

RDI for MAJIC

User's Manual



Embedded Performance, Inc.

January, 2005

EPI has made every attempt to ensure that the information in this document is accurate and complete. However, EPI assumes no responsibility for any errors, omissions, or for any consequences resulting from the use of the information included herein or the equipment it accompanies. EPI reserves the right to make changes in its products and specifications at any time without notice.

Any software described in this document is furnished under a license or non-disclosure agreement. It is against the law to copy this software on magnetic tape, disk, or other medium for any purpose other than the licensee's personal use.

Embedded Performance, Incorporated
606 Valley Way
Milpitas, California 95035
USA

Voice: (408) 957-0350

FAX: (408) 957-0307

email: sales@epitools.com
support@epitools.com

WWW: <http://www.epitools.com>

Acknowledgments:

ARM, ADW, AXD are trademarks of ARM Technologies, Inc.

Windows is a trademark of Microsoft Corporation.

UNIX is a trademark of AT&T.

Ethernet is a trademark of XEROX.

MAJIC, MONICE, EDBICE, and EPI are trademarks of Embedded Performance, Inc.

Multi is a trademark of Greenhills Software.

All other trademarks are trademarks of their respective companies.

© 2000-2005 Embedded Performance, Incorporated.

All rights reserved.

Table of Contents

Table of Contents	iii
Introduction	1
About this Manual	1
What is RDI	1
Related Documentation	2
Notational Conventions	2
Getting Started	3
Configuring the RDI Environment	3
RDI/MAJIC Configuration	3
Using the MAJIC Setup Wizard	4
Debugger Configuration	9
ARM ADS/AXD	9
ARM ADW	11
GreenHill's Multi	13
RDI MAJIC Console Window	16
Advanced Configuration	17
Configuration File	17
Contents	17
Organization	18
Global	18
Device	19
Controller	19
RDIDeviceList	20
Startup Command File	21

1 ***Introduction***

This chapter provides a brief overview of the scope of this manual, what "RDI" is, and EPI's implementation specifics. It also identifies specific reference documentation for topics outside the scope of this document.

About this Manual

This is the user manual for the Embedded Performance RDI interface library for EPI's MAJIC Series of Intelligent Debug Probes. It is aimed at end-users who intend to use the MAJIC with debuggers that support the ARM RDI debugger interface specification.

Note: Except where explicitly stated to the contrary, the term MAJIC refers to all models in the MAJIC Series of Intelligent Debug Probes.

What is RDI

The RDI (Remote Debug Interface) is an Application Programming Interface (API) that defines a standard set of data structures and functions that abstract debug services. Having a standard interface allows development tools (debuggers, debug kernels, ICEs, JTAG probes etc.) from different vendors to work together.

The RDI specification is provided by ARM Inc. The current version as of the writing of this document is RDI 1.5.1. See the Related Documentation below for details on other manuals.

The RDI API is implemented as a shared library (DLL file in Windows), called RDIMAJIC.DLL. RDIMAJIC.DLL supports both the RDI 1.5 and RDI 1.5.1 specifications.

Related Documentation

<i>RDI 1.5.1:</i>	ARM's RDI 1.5.1 document. ARM document Number: <i>RDI-0057-CUST-ESPC-A</i> . Dated: 24 January, 2001
<i>RDI 1.5.1tx and RDI 1.5.1rt:</i>	Documents RDI 1.5.1 additions for trace and real-time debug. ARM document Number: <i>RDI-0032-CUST-ESPC-A</i> . Dated: 19 May, 2000
<i>RDI 1.5 Terms and Abbreviations:</i>	Documents RDI definitions.. ARM document Number: <i>RDI-0005-CUST-ESPC-B</i> . Dated: 11 November 1998
<i>RDI 1.5:</i>	ARM's RDI 1.5 document. ARM document Number: <i>RDI-0003-CUST-ESPC-B</i> . Dated: 11 November 1998

Notational Conventions

The following conventions are used in the syntax descriptions of this manual.

Bold face	Bold identifies characters that must be entered exactly as shown.
<i>Italic</i>	Indicates a general category of input described in detail in the command operand's section.
[]	Square brackets enclose an optional operand or group of operands. The brackets are not entered in the command.
{ }	Curly braces are used for grouping purposes. These are not entered in the command. They either enclose a list of alternatives, one of which must be chosen, or they enclose a group of operands that are to be taken together in the context of a list of alternatives or a subsequent repetition.
...	Ellipsis (three dots in succession) indicate a preceding operand, or group of operands if enclosed by [] or { }, may optionally be repeated one or more times.
	A vertical bar indicates an operand, or group of operands if enclosed by [] or { }, on either side of the bar may be entered, but not both.
<i>WHITESPACE</i>	This symbol indicates one or more space or tab characters.

2 *Getting Started*

This chapter explains the steps to configure the RDI/MAJIC interface library to work with an RDI complaint debugger. There are three basic steps:

1. Install the EPI Development Tools (EDT) package, and your RDI compliant debugger. This manual assumes that you have already installed both packages.
2. Run the MAJIC Setup Wizard to configure the RDI environment.
3. Run your debugger and select the RDI/MAJIC environment.

Note: Please see Chapter 2 of the *MAJIC User's Manual* for information on setting up the MAJIC hardware. Additional details on the configuration process and advanced configuration options are covered in Chapter 3 of the *MAJIC User's Manual*.

Configuring the RDI Environment

The EPI Development Tools (EDT) for ARM includes the RDI/MAJIC interface library software that provides the connection from your debugger to the MAJIC. It also includes sample start up files for standard reference boards, and on-line documentation. So the first step is to install the EDT package on your computer.

RDI/MAJIC Configuration

The easiest way to configure the RDI environment is with the MAJIC Setup Wizard. It will guide you through a few forms to construct an *RDI/MAJIC Configuration*, and save that for subsequent use by your debugger. An RDI/MAJIC Configuration consists of the following files:

`rdimajic.dll`

The RDI/MAJIC library that will be loaded by your third party debugger to provide the connection to your MAJIC. This file will be copied to the RDI/MAJIC Configuration directory that you select in the MAJIC Setup Wizard.

