Note

Before using this information and the product it supports, be sure to read the general information under "Notices" at the end of this book.
Contents

Figures ........................................ vii
Tables ........................................... ix
About this guide ............................... xi
Who should read this guide ..................... xi
What this guide contains ........................ xi
Publications ..................................... xii
IBM Tivoli Monitoring library ................. xii
Related publications ............................ xiii
Accessing publications online ................. xiv
Ordering publications ........................... xiv
Accessibility ..................................... xv
Contacting IBM Software Support ............. xv
Conventions used in this guide ............... xv
Typeface conventions ........................... xv
Operating system-dependent variables and paths xv
Summary of changes to this edition of the User’s Guide xvi

Part 1. Installing, configuring, and using ........ 1

Chapter 1. Introduction ......................... 3
Overview ........................................ 3
Understanding Tivoli Monitoring .............. 4
Resources ....................................... 4
Resource models ............................... 4
Cycles .......................................... 4
Thresholds ...................................... 4
Parameters ...................................... 5
Indications ..................................... 5
Occurrences and holes .......................... 6
Events .......................................... 6
Clearing events ................................ 8
Correlated events .............................. 8
Monitoring of events and indications ......... 9
At the Tivoli Enterprise Console server ...... 9
At the Web Health Console .................... 9
At the Tivoli Business Systems Manager .... 10
Recovery actions ................................ 10
Built-in actions ................................ 10
Tivoli Framework tasks ....................... 10
Scheduling ...................................... 10
Logging ........................................ 11
Profiles ........................................ 11
The heartbeat function ......................... 11
Components ..................................... 11
Tivoli Monitoring Base component .......... 12
Web Health Console ........................... 13
Endpoint component .......................... 13
Tivoli Business Systems Manager Adapter .. 13
Gathering Historical Data component ....... 13

Tivoli Enterprise Data Warehouse Support component ....................... 13
Collecting the data ............................ 14
Resource model data flow ...................... 14
Obtaining resource data at the endpoint ... 15
Security considerations ........................ 15
Tivoli Monitoring Resource Model Builder 16

Chapter 2. Installing ......................... 17
Supported platforms ........................... 17
Hardware requirements ........................ 18
Prerequisite software .......................... 19
Tivoli Management Framework ............... 19
Required Tivoli Management Framework patches .......................... 19
Other software ................................ 19
Coexistence with other versions of Tivoli Monitoring ..................... 22
Products within the Tivoli Monitoring family ........................ 23
Products within the Tivoli Distributed Monitoring (Classic Edition) family .. 23
Backward compatibility ......................... 23
Installing/upgrading the product ............. 24
Before you begin ............................... 25
Installing using the installation wizard ...... 25
Installing with Tivoli Software Installation Service ....................... 25
Installing from the Tivoli desktop .......... 26
Installing from the command line ........... 29
Upgrading from Tivoli Monitoring Version 5.1.1 ........................ 30
Upgrading a default resource model .......... 30
Upgrading from Workbench to Resource Model Builder .................. 31
Adding the notice group ........................ 31
Installing Java Runtime Environment ....... 31
Installation options ............................ 32
Enabling Tivoli Monitoring to work with Tivoli Data Warehouse using Gathering Historical Data .......... 34
Uninstalling the product ...................... 34
Uninstalling the endpoint components ....... 34
Uninstalling from servers and gateways .... 37

Chapter 3. Using the product ................. 39
Overview ........................................ 39
Profiles ......................................... 40
Key points about profiles ...................... 40
Setting up managed resources ................. 40
Creating profile managers and profiles ....... 41
Using the CLI to manage profiles at the server ......................... 43
Using default resource models ................ 43
Adding a default resource model to a profile ......................... 43
Using the CLI to add a default resource model to a profile ............ 44
Adding a customized resource model to a profile ....................... 45
Using the CLI to add a customized resource model to a profile ....... 46
Figures

1. Graph showing CPU usage generating indications, holes and an event . . . . . . 7
2. Components and event monitors of IBM Tivoli Monitoring. . . . . . . . . . 12
3. Data flow for profile distribution and resource model monitoring activities . . . . . . 14
4. Data flow for heartbeat control . . . . . . 71
5. Data flow for endpoint registration . . . . . . 72
6. Data flow for heartbeat function at gateway 73
7. Data flow using the wdmngcache command 74
8. Data flows for bulk/delta discovery . . . . 90
9. Data flow for viewing Tivoli Monitoring data in Tivoli Business Systems Manager . . . . 91
10. Overview of Tivoli Data Warehouse . . . . 96
11. Timeline for the logging of data for the central data warehouse . . . . . . . . . 99
12. The request manager processor . . . . . . 177
13. Tivoli Monitoring endpoint engine architecture. . . . . . . . . . . . . . 227
### Tables

1. Example threshold defaults .............................................. 5
2. Counting occurrences .................................................. 7
3. High CPU usage percentages over 12 cycles: holes and occurrences ................................................................. 7
4. High CPU usage percentages over 12 cycles: health .................................................................................. 9
5. Supported platforms .......................................................... 17
6. Hardware prerequisites for Tivoli Monitoring ....................... 18
7. Hardware prerequisites for Tivoli Business Systems Manager Adapter ................................................................. 19
8. Prerequisite software .......................................................... 19
9. Slots of the TMW_Event class .............................................. 80
10. Lifecycle of eventid property (no clearing event) ................. 81
11. Lifecycle of eventid property (with clearing event) ............... 81
12. Slots of the HeartBeat_Event class ...................................... 82
13. Install options for the Tivoli Enterprise Data Warehouse Support component ......................................................... 100
14. Vendor-specific information for RIM object attributes ........... 106
15. New and old command names ............................................. 109
16. Return codes from commands ............................................ 169
17. Health determination example ........................................... 175
18. Navigation icons .............................................................. 191
19. Health alert icons .............................................................. 192
20. Environment variables set by events ................................... 203
21. Other environment variables ............................................. 203
22. Procedures for creating Windows endpoints ......................... 208
About this guide

IBM® Tivoli® Monitoring allows you to monitor the availability and performance status of resources on your systems to identify bottlenecks and potential resource problems. This guide describes how to install, customize, and use IBM Tivoli Monitoring.

IBM Tivoli Monitoring was formerly known as Tivoli Distributed Monitoring (Advanced Edition). You should note that a separate product exists, called Tivoli Distributed Monitoring (Classic Edition), formerly known as Tivoli Distributed Monitoring. Descriptions of the IBM Tivoli Monitoring and Tivoli Distributed Monitoring (Classic Edition) products are given in the glossary on page 263.

Who should read this guide

This guide is for system administrators.

To make effective use of the product you require knowledge as well as practical experience of the following:

- Importing, installing, and managing the Tivoli Management Framework and the Tivoli Management Environment®
- System administration on the systems where you plan to install IBM Tivoli Monitoring components and which you plan to monitor using the product

You should also be familiar with the Tivoli Enterprise Console® product.

What this guide contains

This guide contains the following parts and chapters:

Part I. Installing, configuring, and using

- **Chapter 1, “Introduction”**
  Provides an introduction to Tivoli Monitoring, its features and functions and its role in the Tivoli environment.
- **Chapter 2, “Installing”**
  Describes the steps for installing Tivoli Monitoring in your Tivoli environment.
- **Chapter 3, “Using the product”**
  Describes the purpose of profiles in the Tivoli environment and presents the range of customization options that you can apply to the default resource models supplied with the product or to the resource models that you have created. It also describes how to distribute resource models to endpoints, and what to do if the distribution fails at one or more endpoints.
- **Chapter 4, “Heartbeat function”**
  Gives full details of the purpose, use, and management of the heartbeat function.
- **Chapter 5, “Integration with Tivoli Enterprise Console server”**
  Describes how to enable the monitoring of Tivoli Monitoring events on a Tivoli Enterprise Console server.
- **Chapter 6, “Integration with Tivoli Business Systems Manager”**
Describes the integration of Tivoli Monitoring with Tivoli Business Systems Manager and provides the steps for enabling the monitoring of Tivoli Monitoring events on the Tivoli Business Systems Manager workstation.

- **Chapter 7, “Integration with Tivoli Data Warehouse”**
  Describes the integration of Tivoli Monitoring with Tivoli Data Warehouse and the interaction between the two products.

- **Chapter 8, “Commands”**
  Lists and describes the IBM Tivoli Monitoring commands that you can issue from the command line.

**Part II. Web Health Console**

- **Chapter 9, “Introduction to the Web Health Console”**
  Describes the purpose and benefits of the Web Health Console in day-to-day monitoring of resources.

- **Chapter 10, “Using the Web Health Console”**
  Describes the steps for starting, configuring, and using the Web Health Console to monitor the resources of a specific list of endpoints.

**Part IV: Appendixes**

- **Appendix A, “Environment variables”**
  Documents the environment variables used by the product.

- **Appendix B, “Installing using the installation wizard”**
  Documents how to use the installation wizard to install the product.

- **Appendix C, “Migration considerations”**
  Describes the migration from Tivoli Distributed Monitoring (Advanced Edition) 4.1 and the migration from Tivoli Web Component Manager.

- **Appendix D, “Supporting clusters”**
  Describes the clustering solution offered by Tivoli Monitoring.

- **Appendix E, “Accessibility”**
  Provides information about the support provided in the product and its documentation for users with a visual or physical handicap.

- **Appendix F, “Use of Rhino: JavaScript for Java”**
  Provides the text of the license for the use of Rhino: JavaScript™ for Java™, which is incorporated in the product.

### Publications

This section lists publications in the IBM Tivoli Monitoring library and related documents. It also describes how to access Tivoli publications online, and how to order Tivoli publications.

**IBM Tivoli Monitoring library**

The following documents are available in the IBM Tivoli Monitoring library:

  Describes how to install, customize, and use IBM Tivoli Monitoring to manage system and application resources.

- **IBM Tivoli Monitoring: Problem Determination Guide, SH19-8520**
  Describes the problem determination tools and techniques available with IBM Tivoli Monitoring.
  Describes how to use IBM Tivoli Monitoring Resource Model Builder (formerly known as Workbench) to create new resource models or to modify existing ones.

• **IBM Tivoli Monitoring: Resource Model Builder Problem Determination Guide, SC32-1392**
  Describes the tools and troubleshooting techniques for doing problem determination for the Resource Model Builder.

  Provides information about using and customizing the resource models that can be used with IBM Tivoli Monitoring.

• **IBM Tivoli Monitoring: Release Notes, GI10-5797**
  Provides the most current information about IBM Tivoli Monitoring.

• **IBM Tivoli Monitoring: Deployment Supplement, SC23-4799**
  Describes how to use the deployment wizard for a custom installation.

• **IBM Tivoli Monitoring: Road Map for a Typical Installation, GI11-0938**
  Describes typical installation scenarios.

Versions of these documents in PDF and HTML formats can be found on the IBM Tivoli Monitoring product CDs. They are stored in the books directory, and can be accessed by selecting the file Books/infocenter.html with your Web browser. This displays an HTML page from which all of the documents can be accessed in either format.

Any updated versions of these documents are placed on the Tivoli Software Information Center web site (see “Accessing publications online” on page xiv, for more details).

## Related publications

Before beginning the installation, read the following related documentation for more information about the management options that the IBM Tivoli Monitoring products provide:

• **Tivoli Distributed Monitoring (Classic Edition): Release Notes**
  Provides updated information about the Tivoli Distributed Monitoring (Classic Edition) product (formerly known as Tivoli Distributed Monitoring).

• **Tivoli Decision Support for Server Performance Prediction: Release Notes**

• **Tivoli Management Framework: User’s Guide**
  Provides information about using the Tivoli Management Framework and Tivoli environment.

• **Tivoli Management Framework: Installation Guide**
  Provides information about installing and setting up the Tivoli Management Framework and Tivoli environment.

• **Tivoli Management Framework: Planning for Deployment Guide**
  Provides information that will help users planning the deployment of their Tivoli Management Framework and Tivoli environment.

• **Tivoli Management Framework: Reference Manual**
  Provides information about the Tivoli Management Framework commands.
• Tivoli Software Installation Service: User’s Guide
  Provides task-oriented information on how to import, manage, and install Tivoli Management Environment software on selected machines and managed nodes within your Tivoli management region.

• Tivoli Software Installation Service: Release Notes
  Provides important information about using and installing the Tivoli Software Installation Service (SIS)

• Tivoli Enterprise Console: Rule Builder’s Guide
  Provides information about using the Tivoli Enterprise Console rule editor and graphical rule builder to modify existing rules and create new rules to match your specific event management needs.

• Tivoli Enterprise Console: Release Notes, Version 3.7 or later
  Provides the most current information about Tivoli Enterprise Console.

• Tivoli Business Systems Manager: User’s Guide (if you intend to use Tivoli Business Systems Manager to monitor Tivoli Monitoring events)
  Describes how to use the Tivoli Business Systems Manager product.

• Tivoli Business Systems Manager: Console User’s Guide (if you intend to use Tivoli Business Systems Manager to monitor Tivoli Monitoring events)
  Describes how to use the Tivoli Business Systems Manager console.

The Tivoli Software Glossary includes definitions for many of the technical terms related to Tivoli software. The Tivoli Software Glossary is available, in English only, at the following Tivoli software library Web site:


Access the glossary by clicking the Glossary link on the left pane of the Tivoli software library window.

Accessing publications online

IBM posts publications for this and all other Tivoli products, as they become available and whenever they are updated, to the Tivoli Software Information Center web site at this link:

http://publib.boulder.ibm.com/tividd/td/tdprodlst.html

Click the Tivoli Monitoring link to access the product library.

Note: If you print PDF documents on other than letter-sized paper, set the option in the File → Print window that allows Adobe Reader to print letter-sized pages on your local paper.

Ordering publications

You can order many Tivoli publications online at the following Web site:


You can also order by telephone by calling one of these numbers:
• In the United States: 800-879-2755
• In Canada: 800-426-4968
• In other countries, for a list of telephone numbers, see the following Web site:

Accessibility

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully. With this product, you can use assistive technologies to hear and navigate the interface. You can also use the keyboard instead of the mouse to operate all features of the graphical user interface.

For additional information, see Appendix E, “Accessibility.”

Contacting IBM Software Support


If you want to contact software support, see the IBM Software Support Guide at the following web site: [http://techsupport.services.ibm.com/guides/handbook.html](http://techsupport.services.ibm.com/guides/handbook.html)

The guide provides information about how to contact IBM Software Support, depending on the severity of your problem, and the following information:

- Registration and eligibility
- Telephone numbers and e-mail addresses, depending on the country you are in
- What information you should gather before contacting support

Conventions used in this guide

This guide uses several conventions for special terms and actions, operating system-dependent commands and paths, and margin graphics.

Typeface conventions

The following typeface conventions are used in this book:

**Bold**
- Lowercase and mixed-case commands, command options, and flags that appear within text appear like this, in bold type.
- Graphical user interface elements (except for titles of windows and dialogs) and names of keys also appear like this, in bold type.

*Italic*
- Variables, values you must provide, new terms, and words and phrases that are emphasized appear like this, in italic type.

**Monospace**
- Commands, command options, and flags that appear on a separate line, code examples, output, and message text appear like this, in monospace type.
- Names of files and directories, text strings you must type, when they appear within text, names of Java methods and classes, and HTML and XML tags also appear like this, in monospace type.

Operating system-dependent variables and paths

This book uses the UNIX® convention for specifying environment variables and for directory notation.

When using the Windows® command line, replace $variable with %variable% for environment variables and replace each forward slash (/) with a backslash (\) in directory paths.
Note: If you are using the bash shell on a Windows system, you can use the UNIX conventions.

<table>
<thead>
<tr>
<th>Summary of changes to this edition of the User’s Guide</th>
</tr>
</thead>
<tbody>
<tr>
<td>All technical changes and new information are marked by change bars in the left margin.</td>
</tr>
<tr>
<td>The following information was added to this edition:</td>
</tr>
<tr>
<td>• The option to send a clearing event has been added to the GUI (see “Customizing thresholds and event rules” on page 47).</td>
</tr>
<tr>
<td>• A new default policy has been defined to be used with the task library (see “Running a task from the Tivoli desktop” on page 66).</td>
</tr>
<tr>
<td>• A new appendix has been added to describe the installation using the wizard (also known as Rapid Deployment).</td>
</tr>
<tr>
<td>• New environment variables were defined (see Appendix A, “Environment variables” for details).</td>
</tr>
<tr>
<td>• A scenario for supporting clusters on Windows has been documented (see Appendix D, “Supporting clusters” for details).</td>
</tr>
<tr>
<td>The following information was changed:</td>
</tr>
<tr>
<td>• The list of supported platforms was updated (see Table 5 on page 17).</td>
</tr>
<tr>
<td>• The procedure for installing Java Runtime Environment using wdmdistrib has been updated with new information regarding HP-UX systems (see “Installation options” on page 32).</td>
</tr>
<tr>
<td>• The procedure for uninstalling endpoints was modified (see “Uninstalling the endpoint components” on page 34).</td>
</tr>
<tr>
<td>• The syntax and options of the following commands were modified (see Chapter 8, “Commands” for details):</td>
</tr>
<tr>
<td>- wdmcmddistrib</td>
</tr>
<tr>
<td>- wdmcollect</td>
</tr>
<tr>
<td>- wdmconfig</td>
</tr>
<tr>
<td>- wdmdistrib</td>
</tr>
<tr>
<td>- wdmeditprf</td>
</tr>
<tr>
<td>- wdmheartbeat</td>
</tr>
<tr>
<td>- wdmiseng</td>
</tr>
</tbody>
</table>
Part 1. Installing, configuring, and using

Chapter 1. Introduction ........................................ 3
Overview ....................................................... 3
Understanding Tivoli Monitoring .......................... 4
  Resources .................................................. 4
  Resource models ....................................... 4
  Cycles ..................................................... 4
  Thresholds ............................................... 4
  Parameters .............................................. 5
  Indications .............................................. 5
  Occurrences and holes .................................. 6
Events ....................................................... 6
  Clearing events ....................................... 8
  Correlated events .................................... 8
Monitoring of events and indications ..................... 9
  At the Tivoli Enterprise Console server .......... 9
  At the Web Health Console ......................... 9
  At the Tivoli Business Systems Manager ........ 10
Recovery actions .......................................... 10
  Built-in actions ..................................... 10
  Tivoli Framework tasks ............................... 10
Scheduling .................................................. 10
Logging ..................................................... 11
Profiles ..................................................... 11
The heartbeat function ................................... 11
Components ............................................... 11
  Tivoli Monitoring Base component ............... 12
  Web Health Console ................................ 13
  Endpoint component ................................ 13
  Tivoli Business Systems Manager Adapter ........ 13
  Gathering Historical Data component ........... 13
  Tivoli Enterprise Data Warehouse Support .... 13
  Collecting the data ................................... 14
  Resource model data flow ......................... 14
  Obtaining resource data at the endpoint ....... 15
Security considerations ................................ 15
Tivoli Monitoring Resource Model Builder ............ 16

Chapter 2. Installing ........................................ 17
  Supported platforms .................................. 17
  Hardware requirements ............................... 18
  Prerequisite software ................................ 19
  Tivoli Management Framework ..................... 19
    Required Tivoli Management Framework patches ........................................ 19
    Other software ...................................... 19
  Coexistence with other versions of Tivoli Monitoring ..................... 22
    Products within the Tivoli Monitoring family ........................................ 23
    Products within the Tivoli Distributed Monitoring (Classic Edition) family ........ 23
  Backward compatibility ................................ 23
  Installing/upgrading the product ................. 24
    Before you begin ................................ 25
    Installing using the installation wizard .... 25
  Installing with Tivoli Software Installation Service ...................................... 25
  Installing from the Tivoli desktop ................ 26
  Installing from the command line ................ 26
  Upgrading from Tivoli Monitoring Version 5.1.1 ..................... 30
  Installing a default resource model .......... 30
  Upgrading from Workbench to Resource Model Builder ............. 31
  Adding the notice group ............................. 31
  Installing Java Runtime Environment ............ 31
  Installation options ................................ 32
  Enabling Tivoli Monitoring to work with Tivoli Data Warehouse using Gathering Historical Data ..................... 34
  Uninstalling the product ......................... 34
  Uninstalling the endpoint components .......... 34
  Uninstalling from servers and gateways ...... 37

Chapter 3. Using the product .............................. 39
Overview .................................................. 39
Profiles .................................................. 40
  Key points about profiles ........................ 40
  Setting up managed resources ................... 40
Creating profile managers and profiles ............. 41
  Using the CLI to manage profiles at the server ...... 43
Using default resource models ....................... 43
Adding a default resource model to a profile ....... 43
  Using the CLI to add a default resource model to a profile ..................... 44
Adding a customized resource model to a profile .... 45
  Using the CLI to add a customized resource model to a profile ..................... 46
Customizing a resource model ......................... 46
  Customizing thresholds and event rules ........ 47
    Specifying built-in actions for an event 49
    Editing a built-in action ..................... 49
    Removing a built-in action ................... 50
    Adding a built-in action ...................... 50
    Specifying tasks for an event ............... 50
    Sending an e-mail in response to an event 52
    Sending a notice in response to an event 53
  Customizing parameters ......................... 54
  Customizing scheduling information ............ 54
    Setting the monitoring period ............... 55
    Creating schedule rules ....................... 55
Customizing data logging information ................ 56
  Using the CLI to customize a resource model for a profile ..................... 58
  Setting distribution options for a profile ...... 58
  Profile distribution usingMDist 2 ............... 59
  Rerunning the failed distributions ............... 60
Deleting profile copies during an unsubscribe ........ 61
  Modifying the profile manager policy when no custom policy object is in use .......... 61
  Using the Tivoli desktop ....................... 61
  Using the CLI .................................. 62
  Modifying the profile manager policy when a custom policy object is in use .......... 62
Managing profiles and resource models at endpoints 63
<table>
<thead>
<tr>
<th>Chapter 4. Heartbeat function</th>
<th>69</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heartbeat statuses</td>
<td>69</td>
</tr>
<tr>
<td>Installing and configuring the heartbeat</td>
<td>69</td>
</tr>
<tr>
<td>Controlling the heartbeat</td>
<td>70</td>
</tr>
<tr>
<td>Heartbeat data flows</td>
<td>71</td>
</tr>
<tr>
<td>Endpoint registration</td>
<td>71</td>
</tr>
<tr>
<td>Heartbeat monitoring</td>
<td>73</td>
</tr>
<tr>
<td>Viewing the endpoint cache</td>
<td>74</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 5. Integration with Tivoli Enterprise Console server</th>
<th>77</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>77</td>
</tr>
<tr>
<td>Identifying the Tivoli Enterprise Console server to Tivoli Monitoring</td>
<td>77</td>
</tr>
<tr>
<td>Enabling monitoring of events and heartbeat messages on the Tivoli Enterprise Console server</td>
<td>78</td>
</tr>
<tr>
<td>Enable monitoring of the events on the Availability Intermediate Manager</td>
<td>80</td>
</tr>
<tr>
<td>Slots of the Tivoli Enterprise Console classes</td>
<td>80</td>
</tr>
<tr>
<td>Slots in the TMW_Event class</td>
<td>80</td>
</tr>
<tr>
<td>Slots in the HeartBeat_Event class</td>
<td>82</td>
</tr>
<tr>
<td>Understanding the Tivoli Enterprise Console rules</td>
<td>82</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 6. Integration with Tivoli Business Systems Manager</th>
<th>85</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>85</td>
</tr>
<tr>
<td>Installing the Tivoli Business Systems Manager Adapter</td>
<td>85</td>
</tr>
<tr>
<td>Configuring the Tivoli Business Systems Manager Adapter</td>
<td>86</td>
</tr>
<tr>
<td>Using the Tivoli Business Systems Manager Adapter</td>
<td>89</td>
</tr>
<tr>
<td>Viewing Tivoli Business Systems Manager data</td>
<td>90</td>
</tr>
<tr>
<td>Enabling and disabling Tivoli Business Systems Manager</td>
<td>92</td>
</tr>
<tr>
<td>Managing the Tivoli Business Systems Adapter from the CLI</td>
<td>93</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 7. Integration with Tivoli Data Warehouse</th>
<th>95</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>95</td>
</tr>
<tr>
<td>Customer scenario using Tivoli Monitoring</td>
<td>97</td>
</tr>
<tr>
<td>Enabling data logging to the central data warehouse</td>
<td>97</td>
</tr>
<tr>
<td>Installing and configuring the Tivoli Enterprise Data Warehouse</td>
<td>99</td>
</tr>
<tr>
<td>Supporting component</td>
<td>99</td>
</tr>
<tr>
<td>Installing the Tivoli Enterprise Data Warehouse Support</td>
<td>100</td>
</tr>
<tr>
<td>Creating and configuring the database structure</td>
<td>101</td>
</tr>
<tr>
<td>Verifying the prerequisites</td>
<td>101</td>
</tr>
<tr>
<td>Creating and configuring the database structure</td>
<td>102</td>
</tr>
<tr>
<td>Verifying the RIM object connection</td>
<td>104</td>
</tr>
<tr>
<td>Configuring Tivoli Enterprise Data Warehouse Support</td>
<td>104</td>
</tr>
<tr>
<td>Uploading data using the CLI</td>
<td>105</td>
</tr>
</tbody>
</table>

| Alternative installation and configuration procedures        | 105 |
| Creating a RIM object using a shell script                    | 105 |
| Creating a database structure using the SQL processor         | 107 |

<table>
<thead>
<tr>
<th>Chapter 8. Commands</th>
<th>109</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upgraded command names</td>
<td>109</td>
</tr>
<tr>
<td>Commands for managing profiles at the Tivoli server</td>
<td>110</td>
</tr>
<tr>
<td>Command for managing default resource models at the Tivoli server</td>
<td>110</td>
</tr>
<tr>
<td>Commands for managing the Tivoli Business Systems Manager Adapter</td>
<td>110</td>
</tr>
<tr>
<td>Commands for managing Tivoli Monitoring at the endpoints</td>
<td>111</td>
</tr>
<tr>
<td>Commands for managing Tivoli Monitoring at managed nodes/gateways</td>
<td>111</td>
</tr>
<tr>
<td>wdmcheckprereq</td>
<td>112</td>
</tr>
<tr>
<td>wdmcmd</td>
<td>113</td>
</tr>
<tr>
<td>wdmcmddistrib</td>
<td>115</td>
</tr>
<tr>
<td>wdmcollect</td>
<td>119</td>
</tr>
<tr>
<td>wdmconfig</td>
<td>121</td>
</tr>
<tr>
<td>wmdiscovery</td>
<td>127</td>
</tr>
<tr>
<td>wmdistrib</td>
<td>129</td>
</tr>
<tr>
<td>wmdumpmprf</td>
<td>132</td>
</tr>
<tr>
<td>wmdatprf</td>
<td>138</td>
</tr>
<tr>
<td>wdmeng</td>
<td>146</td>
</tr>
<tr>
<td>wdmheartbeat</td>
<td>148</td>
</tr>
<tr>
<td>wdmloadprf</td>
<td>150</td>
</tr>
<tr>
<td>wdmldeng</td>
<td>153</td>
</tr>
<tr>
<td>wdmnn</td>
<td>157</td>
</tr>
<tr>
<td>wdmngcache</td>
<td>159</td>
</tr>
<tr>
<td>wdmrmm</td>
<td>162</td>
</tr>
<tr>
<td>wdmtrcng</td>
<td>164</td>
</tr>
<tr>
<td>wtmmaddrm</td>
<td>166</td>
</tr>
<tr>
<td>wtmmdefrm</td>
<td>167</td>
</tr>
<tr>
<td>wtmmrmm</td>
<td>168</td>
</tr>
<tr>
<td>Return codes from commands</td>
<td>169</td>
</tr>
</tbody>
</table>
Chapter 1. Introduction

This chapter describes the main features of IBM Tivoli Monitoring 5.1.2 (hereafter also referred to as Tivoli Monitoring) and its role in the Tivoli environment. It is divided into these main sections:

- ‘Overview’ gives a general overview of the product.
- ‘Understanding Tivoli Monitoring’ explains the concepts behind the product.
- ‘Components’ details the components that can be installed, and explains the relationships between them.
- ‘Collecting the data’ describes the data flows for the various functions of the product.
- ‘Security considerations’ discusses the implementation of the product across firewalls.
- ‘Tivoli Monitoring Resource Model Builder’ provides high level information on this separate component.

