



MIPS[®] Navigator[™] ICS Getting Started Guide

Document Number: MD00708
Revision 1.11
December 22, 2011

MIPS Technologies, Inc.
955 East Arques Avenue
Sunnyvale, CA 94085-4521

Copyright © 2012 MIPS Technologies Inc. All rights reserved.

Copyright © 2012 MIPS Technologies, Inc. All rights reserved.

Unpublished rights (if any) reserved under the copyright laws of the United States of America and other countries.

This document contains information that is proprietary to MIPS Technologies, Inc. ("MIPS Technologies"). Any copying, reproducing, modifying or use of this information (in whole or in part) that is not expressly permitted in writing by MIPS Technologies or an authorized third party is strictly prohibited. At a minimum, this information is protected under unfair competition and copyright laws. Violations thereof may result in criminal penalties and fines.

Any document provided in source format (i.e., in a modifiable form such as in FrameMaker or Microsoft Word format) is subject to use and distribution restrictions that are independent of and supplemental to any and all confidentiality restrictions. UNDER NO CIRCUMSTANCES MAY A DOCUMENT PROVIDED IN SOURCE FORMAT BE DISTRIBUTED TO A THIRD PARTY IN SOURCE FORMAT WITHOUT THE EXPRESS WRITTEN PERMISSION OF MIPS TECHNOLOGIES, INC.

MIPS Technologies reserves the right to change the information contained in this document to improve function, design or otherwise. MIPS Technologies does not assume any liability arising out of the application or use of this information, or of any error or omission in such information. Any warranties, whether express, statutory, implied or otherwise, including but not limited to the implied warranties of merchantability or fitness for a particular purpose, are excluded. Except as expressly provided in any written license agreement from MIPS Technologies or an authorized third party, the furnishing of this document does not give recipient any license to any intellectual property rights, including any patent rights, that cover the information in this document.

The information contained in this document shall not be exported, reexported, transferred, or released, directly or indirectly, in violation of the law of any country or international law, regulation, treaty, Executive Order, statute, amendments or supplements thereto. Should a conflict arise regarding the export, reexport, transfer, or release of the information contained in this document, the laws of the United States of America shall be the governing law.

The information contained in this document constitutes one or more of the following: commercial computer software, commercial computer software documentation or other commercial items. If the user of this information, or any related documentation of any kind, including related technical data or manuals, is an agency, department, or other entity of the United States government ("Government"), the use, duplication, reproduction, release, modification, disclosure, or transfer of this information, or any related documentation of any kind, is restricted in accordance with Federal Acquisition Regulation 12.212 for civilian agencies and Defense Federal Acquisition Regulation Supplement 227.7202 for military agencies. The use of this information by the Government is further restricted in accordance with the terms of the license agreement(s) and/or applicable contract terms and conditions covering this information from MIPS Technologies or an authorized third party.

MIPS, MIPS I, MIPS II, MIPS III, MIPS IV, MIPS V, MIPSr3, MIPS32, MIPS64, microMIPS32, microMIPS64, MIPS-3D, MIPS16, MIPS16e, MIPS-Based, MIPSsim, MIPSpro, MIPS Technologies logo, MIPS-VERIFIED, MIPS-VERIFIED logo, 4K, 4Kc, 4Km, 4Kp, 4KE, 4KEc, 4KEm, 4KEp, 4KS, 4KSc, 4KSd, M4K, M14K, 5K, 5Kc, 5Kf, 24K, 24Kc, 24Kf, 24KE, 24KEc, 24KEf, 34K, 34Kc, 34Kf, 74K, 74Kc, 74Kf, 1004K, 1004Kc, 1004Kf, 1074K, 1074Kc, 1074Kf, R3000, R4000, R5000, ASMACRO, Atlas, "At the core of the user experience.", BusBridge, Bus Navigator, CLAM, CorExtend, CoreFPGA, CoreLV, EC, FPGA View, FS2, FS2 FIRST SILICON SOLUTIONS logo, FS2 NAVIGATOR, HyperDebug, HyperJTAG, IASim, JALGO, Logic Navigator, Malta, MDMX, MED, MGB, microMIPS, OCI, PDtrace, the Pipeline, Pro Series, SEAD, SEAD-2, SmartMIPS, SOC-it, System Navigator, and YAMON are trademarks or registered trademarks of MIPS Technologies, Inc. in the United States and other countries.

All other trademarks referred to herein are the property of their respective owners.

MIPS Navigator ICS

Getting Started Guide

Last Updated December 22, 2011

Table of Contents

1	INTRODUCTION	8
2	WHAT'S NEW	8
2.1	NAVIGATOR ICS 2.8.2	8
2.2	NAVIGATOR ICS 2.8.0	8
3	KNOWN ISSUES	9
3.1	NAVIGATOR ICS 2.8.0	9
4	LICENSE FILES	9
4.1	SETTING UP LICENSING.....	9
4.1.1	<i>Manually Configuring Your License File.....</i>	<i>10</i>
4.2	OBTAINING A LICENSE FILE	10
4.2.1	<i>Obtaining an Evaluation License.....</i>	<i>10</i>
4.2.2	<i>Obtaining a Permanent License.....</i>	<i>10</i>
4.2.2.1	Method 1	10
4.2.2.2	Method 2.....	11
5	INSTALLING THE SOFTWARE.....	12
5.1	RUNNING THE INSTALLER	12
5.1.1	<i>Hardware & OS requirements:.....</i>	<i>12</i>
5.1.2	<i>Windows Install.....</i>	<i>12</i>
5.1.2.1	Prerequisites.....	12
5.1.2.2	Execute the Installer.....	13
5.1.3	<i>Linux Install</i>	<i>14</i>
5.1.3.1	Prerequisites.....	14
5.1.3.2	Execute the Installer.....	14

5.1.3.3	Installing MIPS Navigator Console Software	15
6	INSTALLING THE MIPS NAVIGATOR PROBE	15
7	DEBUGGING AN EXAMPLE APPLICATION.....	15
7.1	STARTING MIPS NAVIGATOR ICS.....	16
7.1.1	<i>Starting the MIPS Navigator ICS on Windows</i>	<i>16</i>
7.1.2	<i>Starting MIPS Navigator ICS on Linux.....</i>	<i>16</i>
7.2	GETTING STARTED	17
7.2.1	<i>Specifying the license file</i>	<i>17</i>
7.2.2	<i>The Welcome Screen</i>	<i>18</i>
7.2.3	<i>The Workbench.....</i>	<i>18</i>
7.3	SETTING UP THE DEMO PROJECT	19
7.3.1	<i>Creating A Project in MIPS Navigator ICS.....</i>	<i>19</i>
7.4	IMPORTING EXISTING SOURCE CODE	23
7.5	BUILDING THE PROJECT.....	27
7.6	DEBUG CONFIGURATION	28
7.6.1	<i>Create a Debug Configuration.....</i>	<i>29</i>
7.6.2	<i>Debug Configuration Settings.....</i>	<i>31</i>
7.6.3	<i>Using a Navigator Probe</i>	<i>32</i>
7.6.4	<i>Using the MIPSSim Simulator.....</i>	<i>33</i>
7.6.5	<i>Using the MIPS IASim Simulator.....</i>	<i>35</i>
7.6.5.1	<i>Advanced IASim Debug Configuration</i>	<i>37</i>
7.7	STARTING A DEBUG SESSION	38
8	TRIGGER EXPRESSIONS.....	41
8.1	EXECUTION PATTERNS	41
8.1.1	<i>Basic forms for creating execution triggers.....</i>	<i>41</i>
8.2	DATA PATTERNS.....	42

8.2.1	<i>Basic forms for creating data triggers</i>	42
8.2.1.1	Basic Data Trigger Modifiers:.....	42
8.2.1.2	Data Value Modifiers.....	43
8.3	ADDRESS MODIFIERS.....	43
8.4	RANGE EXPRESSIONS	44
8.5	EXPRESSION NOTES	45
9	CACHE VIEWER	45
10	APPENDIX A – FLEXLM LICENSING.....	46
10.1	LICENSE TROUBLE SHOOTING	46
10.1.1	<i>Environment Variables</i>	46
10.1.2	<i>Testing Your License</i>	46
10.1.2.1	Within the IDE.....	46
10.1.2.2	From the Command Line.....	48
10.1.3	<i>Troubleshooting Tips</i>	48
10.2	LICENSE DESCRIPTIONS	48
10.2.1	<i>Code Sourcery License</i>	48
10.2.2	<i>MIPS Navigator ICS License</i>	48
10.2.3	<i>MIPSim License</i>	48
11	APPENDIX B – CONNECTING MIPS NAVIGATOR PROBES	49
11.1	ESD HANDLING PRECAUTIONS.....	49
11.2	CONNECTING THE HARDWARE	49
11.2.1	<i>Navigator and Navigator Pro</i>	49
11.2.1.1	Hardware Connections.....	49
11.2.1.2	Connecting the Hardware Using USB	50
11.2.1.3	Connecting the Hardware Using Ethernet.....	51
12	REFERENCES.....	53

13 REVISION HISTORY 54

1 Introduction

This document provides a getting-started guide to using the MIPS® Navigator™ ICS (Integrated Component Suite). MIPS® Navigator™ ICS is built on the Eclipse IDE and incorporates the following major components:

- Bare-iron toolchain for MIPS cores provided by CodeSourcery, Inc.
- GNU/Linux toolchain for MIPS cores provided by CodeSourcery, Inc.
- Navigator Console Software
- MIPS Navigator Probe debug plugins for Eclipse
- MIPSsim Simulator
- IASim Simulator

Note

Prior to downloading the product installer, ensure that any anti-virus software is disabled. Many anti-virus programs attempt to scan files when reading, copying, and opening them. Some of these programs do not handle extremely large files very well and the system can appear to hang for long periods of time. It is recommended that any real-time scanning be disabled during the download and install of the product.

2 What's New

2.1 Navigator ICS 2.8.2

- New Cache Browser Views for Data and Instruction L1/L2 cache (**Window** → **Show View** → **Cache: D\$ Browser** (or **Cache: I\$ Browser**).
- New Trace View with significantly improved trace dequeuing performance.
- Fixed bug in trigger view that could result in erroneously limiting available trigger resources when the “Toggle hardware breakpoint” feature was used.

2.2 Navigator ICS 2.8.0

- Updated compiler toolchain based on CodeSourcery version 2011-03.107 based on GCC 4.5.2 and GDB 7.2.

- Updated Eclipse IDE based on Eclipse 3.6 (Helios) and CDT 7. Follow [this link](#) to see a summary of new changes in CDT 7.
- Updated Navigator Console release 4.4.2.
- New Expression based Pattern creator for the Trigger View. This is an early release. Feedback is appreciated, and changes may be made in future releases to the expression syntax.
- Updated Hot Spot Analyzer.
- New Symbolizer symbol browsing dialog. This dialog is accessible from several places within Navigator ICS to provide easy lookup of addresses from symbol name.
- Many bug fixes and improvements to address debugger stability.

3 Known Issues

3.1 Navigator ICS 2.8.0

- When debugging the “run to line” feature does not work if the target in is not inside the current function scope.
- When an attempt is made to source-line step over a line of code that is an infinite loop, the debugger will not return control to the user when the halt button is pressed. The debug session must be terminated to recover. This will be fixed in the next maintenance release.
- The compiler does not produce pure micromips code. Startup code and some libraries are still implemented using MIPS32 code. This will be fixed in the next maintenance release.

4 License Files

MIPS Navigator ICS uses the FlexLM licensing engine. Different components of MIPS Navigator ICS are licensed in different ways. The MIPS Product Key that was delivered with your order is used to fetch a license from the MIPS Website.

The information and processes in this section can be read and completed prior to installing MIPS Navigator ICS.

4.1 Setting up Licensing

It is highly recommended that you obtain your license file prior to installing MIPS Navigator ICS. The MIPS Navigator ICS installer will ask you for your license file and install it so that no further license configuration is needed. This is the simplest way to setup product licensing.

You can, optionally, skip the installation of the license file during product installation. If you do this, then MIPS Navigator ICS will prompt you for your license when started. If you intend on using the MIPS

Navigator Console outside of the MIPS Navigator ICS IDE, further manual license configuration will be also required.

When you install the license file during the installation process copies of the license file are placed in predefined locations within the installation directories. The Navigator ICS application launcher script then defines (or modifies if it is already defined) the MIPS_LICENSE_FILE to point to these pre-defined directories.

If you open the licensing preference page in Navigator ICS you will see the current value of the MIPS_LICENSE_FILE variable. This value will reflect changes made by the application launcher.

4.1.1 **Manually Configuring Your License File**

Manually configuring your license file requires defining the MIPS_LICENSE_FILE and CSLMRD_LICENSE_FILE environment variables and pointing them to your license files. Since the Navigator ICS license contains license information for both MIPS and CodeSourcery, both environment variables should point to the same license files.

If you choose to configure your license file after the installation, follow these steps:

1. Create a directory someplace on your system and place your MIPS license files in this directory.
2. Modify your environment to create a MIPS_LICENSE_FILE variable that points to this directory.
3. Modify your environment to create a CSLMRD_LICENSE_FILE variable that points to this directory.

4.2 **Obtaining a License File**

4.2.1 **Obtaining an Evaluation License**

If you are evaluating MIPS Navigator ICS, then you can obtain a 30 day evaluation license from the MIPS Website. The evaluation license is node-locked to a single PC. You do not need a Product Key to obtain an evaluation license.

4.2.2 **Obtaining a Permanent License**

To obtain a permanent node-locked license you must go to the licensing webpage for MIPS Navigator ICS and fill in the online form. You will need a valid Product Key and the Ethernet NIC address (not the IP Address) of the computer being licensed. A permanent license file will be emailed to you.

