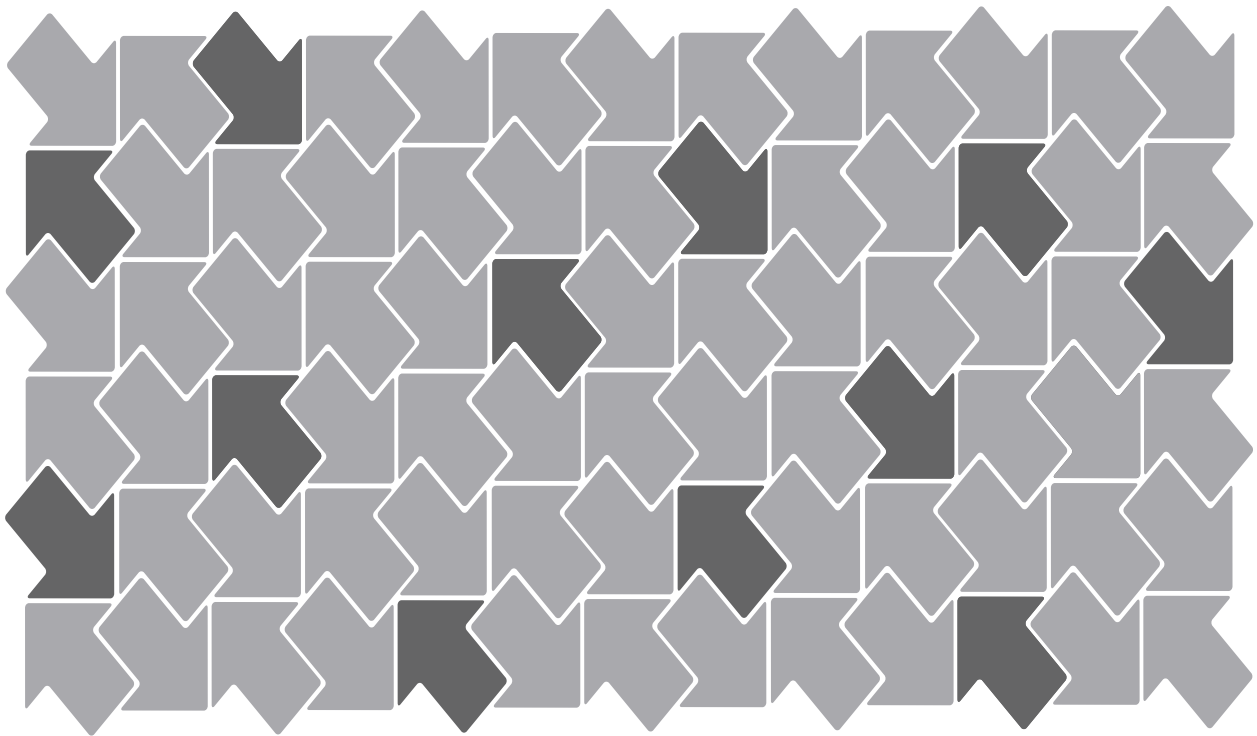


Eclipse Integrated Virtual Debugger User's Manual

Workstation 6.0



Eclipse Integrated Virtual Debugger User's Manual

Revision: 20070426

Item: WS-ENG-Q107-297

You can find the most up-to-date technical documentation on our Web site at

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

The VMware Web site also provides the latest product updates.

If you have comments about this documentation, submit your feedback to:

docfeedback@vmware.com

© 2007 VMware, Inc. All rights reserved. Protected by one or more of U.S. Patent Nos. 6,397,242, 6,496,847, 6,704,925, 6,711,672, 6,725,289, 6,735,601, 6,785,866, 6,789,156, 6,795,966, 6,880,022, 6,944,699, 6,961,806, 6,961,941, 7,069,413, 7,082,598, 7,089,377, 7,111,086, 7,111,145 and 7,117,481; patents pending.

