



AN INTEL COMPANY

# WIND RIVER® LINUX

## GETTING STARTED WORKBENCH TUTORIALS

8.0



## Copyright Notice

Copyright © 2017 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, Simics, Tornado, and VxWorks are registered trademarks of Wind River Systems, Inc. Helix, Pulsar, Rocket, Titanium Cloud, Titanium Server, and the Wind River logo are trademarks 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](http://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 for your product on the Wind River download and installation portal, Wind Share:

<http://windshare.windriver.com>

Wind River may refer to third-party documentation by listing publications or providing links to third-party websites 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.): +1-800-545-WIND  
Telephone: +1-510-748-4100  
Facsimile: +1-510-749-2010

For additional contact information, see the Wind River website:

[www.windriver.com](http://www.windriver.com)

For information on how to contact Customer Support, see:

[www.windriver.com/support](http://www.windriver.com/support)

*Linux*

*Getting Started Workbench Tutorials, 8.0*

26 April 2017

# Contents

<b>1 Developing Platform Projects .....</b>	<b>1</b>
Platform Projects Development Tutorial .....	1
Creating and Configuring a Platform Project .....	1
Building a Platform Project Image .....	4
Configuring a New Project to Add Application Packages .....	5
Adding New Application Packages to an Existing Project .....	8
Verifying that the Project Includes the New Application Package .....	9
Deploying a Platform Project Image .....	10
Platform Project Development Summary .....	14
<b>2 Developing Application Projects .....</b>	<b>15</b>
Application Project Development Tutorial .....	15
Creating an Application Project with Workbench .....	15
Building an Application Project .....	17
Creating a Target Connection .....	18
Deploying an Application Project with Workbench .....	21
Application Project Development Summary .....	23
<b>3 Debugging an Executable .....</b>	<b>25</b>
Executable Debugging Tutorial .....	25
Debugging an Executable with Workbench .....	25
<b>Appendix A: Wind River Education Resources .....</b>	<b>29</b>



## *Developing Platform Projects*

Platform Projects Development Tutorial 1

### **Platform Projects Development Tutorial**

Learn about platform project development on Workbench through a series of self-paced exercises.

#### **Overview**

When you have completed this tutorial, you will know how to:

- create a new platform project and configure it for your needs
- build your new platform project
- optionally add application packages when you create the project, or
- add application packages to an existing project
- verify that applications have been successfully added to your project, and
- deploy your platform image

### **Creating and Configuring a Platform Project**

You define parameters such as the BSP, the kernel, and the size of the target file system when creating and configuring a platform project.

To configure your platform project with Workbench:

## Procedure

1. Start Workbench.

Options	Description
<b>From the System menu</b>	Select <b>WindRiver &gt; Workbench 4 &gt; Workbench 4</b> .
<b>From the desktop</b>	Double-click the <b>Workbench</b> icon on your desktop.
<b>From the command line</b>	<ol style="list-style-type: none"><li>1. Navigate to the Workbench install directory: <pre>\$ cd installDir</pre></li><li>2. Enter the following command: <pre>\$ workbench-4/startWorkbench.sh</pre></li></ol>

2. Configure the platform project.

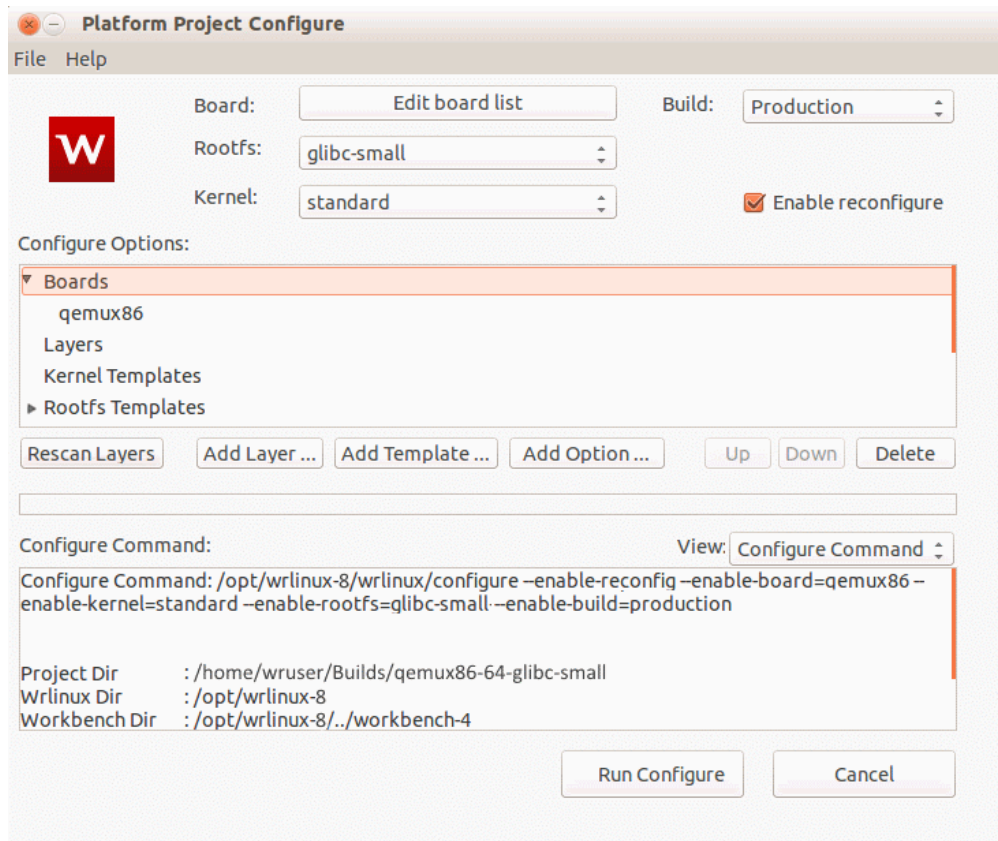
- a) Create a new project.

Right-click in the Workbench **Project Explorer** and select **New > Project**.

- b) Expand the **Wind River Linux** option, select **Wind River Linux Platform Project**, then click **Next**.

- c) Give your project a descriptive name, such as **qemux86-64-glibc-small**, and click **Finish**.

The Platform Project Configure dialog box appears.



In this example, you will configure and build a platform project using the following settings:

**Board:** qemux86-64  
**Rootfs:** glibc-small  
**Kernel:** standard

Accept the rest of the default settings in the Platform Project Configure dialog box. This creates a project with a standard kernel and a small file system for a qemux86-64 board.

- d) Optionally set additional configure options to greatly improve the time it takes to build your project.

Use the following configuration options to set values for parallel builds:

- **--enable-jobs=*n*** specifies the maximum number of parallel jobs (threads) that make can perform while building a single package. If this is not set, the default is 1.
- **--enable-parallel-pkgbuilds=*n*** specifies the maximum number of packages that may be built in parallel.



---

**NOTE:** BitBake automatically sets defaults for parallel jobs and packages to the number of cores on the machine.

---



---

**NOTE:** Workbench uses **ccache** by default, and it does not need to be manually configured.

---

e) Click **OK**.

3. Click **Finish** to configure the project.

### Postrequisites

When you have completed these steps and created a new platform project, you can build an image as described in [Building a Platform Project Image](#) on page 4.

## Building a Platform Project Image

After configuring a platform project you must build it.

### Prerequisites

Before proceeding, you should have a created a project as described in [Creating and Configuring a Platform Project](#) on page 1.