You will need to obtain the host ID for the host system on which the product is installed. There are two methods available:

4.2.2.1 **Method 1**

The `tools/flexlm` subdirectory of the product installation directory contains a tool called `lmhostid.exe` (`lmhostid` on Linux). Executing this utility in a command shell will display all valid hostids for the system. If multiple hostids are displayed, select only one. It does not matter

which one is selected as long as it is a permanent hostid (i.e. try not to choose an Ethernet id associated with a VPN virtual NIC). Below is an example of the output from the `lmhostid` command:

```
cmd>lmhostid
lmhostid - Copyright (c) 1989-2006 Macrovision Europe Ltd. and/or
The FLEXnet host ID of this machine is "0019b9381a82"
```

4.2.2.2 Method 2

On Windows PC's you can get the Ethernet address by opening a command prompt window and typing "`ipconfig /all`". The output will be similar to (with the Ethernet address in bold):

```
Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix . : mydomain.com
    Description . . . . . : Broadcom NetXtreme 57xx Gigabit Controller
    Physical Address. . . . . : 00-00-65-89-AB-F2
    Dhcp Enabled. . . . . : Yes
    Autoconfiguration Enabled . . . . : Yes
    IP Address. . . . . : 192.168.0.123
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.0.1
    DHCP Server . . . . . : 192.168.0.1
    DNS Servers . . . . . : 192.168.0.2
    Primary WINS Server . . . . . : 192.168.0.3
    Secondary WINS Server . . . . . : 192.168.0.4
```

On Linux PC's you can get the Ethernet address by opening a command prompt window and typing "`ifconfig`". The output will be similar to (with the Ethernet address in bold):

```
eth0    Link encap:Ethernet  HWaddr 00:13:72:34:B8:AA
        inet addr:192.168.0.158  Bcast:192.168.0.255  Mask:255.255.255.128
        inet6 addr: fe80::213:72ff:fe34:b8aa/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:52 errors:0 dropped:0 overruns:0 frame:0
        TX packets:41 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:6988 (6.8 KiB)  TX bytes:4099 (4.0 KiB)
        Interrupt:177

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:16436  Metric:1
        RX packets:1295 errors:0 dropped:0 overruns:0 frame:0
        TX packets:1295 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:2145250 (2.0 MiB)  TX bytes:2145250 (2.0 MiB)
```

5 Installing the Software

This section details the installation of the MIPS Navigator ICS Software on Windows and Linux host machines.

Note

If you are using a USB MIPS Navigator probe, you should install MIPS Navigator ICS prior to connecting your probe to your host PC.

If you have any MIPS Navigator Probes connected to your computer's USB, make sure to disconnect them from USB before installing the software.

5.1 Running the Installer

5.1.1 Hardware & OS requirements:

- Pentium III+ class PC with minimum 1024 Mbytes of memory (more is highly recommended and will significantly improve performance)
- Operating Systems:
 - Windows XP Service Pack 3 (32 and 64 bit)
 - Windows 7 (32 and 64 bit)
 - Red Hat Enterprise Linux 4 32bit
 - Red Hat Enterprise Linux 5 (32 and 64 bit)

5.1.2 Windows Install

5.1.2.1 Prerequisites

MIPS Navigator ICS for Windows requires the following software be previously installed on the host Windows system:

- A PDF Document Reader
- Windows .NET Framework 3.5

Prior to installing Navigator ICS:

- Ensure that previous versions of Navigator ICS, Navigator IDE, System Navigator, and Navigator Console are uninstalled.
- Ensure that any anti-virus software is disabled. Many anti-virus programs attempt to scan files when reading, copying, and opening them. Some of these programs do not handle extremely large files very well and the system can appear to hang for long periods of time. It is

recommended that any real-time scanning be disabled during the download and install of the product.

5.1.2.2 *Execute the Installer*

The Windows installer is named **mips-navigator-ics-<version>.exe**. Follow these steps to start the installer:

Driver Installation on Windows XP

If anytime during installation a Windows dialog box appears stating that the software being installed has not passed Windows Logo testing, click the “Continue Anyway” button. It is safe and necessary to do so if you wish to use Navigator Console with a USB probe.

1. Execute this program to launch the installer. The installer will present several screens with questions and options.
2. **Introduction:** The first screen is an introduction and recommends stopping all other application before continuing.
3. **License Agreement:** You must accept the terms of the license agreement before continuing.
4. **Choose Install Set:** You must select an install set to install. You can choose among Complete, MIPS SDE ELF, MIPS GNU Linux, and Navigator Console. Choose the install set that best matches your development needs. MIPS SDE ELF is the default install set.
5. **Choose Install Folder:** You must install the software to a location within your home directory or to a directory where you have complete read/write/execute access.
6. **Choose Shortcut Folder:** You can select a folder where you would like shortcuts to product documentation to be. You can also prevent these shortcuts from being created.
7. **Install License File:** You can optionally (recommended) install your license file during the product installation process. Use the [Choose...] button to locate your license file so that the installer can install it.
8. **Pre-installation Summary:** This screen presents a summary of your choices prior to starting the installation process as well as a summary of disk space requirements. If everything looks good, press the Install button.
9. **Installing...:** This screen shows the progress of the installation. Depending on the speed of the system, installation can take several minutes to complete.
10. **SysNav Installation Instructions:** Only applies to the Linux Installer. This screen is skipped on Windows.
11. **Install Complete:** This screen announces the completion of this Navigator ICS installation.

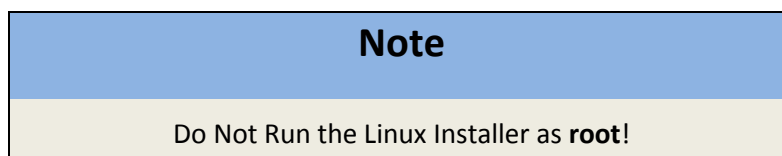
5.1.3 Linux Install

5.1.3.1 Prerequisites

MIPS Navigator ICS for Linux requires the following software be previously installed on the host Windows system:

- A PDF Document Reader
- PERL (Version 5 or later)

5.1.3.2 Execute the Installer



The Linux installer is named **mips-navigator-ics-<version>.bin**. Follow these steps to start the installer:

1. First make sure that the installer file has execute permissions. This can be done using the `chmod` command: `chmod 755 <installer-file>`
2. Execute this program to launch the installer. The installer will present several screens with questions and options.
3. **Introduction:** The first screen is an introduction and recommends stopping all other application before continuing.
4. **License Agreement:** You must accept the terms of the license agreement before continuing.
5. **Choose Install Set:** You must select an install set to install. You can choose among Complete, MIPS SDE ELF, MIPS GNU Linux, and Navigator Console. Choose the install set that best matches your development needs. MIPS SDE ELF is the default install set.
6. **Choose Install Folder:** You must install the software to a location within your home directory or to a directory where you have complete read/write/execute access.
7. **Choose Shortcut Folder:** You can select a folder where you would like shortcuts to product documentation to be. You can also prevent these shortcuts from being created.
8. **Install License File:** You can optionally (recommended) install your license file during the product installation process. Use the [Choose...] button to locate your license file so that the installer can install it.
9. **Pre-installation Summary:** This screen presents a summary of your choices prior to starting the installation process as well as a summary of disk space requirements. If everything looks good, press the Install button.
10. **Installing...:** This screen shows the progress of the installation. Depending on the speed of the system, installation can take several minutes to complete.

11. **SysNav Installation Instructions:** The MIPS Navigator ICS Installer unpacks, but does not install the device driver components of the Navigator Probe software. Unlike the IDE installer, the Navigator Probe Software Installer must be run as root. See the next section.
12. **Install Complete:** This screen announces the completion of this Navigator ICS installation. Proceed to the next section to complete the installation of the Navigator Console software for MIPS Navigator Probes.

5.1.3.3 *Installing MIPS Navigator Console Software*

Most of the Navigator Console software is installed with the IDE. Some system level device driver files cannot be installed with the IDE installer because root privileges are required to install these components. A single installation script is provided that will install these components. This script must be run manually from a root shell. Follow these steps to install the driver components:

1. Open a shell console and **su** to a root shell
2. Change to Directory: `cd <install-dir>/NavigatorConsole/install`
3. Execute the `./sysnav_install.pl` script (this requires PERL)

Upon completion a message is printed to the console indicating the successful installation of the software. **Be sure to close the root shell before starting Navigator ICS.**

6 Installing the MIPS Navigator Probe

See **Appendix B – Connecting MIPS Navigator Probes** for instructions on connecting the MIPS Navigator probe to the host PC and target systems.

7 Debugging an Example Application

This section of the Getting Started Guide will cover:

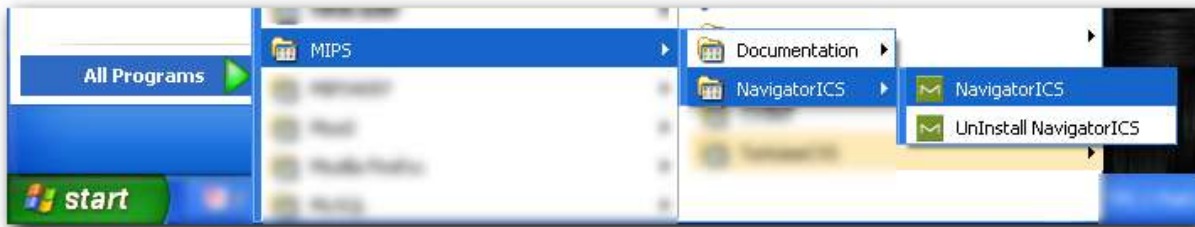
- Starting the MIPS Navigator ICS for the first time.
- A brief introduction to the MIPS Navigator ICS.
- Creating a project using existing/example source code.
- Setting up a debug configuration.
- Launching the debugger.
- Interacting with the target through the MIPS Navigator ICS.

7.1 Starting MIPS Navigator ICS

The MIPS Navigator ICS is built around the Eclipse platform (www.eclipse.org). If you are familiar with the Eclipse platform, then the MIPS Navigator ICS will be very familiar. If this is your first time working on an Eclipse based IDE, then it is strongly suggested that you visit the Eclipse home on the internet and read some of the articles and help files that describe the environment and the concepts that lay underneath. You can view the official [Eclipse Help System](#) online. Take a look at the *Workbench User Guide* for an excellent and thorough introduction to the Eclipse based IDE. Keep in mind that the *Workbench User Guide* does not describe the world of embedded development and, thus, should be used only to become familiar with the Eclipse environment, not with embedded C development.

7.1.1 Starting the MIPS Navigator ICS on Windows

The product installer creates a start-menu group called *MIPS* (this name can be changed during product installation, if you specified a different group, look for the name you specified). Within this group is a folder called **Navigator ICS** (see the screen snapshot below). Select the **Navigator ICS** icon to launch the IDE.



It is important to use the start menu (or a shortcut to the start menu) when launching the IDE because this launch shortcut sets up important details with regard to the development environment workspace.

For a more thorough understanding of the launch sequence examine the properties of the start menu item.

7.1.2 Starting MIPS Navigator ICS on Linux

The Linux installer places a shortcut icon on your desktop. Note: We have seen some flavors of Linux that may not update the desktop without first logging out and logging back in. Double-clicking on the shortcut (or executing it from a command line) will launch MIPS Navigator ICS. Always use this shortcut to launch MIPS Navigator ICS. The shortcut sets up an environment required to properly run MIPS Navigator ICS.

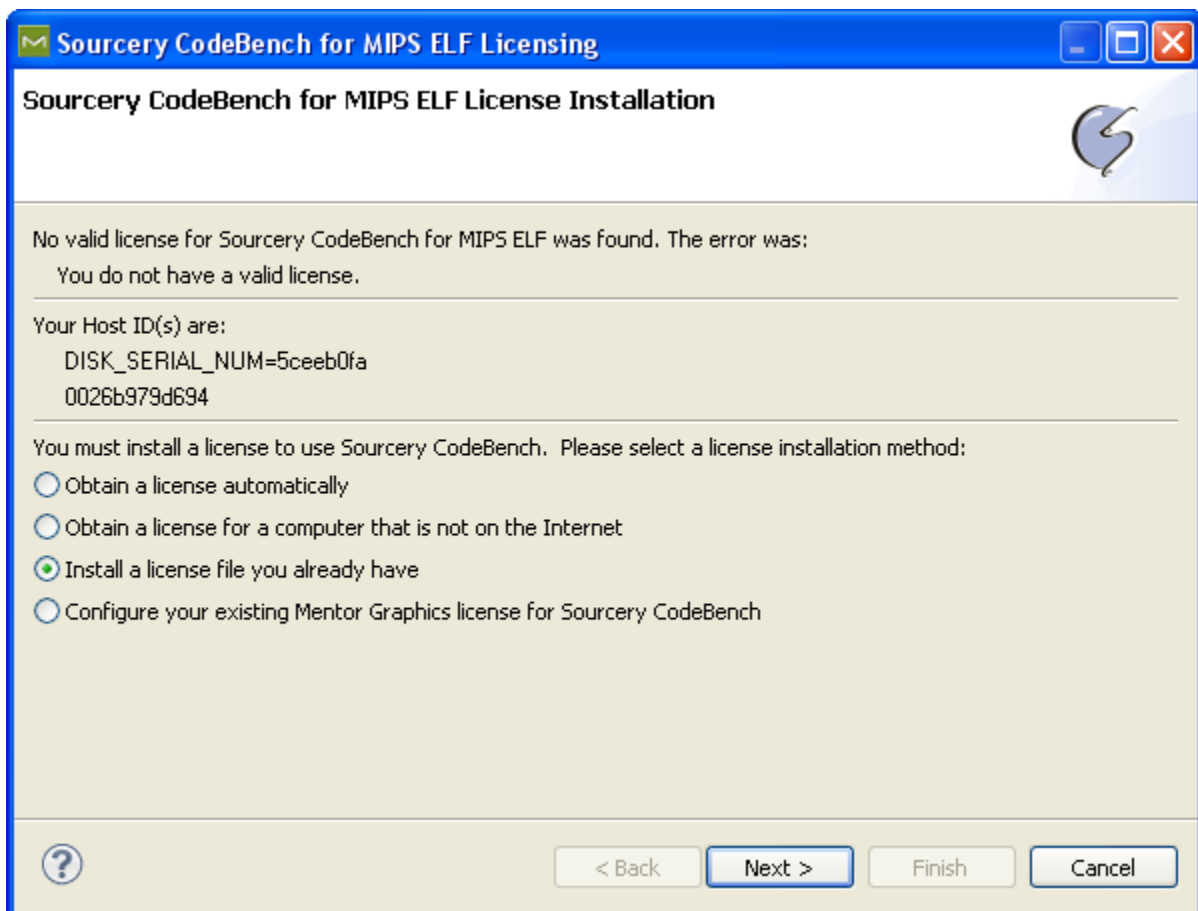
7.2 Getting Started

7.2.1 Specifying the license file

If you did not specify your license file during product installation then when the IDE is first started you will be prompted to locate your license file. You will see a dialog box (shown below). At this point you should have a valid license file accessible on the host machine. Select the radio button option titled “Install a license file you already have”. Click **Next**. Click **Browse**. Browse to and select the license file. Click **Open**. Click **Finish**.