VMware, the VMware "boxes" logo and design, Virtual SMP and VMotion are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

VMware, Inc.
3145 Porter Drive
Palo Alto, CA 94304
www.vmware.com

Contents

Preface 5

Overview of the Eclipse Integrated Virtual Debugger 7

Setting Up the Eclipse Integrated Virtual Debugger Environment 9

Host System Requirements 9

Supported Host Operating Systems 9

Supported JRE Versions 10

Eclipse Requirements 10

Virtual Machine Requirements 10

Supported Guest Operating Systems 10

VMware Tools Requirements 11

Java and JRE Requirements 11

Installing PSAPI.DLL on Windows NT 11

Disabling the Firewall on Linux Guest Systems 11

Configuring the Firewall on Windows Guest Systems 11

Using the Eclipse Integrated Virtual Debugger 13

Managing Virtual Machine Launch Configurations 13

Using Application Configurations to Start Applications in a Virtual Machine 13

Using Application Configurations to Attach to Applications Running in a Virtual Machine 14

Deleting Configurations 15

Starting an Application Debugging Session in a Virtual Machine 15

Starting an Application in a Virtual Machine Without Debugging 15

Attaching the Debugger to an Application Running in a Virtual Machine 15

Index 17

Preface

This preface provides information about the *Eclipse Integrated Virtual Debugger User's Manual* and links to VMware® technical support and educational resources.

This preface contains the following topics:

- [“About This Book”](#) on page 5
- [“Technical Support and Education Resources”](#) on page 6

About This Book

This manual, the *Eclipse Integrated Virtual Debugger User's Manual*, provides information about setting up and using the Eclipse Integrated Virtual Debugger.

Revision History

This manual is revised with each release of the product or when necessary. A revised version can contain minor or major changes.

Table P-1.Revision History

Release	Description
20070412	First version of the Eclipse Integrated Virtual Debugger documentation.

To view the most current version of the manual, see the VMware Web site:

http://www.vmware.com/products/beta/ws/ws_eclid.pdf

Intended Audience

This book is intended for anyone who needs to set up or use the Eclipse Integrated Virtual Debugger. The integrated virtual debugger is intended primarily for people who develop, test, and debug Java applications in multiple operating systems or computing environments.

Document Feedback

If you have comments about this documentation, submit your feedback to:

docfeedback@vmware.com

Conventions

Table P-2 illustrates the typographic conventions used in this manual.

Table P-2.Conventions Used in This Manual

Style	Elements
Blue (online only)	Cross-references and email addresses
Blue boldface (online only)	Links
Black boldface	User interface elements such as button names and menu items
Monospace	Commands, filenames, directories, and paths
Monospace bold	User input

Table P-2.Conventions Used in This Manual (Continued)

Style	Elements
<i>Italic</i>	Document titles, glossary terms, and occasional emphasis
<Name >	Variable and parameter names

Technical Support and Education Resources

The following sections describe the technical support resources available to you.

Self-Service Support

Use the VMware Technology Network (VMTN) for self-help tools and technical information:

- Product information – <http://www.vmware.com/products/>
- Technology information – <http://www.vmware.com/vcommunity/technology>
- Documentation – <http://www.vmware.com/support/pubs>
- VMTN Knowledge Base – <http://www.vmware.com/support/kb>
- Discussion forums – <http://www.vmware.com/community>
- User groups – <http://www.vmware.com/vcommunity/usergroups.html>

For more information about the VMware Technology Network, go to <http://www.vmtn.net>.

Online and Telephone Support

Use online support to submit technical support requests, view your product and contract information, and register your products. Go to <http://www.vmware.com/support>.

Customers with appropriate support contracts should use telephone support for the fastest response on priority 1 issues. Go to http://www.vmware.com/support/phone_support.html.

Support Offerings

Find out how VMware support offerings can help meet your business needs. Go to <http://www.vmware.com/support/services>.

VMware Education Services

VMware courses offer extensive hands-on labs, case study examples, and course materials designed to be used as on-the-job reference tools. For more information about VMware Education Services, go to <http://mylearn1.vmware.com/mgrreg/index.cfm>.

CHAPTER 1 Overview of the Eclipse Integrated Virtual Debugger

The Eclipse Integrated Virtual Debugger provides a configurable interface between Eclipse and virtual machines, making it easy to develop and debug applications that run in multiple operating system environments on a single PC. Debugging your applications in virtual machines enables you to reproduce and record errors while maintaining the integrity of the host machine. You can perform typical debugging tasks such as pausing at breakpoints, stepping through code, and viewing and modifying the state of your application, all without impacting the host environment.