Overview

IBM® Tivoli® Monitoring 5.1.2 is a Tivoli application that applies preconfigured best practices to the automated monitoring of essential system resources. The application detects bottlenecks and other potential problems and provides for the automatic recovery from critical situations, which eliminates the need for system administrators to manually scan through extensive performance data. The application also integrates seamlessly with other Tivoli availability solutions, including Tivoli Business Systems Manager® and Tivoli Enterprise Console®. This application was previously called Tivoli Distributed Monitoring (Advanced Edition).

Most features of Tivoli Monitoring can be used as supplied or can be modified manually using the graphical user interfaces (GUIs) or the command line interface (CLI) provided.

The main features of Tivoli Monitoring are:

- An off-the-shelf solution for monitoring Windows®, UNIX®, Linux, and OS/400® systems. Data collection and problem analysis is performed locally on the system.
- Ready-to-use resource models that report on specific aspects of a system’s status. For example, the Process resource model provides information on the status of processes, CPU usage, and so forth. Resource monitoring is an implementation of the Common Information Model (CIM). CIM is an approach to system and network management that applies object-oriented techniques to model the system.
- Resource models that can easily be added to a Tivoli profile, which can be distributed to multiple systems simultaneously.
- The ability to modify resource models by changing, for example, threshold levels to match specific requirements.
- The ability to view both real-time and historical data for any system from a centralized monitoring application called the Web Health Console, which is supplied with the product.
• The ability to send the results from the data collection and analysis to the Tivoli Enterprise Console or to the Tivoli Business Systems Manager.
• The ability to specify automatic corrective or preventive actions to resolve situations that could develop into real problems.
• A scheduling feature that enables monitoring to take place at user-specified times.
• A heartbeat function, running at gateways, that regularly checks the availability and status of attached endpoints and makes the information available to the Tivoli Enterprise Console server, Tivoli Business Systems Manager, or Tivoli Monitoring Notice Group.

Understanding Tivoli Monitoring

This section explains the main concepts behind the product.

Resources

Tivoli Monitoring monitors resources at distributed systems. In this context, a resource is anything that affects the operation of a computer system and includes physical and logical disks, CPUs, memory, printers, as well as the processes and services running, such as LanMan, the Windows event log, the UNIX (logging system daemon) syslogd, and TCP/IP.

Resource models

Tivoli Monitoring uses out-of-the box, predefined resource models to specify which resource data are accessed from the system at runtime and how this data is processed. For example, the Process resource model obtains data related to processes running on the system. Performance data is automatically collected by the resource model and processed by an appropriate algorithm to determine whether or not the system is performing to your expectations. Generally, you can use the resource model default values and still obtain useful data. However, if necessary, you can customize the resource models to suit your requirements.

Full details about the resource models supplied with the product are given in the IBM Tivoli Monitoring: Resource Model Reference Guide.

Cycles

When a resource model is run at an endpoint, it gathers data at regular intervals, known as cycles; the duration of a cycle is the cycle time. A resource model with a cycle time of 60 seconds gathers information every 60 seconds. The data collected is a snapshot of the status of the resources specified in the resource model. Each of the supplied resource models has a default cycle time, which you can modify as required.

Thresholds

Each resource model defines one or more thresholds. A threshold is a named property of the resource with a default value that you can modify in the customization phase. Typically, the value specified for a threshold represents a significant reference level of a performance-related entity, which, if exceeded or not reached, a system administrator might want to know about. However, some thresholds are used as reference values to limit the scope of the resource model. For example, in the Windows Process resource model, the following are amongst the thresholds used:
Table 1. Example threshold defaults

<table>
<thead>
<tr>
<th>Threshold</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>High CPU Usage</td>
<td>60</td>
</tr>
<tr>
<td>Maximum Processes</td>
<td>5</td>
</tr>
</tbody>
</table>

The threshold High CPU Usage is used to monitor excessive CPU usage, that perhaps might be damaging to other processes running at the same time. The value for this threshold is measured as a percentage of the CPU capacity, the default being 60%. The Maximum Processes threshold, instead, determines that only the top 5 (the default value) highest CPU-using processes are considered in the monitoring of high CPU usage.

Alternatively, a threshold can be a minimum below which a certain performance characteristic should not fall. For example, the Windows Logical Disk resource model has a Low Disk Space threshold showing the minimum percentage disk space that must always be available (the default is 5%).

Parameters

Some resource models have one or more parameters. Each parameter can take the form of a list of strings, a list of numeric values, a Boolean list of predetermined values from which you can make any combination of selections, or a choice list of mutually exclusive alternatives. For example, the Windows Parametric TCP/IP Ports resource model has a parameter where you list the ports to be monitored and another to choose the port states to monitor.

Indications

Each resource model generates an indication if certain conditions implied by the resource model’s thresholds are not satisfied in a given cycle. Each resource model has its own algorithm to determine which combinations of thresholds should generate an indication. Indications might be generated in any one of the following circumstances:

- A single threshold is exceeded. For example, in the Windows Process resource model, the Process High CPU indication is generated when the High CPU Usage threshold is exceeded (for any process that has a nonzero process ID).
- A combination of two or more thresholds are exceeded. For example, in the Windows Logical Disk resource model a High Read Bytes per Second indication is generated when both the following thresholds are exceeded:
  - The amount of bytes transferred per second (being written or read) exceeds the High Bytes per Second threshold
  - The percent of time that the selected disk drive spends for read or write requests exceeds the High Percent Usage threshold.
- A combination of other factors has changed. For example, in the Windows Process resource model the Process Handle Leak indication is generated when a process is leaking handles. There is no threshold for this indication. The resource model compares the number of handles of the five processes with the most handles in consecutive cycles. If the number of handles has increased, the indication is generated.

The specific definition of the indications that have been created for each resource model are documented in IBM Tivoli Monitoring: Resource Model Reference Guide.


**Occurrences and holes**

An occurrence is the term used to refer to a cycle during which an indication occurs for a given resource model.

A hole is the term used to refer to a cycle during which an indication does not occur for a given resource model. In other words, none of the conditions specified for the generation of any indication have been met. This does not mean that none of the thresholds have been exceeded. For example, in the Windows Logical Disk resource model a High Read Bytes per Second indication is not created when the percentage disk time is higher than the High Percent Usage threshold, provided that the Low Disk Space threshold is exceeded.

**Events**

An event is used to verify the persistence of a given indication by eliminating unrepresentative peaks and troughs for the indication. For example, a process that generates the Process High CPU indication in one cycle may be behaving perfectly normally, and be of no threat to other processes if the high usage is not repeated. However, an indication that persists over several cycles is more likely to be a problem. Thus, an event defines the number of consecutive occurrences of the indication that are significant.

However, given that you have decided that a certain number of consecutive cycles of Process High CPU greater than the chosen threshold value is significant, you may feel that if, during the accumulation of the consecutive occurrences, one or two cycles fall below the threshold, it should not stop the counting of consecutive occurrences. Thus, an event allows you to define how many consecutive holes in the sequence of consecutive occurrences are permitted.

So, an event is an aggregation of a defined number of consecutive occurrences during which there can be a defined number of consecutive holes. For example, consider an event that is defined as three occurrences and two holes. This means that one or two consecutive holes will be ignored by the aggregation algorithm in detecting three consecutive occurrences of the indication.

In the following two series of events, 1 represents an occurrence and 0 represents a hole.

- \( 0 1 0 1 0 0 1 \)
- \( 0 0 1 0 0 1 1 \)

Notice that, ignoring the sequences of one or two consecutive holes as stated in this example event definition, each series contains three consecutive occurrences. Therefore, in this example, each sequence represents an event.

Defining an event as three occurrences and two holes also means that the maximum number of cycles needed to trigger an event is seven cycles. In this case, each occurrence would be following by the maximum of two holes as shown below.

- \( 1 0 0 1 0 0 1 \)

Further, if the monitoring algorithm has counted two consecutive occurrences and then observes three consecutive holes – more holes than that allowed in the definition of an event for this example – it sets the count of occurrences to zero. See Table 2 on page 7.
Table 2. Counting occurrences

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Count of occurrences</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0 0</td>
<td>1</td>
</tr>
<tr>
<td>1 0 0 1</td>
<td>2</td>
</tr>
<tr>
<td>1 0 0 1 0</td>
<td>2</td>
</tr>
<tr>
<td>1 0 0 1 00</td>
<td>2</td>
</tr>
<tr>
<td>1 0 0 1 00 0</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 1 shows a graph of the actual level of CPU usage on a hypothetical system, where the indication threshold is at 60% of CPU usage.

Example High CPU Usage (threshold 60%)

![Example High CPU Usage Graph](image)

Figure 1. Graph showing CPU usage generating indications, holes and an event

The snapshots of the CPU usage obtained by the resource model are shown in Table 3. The first row shows the cycle number and the second the percentage of CPU usage.

Table 3. High CPU usage percentages over 12 cycles: holes and occurrences

<table>
<thead>
<tr>
<th>Cycle</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU%</td>
<td>55</td>
<td>73</td>
<td>54</td>
<td>63</td>
<td>68</td>
<td>42</td>
<td>50</td>
<td>70</td>
<td>90</td>
<td>55</td>
<td>70</td>
<td>80</td>
</tr>
<tr>
<td>Occurrence or Hole</td>
<td>H</td>
<td>O</td>
<td>H</td>
<td>O</td>
<td>O</td>
<td>H</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>Occurrence Count</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

Cycles that exceed the threshold, and are thus occurrences, are shown in bold type; the other cycles are holes.

For this example, it has been determined that an event is going to be created for this indication if there are four consecutive occurrences with only one hole permitted. The final row in Table 3 shows the count that the resource model would make of the consecutive occurrences. The count is set to zero at the 7th cycle.
snapshot because the resource model has encountered two consecutive holes. In this example, an event is triggered in the 12th cycle. The count is then set to zero and the process continues.

Events may be generated multiple times (when the problem persists and no clearing event function is enabled) or just once (when the clearing event function is enabled). See "Clearing events" to understand how they work.

**Clearing events**
A clearing event is a resource model function that, if enabled, allows Tivoli Monitoring to close an error event when the circumstances that have caused the event are no longer present.

When the clearing event function is enabled for a resource model, events (carrying a specified event ID) are sent only the first time the aggregation rule is satisfied.

A clearing event (carrying the same event ID) is sent when both of the following conditions are true:
- one event was sent
- and the current number of holes exceeds the number of holes defined in the aggregation rule for sending the event

Clearing events can be processed by the Tivoli Enterprise Console server and by Tivoli Business Systems Manager. Clearing events have a severity of *harmless*, regardless of the severity of the original event, but have the same event ID as the original event. See "Slots in the TMW_Event class" on page 80 to understand the lifecycle of the event ID property when clearing events are used.

To give an example of clearing event, let’s consider the following scenario: a service stops and the Tivoli Enterprise Console server receives an event notifying it of this problem. Until the service restarts, the problem is still present on the endpoint, but Tivoli Monitoring does not send any further event notifications to the Tivoli Enterprise Console server. If the clearing event function is enabled, as soon as the service restarts, a clearing event is sent to the Tivoli Enterprise Console server, thereby closing the original event. The clearing event itself does not normally appear on the server, as its only function is to clear the original error event.

**Note:** The clearing event function is not always enabled. For example for the Parametric Event Log and the Event Log resource models the clearing event function is disabled.

**Note:** Correlated events cannot be cleared.

**Correlated events**
Use event correlation when you need to understand in more detail the kind of problem your resources are having. This function correlates two or more events defined in different resource models that belong to the same profile. Correlation makes it possible to highlight specific resource problems that would not be evident only on the basis of single separated events. For example, an event notifies you that a running process is leaking memory; then another event notifies you that you are running out of memory. Finally, a correlated event notifies you that you have low memory because there is a process that is leaking memory.
To enable event correlation, it is necessary to load the resource models where the events to correlate are specified and it is necessary to have these resource models running in the same profile.

Correlated events are sent to the Tivoli Enterprise Console server if in the profile you have specified to send an event notification to that server. This notification cannot be disabled through the GUI, nor through the command line interface. Each correlated event is identified by an event ID, but this ID is not connected in any way to the events that generated the correlated event.

The clearing function is not active for correlated events. Therefore, when a problem, highlighted by a correlated event, is resolved, no notification is sent to the Tivoli Enterprise Console.

The event correlation function is enabled only on Windows platforms.

**Monitoring of events and indications**

Events can be sent to the Tivoli Enterprise Console server, the Web Health Console, and Tivoli Business Systems Manager.

**At the Tivoli Enterprise Console server**

Events can be viewed by a Tivoli Enterprise Console server provided that you have compiled and loaded the relevant Basic Recorder of Objects in C (BAROC) files on the server. The event contains a set of properties that can help to identify the problem. For example, the information in the ProcessHandleLeak event includes values for the following:

- Current Process ID
- Number of handles allocated to the process
- Name of the process

Clearing events can also be processed by the Tivoli Enterprise Console server. If the default procedure is used to enable the monitoring of events (see "Enabling monitoring of events and heartbeat messages on the Tivoli Enterprise Console server" on page 78), the Tivoli Enterprise Console server uses the clearing event to close the associated error event. However, if you choose not to install the clearing events rule incorporated in the default procedure, the clearing event is displayed as a separate entity with the same ID as the original error event.

**At the Web Health Console**

The Web Health Console, which is an optional part of Tivoli Monitoring, obtains events and indications from endpoints. The Web Health Console displays the health of each potential problem as a numeric value between 100 (perfect health) and zero (with zero meaning that the conditions for the corresponding event have been met). Intermediate values show the percentage of occurrences currently registered with respect to the total number of occurrences needed to trigger an event. For example, Table 4 is based on Table 3 on page 7 but it shows also the health percentage:

<table>
<thead>
<tr>
<th>Cycle</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU%</td>
<td>55</td>
<td>73</td>
<td>54</td>
<td>63</td>
<td>68</td>
<td>42</td>
<td>50</td>
<td>70</td>
<td>90</td>
<td>55</td>
<td>70</td>
<td>80</td>
</tr>
</tbody>
</table>

Table 4. High CPU usage percentages over 12 cycles: health
Table 4. High CPU usage percentages over 12 cycles: health (continued)

<table>
<thead>
<tr>
<th>Health (%)</th>
<th>100</th>
<th>75</th>
<th>75</th>
<th>50</th>
<th>25</th>
<th>25</th>
<th>100</th>
<th>75</th>
<th>50</th>
<th>50</th>
<th>25</th>
<th>0</th>
</tr>
</thead>
</table>

In this example, the health percentage changes in steps of 25% because 4 occurrences were required to trigger an event; if the indication had required 5 occurrences, the health percentage would have changed by steps of 20%.

**At the Tivoli Business Systems Manager**

Events can also be sent to the Tivoli Business Systems Manager, provided that the Tivoli Business Systems Manager Adapter component has been installed on the gateways of the endpoints that are to be monitored. A full description of Tivoli Business Systems Manager can be found in the Tivoli Business Systems Manager documentation. For more information about the integration of Tivoli Business Systems Manager with Tivoli Monitoring, see Chapter 6, “Integration with Tivoli Business Systems Manager,” on page 85.

Clearing events can also be processed by the Tivoli Business Systems Manager, which uses the clearing event to close the associated error event.

**Recovery actions**

Recovery actions can be run automatically for any event. Recovery actions can be of two types: built-in actions and Tivoli Framework Tasks. The recovery actions can take positive steps to remedy the situation, or can ensure that information about the event is distributed to the appropriate authorities or entities.

**Built-in actions**

Certain events may have one or more actions predefined for them. An action can be either the execution of a CIM class method, or the execution of a program. Both type of actions can be implemented only through the Resource Model Builder. For example, an event that detects the failure of a service could have the restart of that service as its built-in action. Thus, without any human intervention, Tivoli Monitoring would detect the failure of a service and automatically restart it. Built-in actions are defined by default as part of an event, but can be removed. They have the same event ID as the event that they are designed to correct.

**Tivoli Framework tasks**

For each event you can select one or more Tivoli Framework Tasks to be performed when the event is triggered. See the Tivoli Management Framework: User’s Guide and the Tivoli Management Framework: Reference Manual for more details about Tivoli Framework Tasks.

**Note:** The tasks that are triggered by a Tivoli Monitoring event can access the event name and event thresholds of the triggering event by accessing the environment variables. Refer to Appendix A, “Environment variables,” on page 203 for the list of environment variables used by the product.

**Scheduling**

Tivoli Monitoring contains a scheduling feature that allows you to determine a period within which monitoring should take place, and specific scheduling rules.

The monitoring period is determined by defining a from and a to date.
The scheduling rules allow you to define time periods on specific weekdays during which monitoring will take place. Any number of rules can be defined, letting you set up a complex pattern of resource monitoring for a profile, covering the periods relevant to you.

The scheduled times are always interpreted as local times, allowing you to set up a single rule that monitors the same local time period in different time zones. For example, if your region covers several time zones, but you want to monitor morning activities in each time zone, a single rule defining the monitoring period as being between 08:00 and 13:00, is interpreted locally in each of the time zones, so that you monitor the same relative period.

You should note also that all times of events or activities reported from endpoints or gateways are also logged in the local time of the system from where they originated.

Logging
For any endpoint, you can log the data collected by a resource model and write it in a local database. Then you can view it through the History View of the Web Health Console. You can choose to store raw or aggregated data.

For more information, see “Customizing data logging information” on page 56.

Profiles
Tivoli Monitoring is a profile-based application that runs in a Tivoli environment. Different profiles can be defined containing different selections of resource models. All aspects of existing profiles can be modified, including the addition, deletion and customization of resource models. You can distribute multiple profiles to each endpoint.

The heartbeat function
In addition to the monitoring processes described above, Tivoli Monitoring operates a heartbeat function, which monitors the basic system status at endpoints attached to the gateway at which it is enabled. Events may be sent to the Tivoli Business Systems Manager (provided that the Tivoli Business Systems Manager Adapter component is installed at the gateway), the Tivoli Enterprise Console and the IBM Tivoli Monitoring Notice Group. Full details of this function are given in Chapter 4, “Heartbeat function,” on page 69.

Components
Tivoli Monitoring can be installed and configured from any Tivoli management region server (Tivoli server). Tivoli Monitoring has components that can be installed on the Tivoli server and gateways. On the endpoint, the product component is automatically installed at the first profile distribution.

Figure 2 on page 12 is a diagram of the components of the product. It also shows the various monitors that can receive information from the product.
## Tivoli Monitoring Base component

This product component should be installed on the Tivoli management region server and on all gateways to which are attached endpoints that you want to monitor. In this guide, the term *gateway* is used to mean a managed node configured as a gateway. Installation instructions can be found in Chapter 2, “Installing,” on page 17.

The component comprises a graphical user interface and a command line interface, available at both server and gateway, and all functions of the product can be controlled from either node. However, you should note that the database of available default resource models is maintained at the server; commands issued to manage resource models from gateways are routed to and performed on the server.

In addition the component can be configured to operate the heartbeat function for all endpoints directly attached to the system on which it is installed.

For more information, see Chapter 4, “Heartbeat function,” on page 69.
Web Health Console

The Web Health Console is the web-based graphical interface for Tivoli Monitoring that runs on any system that can be connected through TCP/IP to the Tivoli management region. The Web Health Console allows you to view real time information about a specific problem, and check the status (or health) of a set of endpoints. You can use the Web Health Console to work with real time data or with historical data that has previously been logged to a local database.

For more information, see Part 2, “Web Health Console,” on page 173

Endpoint component

The endpoint component (which requires a Tivoli management agent) performs the resource management through one or more resource models that are distributed to the endpoint with a Tivoli Monitoring profile. The endpoint component is installed automatically when a Tivoli Monitoring profile is distributed to the endpoint for the first time.

Tivoli Business Systems Manager Adapter

This component is installed alongside the base component on a gateway (or a server which is directly attached to endpoints). It feeds discovery information and Tivoli Monitoring events to the Tivoli Business Systems Manager.

For more information see Chapter 6, “Integration with Tivoli Business Systems Manager,” on page 85

Gathering Historical Data component

The Gathering Historical Data component enables Tivoli Monitoring to use Tivoli Decision Support for Server Performance Prediction (Advanced Edition) and Tivoli Data Warehouse.

The Gathering Historical Data component uses data collected by specific Tivoli Monitoring resource models to populate a database on the Tivoli server where it is installed. The collected data is aggregated every 24 hours and added to the Tivoli Monitoring database, from where it can be used in analyses that help plan network growth using key system metrics.


For more information about Tivoli Data Warehouse, see the following publications:
- Tivoli Data Warehouse: Installing and Configuring
- Tivoli Enterprise Data Warehouse: Enabling an Application for Tivoli Enterprise Data Warehouse
- Tivoli Monitoring Warehouse Enablement Pack: Implementation Guide

Tivoli Enterprise Data Warehouse Support component

The Tivoli Enterprise Data Warehouse Support component also enables the integration of Tivoli Monitoring with Tivoli Data Warehouse.

As opposed to the Gathering Historical Data component, Tivoli Enterprise Data Warehouse Support interacts with the IBM Tivoli Monitoring for (...) products. Therefore, if your want to exploit any of the features of the IBM Tivoli Monitoring
Collecting the data

This section describes how the product obtains the data required for monitoring at the endpoints.

**Resource model data flow**

Figure 3 shows the data flow for a resource model.

The resource model is distributed (pushed) as part of a profile to the endpoint through the gateway (steps 1, 2, and 3). It starts monitoring the endpoint's resources and sends information about all events to the gateway, and information about unsecure events directly to the Tivoli Enterprise Console server (step 4). The
gateway, depending on which monitors are enabled, distributes events to the Tivoli Business Systems Manager CommonListener and secure events to the Tivoli Enterprise Console server (step 5).

**Obtaining resource data at the endpoint**

Tivoli Monitoring uses processes that form part of the endpoints’ operating systems to obtain resource data, as follows:

**Windows operating systems**

On Windows operating systems, the product uses Windows Management Instrumentation (WMI), which is Microsoft®’s implementation of CIM. WMI enables applications, including Tivoli Monitoring, to retrieve information about the current status of a system.

**Note:** WMI is supplied as part of Windows 2000. On Windows NT®, WMI must be installed on each system.

Other collection agents can also be used, and resource models can be defined that monitor resources not managed by WMI.

**UNIX and Linux operating systems**

On UNIX and Linux operating systems, the information collection agent is incorporated in the product, based on CIM specifications.

**OS/400 operating systems**

On OS/400 operating systems, the information collection agent is incorporated in the product, based on CIM specifications.

