eXDI2RVI Driver for Windows
Embedded CE 5.0
Application Note 192

Released on: 29 January 2011
Table 1 Change history

<table>
<thead>
<tr>
<th>Date</th>
<th>Issue</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>February 2008</td>
<td>A</td>
<td>First release</td>
</tr>
<tr>
<td>March 2009</td>
<td>B</td>
<td>Second release</td>
</tr>
<tr>
<td>May 2009</td>
<td>C</td>
<td>Documentation update</td>
</tr>
<tr>
<td>September 2009</td>
<td>D</td>
<td>Documentation update</td>
</tr>
<tr>
<td>January 2010</td>
<td>E</td>
<td>Update for new version of driver 3.3 to 3.4</td>
</tr>
<tr>
<td>January 2011</td>
<td>F</td>
<td>Documentation update</td>
</tr>
</tbody>
</table>

Proprietary Notice

Words and logos marked with ® or ™ are registered trademarks or trademarks of ARM® Limited in the EU and other countries, except as otherwise stated below in this proprietary notice. Other brands and names mentioned herein may be the trademarks of their respective owners.

Neither the whole nor any part of the information contained in, or the product described in, this document may be adapted or reproduced in any material form except with the prior written permission of the copyright holder.

The product described in this document is subject to continuous developments and improvements. All particulars of the product and its use contained in this document are given by ARM in good faith. However, all warranties implied or expressed, including but not limited to implied warranties of merchantability, or fitness for purpose, are excluded.

This document is intended only to assist the reader in the use of the product. ARM Limited shall not be liable for any loss or damage arising from the use of any information in this document, or any error or omission in such information, or any incorrect use of the product.

Where the term ARM is used it means “ARM or any of its subsidiaries as appropriate”.

Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by ARM and the party that ARM delivered this document to.

Unrestricted Access is an ARM internal classification.

Product Status

The information in this document is final, that is for a developed product.

Web Address

http://www.arm.com
1 Introduction

The *Extended Debugging Interface to RealView ICE* (eXDI2RVI) driver provides connectivity between Platform Builder and ARM RealView ICE. This enables you to use Microsoft *Windows CE* Platform Builder 5.0 to perform hardware-assisted debug of ARM architecture-based target systems using RealView ICE v3.1 or later. The eXDI2RVI driver provides a *Graphical User Interface* (GUI) to allow access to target features that are not available through the Platform Builder GUI.

This document describes how to set up and use the eXDI2RVI driver with Platform Builder. The examples described are built for the ARM RealView Versatile Platform Baseboard for ARM926EJ-S™. It is assumed that you are familiar with RealView ICE and Platform Builder. It is also assumed that you are familiar with the Platform Builder support for hardware-assisted debugging, as described in the appropriate documentation supplied with Platform Builder.

This document contains the following sections:

- Setup and installation on page 4
- Building an image in Platform Builder 5.0 on page 6
- Using the eXDI2RVI driver with Platform Builder on page 10
- Downloading Windows CE images on page 15
- Debugging Windows CE images on page 23
- Debugging eboot on page 28
- Supported connection mechanisms on page 30.

If you are using the eXDI2RVI driver with Windows CE 6.0, see Application Note 220 *eXDI2RVI Driver for Windows Embedded CE 6.0*. 

---

**Note:**

The information provided in this document is subject to change without notice. ARM Limited reserves the right to make changes to the products described in this document at any time. Copyright © 2008-2011 ARM Limited. All rights reserved. Non-Confidential.
2 Setup and installation

The eXDI2RVI driver is intended for use on a PC that is running Windows XP Professional (Service Pack 2, or later). The eXDI2RVI driver also requires Windows CE 5.0 and ARM RealView ICE.

You must follow this installation order:
1. Install Platform Builder
2. Setup RealView ICE
3. Install the eXDI2RVI driver on page 5.

2.1 Install Platform Builder

You must ensure that you install:
1. Platform Builder for Windows CE 5.0.
2. All the updates for Windows CE 5.0 from the Microsoft website.
   Install the updates in ascending date order.

2.2 Setup RealView ICE

To setup RealView ICE correctly:
1. Set up the target hardware. See your target hardware documentation.
2. Connect the RealView ICE hardware to the target hardware and the Windows PC.
3. Start RealView ICE, and configure this as appropriate using the RealView ICE configuration dialog box (RVConfig). RVConfig should report the connection to the target system, and identify the processor to which it is connected.