You can manage configuration settings for each virtual machine in which you want to execute or debug applications. Eclipse Integrated Virtual Debugger launch configurations determine which virtual machine to run the application in and how the application is executed. Using these configurations, you can:

- Start an application debugging session in a virtual machine.
- Start an application in a virtual machine without debugging.
- Start a debugging session that attaches to a process already running in a virtual machine.

Once configured, the integrated virtual debugger finds the virtual machine, powers it on if necessary, sets up the environment based on your configuration settings, and starts or attaches to the application.

To configure how an application is started in a virtual machine, you can specify:

- The name of the virtual machine (.vmx configuration file).
- (Optional) The path to the JVM on the guest system.
- (Optional) The location of folders to be shared between the host and the guest.
- (Optional) Actions to perform before launching an application from Eclipse, including:
 - Revert to the most recent snapshot.
 - Run specified pre-execution commands.
- (Optional) Actions to perform after an application launched from Eclipse is terminated, including:
 - Run specified post-execution commands (for example, to perform cleanup tasks).
 - Set the virtual machine state to:
 - Suspended (default)
 - Revert to the most recent snapshot
 - Powered off

For information on installation requirements for the Eclipse Integrated Virtual Debugger, see [Chapter 2, “Setting Up the Eclipse Integrated Virtual Debugger Environment,”](#) on page 9.

For information on how to manage virtual machine configurations and how to start debugging sessions in virtual machines, see [Chapter 3, “Using the Eclipse Integrated Virtual Debugger,”](#) on page 13.

CHAPTER 2 **Setting Up the Eclipse Integrated Virtual Debugger Environment**

Review the requirements and recommendations in this chapter before following the instructions in the *Workstation User's Manual* to install the Eclipse Integrated Virtual Debugger as an optional component of Workstation 6. You can install the Eclipse Integrated Virtual Debugger on any supported Windows or Linux host system that is running Workstation 6 and has a supported version of Eclipse and the Java Runtime Environment (JRE) installed.

During Windows installation, if Eclipse is not installed in `C:\Program Files\Eclipse` or `C:\Eclipse`, you must use the **Custom** setup to select the Eclipse Integrated Virtual Debugger component and specify the Eclipse directory location.

During Linux installation, you must override the default value of **No** when prompted by `vmware-config.pl` to install the Eclipse Integrated Virtual Debugger.

When you install the Eclipse Integrated Virtual Debugger:

- The integrated virtual debugger plug-in, `ivd.jar`, Foundry Java bindings, and the `plugin.xml` launch configuration file are placed in the `com.vmware.bfg_1.0.0` subdirectory of the Eclipse plug-in directory.
- After you restart Eclipse, the **Debug** menu includes the new launch configuration types **VMware attach to application** and **VMware execute Java application**. These launch configuration types have a VMware tab that enables you to configure virtual machine settings.

You can debug in multiple virtual machines simultaneously. You can also debug multiple sessions in a single virtual machine.

This chapter contains the following sections:

- [“Host System Requirements”](#) on page 9
- [“Eclipse Requirements”](#) on page 10
- [“Virtual Machine Requirements”](#) on page 10

Host System Requirements

The Eclipse Integrated Virtual Debugger can run on any supported host operating system that is running Workstation 6 and has Eclipse installed. Eclipse must be running on the same system as Workstation 6. See [“Eclipse Requirements”](#) for additional Eclipse requirements.

Supported Host Operating Systems

The Eclipse Integrated Virtual Debugger supports the following Windows 32-bit host operating systems:

- Windows Vista Enterprise Edition
Windows Vista Business Edition
Windows Vista Home Basic and Premium Editions
Windows Vista Ultimate Edition
- Windows XP Home Edition, SP1, SP2
Windows XP Professional, SP1, SP2
- Windows 2000 Server SP3, SP4
Windows 2000 Professional, SP3, SP4
Windows 2000 Advanced Server, SP3, SP4

NOTE Windows 64-bit host operating systems are not currently supported.