**Security considerations**

Tivoli Monitoring exploits the infrastructure provided by Tivoli Management Framework, Version 3.7.1 (and beyond), to enable the functionality of the product across firewalls. To protect the privacy and data integrity, Tivoli Management Framework 3.7.1 enables you to either configure Bulk Data Transfer (BDT) proxy mechanism and Secure Socket Layer 3 (SSL3) encryption support, or install the Tivoli Management Framework Firewall Security Toolbox.

For further information, refer to the following documentation:

- **Tivoli Management Framework Firewall Security Toolbox Release Notes, Version 1.2**
  This provides information about installing and configuring the components of the Tivoli Management Framework Firewall Security Toolbox.

- **Tivoli Management Framework: Release Notes, Version 3.7.1**
  This book contains information about Security and New Command Options.

- **Tivoli Management Framework: Planning for Deployment Guide, Version 3.7.1**
  This book contains information about Secure Sockets Layer Data Encryption.

- **Secure Socket Layer Introduction and iKeyman User’s Guide**
  This provides further background information about SSL3.

- **Tivoli Enterprise Management across Firewalls (redbook SG24-5510-01)**
  This provides background information and includes scenarios that refer to Tivoli Distributed Monitoring (Classic Edition).
Tivoli Monitoring Resource Model Builder

Starting from IBM Tivoli Monitoring 5.1.1 Fix Pack 5 (5.1.1-ITM-FP05), the Tivoli Monitoring Workbench is replaced by the new Tivoli Monitoring Resource Model Builder component, which can be downloaded from the following web page:


Tivoli Monitoring Resource Model Builder is a stand-alone component that provides an integrated environment for developing, debugging, and packaging resource models for Tivoli Monitoring. You can use the Resource Model Builder to develop your own resource models or to modify existing resource models provided with Tivoli Monitoring.


In this guide the term Workbench is still used when referring to versions of this component prior to IBM Tivoli Monitoring 5.1.1 Fix Pack 5 (5.1.1-ITM-FP05).
Chapter 2. Installing

This chapter lists the supported operating systems and the prerequisite hardware and software for installing and running IBM Tivoli Monitoring and gives details of how to perform the following tasks:

- Installing the product using the installation wizard
- Installing the product with the Tivoli Software Installation Service
- Installing the product from the Tivoli Desktop
- Installing the product from the command line
- Upgrading from Tivoli Distributed Monitoring 5.1.1
- Adding the Notice Group
- Installing Java Runtime Environment
- Uninstalling the endpoint components
- Uninstalling the product from servers and gateways

For information about integrating Tivoli Monitoring with other Tivoli products, see:

- Chapter 5, “Integration with Tivoli Enterprise Console server,” on page 77
- Chapter 6, “Integration with Tivoli Business Systems Manager,” on page 85
- Chapter 7, “Integration with Tivoli Data Warehouse,” on page 95

### Supported platforms

Table 5 details the supported platforms for servers, gateways, and endpoints:

<table>
<thead>
<tr>
<th>System</th>
<th>Versions</th>
<th>Server</th>
<th>Gateway</th>
<th>Endpoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIX®</td>
<td>4.3.3, 5.1.0.C (3) (4), 5.2 (5) (4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Solaris</td>
<td>2.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7, 8, 9 (3) (4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows NT, Version 4.0</td>
<td>Service Packs 6, and 6a</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows 2000</td>
<td>Server, Advanced Server,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Professional, DataCenter Svr sp3+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows Server 2003</td>
<td>Standard, Enterprise</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows XP</td>
<td>Professional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Turbo Linux Svr</td>
<td>6.1, 6.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SuSE</td>
<td>6.4, 7.0, 7.1, 7.2, 8.0, and 8.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SuSE SLES</td>
<td>7.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLES</td>
<td>7.0 for S/390® and z/Series (5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HP-UX (1)</td>
<td>11, 11i (3)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OS/400</td>
<td>5.1, 5.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RedHat Server (IA32)</td>
<td>7.0, 7.1, 7.2, and 7.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RedHat Ent Linux (IA32)</td>
<td>2.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RedHat for OS/390®</td>
<td>6.0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© Copyright IBM Corp. 2004
Table 5. Supported platforms (continued)

<table>
<thead>
<tr>
<th>System</th>
<th>Versions</th>
<th>Server</th>
<th>Gateway</th>
<th>Endpoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>RedHat for Intel™</td>
<td>7.3, 8.0</td>
<td>■</td>
<td>■</td>
<td>■</td>
</tr>
<tr>
<td>UL (SLES 8)</td>
<td>1.0 for IA32 sp2+</td>
<td>■</td>
<td>■</td>
<td>■</td>
</tr>
<tr>
<td>UL (SLES 8)</td>
<td>1.0 for z/Series sp2+ (3) (4)</td>
<td>■</td>
<td>■</td>
<td>■</td>
</tr>
<tr>
<td>UL (2) (SLES 8)</td>
<td>1.0 for PowerPC</td>
<td>■</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Supported platforms notes:
- (1) The Tivoli Business Systems Manager Adapter cannot be installed on HP-UX gateways.
- (2) Supported starting from 5.1.1-ITM-FP04 and 4.1-TMF-0013,-0014,-0015.
- (3) 32-bit application tested on 32-bit kernel.
- (4) 32-bit application tested on 64-bit kernel under “toleration” mode.
- (5) Servers and gateways support these platforms only if based on Tivoli Management Framework 4.1.

Note: Use of particular components or functions of the product may require the use of other than the minimum operating system versions shown here (see “Prerequisite software” on page 19).

Hardware requirements

Table 6 lists the minimum hardware requirements for installing and using Tivoli Monitoring on servers, gateways and endpoints. For applications running resource models from the “IBM Tivoli Monitoring for ...” products, the memory footprint may be larger than the numbers indicated in the table.

Table 6. Hardware prerequisites for Tivoli Monitoring

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Minimum requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>CPU Same requirements as Framework</td>
</tr>
<tr>
<td></td>
<td>RAM</td>
</tr>
<tr>
<td></td>
<td>Disk space 85 MB</td>
</tr>
<tr>
<td>Gateway</td>
<td>CPU Same requirements as Framework</td>
</tr>
<tr>
<td></td>
<td>RAM</td>
</tr>
<tr>
<td></td>
<td>Disk space 85 MB</td>
</tr>
<tr>
<td>Windows Endpoint</td>
<td>CPU 266 MHz</td>
</tr>
<tr>
<td></td>
<td>RAM 128 MB</td>
</tr>
<tr>
<td></td>
<td>Disk space 30 MB</td>
</tr>
<tr>
<td>UNIX/Linux Endpoint (5)</td>
<td>CPU Same requirements as Framework</td>
</tr>
<tr>
<td></td>
<td>RAM</td>
</tr>
<tr>
<td></td>
<td>Disk space 100 MB</td>
</tr>
<tr>
<td>OS/400 Endpoint</td>
<td>CPU Same requirements as Framework</td>
</tr>
<tr>
<td></td>
<td>RAM</td>
</tr>
<tr>
<td></td>
<td>Disk space 20 MB</td>
</tr>
</tbody>
</table>

Hardware prerequisites notes:
(1) UNIX/Linux endpoints must have at least 256 MB of free memory. This is needed because the maximum allowed heap of the Java process on the Tivoli Monitoring engine is configured to be 256 MB.

Table 7 lists the minimum hardware requirements for installing the Tivoli Business Systems Manager Adapter on gateways.

Table 7. Hardware prerequisites for Tivoli Business Systems Manager Adapter

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Minimum requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gateway</td>
<td></td>
</tr>
<tr>
<td>CPU</td>
<td>Same requirements as Framework</td>
</tr>
<tr>
<td>RAM</td>
<td></td>
</tr>
<tr>
<td>Disk space</td>
<td>2 MB</td>
</tr>
</tbody>
</table>

Prerequisite software

Prerequisite software information is provided in separate sections for Tivoli Management Framework and for other software.

Tivoli Management Framework

Tivoli Monitoring requires as a minimum Tivoli Management Framework Version 3.7B, but Version 3.7.1 or 4.1 is recommended (some platforms require patches, as detailed below).

Required Tivoli Management Framework patches

If you have Tivoli Management Framework Version 4.1, you are recommended to install patches 4.1-TMF-0013, 4.1-TMF-0014, and 4.1-TMF-0015 (this last patch is required to run on iSeries and pSeries Linux endpoints).

If you have Tivoli Management Framework Version 3.7.1, you are recommended to install patches 3.7.1-TMF-0097, 3.7.1-TMF-0098, 3.7.1-TMF-0099, and 3.7.1-TMF-0114, while 3.7.1-TMF-107 must also be installed to run the product on iSeries and pSeries Linux endpoints.

Some platforms have specific patch requirements, as follows:

Windows 2000

On Tivoli Management Framework, 3.7B, patch 3.7-TMF-0010 is a prerequisite for any version of the Windows 2000 operating system, and you are recommended to install it on all other platforms where it is available.

Windows XP

For Windows XP Professional endpoints patch 3.7.1-TMF-0044 is required on the server.

Other software

Table 8 gives details of other software prerequisites, depending on your working environment:

Table 8. Prerequisite software

<table>
<thead>
<tr>
<th>Feature</th>
<th>Required software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Systems using the Web Health Console product component</td>
<td>Netscape 6.x or Internet Explorer 6.x must be installed on the system.</td>
</tr>
</tbody>
</table>
Table 8. Prerequisite software  (continued)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Required software</th>
</tr>
</thead>
</table>
| Endpoints running UNIX or Linux              | • Java Runtime Environment (JRE) 1.3.0 or 1.3.1 must be installed on the endpoint (endpoints running Solaris must have JRE 1.3.1-08, while endpoints running AIX must have JRE 1.3.1). However, it is recommended that you install and use the JRE available with the product CD:  
  - If you already have an appropriate copy of JRE installed on the target system, after installing IBM Tivoli Monitoring you should use the provided task `DMLinkJre` to link the product to your existing JRE (see “Installation options” on page 32). Use task `DMRemoveLinkJre` to remove the link to JRE.  
  - If a UNIX or Linux endpoint does not have an appropriate copy of JRE installed, you can install it from the product CD, using the instructions provided on page 32.  
  Note that, if you are using JRE 1.3.1, it must be from a build dated September 14 2002 (or newer) on UNIX, or dated November 7 2002 (or newer) on Linux (running on i/Series or p/Series).  
• Endpoints running Solaris and using the Java 2 Platform Standard Edition (J2SE) must have the full set of required OS patches needed for the support of J2SE. Visit the following web site for details: [http://java.sun.com/j2se/1.3/install-solaris.html](http://java.sun.com/j2se/1.3/install-solaris.html) |
| Endpoints running Windows NT                 | • Windows Management Instrumentation (WMI) must be installed, before you can use a profile on the endpoint. WMI Version 1.5 is recommended. This software comes with Windows 2000, but must be installed manually on Windows NT endpoints. For more information, search for “WMI” on the following web site: [http://msdn.microsoft.com/](http://msdn.microsoft.com/) |
| Endpoints running Windows and using JavaScript™ resource models | • Microsoft Windows Script Host 5.6 is a prerequisite to run JavaScript resource models on a Windows endpoint. You can obtain the software from the following web site: [http://msdn.microsoft.com/](http://msdn.microsoft.com/) |
| Endpoints running Windows and deploying resource models using Java providers | • This prerequisite applies to Windows endpoints that deploy resource models which use user-defined Java providers, or resource models included in products built on top of IBM Tivoli Monitoring:  
  - Java Runtime Environment (JRE) 1.3.0 or 1.3.1 must be installed on the endpoint. It is recommended that you install and use the JRE available with the product CD. For details refer to section “Installing Java Runtime Environment” on page 31. |
Table 8. Prerequisite software (continued)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Required software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Endpoints running Windows and deploying the Network Interface Card resource model</td>
<td>• SNMP and the Network Monitoring Agent must be installed in the Network properties (SNMP is a prerequisite only on Windows NT systems). The Network Monitoring Agent must be installed prior to installing TCP/IP. The Network Interface performance object must be available from the performance monitor.</td>
</tr>
<tr>
<td>Endpoints running Windows and deploying the Parametric TCP/IP Ports or the TCP/IP resource models</td>
<td>• SNMP and TCP/IP must be installed in the Network properties.</td>
</tr>
<tr>
<td>Endpoints running Windows and deploying the Server Performance Prediction resource model</td>
<td>• The logical and physical disk counters must be enabled (diskperf -y). A reboot is necessary after the enablement. • SNMP and TCP/IP must be installed in the Network properties. The Network Monitoring Agent must be installed prior to installing TCP/IP. The Network Interface performance object must be available from the performance monitor.</td>
</tr>
<tr>
<td>Endpoints running AIX or HP and deploying the Network Interface or the Server Performance Prediction resource models</td>
<td>• The SNMP daemon must be active. • SNMP Version 3 is required on AIX 5.2 endpoints to run the Network Interface resource model.</td>
</tr>
<tr>
<td>Endpoints running OS/400</td>
<td>• OS/400 Option 30, QShell Interpreter, must be installed. • Patch 3.7-TMF-0043 must be installed and the endpoint updated with the new version. • Java Runtime Environment (JRE) 1.3.0 or 1.3.1 must be installed on the endpoint: – On OS/400 V5R1, this is installed in IBM Developer Kit for Java, *BASE and Option 5, 5722-JV1. Also the latest level of Java group PTF SF99069 (or of a group PTF that supersedes it) should be installed. In addition, PTF MF28579 must be installed on the operating system. – On OS/400 V5R2, this is installed in IBM Developer Kit for Java, *BASE and Option 5, 5722-JV1. Also the latest Java group PTF should be installed. – In addition, it is recommended that every two months you request the latest fix (or Group PTF) for the 5722-JV1, SLIC, and XPF products.</td>
</tr>
<tr>
<td>Use of Gathering Historical Data on Tier1 Tivoli management region server</td>
<td>• Patch 3.7-TMF-0035 must be installed, if the Tier1 Tivoli management region server is connected to Tier2 endpoints.</td>
</tr>
</tbody>
</table>
Table 8. Prerequisite software (continued)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Required software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use of the Data Logging facility on endpoints running Windows NT</td>
<td>• The ODBC driver for Microsoft® Access 2000 must be installed on the endpoint. If the endpoint does</td>
</tr>
<tr>
<td></td>
<td>not have Microsoft Access 2000 installed:</td>
</tr>
<tr>
<td></td>
<td>1. Run the mdac_typ.exe file that is provided with Microsoft Data Access Components (MDAC) 2.1 (or</td>
</tr>
<tr>
<td></td>
<td>later), at <a href="http://www.microsoft.com/data">http://www.microsoft.com/data</a></td>
</tr>
<tr>
<td></td>
<td>2. Install Jet 4.0 Service Pack 3. Note that starting from MDAC 2.6 the ODBC driver is not included,</td>
</tr>
<tr>
<td></td>
<td>therefore it must be installed separately.</td>
</tr>
<tr>
<td>Use of Tivoli Enterprise Console server</td>
<td>• Tivoli Enterprise Console server 3.7 plus patch 3.7-TEC-0004, or later</td>
</tr>
<tr>
<td></td>
<td>• If you want to send secure events to the Tivoli Enterprise Console server, the Adapter Configuration</td>
</tr>
<tr>
<td></td>
<td>Facility (ACF) must be installed on both the Tivoli server and Tivoli management gateways used to</td>
</tr>
<tr>
<td></td>
<td>distribute profiles to the endpoints.</td>
</tr>
<tr>
<td>Use of Tivoli Business Systems Manager Adapter</td>
<td>• Tivoli Monitoring must be installed on the gateway.</td>
</tr>
<tr>
<td></td>
<td>• Java Runtime Environment (JRE) must be installed on the gateway (see “Installing the Tivoli Business</td>
</tr>
<tr>
<td></td>
<td>Systems Manager Adapter” on page 85).</td>
</tr>
<tr>
<td></td>
<td>• The Tivoli Business Systems Manager patches 1.5-BSM-0010, 1.5-BSM-0016 should be installed on the</td>
</tr>
<tr>
<td></td>
<td>Tivoli Business Systems Manager server.</td>
</tr>
<tr>
<td></td>
<td>• The adapter cannot be installed on HP-UX gateways.</td>
</tr>
<tr>
<td>Use of Software Installation Service (SIS) for installing the product</td>
<td>• The following patches are required on SIS: 3.7-SISCLNT-0005 and 3.7-SISDEPOT-0005.</td>
</tr>
<tr>
<td>components</td>
<td></td>
</tr>
<tr>
<td>Upgrading Tivoli server or gateway from Tivoli Distributed Monitoring for</td>
<td>• Tivoli Distributed Monitoring for Windows 4.1 server/gateway component already installed.</td>
</tr>
<tr>
<td>Windows 4.1</td>
<td></td>
</tr>
<tr>
<td>Integration with Tivoli Data Warehouse</td>
<td>• Tivoli Enterprise Data Warehouse 1.1 Fixpack 2 (plus latest available patch), or a later version,</td>
</tr>
<tr>
<td></td>
<td>must be available.</td>
</tr>
</tbody>
</table>

Coexistence with other versions of Tivoli Monitoring

There are two families of monitoring products available from Tivoli:

• The Tivoli Monitoring family, which consists of:
  – Tivoli Monitoring 5.1.2
  – Tivoli Monitoring 5.1.1
  – Tivoli Monitoring 5.1
  – Tivoli Distributed Monitoring (Advanced Edition) 4.1, no longer supported
  – Tivoli Distributed Monitoring for Windows 3.7 and previous versions
  – Tivoli Manager for Windows NT, all versions

• The Tivoli Distributed Monitoring (Classic Edition) family, which consists of:
  – Tivoli Distributed Monitoring (Classic Edition) 3.7
  – Tivoli Distributed Monitoring 3.7 and previous versions
This section explains how these products can coexist with, or be upgraded to, IBM Tivoli Monitoring.

**Products within the Tivoli Monitoring family**

Products within the Tivoli Monitoring family must be upgraded to Version 5.1. None of the products of this family can coexist in the same Tivoli management region. Therefore, for example, you cannot install Tivoli Monitoring 5.1 in the same Tivoli management region where Tivoli Distributed Monitoring (Advanced Edition) 4.1 is installed.

Versions of Tivoli Manager for Windows NT and older versions of Tivoli Distributed Monitoring for Windows should be upgraded first to Tivoli Distributed Monitoring for Windows NT 3.7, then to Tivoli Distributed Monitoring (Advanced Edition) 4.1, then to Tivoli Monitoring 5.1, then to Tivoli Monitoring 5.1.1, and, finally, to Tivoli Monitoring 5.1.2.

Full details of the steps required to upgrade from Tivoli Distributed Monitoring for Windows NT, 3.7 to Tivoli Monitoring 5.1.2 are given in Appendix C, “Migration considerations,” on page 217.

The Workbench (or the Resource Model Builder) and the endpoint component of the product can coexist on the same system but only if they are aligned (same product version, release, and patch).

The Workbench and the Resource Model Builder can coexist on the same system. However, if, with the Resource Model Builder, you want to use resource models developed using the Workbench, you need to apply the import procedure described in section “Upgrading from Workbench to Resource Model Builder” on page 31.

**Products within the Tivoli Distributed Monitoring (Classic Edition) family**

All products within the Tivoli Distributed Monitoring (Classic Edition) family can coexist with Tivoli Monitoring 5.1 or with Tivoli Monitoring 5.1.1 or with Tivoli Monitoring 5.1.2. That is, they can be installed on the same server and gateways, and can access the same endpoints.

**Note:** There is no automatic migration from the Tivoli Distributed Monitoring (Classic Edition) product family to Tivoli Monitoring 5.1.2. See Appendix C, “Migration considerations,” on page 217, and, in particular, section “Migrating from Tivoli Distributed Monitoring (Classic Edition) to Tivoli Monitoring”.

**Backward compatibility**

Carefully consider the following information, which applies to backward compatibility:

- BAROC files (needed to enable monitoring of Tivoli Monitoring events on a Tivoli Enterprise Console server; refer to Chapter 5, “Integration with Tivoli Enterprise Console server,” on page 77 for details):
  - The BAROC files available with Tivoli Monitoring 5.1.2 or with Tivoli Monitoring 5.1.1 or with Tivoli Monitoring 5.1 can also be used with Tivoli Distributed Monitoring (Advanced Edition) 4.1 or with Tivoli Distributed Monitoring for Windows 3.7 Patch 3.
- The BAROC files available with Tivoli Distributed Monitoring (Advanced Edition) 4.1 can also be used with Tivoli Distributed Monitoring for Windows 3.7 Patch 3.
- The BAROC files available with Tivoli Monitoring 5.1.1 can also be used with Tivoli Monitoring 5.1.
- The BAROC files available with Tivoli Monitoring 5.1.2 can also be used with Tivoli Monitoring 5.1.1 or with Tivoli Monitoring 5.1.

- Resource models created using Workbench:
  - Resource models created using Tivoli Distributed Monitoring for Windows 3.7 Workbench can be used also with Tivoli Monitoring 5.1.2, with Tivoli Monitoring 5.1.1, with Tivoli Monitoring 5.1, or with Tivoli Distributed Monitoring (Advanced Edition) 4.1.
  - Resource models created using Tivoli Distributed Monitoring (Advanced Edition) 4.1 Workbench can also be used with Tivoli Monitoring 5.1 or with Tivoli Monitoring 5.1.1 or with Tivoli Monitoring 5.1.2 (although resource models for UNIX endpoints will need to be redistributed).
  - Resource models created using Tivoli Monitoring 5.1 Workbench can also be used with Tivoli Monitoring 5.1.1 or with Tivoli Monitoring 5.1.2
  - Resource models created using Tivoli Monitoring 5.1.1 Workbench can also be used with Tivoli Monitoring 5.1.2
  - As a general rule, resource models created using the Workbench can be modified using the Resource Model Builder but you must use the import procedure (see "Upgrading from Workbench to Resource Model Builder" on page 31).

**Note:** When deploying resource models using Tivoli Monitoring 5.1.2, you must carefully consider the following aspects:
- if you have modified any of the resource models that were included in older versions of the product, and you want to retain the modified resource models (rather than overwriting them with the new resource models provided with Tivoli Monitoring 5.1.2), make sure that you select the appropriate options during the installation.
- If you have created new resource models using providers that were included in older versions of the product, make sure that you replace the existing providers with the new ones included in Tivoli Monitoring 5.1.2 and that you rebuild your resource models using the new providers.

### Installing/upgrading the product

Tivoli Monitoring works in the Tivoli Management Environment® and can be installed and configured from any Tivoli management region server.

You can install Tivoli Monitoring 5.1.2 on any node either as a new product, or, if you have previously installed Tivoli Monitoring 5.1.1, you can upgrade to Tivoli Monitoring 5.1.2 using Tivoli Monitoring 5.1.1 Fix Pack 6 (5.1.1-ITM-FP06).

If you install Tivoli Monitoring 5.1.2 as a new product, you can do so in the following ways:
- Using the installation wizard
- Using Tivoli Software Installation Service
- From the Tivoli desktop
Before you begin

Please read the following notes before you begin to install or upgrade the product:

1. Decide on which servers or gateways you are going to install/upgrade the product. You must install the product on the server before installing/upgrading it on any gateway.

2. Ensure that the selected nodes meet the hardware requirements for the installation.

3. Ensure that you have the correct prerequisite software installed on each node.

4. Determine for each node if you need to install the product from scratch or if you can upgrade from a previous version of the product. Ensure that the selected nodes meet the software requirements for the installation or upgrade.

5. Ensure that you have Administrator or root privileges to perform the installation or upgrade.

6. Back up your database before you begin to install or upgrade the product.

**Note:** The IBM Tivoli Monitoring Task Library is also installed with the product. It contains a set of predefined tasks associated with the product. If you want to define additional tasks, you should create a separate task library, in order to avoid that the tasks you have created are lost if an uninstall operation is carried out.

A full description of how to carry out a fresh installation now follows; the product upgrade is described in Appendix C, “Migration considerations,” on page 217.

**Installing using the installation wizard**

The installation wizard (also known as Rapid Deployment) creates a complete Tivoli environment, if none exist, and installs Tivoli Monitoring. Use the installation wizard if you have no previous Tivoli software installations.

Refer to Appendix B, “Installing using the installation wizard,” on page 205 for details.

**Installing with Tivoli Software Installation Service**

Tivoli Software Installation Service (SIS) can install multiple Tivoli products on multiple systems in parallel. This Java-based product can install more products on more systems in much less time than the Framework’s install facility. SIS performs product prerequisite checks and, if defined, user-specified prerequisite checks, ensuring as few install failures as possible. In most cases, failures occur only when machines are turned off or removed from the network.

SIS also creates an installation repository into which you can import the installation image of one or more Tivoli products. You can import only those...
Installing from the Tivoli desktop

You can install the following product components from the Tivoli desktop:

**IBM Tivoli Monitoring, Version 5.1.2**

Includes the files that enable you to create and distribute profiles and run IBM Tivoli Monitoring commands. Install this on the Tivoli Management Region server and the gateways to which profiles will be distributed in your region (it must be installed on the server before installing it on any gateway).

**Tivoli Monitoring TBSM Adapter, Version 5.1.2**

Includes the files that enable you to use Tivoli Business Systems Manager to receive events sent from the resource models. Install this on the Tivoli management region gateway. If the Tivoli management region server also has the Tivoli Framework Gateway component installed on it to provide gateway services to directly-connected endpoints, install Tivoli Business Systems Manager Adapter also on the Tivoli management region server. For full instructions on how to install, configure, and use this component, see Chapter 6, “Integration with Tivoli Business Systems Manager,” on page 85.