   Note
   
   The eXDI2RVI driver works for TCP/IP (RealView ICE v3.1 or later) and USB (RealView ICE v3.3 or later) connections between the host machine and the RealView ICE unit. When connecting using USB, you receive a warning message if you try to use a version earlier than RealView ICE v3.3.

   Note
   
   The RealView ICE host software and the firmware being used must be the same version, otherwise a warning message is displayed. For example:

   Warning: RealView ICE host software and firmware versions do not match. It is suggested to have same version.

For details on how to use RealView ICE, see the RealView ICE and RealView Trace User Guide.
2.3 Install the eXDI2RVI driver

To install the eXDI2RVI driver:

1. Run either
   - the RealView ICE eXDI2 Driver Win32.msi file for a 32-bit Windows OS, or
   - the RealView ICE eXDI2 Driver Win64.msi file for a 64-bit Windows OS.

   **Note**
   - The .msi file is available from ARM on request.
   - If you have an existing eXDI2RVI driver already installed for Windows CE 5.0, you must first uninstall it using the same .msi file that you used to install it. After this, open and close Platform Builder before installing the new version of the driver.

2. Follow the on-screen prompts to install the eXDI2RVI driver into a directory of your choice. The default install directory is C:\Program Files\ARM\exdi2rvi\.
3 Building an image in Platform Builder 5.0

Before you can debug a Windows CE image using Platform Builder and RealView ICE, you must build the image. This section describes the procedures for working with the ARM RealView Versatile Platform Baseboard for ARM926EJ-S as an example. This contains:

- Installing the Board Support Package
- Building a Windows CE image

3.1 Installing the Board Support Package

You must install your Board Support Package (BSP) to enable Platform Builder to build images for your target. This section describes how to install the ARM RealView Versatile Platform Baseboard for ARM926EJ-S:

1. Obtain the BSP for your platform from ARM.
2. Locate and unzip the BSP file ARMVPB926EJS-WINCE500-r0p0-00alp0.zip
   This contains the file ARMVPB926EJS-WINCE500-r0p0-00alp0.msi
3. Run the ARMVPB926EJS-WINCE500-r0p0-00alp0.msi file to display the Import Wizard.
4. Follow the on-screen instructions on the Import Wizard. At the end of the installation, the ARMVPB directory is automatically created in <WINCE ROOT>\PLATFORM, where <WINCE_ROOT> is the Platform Builder installation directory.

3.2 Building a Windows CE image

To build a Windows CE image for an ARM-specific target, using an ARM BSP:

2. Select File → Manage Catalog Items... to display the Manage Catalog Items dialog box.
3. Locate the armvpb.cec file and click OK to display the Manage Catalog Items dialog box. It might be necessary to click the Import... button to add this file. The file is in the <WINCE ROOT>\PLATFORM\ARMVPB\ directory.
4. Select File → New Platform... to display the New Platform Wizard dialog box.
5. Follow the on-screen instructions and input a name for your workspace.
6. In the New Platform Wizard dialog box, from the list of available BSPs, select ARM Realview Platform Baseboard for ARM926EJ-S and click Next.
7. Proceed through the on-screen instructions to make your selections as required. Then exit from the New Platform Wizard.
8. Select Set Active Configuration... from the Build OS menu.
9. From the Set Active Configuration dialog box, select ARM Realview Versatile PB926EJ-S: ARMV4I_Debug, and click OK, see Figure 1 on page 7.
10. Select **Platform** → **Settings...** to access the **Platform Settings** dialog box for your project.

11. In the **Platform Settings** dialog box, access the **Build Options** tab. Ensure that the following checkboxes are not selected:
   - Enable kernel debugger
   - Enable KITL.

If you are creating an image to download to flash, select the **Write run-time image to flash memory (IMGFLASH=1)** checkbox.

See Figure 2.

12. Access the **Environment** tab, then click the **New...** button.

13. Set the environment variable **IMGHDSTUB**. This variable is used to control OS visibility in Platform Builder, allowing you to see information on threads, processes and modules.