To build your platform project with Workbench:

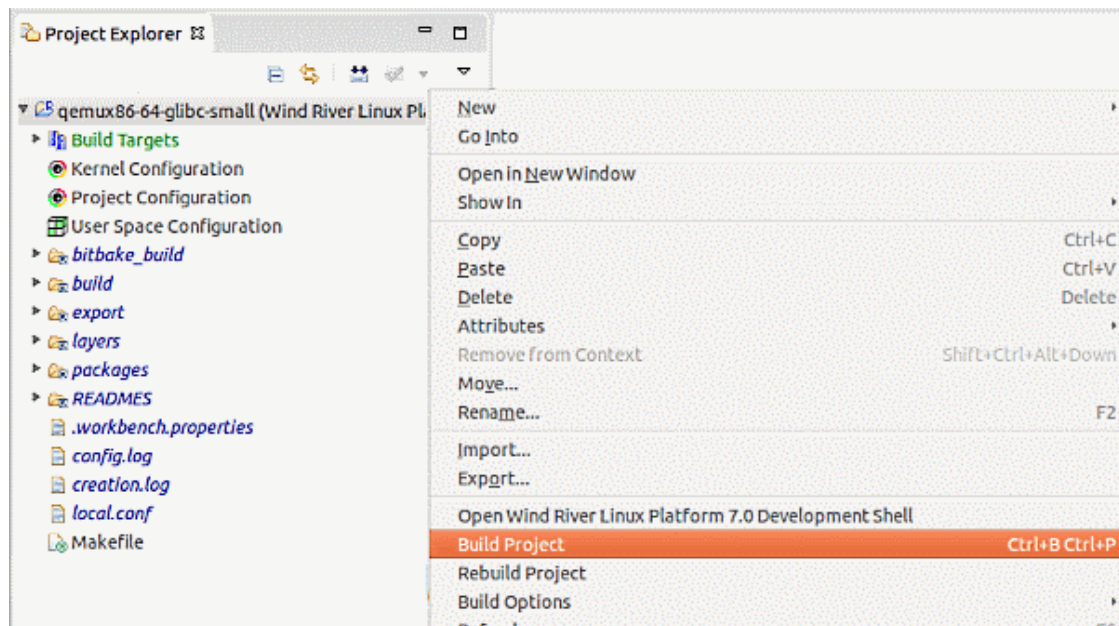
### Procedure

1. Expand the project in Workbench.

If necessary, expand the project by clicking on the arrow to the left of the project name. Inside the project you can see various tools, build targets, and files associated with the project.

2. Build the project.

Right-click on the platform project folder in **Project Explorer** and select **Build Project**.



Building could take from minutes to hours, depending on your development resources and project configuration. When the build is finished, you will have a kernel and file system ready for deployment to a real or simulated target as described in the *Wind River Linux Getting Started Command Line Tutorials: Building a Platform Project Image*.



## Postrequisites

When you are comfortable with this procedure, learn about options for adding an application in [Configuring a New Project to Add Application Packages](#) on page 5 and [Adding New Application Packages to an Existing Project](#) on page 8.

## Configuring a New Project to Add Application Packages

You can add application packages to your platform project using the New Platform Project wizard.

This topic describes how to add Wind River Linux layers and templates to a platform project configuration. For details on adding custom layers and templates, see the *Wind River Linux Platform Developer's Guide: Platform Project Image Configuration Overview*.



**NOTE:** The functionality added in this example (the target-resident debugger) is currently only supported on targets with the x86 architecture (32and 64-bit), but the workflow for adding a non-default layer and templates is the same for all architectures and BSPs.

To customize a platform project with Workbench:

## Prerequisites

Before adding an application, you should have created and built a platform project as described in [Creating and Configuring a Platform Project](#) on page 1 and [Building a Platform Project Image](#) on page 4.

## Procedure

1. Start Workbench.

Options	Description
From the System menu	Select <b>WindRiver &gt; Workbench 4 &gt; Workbench 4</b> .
From the desktop	Double-click the Workbench icon on your desktop.
From the command line	<ol style="list-style-type: none"><li>1. Navigate to the Workbench install directory: <pre>\$ cd installDir</pre></li><li>2. Enter the following command: <pre>\$ workbench-4/startWorkbench.sh</pre></li></ol>

2. Configure the platform project.

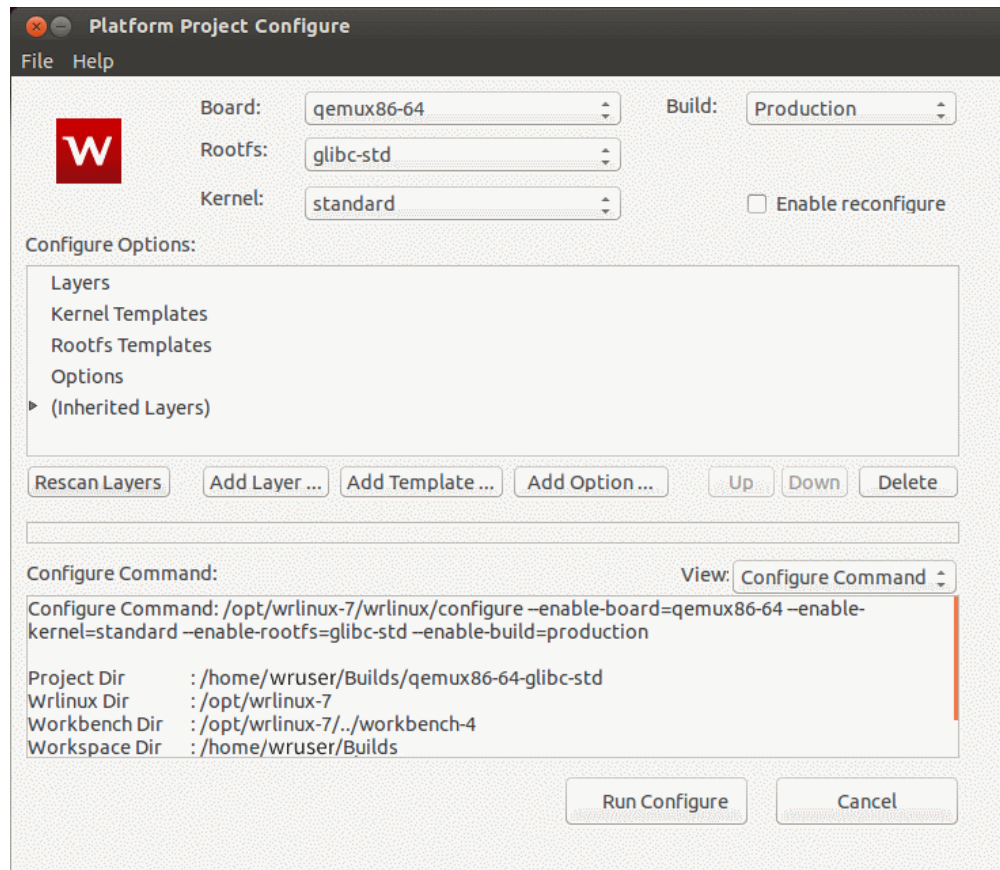
- a) Create a new project.

Right-click in the Workbench **Project Explorer** and select **New > Project**.

- b) Expand the **Wind River Linux** option, select **Wind River Linux Platform Project**, then click **Next**.

- c) Give your project a descriptive name, such as `qemux86-64_glibc-small`, and click **Finish**.

The Platform Project Configure window appears.



In this example, you will configure and build a platform project using the following settings:

**Board:** qemux86-64

**Rootfs:** glibc-small

**Kernel:** standard

- d) Accept the rest of the default settings in the Platform Project Configure dialog box. This creates a project with a standard kernel and a small file system for a qemux86-64 board.
- e) You can configure additional options to greatly improve the time it takes to build your project. This step is optional.

You can use the following configuration options to set values for parallel builds.

- **--enable-jobs=*n*** specifies the maximum number of parallel jobs (threads) that make can perform while building a single package. If this is not set, the default is 1.
- **--enable-parallel-pkgbuilds=*n*** specifies the maximum number of packages that may be built in parallel.



---

**NOTE:** BitBake automatically sets defaults for parallel jobs and packages to the number of cores on the machine.

---



**NOTE:** Workbench uses **ccache** by default, so there is no need to add a configure option for it.

f) Click **OK**.

3. Configure additional functionality.

a) In the Platform Project Configure window, click **Add Template**.

The Add Templates window appears.

b) In the RootFS/Templates section, select the checkbox for the **feature/debug** template.

This template adds **gdb** and **gdbserver** to your platform project to provide native and on-target debugging functionality.



**NOTE:** This feature is visible in the template list because it exists in one of the default layers that are always included in configurations.