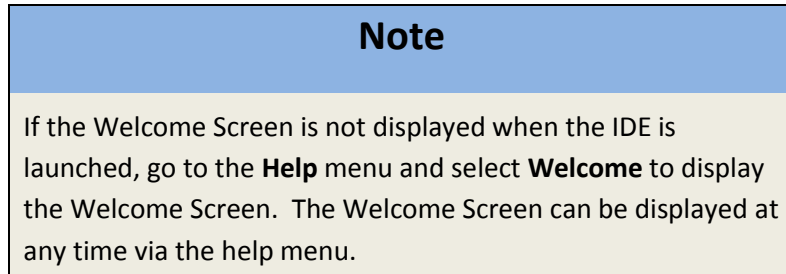
Note

If you have selected to install both bare-iron and Linux toolchains, then you will be asked to specify two license files during this stage. Repeat the steps for the first license file twice.



7.2.2 The Welcome Screen

The first time the IDE is started you will be presented with a Welcome screen. Feel free to explore the options presented in this display especially if you are new to Eclipse based environments.



The descriptions and screenshots in this Getting Started Guide assume that you are running the MIPS Navigator ICS for the first time (or at least assuming that the UI is in its initial installed state). The IDE allows the developer to configure window placement in an almost infinite variety. If you are working through this Getting Started Guide on a previously installed system, then some of descriptions may not match what you see on your screen.

7.2.3 The Workbench

Click on the arrow icon on the upper right corner of the Welcome Screen. This is the “Go to the workbench” icon.

The workbench is the main working area of the IDE. When you first open the workbench you will see a display that looks something like Figure 1. This is an empty workbench shown in the **C/C++ Perspective** (as indicated by the small depressed button near the upper right hand corner of the window, labeled C/C++).

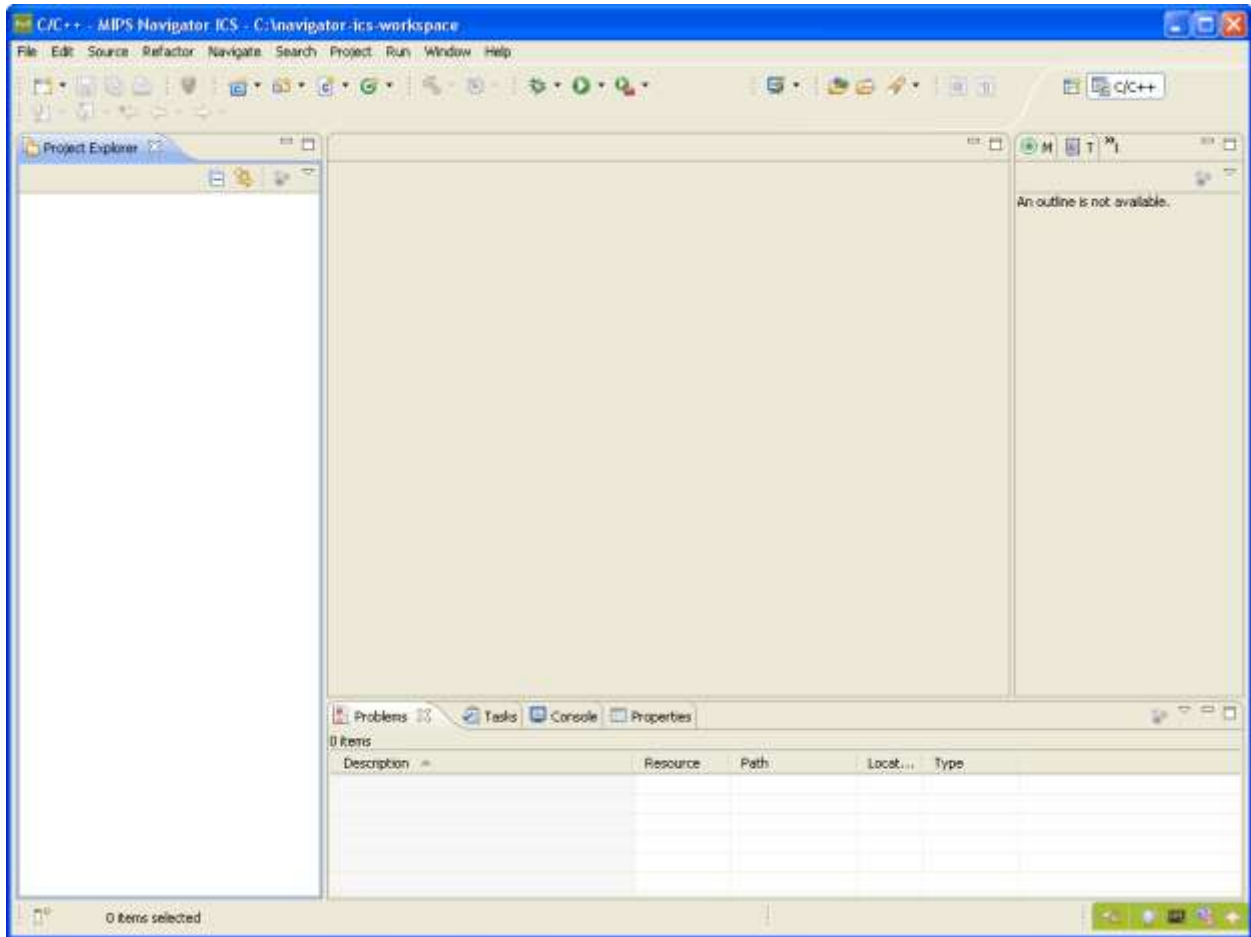


Figure 1 - An Empty Workbench

7.3 Setting Up the Demo Project

The Eclipse IDE uses the term Project to describe a set of files used to build a particular software component. A project is stored on the disk as a folder (or directory) that contains, for example, C/C++ program files.

Like everything else in Eclipse, Projects come in many “flavors”. Eclipse uses the project type to simplify the building and maintenance of the project contents. It also keeps the project source (CVS, SVN, etc.) associated with the project.

7.3.1 Creating A Project in MIPS Navigator ICS

Now we are going to create a project that “wraps” some existing source code. In this walk-through we will use some example source code that is delivered with the MIPS Navigator ICS product.

On the main application menu, select the **File | New | C Project** menu item (Figure 2).

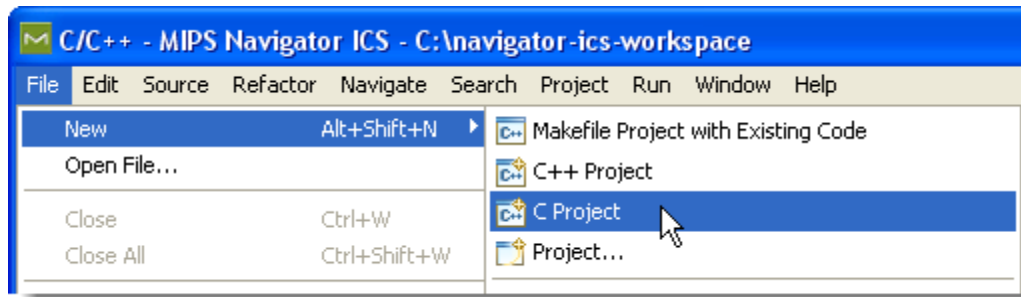


Figure 2 – Create a New C Project

This will open the **New C Project** dialog (Figure 3). Follow these steps:

- Give the project a name. Enter a name in the **Project name:** field. For this example, type “mipsdemo”.
- In the **Project Type** list, select **Executable/Empty Project**.
- In the **Toolchain:** list, select **Sourcery CodeBench for MIPS ELF**.
- At this point, the C Project dialog should look like Figure 3.

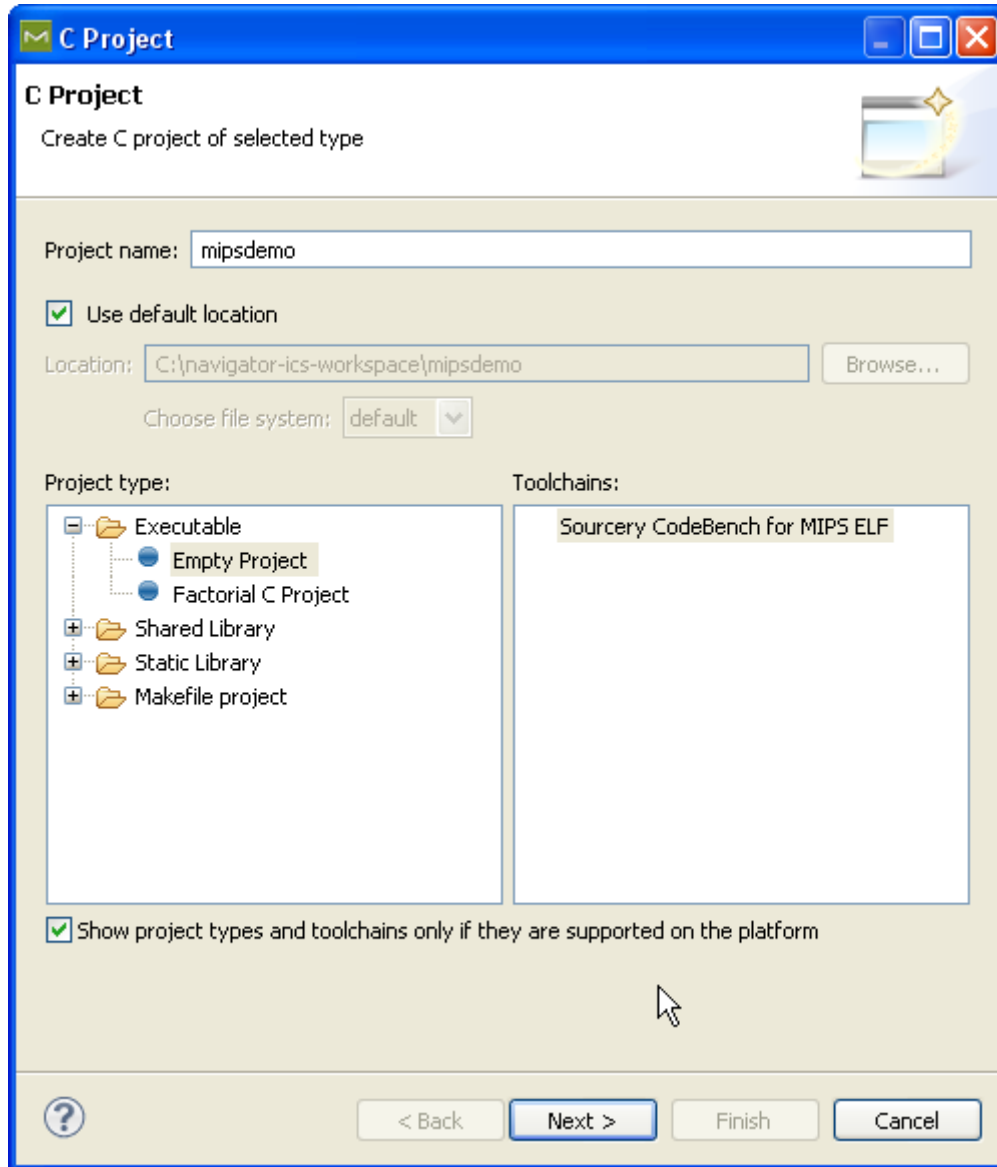


Figure 3 – New C Project Dialog

- Click the **Next** Button. This will present the **C Project** dialog. This dialog is used to set up several components of the project. Select the settings shown in Figure 5 (for the purpose of this example we are using a MALTA board with a 24Kc processor core):