See Figure 3 on page 8.
14. In the Variable Name: field, type IMGHDSTUB. In the Variable value: field, type 1.
15. Click OK to return to the main Platform Settings dialog box. Click OK again to exit this dialog box.
16. Select View → Catalog to display the Catalog window.

    Note

Steps 17 to 20 only mention example components. You must select or unselect image components according to your own requirements.

17. In the Catalog window, open the Core OS folder and its Windows CE Devices folder.
18. Open the File Systems and Data Store folder and its Storage Manager option.
19. Right-click on FAT File System, then select Add to OS Design from the context menu. See Figure 4.

20. Repeat Step 19 for the Partition Driver and Storage Manager Control Panel Applet options, respectively.
21. Access the **Build OS** menu, and ensure that in the **Global Build Settings** command the **Copy Files to Release Directory After Build** and **Make Run-Time Image After Build** commands are selected, see Figure 5.

![Figure 5 Build OS menu and selected commands](image)

22. Select **Build OS → Sysgen** to begin the build process.

23. When the build completes, check that the files **NK.bin** and **NK.nb0** are present in the release directory.

### 3.3 How to generate NK.nb0

If **NK.nb0** is not present in the release directory, use the Platform Builder inbuilt tool, `cvrtbin.exe`, to generate it. To generate the **NK.nb0** file:

1. Select **Build OS → Open Release Directory** to open the command window.

2. In the command window, enter:

   viewbin nk.bin

   See Figure 6.

![Figure 6 Command window](image)

3. Enter `cvrtbin` with appropriate command line options:

   ```
   cvrtbin -r -a <Image Start> -w <width> -l <length> nk.bin
   ```

   The value of `width` can be 8, 16, or 32. The values of `Image Start` and `length` are as shown in the command window.

   For example:

   ```
   cvrtbin -r -a 0x00070000 -w 32 -l 0x0158B940 nk.bin
   ```
4 Using the eXDI2RVI driver with Platform Builder

The eXDI2RVI driver provides connectivity between Platform Builder and RealView ICE. This section describes how to use the eXDI2RVI driver to access and configure RealView ICE. It contains:

- Accessing the eXDI console
- eXDI Console options on page 11.

4.1 Accessing the eXDI console

To download and debug images, you must use the eXDI2RVI console. This section describes how to access the console:

1. In Platform Builder, select Target → Connectivity Options... to display the Target Device Connectivity Options dialog box. See Figure 7.

2. Set up the target device connectivity:
   a. Set the Download and Transport options to None.
   b. Select eXDI2RVI from the Debugger drop-down menu.
   c. Click Apply to save the changes.
   d. Click Close.

3. Select Target → Attach Device to attach the debugger to the target. The eXDI Console dialog box appears. See Figure 8 on page 11.
4.2 eXDI Console options

The full list of user-configurable options in the eXDI Console dialog box are:

- The **Target Control** panel comprises of:
  - **Configure RVI.** If the stored configuration does not match with the current target device, click this button to configure RealView ICE. You must do this at least once to describe your target device to the eXDI2RVI driver. In addition, always set the **Post Reset State** to **RUNNING**, as shown in Figure 30 on page 27. For more information on using the RealView ICE configuration dialog box (RVConfig), see the *RealView ICE and RealView Trace User Guide*. For faster downloads through JTAG, see *Manual RealView ICE configuration* on page 30.
  - **Select Core.** Access this drop-down menu to specify the core to which the debugger is to be connected. In a single-core target, connection is made to this core by default.
  - **Reset on Connect.** Select this checkbox to reset the target hardware after connection.
  - **Run Pre-config.** Select this checkbox to write the initial configuration settings to the target. Doing this, the memory writes specified in the `exdi2rvi_preconfig.ini` file are executed before downloading or booting the Windows CE image.

---

**Note**

The pre-configuration script only supports memory write commands.

You must edit the `exdi2rvi_preconfig.ini` sample file with memory write commands that are suitable to the preconfiguration of your board. The supported commands must be in the same format as mentioned in the sample file:

```
[Preconnect commands]
Command0=MemoryWrite 0x80000000 0xABCDEF00 4
Command1=MemoryWrite 0x20000000 0x00000020 1
Command2=MemoryWrite 0x80000004 0x0000abcd 2
```
The number at the end of each command indicates the write size (4=word, 2=half-word, 1=byte). There must be only one space after each word, as shown in the sample file. The sample file is located in the eXDI2RVI driver installation directory, for example in C:\Program Files\ARM\exdi2rvi.