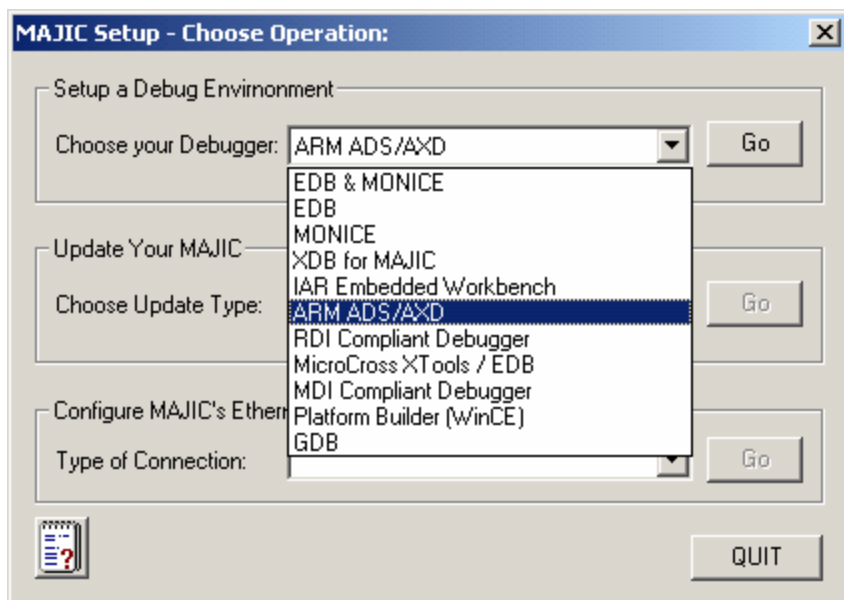
rdimajic.cfg The MAJIC Setup Wizard creates a **rdimajic.cfg** file specific to your configuration (MAJIC and start up file), and saves that in the RDI/MAJIC Configuration directory that you select in the MAJIC Setup Wizard. This file contains configuration information required by **rdimajic.dll**.

A sample **rdimajic.cfg** file showing how you can combine multiple RDI/MAJIC configurations in a single configuration file is provided in the **rdi** folder of your EDT software installation. Advanced configuration files are discussed on page 17.

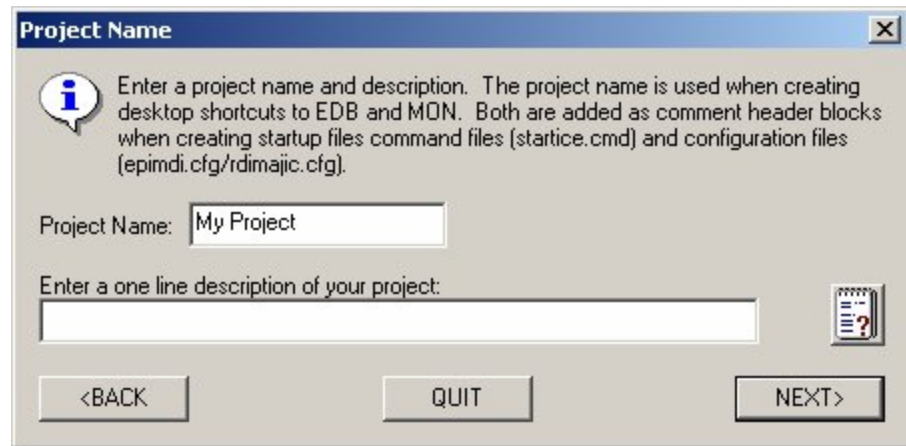
startice.cmd MAJIC and target hardware configuration information is provided by the **startice.cmd** command file. Sample startup command files for standard reference platforms are included in the EDT software installation. If you are using one of these boards or something similar, you can select the appropriate sample in the MAJIC Setup Wizard (all ***.cmd** files in the selected directory are copied to the RDI/MAJIC configuration directory that you select). If no suitable **startice.cmd** file is available, the wizard can create one.

Using the MAJIC Setup Wizard

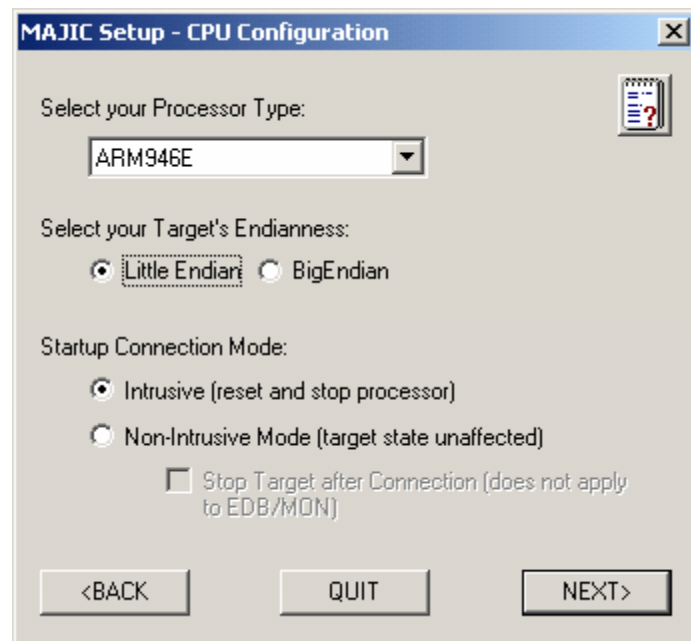
1. Use the Windows **[Start]** button and run the wizard from the **Programs/EPI Tools** folder. Read the introduction form and click **[Next]** to enter the wizard.