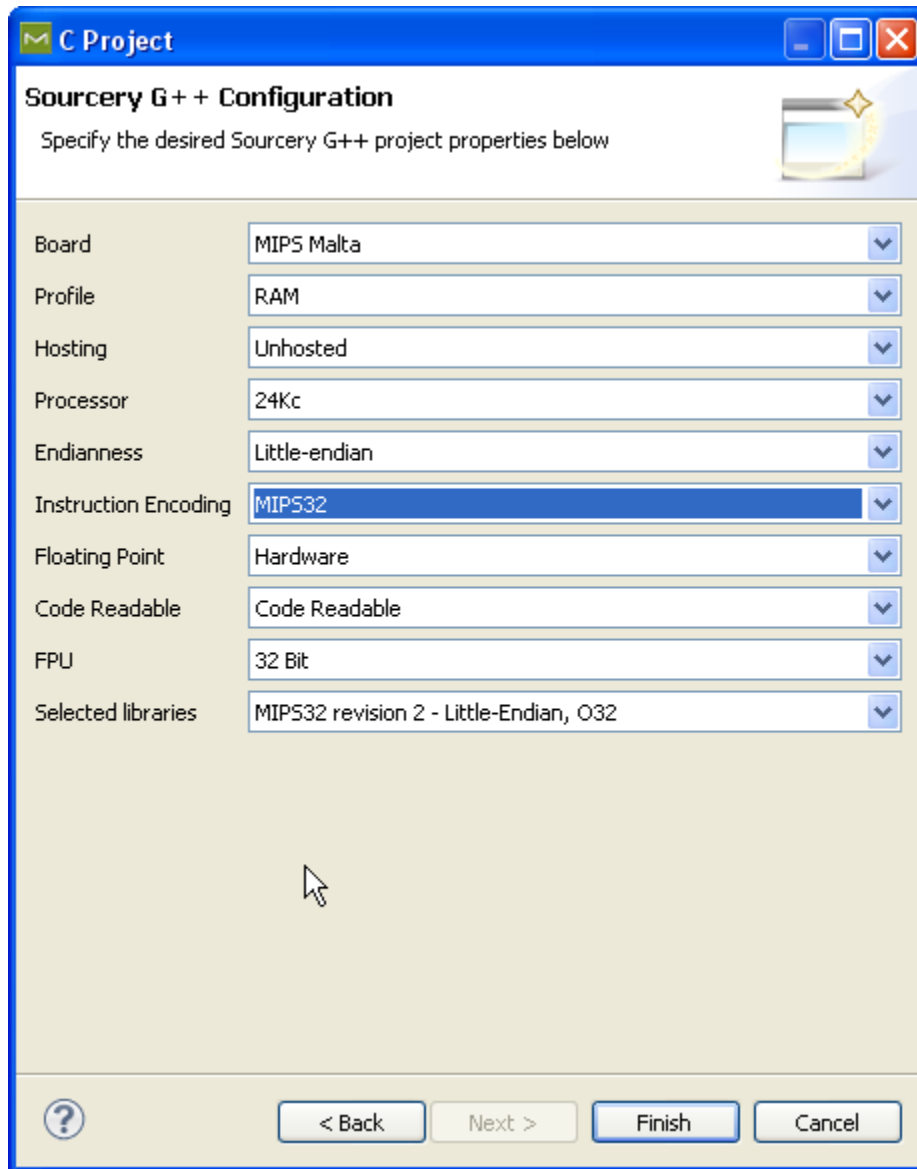


Figure 4 – C Project Dialog

- Click the **Finish** button.

Note

This example assumes a 24Kc target core is being used. If you are using a different core, you will need to adjust the Processor setting to match your core.

Note

Note on Endianess: The endianess must be consistent across the system. Check:

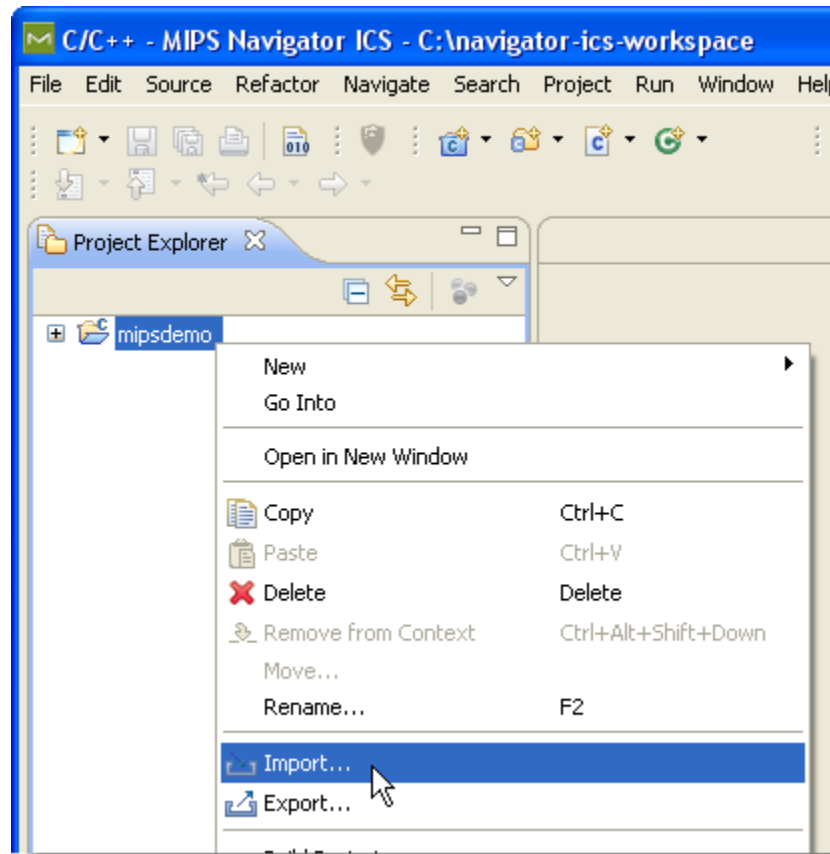
1. The project tool settings to configure whether the project is being compiled for big or little endian
2. The Debug configuration to configure the debugger endianess
3. The target hardware. MALTA boards can be switched between big and little endian.

- You have just created a project.

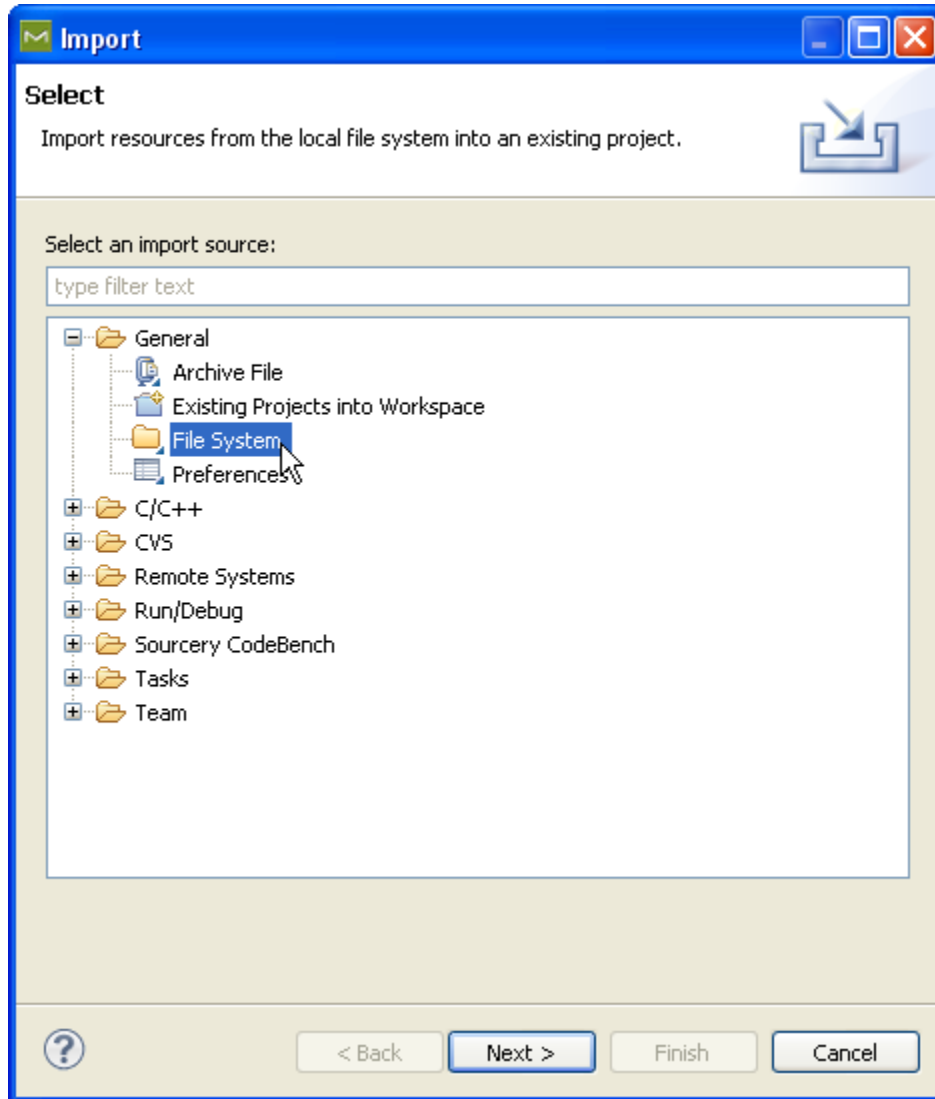
7.4 Importing Existing Source Code

The project you just created has no source code. In this guide we are going to import some example source code that is provided with the installation of MIPS Navigator ICS

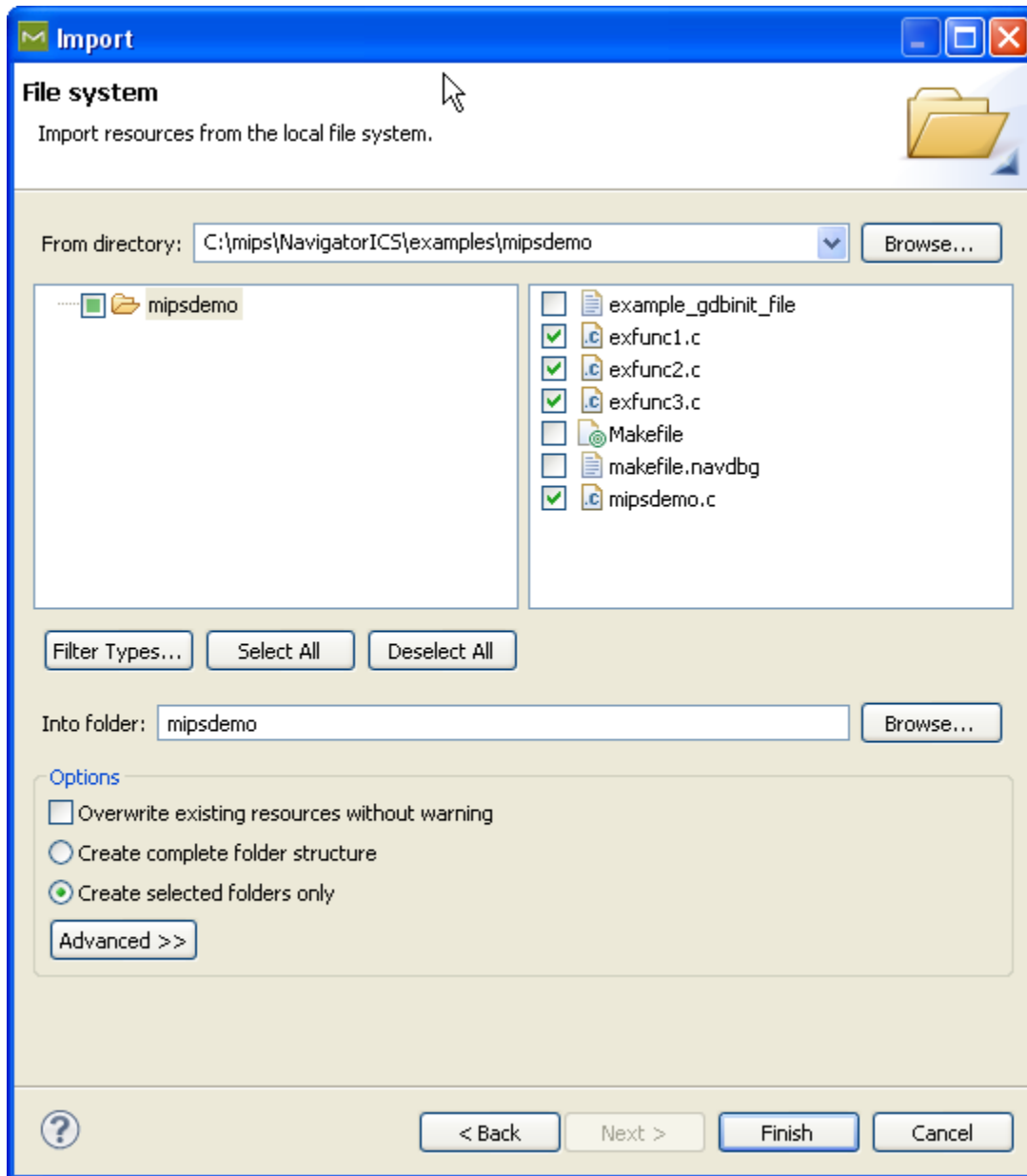
- Right-click on the **mipsdemo** project node in the project view and select **Import...** from the context menu.



- This will open the **Import** dialog. Select the **General -> File System** node then click the OK button.

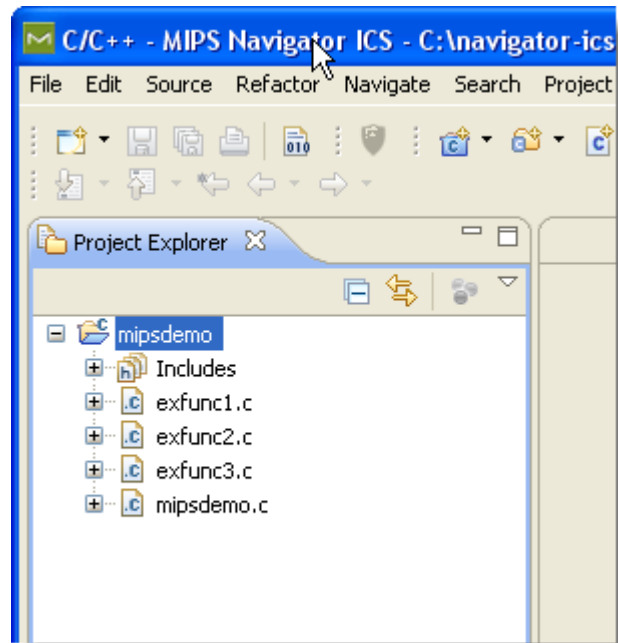


- This will open the **Import – File System** dialog. Click the Browse button and select the **C:\mips\NavigatorICS\examples\mipsdemo** directory as the directory to import from. Check the checkbox next to each of the four source files. Do not check any additional files. Configure the dialog setting according to this screenshot:



- Click the **Finish** button.