— **Connect.** Click this button to connect to the current target system, and to start the debug session.

--- **Note**

If more than one processor is present in the **Select Core** list, the **Connect** button is disabled. It is enabled only when you select a core from the list.

---

- **The Image** panel comprises of:

  — **The Download** checkbox. If you want to debug a program that is already on the target, ensure that this checkbox is clear. If this checkbox is clear, the eXDI driver does not download the image, and connects it to a running target.

  — **Multiple binaries.** Select this to download multiple files to the target. This feature is only available when downloading to RAM through JTAG, see *Downloading multiple binaries* on page 18.

  — In the **Debug from** panel, select the **RAM** radio button to download the image to RAM or select the **Flash** radio button to download the image to flash.

  --- **Note**

  When you select **Download**, the name of the **Debug from** panel changes to **D/L and debug from**.

---

- **Select Platform.** When downloading to flash, you can use JTAG only for supported platforms. This drop-down menu lists the supported platforms, and is shown in Table 2. To download to flash in other platforms, use Ethernet.

---

### Table 2 Supported platforms for download to flash using JTAG

<table>
<thead>
<tr>
<th>Platform</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zoom iMX31 Lite Kit</td>
<td>ARM1136</td>
</tr>
<tr>
<td>phyCORE_i.MX27</td>
<td>ARM926</td>
</tr>
<tr>
<td>phyCORE_i.MX31</td>
<td>ARM1136</td>
</tr>
<tr>
<td>SMDK2450</td>
<td>ARM926</td>
</tr>
<tr>
<td>SMDKC100</td>
<td>Cortex™-A8</td>
</tr>
</tbody>
</table>

---

- In the **D/L through** panel, select the **JTAG** radio button to download the image using JTAG or select the **Ethernet** radio button to download the image using Ethernet. Downloading through Ethernet is faster than downloading through JTAG.

- **Write Check.** This option enables read back verification during image download. If the image written to memory does not match the data read back then the driver will print an error message detailing the location and reason for the mismatch. If you use this facility, the download time increases. In this software release, this feature is only available when downloading to RAM.

- The **D/L and** panel. Select the **Boot** radio button to boot the OS after image download or select the **Stop** radio button if you want the OS program to be stopped after the image is downloaded.
— The **Image File** field is used to denote the location of the image (Nk.nb0) to be downloaded. For example, C:\WINCES001\PBWorkspaces\vs\Test_image_pb926\RelDir\ARMVPB926EJS_ARMV4I_Debug\Nk.nb0

----- Note ----- 
If Nk.nb0 is not present in this directory, you can use one of the Platform Builder inbuilt tools to generate it. See *How to generate NK.nb0* on page 9.

— The **Load Address** field. You must enter the image start address in this field. See *How to generate NK.nb0* on page 9 for more information on how to obtain the image start address. This field accepts only valid 32-bit hexadecimal addresses. The first two characters, 0x, cannot be edited.

----- Note ----- 
If you select *Ethernet* from the *D/L through* panel, the **Load Address** changes to **Start Address**.

— The **Treat Load Address as** panel is enabled only when you have selected the **Download** checkbox. This panel comprises two radio buttons, **Physical** (default) and **Virtual**, which tell the eXDI driver whether to interpret the load address as physical or virtual.

  If the load address mentioned is physical, select **Physical**. If eXDI detects that the Memory Management Unit (MMU) on the target is enabled, eXDI disables the MMU and downloads the image to a physical address specified by you.

  If the load address mentioned is virtual, select **Download** and **Virtual**. If eXDI detects that the MMU on the target is disabled, eXDI enables the MMU and downloads the image to a virtual address specified by you.

- **View** panel. Click the **Registers** button to display the core’s CP15 registers.

- **Log Control** panel comprises of:
  — The **Clear Log** button. Click this to clear messages displayed in the message panel. These messages relate to details of the connected device.
  — The **Separator** button. Click this to add a separator line between messages in the message panel.

- The message panel displays error messages or download progress information for the user.

