correct it.

Starting the HOB on Red Hat Enterprise

You may not be able to start the HOB on a Red Hat Enterprise Edition 6.3 due to lack of Red Hat support for the most recent Gtk, PyGtk, and PyGobject packages [WIND00398074].

The HOB requires Gtk 2.2 and PyGtk2 2.21, but Red Hat provides Gtk 2.18.9-10 and PyGtk2 2.16.0-3.

Receiving do_patch failure on platform project build

If you receive the following warning while building a platform project with the **util_linux** package:

ERROR: No valid terminal found, unable to open devshell

you must perform a distclean on the package, and rebuild the file system:

```
$ make -C build util_linux.distsclean
$ make
```

[LIN5-1431]

Some programs will not build due to missing symbolic constants or functions which are not part of a base POSIX system

Clean-up efforts in glibc have marked a number of extensions and non-standard features to be included only when you have requested them. To re-enable many of these optional features, use **-D_GNU_SOURCE** when compiling.

Statically linking busybox with glibc

You cannot statically link **busybox** with **glibc**. It is possible to use the configure flag **--enable-scalable=mklibs** when static linking, though it might be buggy and is probably a bad idea. See http://sources.redhat.com/bugzilla/show_bug.cgi?id=3400.

Mtab lock file moved

The **mtab** lock file has been relocated to **/var/lock**, from **/etc**. According to the Filesystem Hierarchy Standard, **/etc** is not guaranteed to be writable.

When **/etc** is mounted as part of a read-only partition, you may need to pass **-n** to various **mount** commands in order to modify the existing **mtab** file. (Mounting new file systems will always work, modifying mounts may be an issue.) **mount** attempts to generate a **tmp** file in **/etc**, which is read-only, so it should fail. Even if it succeeds, **/etc/mtab** is **bind** mounted and cannot be overwritten with a rename.

Openssl certificate requests

To generate a certificate request with **openssl**, you should first download a **sign.sh** signing script (such as the one available at <http://www.faqs.org/docs/securing/chap24sec195.html>) and place it in **/usr/sbin**. You should modify it appropriately (for example, **/etc/ssl** should be changed to **/usr/lib/ssl**) to work in the Wind River Linux environment.

Using rpm on a target

When using **rpm** on a target, automatic repackaging of an erasure or upgrade is disabled. To enable it, set the value `%_repackage_all_erasures` to a non-zero value in the **/usr/lib/rpm/5.0/macros** file. You must also create a **/var/spool/repackage** directory.

Using gdb and gdbserver with glibc thread debugging

gdb is unable to trap some of the signals from threaded programs because the build system strips the binaries when the target file system is constructed. **gdb** needs some of the symbols out of the **libpthread.so** to find all of the threading information.

The workaround is to install the **debuginfo** files, or use a debug build of **glibc** in order to debug threaded applications.

Running **gdb** on the Target

1. Copy the **export/RPMS/arch/glibc-debuginfo-2.5-1.arch.rpm** from the host to the target.

2. On the target run (as root):

```
# cd /
# rpm2cpio glibc-debuginfo-2.3.6-1.arch.rpm | cpio -id
```

This will install files into **/usr/lib/debug** and **/usr/src/debug** (if there are any associated sources).

When the (target) **gdb** is run, it will detect that the symbols are stripped from the **libpthread_db.so**, and automatically look at **/usr/lib/debug** for the symbols.

Cross Debugging (using **gdbserver** to attach to a process)

1. On your host system, extract the debug information:

```
$ cd local_debug_path
$ rpm2cpio ../glibc-debuginfo-version.arch.rpm | cpio -id
```

2. On the target, setup **gdbserver**:

```
# gdbserver host_ip:port --attach pid
```

3. Execute the following back on the host:

```
$ ./host-cross/bin/i586-wrs-linux-gnu-gdb path_to_app/mthread.out
$ set sysroot target_sysroot_path
$ set debug-file-director local_debug_path /usr/lib/debug
$ target remote target_ip:port
```

In either case, you can verify that **gdb** has the proper thread information for debugging available by using the **gdb** command **info threads**. You should see a list of all of the threads. If you get either blank or one thread (when you should have more) than **gdb** was unable to load all of the symbols from the debug symbols version.