You should now have a new project with source code. It should look like this:



7.5 Building the Project

Now it is time to build the newly created project.

To start a build, select **Project | Build Project** from the main menu. If the **Build Project** item is greyed out, then you may need to click on the project node in the Project Explorer once to enable the menu item.

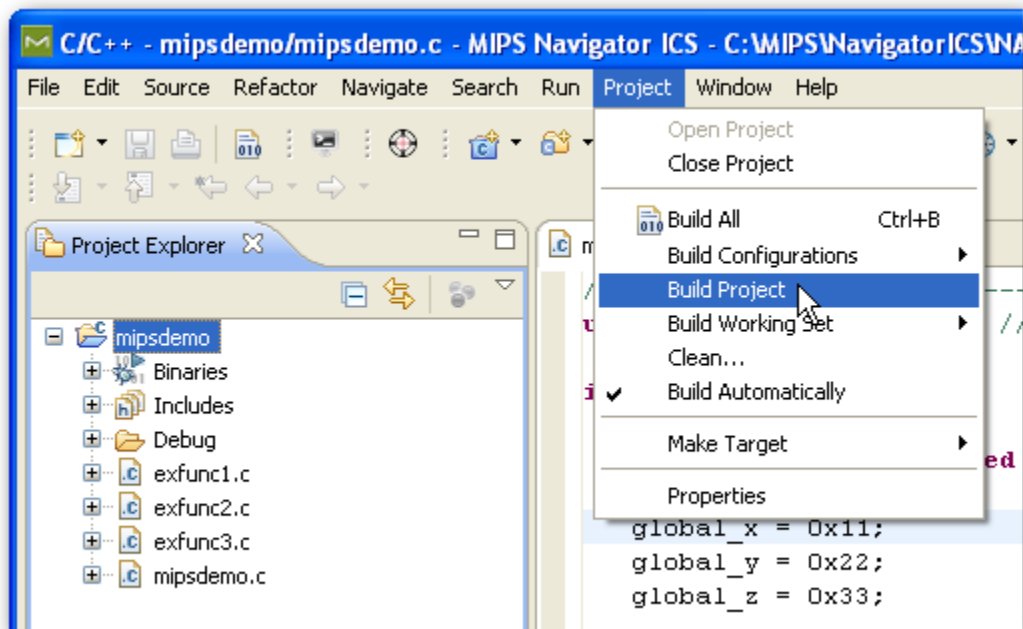


Figure 5 – Kick off a build

When the build is started a build progress dialog is displayed. Depending on how long the build takes, this dialog may disappear quickly. If the build is going to take a long time, then the process can be “backgrounded” by clicking the **Run in Background** button.

When the build is complete, you can expand the **Binaries** folder in the project explorer and see that a new binary image has been created. (Note: The binary image file does not have a file name extension, in this example the binary image file is named “mipsdemo” and is based on the project name.)

7.6 Debug Configuration

This Getting Started Guide assumes that one of the following configurations is being used: a MIPS Navigator probe connected to a Malta board (with a 24K daughter card) via a USB probe, or using MIPSsim or IASim. If using the MIPS Navigator Probe, make sure that you have run at least one debug session using the Navigator Console software. For multi-core devices such as the MIPS 1004K and 1074K cores, you must create a Navigator Console script for your device prior to continuing with this section. Review the Navigator Console Getting Started Guide, MD00732, section 6 for help on starting the console software.

Before we can start a debug session on the target board we must create a Debug Configuration. The Debug Configuration specifies everything outside of the MIPS Navigator ICS that is required to get the debug environment up and running.

7.6.1 Create a Debug Configuration

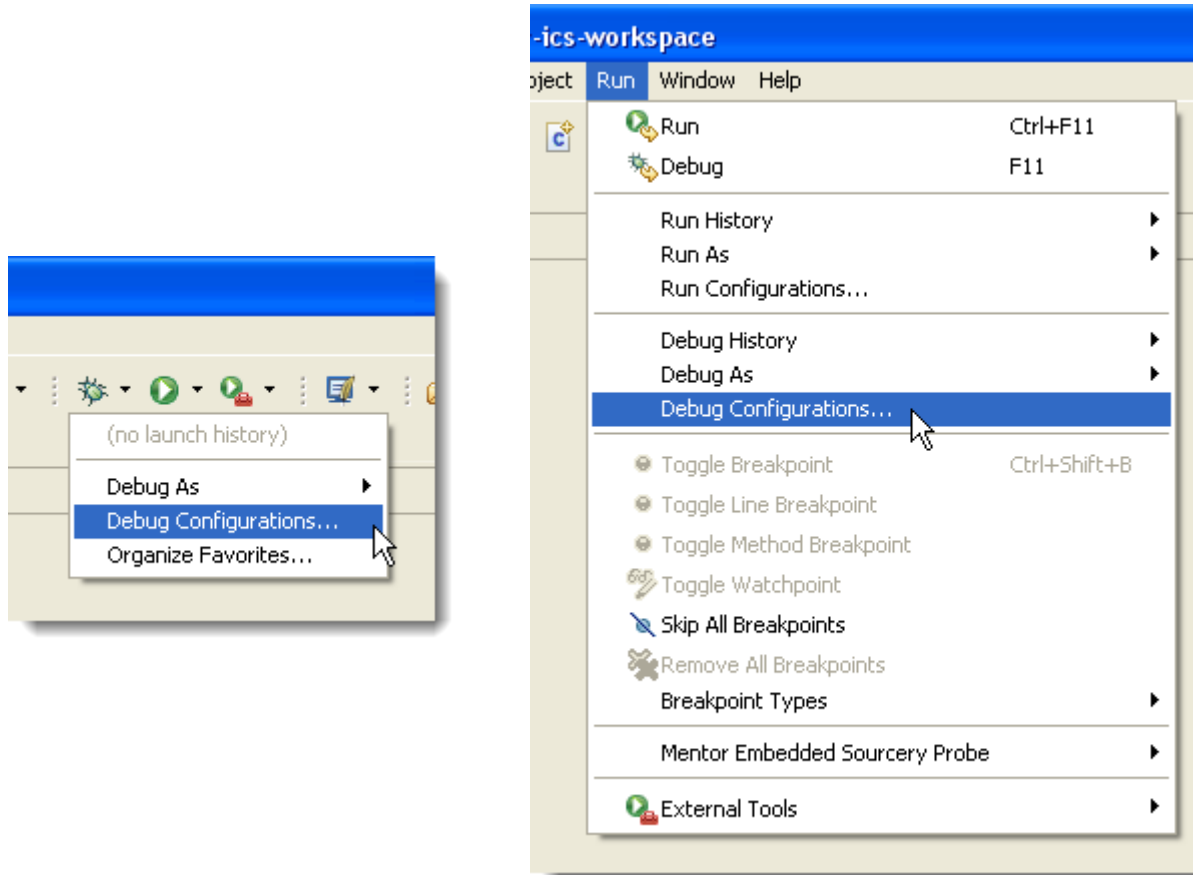

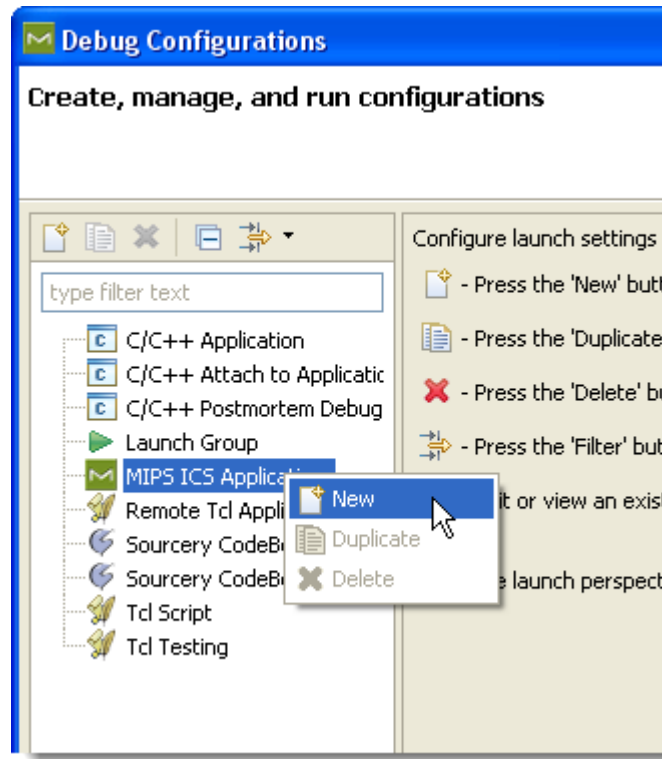


Figure 10 - Create a debug target (step 1)

First, select the **Debug Configurations ...** menu item from the main **Run** menu (or select the **Debug Configurations...** menu item from the debug tool bar button  drop down menu). This will bring up the **Debug Configurations** dialog box.



- Right-click on the **MIPS Navigator ICS Application** node and select **New**.

This will create a new blank debug configuration, named **mipsdemo Debug** (constructed from the name of the sample project), and the **Debug Configuration** dialog box will appear.

7.6.2 Debug Configuration Settings

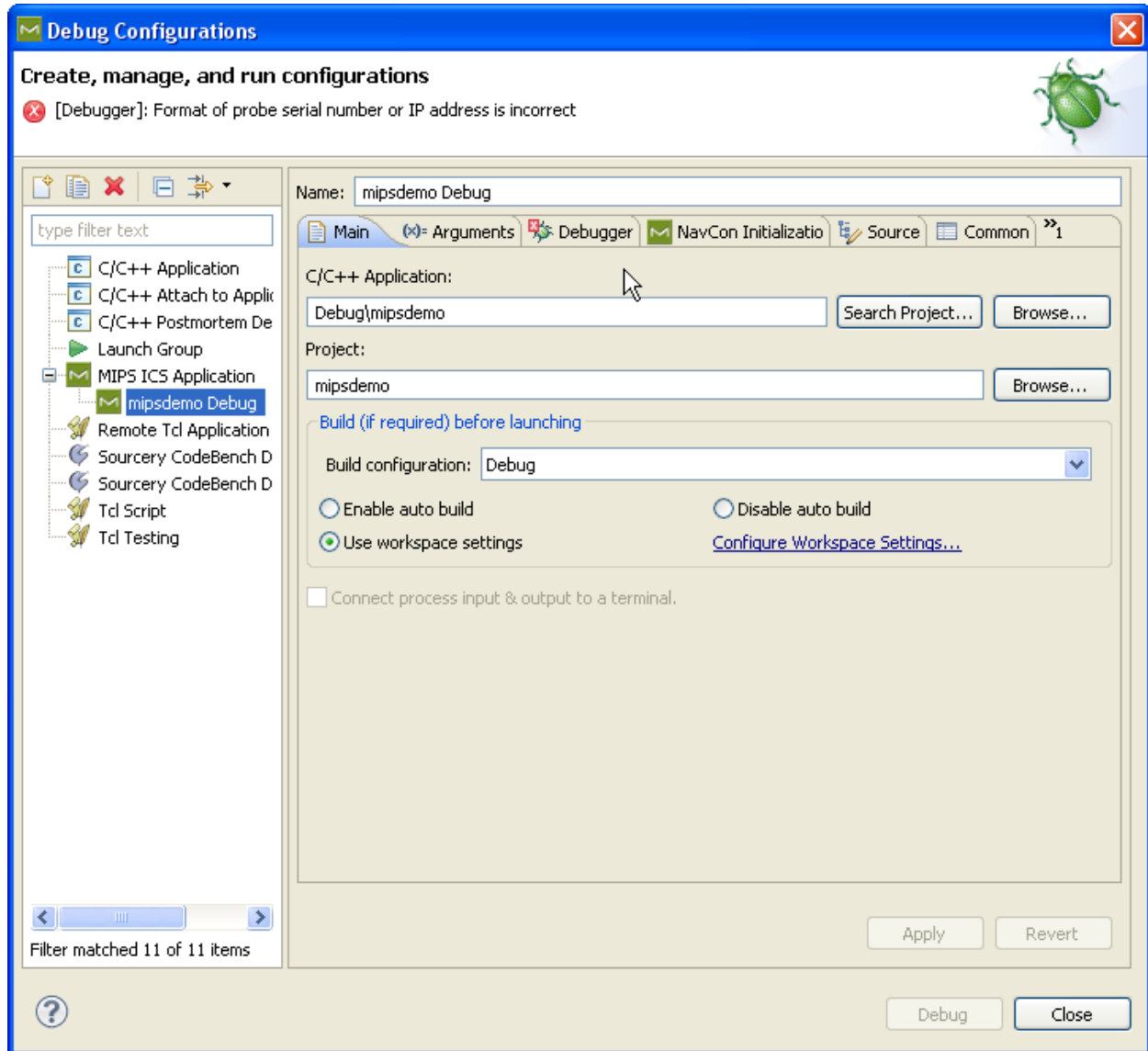


Figure 11 - Debug Configuration

If you had the **mipsdemo** project selected in the Project Explorer prior to opening the Debug Configuration dialog, then the Project and C/C++ Application fields are probably already filled in. If not, you can use the **Browse** buttons to select and fill these fields. Select the **mipsdemo** project and the **Debug\mipsdemo** binary application.

The next three sections describe configuration for these Target Interfaces:

- Using a Navigator Probe
- Using the MIPSSim Simulator

- Using the MIPS IASim Simulator

7.6.3 Using a Navigator Probe

Select the **Debugger** tab. The configuration dialog box will now look similar to the screen shot below. If using the MIPS Navigator Probe with a Malta board then all default values should be sufficient for this example.