**Tivoli Monitoring Gathering Historical Data Component, Version 5.1.2**

Includes the files that enable you to integrate Tivoli Decision Support for Server Performance Prediction (Advanced Edition) and Tivoli Data Warehouse with Tivoli Monitoring, using Tivoli Monitoring as a means of collecting information to be analyzed by those products. To install, configure, and use the Gathering Historical Data component (formerly called TDS Configuration component), refer to the following publications:

- Tivoli Monitoring Warehouse Enablement Pack: Implementation Guide
- Enabling an Application for Tivoli Enterprise Data Warehouse

**Tivoli Monitoring Tivoli Enterprise Data Warehouse Support Version 5.1.2**

Includes the files that enable you to integrate Tivoli Data Warehouse and the Tivoli Monitoring for Web Infrastructure products with Tivoli Monitoring. To install, configure, and use the Tivoli Enterprise Data Warehouse Support component, refer to Chapter 7, “Integration with Tivoli Data Warehouse,” on page 95.

To install the product from scratch from the Tivoli desktop, perform the following steps:
1. Insert the product CD-ROM into the CD-ROM drive.
2. Start the Tivoli desktop.
3. From the Desktop menu select **Install → Install Product**. The Install Product dialog opens.

4. If the IBM Tivoli Monitoring product components are displayed in the **Select Product to Install** scrolling list, go to step 8
   —OR—
   If the IBM Tivoli Monitoring product components are not displayed in the **Select Product to Install** scrolling list, continue with step 5
5. Click **Select Media**.
   The File Browser dialog opens.
6. Type or select the location of the Tivoli CD-ROM in the Path Name field by completing one of the following tasks:
   - Type the complete path name in the Path Name field.
   - Browse the file system by completing the following steps:
     a. In the Hosts scrolling list, select the host (or drive) on which the CD-ROM is mounted, by double-clicking it. Choosing a host updates the Directories scrolling list to show the directories (under root) of the host you selected.
     b. In the Directories scrolling list, double-click the directory that contains the install media. Choosing a directory updates the Files list. Descend the directory tree until you have found the appropriate directory.

7. Click Set Media & Close.
   The Install Product dialog reopens. The Select Product to Install scrolling list shows the products that are available for installation.

8. Select one of the following products from the Select Product to Install scrolling list:
   - IBM Tivoli Monitoring 5.1.2
   - IBM Tivoli Monitoring TBSM Adapter, Version 5.1.2

   Note: This component cannot be installed on HP-UX gateways.
   - IBM Tivoli Monitoring - Gathering Historical Data Component, Version 5.1.2
   - IBM Tivoli Monitoring - Tivoli Enterprise Data Warehouse Support, Version 5.1.2

   Note: If you are installing the Tivoli Business Systems Manager adapter, see “Installing the Tivoli Business Systems Manager Adapter” on page 85 for the additional information that needs to be provided. Similarly, if you are installing the Tivoli Enterprise Data Warehouse Support
component, see “Installing and configuring the Tivoli Enterprise Data Warehouse Support component” on page 99, and if you are installing the Gathering Historical Data Component, see the information given about the TDS Configuration component in Tivoli Decision Support for Server Performance Prediction: Release Notes.

9. Ensure that the appropriate host is displayed in the **Clients to Install On** scrolling list. If it is not displayed, select the server on which you want to install the selected product from the **Available Clients** scrolling list, and use the left arrow to move the selected server to the **Clients to Install On** scrolling list.

10. Click **Install** to begin installing the product on gateways.

   The Product Install dialog opens. It provides a list of the operations to be performed and warns of any problems you might want to correct before installing.

11. Review the status information and click **Continue Install**.

   The Product Install dialog informs you when installation is complete.

12. If you are going to send heartbeat events to the Tivoli Monitoring notice group, follow the instructions in “Adding the notice group” on page 31.

13. Click **Close**.

### Installing from the command line

Use the `winstall` command to install Tivoli Monitoring from the command line.

To install Tivoli Monitoring on a gateway from a CD-ROM, enter the following command:

```
winstall -c path/cdrom -i index_file managed_node
```

where:

- `path` Specifies the complete path to the `/cdrom` directory in the CD-ROM. (Use forward slashes to specify the path, regardless of the operating system you are running.)

- `index_file` Specifies the product index file from which Tivoli Monitoring is installed. This value can be one of the following:

  **DM512**
  
  The index file for Tivoli Monitoring

  **DM512TBS**
  
  The index for Tivoli Business Systems Manager Adapter (Tivoli Monitoring TBSM Adapter, Version 5.1.2). After you have installed this component using the CLI, use the `wdmconfig` command (and, in particular the `tbsma.jre_root` key) to specify the installation directory of Java Runtime Environment.

  **Note:** This component cannot be installed on HP-UX gateways.

  **DM512TDS**
  
  The index for Gathering Historical Data Component Version 5.1.2 (previously known as Tivoli Monitoring TDS Configuration)
**Note:** This product is required to enable integration with Tivoli Decision Support for Server Performance Prediction, refer to the Tivoli Decision Support for Server Performance Prediction: Release Notes. See the same publication also to understand how to create the RIM object using a shell script, after you have installed this component using the CLI.

**DM512TED**

The index file for Tivoli Enterprise Data Warehouse Support Version 5.1.2. To create the RIM object after you have installed this component using the CLI, refer to “Creating a RIM object using a shell script” on page 105.

`managed_node` The selected product will be installed on the gateway you specify. If you do not specify a gateway, the product is installed on all available clients.

If you are going to send heartbeat events to the Tivoli Monitoring notice group, follow the instructions in “Adding the notice group” on page 31.

For more information about all the parameters you can use with the `winstall` command, refer to the Tivoli Management Framework Reference Manual.

**Upgrading from Tivoli Monitoring Version 5.1.1**

To upgrade from Tivoli Monitoring 5.1.1 to Tivoli Monitoring 5.1.2, you must install Tivoli Monitoring 5.1.1 Fix Pack 6 (5.1.1-ITM-FP06).

Tivoli Monitoring 5.1.1 Fix Pack 6 contains exactly the same code level of Tivoli Monitoring 5.1.2 and registers you as a Tivoli Monitoring 5.1.2 customer.

To install Tivoli Monitoring 5.1.1 Fix Pack 6, follow the instructions documented in the corresponding ReadMe file.

**Installing a default resource model**

If you want to install a new default resource model, or an existing default resource model that has been modified with Tivoli Monitoring Workbench or with Tivoli Monitoring Resource Model Builder, enter the following command from command line:

```
wdmrm –add resource_model_tarfile
```

For more information, refer to Chapter 8, “Commands,” on page 109.

If you want to remove an existing default resource model, enter the following command from command line:

```
wdmrm –remove resource_model
```

For more information, refer to Chapter 8, “Commands,” on page 109.

If you want to create a new default resource model or modify an existing one, you can do it using the Resource Model Builder. For more information on how to do it, see IBM Tivoli Monitoring: Resource Model Builder User’s Guide.
Upgrading from Workbench to Resource Model Builder

Starting from IBM Tivoli Monitoring 5.1.1 Fix Pack 5 (5.1.1-ITM-FP05), the Workbench is replaced by the new Resource Model Builder component. For more information, refer to the IBM Tivoli Monitoring: Resource Model Builder User’s Guide.

If you want to use the Resource Model Builder with any resource models created using the Workbench, you must do the following:

1. Install the Resource Model Builder by running ITM_Workbench_WSWB_Setup.exe and following the installation wizard steps. Note that the Workbench and Resource Model Builder components can coexist on the same system.
2. Start the Resource Model Builder.
3. Create a project to hold the imported resource models.
4. Import the resource models by choosing File → Import and selecting Old ITM Workbench Files.

Adding the notice group

If you want to send events to the Tivoli Monitoring notice group, use the following procedure to create the link with the notice group:

1. From the Tivoli Desktop select the Administrators icon. The Administrators dialog opens.
2. Right-click on the Administrators icon for your region and select Edit Notice Group Subscriptions. The Set Notice Group dialog opens.
3. Select the Tivoli Monitoring notice group and click on the left arrow to move it to the Current Notice Groups pane.
4. Press Change & Close to set the notice group definitions to include the selected notice group.

Installing Java Runtime Environment

Various components of Tivoli Monitoring require Java Runtime Environment (JRE), Version 1.3.0 or 1.3.1 (see Table 8 on page 19 for full details).
Note: On OS/400 systems, JRE is obtained by installing IBM Developer Kit for Java and the latest Java group PTF (see Table 8 on page 19 for details).

There are three available installation methods:

- JRE is available on the product CD for installation using the Tivoli Software Installation Service (SIS).
- JRE is also available on the product CD in compressed format, for manual installation or installation through the wdmdistrib –J command (for details on this command, refer to Chapter 8, “Commands,” on page 109).
- On UNIX and Linux endpoints, if the endpoint already has an appropriate version of JRE installed, you need only link the product component to the existing JRE using a task provided with the product.

Important Installation Notes:

1. If JRE is not installed or linked as described here, the product fails on the system in question.
2. The use of an IBM version of JRE is recommended for support purposes.
3. All versions of the JRE on the CD are supplied by IBM, except for the HP-UX version.
4. Prerequisites for JRE can be found on the appropriate readme file in the JRE directory on the product CD.

Installation options

The full details of the installation options are as follows:

Installation using SIS
You can use SIS to install JRE. The JRE product name in SIS is ITM 5.1.2 - JRE 1.3.0.

Installation using the wdmdistrib command
To install JRE from the compressed files on the Tivoli Monitoring 5.1.2 Tools CD, you can use the wdmdistrib –J command (for details on this command, refer to “wdmdistrib” on page 129).

Note that, if you want to install JRE on an HP-UX endpoint using the wdmdistrib command, you first need to create a compressed file (jre13.tar.gz) containing the JRE and then deliver the compressed file to the HP-UX endpoint using wdmdistrib –J. The following is an example of how this can be done (lines that start with # represent the shell command line on an HP-UX system, lines that start with ## are comments):

## Go to the directory where you placed the JRE and tar up that directory

```
# /opt cd java1.3
hpdsps2/opt/java1.3 cd ls
COPYRIGHT demo jre
LICENSE include lib
bin java.os11.release.notes src.jar
# tar cvf /tmp/jre13.tar .
## Then gzip the file.
# cd /tmp
# gzip jre13.tar
## Then use ftp (in binary mode) to send the jre13.tar.gz to your Tivoli server
## On the Tivoli server (in a locally mounted filesystem, not nfs mounted),
place that file in your JRE directory, under an hpux10 subdirectory
# /export/home/JRE # mkdir hpux10
# /export/home/JRE # ls
aix4-r1 hpux10 linux-ix86 linux-s390 solaris2
w32-ix86
# gollum_/export/home/JRE # ls -l hpux10
```

32 IBM Tivoli Monitoring: User’s Guide
Now you can run the wmdistrib -J command

```
# wmdistrib -J "pwd" -p hpdp2 @Endpoint:hpdp2
```

## The `pwd` will expand to the current directory supplying the argument for the -J option
## The -p hpdp2 indicates that a Tmw2kProfile is to be distributed to the hpdp2 endpoint
## The JRE will be distributed, unzipped, and untarred on the endpoint and placed in the $LCF_LOAD_DIR/../JRE/DMAE/ directory

### Manual installation

To install JRE from the compressed files on the Tivoli Monitoring 5.1.2 Tools CD, follow these steps:

1. Copy the jre13.tar.gz file from the directory on the Tools CD that applies to the operating system where JRE is to run, to the directory where you want to install JRE:
   - AIX: JRE/aix4-r1
   - Linux: JRE/linux-ix86
   - Linux S/390: JRE/linux-s390
   - Solaris: JRE/solaris2
   - HP-UX: Refer to the installation instructions for HP-UX Runtime Environment for Java 2 Version 1.3.1, which can be found at the following URL:
   - Windows: JRE/w32-ix86

   See the specific ReadMe file included in each directory to understand the system requirements associated with each platform.

2. From the directory where you copied the jre13.tar.gz file, issue the command:
   ```
   gzip -dc jre13.tar.gz | tar xv -
   ```

3. If JRE is being installed for a UNIX/Linux endpoint, run the DMLinkJre task, as described below.

### Link to an existing version using the DMLinkJre task

On Windows, UNIX, or Linux endpoints, if you already have the appropriate version of JRE installed on the target system, you can use a Tivoli task to link the product to JRE.

1. From the Tivoli Desktop select Desktop → TMR Connections → Top Level Policy Regions. The Top Level Policy Regions dialog opens.
2. Double-click the TivoliDefaultMw2kRegion icon. The Policy Region dialog for that policy region opens.
3. Double-click the Tivoli Monitoring Tasks icon. The Task Library dialog for that library opens.
4. Run the DMLinkJre task. The Execute Task dialog opens.
5. Select the endpoint or endpoints on which you want to link the JRE, and click Execute (you can also identify endpoints using one or more profile managers).

**Note:** Ensure that the path where JRE is physically installed is the same on all selected endpoints.
6. On the DMLinkJre dialog, enter the complete path where JRE is installed (not including the /bin directory). Click Set & Execute. JRE is linked to the product on all selected endpoints.

---

**Enabling Tivoli Monitoring to work with Tivoli Data Warehouse using Gathering Historical Data**

After installing the Gathering Historical Data component, to enable Tivoli Monitoring to collect data that can be used by Tivoli Data Warehouse:

1. Create the RIM object and the database schema, as described in *Tivoli Decision Support for Server Performance Prediction: Release Notes*.

2. From a command prompt on a Tivoli server where the Tivoli environment was set, switch to the $BINDIR/TME/Tmw2k/TDS/rdbcfg directory and run the following script:
   
   twh_enable.sh

3. From a command prompt on the RIM host where the Tivoli environment was set, switch to the $BINDIR/TME/Tmw2k/TDS/rdbcfg directory and run the following script:
   
   run_query.sh twh_enabl_update.extension

   where extension is either db2, mssql, ora, syb, or inf, depending on the RDBMS installed.

   **Note:** If you are using a DB2® or Informix™ RDBMS, the script must be run from a DB2 or Informix command line on the database server. This is necessary because you cannot modify the DB2 database from the DB2 Client Command Line Processor without an existing database connection, and because Informix uses the dbaccess facility, which is shipped as part of the Informix server.

---

**Uninstalling the product**

This section describes how to uninstall the product components from endpoints, gateways, and servers, and how to remove the automatic endpoint reboot capability.

**Note:** The Tivoli Monitoring Task Library is also uninstalled with the product. If it contains any user-defined tasks, they are lost. Therefore you should create any user-defined tasks in a separate library.

To uninstall the product follow either of these procedures:

1. Uninstall the endpoint components.

2. Uninstall from server and gateways.

   or:

   1. Uninstall from server and gateways and automatically uninstall all registered endpoints.

---

**Uninstalling the endpoint components**

Tivoli Monitoring provides a Tivoli task to remove Tivoli Monitoring from endpoints.

The DMEndpointUninstall task performs the following steps:
1. Removes the endpoint from the cache on the managed node.
2. Stops the application if it is running.
3. Removes from the endpoints all the Tivoli Monitoring files that were downloaded to the endpoints with the Tmw2k profile. Removes also the endpoint database.
4. Where appropriate, removes the application keys from the Windows registry.

Run this task on the managed node to which the endpoint is associated.

To uninstall the product from an endpoint, perform the following steps:

1. From the Tivoli desktop select **TMR Connections → Top Level Policy Regions**. The Top Level Policy Regions dialog opens.
2. Double-click the **TivoliDefaultMw2kRegion** icon. The Policy Region dialog for the default policy region opens.
3. Double-click the **IBM Tivoli Monitoring Tasks** icon. The Task Library dialogue for IBM Tivoli Monitoring Tasks opens, showing all available tasks.
4. Double-click the **DMEndpointUninstall** icon. The Execute Task dialog opens.
5. Select the **Output Destination** where the system output from the uninstall is to be displayed or saved.

6. If you want to perform the uninstall on one or more specific endpoints attached to the managed node, select the managed node (not the endpoint) in the **Available Task Endpoints** list. Click the left arrow to move the managed node to the **Selected Task Endpoints** list.

7. Click **Execute** to uninstall the product. The DMEndpointUninstall dialog opens.
8. Enter a list of endpoints or -g followed by the gateway name if the uninstall has to be executed on all endpoints attached to that gateway. Deselect the Print Nested Task Output option if you do not want to view the output produced by the nested task (DMEndpointUninstall invokes the DMRemoveFiles task). You can also change the timeout for the DMRemoveFiles task by specifying the appropriate value in the Nested Task Timeout field. Click Set & Execute.

9. When the uninstall is complete, click Close.

**Uninstalling from servers and gateways**

The Tivoli Management Framework provides the `wuninst` command to remove Tivoli applications from a specified node or from the entire region. You can use the `wuninst` command with product tags that are specific to Tivoli Monitoring to remove the product from any system in your environment or from the entire region.

In particular, use the `-rmeps` option to automatically uninstall also all registered endpoints.

When using the `wuninst` command, make sure you use it in this sequence:

1. Uninstall any Tivoli Monitoring components (for example: the Gathering Historical Data Component, or the Tivoli Business Systems Adapter)
2. Uninstall the Tivoli Monitoring base component (core product)

To uninstall the Tivoli Monitoring components from servers or gateways enter:

```
wuninst tagname destination_target -rmfiles
```

To uninstall the Tivoli Monitoring base component from servers or gateways enter:

```
wuninst tagname destination_target -rmfiles -rmeps
```

Where:

- `tagname` is one of the registered product tags for IBM Tivoli Monitoring that are provided by Tivoli.

**Note:** These tags do not indicate the version or release of the product:

- `DM_Advanced_Edition_TDS`, to indicate Tivoli Decision Support for Server Performance Prediction.
- `DM_Advanced_Edition_TBSMA`, to indicate Tivoli Business Systems Manager Adapter.
- `TMNT_3.6.2`, to indicate Tivoli Monitoring.
- `ITM_TEDW`, to indicate Tivoli Enterprise Data Warehouse Support.

`destination_target` is the gateway or server from which you want to remove the product. If you specify a server, the product is uninstalled from all its gateways.

For more information about command line syntax and usage of the `wuninst` command, refer to the Tivoli Management Framework Reference Manual and the Tivoli Management Framework Release Notes: Server, Gateway, and Endpoint.
Chapter 3. Using the product

This chapter describes the purpose of profiles in the Tivoli environment and presents the range of customization options you can use to fine-tune resource models to ensure optimal availability of resources.

These options are discussed in the context of using the graphical user interface (GUI) to perform them. However, many users want to use the commands provided in the command line interface (CLI), and there are some tasks that can only be performed using the CLI, such as adding new resource models at the Tivoli server. All users are recommended to read through this chapter to understand the concepts of using the product and the structure of profiles and resource models. Those who want to use the command line interface can find full details of the commands in Chapter 8, “Commands,” on page 109. In addition, references are made in the various sections of this chapter to the corresponding commands.

This chapter provides details of how to perform the following tasks:

- Setting up managed resources
- Creating profile managers and profiles
- Using default resource models
- Adding a default resource model to a profile
- Adding a customized resource model to a profile
- Customizing a resource model
  - Customizing thresholds and event rules: it includes:
    - Specifying built-in actions for an event
    - Specifying tasks for an event
  - Customizing parameters
  - Customizing scheduling information: it includes:
    - Setting the monitoring period
    - Creating schedule rules
  - Customizing data logging information
- Setting distribution options for a profile
- Profile distribution using MDist
- Deleting profile copies during an unsubscribe
- Managing profiles and resource models at endpoints
- Managing Tivoli Monitoring at gateways
- Using Tivoli Monitoring tasks

Overview

Before you can monitor resources at an endpoint, you must create one or more monitoring profiles. You then add the required resource models to the profile. You can add resource models with the default values preconfigured by Tivoli, or you can customize the resource models as required.
Profiles

A profile is a collection of information corresponding to a Tivoli application resource. In most Tivoli applications, a profile contains information that is specific to a particular application or a particular database grouping.

Tivoli Monitoring profiles enable you to define resource models and distribute these resource models to subscribing endpoints. When you first set up Tivoli Monitoring, no monitoring occurs until you distribute the profile. Resource models can be added to a profile, deleted from it, or modified. These changes to the profile do not take effect until you redistribute the profile. The subscribers can be endpoints and other profile managers.

Key points about profiles

There is a relationship between profiles, profile managers, policy regions, endpoints and resource models:

- Profile managers are created within a policy region and must be a managed resource of that region. See “Setting up managed resources.” for more details.
- Profile managers contain profiles; they also administer profiles and subscriber lists (groups of endpoints). See “Creating profile managers and profiles” on page 41 for more details.
- Profiles include one or more resource models, which can be customized or can be defined using the defaults. If you want to modify the default resource models, see “Using default resource models” on page 43.

Setting up managed resources

Before creating profile managers and profiles for Tivoli Monitoring, you must make the Tivoli Monitoring profile available for use as a managed resource.

Note: This procedure must be performed only once for each policy region in which Tivoli Monitoring profiles are used.

To set up managed resources, perform the following steps:

1. In the Policy Region dialog, select Properties → Managed Resources.

The Set Managed Resources dialog opens.
2. Ensure that the **Tmw2kProfile** is displayed in the **Current Resources** list.
3. Click **Set & Close**.
   The Policy Region dialog reopens.

### Creating profile managers and profiles

To create a new profile, perform the following steps:

1. From a Policy Region dialog, select **Create → Profile Manager**.
   The Create Profile Manager dialog opens.

   ![Create Profile Manager](image)

   a. Type the name of the new profile manager.
   b. If you want to be able to push the profile to endpoints, click **Dataless Endpoint Mode**.

      **Note:** You cannot subscribe other profile managers to the profile manager using the dataless endpoint mode.
   c. Click **Create & Close**.
      The new profile manager is created in the Profile Manager dialog.

2. In the Policy Region dialog, double-click the **Profile Manager** icon.
   The Profile Manager dialog opens.
3. Select Create → Profile.
   The Create Profile dialog opens.

4. Type the name of the new profile in the Name/Icon Label field.
5. Select Tmw2kProfile from the Type list.
6. Click Create & Close.

The Profile Manager now contains an icon for the new profile.

If you want a profile to be able to send TEC events or TBSM events, you must configure the profile properties. See “Identifying the Tivoli Enterprise Console server to Tivoli Monitoring” on page 77 for details.

Using the CLI to manage profiles at the server

The commands for managing profiles at the server are as follows:

- **wdmloadprf**
  Loads and updates profiles at a Tivoli management region server

- **wdmdumpprf**
  Writes the full details of a profile to the standard output

- **wdmeditprf**
  Allows you to edit the definition of a profile, including all resource model details

- **wdmdistrib**
  Allows you to distribute profiles to the endpoints

Details of all commands can be found in Chapter 8, “Commands,” on page 109.

Using default resource models

As stated above, a profile includes one or more resource models, which can be customized or can be defined using the defaults. Default resource models can be changed, and new default resource models created, using the Tivoli Monitoring Resource Model Builder. Tivoli Monitoring also includes a command that makes those new or changed models available at the server, as follows:

- **wdmrm**
  Adds, lists, or removes a specified default resource model at the Tivoli management region server or managed node/gateway from where it is issued. It also adds the NLS catalog to an already installed default resource model.

Details of all commands can be found in Chapter 8, “Commands,” on page 109.

Adding a default resource model to a profile

To add a resource model to a profile using its default values, perform the following steps:

1. Open the Profile Manager dialog.
2. Double-click the profile icon to which you want to add a resource model.
The Tivoli Monitoring Profile dialog opens.

3. Click **Add with Defaults**.

   The Add Default Models to Profile dialog opens.

4. Select the required resource models to add.

5. Click **Add & Close**.

   The resource models are added to the Tivoli Monitoring Profile dialog.

**Using the CLI to add a default resource model to a profile**

Use the following command to add a default resource model to a profile:

```shell
wdmeditprf
```

Allows you to edit the definition of a profile, including all resource model details.
If the command is submitted with no arguments related to resource model details, the model is added with the defaults. Details of the command can be found in “wdmeditprf” on page 138.

Adding a customized resource model to a profile

To customize the basic settings of a resource model and add it to a profile, perform the following steps:

1. Double-click the profile icon to which you want to add a customized resource model.
2. From the Tivoli Monitoring Profile dialog, click Add. The Add Resource Model to Profile dialog opens.

3. Select the resource model category from the Category list. The defaults supplied with the product contain the following categories:

   **UNIX – Linux**
   Makes selectable in the Resource Model list only the resource models appropriate to all UNIX and Linux platforms.

   **Windows**
   Makes selectable in the Resource Model list only the resource models appropriate to all Windows platforms.
Customizing IBM Tivoli Monitoring: a resource model

Solaris
Makes selectable in the Resource Model list only the resource models appropriate to all Solaris platforms.

OS400
Makes selectable in the Resource Model list only the resource models appropriate to all OS/400 platforms.

However, you should note that these categories can be modified and other categories can be added using the Tivoli Monitoring Resource Model Builder.