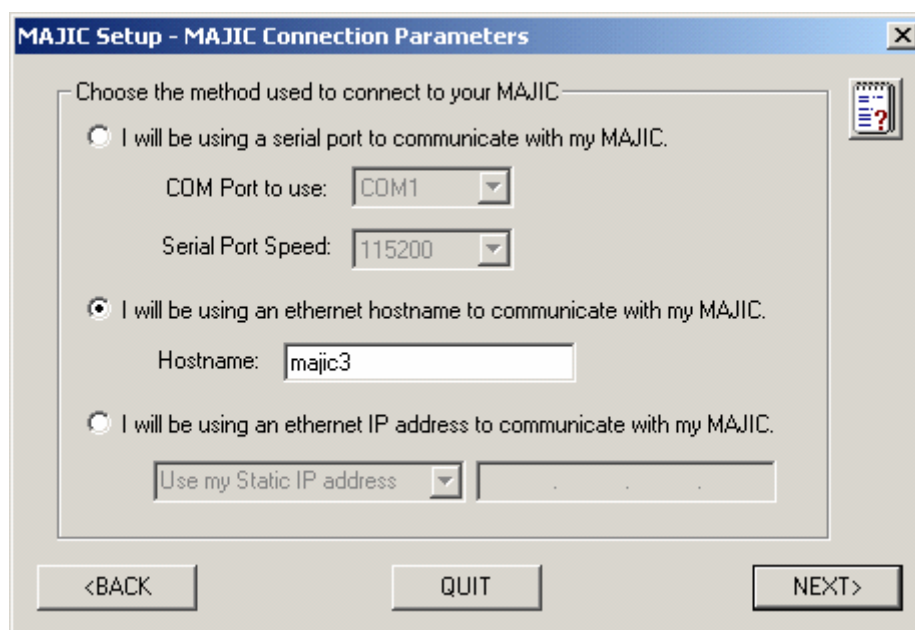
2. If you will be using AXD from the ARM ADS package (available from EPI or ARM), then choose **ARM ADS/AXD** for your debugger. Otherwise, choose **RDI Compliant Debugger**. Click **[Go]** to begin the configuration process.



3. Enter a project name and optional description, then click **[Next]**.

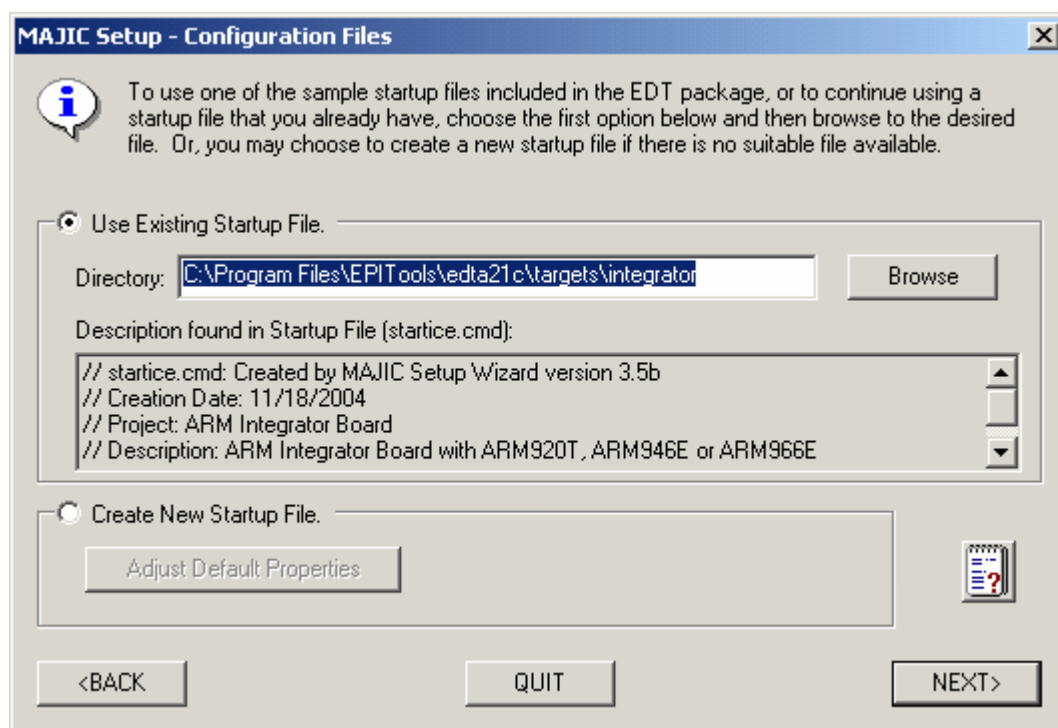


4. Select the processor type from the drop down list, make sure the target endian setting is right for your board, choose which connection mode you want, and click **[Next]**.



5. Select the communication channel; either serial using a COM port, Ethernet using a host name, or Ethernet with an IP address. Then click **[Next]**.

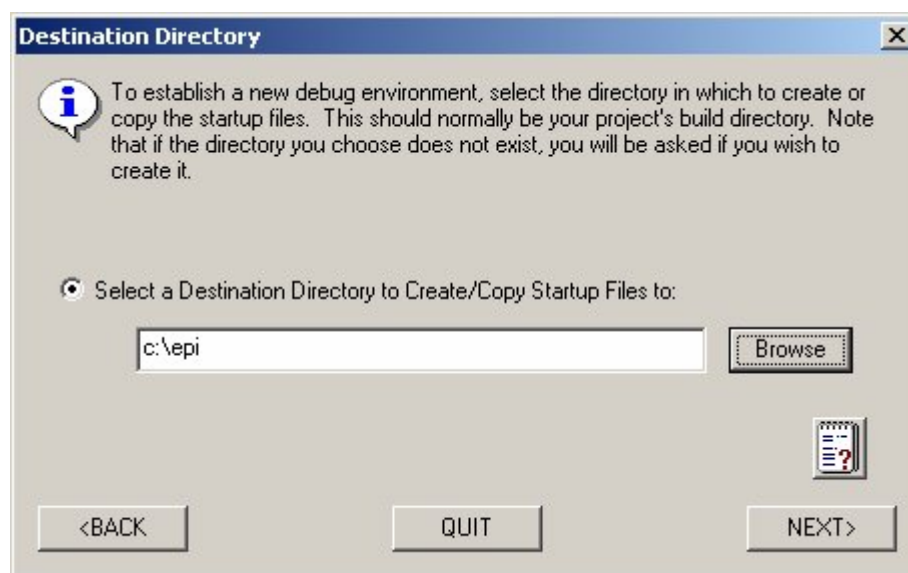
Note: See Chapter 2 of the *MAJIC User's Manual* for details on setting up a network connection.