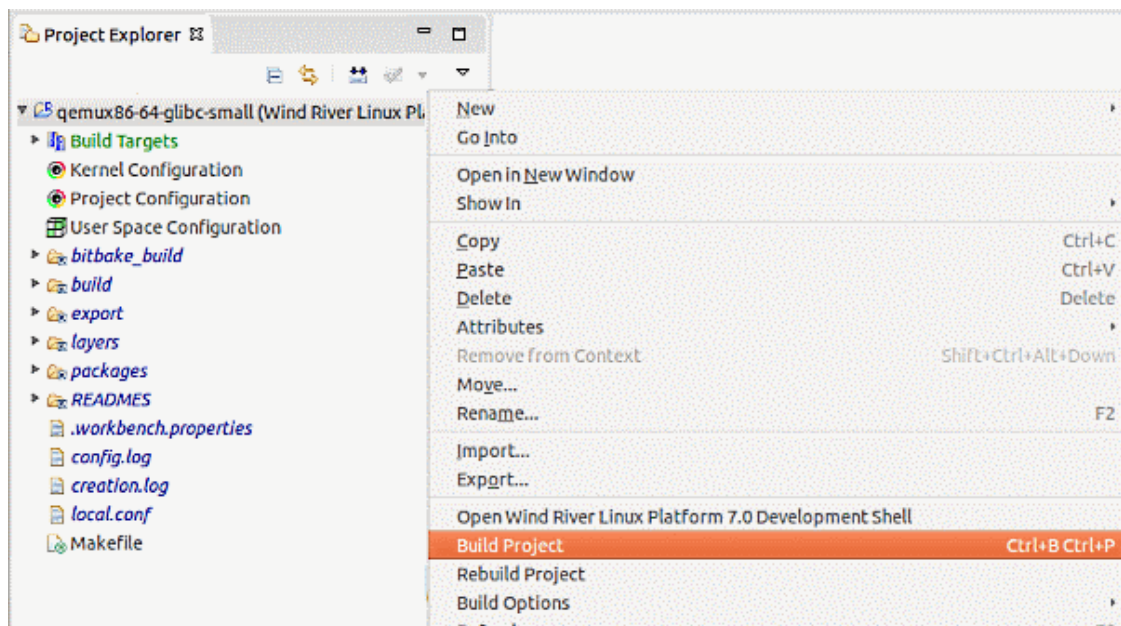
c) Click **Done** to include the **feature/debug** template in your configuration.

d) In the Platform Project Configure window, click **Run Configure**.

Project configuration will begin, and display progress in the window. Once it completes, click **Finish** to close the Platform Project Configure window.

4. Build the platform project.

In the Project Explorer view, right-click on the project and select **Build Project**.



This will take from minutes to hours, depending on your development resources. When the build is finished, you will have a kernel and file system ready for deployment to a target or an emulator.

## Postrequisites

Once you have added a new package, verify that it was added successfully as described in [Verifying that the Project Includes the New Application Package](#) on page 9.

## Adding New Application Packages to an Existing Project

You can add application packages and new features to a platform project using Workbench.

## Prerequisites

In the following example, you will add **gdb** to a previously configured and built platform project using Workbench, as detailed in the following sections:

- [Creating and Configuring a Platform Project](#) on page 1
- [Building a Platform Project Image](#) on page 4

This example assumes that you do not already have **gdb** included with your platform project.

## Procedure

1. Open a development shell in the platform project's directory.

In Project Explorer, right-click on the platform project folder and select **Open Wind River Linux Platform 8.0 Development Shell**.

2. Build and verify the **gdb** package.

- a) Build **gdb**.

From the **project** directory, enter the following command:

```
$ make gdb
```

Building the package takes a couple of minutes, during which you will see the progress on your terminal.

- b) Verify that **gdb** now exists in your package directory.

Issue the following command:

```
$ find export/RPMS/ -name '*gdb*'
```

The system returns the following output:

```
export/RPMS/x86_64/libgdbm-staticdev-1.11-r0.0.core2_64.rpm
...
export/RPMS/core2_64/libgdbm-bin-1.11-r0.1.i586.rpm
export/RPMS/core2_64/libgdbm4-1.11-r0.1.i586.rpm
export/RPMS/core2_64/python-gdbm-2.7.3-r0.3.1.i586.rpm
```

3. Add the **gdb** package to the platform project build.

From the project directory, enter the following command to add the **gdb** package:

```
$ make gdb.addpkg
```

The system will return the following output:

```
make: Entering directory `/Builds/qemux86-64_small/build'
==Checking ../layers/local/recipes-img/images/wrlinux-image-glibc-small.bbappend==
==Checking for valid package for gdb==
...
=== ADDED gdb to ../layers/local/recipes-img/images/wrlinux-image-glibc-
small.bbappend ===
```

**4. Rebuild the root file system.**

In **Project Explorer**, right-click on the project select **Build Project**.

**Postrequisites**

Once you have added a new package, verify that it was added successfully as described in [Verifying that the Project Includes the New Application Package](#) on page 9.

**Verifying that the Project Includes the New Application Package**

You should verify that application packages you added to a platform project work properly.

In this example, you are going to verify that a package was added successfully to the platform project using Workbench.

**Prerequisites**

This procedure requires that you have added the gdb package from [Configuring a New Project to Add Application Packages](#) on page 5, or [Building a Platform Project Image](#) on page 4.

**Procedure**

1. Verify that the **gdb** binary has been added to the platform project's file system.
  - a) Expand the platform project in **Project Explorer** to display the **export** folder.
  - b) Expand this folder to reveal the contents of **/dist/usr/bin**.
  - c) Locate the **gdb** and **gdbserver** files in the list.

2. Run the **gdb** binary on the target.

- a) Deploy the platform project on a QEMU target.

See [Deploying an Application Project with Workbench](#) on page 21.

- b) Run the **gdb** command on the target:

```
# gdb
```

The system returns the following and display the (**gdb**) prompt to confirm **gdb** is working:

```
GNU gdb (Wind River Linux Sourcery CodeBench 4.6-60) 7.2.50.20100908-cvs
Copyright (C) 2010 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-wrs-linux-gnu".
For bug reporting instructions, please see:
<support@windriver.com>
(gdb)
```

You may use the debugger, or type **quit** to return to the command prompt.

### Postrequisites

Having verified that the platform image includes the application, you can deploy it as described in [Deploying a Platform Project Image](#) on page 10.

## Deploying a Platform Project Image

Use the New Connection wizard to create and automatically deploy QEMU simulated targets for development and testing.

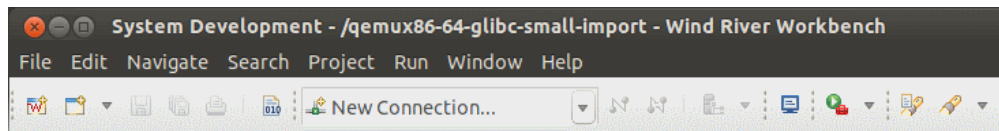
Use the System Management view or toolbar to create, connect to, or disconnect from your QEMU target. The following procedure describes how to create and launch a new QEMU target connection.