----- Note ----- 
If the message panel displays the following message, it means that the rvi.rvc file is missing:

  **Configure RealView ICE**

  You must configure RealView ICE and create the rvi.rvc file using the **Configure RVI** button. See *Manual RealView ICE configuration* on page 30.

----- 

- The target status bar is located along the bottom of the eXDI Console. One part of it displays the target status message as applicable, and the other part displays the current execution world. Examples of target status message are:

  — **Ready to connect**. The target is ready for connection to begin.
  — **Connecting to target**... An attempt is being made to connect to the target.
  — **Target connected**. The target is connected to the debugger.
— Target connection failed. An attempt to connect to the target has failed.
— Target not configured. The target is not yet configured, and acts as a prompt for you to configure the target in order to connect to it.

Note
If the RealView ICE and the target are already configured when you select Attach Device, the eXDI driver tries to connect to RealView ICE using the saved rvi.rvc file. Consequently you must keep the RealView ICE and the target powered ON before selecting Attach Device.

Configuring please wait... The target has been configured, the user interface is updating, and you must wait until this is complete.
— Target Running. The target is in running state.
— Target Stopped. The target is in stopped state.

Examples to denote the current execution world are Secure and Normal. These worlds are applicable only to those cores with TrustZone®, such as ARM1176, Cortex-A8 and Cortex-A9.

• The About button displays version information about the eXDI2RVI driver.
5 Downloading Windows CE images

This section contains:

- Downloading Windows CE images through JTAG
- Downloading multiple binaries on page 18
- Downloading Windows CE images through Ethernet on page 20.

5.1 Downloading Windows CE images through JTAG

You can use JTAG to download a Windows CE image to RAM or flash:

- Downloading to RAM through JTAG
- Downloading to flash through JTAG on page 17.

Downloading to RAM through JTAG

To download an image to RAM through JTAG:

1. Setup the target device connectivity options, and attach the debugger to the target to access the eXDI console, if not already done, see Accessing the eXDI console on page 10.

2. Configure RVI, if not already done, see Accessing the eXDI console on page 10.

3. Select the Download checkbox.

4. Select RAM from the D/L and debug from panel.

5. Select JTAG from the D/L through panel.

6. In the D/L and panel, select Boot to boot the target after the download or select Stop to halt the target at the load address.

7. Set the Load Address field to the Image Start, see How to generate NK.nb0 on page 9.

8. Select either Physical or Virtual as appropriate from the Treat Load Address as panel, see eXDI Console options on page 11.

9. Access the Select Core drop-down menu and select your core, for example ARM926EJ-S. Figure 9 on page 16 shows an example of a completed eXDI Console dialog box.
10. When you are ready to proceed, click the Connect button to connect to the current target system, and to invoke the debugger. See Figure 10.

Figure 9 Example eXDI Console dialog box

Figure 10 eXDI Console dialog box showing target status and connection information

Downloading starts. The message panel provides target details and the status of the image being downloaded to the target. At the same time, an Image Download dialog box displays the progress of the download, see Figure 11 on page 17.
If **Boot** is checked in the **D/L and** panel, then when downloading completes, the target starts running and the status bar displays **Target Running**, otherwise the target stops at the load address and the status bar displays **Target Stopped**, see Figure 12.

**Downloading to flash through JTAG**

To download an image to flash through JTAG:

1. Setup the target device connectivity options, and attach the debugger to the target to access the eXDI console, if not already done, see *Accessing the eXDI console* on page 10.
2. Configure RVI, if not already done, see *Accessing the eXDI console* on page 10.
3. Select the **Download** checkbox.
4. Select **Flash** from the **D/L and debug from** panel.
5. Type the full path name of the image in the **Image File** panel.
6. Select your platform from the **Select Platform** drop down menu, see Table 2 on page 12.
7. Select **JTAG** from the **D/L through** panel.
8. In the **D/L and** panel, select **Boot** to boot the target after the download or select **Stop** to halt the target at the load address.

9. Set the **Load Address** field.

10. Select either **Physical** or **Virtual** as appropriate from the **Treat Load Address as** panel, see **eXDI Console options** on page 11.

11. Access the **Select Core** drop-down menu and select your core, for example ARM926EJ-S. Figure 13 shows an example of a completed eXDI Console dialog box.