The Eclipse Integrated Virtual Debugger supports the following Linux 32-bit and 64-bit host operating systems:

- Red Hat Enterprise Linux WS 4.5 (Beta, formerly called 4.0 Update 5)
 - Red Hat Enterprise Linux AS 4.0, updates 1, 2, 3, 4
 - Red Hat Enterprise Linux ES 4.0, updates 1, 2, 3, 4
 - Red Hat Enterprise Linux WS 4.0, updates 1, 2, 3, 4

- Red Hat Linux 9.0 — stock 2.4.20-8, upgrade 2.4.20-20.9

- Ubuntu Linux 6.10
 - Ubuntu Linux 6.06

Supported JRE Versions

The host system must be running a Java Runtime Environment (JRE) meeting Java 2 Platform Standard Edition (J2SE) 5.0 or higher specifications. J2SE consists of the JRE and developer tools for compiling, debugging, and running applications written in the Java language.

NOTE Eclipse displays the error message `unable to load class` if an unsupported version of J2SE is being used on the host system.

Eclipse Requirements

You must have Eclipse 3.2 installed on the host. On Windows Vista hosts, you must have Eclipse 3.2.2 installed.

The Java language is supported. You cannot have GCJ Java installed on the guest operating system.

Review [“Virtual Machine Requirements”](#) for information about guest system requirements.

Virtual Machine Requirements

The Eclipse Integrated Virtual Debugger is supported on any Workstation 6 virtual machine that is running a supported Windows or Linux guest operating system.

Supported Guest Operating Systems

This section provides a simplified list of guest operating systems supported for debugging in virtual machines. For the most recent list of supported guest operating systems, including detailed information about the specific operating system versions, service packs, and updates supported, see the *VMware Guest Operating System Installation Guide*, at <http://pubs.vmware.com/guestnotes/>. This guide also provides notes on installing the most common guest operating systems.

Operating systems that are not listed are not supported for debugging in a virtual machine.

The following Windows 32-bit guest operating systems are supported:

- Windows Vista (all except Vista Home Edition, which cannot be run in a virtual machine due to Microsoft licensing restrictions)
- Windows Server 2003, Enterprise Edition and R2
- Windows XP Professional and Home Edition
- Windows 2000 Professional
 - Windows 2000 Server
 - Windows 2000 Advanced Server

The following Windows 64-bit guest operating systems are supported:

- Windows Vista x64 Edition (3-D effects not yet supported)
- Windows Server 2003 x64 Edition
- Windows XP Professional x64

The following Linux 32-bit and 64-bit guest operating systems are supported:

- Red Hat Linux 8 and 9
- Red Hat Enterprise Linux Advanced Server, Enterprise Server, and Workstation 4 and 5
- Ubuntu Linux 6.10
Ubuntu Linux 6.06
- SUSE Linux 10
- SUSE Linux Enterprise Server 10

VMware Tools Requirements

Make sure that the version of VMware Tools on the guest operating system matches the version of Workstation 6 (which the Eclipse Integrated Virtual Debugger is a component of) on the host.

Java and JRE Requirements

You cannot have GCJ Java installed on the guest operating system.

The guest operating system must be running JRE 1.4.2 or higher. If you are not using JRE 5.0 on the guest, you must update the build settings in Eclipse to be compatible with the older JRE.

To update the Eclipse build settings to use a 1.4.x JRE on the guest

- 1 In the Eclipse Package Explorer, right-click the topmost folder (Project item) and choose **Properties**.
- 2 In the left pane of the Properties page, select **Java Compiler**.
- 3 Select **Enable project specific settings**, and set the JDK Compliance Compiler compliance level to **1.4**.

Installing PSAPI.DLL on Windows NT

On Windows NT, you must install the `psapi.dll` library file to retrieve process status information so that the Eclipse Integrated Virtual Debugger can attach to a process. You can download `psapi.dll` from <http://www.microsoft.com/downloads/release.asp?releaseid=30337>.

Disabling the Firewall on Linux Guest Systems

You must disable the firewall on Linux guest operating systems. The Eclipse Integrated Virtual Debugger opens an available port (searching from port 49152) for each debugging session.

Configuring the Firewall on Windows Guest Systems

If you are using a 1.4.x JRE on Windows guest systems, you must either disable the firewall or allow incoming connections to the JVM. If you have a Windows system (such as Windows XP SP2, Windows 2003, and Windows Vista) that allows you to configure exceptions to the firewall, you can add the JVM to the exceptions list.