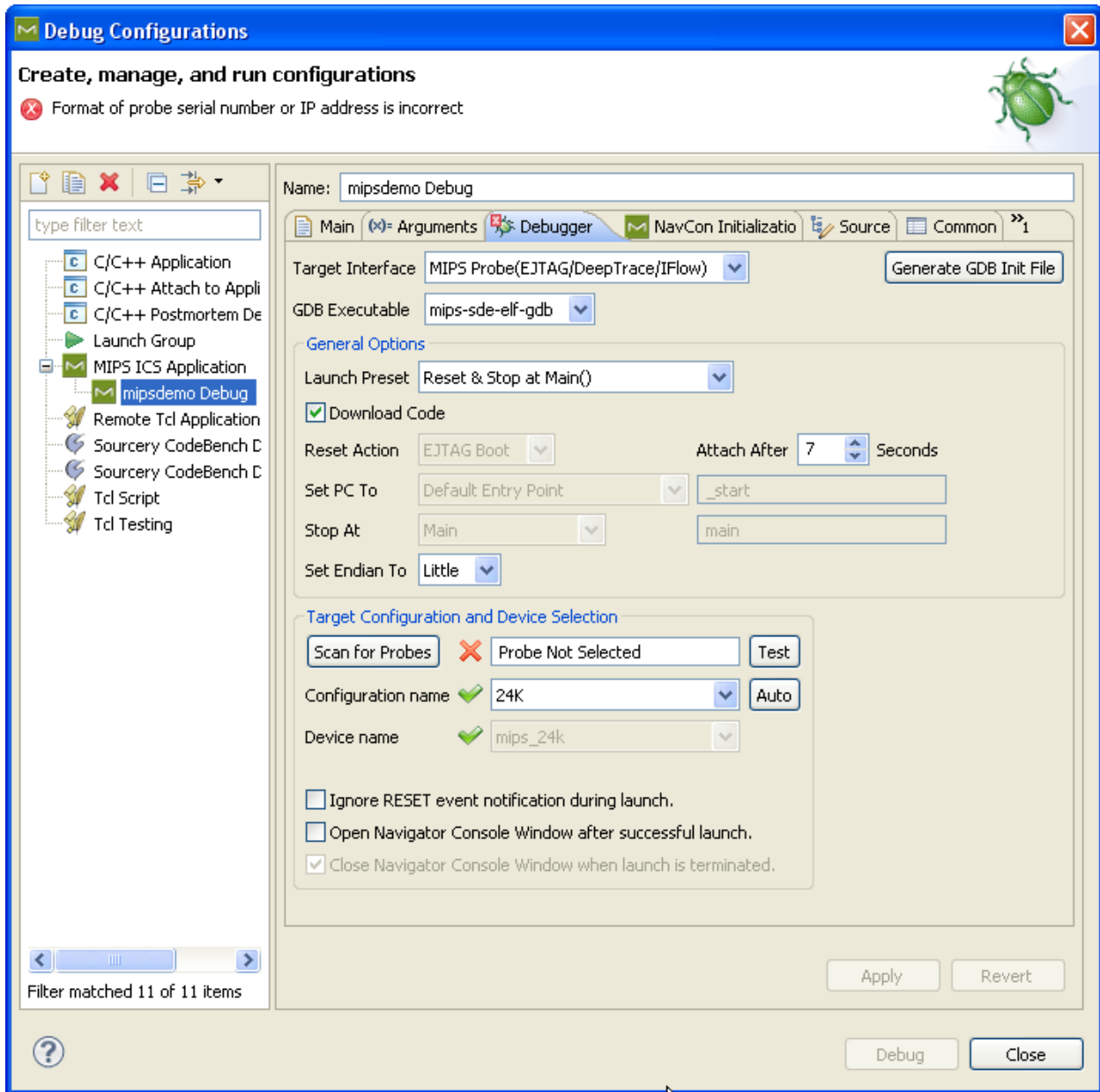


Figure 126 - Debugger Hardware Configuration (MIPS Navigator Probe Configuration)

In general, the default setting should suffice for this example. Make the following changes to the default settings:

- **Configuration Name/Device Name**

If you selected a specific processor in the project settings, then this setting may be set automatically according to your project setting. If not change this option to match the core and device that you are using. If the selected core only has a single device, then the device will be set automatically.

- **Probe Identifier**

Specify the serial number or ip-address of the Navigator Probe in this box. If the probe is connected via USB or via Ethernet on the local subnet you can use the **Scan for Probes** button to locate the probe automatically.

Set Your Configuration & Probe Preferences

If you will be using the same probe and configuration for a majority of your work, you can set the default probe serial number and default configuration and device in the global preferences. On the main menu bar select **Window → Preferences → MIPS Navigator → Probes**. When this is done, each new debug launch configuration will use these default values.

7.6.4 Using the MIPSSim Simulator

Select the **Debugger** tab. Change the **Target Interface** dropdown to select an installed version of MIPSSim. Some of the GUI elements on the **Debugger** tab will now change and the **Debugger** tab should look similar to the figure below.

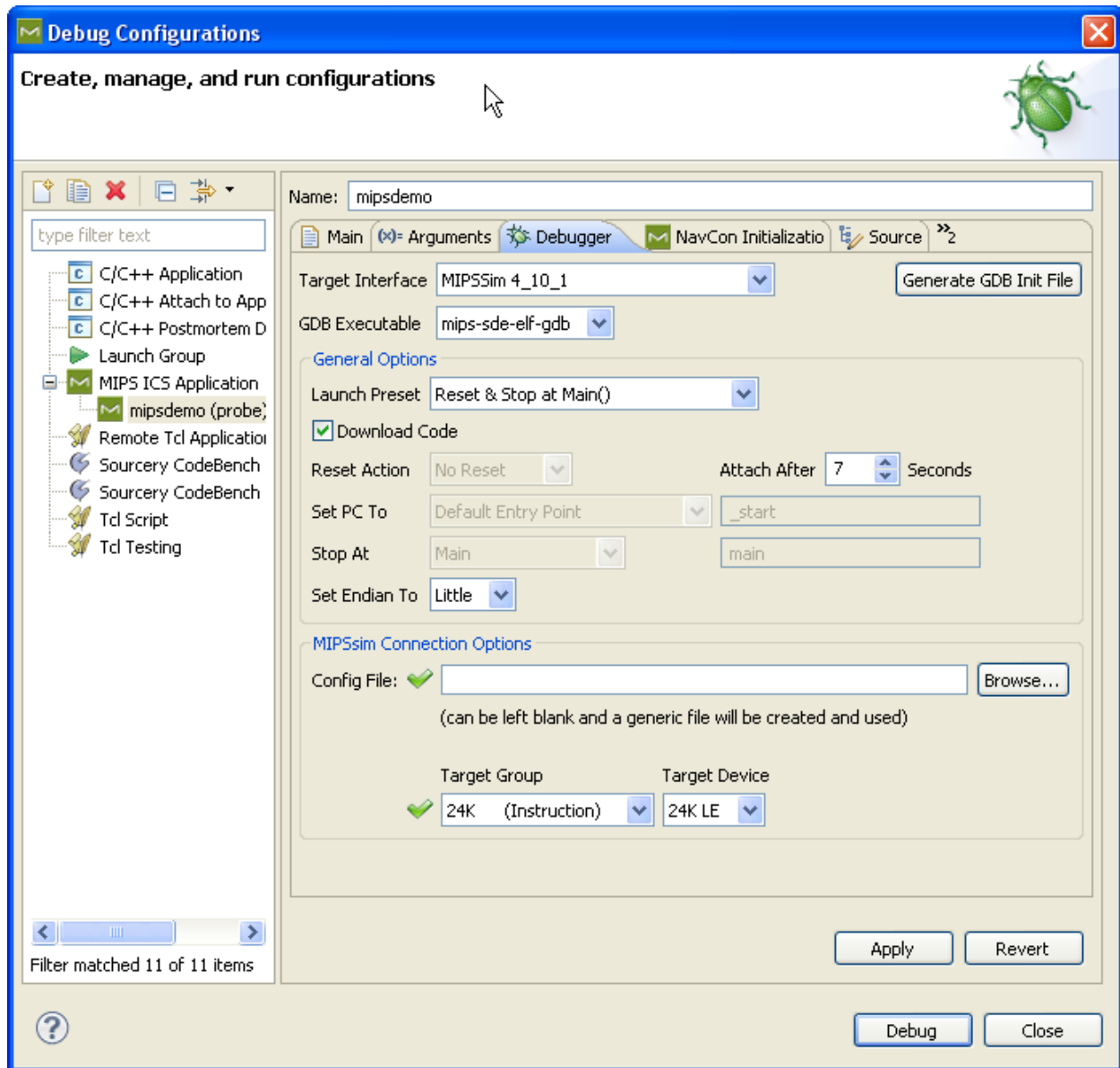


Figure 13 - Debugger Hardware Configuration (MIPSSim 4KE Configuration)

There is now a **MIPSSim Configuration File** textbox which is currently blank. If no MIPSSim configuration file is specified then an empty default file will be created and used. We do not need to specify a file at this time, but if it is desired that MIPSSim more closely mimic the targeted hardware (especially when Cycle Accurate mode is used), then a MIPSSim configuration file should be specified. Please refer to the MIPSSim documentation for information on how to create a MIPSSim configuration file.

Set the **Target Group** and **Target Device**. If you selected a specific processor in the project settings, then this setting may be set automatically according to your project setting. If not, set the target group and

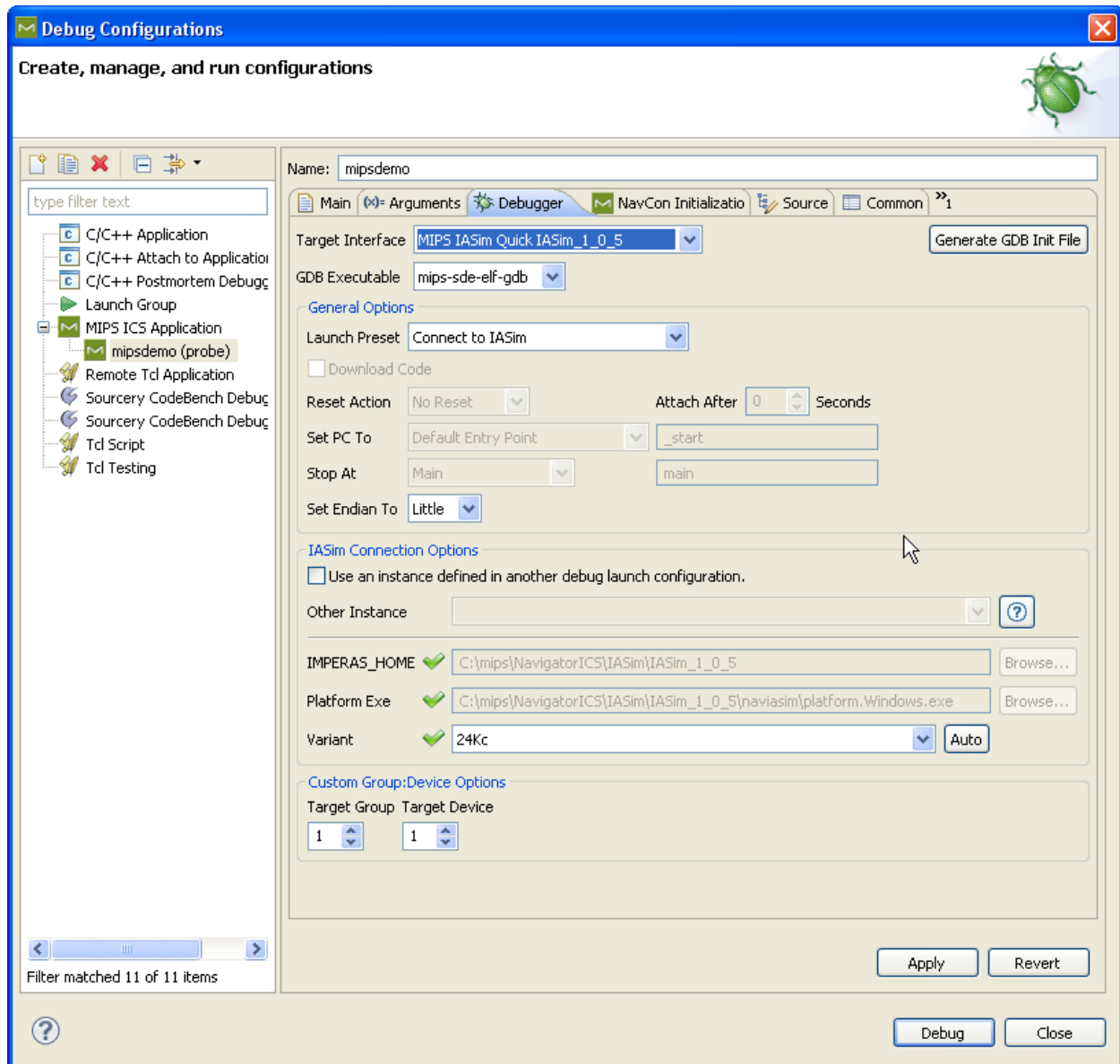
device. To continue with the targeted configuration of a 24K core change the **Target Group** dropdown to **24K (Instruction)** and change the **Target Device** to **24K LE**. It is important that the endianness in the **Device name** dropdown match the endianness which was specified in the project properties and which the project was compiled for.

Note
<p>Note on Endianness: The endianness must be consistent across the system. Check:</p> <ol style="list-style-type: none">1. The project tool settings to configure whether the project is being compiled for big or little endian2. The Debug configuration to configure the debugger endianness

The remaining tabs on the **Debug Configurations** dialog (Arguments, Source, Common, Additional GDB Commands) do not require any changes when connecting to a MIPSsim board using the **mipsdemo** example.

7.6.5 Using the MIPS IASim Simulator

Select the **Debugger** tab. Change the **Target Interface** dropdown to select **MIPS IASim Quick IASim_1_0_5**. Some of the GUI elements on the **Debugger** tab will now change and the **Debugger** tab should look similar to the figure below.



Notice the **Launch Preset** setting. It is set to **Connect to IASim**. This is the recommended setting for most circumstances. If needed, this can be changed to **Advanced** for additional control of the launch.

The **Variant** setting should be set automatically according to your project settings. If not, select the appropriate variant from the dropdown list.