4. Select the required resource model from the Resource Model list.
5. To view all the settings of a resource model, select the resource model.
   An explanation of the resource model and its parameters, together with help on setting parameters is displayed in the Description text box.
6. In the Cycle Time text box, set the frequency with which the resource model monitors the data.
7. Change any of the Threshold values as required:
   a. Select the Threshold Name that you want to change.
      A description of the threshold name is displayed. The currently assigned threshold value for the threshold name is also displayed above the description.
   b. Change the currently assigned threshold value to one that matches your requirements.
   c. Click Apply.
      The new value is set and displayed in the Threshold Value text box.
8. If required, click the appropriate buttons to modify the Indications (see “Customizing thresholds and event rules” on page 47) and Parameters (for parametric resource models only - see “Customizing parameters” on page 54) or add Schedule information (see “Customizing scheduling information” on page 54). If you want to, you can leave these items to take their default values and customize them later (see “Customizing a resource model”).
9. Click Add & Close to save any changes you have made.
   The Tivoli Monitoring Profile dialog now shows the customized resource model.

Using the CLI to add a customized resource model to a profile
Use the following command to add a customized resource model to a profile:

```
wdmeditprf
```

Allows you to edit the definition of a profile, including all resource model details.

The command can be submitted to add the resource model, changing any combination of the resource model details. Details of the command can be found in “wdmeditprf” on page 138.

---

Customizing a resource model

You can customize resource models by doing any or all of the following:

- Customizing indications and event rules, including:
  - Specifying built-in actions
  - Specifying corrective tasks
- Customizing parameters (parametric resource models only)
- Customizing scheduling information
Customizing data logging information

**Customizing thresholds and event rules**

To customize the rules that determine how indications generated by a resource model are aggregated into events, perform the following steps:

1. Open the Tivoli Monitoring Profile dialog.
2. Select the resource model you want to customize.
3. Click **Edit**.

   The Edit Resource Model dialog opens.

4. Click **Indications**.

   The Indications and Actions dialog opens.
5. The indications appropriate to the selected resource model are displayed. If you want to change the values of the event associated with an indication, select the indication.

The parameters you can change are as follows:

- **Number of Occurrences**
  This number represents how many consecutive times the problem reported by the indication occurs before sending an indication. You can change this value to specify how persistent a problem is before an indication is triggered. The number of occurrences must be greater than zero.

- **Number of Holes**
  This is used to determine how many cycles that do not produce an indication can occur between cycles that do produce an indication, in order for the occurrences of the indication to still be considered consecutive. Each cycle without an indication is termed a hole. The number of holes must be greater than, or equal to, zero. Thus, an event that has a value of 2 holes means that if up to 2 cycles without an indication occur between any two cycles with an indication, the cycles with an indication are considered consecutive.

Use the value for the Number of Holes in conjunction with the Number of Occurrences parameter (see the above paragraph) and the Cycle Time (defined on the Add Resource Models to Profile dialog) to define a time window for the generation of an event. If, for example, you define Cycle Time as 10 seconds, Number of Occurrences as 5, and Number of Holes as 2, the time that must elapse before an event is created is between 40 and 120 seconds. The minimum elapsed time is the number of occurrences multiplied by the cycle time. The maximum time window assumes that the maximum number of holes occurs between each pair of occurrences, and is determined by the equation:

\[ TW = CT \times ((Oc - 1) + (H \times (Oc - 1)) \]

where:

- **TW** Specifies the time window being calculated
- **CT** Specifies the cycle time
- **Oc** Specifies the number of occurrences
H Specifies the number of holes

- **Send TEC Events**
  Select this check box to specify that if an event occurs, a Tivoli Enterprise Console event is to be sent to the specified event server (see "Identifying the Tivoli Enterprise Console server to Tivoli Monitoring" on page 77).

- **Send to TBSM**
  Select this check box to specify that if an event occurs, it is to be sent to the Tivoli Business Systems Manager’s Common Listener.

- **Severity**
  Defines the severity of the indication. It can have either of the following values: fatal, critical, warning, harmless, or minor. Select the appropriate severity from the drop-down list.

- **Clearing**
  Defines whether a clearing event should be sent when the circumstances that generated the event are no longer present. Select the clearing policy from the drop-down list.

6. If you have previously deleted any of the built-in actions, the **Built-In** button is enabled; see “Specifying built-in actions for an event”

7. If you want to specify tasks to be carried out when an event is recognized, click the **Tasks** button. See “Specifying tasks for an event” on page 50. If corrective tasks have already been defined you can remove them from the event definition.

8. When you have made the required changes to an indication, click **Apply** to have your changes accepted. Repeat the procedure if you need to change any other indication. When you have made the required changes to all indications, click **Apply Changes & Close**.

**Specifying built-in actions for an event**

Built-in actions are available for some events for some resource models, and are performed when an event is triggered. They are listed in the Action List panel of the Indications and Actions dialog.

**Editing a built-in action:** To change the parameters of an action, perform the following steps:

1. Select the action to edit in the Action List panel of the Indications and Actions dialog.

2. Click **Edit**.
   The Edit Current Action dialog opens.
3. In the Retry field, modify the number of times the action should try to perform when an event is triggered. For example, if you set the value to three, the product tries to perform the action three times only. If you set this value to 0, the product continually tries to perform the action until it is successful.

4. Click Close if you want to leave the original value, or click Modify & Close to modify the selected Action with the value shown and close the dialog.

Removing a built-in action: To remove a built-in action, select it in the Action list and press Remove.

Adding a built-in action: You can only add a built-in action if you have previously removed one. Only in these circumstances the Built-In button in the Indications and Actions dialog becomes active.

To reinstate a previously removed built-in action, perform the following steps:

1. Click Built-In on the Indications and Actions dialog.

   The Add Action dialog opens.

   Select the action to run when an event occurs. The dialog shows the name of the currently selected indication. It also displays a list of selectable actions.

2. Select the required action to be run when an event occurs. In the Retry field, type the number of times the action tries to perform the action. For example, if you set the value to three, the product tries to perform the action three times only. If you set this value to 0, the product continually tries to perform the action until it is successful. Click Apply to save the new value.

3. Click Add to add the selected action with the values shown and keep the dialog open, or click Add & Close to add the selected Action with the values shown and close the dialog.

Specifying tasks for an event

Tasks can be triggered when an event occurs. The target for the execution of the task is the endpoint that produced the event.
In the Indications and Actions dialog, perform the following steps to specify corrective or reporting tasks when an event is triggered:

1. Select an event for which a task is to be triggered.
2. Click Tasks to add a task that will be performed when the event is triggered. The Tasks dialog opens.

3. Double-click a task library in the Libraries list. The tasks contained in the library are displayed in the Tasks list.
4. Double-click a task in the Tasks list, then click Change & Close to add the task to the Action list in the Indications and Actions dialog.

**Note:** A task is triggered when the event occurs. Subsequent occurrences of the same event do not cause the task to be triggered again unless a clearing event is enabled to close the original error event.

You can send an event (TMWTaskResult, associated with the submission of a set of tasks) to the Tivoli Enterprise Console server. If the submission of the tasks was successful, the event message is the following:

*The tasks associated to the event EventTriggerName have been submitted for execution*

The event also contains the name of the tasks, the task libraries, and a list of return codes. The return codes are not related to the execution of each individual task,
but to the result of the submission of the task to the task engine (a return code equal to 1 indicates that the task was not submitted; a return code equal to 0 indicates that the task was submitted).

If the submission of all tasks was not successful (because, for example, the task engine is not running) then all return codes are set to -1 and the event message is the following:

"The tasks associated to the event EventTriggerName could not be submitted for execution"

**Sending an e-mail in response to an event:** In the Tasks dialog, perform the following steps:

1. Double-click **IBM Tivoli Monitoring Utility Tasks** in the Libraries list. The tasks contained in the IBM Tivoli Monitoring Utility Tasks library are displayed in the Tasks list.
2. Double-click `dm_mn_send_email` in the **Tasks** list. The Configure Task dialog opens.

![Configure Task Dialog](image1)

3. Specify the appropriate e-mail address. Refer to Chapter 6. “Configuring for SMTP E-Mail” of the *Tivoli Management Framework: Installation Guide* for information on configuring Tivoli for use with e-mail services.

4. Click **Change & Close** to add the task to the Action list.

**Note:** The task cannot be run manually. It must always be associated to an event.

**Sending a notice in response to an event:** In the Tasks dialog, perform the following steps:

1. Double-click **IBM Tivoli Monitoring Utility Tasks** in the **Libraries** list. The tasks contained in the IBM Tivoli Monitoring Utility Tasks library are displayed in the **Tasks** list.

2. Double-click `dm_mn_send_notice` in the **Tasks** list. The Configure Task dialog opens.

![Configure Task Dialog](image2)

3. Specify the appropriate parameters in the Configure Task dialog (you may run the Tivoli Framework command `wlsnotif -g` to see the available Notice Groups, while Priority is either Critical, Error, Warning, Notice, or Debug), then click **Change & Close** to add the task to the Action list.

**Note:** The task cannot be run manually. It must always be associated to an event.
Customizing parameters

In parametric resource models, you can modify any parameters to customize and optimize the monitoring process. For more information about the parameters, refer to IBM Tivoli Monitoring: Resource Model Reference Guide.

To customize parameters for a parametric resource model, perform the following steps:

1. In the Edit Resource Model dialog, click the Parameters button.
   The Parameters dialog opens.

2. Select the type of parameter from the Name drop-down list.
3. Modify the parameter values as required.
4. When you have added or deleted the values as required, click Apply Changes and Close to apply and save the changes.
   The Parameters dialog closes.

Customizing scheduling information

Customizing the scheduling of the resource model is a two-stage process:

1. Set the schedule times to control the time period within which the resource model monitors resources.
2. Define the schedule rules to set the specific time intervals within the overall monitoring period when data is to be collected.

**Note:** The frequency with which data is collected is determined by the cycle time of the resource model.

In the Edit Resource Model dialog of the selected resource model click Schedule. The Scheduling dialog opens.
This dialog has the following group boxes:

**Schedule**
For setting the data collection period. By default, all resource models are set to always collect data.

**Schedule Rules**
For managing time intervals during which the resource model is active.

**Rule Editor**
For creating and editing schedule rules.

### Setting the monitoring period
To set the monitoring period, perform the following steps:

1. In the Schedule group box, deselect **Always**.
2. Set a start and stop date to define the monitoring period.
3. If required, add one or more schedule rules that determine time periods on selected days during which monitoring takes place.

For information about creating schedule rules, see “Creating schedule rules.”

### Creating schedule rules
The schedule controls the overall period for monitoring a resource and the timing windows in which resource model data is to be collected. The overall collection period can be divided into active and inactive intervals by applying one or more rules.
To create a schedule rule, perform the following steps:

1. In the Scheduling dialog, click the **New Rule** button.
2. In the **Rule Editor** group box, type a name for the rule.
3. Specify the day or days on which you want the collections to be active during the collection period, by selecting one or more items in the list of weekdays.
   
   **Tip:** Use the **Shift** key or **Ctrl** key as necessary, to select more than one day from the list.

4. Set start and stop times for the collection activity or select the **All Day** check box. Times are always interpreted as local time where the endpoint engine is running, so that, for example, setting a time interval of 08:00 to 13:00 ensures that monitoring takes place between those times in all time zones to which you distribute the profile.

5. Click **Set Rule**.
   
   Your new rule is added to the **Schedule Rules** list.

   You can create another rule for addition to the list or delete an existing rule from the list.

   **Tips:** To display the details of any rule, select the rule in the **Schedule Rules** list. Its settings are displayed in the **Rule Editor** group box.

   If the Schedule Rules list contains more than one schedule rule, all the time intervals are respected, and rules are combined by adding together the time periods they define. For example, if you specify a rule that requests monitoring between 8:00 and 14:00 every day and another that requests all-day monitoring on Fridays, the sum of the two rules gives all-day monitoring only on Fridays, and monitoring between 8:00 and 14:00 on all other days. If the second rule instead requested monitoring from 12:00 to 18:00 on Fridays, the sum of the rules would give monitoring between 8:00 and 18:00 on Fridays and between 08:00 and 14:00 on all other days.

6. To save your rule, click **Add & Close**.
   
   The Scheduling dialog closes.

### Customizing data logging information

Using this feature, you can log data collected by a resource model and write it in a local database. Then you can view it through the Web Health Console. In the database, you can choose to store one of the following types of data:

**Raw data**

Data is written exactly as it is collected by the resource model. All the monitored values are collected and copied in the database.

**TEDW data**

Data is collected and copied in the database for later use by Tivoli Data Warehouse.

**Aggregated data**

Data is collected and aggregated at fixed intervals you define (**Aggregation Period**). Then only the aggregated values are written in the database. The aggregated data can be calculated on the basis of one or more of the following options:

- Maximum
- Minimum
- Average
Note that the Web Health Console can only use Raw Data or Aggregated Data, while Tivoli Data Warehouse can only use TEDW Data.

By default, data logging is not enabled.

To add data logging information to a resource model, perform the following steps:

1. In the Edit Resource Model dialog, click Logging. The Logging dialog opens.

2. In the Data Logging Settings group box, select the Enable Data Logging check box. This enables logging of all data collected by the resource model to a database.

3. Choose the type of data you want to store in the database by selecting one of the following check boxes:
   - Raw Data
   - TEDW Data
   - Raw Data and TEDW Data
   - Aggregated Data

   Note that Raw Data and Aggregated Data can only be used by the Web Health Console, while TEDW Data can only be used by Tivoli Data Warehouse. Therefore if you have selected both the Raw Data and TEDW Data check boxes, data will be logged for both the Web Health Console and Tivoli Data Warehouse.

4. To specify the aggregation rule to be applied to the data before it is written to the database, perform the following steps:
   a. To specify the period for which data is to be aggregated, set Hours and Minutes of the Aggregation Period to the required values.
   b. Select one or more of the following functions to be performed on the numerical data collected during the aggregation period before it is written to the database:
Maximum  Calculates and logs the peak value in each aggregation period
Minimum  Calculates and logs the lowest value in each aggregation period
Average  Calculates and logs the average of all values in each aggregation period

5. To specify the period for which data is to be stored in the database, set Hours and Minutes of the Historical Period to the required values.

6. To save the changes, click Apply Changes and Close.
   The Logging dialog closes.

Using the CLI to customize a resource model for a profile

Use the following command to customize a resource model for a profile:

**wdmeditprf**  Allows you to edit the definition of a profile, including all resource model details.

The command can be submitted to edit the resource model for a selected profile, changing any combination of the resource model details. Details of the command can be found in "wdmeditprf" on page 138.

Setting distribution options for a profile

To set the distribution options for the profile, perform the following steps:

1. In the Profile Manager dialog, open a profile to be distributed.
   The Tivoli Monitoring Profile dialog opens.

2. Click Profile → Distribute.
   The Distribute Profile dialog opens.

3. Select one of the Distribute To options based on the following descriptions:
   • Next level of subscribers
     Distributes the profile only to the subscribers named in the Distribute To these Subscribers scrolling list of the Distribute Profile dialog.
This selection distributes the profile only to the subscribers of the profile manager. It does not distribute to lower-level subscribers. If a profile manager with subscribers resides at the next lower level, you may need to perform the distribution process from profile managers at more than one level to reach all the profile endpoints.

- All levels of subscribers
  Distributes the profile to all subscribers in the hierarchy. An example follows to illustrate the difference between distributing to the two levels of subscribers. You have a profile hierarchy in which a dataless profile manager is subscribed to a profile manager, and the dataless profile manager has an endpoint subscribed. If you distribute to the next level of subscribers, the profile is distributed only to the dataless profile manager. If you distribute to all levels of subscribers, the profile is distributed to the dataless profile manager, and to the endpoint.

Select this option if you want to distribute a profile in which your endpoint is the only subscriber.

- Select the following Distribution Will option: Make subscribers profile an EXACT COPY of this profile
  Overwrites the subscribers profile with an exact copy of the profile being distributed.
  Do not distribute a profile to a Tivoli Enterprise™ endpoint with the preserve modifications option set. You must always use the make exact copy option.
- Select subscribers to receive the profile by choosing them from the Don’t Distribute To These Subscribers scrolling list and moving them to the Distribute To These Subscribers scrolling list.

**Note:** Make sure that each of the subscribers in the **Distribute To These Subscribers** scrolling list is either a profile manager or a supported TMA endpoint. Tivoli Monitoring does not support other types of endpoints.

4. Click **Distribute & Close** to distribute the profile immediately and close the Distribute Profile dialog, saving the settings you have made, and return to the Tivoli Monitoring Profile dialog.

You can also click **Distribute** to distribute the profile immediately, saving the settings you have made, but leaving the Distribute Profile dialog open.

Alternatively, you can click **Schedule** to schedule the distribution of the profile with the Tivoli Scheduler. For details about using the Tivoli Scheduler, refer to the *Tivoli Management Framework User’s Guide*.

---

### Profile distribution using MDist 2

Tivoli Monitoring uses Multiplexed Distribution (MDist 2) to perform asynchronous profile data transfers through a hierarchy of repeaters. MDist 2 returns a sequence of responses containing the distribution status from each endpoint to the application initiating the distribution. These responses are sent back to Tivoli Monitoring in a log file (see “Distribution Logs” in the *IBM Tivoli Monitoring: Problem Determination Guide* for details).

Tivoli Monitoring makes full use of these MDist 2 functions:

- Asynchronous delivery
- Assured delivery
- Check-point and restart
• Data depoting

Asynchronous delivery
MDist 2 uses an asynchronous interface to applications, which means that when Tivoli Monitoring submits a distribution request, it immediately gets back a distribution identifier and confirmation that the distribution is in progress. Only when the distribution is complete does MDist 2 send the final distribution status for each endpoint through the callback function. This means that Tivoli Monitoring does not need to be kept waiting until the distribution is complete for every endpoint.

Assured delivery
The distribution of Tivoli Monitoring profiles is assured even when there are network interruptions, power-offs on the machines, or disconnected endpoints. MDist 2 is able to continue the distribution from where it reached before the network interruption. It also retries to establish connections to endpoints until it is successful or the distribution time expires.

Check-point and restart
A data stream that has been interrupted can be resumed from the last successful check-point. This means that it is not necessary to resend all the Tivoli Monitoring Profile data when the distribution is resumed, but only the data that had not yet been sent when the interruption occurred.

Data depoting
MDist 2 allows the storage of distribution segments at a depot close to the endpoint, and for the distribution to be submitted to the endpoints from that depot. The data transferred to the endpoints is thus taken from the depot, and not from the source host. This considerably reduces network traffic and speeds up the distribution.

Rerunning the failed distributions
When a distribution fails, Tivoli Monitoring creates a profile manager containing the endpoint subscribers that failed. To see the profile managers, go to the Region dialog and select View → Refresh.

If the distribution failed because of error Bad_Interpreter, then the profile manager name is derived as follows:

OriginalProfileName_Push_Failed_Bad_Interpreter

where:

OriginalProfileName
This is the name of the profile that you were distributing when the error occurred.

Error message AMW089E might be displayed at this point, indicating that the resource model type is not compatible with the endpoint operating system. For example, you might have distributed a Windows resource model to a UNIX endpoint, or vice versa.

If the distribution fails because of any other error, then the profile manager name is derived as follows:

OriginalProfileName_Distribution_Failed

Provided that the profile manager that you used for the original distribution was created without checking the “dataless endpoint mode” option, you can use these
profile managers to redistribute the profile to the failed endpoints when you have fixed the problem that caused the original failure. To do this, just subscribe the profile managers that contain the failed endpoints to the profile manager that contained the original profile. You can then distribute the original profile to the failed endpoints by selecting these profile managers as the target for the distribution. The profile managers can also be edited to delete an endpoint from a group of failed endpoints before retrying the distribution.

If the redistribution is successful, the profile manager relating to the error condition is deleted. If the redistribution is wholly or partially unsuccessful, the profile manager is retained, containing only the subscribers that were unsuccessful in the redistribution. You should continue the process of correcting the errors and retrying the distribution until all errors have been fixed.

If any of these problems persist and cannot be resolved, call IBM Software Support.

### Deleting profile copies during an unsubscribe

This section explains how you can enable the profile manager resources to remove profile copies at target subscribers. As a general rule, the unsubscribe operation is performed through the Tivoli Desktop or through the Tivoli Management Framework `wunsub` command.

Before any unsubscribe operation, if you want to allow the removal of Tivoli Monitoring profile copies at the target subscribers (“Delete all profile copies” option on the Tivoli Desktop, or `wunsub` without the option `-I`), you must enable the profile manager resource to do this action. You do so by implementing one of the procedures described in this section.

The procedures use the profile manager validation policy method `pm_val_remove_subscribers` to initiate an MDist 2 delete distribution that removes local copies of profiles at the target subscribers (endpoints, application proxies, or profile managers).

The procedures are different depending on whether you have (or not) a policy object for profile manager already enabled in your policy region other than the policy object provided by the Tivoli default installation.

The results of the unsubscribe operation are written to the files (one for each profile):

```
$DBDIR/AMW/logs/msg_profile#region_remove.log
```

**Note:** The unsubscribe operation does not prevent the endpoint engine from running. All profiles not affected by the unsubscribe remain active on the endpoint.

### Modifying the profile manager policy when no custom policy object is in use

You can implement the procedure using the Tivoli Desktop or using the command line interface.

**Using the Tivoli desktop**

Do the following:

![Managed Resource Policies dialog](image)

2. Select **Profile Manager** from the Managed Resources list.
3. Select **ITMUnsubscribe** from the Validation Policy drop-down list. Make sure that the Validation Enabled check box is selected.
4. Click **Set & Close**.

Repeat steps 1 through 4 for each policy region.

To disable, follow the previous steps, but select None or another validation policy from the Validation Policy drop-down list.

Note that the product uninstall removes the ITMUnsubscribe validation policy and sets it to None on all policy regions where it was enabled.

**Using the CLI**

For each policy region run the command:

```
$ wsetpr -v ITMUnsubscribe -e ProfileManager policy_region_name
```

To disable, for each policy region run the command:

```
$ wsetpr -v None -e ProfileManager policy_region_name
```

Note that the product uninstall removes the ITMUnsubscribe validation policy and sets it to None on all policy regions where it was enabled.

**Modifying the profile manager policy when a custom policy object is in use**

If you already have a policy object, you can modify it according to the following steps:

1. Redirect to a file the policy method that you are currently using:
wgetpolm -v ProfileManager your_policy_object pm_val_remove_subscribers > temp_policy_script_file

2. Append to the temporary script the contents of file 
   itm_remove_subscribers.validation which is installed on directory
   $BINDIR/TME/Tmw2k on the managed node.

3. Include the modified script in your policy object:
   wgetpolm -v ProfileManager your_policy_object pm_val_remove_subscribers < temp_policy_script_file

4. Make sure that your policy is enabled on the resource for the policy regions
   in your Tivoli environment (see the wsetpr command in the Tivoli Management

Managing profiles and resource models at endpoints

Tivoli Monitoring provides you with a set of commands to manage profiles and
resource models after they have been distributed to endpoints:

```
wdmcmd    Stops or starts Tivoli Monitoring on one or more endpoints from a
gateway or server.

wdmcmdddistrib
   Using the MDist 2 service, it stops, starts, or upgrades Tivoli
   Monitoring on one or more endpoints from a gateway or server.

wdmcCollect
   Starts or stops the collection of data from one or more endpoints
   and loads the data into a relational database management system
   used by Tivoli Monitoring.

wdmdistrib
   Distributes a profile to one or more subscribers.

wdmeng    Stops or starts profiles or resource models at endpoints; also
          deletes profiles at endpoints.

wdmlseng  Returns a list and the status of all resource models that have
          been distributed on a specified endpoint.

wdmtrceng Sets the trace parameters of the Tivoli Monitoring engine at the
          endpoint.
```

Details of all Tivoli Monitoring commands can be found in Chapter 8, "Commands," on page 109. Details of Tivoli Management Framework commands can be found in the Tivoli Management Framework: Reference Manual.

Managing Tivoli Monitoring at gateways

Tivoli Monitoring provides you with a command to manage the product on
gateways:

```
wdmmm    Stops or starts selected Tivoli Monitoring processes on one or all
gateways.
```

Commands to manage the Tivoli Business Systems Manager Adapter component
on gateways are described in “Managing the Tivoli Business Systems Adapter from
the CLI” on page 93.

Using Tivoli Monitoring tasks

Tivoli Monitoring provides the following set of built-in tasks that you can use to
perform specific product functions:
DMEndpointRemoveFiles
Use this task to remove the engine code from an endpoint (without affecting any definitions in the managed node).

DMEndpointUninstall
Use this task to uninstall an endpoint (see "Uninstalling the endpoint components" on page 34).

DMRebootUninstall
Use this task to prevent the automatic reboot of a specific endpoint from the parent gateway, or of all endpoints associated to a gateway. The task must be run on the corresponding managed node. After you run this task, the engine is not restarted on the endpoint at reboot time until you distribute the profile again.

DMLinkJRE
Use this task on UNIX or Linux endpoints, which already have the appropriate version of Java Runtime Environment (JRE) installed, to link Tivoli Monitoring to the existing JRE. For details see "Installing Java Runtime Environment" on page 31.

DMRemoveLinkJRE
Use this task on UNIX or Linux endpoints to remove the link to an existing JRE.

DMCollectEpLog
Use this task to collect troubleshooting data on an endpoint. For details, see “Serviceability Tasks” in the IBM Tivoli Monitoring: Problem Determination Guide.

DMCollectMnLog
Use this task to collect troubleshooting data on a managed node. For details, see “Serviceability Tasks” in the IBM Tivoli Monitoring: Problem Determination Guide.

DMCollectEpEnv
Use this task to collect information about the environment of an endpoint. For details, see “Serviceability Tasks” in the IBM Tivoli Monitoring: Problem Determination Guide.