![Figure 13 Example eXDI Console dialog box](image)

12. When you are ready to proceed, click the **Connect** button to connect to the current target system, and to invoke the debugger.

Downloading starts. The message panel provides target details and the status of the image being downloaded to the target.

If **Boot** is checked in the **D/L and** panel, then when downloading completes, the target starts running and the status bar displays **Target Running**, otherwise the target stops at the load address and the status bar displays **Target Stopped**.

### 5.2 Downloading multiple binaries

To download a Windows CE image as multiple binaries:

1. Setup the target device connectivity options, and attach the debugger to the target to access the eXDI console, if not already done, see **Accessing the eXDI console** on page 10.

2. Configure RVI, if not already done, see **Accessing the eXDI console** on page 10.

3. Select the **Download** checkbox.

4. Select **RAM** from the **D/L and debug from** panel.

5. Open the `exdi2rvi_multi_download.ini` file, which is located in the eXDI2RVI driver installation directory, for example C:\Program Files\ARM\EXDI2RVI\.
6. In the exdi2rvi_multi_download.ini file, write the absolute paths and respective start addresses of all the binary files to download. The file paths must be labeled as File\n and the start addresses must be labeled as LoadAddress\n, when \n is an integer counting from 1. Figure 14 shows an example.

7. Select JTAG from the D/L through panel.

8. Select the Multiple binaries checkbox.

9. In the D/L and panel, select Boot to boot the target after the download or select Stop to halt the target at the load address.

10. Set the Load Address field. This is the address where the target boots from or stops at after the download completes.

11. Select either Physical or Virtual as appropriate from the Treat Load Address as panel, see eXDI Console options on page 11.

12. Access the Select Core drop-down menu and select your core, for example ARM926EJ-S. Figure 9 on page 16 shows an example of a completed eXDI Console dialog box.
13. When you are ready to proceed, click the **Connect** button to connect to the current target system, and to invoke the debugger.

Downloading starts. The message panel provides target details and the status of the image being downloaded to the target.

### 5.3 Downloading Windows CE images through Ethernet

You can download an image to RAM or flash using Ethernet. Before you can download through Ethernet, you must load the Ethernet bootloader (eboot) on to the target.

To download using Ethernet:

1. In Platform Builder, select **Connectivity Options**... from the **Target** menu. The Target Device Connectivity Options dialog box appears, see Figure 16.

![Figure 16 Target Device Connectivity Options dialog box](image)

2. Set the **Download** option to **Ethernet**.

3. Click **Settings** beside the **Download** option to open the **Ethernet Download Settings** dialog box, see Figure 17.

![Figure 17 Ethernet Download Settings](image)
4. Ensure that eboot is running on the target.
5. Ensure that eboot sends a BootMe signal so that it is listed in the Active target devices.
6. Select the target device from the Active target devices panel, see Figure 18.

![Figure 18 Download settings for ARMVPB4876](image)

7. Set the Transport option to Ethernet.
8. Select the eXDI2RVI option from the Debugger drop-down menu, see Figure 19.

![Figure 19 Connectivity options for downloading through Ethernet](image)

9. Click Apply and then Close.
10. Attach the debugger to the target to access the eXDI console, if not already done, see Accessing the eXDI console on page 10.
11. Configure RVI, if not already done, see Accessing the eXDI console on page 10.
12. Select the Download checkbox and select Flash from the D/L and debug from panel to download to flash, or select RAM to download to RAM.
13. If you select Stop from the D/L and panel, set the Start Address field to the start address of the image, see How to generate NK.nb0 on page 9.
14. Access the **Select Core** drop-down menu and select your core, for example ARM926EJ-S.

15. Select **Reset on Connect**. Figure 20 shows an example of the eXDI Console dialog box.

![Figure 20 Example eXDI Console dialog box](image)

16. When you are ready to proceed, click the **Connect** button to connect to the current target system, and to invoke the debugger.

Downloading starts. The message panel provides target details and the status of the image being downloaded to the target.

If **Boot** is checked in the **D/L and** panel, then when downloading completes, the target starts running and the status bar displays **Target Running**, otherwise the target stops at the Start Address and the status bar displays **Target Stopped**.
6 Debugging Windows CE images

When downloading completes, the image starts booting if the **Boot** radio button is checked in the **D/L and** panel. To debug the image running on the target, select **Debug → Break All** from the main menu. If you want to view the debug features and manipulate debug processes, select **View → Debug Windows** and then select any of the available sub-options. For example, these are **Processes, Modules and Symbols, Threads, and Registers.** See Figure 21.