To add the JVM to the exceptions list

- 1 Choose Start > Control Panel > Windows Firewall and select the **Exceptions** tab.
- 2 Click **Add Program** and browse to the Java executable.
- 3 Click **OK**.

NOTE On Windows Vista guests, you might have to restart the firewall after configuring it to allow incoming connections to the JVM.

CHAPTER 3 Using the Eclipse Integrated Virtual Debugger

The Eclipse Integrated Virtual Debugger enables you to:

- Manage launch configuration settings for application execution and debugging in virtual machines.
- Start an application debugging session in a virtual machine.
- Start an application in a virtual machine without debugging.
- Start a debugging session that attaches to a process already running in a virtual machine.

Using Eclipse launch configurations, you can choose what virtual machine to run your application in and how it is executed.

This chapter contains the following sections:

- [“Managing Virtual Machine Launch Configurations”](#) on page 13
- [“Using Application Configurations to Start Applications in a Virtual Machine”](#) on page 13
- [“Using Application Configurations to Attach to Applications Running in a Virtual Machine”](#) on page 14
- [“Deleting Configurations”](#) on page 15
- [“Starting an Application Debugging Session in a Virtual Machine”](#) on page 15
- [“Starting an Application in a Virtual Machine Without Debugging”](#) on page 15
- [“Attaching the Debugger to an Application Running in a Virtual Machine”](#) on page 15

Managing Virtual Machine Launch Configurations

Launch configurations determine how Eclipse applications are executed. The launch configuration types **VMware attach to application** and **VMware execute Java application** have a **VMware** tab. The values you enter in the **VMware** tab determine virtual machine configuration settings. Once configured, you can start and attach to applications in virtual machines from the Eclipse **Debug** and **Run** menus.

Choose **Run > Debug** to manage launch configurations. You can create, duplicate, edit, and remove configurations.

Using Application Configurations to Start Applications in a Virtual Machine

This section describes the virtual machine settings you can configure to start an application in a virtual machine.

To create, duplicate, or edit a launch configuration to start an application in a virtual machine

- 1 Choose **Run > Debug**.

The **Debug** page is displayed. You can create, manage, and run configurations from this page.

- 2 You can create a launch configuration based on default settings or based on another configuration. You can also edit an existing configuration. Do one of the following:
 - Create a configuration based on default settings by selecting **VMware execute Java application** in the left pane, and clicking the **New launch configuration** icon at the top of the pane.

- Create a configuration based on another configuration by selecting the configuration you want to duplicate under **VMware execute Java application** in the left pane, and clicking the **Duplicates the currently selected configuration** icon at the top of the pane.
- Edit an existing configuration by selecting the configuration you want to edit under **VMware execute Java application** in the left pane.

Perform the remaining steps in the **VMware** tab of the right pane.

- 3 Choose a virtual machine from the drop-down menu of recently used and currently running virtual machines.
Click **Browse** to select from `.vmx` files on the system.
- 4 (Optional) If you want to use a JVM other than the one that is automatically selected, select an alternate JVM path.
- 5 (Optional) Expand the list of shared folders to add, edit, or remove folders to be shared between the host and the guest systems.
For each folder, enter the share name and the location on the host system.
By default, the project folder is shared.
- 6 (Optional) Indicate actions to be performed before the application is launched:
 - Select **Set virtual machine state to most recent snapshot** to revert to the most recent snapshot before the application is launched.
 - Select **Run script** and enter one or more shell commands to be executed in the guest operating system before the application is launched. No syntax checking is performed. Either enter one command per line, or enter multiple commands on the same line using a semicolon as a separator.
- 7 (Optional) Indicate actions to be performed after the application has terminated:
 - Select **Run script** and enter one or more shell commands to be executed in the guest operating system after the application has terminated. No syntax checking is performed. Either enter one command per line, or enter multiple commands on the same line using a semicolon as a separator.
 - Select **Set virtual machine state**, and select one of the following options:
 - **to suspended** (default)
 - **to most recent snapshot**
 - **to powered-off**
- 8 Click **Apply**.
If you click **Revert**, settings revert to previous values.
If newly created, the launch configuration is added to the left pane.