The **Endian** setting should be set automatically when the launch is created (if the project executable is automatically selected). If the project executable is not automatically set upon creation (for instance when no project was selected or the project has multiple executables), then the **Endian** setting may

need to be set manually. When set manually, it is validated against the currently selected project executable to ensure correctness.

The **Target Group** and **Target Device** must be set manually. For single core models, these will always be 1 and 1.

7.6.5.1 *Advanced IASim Debug Configuration*

Changing the **Target Interface** setting to **IASim Advanced** enables several additional controls that allow you to customize several aspects of **IASim**. When the “Quick” **IASim** interface is selected, most of these controls are hidden and set to values that work with the supplied precompiled **IASim** platforms installed with Navigator ICS. Each of these controls are described briefly.

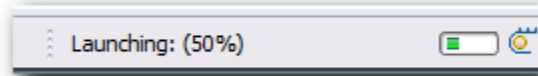
A brief description of each item in the **IASim Connection Options** group:

- **Use an instance defined in another debug launch configuration**
This control should remain unchecked when working with single-core processors.
Only use this option when you are debugging a multicore system. This checkbox allows you to use an instance of IASim that has been configured in a different launch configuration. For instance, on a MIPS 34K processor with 2 VPEs you would setup a launch configuration for vpe-0, configure an instance of IASim within that launch configuration, then create a second launch configuration for vpe-1, then, in the vpe-1 launch configuration, check this checkbox and select the vpe-0 launch configuration as the source for the IASim instance. The vpe-0 launch must be launched and active prior to launching the vpe-1 launch. When this checkbox is checked all other controls for configuring an IASim instance (those below the horizontal line) are disabled.
- **Other Instance**
This control specifies the other launch configuration (by name) that contains the IASim configuration to use for this launch.
- **Use inherited system environment variables**
When this control is checked, the four IMPERAS_? attributes will be filled in using the values inherited from the system environment when Navigator ICS was started. When a new launch configuration is created this checkbox will be checked automatically if these environment variable exists (Navigator ICS assumes you will want to use them). Clearing the checkbox will allow for manual configuration of the attribute variables. You will want to do this if you have multiple instances of IASim installed and want to manually configure which instance to use.
- **IMPERAS_HOME**
This control specifies the home (root) directory of the IASim installation that you want to use. Click the **Browse** button to browse for and select the home directory for the instance that you want to use.
- **IMPERAS_RUNTIME**
This control should always be **OVPsimMIPS**. The future may support different values.

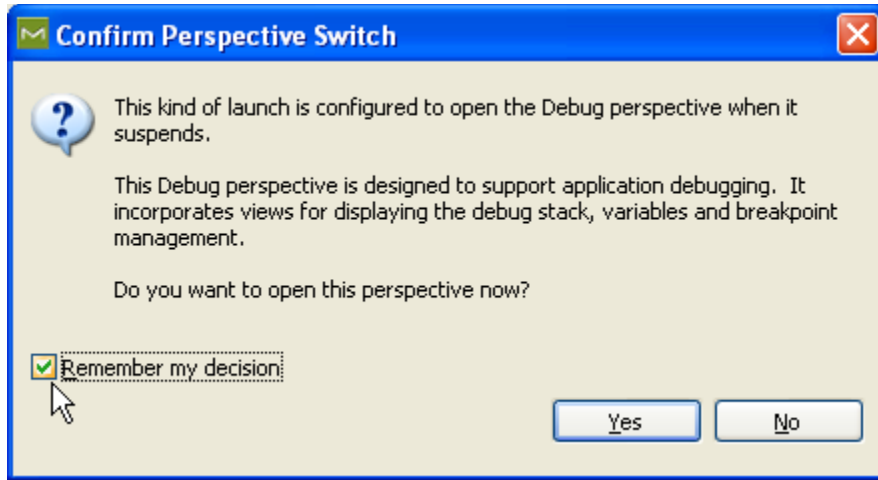
- **IMPERASE_ARCH**
This is read-only and will always reflect the host platform OS (**Windows** or **Linux**)
- **IMPERAS_VLNV**
This control will be automatically set when you change the **IMPERAS_HOME** attribute. It can also be set manually. Consult the IASim documentation for further details on this setting.
- **Platform Exe**
This control specifies the platform executable for the IASim simulator. Please ensure that the platform executable was built using the same IASim installation as referenced by the **IMPERAS_HOME** attribute. (See the section below with regard to building a platform that is compatible with the Navigator ICS debug model)
- **Variant**
Use this dropdown to select the MIPS processor variant to model in the simulator. This should match the processor variant used when compiling the project executable. The entries in the dropdown are extracted from an XML file that is included in the IASim installation and may change if you point to a different **IMPERAS_HOME** directory.
- **MDI Library**
This control is a read-only field that is automatically filled in when the **IMPERAS_HOME** attribute changed. It is provided for reference and as a double-check to ensure that an MDI library is present in the specified IASim installation.
- **Debug Port**
This control is used to specified the debug socket port (used by the MDI Library to communicate with the IASim model process). Setting this to **0** means to dynamically choose a port. This is the recommended setting since it will avoid selecting a port that may already be in use by another program. If, for some reason, you need to specify the port manually, you can specify a port number greater than 1024 here.
- **Log IASim STDOUT/STDERR to file?**
Not implemented yet.
- **Log File**
Not implemented yet.

7.7 Starting a Debug Session

Starting a debug session is now as simple as clicking the Debug button at the bottom right of the debug configuration dialog. Do that now. The MIPS Navigator ICS Progress Bar will indicate that the debug session is launching.



After several seconds the MIPS Navigator ICS may display the following message box.



Go ahead and click the “Remember my decision” checkbox, then press the **Yes** button. MIPS Navigator ICS will then reconfigure itself to present the debug perspective.

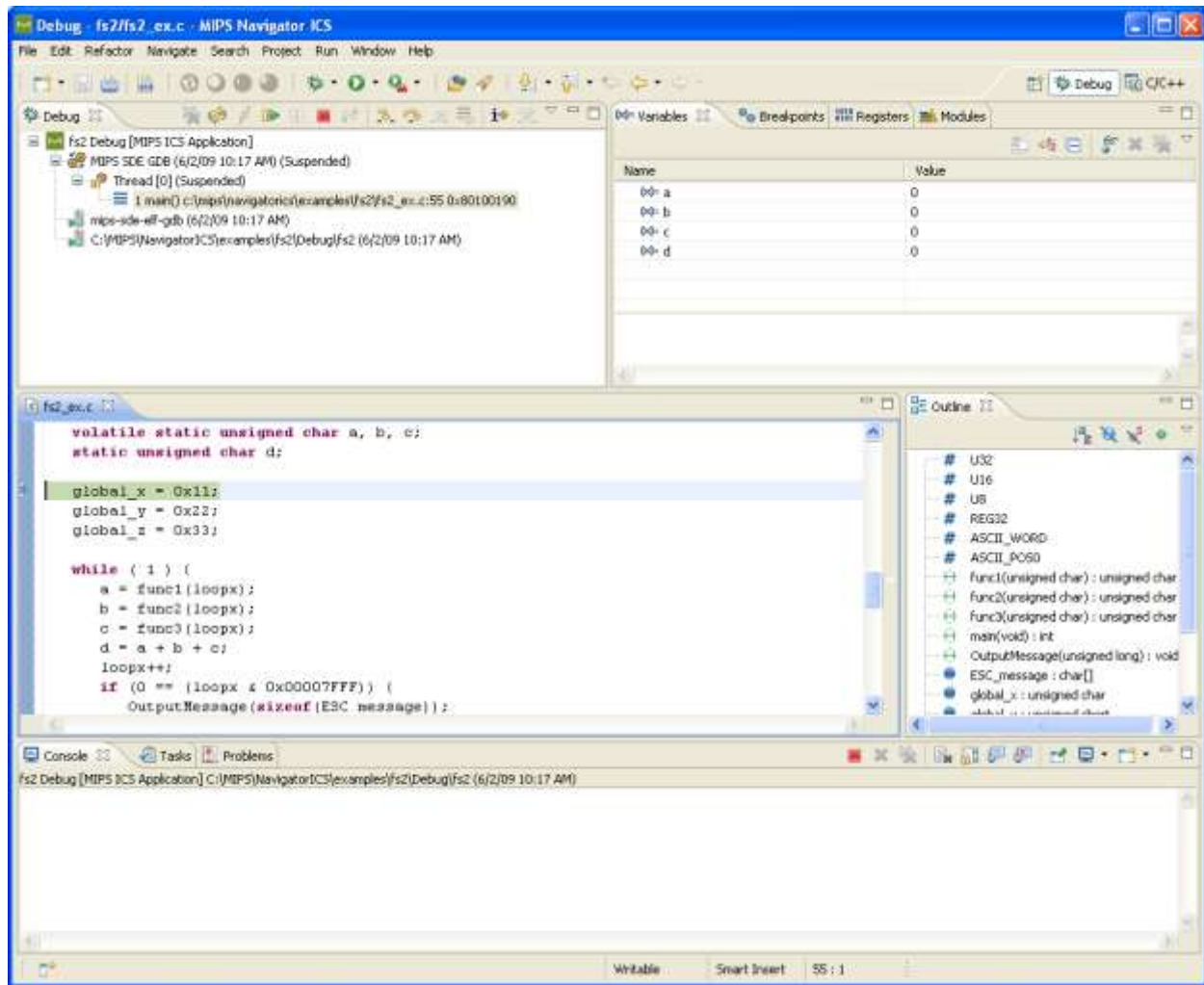


Figure 15 - The Debug Perspective

Within the debugging perspective the source code view is shown along with several other views: Debug view (upper left), the Variables, Breakpoints, Registers, and Modules views (upper right), and the Tasks and Memory views (bottom pane).

Note

If you have reached this point, you have successfully installed and configured the MIPS Navigator ICS and connected it to the target.

8 Trigger Expressions

Navigator ICS, starting at version 2.8.0, contains an updated trigger view that uses expressions to create patterns for triggers. Expressions make it very simple to create complicated patterns and provide full access to program symbolic information. This section describes the basic syntax used to create patterns.

All triggers in Navigator ICS are created from expressions that evaluate to a 32bit address (or address range on supported cores). The expressions syntax also contains syntax to specify data values, access masks, value masks, and trigger types.

Expressions are evaluated as you type them. This aids in ensuring that you are creating valid expressions. Any expressions errors are reported in real time at the top of the pattern edit dialog.

Content Assist is available when editing patterns. The default key to activate content assist is <control>-<space>. Activating content assist will pop up a companion window offering context sensitive assistance in creating a pattern. If no text is present in the pattern edit box a list of general expressions is provided.

8.1 Execution Patterns

8.1.1 Basic forms for creating execution triggers

Expression	Example	Description
<symbolname>	myfunc	Specify the name of a symbol that resolves to an execution address.
<address>	0x80101234	Specify an address literal of an execution address.
<file>:<line>	main.c:50	Creates an execution trigger at the address at the start of the specified file name and line number. File name extensions must be specified. An error is reported if the line contains no code.
:<line>	:50	Same as above but uses the filename of the currently focused file in the text editor. Once the filename is resolved it remains fixed (it is not updated if the editor is focused on another file).
<filename>	main.c	On cores that do not support range triggers this creates a pattern that triggers on the first executable address in the module. On cores that do support range triggers this creates a range pattern that triggers when any instruction in the module is

executed.

8.2 Data Patterns

8.2.1 Basic forms for creating data triggers

Expression	Example	Description
<symbolname>	myvar mystruct.myval myarray[5]	Specify the name of a symbol that resolves to a data symbol address. This should not be a stack variable. This form will create a read/write trigger on any value. Structure fields can be specified. Array indexes can be specified.
<address>	0x80101234	Specify an address literal of a data address. This form will create a read/write trigger on any value.

8.2.1.1 Basic Data Trigger Modifiers:

Expression	Example	Description
<dataexpr>=	myvar=	The addition of the '=' sign modifies the pattern so that it only triggers on data writes of any value to the specified data expression address (modeled after the assignment operator in C). Alternatively, you may prefix the symbol with a "w:", like "w:myvar", but this form does not allow for specifying values to match against.
<dataexpr>=<val>	myvar=5	Creates a pattern that only triggers when the specified value is written to the data expression address.
<dataexpr>==	myvar==	The addition of the '==' sign modifies the pattern so that it only triggers on data reads of any value to the specified data expression address (modeled after the comparison operator in C). Alternatively, you may prefix the symbol with a "r:", like "r:myvar", but this form does not allow for specifying values to match against.
<dataexpr>==<val>	myvar==5	Creates a pattern that only triggers when the specified value is read from the data expression address.

<dataexpr>===	<code>myvar===</code>	The addition of the '===' sign modifies the pattern so that it triggers on data reads or writes of any value to the specified data expression address. This is the same behavior as not using any '=' signs. Alternatively, you may prefix the symbol with a "rw:", like "rw:myvar", but this form does not allow for specifying values to match against.
<dataexpr>===<val>	<code>myvar===5</code>	Creates a pattern that only triggers when the specified value is read from or written to the data expression address.

8.2.1.2 Data Value Modifiers

Data value modifiers allow some control over what portion of the data value is used when comparing the value being read/written to the pattern value. You will want to use the value mask when attempting to trigger on data symbols that are not 32bits wide, for instance when triggering on a read from a byte or char type variable. Since the variable is only 8 bits wide you will want to mask off the 24 bits of the compare value not being used. You must also be careful to mask off the correct bytes depending on the data alignment, order, and packing used in by the compiler/linker.

Data value modifiers can only be used when specifying a hexadecimal data value. Modifiers must be specified on byte boundaries.

Modifier	Example	Description
--	<code>myvar=0x----0005</code>	Specifies an access mask on the specified byte(s).
XX	<code>myvar=0XXXXXXXX00</code>	Specifies a value mask on the specified byte(s). Values in the masked bytes will be ignored.