![Figure 21 Debug features](image)

By default, Platform Builder sets software breakpoints. To set a new hardware breakpoint:

1. Select **View → Debug Windows → Breakpoints List** to open the **Breakpoints List** dialog box.

2. Click the **New Breakpoint** icon to open the **New Breakpoint** dialog box. See Figure 22.

![Figure 22 New Breakpoint icon](image)

3. In the New Breakpoint dialog box, enter the new address into the **Break at:** field and click the **Hardware...** button. See Figure 23 on page 24.
4. In the Breakpoint Selection dialog box, click the **Hardware** radio button and then click **OK**. See Figure 24.

5. In the New Breakpoint dialog box, click **OK**.

To change an already-set software breakpoint to a hardware breakpoint:

1. Select **View** → **Debug Windows** → **Breakpoints List** to open the **Breakpoints List** dialog box.

2. Right-click on the address you want to set a hardware breakpoint, and select **Breakpoint Properties** from the context menu. See Figure 25.

3. In the Breakpoint Properties dialog box, click the **Hardware**... button. See Figure 26 on page 25.
4. In the Breakpoint Selection dialog box, click the **Hardware** radio button and click **OK**. See Figure 27.

5. In the Breakpoint Properties dialog box, click **OK**.

The eXDI Console displays the number of hardware breakpoints that are available, shown in the message panel in Figure 12 on page 17. If you attempt to set a hardware breakpoint when there are none available, an error message displays in the **eXDI Console** dialog box. For example:

```
ERROR: Insufficient bkpt resources. Setting software bkpt at 0x71024
```

Note
Breakpoints can be set in virtual address space only after enabling the MMU. If you have set breakpoints before enabling the MMU, these breakpoints must be cleared before MMU enable.

6.1 The eXDI Coprocessor And Peripheral Registers window

You can access CP15 registers in the **eXDI Coprocessor and Peripheral Registers** window. The window show non-readable registers as `!!!!!`, see Figure 28 on page 26.

For example if you connect to a Cortex-A8 target, the Secure World register values appear as `!!!!!`.
Click the **Secure World** button to list the Secure World coprocessor registers in a new window, for example the CortexA8 - CP15 - Secure World window, see Figure 29.

Register details here are also hidden by exclamation marks, and to enable the debug privileges you must contact the device vendors.

--- **Note** ---

Platform Builder Registers, and eXDI coprocessor and peripheral registers are editable only in Target Stop mode.

### 6.2 Connecting to a running or halted target

eXDI can connect to a running or a halted target. If you connect eXDI to a running target, you must halt the target before debugging it, by selecting **Debug → Break All** from the Platform Builder main menu. Otherwise if you connect eXDI to a halted target, you can start debugging from the instruction where the target is halted.

To connect to a running program without resetting or stopping the target:

1. In the RealView ICE RVConfig dialog box, click **Advanced** in the tree diagram, then ensure that in the advanced controls list you unselect the **Perform TAP reset on first connect** checkbox.
2. Expand Devices in the tree diagram and click the required device name, for example ARM926EJ-S.

3. In the **Post Reset State** drop-down menu, select **0 - RUNNING**. See Figure 30.

![Figure 30 RVConfig - Post Reset State](image)

4. Select **File → Save**, and then select **File → Exit**.

   For more information on using the RVConfig dialog box, see the *RealView ICE and RealView Trace User Guide*.

5. In the eXDI Console dialog box, ensure that you unselect the **Download** and the **Reset on Connect** checkboxes, then click the **Connect** button.

   **Note**

   The **Post Reset State** item is not available in RVConfig for Cortex targets prior to RealView ICE v3.4.
7 Debugging eboot

This section describes the method of debugging eboot with Platform Builder and eXDI2RVI driver. It also covers the creation of the eboot Project.