The **--cref** option to the GNU linker causes the linker to abort

Omit this option to create binaries successfully. [LIN2-1661]

Warnings when building the kernel

Some warnings remain when building the kernel, these warnings have been audited for safety and remain due to constraining factors such as upstream compatibility, code stability, compiler requirements and patch stack maintenance.

ALSA library compatibility

The user space ALSA libraries and utilities will only build/run in a glibc rootfs. They are included in **glibc_std** and **feature/demo pkglist.add** files.

Production build default optimization

To ensure proper compilation and runtime behavior, the default optimization levels for production builds is **-O2**. The optimization flags are:

- **-O** Some optimizations are enabled.
- **-O2** More optimizations are enabled, but none that increase the size of binaries.
- **-O3** Maximum optimization.
- **-Os** Optimize for binary size.

For builds with **--enable-build=debug**, no optimization is specified.

Some packages will not compile or execute properly without optimization turned on: **glibc**, **openssh/ssl** and the kernel are examples; for these packages the lowest possible optimization setting will typically be used.

The default optimization level can make debugging user applications difficult due to instruction pointer bouncing. User applications that are not part of the default distribution can be built via workbench or from the command line with lower or no optimization. The system wide optimization level for debug builds can be modified in the CPU template **config.sh** file by altering/append to the following variable:

```
TARGET_CDEBUG_LEVEL="-g -O value"
```

Similarly the production optimization level is controlled by:

```
TARGET_COPT_LEVEL="-O value"
```

where *value* is replaced by the desired optimization value. Note this value affects all packages and if changed from the default value, the runtime behavior cannot be predicted.

Custom patching packages

It may be necessary, when custom patching a package, to remove old directories, sstate files, and so forth. In these cases, use **make -C buildpackage.distclean** (within the project build directory), to remove the package build directory, the package source directory, and the package's associated sstate files. If that is not sufficient, the use **make -C buildpackage.cleansstate** to force a clean build state.

MIPS board limitations

MIPS boards with 32-bit PCI can only access 1GB of DMA memory.

MIPS kernel limitations

There may be MIPS 64-bit kernel DMA problems with 32-bit PCI cards.

Documentation Updates and Errata

Documentation may be updated periodically as new features are added or issues are discovered.

This may include updates to the *Wind River Linux User's Guide*, *Wind River Linux Getting Started Guide*, and the *Workbench by Example Guide, Linux Version*.

To get the latest versions of these guides, as well as the other Wind River Linux documents including a complete, visit the Wind River Knowledge Library at: knowledge.windriver.com.

5

Wind River Workbench 4

Changes in This Workbench Release 29

Changes in This Workbench Release

This release of Workbench includes a number of changes.

For details on Workbench integration with these changes, see the *Wind River Workbench User's Guide*.

Wind River Workbench 4

Wind River Workbench 4 provides product-specific features to complementary Wind River products.

Wind River Linux 7.0 Support

Wind River Workbench supports Wind River Linux 7.0 with the following features:

- A new wizard is available for creating, launching and managing QEMU connections
- The Feeds View is available to obtain the latest information on Wind River Linux.

To access it, select **Window > Show View > Feeds** from the main Workbench menu.

- Windows host is supported for application development, with build, debug, and analysis applications for remote Wind River Linux targets
- Application project creation and debugging with TCF/GDB
- New platform project configuration and management tools
- New CPU Profiler tool
- New Memory Analyzer tool

Workbench 4 provides four key value enhancements for Wind River Linux, compared to Workbench 3:

Performance Profiling

Performance Profiling is now based on the Open Source perf engine instead of oprofile. A flexible UI Framework is built on top of perf, to support back-in-time navigation on profiling results, live reconfiguration of the profiling session, graphical correlation of multiple data elements, and both live and offline data collection. Arbitrary data sources can be used through perf uprobe and kprobe support with on-target aggregation to manage huge data streams.

Memory Analysis

Memory Analysis is now based on perf uprobes, using the same unique data aggregation engine on the target to deliver a 20x performance improvement compared to mpatrol, the unique ability to attach to a running system for memory leak detection, and a unique graphical view to analyze the data in multiple domains including graphical leak suspects and a histogram. For a video demo of this capability, visit <http://knowledge.windriver.com>.

Indexed Text Search

The Indexed Text Search can now access directory structures outside the Eclipse workspace. This helps for navigating and understanding even huge Platform projects quickly.

wrttool Scripting

The wrtool command-line utility provides scripting support of the Eclipse IDE to create and manage project structures in an automated way.