6. The EDT package includes sample start up files for many standard reference boards. If you will be using one of these boards or something similar, or you already have a start up file, choose **Use Existing Startup File** and use the **[Browse]** button to find to it.

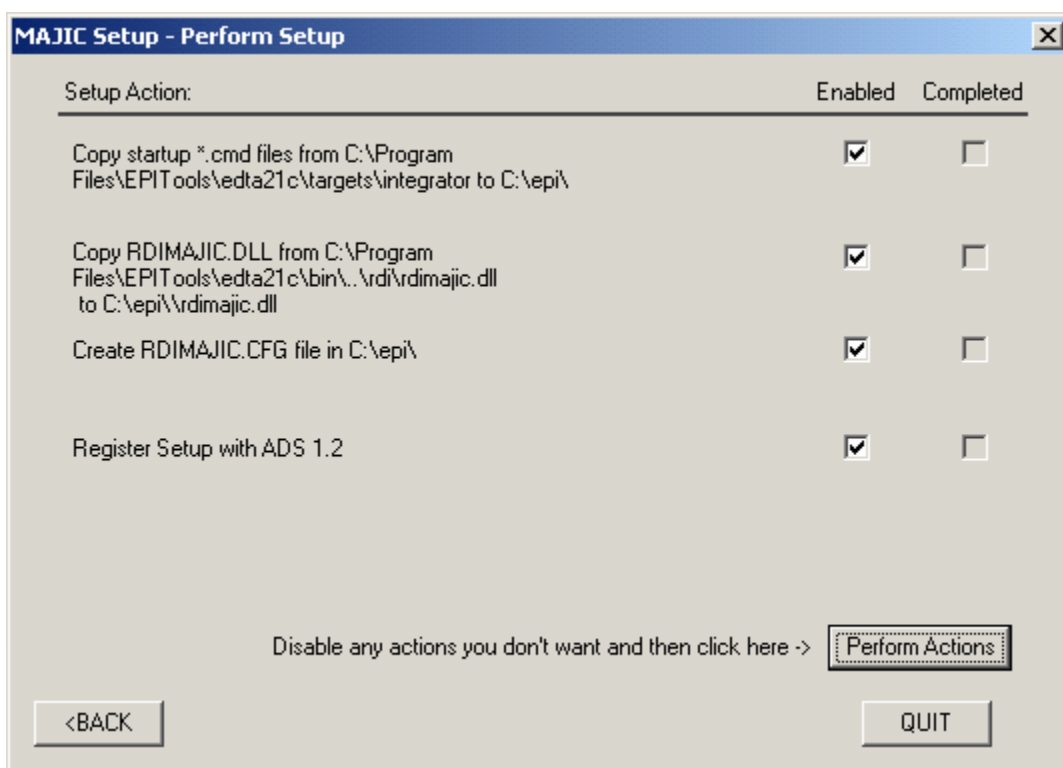
Or, if no suitable start up file exists, you can choose **Create New Startup File**. The default properties are based on the reference board for the CPU you selected, but they may be adjusted to accommodate differences on your board.

Note: If you choose **Use Existing Startup File** the endian selection in the selected file will override the endian selection in step 4. If you choose **Create New Startup File** then the endian selection in step 4 will be saved in the newly created file.



7. Choose the directory where the start up files and RDI interface library should be placed. If you frequently change between dissimilar targets or multiple MAJICs, then you may want to put each configuration in a different subdirectory so you won't have to rerun the wizard each time.

Note: AXD allows you to put the files anywhere, but ADW requires that you use a directory that is included in your **PATH** environment. For other debuggers you should check their documentation for their requirements are.



- View the list of actions, which may vary depending on how you answered certain questions. Then click **[Perform Actions]** to complete the RDI configuration process. Note that the Register Setup with ADS 1.2 action is only available if you have ADS 1.2 installed.

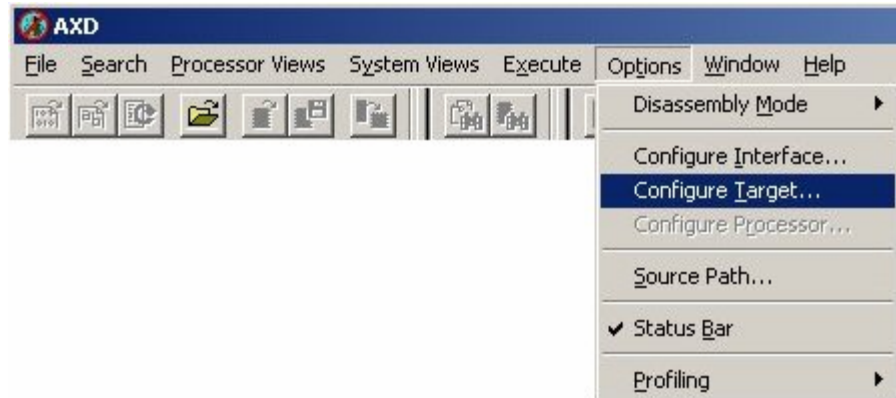
Debugger Configuration

Once you have established the MAJIC/RDI configuration, you are ready to start your debugger and select it. Every RDI compliant debugger does this differently, so check your debugger documentation for details. The following sections provide examples for AXD, ADW, and Greenhills Multi-2000.