### Prerequisites

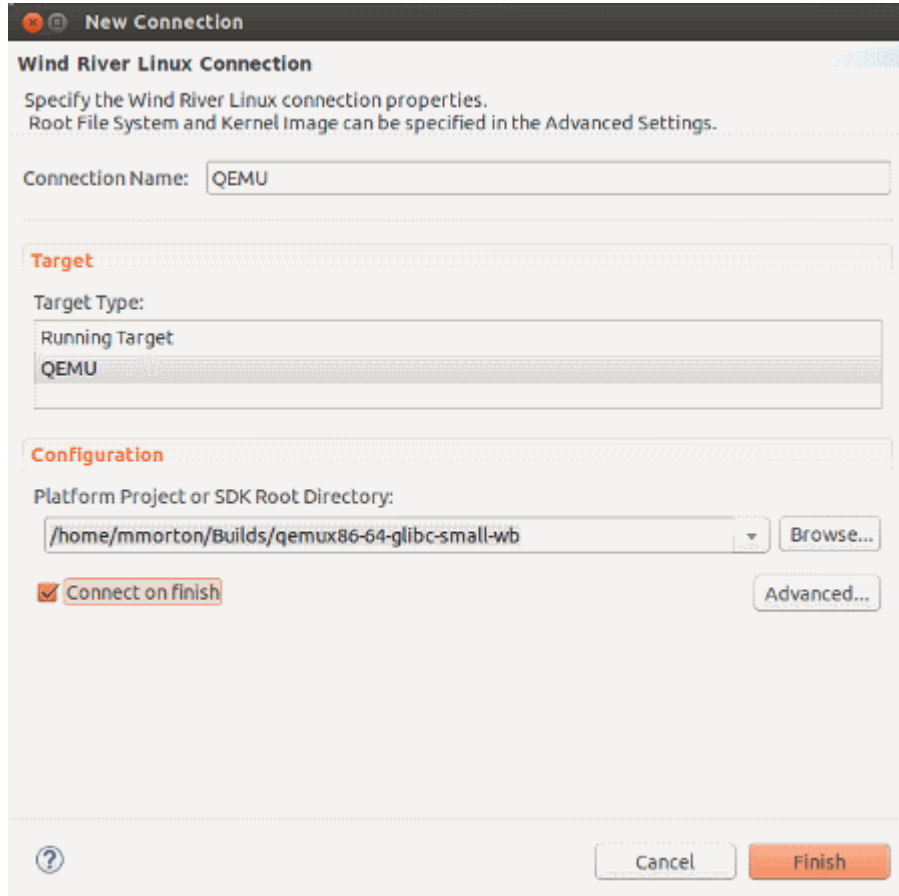
Before deploying your platform project image, ensure that any packages you added are installed correctly as described in [Deploying a Platform Project Image](#) on page 10. You will also need a connection to your target (see [Creating a Target Connection](#) on page 18).

### Procedure

1. Create a new QEMU connection in Workbench.
  - a) Click **New Connection** in the System Management toolbar to open the New Connection window.



- b) In the New Connection window, select a target option.  
Select **QEMU** in the **Target Type** section.



- c) Enter target connection information.

Click **Browse** and navigate to the platform project or SDK root directory, then click **OK** once selected.



**NOTE:** Windows-based development does not support launching QEMU targets from the SDK root directory. If you are using Workbench on Windows, you must connect to a running target.

- d) Select the **Connect on finish** checkbox.

This will automatically connect Workbench to the target once the launch completes.

2. Optionally set QEMU configuration options.

Click **Advanced** to view the advanced settings available for a QEMU target.



The image shows a screenshot of the 'Advanced Settings' dialog box in the Wind River Linux Workbench. The dialog has a title bar with a close button and the word 'Advanced'. Below the title bar, it says 'Advanced Settings' and 'Please enter the advanced settings.' The main section is titled 'QEMU' in orange. It contains several fields: 'Platform' is a dropdown menu showing 'latest installed (currently 'Wind River Linux Platform 7.0')'; 'Instance Number' is a text input field with '0'; 'Configuration' is a table with two columns, 'Name' and 'Value', listing various QEMU configuration options and their values. Below the table is a 'Description' section with a large empty text area. At the bottom, there is a section titled 'Tools Host/Target Communication Logging' in orange, with a 'Verbosity (0 = off):' dropdown menu set to '0'. At the very bottom are 'Cancel' and 'OK' buttons, along with a help icon (?) on the left.

Name	Value
TARGET_VIRT_BOOT_TYPE	usernfs
NFS_EXPORT_DIR	/home/mmorton/Builds/qemux86-6
NFS_MOUNT_DIR	export/dist
NFS_MOUNTPROG	21211
NFS_NFSPROG	11211
NFS_PORT	3149
MOUNT_PORT	3148

Use the **Instance Number** field to specify a QEMU instance manually. This allows you to launch multiple QEMU simulations in parallel, and is the same as using the **TOPTS=-n** from the command line. Note that instances are assigned automatically when a new QEMU connection is created.

In the **Configuration** section, the available options are the equivalent of running the **make config-target** command from the command line. Use these to specify many aspects of your simulation.

Use the **Tools Host/Target Communication Logging** section to specify the level of logging detail the system provides, where a setting of **0** (zero) results in no logging.

Once you are finished, click **OK** to save your settings.

3. Finish the wizard and launch the QEMU target window.
  - a) Click **Finish**.

The QEMU window will launch and complete the boot process.
  - b) Log into the QEMU target.

Enter user name and password **root** to log into the target.



```

Virtual-WRLinux0
Starting Permit User Sessions...
Starting Login Service...
[ OK ] Started System Logging Service.
[ OK ] Started /etc/rc.local Compatibility.
[ OK ] Started Permit User Sessions.
[ OK ] Started Berkeley Internet Name Domain (DNS).
WARNING: tcf-agent listening on port for commands to monitor/modify the system.
[ OK ] Started Target Communication Framework agent.
[ OK ] Reached target Host and Network Name Lookups.
Starting Getty on tty1...
[ OK ] Started Getty on tty1.
Starting Serial Getty on ttyS0...
[ OK ] Started Serial Getty on ttyS0.
[ OK ] Reached target Login Prompts.
[ OK ] Started Login Service.
[ OK ] Reached target Multi-User System.
Starting Update UTMP about System Runlevel Changes...
[ OK ] Started Update UTMP about System Runlevel Changes.

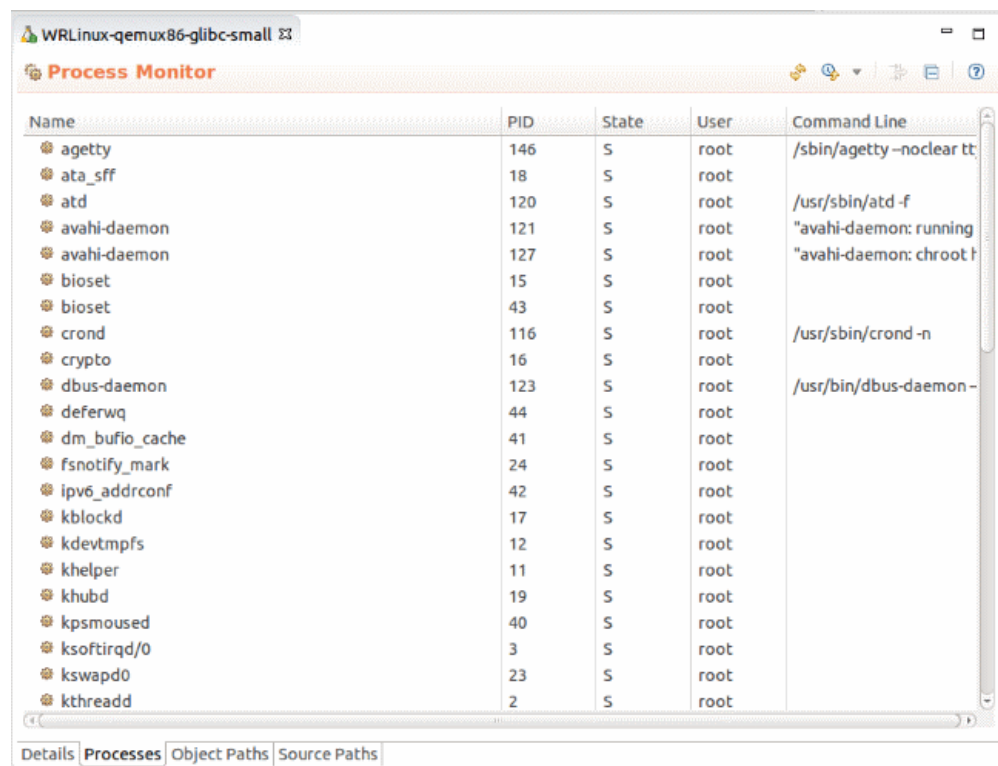
Wind River Linux 7.0.0.0 qemu0 ttyS0

qemu0 login: root
Password:
root@qemu0:~#

```

4. View the target's processes in Workbench.

In the **Connection** tab that appears in the main Editor view, click the **Processes** tab. Notice the target processes list, indicating a successful connection.