Other capabilities have been re-done in a different way compared to Workbench 3:

- Debugging has been re-based on Open Source gdb, thus standardizing on common workflows. As a result, the graphical Debug Symbol Browser is no longer available in Workbench 4. Instead, you can use gdb commands on the GDB Console to look up symbols. Multi-thread debugging and follow-fork-mode work differently than in Workbench 3; see the documentation for details.
- Target Connectivity has been simplified; most target settings can now be auto-discovered and a single wizard page is sufficient for setting up any target. Special connection types for QEMU and Coredumps do no longer exist. Instead of the specialized QEMU connection type, a QEMU Launch Type will be provided with a Workbench Update in the future. For now, use command-line operations (**make start-target**) to launch QEMU instances.
- For LTTng Tracing, the Eclipse Open Source LTTng Viewer is now provided as a powerful, flexible mechanism for configuring, controlling, and uploading trace. In addition to the Open Source visualization of LTTng traces, the System Viewer visualization is still provided as an optional add-on for enhanced scalability through its database-driven approach and advanced filtering. Especially for follow-the-CPU visualization, the System Viewer visualization is still superior to the Open Source visualization.
- The Wind River GIT and Quilt integrations have been replaced with the more powerful, flexible Eclipse EGit plug-in for history display and team operations.

Some other Workbench 3 elements are not currently available in Workbench 4:

- Package Projects and Layer Projects are no longer provided as special project types. Advanced editing support for BitBake scripts is planned for a future update instead.
- UI integrations for Valgrind and Coverage are no longer provided in Workbench 4. Please use the command-line gcov and valgrind tools instead. A re-base of this functionality on Open Source Linuxtools is considered for a future update instead.
- The Getting Started Screen is no longer provided in this release. For getting late-breaking information and question and answers with experts, use <http://knowledge.windriver.com>, as well as other advanced online tools hosted by Wind River instead.

Workbench support has the following exceptions:

Wind River Run-Time Analysis Tools

Learn about enhancements and improvements to the Run-Time Analysis Tools (formerly ScopeTools) provided with this release.

This section describes those changes.

Heapscope

When heapscope is used to start a process, all child processes will also be instrumented and their allocations will be tracked.

In order to use heapscope on 32bit processes when the default configuration is 64 bit package builds, you must include the 32bit heapscope package using the usual commands:

```
$ make -C build lib32-heapscope.addpkg  
$ make [fs]
```

Heapscope will then correctly instrument and track allocations for the children of the different multi-lib types when you use heapscope on a 64 (or 32) bit process that spawns a 32 (or 64) bit process.

6

The GNU Toolchain

GNU Toolchain Introduction	33
Toolchain Changes in This Release	34
Migration and Backward Compatibility	34
Known Problems	35

GNU Toolchain Introduction

The Wind River Linux toolchain provides C and C++ compilers, as well as a linker, assembler, debugger, and related utilities. The toolchain is based on the OpenEmbedded-Core sources, with additional patches to enable new processors, boards, and various features. It is delivered as a precompiled binary to help avoid the time penalty of compiling the toolchain, as well as eliminate common problems due to host distribution bugs.

As of the release of Wind River Linux 7, the toolchain versions align with the Yocto Project 1.7 (Dizzy) release. Collectively, these components are packaged into a binary toolchain SDK, with a version of wr17-5. Future toolchain patches will update the number after the hyphen which identifies a specific toolchain release, or "drop." When submitting bug reports about the toolchain, please be sure to indicate the specific toolchain drop you are using, as issues present in earlier drops may be corrected in later drops.

Users are often curious as to the "exact" versions of various toolchain components. Because Wind River Linux is extensively updated from the upstream versions, the "exact" version is really the base version (indicated above) plus the drop number revision for all components. This corresponds roughly to gcc version 4.9.1, binutils version 2.24, glibc version 2.20, and gdb version 7.7.1. When checking for feature availability, be sure to check the features actually provided by the toolchain, rather than looking at upstream project feature checklists; the Wind River Linux toolchain often has enhancements or features that were not in the corresponding upstream tree until later versions. Note that the toolchain version reported may vary with architecture.

This release supports the GNU tools for the host platforms and target architectures listed in [Host System Requirements](#) on page 11. Further, Wind River ships and supports only the C and C++ compilers, although other languages are included in the GNU Compiler Collection.