ARM ADS/AXD

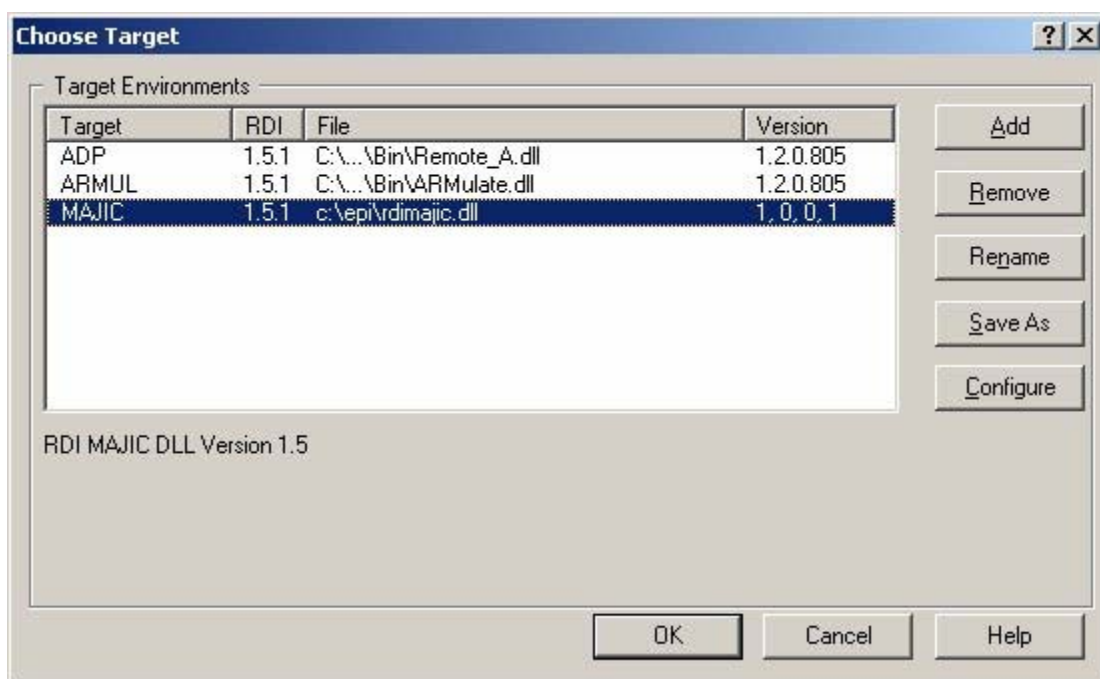
If you selected **ARM ADS/AXD** when you ran the MAJIC Setup Wizard, then it will have registered your MAJIC/RDI configuration as a *Target Environment* within AXD. However, the wizard does not automatically select your MAJIC/RDI configuration. This section explains how to set the AXD Target Environment to your MAJIC/RDI configuration.

- Start AXD. It may automatically reconnect to the target environment that you last used (the default is ARMulate mode), or it may open the **Choose Target** form. If it does automatically reconnect to your previous target, then use the **Options/Configure Target...** menu to open the **Choose Target** form.



Note: If AXD tries to reconnect to your previous target environment but that target is not available, AXD will report connection errors and then give you a chance to open the **Choose Target** form.

Note: If you frequently switch between multiple target environments, we recommend that you set AXD to automatically open the **Choose Target** form. That will allow you to select the right one at the start of each debug session. To do this, open the **Configure Interface...** form, go to the **Session File** tab and disable the **Reselect Target** checkbox.



2. Select MAJIC as the Target Environment, then click the **[OK]** button. AXD now attempts to load **rdimajic.dll** and connect to your MAJIC. If there are any problems in the connection process, be sure to look in the RDI Console Window (see page 16) for details.

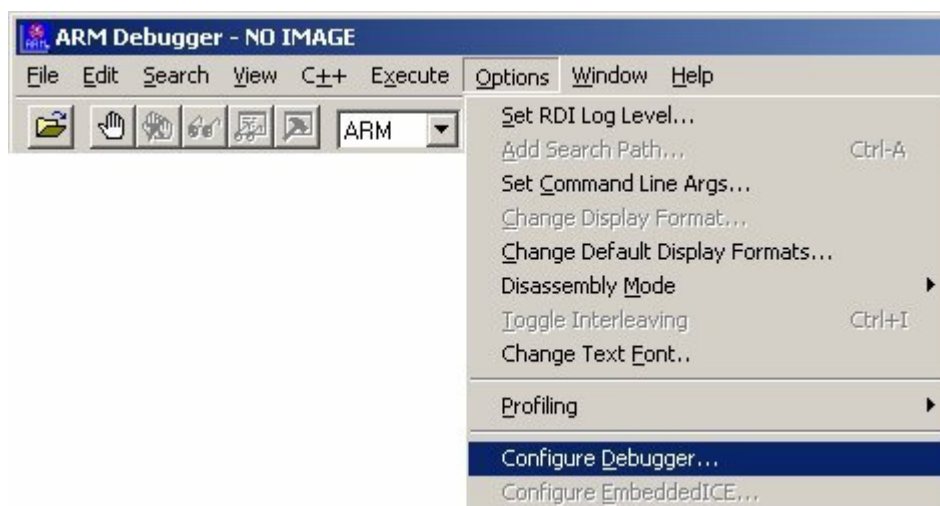
Note:

If you have multiple MAJIC probes or multiple dissimilar target boards, then you may want to create separate Target Environments for each. To do this, use the MAJIC Setup Wizard to create the first environment, then use the **[Save As]** button in AXD's **Configure Target** form save the **MAJIC** environment as a new name. Then run the MAJIC Setup Wizard to create each additional environment, in different directories, and save each new **MAJIC** environment as a unique name in AXD's **Configure Target** form