## **Platform Project Development Summary**

In this tutorial, you learned to create, configure and build a platform project. You explored the differences between adding an application package to a new and an existing project. You verified that your application package was added correctly and, finally, you deployed the project image.

# 2

## *Developing Application Projects*

Application Project Development Tutorial 15

### Application Project Development Tutorial

Learn how to develop application projects using Workbench through a series of self-paced exercises.

#### Overview

In this tutorial, you will

- learn about the relationship between the local layer and applications
- create and deploy a new application

### Creating an Application Project with Workbench

Create the Hello World application project from the samples included with Wind River Linux.

This procedure assumes you have previously configured and built a platform project using Workbench. See: [Creating and Configuring a Platform Project](#) on page 1.

#### Procedure

1. Start Workbench.

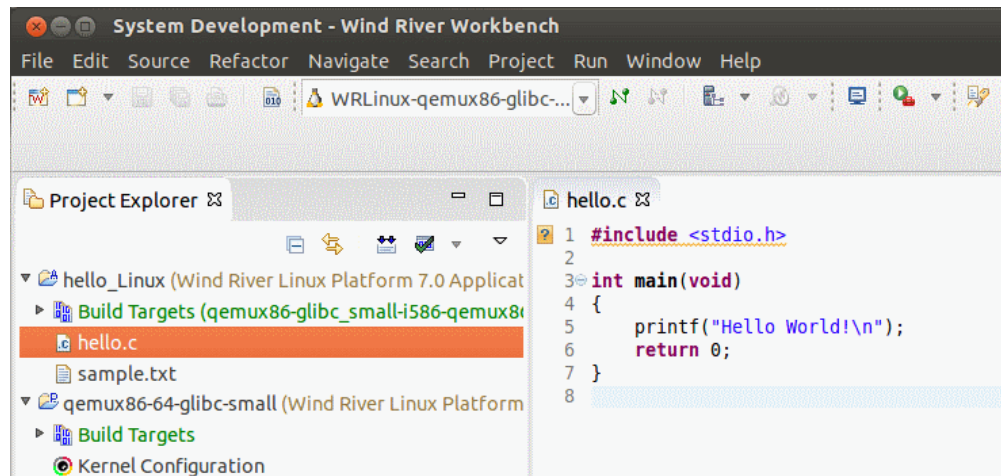
Options	Description
From the System menu	Select <b>WindRiver &gt; Workbench 4 &gt; Workbench 4</b> .
From the desktop	Double-click the Workbench icon on your desktop.

Options	Description
From the command line	<ol style="list-style-type: none"> <li>1. Navigate to the Workbench install directory:  <pre>\$ cd installDir</pre> </li> <li>2. Enter the following command:  <pre>\$ workbench-4/startWorkbench.sh</pre> </li> </ol>

2. Create a new application project using the example Hello World sample project.
  - a) Select **File > New > Example**.
  - b) Select **Wind River Linux Application Sample Project**, then click **Next**.
  - c) Select **The Hello World Demonstration Program**, then click **Finish**.

The project appears in Project Explorer as **hello\_Linux**.

To view the source file of the application, expand the **hello\_Linux** project and double-click the **hello.c** file to display it in the Editor view.



3. Build the application project.  
Right-click on the application project folder and select **Build Project**.

### Postrequisites

Once you have created the application, the next step is to build it. See: [Building an Application Project](#) on page 17.

## Building an Application Project

After you create your application project, you must build it using the same build specification you used to build your platform project.

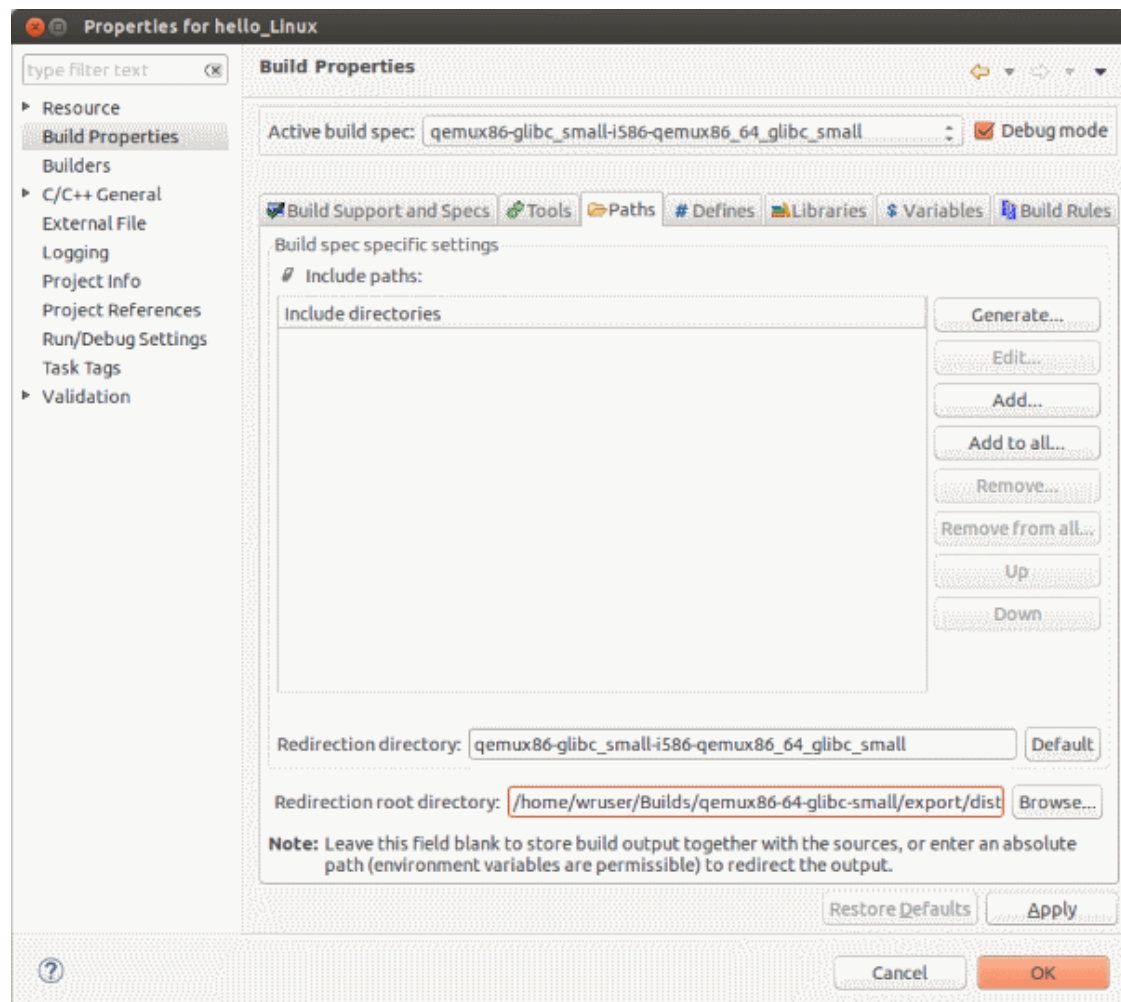
### Prerequisites

Before building an application project, you must have created one as described in [Creating an Application Project with Workbench](#) on page 15.

### Procedure

1. Right-click on the **hello\_Linux** application project and select **Properties**.
2. Set the redirection root directory.

Click **Build Properties > Paths** in the left column the tab. In the **Redirection root directory** field, enter the path to the previously created platform project's **export/dist** directory (see [Building a Platform Project Image](#) on page 4) that contains the target file system, or use the **Browse...** button to navigate there.



For example, the path might be *projectDir/qemux86-64\_small/export/dist/*. When you build the application, Workbench will copy the executable to a directory structure in the root directory of the target file system. The top-most directory will be called **hello\_world**.

3. Commit your changes.

Once you have set the path, click **OK** to close the Build Properties window. When prompted, click **Yes** to continue.

4. Right-click on the **hello\_Linux** application project, select **Build Options > Set Active Build Spec**, and then select the option for **qemux86-64-glibc-small-x86-64-qemux86-64\_small\_prj**.