The most extensive information on GCC is available from the Free Software Foundation (FSF) at <http://gcc.gnu.org/releases.html>, which developers are encouraged to consult. Some of the information on the FSF Web site, however, may not apply to Wind River products, and Wind River does not support all features of GCC.

Toolchain Changes in This Release

Enhancements

Learn about enhancements to this version of the GNU tool chain.

Fixed Problems

A list of known issues that have been addressed in toolchain updates may be found by searching the Wind River Knowledge Library.

Changes were made to address rare code generation issues, prelinker issues, and occasional issues with the debugger.

Here is a link to the Knowledge Library:

<https://knowledge.windriver.com>

Rebuilding the Toolchain from Source

Learn about caveats and restrictions when rebuilding the GNU toolchain from source.

While building the toolchain from source is supported, the resulting binaries are not supported—all defects must be reproduced using the prebuilt binary toolchain.

Migration and Backward Compatibility

If you are migrating projects from earlier versions of GCC, you may need to recompile modules because of incompatible binary formats.

Also, some nonstandard language constructs allowed by earlier versions of GCC are no longer supported in the most recent releases; the **-fpermissive** option may provide a short-term workaround for some projects.

The default semantics of inline functions changed from previous releases. If you have problems with inline functions you may wish to try the **-fgnu89-inline** compiler flag.

More information on changes between 4.x and 4.9 may be found at http://gcc.gnu.org/gcc-4.4/porting_to.html, and http://gcc.gnu.org/gcc-4.6/porting_to.html.

Known Problems

Learn about known problems with this release of the GNU toolchain.

See the FSF Web site at <http://gcc.gnu.org/releases.html> for information about known problems with the GNU tools.

Package Lists

[About Available Software Packages](#) 37

[Obtaining Package Lists and SPDX Licensing Data and Reports](#) 37

About Available Software Packages

A wide range of software packages are available for Wind River Linux 7.0

Refer to the package maintainer sites for details on new package features. In some cases, there may be more than one package version. Wind River Linux provides different versions to ensure compatibility with certain packages and their dependency requirements. This may cause a platform project build to use an earlier package version to ensure compatibility.

Obtaining Package Lists and SPDX Licensing Data and Reports

Wind River provides software package licensing information using the Linux Foundation's Software Data Package Exchange (SPDX) format. In addition, Wind River provides high-level human readable html-based reports for your review.

SPDX data is available in a git repository located in your Wind River installation. To use and view the data, you must clone the repository.

Step 1 Navigate to a location on the host system to clone the layer to.

In this example, you will navigate to the **/tmp** directory.

```
$ cd /tmp
```

Substitute the path as necessary to place the layer in a different directory.

Step 2 Clone the folder.

```
$ git clone installDir/wrlinux-7/ip-disclosures
```

Once complete, the contents of the repository will be available in the **/tmp** directory, for example:

```
$ ls /tmp/ip-disclosures  
Report Scripts standard_Notices Style TOC.htm
```

Step 3 Open the **TOC.htm** file to review SPDX data in a web browser.

The data is organized alphabetically. Use the A to Z links to locate licensing information for a specific package.

Customer Services

Customer Services 39

Customer Services

Wind River is committed to meeting the needs of its customers. As part of that commitment, Wind River provides a variety of services, including training courses and contact with customer support engineers, along with a Web site containing the latest advisories, FAQ lists, known problem lists, and other information resources.

Customer Support

For customers holding a maintenance contract, Wind River offers direct contact with support engineers experienced in Wind River products. The Customer Support program is described in the Standard Support User's Guide available at:

<http://www.windriver.com/support>

The guide describes the services available, including assistance with installation problems, product software, documentation, and service errors.

You can reach Customer Support by e-mail or telephone:

Location	Phone	Email
North and South America, Asia/Pacific (outside Japan)	800-872-4977 (toll-free)	support@windriver.com
Europe, Africa, Middle East	+(00) 800-4977-4977 (toll-free)	support-EC@windriver.com
Japan	81-3-5778-6001	support-jp@windriver.com

For detailed contact information, including contact information specific to your products, see the Support Web site shown above.

Wind River Knowledge Library

Wind River Customer Services also provides the Knowledge Library as part of the Wind River Support Network. The Knowledge Library provides advisories, online documentation, a list of training courses and schedules, and also provides access to software downloads, known problems lists, patches, answers to frequently asked questions, and demo code.