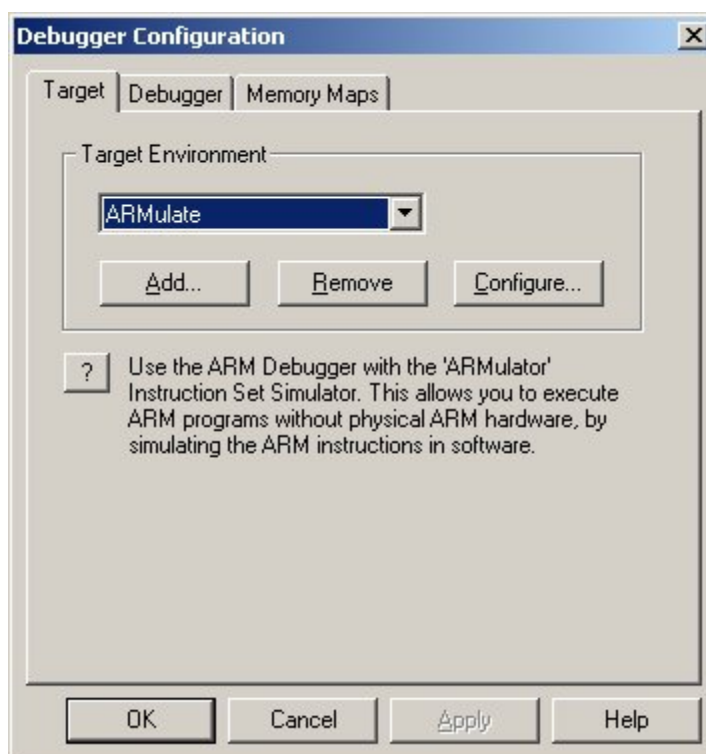
ARM ADW

The MAJIC Setup Wizard cannot register your MAJIC/RDI configuration in ADW, you must add MAJIC/RDI as a *Target Environment* from within ADW itself. This section explains the procedure.

1. Start ADW. It will automatically reconnect to the target environment that you last used (the default is ARMulate mode). Use the **Options/Configure Debugger...** menu to open the **Debugger Configuration** form.



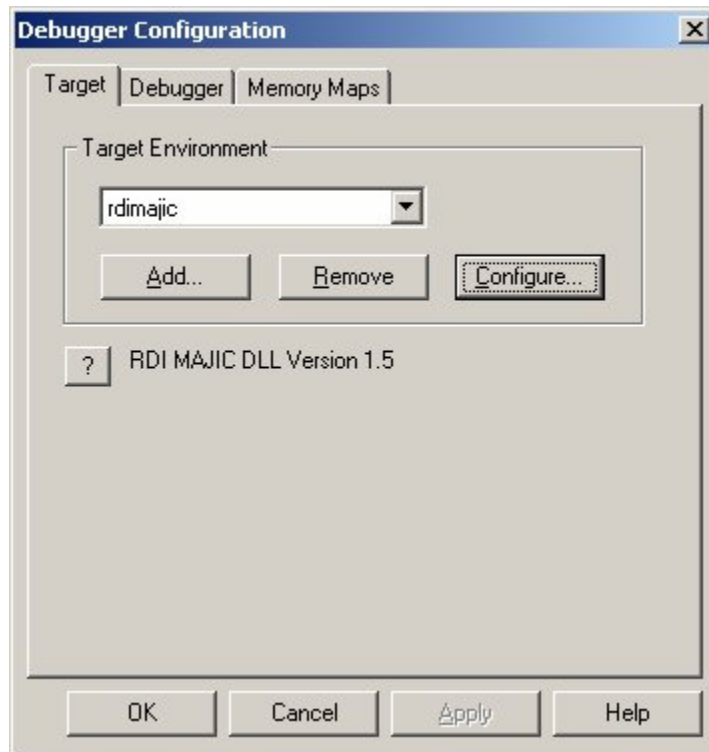
Note: If ADW tries to reconnect to your previous target environment but that target is not available, ADW will report connection errors and then give you a chance to open the **Debugger Configuration** form.



- Click the **[Add...]** button and browse to the directory that you told the MAJIC Setup Wizard to use for your RDI/MAJIC configuration (see page 8). Select the **rdimajic.dll** in this directory.

Note: ADW requires that the directory you select for your RDI/MAJIC configuration be included on your PATH environment variable.

The **rdimajic.dll** interface library should now be selected as the Target Environment.



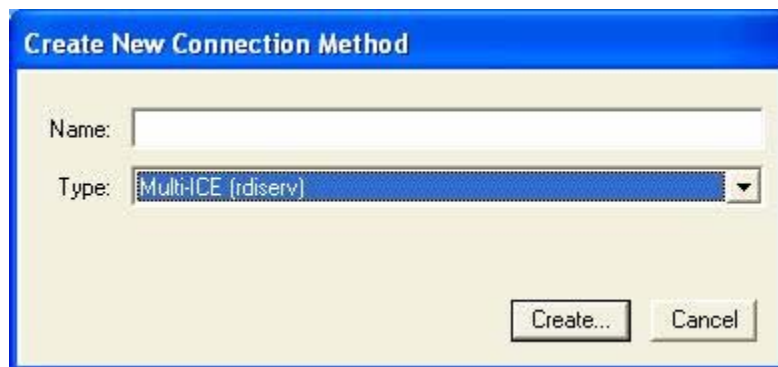
3. Click **[OK]** to complete the configuration process. ADW now attempts to load **rdimajic.dll** and connect to the MAJIC. If there are any problems in the connection process, be sure to look in the RDI Console Window (see page 16) for details.

GreenHill's Multi

Multi uses RDI 1.5 to connect to the MAJIC probe via EPI's RDI library named **rdimajic.dll**. The MAJIC Setup Wizard cannot automatically register your MAJIC/RDI configuration in Multi, but you can specify the RDI library when you create your connection within Multi. The procedure for this follows below:

1. Use the MAJIC Setup Wizard to define your RDI/MAJIC configuration (see page 4), but use the **RDI Compliant Debugger** option. When it asks for the destination directory for your RDI/MAJIC configuration, choose a project specific directory such as **C:\epi\MyProject** so that you can create different configurations for different projects.

2. Start Multi and create a new connection method, making sure to select *Multi-ICE (rdiserv)* as the type.



- Directly reference in the EPI RDI library by clicking on the **Use RDI DLL** checkbox from within the **Advanced** tab of the Multi connection editor. Then browse to the **rdimajic.dll** from the destination directory that you told the MAJIC Setup Wizard to use for your RDI/MAJIC configuration in step 1, above.