Click **Yes** if prompted to rebuild the index.

5. Build the project.

Right-click on your **hello\_Linux** application project folder and select **Build Project**.

The project builds and creates a **hello\_Linux.out** executable for glibc small (32-bit) in the **Binaries** folder of the application project.

### Postrequisites

You now have an executable ready to deploy and debug, and you are ready to connect to the target [Creating a Target Connection](#) on page 18.

## Creating a Target Connection

You use the Wind River Workbench System Management view or toolbar to create or connect to a simulated or hardware target.

You can create a target connection to a running target, or to a QEMU instance. For running targets, you will require the IP address or hostname, and the TCF port to create a connection. To create and deploy a QEMU target from Workbench, see [Deploying a Platform Project Image](#) on page 10. Once you have previously created a QEMU target connection, you can connect or reconnect to it using the following procedure.


### Prerequisites

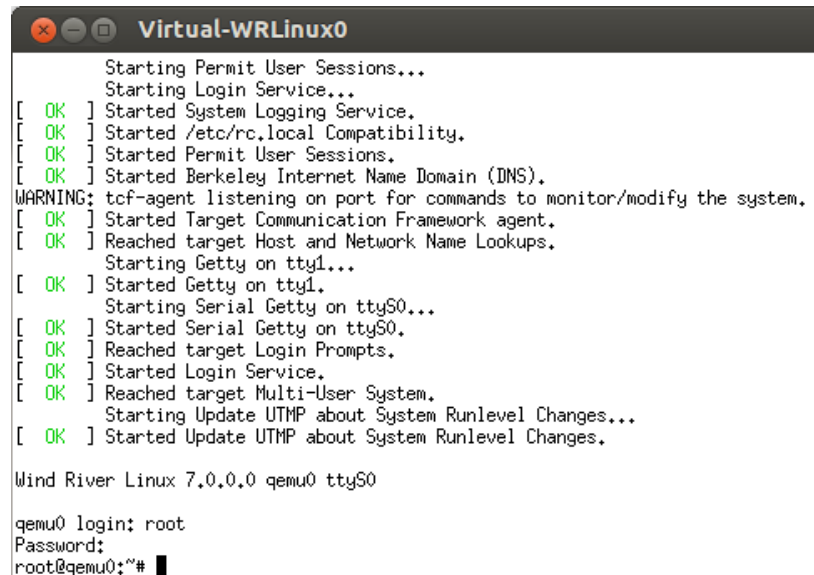
Before connecting to your target for the purpose of deploying an application, you should have created and built it as described in [Creating an Application Project with Workbench](#) on page 15 and [Building an Application Project](#) on page 17.

### Procedure

Connect to the target in Workbench.

If you want to connect to a running target, such as a QEMU instance started from the command line, you should start a QEMU connection from the command-line to use to simulate a hardware connection. If you already have a hardware target running, go to the next step.

Options	Description
<b>Existing QEMU connection</b>	<p>Complete the following steps to connect to a QEMU target. If you have previously established a QEMU target connection and it is still running as described in <a href="#">Deploying a Platform Project Image</a> on page 10, then you already have a target connection.</p> <ol style="list-style-type: none"> <li>1. Start the connection.</li> </ol> <p>In the <b>Systems Management</b> toolbar, select the connection from the drop-down list. If you created the connection in this running Workbench session, the QEMU window will launch and boot.</p> <p>If you have restarted Workbench since creating the connection, you will also need to click <b>Connect</b>  on the Systems Management toolbar.</p>



```

Virtual-WRLinux0
Starting Permit User Sessions...
Starting Login Service...
[ OK ] Started System Logging Service.
[ OK ] Started /etc/rc.local Compatibility.
[ OK ] Started Permit User Sessions.
[ OK ] Started Berkeley Internet Name Domain (DNS).
WARNING: tcf-agent listening on port for commands to monitor/modify the system.
[ OK ] Started Target Communication Framework agent.
[ OK ] Reached target Host and Network Name Lookups.
Starting Getty on tty1...
[ OK ] Started Getty on tty1.
Starting Serial Getty on ttyS0...
[ OK ] Started Serial Getty on ttyS0.
[ OK ] Reached target Login Prompts.
[ OK ] Started Login Service.
[ OK ] Reached target Multi-User System.
Starting Update UTMP about System Runlevel Changes...
[ OK ] Started Update UTMP about System Runlevel Changes.

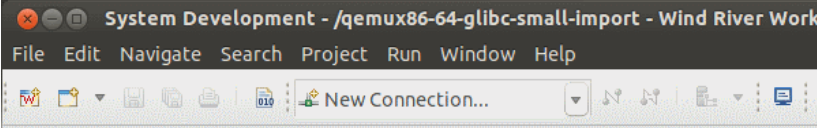
Wind River Linux 7.0.0.0 qemu0 ttyS0

qemu0 login: root
Password:
root@qemu0:~#

```

2. Enter **root** for the user name and password to log into the QEMU instance.

The target is now connected.

Options	Description
<b>Running target connection</b>	<p>Use the following steps to create a connection to a running target, such as a hardware target or a QEMU instance started from the command line with the <b>make start-target</b> command.</p> <ol style="list-style-type: none"> <li>1. Create a new target connection.</li> </ol> <p>Click <b>New Connection</b> in the <b>System Management</b> toolbar to add the QEMU target to your connection options.</p>  <ol style="list-style-type: none"> <li>2. In the New Connection window, select <b>Running Target</b>.</li> <li>3. Enter the target information.</li> </ol> <p>You will require the target hostname and TCF port to complete the connection. The following values are for the QEMU connection created in <i>Step 1</i>.</p> <p><b>Target Address</b></p> <p>Enter <code>localhost:4447</code></p> <p>This is the default setting for a Wind River QEMU BSP-based platform project image. You may substitute <b>localhost</b>, which is the hostname in the <code>/etc/hosts</code> file for the default IP address, <b>127.0.0.1</b>.</p> <p>The port number <b>4447</b> represents the default TCF port for the QEMU instance. To change this port number, use the <b>make config-target</b> command in the platform project directory. For additional information, see the <i>Wind River Linux Platform Developer's Guide: Setting QEMU Configuration Options in Workbench</i>.</p> <p><b>Platform Project or SDK Root Directory</b></p> <p>Click <b>Browse</b>, and navigate to the platform project directory.</p> <p><b>checkbox</b></p> <p>Select this to automatically connect Workbench to the target.</p> <ol style="list-style-type: none"> <li>4. Select the <b>Connect on finish</b> checkbox to start the connection automatically.</li> <li>5. Click <b>Finish</b>.</li> </ol> <p>The Wind River Linux Connection - Configuration appears in the center development view in the System Development perspective.</p> <ol style="list-style-type: none"> <li>6. Optionally click the <b>Processes</b> tab.</li> </ol> <p>Notice the target processes list, indicating a successful connection.</p>



## Postrequisites

Having created a connection to your target, you can deploy as described in [Building an Application Project](#) on page 17.

## Deploying an Application Project with Workbench

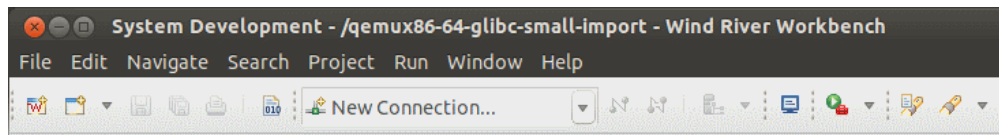
You can deploy an application to a simulated target platform with Wind River Workbench.

This example uses the **hello\_Linux** application created in [Creating an Application Project with Workbench](#) on page 15, and the built platform project image created in [Creating and Configuring a Platform Project](#) on page 1.

To deploy an application project with Workbench:

### Procedure

1. Connect Workbench to the QEMU target.
  - a) Click **New Connection** in the System Management toolbar to add the QEMU target to your connection options.



- b) In the **New Connection** window, enter target information.

#### Target Address

Enter `localhost:4447`

This is the default setting for a Wind River QEMU BSP-based platform project image. You may substitute **localhost**, which is the hostname in the `/etc/hosts` file for the default IP address, **127.0.0.1**.