DMRemoveQuadcapDB
Use this task to remove from UNIX endpoints the Quadcap Embeddable Database (which was used by previous releases of Tivoli Monitoring). On UNIX endpoints the tool currently used is Cloudscape™, and the existing Quadcap data is automatically migrated to Cloudscape. Use the DMRemoveQuadcapDB task if you do not use the Quadcap data or if you have been experiencing problems with the Quadcap database becoming too large on your endpoints. Also, note that the migration procedure does not delete the existing Quadcap database, therefore you may want to use this task to do so after the data has been migrated.

DMCleanDB
Use this task to remove the Data Warehouse database files from an endpoint (Windows or UNIX). The endpoint engine must not be running when the task is executed. The database is created again when the endpoint engine is restarted. The files are located in the following directories:

- on Windows: %LCF_DATDIR%\LCFNEW\Tmw2k\db
- on UNIX: $LCF_DATDIR/LCFNEW/OS_type/ITMLogger/dblogger_cloud
DMPurge_ITM_RIM_DB
Use this task to remove data from the Tivoli Monitoring RIM database. The task runs according to the value specified for the datacollector.db_purge_interval key using the wdmconfig command. For details, see this command on page 121.

DMSetCloudscapeNetworkServerProperties
Use this task on UNIX endpoints to enable the Cloudscape Network Server functionality, or to set configuration parameters for Cloudscape Network Server. The task creates the file itm.properties in the directory $DATDIR/LCFNEW/Tmw2k/Unix/data. Note that the configuration changes have effect only after the engine is restarted. The configuration parameters are:

- The enabling/disabling of the Cloudscape Network Server functionality (boolean value): the default value is that Cloudscape Network Server is enabled (“true”).
- The port number used by Cloudscape Network Server: the value must be in the range 1 to 65536, the default is 1527.
- The password for the read-only user named “guest”: the default password is “guest”.
- The enabling/disabling of connections tracing (boolean value): the default is that tracing is not enabled (“false”). When enabled, the trace is created in the directory $DBDIR/AMW/logs.

DMGetCloudscapeNetworkServerProperties
Use this task on UNIX endpoints to view the configuration parameters for Cloudscape Network Server. If no settings can be viewed, it means that Cloudscape Network Server is not enabled.

DMCompressDB
Use this task to compress the Cloudscape database on a UNIX endpoint. The task is executed only if the engine is not running. The results are stored in the file dmtaskout.log in the directory:
$LCF_DATDIR/LCFNEW/Tmw2k/Unix/data/ITMLogger/dblogger_cloud

You should set the task timeout to a value greater than the default value and in relation with the size of the database to be compressed. Before setting the task timeout value in the Execute Task dialog, you should check the timeout value at the gateway (the default is 300 seconds). If the timeout at the gateway is smaller than the task timeout, then the gateway timeout supersedes the task timeout. To change the timeout at the gateway, you can use the wgateway command, using the set_session_timeout option. Refer to the Tivoli Management Framework: Reference Manual for details on this command.

DMCreateRuleAndLoadBaroc
Use this task to import the Tivoli Monitoring BAROC files into the rule base used by the Tivoli Enterprise Console server. The task implements all steps of the procedure described on page 79 and requires the super or senior authorization roles to be executed. You should specify a timeout higher than the default value: a timeout of 600 seconds is recommended. The task stops and restarts the Tivoli Enterprise Console server if the corresponding option is selected.

To run a Tivoli Monitoring task, you can use the Tivoli Desktop or the command line interface. Either way, you need to have one of the following authorization roles: admin, senior, super, or itm_tasks.
Running a task from the Tivoli desktop

To run a Tivoli Monitoring task using the Tivoli Desktop, do the following:

1. From the Tivoli desktop select **TMR Connections → Top Level Policy Regions**.
   The Top Level Policy Regions dialog opens.

2. Double-click the **TivoliDefaultMw2kRegion** icon.
   The Policy Region dialog for the default policy region opens.

3. Double-click the **IBM Tivoli Monitoring Tasks** icon.
   The Task Library dialog opens showing all tasks available for Tivoli Monitoring.

4. Double-click the icon for the task that you want to run and define the parameters for the chosen task in the Execute Task dialog. Click **Execute**.

   **Note:** You can define **DMTaskPolicy** as the default policy to be used with the task library on any policy region: using this policy enables the appropriate list of targets to be displayed in the **Available Task Endpoints** list on the Execute Task dialog. The Available Task Endpoints list will display only those targets on which the task can successfully run.

Running a task from the CLI

You can use the Tivoli Management Framework **wruntask** command to run a Tivoli Monitoring task (refer to *Tivoli Management Framework: Reference Manual* for the complete description of this command).

To run a task from the command line use the following syntax (use quotation marks around the name of a variable that contains spaces):

```
wruntask -t "task_name" -l "IBM Tivoli Monitoring Tasks" -h node_name -a "additional_parameters"
```

Where:

- `-t "task_name"` specifies the name of the task to be run.
–l “IBM Tivoli Monitoring Tasks”
Indicates the name of the task library that includes the task to be run.

–h node_name
Specifies the name of the node (managed node or endpoint) on which to run the task.

–a “additional_parameters”
Specifies the set of parameters to be passed to the task for its execution. If the parameter includes an option flag followed by its own parameter, enclose both in quotation marks (for example, –a “–o option”). The –a option must be repeated for each option specified. The following Tivoli Monitoring Tasks requires specific parameters:

- **DMEndpointUninstall** parameters:
  - the name of the endpoint or –g gateway_name
  - the possibility to disable the view of the output produced by the execution of the nested task (DMEndpointUninstall invokes DMRemoveFiles). The view of the output is enabled by default.
  - the timeout value in seconds for DMRemoveFiles (the default is 300 seconds)

- **DMRebootUninstall** parameter: the name of the endpoint or –g gateway_name

- **DMLinkJre** parameter: the fully qualified path where JRE is installed (not including the /bin directory)

- **DMCollectEpLog** and **DMCollectMnLog** parameters: the name of the directory and filename to log the information

- **DMSetCloudscapeNetworkServer** parameters: enabling/disabling of the Cloudscape Network Server functionality (boolean value), the port number used by Cloudscape Network Server, the password for the read-only user named “guest”, and the enabling/disabling of connections tracing (boolean value)

- **DMCreateRuleAndLoadBaroc** parameters:
  - the name of the rule_base of the Tivoli Enterprise Console server where the BAROC rules are to be stored
  - whether or not the Tivoli Enterprise Console should be restarted at completion of the task
  - the indication that Tivoli Enterprise Console 3.6.2 is installed (even though this level is no longer supported)

The following are examples about using `wruntask`:

1. to run the **DMEndpointUninstall** task on a managed node to uninstall one of its endpoints, specifying for the nested task (DMRemoveFiles) a 500 seconds timeout and the disabling of the output (–noheader option):
   `wruntask –t DMEndpointUninstall –l IBM Tivoli Monitoring Tasks
   –g @ManagedNode:mn1 –a ep1 –a “–noheader” –a “–t 500”`

2. to run the **DMLinkJre** task on an endpoint:
   `wruntask –t DMLinkJre –l IBM Tivoli Monitoring Tasks
   –g @Endpoint:ep1 –a /opt/IBMJava2-131/jre`

3. to run the **DMCollectEpLog** task on an endpoint:
   `wruntask –t DMCollectEpLog –l IBM Tivoli Monitoring Tasks
   –g @Endpoint:ep1 –a /opt/Tivoli/lcf/dat/1 –a eplog.tar`

4. to run the **DMCollectMnLog** task on a managed node:
wruntask –t DMCollectMnLog –l "IBM Tivoli Monitoring Tasks"
   –h @ManagedNode:mn1 –a /opt/Tivoli/lcf/dat/1 –a mnlog.tar

5. to run the **DMSetCloudscapeNetworkServer** task on an endpoint:
   wruntask –t DMSetCloudscapeNetworkServer –l "IBM Tivoli Monitoring Tasks"
   –h @Endpoint:ep1 –a true –a 1527 –a abc123 –a false

6. to run the **DMCreateRuleAndLoadBaroc** task using a timeout of 600 seconds on a managed node where the Tivoli Enterprise Console server (level 3.6.2) is installed:
   wruntask –t DMCreateRuleAndLoadBaroc –l "IBM Tivoli Monitoring Tasks" –m 600
   –h @ManagedNode:mn1 –a Test_Rule –a "$restartsvr" –a "$362"
Chapter 4. Heartbeat function

Tivoli Monitoring includes a heartbeat function, which monitors the basic signs of life at endpoints attached to the gateway at which it is enabled.

The heartbeat monitor regularly monitors the endpoints, checking if they are running correctly. Events may be sent to the Tivoli Business Systems Manager (provided that the Tivoli Business Systems Manager Adapter component is installed at the gateway), the Tivoli Enterprise Console or the Tivoli Monitoring Notice Group.

Heartbeat statuses

The function can register the following statuses for an endpoint in its cache. They are divided into two groups, depending on whether an information or an Error event is sent to the monitors:

**Statuses for which an information event is sent**
- Alive
- Heartbeat has been stopped

**Statuses for which an error event is sent**
- Resource model in error status
- Tivoli Monitoring engine has been stopped
- Endpoint not available in network

Installing and configuring the heartbeat

The heartbeat function is an integral part of Tivoli Monitoring as installed on a gateway, and needs no specific installation. However, you can configure the recipients of heartbeat information using the `wdmconfig` command (see “wdmconfig” on page 121). To enable the configuration changes applied using `wdmconfig`, you must stop the heartbeat using the `wdmmn` command and restart it using the `wdmheartbeat` command.

The `wdmconfig` command lets you amend various aspects of the configuration of the Tivoli Monitoring components at a gateway. In particular, for the heartbeat, you can change the values of the following configuration keys:

**heartbeat.send_events_to_tbsm**
Set this to `true` if you want to send heartbeat events to the Tivoli Business Systems Manager (the Tivoli Business Systems Manager Adapter must be installed and configured at the gateway); otherwise leave as the default value of `false`.

**heartbeat.send_events_to_tec**
Set this to `true` if you want to send heartbeat events to the hostname of the Tivoli Enterprise Console server; otherwise leave as the default value of `false`.

**heartbeat.tec_server**
If you have set `heartbeat.send_events_to_tec` to `true`, enter here the name of the Tivoli Enterprise Console server (for example: EventServer or EventServer@regionname). The server must be a Tivoli-secure server.
heartbeat.send_events_to_notice
Set this to true if you want to send heartbeat events to the Tivoli Monitoring notice group; otherwise leave as the default value of false.

heartbeat.reboot_engine_if_down
Set this to true if you want to enable the recovery function of the agent where Tivoli Monitoring is running; otherwise leave as the default value of false. When the configuration key is set to true, the recovery function restarts the engine whenever the engine stops running. However, if the engine was stopped using wdmcmd –stop, the recovery function does not restart the engine.

All these configuration keys and their values are contained in file:
$DBDIR/dmml/.config

This file can be read but it must not be modified manually. If you want to modify it, use the wdmconfig command (note that the applied changes might not show).

Controlling the heartbeat

The heartbeat function is controlled by issuing one of the following commands from the Tivoli Console:

wdmheartbeat
Stops or starts the 'heartbeat' monitoring of endpoints, changes its frequency and queries the status of the heartbeat processor (see “wdmheartbeat” on page 148, for more details).

wdmmn
Stops the heartbeat engine on one or all managed nodes/gateways (see “wdmmn” on page 157, for more details).

Figure 4 on page 71 shows the data flow for the heartbeat control.
The heartbeat control commands are routed to the gateway (steps 1 and 2), where they are processed by the heartbeat processor.

**Note:** The heartbeat processor is protected from closures of the oserv such that if the oserv is closed and then re-opened, the heartbeat processor restarts at the same frequency it was using when the oserv was closed.

### Heartbeat data flows

The heartbeat function has these principal activities:

- Endpoint registration
- Heartbeat monitoring
- Viewing the endpoint cache

### Endpoint registration

When Tivoli Monitoring pushes a resource model to an endpoint for the first time, or restarts the Tivoli Monitoring engine at an endpoint, the information in the endpoint cache is updated when the gateway receives a message from the endpoint saying that its engine has been started. Figure 5 on page 72 illustrates the data flow.
Figure 5. Data flow for endpoint registration

The figure shows a profile being distributed (pushed) from the Tivoli Desktop, or a command being issued to start an endpoint engine (steps 1, 2 and 3). The endpoint sends an upcall to the gateway (step 4), which registers the presence of the endpoint in the endpoint cache (step 5).
Heartbeat monitoring

Figure 6 shows the data flow for the heartbeat function at a gateway in a Tivoli management region, with all possible monitors configured to receive the heartbeat events.

The gateway issues periodic heartbeat requests to all attached endpoints (step 1). The data that is returned by the endpoints (step 2) is stored in the endpoint cache (step 3) and sent to whichever of the monitors is configured to receive it (step 4).
**Viewing the endpoint cache**

In addition to the monitors described in the previous section, it is also possible to view the heartbeat information in the endpoint cache, using the `wdmmngcache` command.

![Diagram showing the data flow using the `wdmmngcache` command](image)

**Figure 7. Data flow using the `wdmmngcache` command**

This diagram shows the data flow when a `wdmmngcache` command is issued at the Tivoli Desktop. A request is sent to the gateway (steps 1 and 2), which interrogates the cache (step 3) and sends the information back to the desktop (steps 4 and 5). There is no interaction with the endpoints in this data flow. Data can also be deleted from the cache using this command.

The `wdmmngcache` command has two output formats. If the `-v` option is not used, a line is provided for each endpoint, showing the heartbeat status, as follows:

```
Processing ManagedNode mpulp...
Processing ManagedNode dmw2k2...
```

<table>
<thead>
<tr>
<th>Endpoint</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>mpulp-ep</td>
<td>Alive</td>
</tr>
<tr>
<td>dmw2k2-ep</td>
<td>Alive</td>
</tr>
</tbody>
</table>

If the `-v` option is used, the command provides information about the discovery status of the endpoints, as follows:

```
Processing ManagedNode mcruedele...
Processing ManagedNode dmw2k2...
```

*Warning:* DM_Adv_Edition feature not installed on the Managed Node 'boccaccio'.

---

**IBM Tivoli Monitoring: User’s Guide**

74
<table>
<thead>
<tr>
<th>Endpoint</th>
<th>HB status</th>
<th>TBSM status</th>
</tr>
</thead>
<tbody>
<tr>
<td>dmw2k2-ep</td>
<td>DMEngineOff</td>
<td>Not discovered</td>
</tr>
<tr>
<td>mcrudele-ep</td>
<td>DMEngineOff</td>
<td>Not discovered</td>
</tr>
</tbody>
</table>
Chapter 5. Integration with Tivoli Enterprise Console server

This chapter describes how to integrate Tivoli Monitoring with the Tivoli Enterprise Console server to enable Tivoli Monitoring events to be used by the Tivoli Enterprise Console server.

Overview

Before you can send Tivoli Monitoring events to a Tivoli Enterprise Console server, you must:

- Identify the Tivoli Enterprise Console to Tivoli Monitoring
- Enable monitoring of product events and heartbeat messages on the Tivoli Enterprise Console server

Identifying the Tivoli Enterprise Console server to Tivoli Monitoring

For each profile you create, you must identify the Tivoli Enterprise Console server (or multiple servers) that you want to use to monitor the Tivoli Monitoring events. You can also choose to send unsecure events to the Availability Intermediate Manager (AIM).

To set the identity of the Tivoli Enterprise Console server for all resource models in a profile, perform the following steps:

1. Open the Tivoli Monitoring Profile dialog.
2. Select Edit → Properties.
   The Properties dialog opens and displays the name of the profile for which the server identity is to be set, and the profile manager in which the profile is located. It also displays the option **Send TBSM**.

3. Check the **Send TEC Events** check box to send events to a Tivoli Enterprise Console server (or multiple servers) or to AIM (only for unsecure delivery events).
Note: If you do not select this check box, no events are sent to the Tivoli Enterprise Console server, even if you selected the Send TEC Events check box in the Indications and Actions dialog.

Once this is checked, the Using section of the dialog is enabled. The Using section is disabled if the Send TEC Events check box is not selected. You must select a delivery mode, the mutually exclusive choices are:

- **Event Server(s)**
  
  If you choose this option, all events are sent to the specified Tivoli Enterprise Console server, or multiple servers. Specify the server or select it from the Servers drop-down list. If you specify a server with no region name, the events are sent to the local Tivoli Enterprise Console server. To indicate multiple servers, use a comma as a separator. If the server is an unsecure server, specify the server location followed by the plus ‘+’ sign and the server port.

- **Fail-over**

  If you choose this option, all events are sent to one unsecure Tivoli Enterprise Console server. If you specify multiple servers, the events are sent to the first available server in the list. Servers are specified by the server location followed by the plus ‘+’ sign and the server port.

  In the case of secure servers, it is the responsibility of the Tivoli Enterprise Console gateway to ensure that events are sent to the first available server in the recovery list (the Tivoli Enterprise Console configuration file on the gateway can be modified for this purpose).

Note: When Fail-over is selected as the delivery mode, and the server to which the event has been sent goes down, the clearing event is sent to the first available server in the list. The original event needs to be closed manually once the server is active again.

Note: On OS/400 endpoints, events can only be sent to an unsecure Tivoli Enterprise Console server. Any attempt to send an event to a secure server on OS/400, results in the event not being sent and an error being logged.

4. Click OK to save the properties.

The Tivoli Monitoring Profile dialog is redisplayed.

---

**Enabling monitoring of events and heartbeat messages on the Tivoli Enterprise Console server**

To enable monitoring of Tivoli Monitoring events on a Tivoli Enterprise Console server, you need to import the Tivoli Monitoring BAROC files into the rule base used by the Tivoli Enterprise Console server.

To accomplish this you need to implement one of the following options:

- run a Tivoli-provided script on each Tivoli Enterprise Console server
- or run a Tivoli-provided task on each Tivoli Enterprise Console server
- or execute a procedure

This section describes the three options:

**Running the script**

A script is provided to enable you to update the rule base in a semi-automatic way. The script is named dmae_tec_inst.sh, and is stored in the $BINDIR/TMNT_TEC directory.
The script must be run on the Tivoli Enterprise Console server. It takes as an argument the name of the rule_base of the Tivoli Enterprise Console server where the BAROC rules are to be stored. As an option, you can also specify the following parameters:

- `-restartsvr` to restart the server at the end of the script.
- `-362` to be used when Tivoli Enterprise Console 3.6.2 is installed (even though this level is not supported).

**Running the task**

A task is provided to perform the same actions of the script but, as opposed to the script, it can address multiple Tivoli Enterprise Console servers simultaneously. The task is called **DMCreateRuleAndLoadBaroc**, refer to “Using Tivoli Monitoring tasks” on page 63 for details.

**Following the procedure**

If you do not want to run the script nor the task, you can follow the individual steps that the script and task perform, which are detailed in the procedure set out below. For full information about performing each step of the procedure, refer to the *Tivoli Enterprise Console Rule Builder’s Guide*.

Perform the following steps:

1. Select an existing rule base or create a new rule base to contain the Tivoli Monitoring BAROC files.
2. Import the required BAROC files from the `$BINDIR/TMNT_TEC` directory into the rule base.
   
   It is important that you import the files in the following order:
   
   a. Import the `Tmw2k.baroc` file. For more information about it, see the *Tivoli Enterprise Console Rule Builder’s Guide*.
   
   b. Import the BAROC files for all the resource models whose events you want to send to Tivoli Enterprise Console server.
   
   c. Import the `hb_events.baroc` file to enable heartbeat messages.
3. Import the required heartbeat rules file from the `$BINDIR/TMNT_TEC` directory into the rule base to enable the support of heartbeat messages. The file is called `hb_events.rls`. For more information on the rules contained in this file, see “Understanding the Tivoli Enterprise Console rules” on page 82.
4. Import the required clearing event rules file from the `$BINDIR/TMNT_TEC` directory into the rule base to enable clearing events to close the error events to which they relate. The file is called `dmae_events.rls`. For more information on the rules contained in this file, see “Understanding the Tivoli Enterprise Console rules” on page 82.
5. Compile and load the rule base.
6. Stop and restart the Tivoli Enterprise Console server.

The Tivoli Enterprise Console server is now ready to receive Tivoli Monitoring events from the monitoring sources whose corresponding BAROC files you have imported into the active rule base.

To see the events sent by Tivoli Monitoring, from the Tivoli Enterprise Console main dialog, click the All icon. See the *Tivoli Enterprise Console: User’s Guide* for more details.
Note: The BAROC available with this version of Tivoli Monitoring can also be used with Tivoli Distributed Monitoring (Advanced Edition) 4.1 or with Tivoli Distributed Monitoring for Windows 3.7 Patch 3.

Enable monitoring of the events on the Availability Intermediate Manager

To enable monitoring of Tivoli Monitoring events on Availability Intermediate Manager (AIM), follow the same procedure described for Tivoli Enterprise Console server.

Only unsecure events can be sent to AIM.

Slots of the Tivoli Enterprise Console classes

This section describes the slots (fields) related to the TMW_Event and HeartBeat_Event classes.

Slots in the TMW_Event class

The following table lists all the slots pertaining to the TMW_Event class. All the event classes of the Tivoli Monitoring resource models inherit from the TMW_Event class.

Table 9. Slots of the TMW_Event class

<table>
<thead>
<tr>
<th>Slot</th>
<th>Slot description</th>
<th>Slot value</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
<td>Event adapter known to the event server</td>
<td>TMNT</td>
</tr>
<tr>
<td>sub_source</td>
<td>A more detailed description of the source</td>
<td>N/A</td>
</tr>
<tr>
<td>origin</td>
<td>Protocol address or host name of the source system</td>
<td>endpoint_IP_address</td>
</tr>
<tr>
<td>sub_origin</td>
<td>A more detailed description of the origin</td>
<td>N/A</td>
</tr>
<tr>
<td>hostname</td>
<td>The name of the endpoint where the event occurred</td>
<td>endpoint_label</td>
</tr>
<tr>
<td>fqhostname</td>
<td>The fully qualified name of the host where the event occurred</td>
<td>fq_hostname</td>
</tr>
<tr>
<td>adapter_host</td>
<td>The name of the system where the event adapter that reported the event is located</td>
<td>endpoint_label</td>
</tr>
<tr>
<td>msg</td>
<td>A descriptive message of the event</td>
<td>message</td>
</tr>
<tr>
<td>msg_catalog</td>
<td>The name of the message catalog to be used when displaying internationalized messages</td>
<td>none</td>
</tr>
<tr>
<td>msg_index</td>
<td>The offset to the event message catalog</td>
<td>0</td>
</tr>
</tbody>
</table>
Table 9. Slots of the TMW_Event class (continued)

<table>
<thead>
<tr>
<th>Slot</th>
<th>Slot description</th>
<th>Slot value</th>
</tr>
</thead>
<tbody>
<tr>
<td>repeat_count</td>
<td>Repeat counter</td>
<td>0</td>
</tr>
<tr>
<td>severity</td>
<td>Event severity</td>
<td>event_severity</td>
</tr>
<tr>
<td>short_msg</td>
<td>A shorter description of the message (max 20 characters)</td>
<td>N/A</td>
</tr>
<tr>
<td>modelname</td>
<td>The name of the resource model</td>
<td>resource_model_name</td>
</tr>
<tr>
<td>modelinstance</td>
<td>The resource model instance</td>
<td>N/A</td>
</tr>
<tr>
<td>profilename</td>
<td>The name of the profile</td>
<td>profile_name</td>
</tr>
<tr>
<td>eventid</td>
<td>A unique event ID (see the specific information below)</td>
<td>event_id</td>
</tr>
<tr>
<td>event_key</td>
<td>A list of key information (see the specific information below)</td>
<td>key_list</td>
</tr>
</tbody>
</table>

In particular, the slots eventid and event_key have the following meaning:

- **eventid**: the value of this slot identifies events in relation to the problem they describe. Events associated to the same problem have the same eventid value. Also, clearing events have the same eventid value of the event they clear.

To clarify this, consider the following example. Let’s assume that for event E the aggregation rule is 3 consecutive occurrences of the indication (I) with a maximum of 1 hole (H) between two indications. The following tables show the life cycle of the eventid value (between brackets) with and without a clearing event.

When clearing events are not used (Table 10), the events are generated after every 3 consecutive occurrences with a maximum of 1 hole in between. The eventid value changes when the event becomes associated to a different problem instance (at T13 in the table).

Table 10. Lifecycle of eventid property (no clearing event)

<table>
<thead>
<tr>
<th>Cycles</th>
<th>T0</th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
<th>T6</th>
<th>T7</th>
<th>T8</th>
<th>T9</th>
<th>T10</th>
<th>T11</th>
<th>T12</th>
<th>T13</th>
<th>T14</th>
<th>T15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indications</td>
<td>I</td>
<td>I</td>
<td>H</td>
<td>I</td>
<td>H</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>H</td>
<td>H</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>H</td>
<td></td>
</tr>
<tr>
<td>Events</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>E(id1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>E(id2)</td>
</tr>
</tbody>
</table>

When clearing events are used (Table 11), the events are generated only once after 3 consecutive occurrences with a maximum of 1 hole in between. The clearing event is sent (at T10 in the table) when the number of holes exceeds the maximum number allowed by the aggregation rule. The eventid value changes when the event becomes associated to a different problem instance (at T13 in the table).