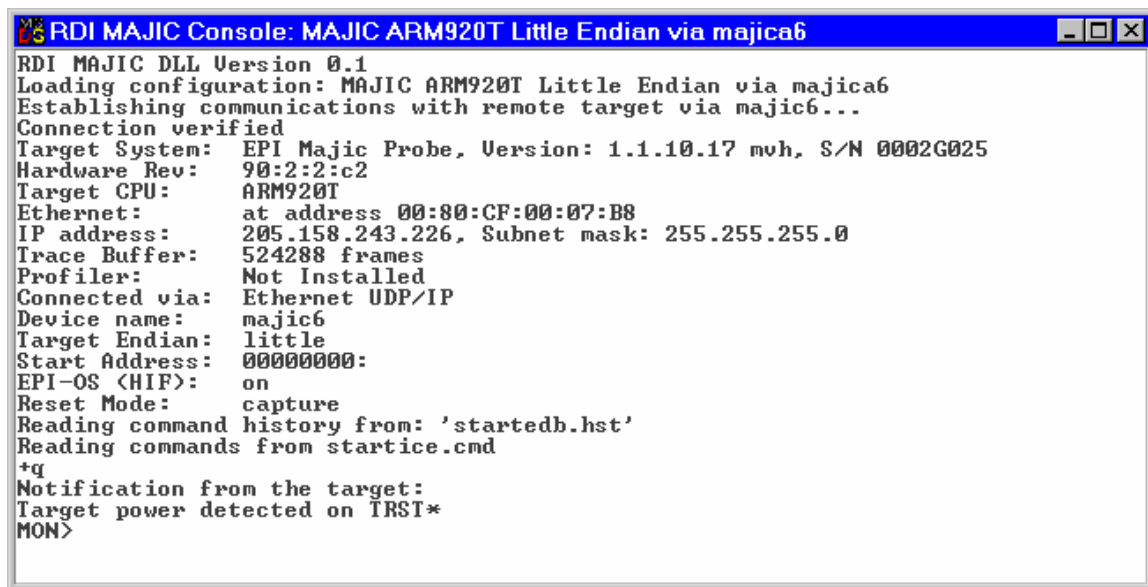
Note:

The **Attach (Debug application already on target)** radio button option is for non-intrusive JTAG connection while the other two options are for intrusive JTAG connection. Make sure that this matches what you selected in the MAJIC Setup Wizard (see page 5).

- Finish editing the connection, if necessary, and click **[OK]**. Multi will attempt to load the **rdimajic.dll** and connect to the MAJIC probe after you click on the **[Connect]** button. If there are any problems in the connection process, be sure to look in the RDI Console Window (see page 16) for details.

RDI MAJIC Console Window

Once you have established a successful connection, this window automatically minimizes itself, as users do not typically need access to this window during their debug session. An example of this window is shown below:



```

RDI MAJIC Console: MAJIC ARM920T Little Endian via majica6
RDI MAJIC DLL Version 0.1
Loading configuration: MAJIC ARM920T Little Endian via majica6
Establishing communications with remote target via majica6...
Connection verified
Target System: EPI Majic Probe, Version: 1.1.10.17 mvh, S/N 0002G025
Hardware Rev: 90:2:2:c2
Target CPU: ARM920T
Ethernet: at address 00:80:CF:00:07:B8
IP address: 205.158.243.226, Subnet mask: 255.255.255.0
Trace Buffer: 524288 frames
Profiler: Not Installed
Connected via: Ethernet UDP/IP
Device name: majica6
Target Endian: little
Start Address: 00000000:
EPI-OS (HIF): on
Reset Mode: capture
Reading command history from: 'startedb.hst'
Reading commands from startice.cmd
+q
Notification from the target:
Target power detected on TRST*
MON>
  
```

The RDI Console Window is similar to running MONICE in a console window (DOS box). It has a MON prompt, and many of MON commands can be used at this prompt, allowing you to check and set the state of the MAJIC or target system, read and write target resources, and display trace data. On-line help is available with the **H** command. Please refer to the *MAJIC User's Manual* for full information on the MON Command Language.

Notes:

- Breakpoint and execution control commands are not available in the RDI Console Window. These types of operations should be done with the third-party debugger which launched RDI.
- Commands can only be entered and executed when the controlling debugger is not itself attempting to access the target.
- Some output operations may cause the MON prompt to disappear at times. If you don't see the prompt and want to type in a command, try hitting the **[Enter]** key first.

3 *Advanced Configuration*

The RDI specification requires the debugger to query the RDIMAJIC library for a list of “devices” that are available, and to give the user a way to select which device to connect to. In the case of the RDI MAJIC Interface, there is also the possibility that there is more than one physical MAJIC probe available to connect to. The RDI configuration file (**`rdimajic.cfg`**) provides the necessary device and probe identification information to the RDIMAJIC library, so that it can respond to the debugger’s query request.

Note: The MAJIC Setup Wizard automatically creates an RDI configuration file describing one configuration. This chapter is for advanced users who want to create a custom configuration file describing multiple configurations.

Configuration File

When the debugger first connects to the RDIMAJIC library, the library will try to open the **`rdimajic.cfg`** file by looking first in the current working directory, then in the directory containing the library, then in the directories specified in your **`PATH`** environment variable. It is usually most convenient to have a single **`rdimajic.cfg`** file located in the same directory as the library. However, if you are using the same installation tree to run multiple connections at the same time, then it can be useful to configure a different default device via different configuration files in different directories.

Contents

The RDIMAJIC configuration file is an ASCII file containing comments (C++ style) and keyword-value pairs. Keywords are not case-sensitive. There are two types of values, strings and numbers. Strings must be enclosed in double quotes (“”), unless they are valid identifiers (start with alphabetic or underscore, and contain only alphanumerics and underscore). Numbers and quoted strings conform to C/C++ language syntax.

The sample configuration file uses line breaks and indentation to improve readability, but this is not required. As long as there is at least one

whitespace character (space, tab, or line break) between all keywords and values the file will be processed correctly.