The port number **4447** represents the default TCF port for the QEMU instance. To change this port number, use the **make config-target** command in the platform project directory. For additional information, see the *Wind River Linux Platform Developer's Guide: Setting QEMU Configuration Options in Workbench*.

#### Platform Project or SDK Root Directory

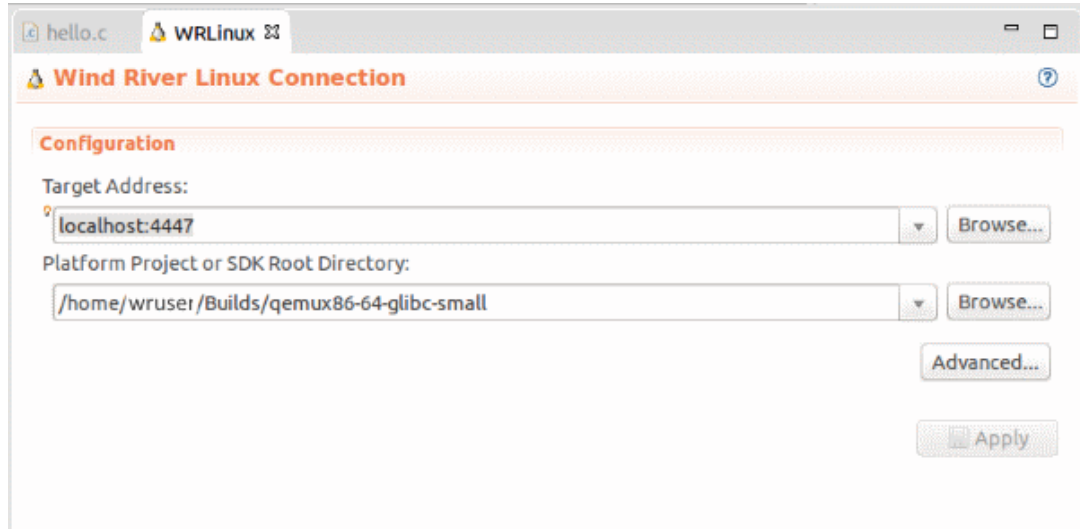
Click **Browse**, and navigate to the platform project directory.

#### Connect on finish checkbox

Select this to automatically connect Workbench to the target.

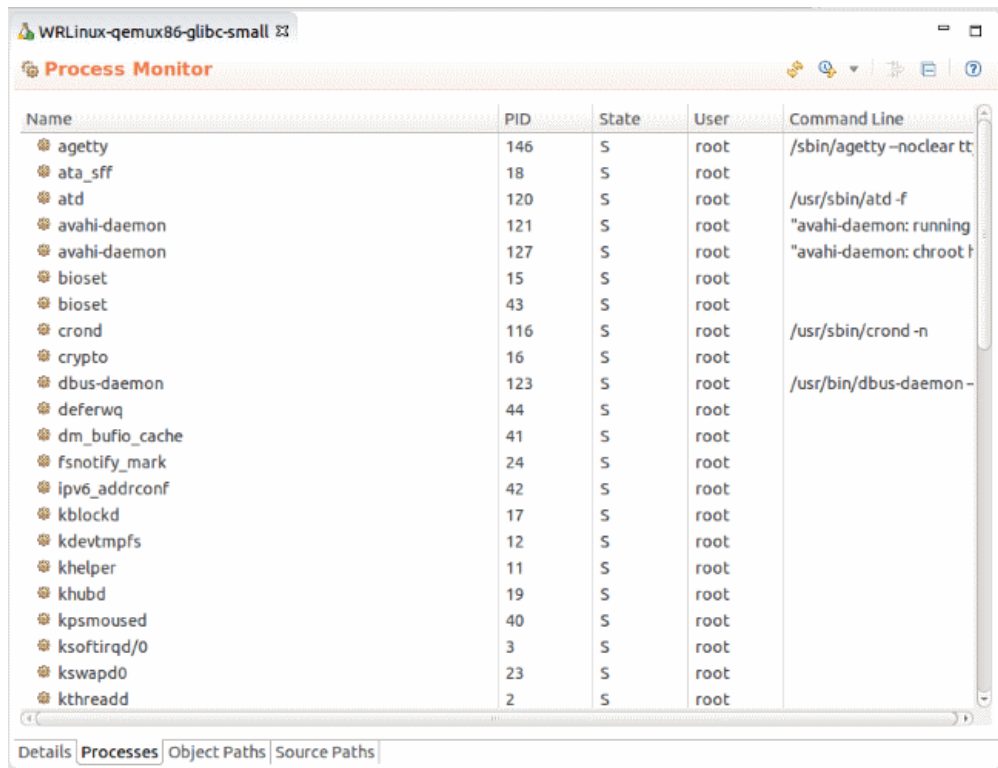
- c) Click **Finish**.

The Wind River Linux Connection - Configuration appears in the center development view of the System Development perspective.



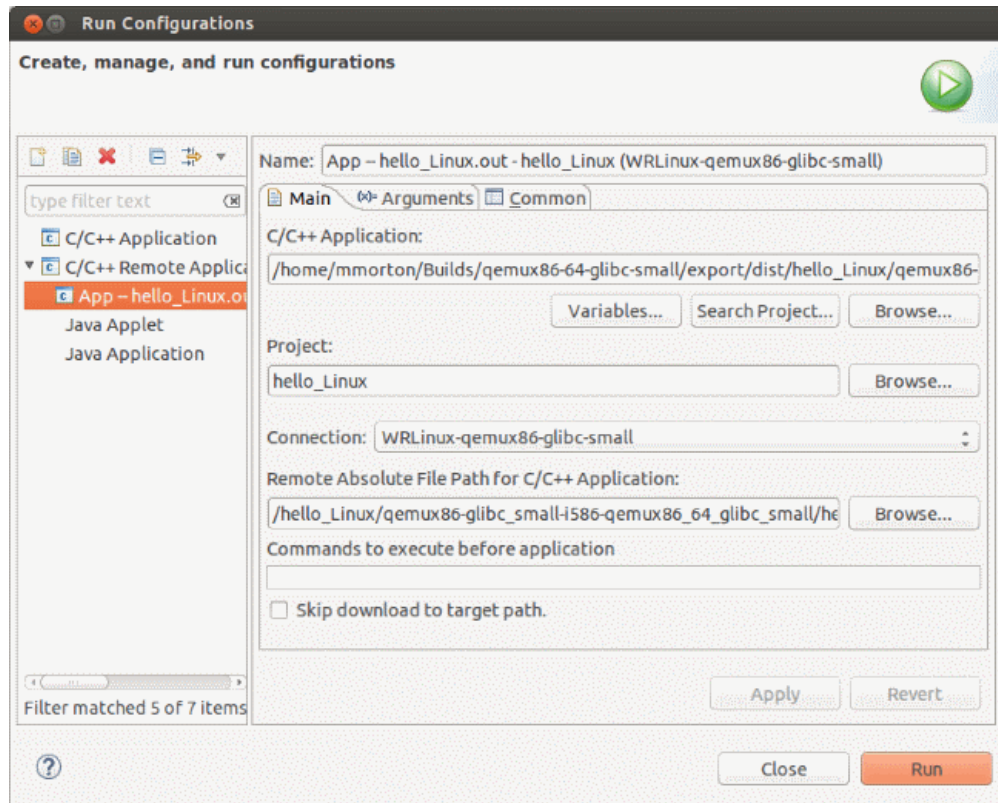
d) Click the **Processes** tab.

Notice the target processes list, indicating a successful connection.



2. Navigate to the **hello\_Linux** application in the **Project Explorer**.
3. Expand the **Build Targets** section, right-click on the **hello\_Linux.out** application, and select **Run Remote Linux Application**.

The Run Configurations window appears.



Notice that the **Project**, **Connection**, and **Remote Absolute Path for C/C++ Application** fields are populated automatically.

4. Click **Run**.

**Hello World** appears in the **Terminals** view in Workbench.



## Application Project Development Summary

In this tutorial, you learned about the local layer and how to use it for application development. You also learned the basic life cycle of a new application, from creating the application project to building, deploying and testing.