Table 11. Lifecycle of eventid property (with clearing event)

<table>
<thead>
<tr>
<th>Cycles</th>
<th>T0</th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
<th>T6</th>
<th>T7</th>
<th>T8</th>
<th>T9</th>
<th>T10</th>
<th>T11</th>
<th>T12</th>
<th>T13</th>
<th>T14</th>
<th>T15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indications</td>
<td>I</td>
<td>I</td>
<td>H</td>
<td>I</td>
<td>H</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>H</td>
<td>H</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>H</td>
<td></td>
</tr>
<tr>
<td>Events</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>E(id1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C(id1)</td>
</tr>
</tbody>
</table>
• **event_key**: the value of this slot identifies the set of name=value pairs of key properties. Usually, the key properties of an indication define the resource instance to which the event is related. The character ";" is used as a separator between each name=value pair (for example, name1=value1;name2=value2). As opposed to eventid, the event_key can have the same value for different events.

### Slots in the HeartBeat_Event class

The following table lists all the slots pertaining to the HeartBeat_Event class. All the Tivoli Enterprise Console events that are sent by the heartbeat inherit from the HeartBeat_Event class and are included in `$BINDIR/TMNT_TEC/hb_events.baroc`.

**Table 12. Slots of the HeartBeat_Event class**

<table>
<thead>
<tr>
<th>Slot</th>
<th>Slot description</th>
<th>Slot value</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
<td>Event adapter known to the event server</td>
<td>TMNT</td>
</tr>
<tr>
<td>sub_source</td>
<td>A more detailed description of the source</td>
<td>N/A</td>
</tr>
<tr>
<td>origin</td>
<td>Protocol address or hostname of the source system</td>
<td>Endpoint <em>IP</em> address</td>
</tr>
<tr>
<td>sub_origin</td>
<td>A more detailed description of the origin</td>
<td>N/A</td>
</tr>
<tr>
<td>hostname</td>
<td>The name of the endpoint where the event occurred</td>
<td>endpoint_label</td>
</tr>
<tr>
<td>fqhostname</td>
<td>The fully qualified name of the host where the event occurred</td>
<td>fq_hostname</td>
</tr>
<tr>
<td>adapter_host</td>
<td>The name of the system where the event adapter that reported the event is located</td>
<td>managednode_label</td>
</tr>
<tr>
<td>severity</td>
<td>Event severity</td>
<td>event_severity</td>
</tr>
<tr>
<td>msg</td>
<td>A descriptive message of the event</td>
<td>message</td>
</tr>
<tr>
<td>msg_catalog</td>
<td>The name of the message catalog to be used when displaying internationalized messages</td>
<td>none</td>
</tr>
<tr>
<td>msg_index</td>
<td>The offset to the event message catalog</td>
<td>0</td>
</tr>
<tr>
<td>repeat_count</td>
<td>Repeat counter</td>
<td>0</td>
</tr>
</tbody>
</table>

### Understanding the Tivoli Enterprise Console rules

The directory `$BINDIR/TMNT_TEC` contains a number of files that include Tivoli Enterprise Console rules. This section lists all rules, the file which contains them, a description of the rule, the slots related to the rule, and the actions.

- **Rule: all_events_clearing**
  - File: `hb_events.rls`
Description: this rule is related to the heartbeat function. When the heartbeat status changes and the Tivoli Enterprise Console receives one of the events listed below, the rule closes the old status event associated to the same endpoint:
- ‘Heartbeat_Off’
- ‘Heartbeat_EndpointUnreachable’
- ‘Heartbeat_EndpointMigrated’
- ‘Heartbeat_DMAgentDown,
- ‘Heartbeat_DMAgentAlive’
- ‘Heartbeat_ResourceModelsInError’

For example, if Tivoli Enterprise Console received the event ‘Heartbeat_DMAgentDown’ related to host “myhost” from IP address “146.84.112.33”, and then receives event ‘Heartbeat_DMAgentAlive’ associated to the same endpoint, at that point the rule closes the ‘Heartbeat_DMAgentDown’ event.

Note: The Heartbeat_EndpointMigrated event can only be sent if the heartbeat.send_endpoint_migrated_event key of wdmconfig is set to true. The default key value is false.

- Used Slots:
  - origin
  - hostname
- Actions:
  - it changes to Closed the event status of the old event with the same slot values
  - it changes to ‘DM(AE) Heartbeat Clearing Event Rule’ the event administrator of the old event with the same slot values

• Rule: tmw_clearing
  – File: dmae_events.rls
  – Description: this rule closes the TMW_Event with the same event id, hostname, and origin slot values. In other words, it closes the TMW_Event for which the clearing event was generated.
  – Used Slots:
    - origin
    - hostname
    - eventid
  – Actions: if the related TMW_Event is found,
    - it changes to Closed the event status of the old event with the same slot values
    - it changes to ‘DM(AE) Clearing Event Rule’ the event administrator of the old event with the same slot values
    - it drops the current clearing event

• Rule: drop_tmw_clearing
  – File: dmae_events.rls
  – Description: this rule drops the clearing event even if the related TMW_Event was already closed or dropped.
  – Used Slots:
    - origin
- hostname
- eventid
- Actions: if the related TMW_Event is found,
  - it drops the current clearing event

**Rule: transform_heartbeat**
- File: hb_ext.rls
- Description: this rule creates a ‘TEC_IS_UNREACHABLE’ event every time the Tivoli Enterprise Control server receives a ‘Heartbeat_EndpointUnreachable’ event. This enables the integration between Tivoli Monitoring and NetView®.
- Used Slots:
  - origin
  - hostname
  - eventid

**Note:** To enable the forwarding of 'Heartbeat_EndpointUnreachable' to NetView, you must import the rules in the following order:
1. import hb_events.rls
2. import hb_ext.rls
3. import netview.rls
Chapter 6. Integration with Tivoli Business Systems Manager

This chapter describes how to integrate Tivoli Monitoring with Tivoli Business Systems Manager, so that Tivoli Monitoring events can be sent to and analyzed by Tivoli Business Systems Manager. The following topics are covered:

- Installing the Tivoli Business Systems Manager Adapter
- Configuring the Tivoli Business Systems Manager Adapter
- Using the Tivoli Business Systems Manager Adapter
- Viewing Tivoli Business Systems Manager data
- Enabling and disabling Tivoli Business Systems Manager integration
- Managing the Tivoli Business Systems Adapter from the CLI

Overview

Tivoli Business Systems Manager is an application that enables you to perform distributed management, OS/390® management, or both. Even when a business system spans multiple platforms, Tivoli Business Systems Manager enables you to graphically monitor and control interconnected business components and operating system resources. A business component and its resources are referred to as a Line of Business (LOB). Using the LOB concept Tivoli Business Systems Manager helps you plan, define, and control your business system. Tivoli Business Systems Manager, together with other Tivoli management components, helps you manage the dependencies between business components and their underlying infrastructure.

The management facilities in Tivoli Business Systems Manager include the management of events generated by Tivoli Monitoring, so when you are using Tivoli Business Systems Manager, you can direct some or all of the events generated by Tivoli Monitoring to the Tivoli Business Systems Manager interface, letting you manage them in the same way as other aspects of your business systems.

Tivoli Monitoring events are sent to Tivoli Business Systems Manager by the Tivoli Business Systems Manager adapter, a Tivoli Monitoring component that should be installed on all gateways in a region.

The adapter feeds Tivoli Business Systems Manager with information about resources managed by Tivoli Monitoring by means of a bulk/delta discovery. For discovered systems it can then supply heartbeat information and Tivoli Monitoring events.

Installing the Tivoli Business Systems Manager Adapter

The Tivoli Business Systems Manager adapter is installed as a product in the same way as Tivoli Monitoring is installed; see “Installing/upgrading the product” on page 24.

Note: The Tivoli Business Systems Manager Adapter cannot be installed on HP-UX gateways.
During the installation, you must define the installation path of Java Runtime Environment to the Tivoli Business Systems Manager Adapter. You do so by following these steps on the Install Options dialog:

1. Enter the following details:
   
   **JRE path for ...**
   
   For each platform type on which you want to install the adapter, enter the complete path of the root installation directory of Java Runtime Environment (not including the directory /bin).

2. Press **Set** to save the paths.

If not all of your platforms of the same type have the same root directory for Java Runtime Environment the following is proposed:

1. Enter the most common path for the platform in question.
2. For each gateway that does not use the supplied path, use the **wdmconfig** command to amend the **tbsma.jre_root** variable to the correct path.

If you press **Close** on the Install Options dialog, the installation path of the JRE is not defined. In that case use the **wdmconfig** command to set the **tbsma.jre_root** variable to the correct path.

---

**Configuring the Tivoli Business Systems Manager Adapter**

The Tivoli Business Systems Manager adapter is configured by using the command **wdmconfig**. You can address this command to one, a list, or all gateways in the region, and it allows you to supply any number of key/value pairs as parameters. The key/value pairs that you need to configure are as follows:

**adapter.trace.enable**

Set this to **true** if you want to write to the file identified in **trace.filename**, all trace messages regarding the operations of the adapter. The default is **false**.

**adapter.trace.level**

Set this to **low**, **medium** or **high**, according to the level of details you require if you have enabled adapter trace messages. The default is **low**.
adapter.working.dir
Working directory that is used by the adapter. The default, which you are recommended to use, is the Tivoli Monitoring middle layer directory ($DBDIR/dmml).

adapter.xml.validation
Set this to true if you want to enable the xml validation. The default is false.

tbsma.jre_root
This parameter is set during the installation of the Tivoli Business Systems Manager Adapter (see “Installing the Tivoli Business Systems Manager Adapter” on page 85), and you do not normally need to change it manually. However, if, for example, you want to install the adapter on a group of gateways using one instance of the install action or command, you need to change this parameter on any gateways in the group that have JRE installed at a location different to that supplied on the Install Options dialog.

Set this parameter to the complete path of the root directory of Java Runtime Environment, 1.3.0 (excluding the directory /bin).

Note: In a Windows NT workstation, if the install target path contains a directory with spaces in the name, the directory name must be specified between single quotes, as in this example:
D:\'Program Files'\jre

tbsma.listener_timeout
This parameter is set to define the maximum timeout to create a connection with the Tivoli Business Systems Manager CommonListener. The timeout is expressed in minutes: the minimum value is 1 minute, the default is 30 minutes.

tbsma.trace_level
Specifies the level of the Tivoli Business Manager engine traces (tmnt_tbsm_eng and tmnt_tbsm_wrapper processes). The minimum value is 0, the maximum is 2, the default is 1.

tbsma.trace_size
Specifies the size in bytes of the Tivoli Business Manager engine traces (tmnt_tbsm_eng and tmnt_tbsm_wrapper processes). The default is 500000.

trace.filename
Filename to which the trace messages from the adapter are written. The default filename is DM.trc.

transport.local.ip.address
Specifies the IP address of the node running the Tivoli Business Systems Manager Adapter. The default value is the IP address of the node where the configuration file has been created.

transport.request.address
Specifies the address for requests. The default address is local_IP_address.DM.QM+DM.Q (for example, 146.84.112.165.DM.QM+DM.Q)

transport.response.address
Specifies the address for responses. The default address is local_IP_address.DM.QM+DM.Q (for example, 146.84.112.165.DM.QM+DM.Q)
transport.request.port
Specifies the port number used to receive requests. The default value is 6969.

transport.response.port
Specifies the port number user to send responses. The default value is 6969.

transport.mqe.fileregistry
Specifies the file registry for the message queue. The default is com.ibm.mqe.registry.MQeFileSession.

transport.mqe.maxchannels
Specifies the maximum number of channels for the message queue. The default is 1.

transport.mqe.local.queue.store
Specifies how the local message queue stores incoming data. The value can be either of the following (the default is file):

- file Indicates that the data is written to disk until received by the CommonListener
- memory Indicates that the data is kept in memory
- reduced Indicates that some data is kept in memory but, beyond a certain amount, the data is written to disk

transport.mqe.remote.queue.store
Specifies how the remote message queue stores incoming data. The value can be either of the following (the default is file):

- file Indicates that the data is written to disk until received by the CommonListener
- memory Indicates that the data is kept in memory
- reduced Indicates that some data is kept in memory but, beyond a certain amount, the data is written to disk

transport.mqe.usefiller
Set this to true if the gateway on which the adapter is installed is running Windows NT, 4.0, Service Pack 5; otherwise leave as the default value of false.

transport.server.ip.address
IP address or hostname of the CommonListener component of Tivoli Business Systems Manager, which listens out for messages from the systems being managed. For example:
transport.server.ip.address=193.202.74.21

transport.server.mqe.address
Address of the CommonListener component. The default address is ServerQM+ServerQ.

transport.server.mqe.port
Port number of the CommonListener component. For example:
transport.server.mqe.port=8082
**transport.trace.enable**

Set this to `true` if you want to write to the file identified in `trace.filename`, all messages regarding the transport of adapter-acquired data to the CommonListener. The default is `false`.

**transport.trace.level**

Set this to `low`, `medium` or `high`, according to the level of details you require if you have enabled transport trace messages. The default is `low`.

All these key/value pairs and their values are contained in file:

$DBDIR/dmml/.config

This file can be read but it must not be modified manually. If you want to modify it, use the `wdmconfig` command (note that the applied changes might not show).

For full details of the command syntax see [“wdmconfig” on page 121](#).

---

**Using the Tivoli Business Systems Manager Adapter**

Once installed and configured, the adapter provides its services in a fully automatic way, not needing to be independently started or stopped.

It carries out the following activities:

**Bulk discovery**

The `wdmdiscovery` command can be issued from the Tivoli desktop to carry out a bulk discovery. The adapter sends details of all systems it has in its cache to the Tivoli Business Systems Manager’s CommonListener. For full details of the command syntax see [“wdmdiscovery” on page 127](#).

[Figure 8 on page 90](#) shows the data flows for bulk discovery.

**Delta discovery**

The same `wdmdiscovery` command can be issued from the Tivoli desktop to carry out a delta discovery. The adapter sends details of all changes since the last discovery, to the Tivoli Business Systems Manager’s CommonListener. For full details of the command syntax see [“wdmdiscovery” on page 127](#).

[Figure 8 on page 90](#) shows the data flows for delta discovery.
The figure shows the data flow for bulk and delta discovery. The `wdmdiscovery` command is issued at the Tivoli Desktop (steps 1 and 2). At the target gateway, the Tivoli Business Systems Manager Adapter obtains the appropriate discovery information (bulk or delta) from the endpoint cache (step 3) and sends it to the Tivoli Business Systems Manager’s CommonListener (step 4).

**Note:** The Adapter expects that the CommonListener responds to a discovery command within a configurable time-out (the default is 300 seconds). If the systems on which these components are running have misaligned time settings, the time-out value may seem to have been exceeded, with the discovery being rejected. The solution is either to align the systems’ time settings or to increase the time-out value. Full details are given in “Problem Determination” in the *IBM Tivoli Monitoring: Problem Determination Guide*.

---

**Viewing Tivoli Business Systems Manager data**

Tivoli Business Systems Manager data can be viewed from the Java Console, as shown in Figure 9 on page 91. The activity takes place without any interaction with Tivoli Monitoring components.
The data is viewed with Tivoli Business Systems Manager, as shown in the following Tivoli Business Systems Manager dialog.

Figure 9. Data flow for viewing Tivoli Monitoring data in Tivoli Business Systems Manager
This dialog shows the All Resources view of Tivoli Business Systems Manager. A Business Object Container has been opened to show an enterprise with at least two IP networks. IP network 69 has been opened to show a single subnet and segment within which are a number of computers. Computer lab03110-nt has been opened to show two entities:

**Operating System**

The properties of this object contain information about the resources of the computer, including the Tivoli Monitoring events.

**Software Element**

The properties of this object contain the heartbeat information.

**Note:** For every subnet in the enterprise hierarchy you can configure a subnet mask using the `wdmconfig` command (see “Configuring the Tivoli Business Systems Manager Adapter” on page 86).

---

### Enabling and disabling Tivoli Business Systems Manager integration

For each profile you create, you can choose to disable the Tivoli Business Systems Manager integration for all resource models in the profile. This is an override that ignores the Send to TBSM option that may be set up for any indication in any resource model in the profile (see “Customizing thresholds and event rules” on page 47).

To disable the Send to TBSM option, perform the following steps:

1. Open the Tivoli Monitoring Profile dialog.
2. Select Edit > Properties.

   The Properties dialog opens and displays the name of the profile for which the server identity is to be set, and the profile manager in which the profile is located. It also displays the check box **Send to TBSM**.

   ![Profile Properties](image)

3. If you want to send events in this profile to Tivoli Business Systems Manager, check the **Send to TBSM** option.

   **Note:** If you do not select this check box, no events are sent to the Tivoli Business Systems Manager, even if you selected the Send to TBSM check box in the Indications and Actions dialog.

4. Click **OK** to save the properties.
The Tivoli Monitoring Profile dialog reopens.

Managing the Tivoli Business Systems Adapter from the CLI

The following commands are available for managing the Tivoli Business Systems Manager Adapter from the command line interface:

`wdmconfig`
Updates the configuration file of the Tivoli Business Systems Manager Adapter.

`wdmdiscovery`
Runs a bulk or delta discovery on selected endpoints at selected gateways. The command should only be used if the Tivoli Business Systems Manager Adapter component is installed.

`wdmmngcache`
Deletes or lists part or all of the contents of the endpoint cache.

For full details of the commands, see [Chapter 8, “Commands,” on page 109](#).
Chapter 7. Integration with Tivoli Data Warehouse

This chapter describes how to integrate Tivoli Monitoring with Tivoli Data Warehouse, so that data collected with a user-defined resource model can be used by Tivoli Data Warehouse.

The following topics are covered:

- Overview
- Customer scenario using Tivoli Monitoring
- Enabling data logging to the central data warehouse
- Installing and configuring the Tivoli Enterprise Data Warehouse Support component

For more information about Tivoli Data Warehouse, see the following publications:

- Tivoli Data Warehouse: Installing and Configuring
- Tivoli Enterprise Data Warehouse: Enabling an Application for Tivoli Enterprise Data Warehouse
- Tivoli Monitoring Warehouse Enablement Pack: Implementation Guide

Overview

As shown in the following figure, Tivoli Data Warehouse consists of a central data warehouse where historical data from management applications (Source “n” in the figure) is stored, aggregated, and correlated.
A data mart is a subset of a data warehouse that contains data tailored and optimized for specific analysis and reporting needs. Both the initial extraction of data from the source application into the central data warehouse and the extraction of data from the central data warehouse into the specific data marts is handled by extract, transform, and load (ETL) processes.

The central data warehouse ETL reads the data from selected external sources, verifies the data, makes the data conform to the generic schema used by the warehouse, and places the data into the central data warehouse.

The data mart ETL extracts a subset of data from the central data warehouse, transforms it, and loads it into a data mart data model or into an application database to be used for application-specific purposes.

Tivoli Monitoring and any developed applications, which interact with Tivoli Data Warehouse, are responsible for running both ETL processes. In Tivoli Monitoring the central data warehouse ETL is called ETL1 (AMX in the product CD), while the data mart ETL is called ETL2 (AMY in the product CD for operating system data).

Tivoli Monitoring uses a proprietary database (accessible through an RDBMS Interface Module - RIM). The Tivoli Monitoring ETL1 process extracts data from the Tivoli Monitoring RIM database and loads it to the central data warehouse. The Tivoli Monitoring ETL2 process moves the data from the central data warehouse to a data mart.

The following sections describe these operations in more detail.
Customer scenario using Tivoli Monitoring

Using Tivoli Monitoring, this is a high level customer scenario that allows the integration with Tivoli Data Warehouse:

- The Independent Software Vendor (ISV Business Partner) developer or the Warehouse Pack developer use the Tivoli Monitoring Resource Model Builder tool to create a resource model with logging capability. In addition, the developer must create a DML SQL script containing information that will facilitate the logging of data into the central data warehouse. The script must be created according to the rules dictated by the following publications:
  - *Tivoli Enterprise Data Warehouse: Enabling an Application for Tivoli Enterprise Data Warehouse*
  - *Tivoli Monitoring Warehouse Enablement Pack: Implementation Guide*

- Alternatively, the developer may decide to use one of the resource models provided with the product. In this case, there is no need to create a DML SQL script because, for operating system data, Tivoli Monitoring takes care of logging the appropriate information to the central data warehouse.

- The Tivoli Monitoring operator installs the resource model on the Tivoli server, includes it in a profile, enables it for data logging into the central data warehouse, and distributes the profile to the endpoints to be monitored.

- The endpoint runs the resource model and data gets collected into the endpoint database.

- The operator uses the `wdmcollect` command to schedule the upload of data (through the MDist 2 service) from the endpoint database into the RIM database used by Tivoli Monitoring.

- Once the data is loaded to the database, the Tivoli Monitoring ETL1 process (AMX in the product CD), installed and scheduled on the Tivoli Data Warehouse, extracts the data from the database and loads it to the central data warehouse.

- As soon as ETL1 completes, the ETL2 process moves the information from the central data warehouse into a data mart that can provide sample reports. Although Tivoli Monitoring provides an ETL2 process (AMY in the product CD for operating system data), the various applications (Warehouse Packs) that need to use Tivoli Monitoring data are actually responsible for executing their own ETL2 process. In fact each application knows what kind of data it needs, what type of analysis needs to be performed on the data, and what type of reports need to be created.

Enabling data logging to the central data warehouse

You enable data logging to the central data warehouse by editing the data logging settings for each resource model. To do so, perform the following steps:

1. Open the Tivoli Monitoring Profile dialog.
2. Select the resource model that you want to customize.
3. Click **Edit**. The Edit Resource Model dialog opens.
4. In the Edit Resource Model dialog, click **Logging**.
   The Logging dialog opens.
5. In the **Data Logging Settings** group box, select the **Enable Data Logging** check box.

   This enables logging of all data collected by the resource model to the Tivoli Monitoring endpoint database.

6. To enable data logging to the central data warehouse, select one of the following check boxes:
   - **TEDW Data**
   - **Raw Data** and **TEDW Data**

   If you select **TEDW Data**, the Tivoli Monitoring engine aggregates the data collected during each cycle time and stores the results in the endpoint database after an aggregation period of 1 hour.

   If you select both **Raw Data** and **TEDW Data**, in addition to the data aggregated on the hour (**TEDW Data**), also the data logged at each cycle time (**Raw Data**) is stored in the endpoint database. However, only the data aggregated on the hour will be considered during the export towards the RIM database.

7. To save the changes, click **Apply Changes and Close**.

   The Logging dialog closes.

   The Tivoli Monitoring engine starts collecting data (for the central data warehouse) at the next full hour after profile distribution time. At that point, after an aggregation period of 1 hour, the data is stored into the endpoint database. Finally, 20 minutes later, the data is exported from the endpoint database into an XML file.

   The following figure shows the data logging process with respect to the time of the individual activities:
Activities such as the redistribution of the same profile, or the stop and restart of the engine, remove the data from cache, causing the data collection to start again at the next full hour and delaying the loading into the data warehouse.

Installing and configuring the Tivoli Enterprise Data Warehouse Support component

The Tivoli Enterprise Data Warehouse Support component of IBM Tivoli Monitoring Version 5.1.2 includes the files that enable the integration with Tivoli Data Warehouse.

This section explains how to install and configure the Tivoli Enterprise Data Warehouse Support component. The installation requires you to complete each of the following procedures in the order given:

1. Verifying the prerequisites
2. Installing the Tivoli Enterprise Data Warehouse Support component
3. Creating and configuring the database structure

Verifying prerequisites

The following are the prerequisites for using the Tivoli Enterprise Data Warehouse Support component:

- Ensure that an appropriate relational database management system (RDBMS, such as DB2, Informix, Microsoft SQL, Oracle, or Sybase) is installed either within the Tivoli region or on an accessible system outside it. If the RDBMS server software is installed within the Tivoli management region, the configuration steps below should be implemented to provide direct access to the RDBMS server. If the RDBMS server is installed outside the Tivoli management region, the RDBMS client component must be installed within the region, and the configuration steps must refer to the RDBMS client software.
Your database administrator is responsible for installing and maintaining the database. For information about installing and using your database, refer to the documentation provided with it.

- Ensure that Tivoli Monitoring is installed and updated on the Tivoli Management region server.

## Installing the Tivoli Enterprise Data Warehouse Support component

This component must be installed on the Tivoli management region server, and, optionally, on one or more managed nodes. You must install the component on the server before you install it on the managed nodes.

To install the component, follow this procedure:

1. Ensure that Tivoli Monitoring is installed and updated on all managed nodes (including the Tivoli Management region server) in your system.
2. In the Tivoli Desktop dialog, select Desktop → Install → Install Product. The Install Product dialog is displayed.
3. Select IBM Tivoli Monitoring 5.1.2 - Tivoli Enterprise Data Warehouse Support and click Install & Close. The Install Options dialog is displayed:

![Install Options](image)

4. This step creates the RDBMS Interface Module (RIM) object needed to interface with the RDBMS database.

   Fill out the Install Option fields as indicated in Table 13 and click Set. If you click Set without filling out the Install Options, the RIM object is created with the default parameters (for DB2).