Organization

The configuration file is organized into sections, where each section defines Global settings, a “Device”, a “Controller”, or the “RDIDeviceList”. Conventionally, the Global section is first, followed by all the Device sections, followed by Controller sections and finally the MDIDeviceList section. But the only requirement is that the RDIDeviceList section follow all the Device and Controller sections.

Global

The **Global** section defines values for configuration settings that are not associated with a particular target device or connection. This section is optional, but it is strongly recommended to include it and specify the **EDTPath** value. The syntax is:

Define Global

EDTPath	<i>EDTPathString</i>
LogFile	<i>LogPathString</i>
CommandFile	<i>CmdPathString</i>

EDTPathString is the full path name of the directory into which your EDT package was installed. It is used to locate various files needed by the MDILib without having to add EPI specific directories to the PATH environment variable.

LogPathString is the full or relative path name of a file which will have detailed MDI session log information and console output written to it. This entry should only be used when requested by EPI Technical Support.

CmdPathString is the full or relative path name of a default startup command file that will be processed automatically when a device is opened which does not specify a more specific **CommandFile** value in its **MDIDeviceList**, **Device**, or **Controller** definitions.

The **CommandFile** entry is optional in every definition section, but at least one applicable value is required for every *DevNameString* defined in the **MDIDeviceList** section. If multiple values are available, the order of precedence is the value specified in the *DevNameString* definition itself, followed by the value specified in the referenced **Device** definition, followed by the referenced **Controller** definition, followed by the **Global** definition.

CommandFile is normally specified in the **Device** definition, since there are commonly different configuration commands needed for different devices. Specifying a default in the **Global** definition is useful if you have several devices sharing a common startup command file.

Device

Each **Device** section defines a target device and includes the CPU type, memory organization, and optionally the startup command file. If you have multiple targets with different CPU types or memory organizations, you will use multiple **Device** sections to define them. The syntax is:

```
Define Device DevIDString
    Endian          { Big | Little }
    EPICpuId        CPUString
    CommandFile     PathString
```

DevIDString is an arbitrary string value used to identify the **Device** section when it is referenced from the **RDIDeviceList** section.

CPUString is a string value giving the EPI-specific CPU type identifier. This is the same value specified by the **-v** command line option when running an EPI debugger (see the MAJIC User's Manual for more information). The **Endian** value specifies the target system's memory organization (big-endian or little-endian).

CmdPathString is the full or relative path name of a startup command file that will be processed automatically when a device is opened which does not specify a more specific **CommandFile** value in its **RDIDeviceList** definition. If **CommandFile** is specified here, the value overrides values specified in the **Global** and **Controller** definitions.

Controller

Each **Controller** section defines a particular MAJIC to connect to and includes the communication port (serial port or Ethernet host name), baud rate, and optionally the startup command file to load for initialization. If you have more than one MAJIC probe, or more than one startup initialization file, you will use multiple **Controller** sections to define them. The syntax is:

```
Define Controller ControllerIDString
    Port           PortString
    Speed          SpeedNumber
    CommandFile    CmdPathString
```

ControllerIDString is an arbitrary string value used to identify the **Controller** section when it is referenced from the **RDIDeviceList** section. *PortString* is a string value giving the serial port or Ethernet hostname to use to connect to the MAJIC probe. This is the same value specified by the **-d** command line option when running an EPI debugger (see the MAJIC User's Manual for more information). *SpeedNumber* is the EPI-specific baud rate value to use when **PortString** specifies a serial interface. Valid values are 0 (1200 baud) through 7 (115,200 baud).

CmdPathString is the full or relative path name of a startup command file that will be processed automatically when a device is opened which does not specify a more specific **CommandFile** value in its **RDIDeviceList** or **Device** definition. If **CommandFile** is specified here, the value overrides a value specified in the **Global** definition.

RDIDeviceList

The last section in the configuration file is the **RDIDeviceList** section. It defines the RDIMAJIC device name strings that the debugger will display to the user, and associates each name with a specific **Device** and **Controller** section. The syntax is:

```
Define RDIDeviceList
{   DevNameString
    [ DefaultDevice ]
    Device      DevIDString
    Controller  ControllerIDString
    CommandFile CmdPathString
} ....
```

If you have more than one **Device** or **Controller** section, the **RDIDeviceList** section will contain multiple sets of *DevNameString* + **Device** + **Controller** entries to list and name all the valid combinations of **Device** and **Controller**.

DevNameString is a string value to be passed to the debugger to identify a particular “RDI Device” that the debugger can connect to. If there is only one *DevNameString* entry in the **RDIDeviceList** section, the debugger may just automatically open the device. Otherwise, the debugger should present you with a list of the *DevNameString* values and let you select which one to connect to.

DefaultDevice identifies a particular *DevNameString* as the active device. Note that some debuggers may not provide a configurable user interface for selecting a device name. If so, then editing this configuration file is the only way to change the selected target device.

CmdPathString is the full or relative path name of a startup command file that will be processed automatically when the device is opened. If **CommandFile** is specified here, the value overrides values specified in the **Global**, **Controller**, and **Device** definitions.

Startup Command File

The MAJIC has a number of configuration options that must be set correctly for the target system. These options are set by debugger commands, and EPI debuggers automatically look for a command file named **startice.cmd** at start up. The EDT software package includes a library of sample start up files for standard reference boards, and you can create a custom start up file with the MAJIC Setup Wizard.

Note: Refer to the *MAJIC User's Guide* for information on customizing the startup command file for your target.

When using a third-party debugger with the RDIMAJIC library, the name of the startup command file is specified in the RDIMAJIC configuration file. A full path name can be provided, or a relative path name, or just the file name. If the full path is not specified, the RDIMAJIC library will try to open the file by looking first relative to the current working directory, then the directory containing the library, then the `bin` subdirectory of the EDT installation directory if **EDTPath** is specified, then the directories specified in your **PATH** environment variable. The directory containing the startup command file will also added to the list of directories to search for other files.

Note: The RDI configuration file created by the wizard always uses **startice.cmd** as the start up command file name. But if one configuration file needs to describe dissimilar targets, then separate start up command files are required, and your custom configuration file will need to reference the different file or path names.