# 3

## *Debugging an Executable*

Executable Debugging Tutorial 25

### Executable Debugging Tutorial

#### Overview

The steps in this tutorial will give you hands-on experience debugging the application on your target using Workbench.

Before you perform any application debugging with Workbench, you should first verify that the application runs successfully on the target system. Doing so ensures that the application has all of the required libraries to function properly, and will ensure a successful debugging session.

#### Debugging an Executable with Workbench

You can debug an application that has been added to a platform project image.

This procedure builds upon previous procedures in these tutorials and requires:

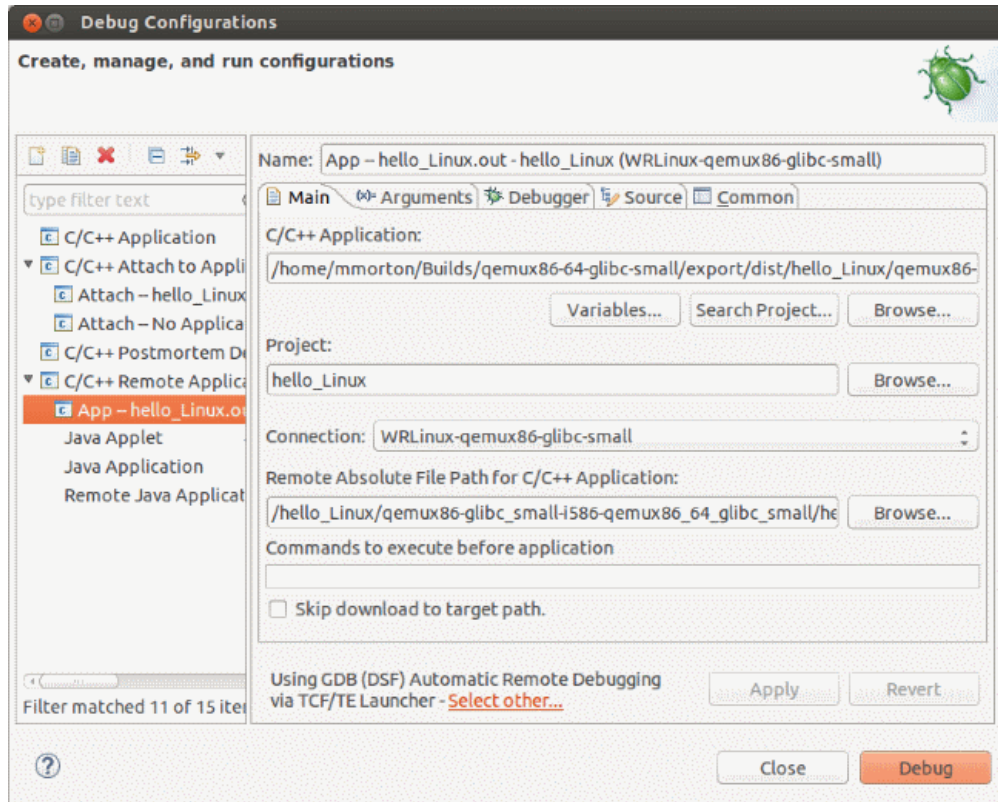
- A platform project configured and built using the **feature/debug** template as described in [Configuring a New Project to Add Application Packages](#) on page 5.
- The **Hello World** application built to match the target architecture as described in [Creating an Application Project with Workbench](#) on page 15.
- A QEMU target connected to Workbench as described in [Deploying an Application Project with Workbench](#) on page 21.

To debug an application using Workbench:

#### Procedure

1. Attach the Debugger to the **hello\_Linux** program.

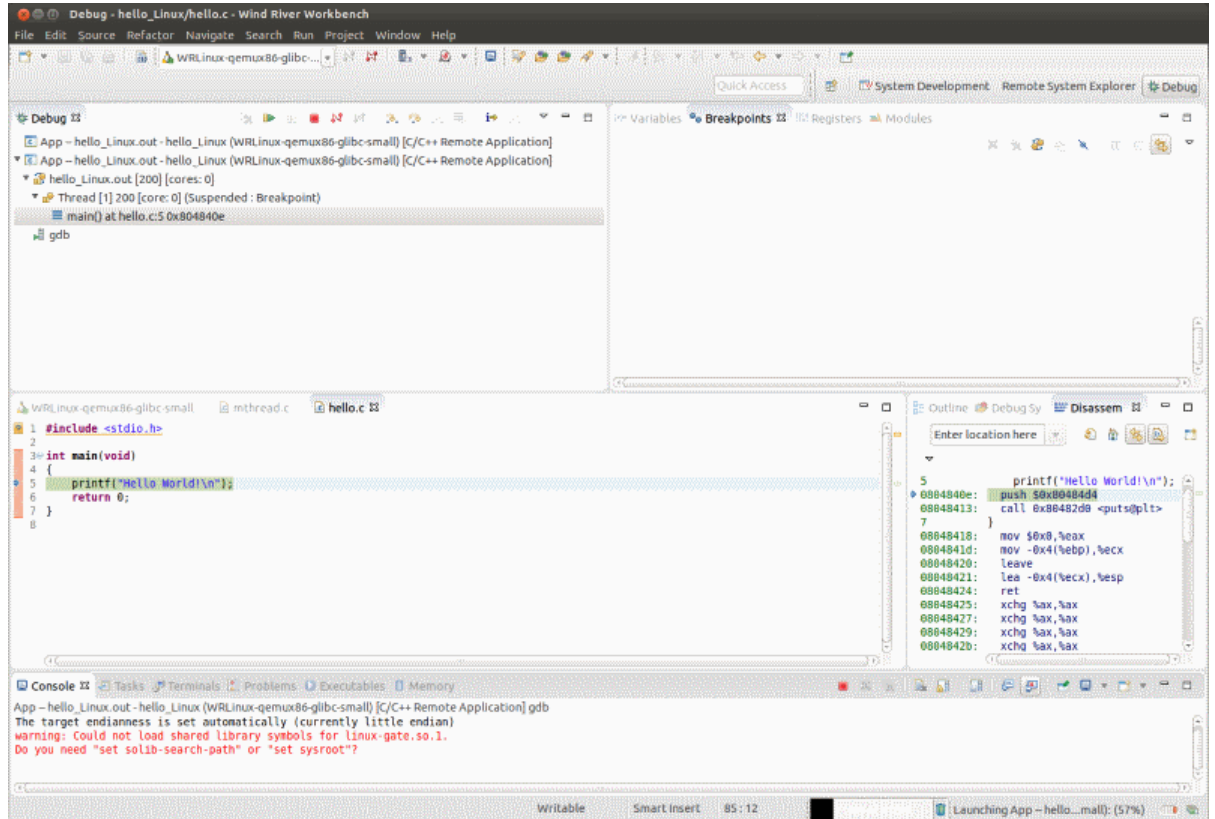
- a) Right-click on the binary and select **Debug Remote Linux Application Process**.  
The Debug Configurations window appears.



Notice that the **Project**, **Connection**, and **Remote Absolute Path for C/C++ Application** fields are populated automatically.

- b) Click **Debug**.

Workbench displays a new **Debug** perspective. This perspective displays **Debug** and **Breakpoint** views by default. The process has stopped at **main( )** in the **Debug** view.



The warning that appears concerning shared library symbols is the result of using a virtual QEMU simulator, which does not have a kernel with libraries that the debugger is looking for.



**NOTE:** To get back to the System Development perspective at any time, select **Window > Open\_Perspective > System Development**.

## 2. Resume execution.

To resume execution, click the green **Resume** icon in the Debug view. The process completes and terminates.







## *Wind River Education Resources*

Wind River Education Services offers customized on-site and self-paced courses.

If you need a more detailed learning experience than is provided by these tutorials, you can purchase instructor-led courses on a wide range of Wind River Linux subjects from Wind River Education Services.

Courses are offered at your location or over the internet by professional instructors with years of experience working with Wind River Linux, and tailored to cover exactly the topics you choose.

See <http://windriver.com/education/> for more information.