Using Application Configurations to Attach to Applications Running in a Virtual Machine

This section describes the virtual machine settings you can configure to attach to an application that is already running in a virtual machine.

To create, duplicate, or edit a configuration that attaches to a running application in a virtual machine

- 1 Choose **Run > Debug**.
The **Debug** page is displayed. You can create, manage, and run configurations from this page.
- 2 You can create a launch configuration based on default settings or based on another configuration. Do one of the following:
 - Create a configuration based on default settings by selecting **VMware attach to application** in the left pane, and clicking the **New launch configuration** icon at the top of the pane.

- Create a configuration based on another configuration by selecting the configuration you want to duplicate under **VMware attach to application** in the left pane and clicking the **Duplicates the currently selected configuration** icon at the top of the pane.
- Edit an existing configuration by selecting the configuration you want to edit under **VMware attach to application** in the left pane.

Perform the remaining steps in the **VMware** tab of the right pane.

- 3 Choose a virtual machine from the drop-down menu of recently used and currently running virtual machines.

Click **Browse** to select from `.vmx` files on the system.

- 4 Click **Apply**.

If you click **Revert**, settings revert to default values.

If newly created, the launch configuration is added to the left pane.

Deleting Configurations

You can remove an existing configuration.

To remove a configuration

- 1 Choose **Run > Debug**.

The **Debug** page is displayed. You can create, manage, and run configurations from this page.

- 2 In the left pane, select one or more configurations you want to delete and click the **Delete selected launch configuration(s)** icon at the top of the pane.

The configuration is removed in the left pane.

Starting an Application Debugging Session in a Virtual Machine

You can debug an application in any configured virtual machine.

To start a debugging session in a virtual machine

- 1 Begin the session in one of the following ways:

- From the **Debug** menu, choose the configuration for the application you want to start debugging.
- In the **Debug** page, select the configuration under **VMware execute Java application** in the left pane and click **Debug** in the right pane.

- 2 Perform debugging tasks as you would in a local debugging environment.

NOTE Do not suspend a virtual machine while the integrated virtual debugger is connected to an application. If you do, the integrated virtual debugger will disconnect from the application.

Starting an Application in a Virtual Machine Without Debugging

You can start an application without debugging in any configured virtual machine.

Begin the session in one of the following ways:

- From the **Run** menu, choose the configuration for the application you want to start.
- In the **Run** page, select the configuration under **VMware execute Java application** in the left pane and click **Run** in the right pane.

Attaching the Debugger to an Application Running in a Virtual Machine

You can attach the debugger to any Java application that is running in a virtual machine.

To attach to an application that is running in a virtual machine

- 1 In the **Debug** page, select the configuration under **VMware attach to application** in the left pane and click **Debug** in the right pane.
- 2 If more than one instance of the Java application is running in the virtual machine, a dialog box appears with a list of the running instances, each identified by their process ID, port number, and arguments.
Select the process you want to attach to.
- 3 Perform debugging tasks as you would in a local debugging environment.

NOTE Do not suspend a virtual machine while the integrated virtual debugger is connected to an application. If you do, the integrated virtual debugger will disconnect from the application.

Index

A

attaching to processes in a virtual machine **15**

C

changing the JVM path **14**

configurations

 creating to attach to applications **14**

 creating to start applications **13**

 deleting **15**

 duplicating to attach to applications **14**

 duplicating to start applications **13**

 editing to attach to applications **14**

 editing to start applications **13**

 managing **13**

D

debugging

 attaching to processes in a virtual machine **15**

 starting applications in virtual machine without **15**

 starting in a virtual machine **15**

deleting configurations **15**

J

JVM

 automatically selected **14**

 changing the path **14**

K

knowledge base

 accessing **6**

L

launch configurations **13**

 creating to attach to applications **14**

 creating to start applications **13**

 duplicating to attach to applications **14**

 duplicating to start applications **13**

 editing to attach to applications **14**

 editing to start applications **13**

S

starting

 applications in a virtual machine without
 debugging **15**

 debugging session in a virtual machine **15**

T

technical support resources **6**

U

user groups

 accessing **6**

V

VMware community forums

 accessing **6**