8.3 Address Modifiers

Address modifiers allow some control over what portion of the address value is used when comparing the address with the pattern.

Modifier	Example	Description
<addr>#<mask>	<code>myaddr#0xFFFF0000</code>	Specifies a bit-wise address mask. Set bits in the mask specify that the corresponding bits in the address are ignored during comparison.
startof:	<code>startof:mystruct</code>	Specifies that the starting address of the specified symbol

		should be used. This is the default behavior if no prefix is specified. This is provided for completeness.
endof:	endof:mystruct	Specifies that the end address of the specified symbol should be used rather than the start address. This is primarily useful for range expressions when you need to specify the end address of the range to be the end address of some symbol that is larger than 32bits (like a structure or array).
<addr>+<ofs>	myvar+10 myvar+sizeof(long)	Specifies a byte offset from the specified address expression. The offset can be a literal constant or any other expression that gdb can evaluate.

8.4 Range Expressions

These expressions can be used on MIPS cores that support range triggers. If you attempt to create a range expression on a target that does not support ranges an error is thrown.

Range expressions cannot be used with data value qualifiers.

Expression	Example	Description
!<symbol>	!myfunc	Creates an exclude-range pattern that excludes all addresses covered by the specified symbol.
@<symbol>	@mystruct	Creates an include-range pattern that includes all addresses covered by the specified symbol.
[addr1]<->[addr2]	sym1<->sym2	Creates an include-range pattern that includes all addresses spanned by the start address of sym1 and the end address of sym2. If sym2's address range is less than sym1 then the addresses will be reversed to create a valid range.
[addr1]>-<[addr2]		Creates an exclude-range pattern that excludes all addresses spanned by the start address of sym1 and the end address of sym2. If sym2's address range is less than sym1 then the addresses will be reversed to create a valid range.

8.5 Expression Notes

- Symbols do not need to be prefixed with an ‘&’ character to denote the “address of”. This is assumed. Any ‘&’ characters in the expression are removed prior to evaluating the expression.
- Some expression elements will be evaluated using gdb’s eval function. This requires that the target be halted at the time (so that gdb is available).
- An expression of “*” creates a pattern that will always fire on the next execution cycle.

9 Cache Viewer

This release of Navigator ICS incorporates a new Cache Viewer plugin. Below are some basic notes on the behavior of these new viewers:

- The best way to get a feel for how the cache viewers behave is to open them up and play around with them for a while.
- The cache views do not auto update on halt. They must be refreshed manually (using the refresh button on the cache view toolbar). This is due to the large size of the cache and the time it takes to read all the data from the target.
- The cache viewer can be configured by default to skip reading invalid cache line data from the target. This will significantly speed up reading cache from the target. If you want to read invalid cache data, uncheck the “Don’t read invalid data” checkbox in the cache view drop down menu.
- The cache views also read corresponding memory from the target in order to display an icon indicating if the value stored at memory is different than the values stored in cache. Auto memory read can be disabled in the cache view drop down menu.
- Hovering the mouse over a cell in the cache viewer will show detailed information about the cache entry including the corresponding physical address, the L2 cache value (if present), L1 cache value, the memory value (with match/mismatch icon).
- You can invalidate the entire cache with or without write back using the invalidate buttons (look like pencils) on the cache view toolbar. This is helpful if you want to see how the cache is updated when executing a specific block of code by placing a breakpoint before and after the block. When stopped at the first breakpoint invalidate the cache; then run to the second breakpoint and refresh the cache. Only changes to the cache due to the executed code will be displayed.
- The data order in the table can be toggled from the drop down menu to suit your taste.
- The cache view can show either the Physical Address of a mapped cache line or an assumed KSEG0 address. This is toggled by a checkbox in the drop down menu.
- Tips on using multiple cache viewers:
 - Multiple cache views can be opened on the same core or on different cores.
 - Multiple views can be display synchronized so that scrolling one view will scroll other views as well.

- Multiple views of a single cache space can be filtered differently.

10 Appendix A – FlexLM Licensing

10.1 License Trouble Shooting

10.1.1 Environment Variables

A single license file is provided and licenses all the features that are included in the purchased product.

MIPS Navigator ICS uses environment variables to locate license files. The following environment variables are used:

- MIPS_LICENSE_FILE – Used for all MIPS provided features
- CSLMRD_LICENSE_FILE – Used for CodeSourcery provided features

In the default/general usage case, these environment variables are defined by the application launcher script to point to a known fixed directory in the installation path. The product installer will copy your license file to these fixed license directories.

If MIPS Navigator ICS does not detect a valid license when started, MIPS Navigator ICS will prompt you for the license file. The license file will be copied to this known location and the application will be fully licensed as expected.

You may also define these environment variables in their system environment and place the license file anywhere you like. If you do this, and everything is configured accurately, then the application will not prompt you for a license file when first starting. If, however, you manually have configured these environment variables and still receive the license file prompt when the application starts, then it is likely that there is an error in the configuration. In this case, check:

- The spelling of the environment variable names
- Check that the directories specified exist
- Check that the license file is in the specified directory
- Check that there are no old, expired, or invalid licenses

10.1.2 Testing Your License

10.1.2.1 Within the IDE

This section assumes that you already have MIPS Navigator ICS installed. You must have the product installed in order to test your license file using the procedure described here.

You can enter your probe serial number into the probe serial number box next to the “Scan For Probes” button (or you can press the “Scan For Probes” button to scan for locally attached probes). If a serial number is provided when the “Check License” button is pressed then the license check will also tell you what features are enabled for the provided probe serial number.

10.1.2.2 From the Command Line

A command line tool is provided that can display diagnostic information about a license file:

```
<INSTALL-DIR>/Tools/FlexLM/lmutil lmdiag -c <license-file> -n
```

Run this command from a command shell specifying your license file to print out diagnostic information. You may also capture this information in a file and send it to support@mips.com if you continue to have license problems.

10.1.3 Troubleshooting Tips

The following tips can be used to help resolve licensing problems.

1. Remove any old CodeSourcery, MIPS Navigator ICS, Navigator IDE, System Navigator, and Navigator Console license files.
2. Check the “HKEY_LOCAL_MACHINE\SOFTWARE\FLEXlm License Manager” registry key in your registry. This key caches license file locations. Make sure that any files referenced by this registry key are valid and up to date. Make sure that there are no old expired license files.
3. Check if you have MIPS_LICENSE_FILE and/or CSLMRD_LICENSE_FILE defined in your system or user environment. If yes, ensure that they are referencing a valid license file.

10.2 License Descriptions

10.2.1 Code Sourcery License

MIPS Navigator ICS is built on the Code Sourcery SG++ Professional IDE. MIPS will provide a license that enables the SG++ IDE for both Linux and bare-iron toolchains. This license is locked to the hostid (usually a NIC MAC address) of the host machine or license server.

10.2.2 MIPS Navigator ICS License

The MIPS provided components of MIPS Navigator ICS will require one or more license features to enable all purchased features. Additional components purchased at a later date may require additional license files. This license file is node locked to the host PC running MIPS Navigator ICS or the license server in a served-license environment.

10.2.3 MIPSsim License

If the program is going to be debugged using the MIPS simulator (MIPSsim), then a MIPSsim license file is also needed. The MIPSsim license file can specify each of the Instruction Accurate (IA) cores or all of the Cycle Accurate (CA) cores, as well as other MIPSsim features. Make sure the MIPSsim license file

contains an entry for the MIPSsim core you are trying to debug. **Note:** the MIPS Navigator ICS license file is also needed in order to debug MIPSsim.

11 Appendix B – Connecting MIPS Navigator Probes

11.1 ESD Handling Precautions

The electronic components installed on the printed circuit board (PCB) are extremely sensitive to Electro-Static Discharge (ESD). Ordinary amounts of static from your clothing or work environment can damage or degrade electronic equipment. You should wear a grounding wrist strap whenever you handle a printed circuit board. The grounding strap provides a conductive path between your body and ground for discharging static electricity to ground.

When operating the MIPS Navigator probe, ESD precautions must be observed to ensure that the unit continues to function correctly. These precautions would include but are not limited to:

- Using ESD mats and wrist strap with the unit in a bench top configuration.
- The use of an ESD strap connected, with a 50ohm resistor in series, to any attachable metallic surface on the unit in any situation where ESD is a hazard.
- In the unlikely event that the equipment may be subjected to power source surges, additional surge protection in the form of a surge protected powerstrip is recommended.

11.2 Connecting the Hardware

This section describes how to connect the Navigator probes to hardware. Refer to the appropriate section for your hardware. System Navigator, Original Edition probes are no longer supported.

11.2.1 Navigator and Navigator Pro

This section details connecting to the Navigator and Navigator Pro probes.

11.2.1.1 Hardware Connections

The Navigator and Navigator Pro probes connect to the host PC via a USB 2.0 Type A to Mini-B cable and to the target system via a flat ribbon cable. USB 1.1 and USB 2.0 are supported with the USB cable supplying power. Navigator probes with Ethernet interfaces can also connect to the host PC through an Ethernet switch, hub, or crossover cable. For Ethernet mode, the power source is the 5V power supply provided with the probe. There are no user-serviceable parts in the probes.

11.2.1.2 Connecting the Hardware Using USB

Note

Never apply power to the target without the probe turned on.

Note

When power-cycling the probe, leave it turned off for at least 5 seconds.

To start a session:

- Turn off target power.
- Connect the flat ribbon cable to the target. The red strip is pin 1.
- Connect the USB cable to the computer.
- Connect the USB cable to the probe. On Windows hosts, when you first power on the unit with the USB cable connected, you may see a series of dialog boxes appear (i.e. “USB 2.0 Device”, followed by “MIPS Jnet Firmware Loader”, followed by “MIPS Navigator Probe”). On Windows XP, you may see the Found New Hardware Wizard, in which case you should choose to install the software automatically. Also on Windows XP, if you see a warning box stating that the driver you are installing for MIPS Navigator Probe has not passed Windows logo testing, it is safe to choose the Continue Anyway option. As a confirmatory step to ensure that the driver has been installed successfully, it may be useful to open the Windows Device Manager and verify that “MIPS Navigator Probe” shows up under “Universal Serial Bus controllers” section. In a few seconds, you will see the POWER LED turn on and the COMM LED flash once. This indicates the probe is properly initialized.
- Turn target power on.
- Start software.

To end a session:

- Exit the software application.

- Turn off target power.
- Disconnect the USB cable from the Navigator probe.

11.2.1.3 *Connecting the Hardware Using Ethernet*

Note

If you have a system administrator available :

It is recommended that you consult with your system administrator to determine if a DHCP server covers the network segment onto which you will attach the probe. If no such DHCP server is in place, then consult with the system administrator to determine the static IP addressing settings to be explicitly assigned to the probe.

If the local network has a DHCP server, then when the probe is powered up in Ethernet mode, the probe will acquire its TCP/IP settings using DHCP negotiation. This is the simplest connection and highly recommended.

If no DHCP server exists on the local network, then the TCP/IP settings must be set up explicitly before making an Ethernet connection for the first time. To set up TCP/IP settings:

- Connect to the probe in USB mode. See the previous section *Connecting the Hardware Using USB*.
- Start the Navigator Console software and configure the Ethernet settings manually (substituting the values appropriate for your network). Contact your system administrator for assistance on these values, as values not consistent with local network structure/policy will prevent the probe from achieving connectivity:

```
netsetup dhcp off ip 192:168:0:7 netmask 255:255:255:0 gateway  
192:168:0:1
```

- If you are connecting your host computer's Ethernet interface directly to the probe in point-to-point fashion, it is recommended that you set the probe's gateway parameter to refer to the host computer's address.

- Changes in the TCP/IP settings do not take effect until the next time that the probe is powered on in Ethernet mode. The TCP/IP settings are stored persistently in the probe flash memory, so configuring these settings is typically something that needs to be performed only once.

By default, an Ethernet-enabled Navigator probe is set up to use DHCP negotiation to acquire its TCP/IP settings on power-up.

Note

Never apply power to the target without the probe turned on.

Note

When power-cycling the probe, leave it turned off for at least 5 seconds.

To start a session:

- Turn off target power.
- If the probe is already connected to the host PC via USB, disconnect the USB cable from the probe.
- Establish a physical Ethernet connection between host and probe. This can be accomplished by connecting both host PC and probe to an Ethernet switch or hub, or by connecting the host PC directly to the probe using an Ethernet crossover cable.
- Apply power to the probe using the 5V power supply provided. The COMM LED of the probe will blink until the probe configures its TCP/IP settings.
- If the probe is set up for DHCP, then the LED blinking will last approximately 5 to 10 seconds while the negotiation is in progress. If the probe is set for static IP settings, then the blinking will last for approximately 5 seconds. If the LED doesn't stop blinking, then this indicates that DHCP negotiation has not succeeded. If this happens, verify that the local network's DHCP server is operating correctly.
- Turn target power on.
- Start software.

To end a session:

- Exit the software application.
- Turn off target power.
- Disconnect power from the probe.

12 References

1. Navigator® Console Command Reference for MIPS® Processor Cores
MIPS Document: MD00705
2. Setting Up Floating License Servers for MIPS® Navigator™ ICS 2.0.0
MIPS Document: MD00706
3. Setting Up Floating License Servers for MIPS® Navigator™ ICS 2.0.1
MIPS Document: MD00707

13 Revision History

Revision	Date	Description
01.00	February 14, 2007	Initial release
01.01	April 10, 2009	Updates to copyrights and document template
01.02	August 28, 2009	Minor edits, formatting
01.03	September 2, 2009	Updated installer instructions for 2.5.2
01.04	September 2, 2009	Minor Edits & Corrections
01.06	January 4, 2010	Updated for 2.6.0
01.07	February 18, 2010	Change path to FlexLM tools.
01.08	April 14, 2010	Updated Copyright statement
01.09	June 7, 2010	Updated Copyright statement
01.10	September 30, 2011	Updated for 2.8.0
01.11		Updated for 2.8.2