7.1 Creating the eboot project

Platform Builder debugs images only of the type nk.bin, therefore you must first rename all eboot.* filenames to nk.* before attempting to debug eboot. To do this:

1. Create a folder, for example eboot, under the directory C:\WINCE500\PBWorkspaces\<current workspace>\RelDir
2. Open the Workspace in Platform Builder, then select View → File View.
3. In the File View window, drop the tree to view the eboot node of the current target board. For example, C:\WINCE500\PLATFORM\<BoardBSP_node>\eboot
4. Right-click on the eboot node and select Settings from the context menu to display the window Editing C:\WINCE500\PLATFORM\<BoardBSP>\eboot\sources
5. Access the General tab, change Target .pdb Name and Target Name to eboot, then click OK.
6. In the File View window, right-click on the node C:\WINCE500\PLATFORM\<BoardBSP_node>\eboot and select Clean Before Building and Make Run-Time Image After Build. Then select Build Current Project.
7. Ensure that the project is built without any errors. If this is not the case, rebuild the project.
8. From the directory C:\WINCE500\PBWorkspaces\<current workspace>\RelDir\<current workspace>_Debug, copy eboot.* to the eboot directory created in Step 1.
9. Then copy the eboot.pdb file from C:\WINCE500\PLATFORM\<BoardBSP_node>\eboot to the eboot directory created in Step 1.
10. In the directory created in Step 1, rename all eboot.* file names to read as nk.* (for example, rename eboot.bin to nk.bin and eboot.nb0 to nk.nb0).

7.2 Debugging eboot

To begin debugging eboot:

1. Open Platform Builder, then select File → Open Workspace.
2. Locate and select nk.bin, for example in C:\WINCE500\PBWorkspaces\vs\Test_image_pb926\RelDir\eboot, as the Project. See Figure 31 on page 29.
3. Select **Target** → **Connectivity Options** to open the **Target Device Connectivity Options** dialog box.

4. Ensure that the **Download** and **Transport** options are set as **None**, and that the **Debugger** option is set to **eXDI2RVI**.

5. Select **Target** → **Attach Device** to connect to the target and open the **eXDI Console** dialog box.

6. In the **eXDI Console** dialog box, select the **Download** checkbox.

7. Select the required core, for example ARM926EJ-S.

8. Enter the appropriate **Load Address** where you want to download nk.nb0.
   For more details on the **Load Address** field, see the **Image** panel information in Accessing the eXDI console on page 10.

9. Select the **Stop** radio button from the **D/L and** panel.

10. Click the **Connect** button to connect to the target and to start the image download process.

11. Select **File** → **Open** and then browse to the appropriate eboot directory. For example, C:\WINCE500\PLATFORM\<BoardBSP_directory>\eboot.

12. Open the source files to view and add breakpoints in them.
8  Supported connection mechanisms

This section describes the mechanisms that are supported when using the eXDI driver:

- Auto connection
- Manual RealView ICE configuration.

8.1  Auto connection

The eXDI driver supports auto connection by means of the Connect button located in the eXDI Console dialog box. See Accessing the eXDI console on page 10. When you click the Connect button, connection is automatically made to the target.

Connection is automatically made to the target only if the stored configuration matches that of the current target environment. If this is not the case, you must attempt a manual configuration of RealView ICE. See Manual RealView ICE configuration.

8.2  Manual RealView ICE configuration

To perform a manual configuration of RealView ICE when using the eXDI driver:

1. In the eXDI Console dialog box, click the Configure RVI button to open the RVConfig dialog box.

2. In the RVConfig dialog box select the RVI from the RealView ICE browser list or enter the IP address or host name of the RVI in the IP Address / Host Name field. Then click the Connect button.

3. Click the Auto Configure button.

4. When this process is completed, click the Advanced node in the tree diagram, then set appropriate values for the relevant RealView ICE settings. See Figure 32 on page 31 for example Reset Type values.
1. Select **File → Save** to save the RealView ICE configuration for the new target.

1. Select **File → Exit**.

2. Click the **Connect** button in the **eXDI Console** dialog box. If the connection fails, follow the prompt messages.

For more information on using the RVConfig dialog box, see the *RealView ICE and RealView Trace User Guide*.

**Note**

You might achieve faster image download speeds by using the Clock Speed controls in the RVConfig dialog box to change the clock speed setting from **Adaptive** to a numeric frequency value. To do this successfully, you must know the frequency range applicable to each core, because not all cores support all the frequency ranges that are available in the RVConfig Clock Speed controls list.