### Table 13. Install options for the Tivoli Enterprise Data Warehouse Support component

<table>
<thead>
<tr>
<th>Database Vendor</th>
<th>The vendor name of the RDBMS product to manage the component data. The supported databases are Sybase, Oracle, MS-SQL, DB2, and Informix.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Home</td>
<td>The directory on the RIM host where the database software (either the RDBMS server or the client, if the server is outside the TMR) is installed.</td>
</tr>
<tr>
<td>Database ID</td>
<td>A unique name for the database. The default value is itm_db.</td>
</tr>
</tbody>
</table>
If you leave all fields blank and click Close (which prevents the RIM object from being created), you can set up the RIM object manually as described in “Creating a RIM object using a shell script” on page 105. You should then configure the data collectors already installed to work with the RIM object by using the wdmconfig command, see “[Configuring Tivoli Enterprise Data Warehouse Support” on page 104] for details. The name of the created RIM object is \texttt{itm\_rim\_RIM\_host\_name}. A default database user password is also associated with the RIM object: the default password is \texttt{itmitm}. After the RIM object is created, you may need to change the password (by using the Tivoli Management Framework command \texttt{wsetrimpw}) in order to have it match with the database password used by the Database User ID to connect to the database instance. Please note the following restrictions when changing the database password:

- For DB2 users, the password must match the password of the DB2 instance owner
- For Sybase users, the password must be at least 6 characters long
- For Informix users, the password must match the Informix NT or UNIX user password

5. The Product Install dialog opens. It provides a list of the operations to be performed and warns of any problems you might want to correct before installing.

6. Review the status information and click Continue Install. The Product Install dialog informs you when installation is complete. At this point, the installation has created a RIM object to interface the RDBMS on the RIM host, and configured all data collectors to work with it.

7. Verify that the RIM object has been created correctly by issuing the command \texttt{wgetrim itm\_rim\_RIM\_host\_name}. If any of the settings are not correct, use the \texttt{wsetrim} command with the appropriate options to change the object labeled \texttt{itm\_rim\_RIM\_host\_name}. If you change the RIM name you must update the configuration of the data collectors already installed (which are using the RIM object whose name has been modified) using the \texttt{wdmconfig} command.

Creating and configuring the database structure

This section explains how to create and configure the database structure (for the Tivoli Enterprise Data Warehouse Support component) from a shell script (an

\begin{table}[h]
\centering
\begin{tabular}{|c|l|
\hline
Database User ID & ID of the user who is authorized to access the database. The default value is blank. If the database is DB2, due to a known RIM limitation, the user ID must be equal to the instance name label and it must be defined with administrative privileges. If the database is Sybase, or any case sensitive database, the user ID must be in uppercase characters. \\
\hline
Database Server ID & The name of the RDBMS server. This is an alias to enable client/server connection. \\
\hline
Instance Name (DB2 only) & DB2 instance name. \\
\hline
Purge Time Interval (days) & All data older than the specified number of days is removed from the database. \\
\hline
Rim Host Name (label) & The label of the RIM host system, where the RDBMS server or client is installed. If this field is filled in, the RIM object is created on the managed node indicated by the field. If this field is not filled in, the RIM object is created on the TMR server. \\
\hline
\end{tabular}
\caption{Install options for the Tivoli Enterprise Data Warehouse Support component (continued)}
\end{table}
alternative method using the SQL processor is described in “Creating a database structure using the SQL processor” on page 107. There are three basic steps:

- Verifying the prerequisites
- Creating and configuring the database structure
- Verifying the RIM object connection

### Verifying the prerequisites

The prerequisites for this procedure are as follows:

1. Read the description of the procedure in “Creating and configuring the database structure,” and determine the system on which you run the procedure. If this system is inside the Tivoli region, ensure that it has the Tivoli environment set. To set the Tivoli environment run the following command, depending on your environment (the location of the command shown here is the default):

   **UNIX/Linux**
   ```bash
   /etc/Tivoli/setup_env.sh
   ```

   **Windows**
   ```cmd
   /system32/winnt/drivers/etc/Tivoli/setup_env.cmd
   ```

2. Before creating the database structure for Tivoli Enterprise Data Warehouse Support, ensure you have backed up your RDBMS database.

### Creating and configuring the database structure

The steps are as follows:

1. Determine the system from where the procedure is to be run, which depends on the database vendor and the operating system:

   **Creating Oracle, Sybase, or Microsoft-SQL Databases**
   The procedure should be run from the RIM host where the database server or client is installed, and in the following way:
   - For Windows NT and Windows 2000, from a Tivoli bash shell
   - For UNIX, from any UNIX shell

   **Creating DB2 and Informix Databases**
   The procedure should be run from a DB2 or Informix command line on the database server. This is because you cannot execute the DB2 database creation script from the *DB2 Client Command Line Processor* without an existing database connection and because Informix uses the dbaccess facility, which is shipped as part of the Informix server.

2. If the database server or client on which you need to run the procedure has Tivoli Enterprise Data Warehouse Support installed, from the `$BINDIR/TME/Tmw2k/Warehousecfg` directory run the `cr_itm_db.sh` script to create the database structure.

3. If the database server or client on which you need to run the procedure is inside the Tivoli region, and does *not* have Tivoli Enterprise Data Warehouse Support installed, you should proceed as follows, depending on the operating system of the database server or client:

   **UNIX/Linux**
   Mount the `$BINDIR/TME/Tmw2k/warehousecfg` directory of any system with Tivoli Enterprise Data Warehouse Support installed (for example, the Tivoli server) as an NFS mount on the database server or client, and run the procedure described in step 2 from within that directory.

   **Windows**
   Follow these steps:
a. Copy the following files from the $BINDIR/TME/Tmw2k/Warehousecfg directory from any system where Tivoli Data Warehouse has been installed:
   cr_itm_db.sh
   cr_db.<database_extension>
   cr_tbl.<database_extension>

   where database_extension is one of the following, depending on your database vendor:
   - DB2 db2
   - Informix inf
   - Microsoft-SQL 6.x mssql
   - Microsoft-SQL 7.0 mssql7
   - Oracle ora
   - Sybase syb

b. Run the cr_itm_db.sh script to create the database structure.

4. If the database server on which you need to run the procedure is outside the Tivoli region, you should proceed as follows, depending on the operating system of the server:

   Follow these steps:
   a. Run the procedure described in step [2 on page 102] on any system with Tivoli Enterprise Data Warehouse Support installed, ignoring any error messages that are displayed because the database is not found.
   b. Copy the resulting cr_db.<database_extension>.sql file to the database server or client, where database_extension is one of the following, depending on your database vendor:
      - DB2 db2
      - Informix inf
      - Microsoft-SQL 6.x mssql
      - Microsoft-SQL 7.0 mssql7
      - Oracle ora
      - Sybase syb
   c. Run the file in the SQL processor on the database server.

   **Note:** If you run the database creation script from a shell with the Tivoli environment set on the Tivoli Enterprise Data Warehouse Support RIM host, the script cr_itm_db.sh attempts to fetch your database configuration from the RIM object attributes. If you do not have the Tivoli Monitoring RIM object already created, you are asked for the parameters identified in [Table 13 on page 100] in a series of questions. You are also asked to enter other parameters, as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Userld password</td>
<td>Password of database to be configured</td>
</tr>
<tr>
<td>Database Device (Sybase and Microsoft-SQL only)</td>
<td>Do not use the master device for the Tivoli Enterprise Data Warehouse Support database. The master database, model database, and temporary database all reside on the master device. Currently, the master device cannot be expanded onto any other device.</td>
</tr>
<tr>
<td><strong>Database Space</strong> (Informix only)</td>
<td>Do not use the rootdbs dbspace for the Tivoli Enterprise Data Warehouse Support database. You must create a separate dbspace for this database.</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Database Size</strong></td>
<td>The size (in MB) of the Tivoli Enterprise Data Warehouse Support database to be created. For Sybase, Microsoft-SQL, and Informix only, the maximum size is the size of the device (or dbspace) dedicated for the Tivoli Enterprise Data Warehouse Support database.</td>
</tr>
<tr>
<td><strong>Database administrator password</strong></td>
<td>Administrator password for database (sa password for DB2)</td>
</tr>
</tbody>
</table>

5. When prompted to do so, enter the database administrator passwords.

**Verifying the RIM object connection**

After creating the database structure for Tivoli Enterprise Data Warehouse Support, test the RIM connection to the database by means of the following Tivoli Management Framework command issued from the Tivoli server:

```bash
wrimtest -l itm_rim_RIM host name
```

For example:

```
Resource Type: RIM
Resource Label: itm_rim_RIM host name
Host Name: amadeus
User Name: DM
Vendor: MS_SQL
Database: dm_db
Database Home:d:\mssql7
Server ID: Amedeus
Instance Home:
Opening Regular Session...Session Opened
RIM: Enter Option >
```

Type x and press Enter to release the session.

**Configuring Tivoli Enterprise Data Warehouse Support**

The Tivoli Enterprise Data Warehouse Support component is configured by using the command `wdmconfig`. You can address this command to one, a list, or all gateways in the region, and it allows you to supply any number of key/value pairs as parameters. The key/value pairs that you need to configure are as follows:

- **datacollector.rim_name**
  Specifies the name of the RIM object that the data collection process uses to load data to the database. The default is `itm_rim_RIM host name`.

- **datacollector.db_purge_interval**
  Specifies the number of days the data is kept on the database: older data is automatically removed from the database. The value can range from 10 to 60. The default is 30 days.

- **datacollector.delay**
  Specifies the time delay (in minutes, compared to the hour) after which the data collector process uploads data from the endpoints. The value can range from 10 to 60 minutes. The default is 30 minutes.

- **datacollector.sleep_time**
  Specifies the time interval (in minutes) between two consecutive requests of data uploading generated by the data collector processor. The value can range from 10 to 60 minutes. The default is 10 minutes.
**datacollector.max_retry_time**

Specifies the maximum number of times an XML data file must be processed before being archived when an error occurs. The default is 3 times.

All these key/value pairs and their values are contained in file: $DBDIR/dmml/.config

This file can be read but it must not be modified manually. If you want to modify it, use the `wdmconfig` command (note that the applied changes might not show).

For full details of the command syntax see “wdmconfig” on page 121.

**Uploading data using the CLI**

Use the `wdmcollect` command to schedule or stop the upload of data from the endpoint database into the RIM database. Then, during the ETL1 process, the data is extracted, transformed, and loaded into the central data warehouse.

The `wdmcollect` command options specify:

- The endpoints from which the data is uploaded
- The interval time between two upload operations

Refer to the “`wdmcollect`” on page 119, for details about this command.

The data collection process uses the MDist 2 service to move data from the endpoints to the managed node, therefore, the managed node must be configured as a Repeater.

**Alternative installation and configuration procedures**

This section explains in detail the following alternative installation and configuration procedures:

- Creating a RIM object using a shell script
- Creating a database structure using the SQL processor

**Creating a RIM object using a shell script**

This section contains instructions for creating the RIM object using a shell script.

To create a RIM object, Tivoli Framework applications use the `wcrtrim` command. Tivoli Enterprise Data Warehouse Support provides you with a shell script called `cr_tedw_rim.sh` that prompts you for the required input and creates a RIM object called `itm_rim_<RIM host name>`.

You should run this script from the Tivoli management region server or managed node where you have installed the Tivoli Enterprise Data Warehouse Support component. You should have the role of a Tivoli administrator.

Note that once you have run the script, you must use the `wdmconfig` command to define the following parameters (see “Configuring Tivoli Enterprise Data Warehouse Support” on page 104 for details):

- `datacollector.rim_name`
- `datacollector.db_purge_interval`
The `cr_tedw_rim.sh` script attempts to retrieve your RIM object attributes. If the `cr_tedw_rim.sh` cannot find the RIM object, you are prompted to enter the required information that will be used by the `wcrtrim` command. The procedure has the following steps:

1. Change to the following directory: `$BINDIR/TME/Tmw2k/warehousecfg`
2. Run the RIM creation script as follows:

<table>
<thead>
<tr>
<th>Platform</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows NT</td>
<td><code>sh cr_tedw_rim.sh</code></td>
</tr>
<tr>
<td>UNIX</td>
<td><code>cr_tedw_rim.sh</code></td>
</tr>
</tbody>
</table>

Information you are required to supply depends on the vendor product requirements for the specific database. You can also use this script to change the RIM object attributes. If a RIM object `itm_rim_RIM host name` already exists, you are asked to remove it. If the displayed values are correct, reply `Y`, and they remain unaltered. Otherwise, reply anything other than `Y`, and create a new RIM object by supplying the parameters described in Table 13 on page 100.

Table 14 gives vendor-specific information for the RIM parameters. The titles to each column show not only the name of the attribute, but also the option used with the `wcrtrim` command in the `cr_tedw_rim.sh` script.

You should note that some of the parameters use database-specific environment variables, highlighted in bold.

<table>
<thead>
<tr>
<th>Database Vendor (-v)</th>
<th>Database ID(-d)</th>
<th>Database User ID (-u)</th>
<th>Database Home (-H)</th>
<th>Database Server ID (-s)</th>
<th>Instance Home (-I)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2</td>
<td>Instance owner id</td>
<td>DB2 CAE install directory</td>
<td>You must specify the string <code>tcpip</code> $DB2DIR</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Informix</td>
<td>Database source name (DSN) defined in the ODBC control panel (NT) or <code>.odbc.ini</code> file (UNIX)</td>
<td>Informix user</td>
<td>Informix client install directory</td>
<td>Informix server name defined in sqlhosts</td>
<td>N/A</td>
</tr>
<tr>
<td>Microsoft_SQL</td>
<td><code>itm_db</code></td>
<td>Microsoft-SQL user for Tivoli Monitoring</td>
<td>Microsoft-SQL client install directory</td>
<td>Hostname for Microsoft-SQL Server host</td>
<td>N/A</td>
</tr>
<tr>
<td>Oracle</td>
<td><code>$ORACLE_SID</code> (or service name for Oracle 8.1+)</td>
<td>Oracle user for Tivoli Monitoring</td>
<td>Oracle client install directory <code>$ORACLE_HOME</code></td>
<td><code>$TWO_TASK</code></td>
<td>N/A</td>
</tr>
</tbody>
</table>
### Creating a database structure using the SQL processor

Your database administrator can customize the SQL templates, such as `cr_db.xxx` and `cr_tbl.xxx`, (where `xxx` can be `syb`, `ora`, `mssql`, `mssql7`, `inf`, or `db2`), and then run these scripts on the RDBMS client or server using the interactive SQL processor.

<table>
<thead>
<tr>
<th>RDBMS Vendor</th>
<th>Configuration File</th>
<th>Interactive SQL Processor</th>
<th>RDBMS Administrator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle</td>
<td>tnsnames.ora</td>
<td>sqlplus</td>
<td>sys</td>
</tr>
<tr>
<td>Sybase</td>
<td>interfaces</td>
<td>isql</td>
<td>sa</td>
</tr>
<tr>
<td>Microsoft-SQL</td>
<td>interfaces</td>
<td>isql</td>
<td>sa</td>
</tr>
<tr>
<td>DB2</td>
<td></td>
<td>db2</td>
<td>instance owner</td>
</tr>
<tr>
<td>Informix</td>
<td>sqlhost</td>
<td>dbaccess</td>
<td>informix</td>
</tr>
</tbody>
</table>

To run the RDBMS configuration scripts on the RDBMS client or server using the SQL processor, perform the following steps as shown for Sybase:

1. On the Tivoli management region server where Tivoli Enterprise Data Warehouse Support is installed, customize the `cr_db.syb` and `cr_tbl.syb` scripts to meet your needs and then save them as `cr_db_syb.sql` and `cr_tbl_syb.sql`.

2. Copy the `cr_db_syb.sql` and `cr_tbl_syb.sql` files from the `$BINDIR/TME/Tmw2k/Warehousecfg` directory on the managed nodes where Tivoli Monitoring is installed, to a temporary directory on the RDBMS server.

3. From the directory that now contains the script, start an `isql` session as super administrator (sa) and run the `cr_db_syb.sql` script as follows:

   ```
   isql -U sa -P pwd -i cr_db.syb.sql
   ```

   where `pwd` is the RDBMS password for the RDBMS user system administrator.

   The script creates the Tivoli Monitoring user and the Tivoli Monitoring database in the Sybase RDBMS.

4. Install the layout by entering the following command:

   ```
   isql -U ITM -P pwd -i cr_tbl.syb.sql
   ```

   where `pwd` is the RDBMS password for the user. The password is the one you specified in the creation of the Tivoli Monitoring RIM object.

5. Prepare to test the configuration by entering the following command:

   ```
   isql -U ITM -P pwd
   ```

   where `pwd` is the RDBMS password for the user.

6. In the SQL session, check that the Tivoli Monitoring repository was installed, by entering the following:
> select * from table
> go

where table is one of the following tables:
- CATEGORIES
- CATEGORIESDATA
- ENDPOINTS
- INSTANCES
- METRICS
- METRICSDATA
- RESOURCES
- RMPROFILES

Results should indicate that zero rows were found. If results indicate that <table> is unknown, the Tivoli Monitoring repository was not installed.

7. Log out of ISQL by entering the following command:
   > quit
Chapter 8. Commands

This chapter describes the Tivoli Monitoring commands you can issue from the command line in the Tivoli environment, and how to use them.

The following special characters are used to define the command syntax:

- `[ ]` Identifies optional arguments. Arguments not enclosed in brackets are required.

- `...` Indicates that you can specify multiple values for the previous argument.

- `|` Indicates mutually exclusive information. You can use the argument to the left of the separator or the argument to the right of the separator. You cannot use both arguments in a single use of the command.

- `{ }` Delimits a set of mutually exclusive arguments when one of the arguments is required. If the arguments are optional, they are enclosed in brackets `[ ]`.

Upgraded command names

In Tivoli Monitoring, all commands have the prefix `wdm`. The following steps have been taken to assist the transition from the old command prefixes to the new prefixes:

- Five of the old command names have been included with this release only of the product as aliases for the new equivalent commands.
- Three of the old commands (wtmntaddrm, wtmntdefrm, wtmntrmrm) have been merged into one new command (wdmrm). The old commands have been made available with this release only, with their old unchanged syntax and options.
- Eight new commands have been added, for which aliases with the old prefix are not supplied.

Thus, you are recommended to switch to the new command names as soon as possible, and in case before any subsequent release of the product. Table 15 shows full details:

Table 15. New and old command names

<table>
<thead>
<tr>
<th>New command names</th>
<th>Old command names, retained for this release</th>
<th>Obsolete merged commands, retained for this release</th>
</tr>
</thead>
<tbody>
<tr>
<td>wdmcheckprereq</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wdmcmd</td>
<td>wtmntcmd</td>
<td></td>
</tr>
<tr>
<td>wdmcmddistrib</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wdmcollect</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wdmconfig</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wdmdiscovery</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wdmdistrib</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wdmdumpprf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wdmeditprf</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The commands are described in this appendix using their new names only.

The commands in this appendix are presented in alphabetic order, and are also listed alphabetically in the index. However, what follows is a grouping of the available commands according to their primary function.

Commands for managing profiles at the Tivoli server

The following commands are available for managing profiles at the Tivoli server:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>wdmeng</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wdmheartbeat</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wdmloadprf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wdmnseng</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wdmnn</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wdmnngcache</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wdmtrceng</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wdmrrm</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wdmloadrp</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Command for managing default resource models at the Tivoli server

The following command is available for managing default resource models at the Tivoli server:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>wdmrm</td>
<td>Adds, lists or removes a specified default resource model at the Tivoli management region server or managed node/gateway from where it is issued. It also adds the NLS catalog to an already installed default resource model.</td>
<td>162</td>
</tr>
</tbody>
</table>

Commands for managing the Tivoli Business Systems Manager Adapter

The following command is available for managing the Tivoli Business Systems Manager Adapter:
<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>wdmdiscovery</td>
<td>Runs a bulk or delta discovery on selected endpoints at selected managed nodes/gateways. The command should only be used if the Tivoli Business Systems Manager Adapter component is installed.</td>
<td>127</td>
</tr>
</tbody>
</table>

**Commands for managing Tivoli Monitoring at the endpoints**

The following commands are available for managing Tivoli Monitoring at the endpoints:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>wdmcmd</td>
<td>Stops or restarts Tivoli Monitoring on one or more endpoints from a managed node/gateway or server.</td>
<td>113</td>
</tr>
<tr>
<td>wdmcmdistrib</td>
<td>Stops or starts Tivoli Monitoring on one or more endpoints from a managed node/gateway or server. It uses the MDist 2 service.</td>
<td>115</td>
</tr>
<tr>
<td>wdmeng</td>
<td>Stops or starts profiles or resource models at endpoints; also deletes profiles at endpoints.</td>
<td>146</td>
</tr>
<tr>
<td>wdmlseng</td>
<td>Returns a list and the status of all resource models that have been distributed on a specified endpoint.</td>
<td>153</td>
</tr>
<tr>
<td>wdmtrceng</td>
<td>Sets the trace parameters of the Tivoli Monitoring engine at the endpoint.</td>
<td>164</td>
</tr>
<tr>
<td>wdmcollect</td>
<td>Starts or stops the collection of data (which is stored into a central data warehouse) from selected endpoints</td>
<td>119</td>
</tr>
<tr>
<td>wdmcheckprereq</td>
<td>Checks the prerequisite software on a Windows endpoint.</td>
<td>112</td>
</tr>
</tbody>
</table>

**Commands for managing Tivoli Monitoring at managed nodes/gateways**

The following commands are available for managing the product at managed nodes/gateways:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>wdmconfig</td>
<td>Updates the configuration file of the Tivoli Business Systems Manager Adapter</td>
<td>121</td>
</tr>
<tr>
<td>wdmheartbeat</td>
<td>Stops or starts the ‘heartbeat’ monitoring of endpoints, changes its frequency and queries the status of the heartbeat processor.</td>
<td>148</td>
</tr>
<tr>
<td>wdmnn</td>
<td>Stops or starts selected Tivoli Monitoring processes on one or all managed nodes/gateways.</td>
<td>157</td>
</tr>
<tr>
<td>wdmnnmgcache</td>
<td>Deletes or lists part or all of the contents of the Tivoli Business Systems Manager Adapter cache.</td>
<td>159</td>
</tr>
</tbody>
</table>
**wdmcheckprereq**

Checks a number of software prerequisites on a Windows endpoint, from a managed node/gateway or server.

**Syntax**

`wdmcheckprereq -e endpoint`

**Description**

The `wdmcheckprereq` command checks a set of prerequisites on a Windows endpoint and returns the results. The set of prerequisites verified by the command is the following:

- WMI version
- ODBC driver version
- Jet Engine version
- JRE version
- Windows Script Host (CScript) version

Refer to the *IBM Tivoli Monitoring: User’s Guide*, chapter "Installing", section "Prerequisite Software", for the minimum versions required by each of the above prerequisites.

When the command is directed to verify the prerequisites of a UNIX or OS/400 endpoint, the command line returns an error.

**Options**

- `-e endpoint`
  
  Specifies the name of the Windows endpoint whose software needs to be checked.

**Authorization**

Requires the super, senior or admin roles.

**Examples**

1. The following example checks the prerequisites on endpoint MyEndpoint:

   `wdmcheckprereq -e MyEndpoint`
wdmcmd

Stops or restarts Tivoli Monitoring on one or more endpoints, from a managed node/gateway or server.

Syntax

```text
wdmcmd –restart {–p profile_manager#region | –e endpoint | –m {managed_node | all}}

wdmcmd –stop {–p profile_manager#region | –e endpoint | –m {managed_node | all}}
```

Description

The `wdmcmd` command stops or restarts the Tivoli Monitoring endpoint engine on all the endpoints that are subscribed to the profile manager if the `-p` option is used, or only on the endpoints listed if the `-e` option is used. When the restart (or stop) is requested using the `-m` option, all endpoints of a specified managed node (or all endpoints for all managed nodes in the Tivoli region) are restarted (or stopped).

Options

- **-e endpoint**
  Specifies one or more endpoints by endpoint label: if more than one endpoint label is to be targeted, they should be separated by spaces.

- **-p profile_manager#region**
  Specifies the profile manager whose subscribers are the target of the command. The profile manager name must always be fully qualified with the policy region name as shown in the syntax.

  If you use this option in conjunction with the `-restart` option, all the profiles that were distributed to the target endpoints are restarted, including those belonging to other profile managers.

- **-m {managed_node | all}**
  Specifies the managed nodes/gateways whose endpoints will be restarted or stopped. Only the endpoints registered to the managed node are affected.

  The options are as follows:

  - **managed_node**
    The name of the managed node/gateway on which the command is to be performed.

  - **all**
    All managed nodes/gateways in the Tivoli region

- **-restart**
  Restarts the endpoint engine on the endpoint or endpoints specified.

- **-stop**
  Stops the endpoint engine on the endpoint or endpoints specified.

Authorization

Requires the super, senior or admin roles.
Examples

1. The following example stops all the endpoints that belongs to the profile manager MyProfileManager.
   wdmcmd –stop -p MyProfileManager

2. The following example restarts the endpoint engine on the endpoints myEP1 and myEP2.
   wdmcmd –restart -e myEP1 myEP2

See also

   wdmcmddistrib

   wdmeng

   wdmilseng

   wdmtrceng
wdmcmddistrib

Stops or starts (if not already running) Tivoli Monitoring on one or more endpoints in an asynchronous fashion, from a managed node/gateway or server. The upgrade option upgrades the endpoint binaries and redistributes all profiles.

Syntax

wdmcmddistrib -start { -p profile_manager\#region | -e endpoint | @endpoints_file } | -m {managed_node | all} | [ -t timeout ] | -d dir }

wdmcmddistrib -stop { -p profile_manager\#region | -e endpoint | @endpoints_file } | -m {managed_node | all} | [ -t timeout ] [ -d dir ]

wdmcmddistrib -upgrade { -p profile_manager\#region | -e endpoint | @endpoints_file } | -m {managed_node | all} | [ -t timeout ] [-d dir ] -M { none | all } | @models_file }

Description

The wdmcmddistrib command works like wdmcmd but, as opposed to wdmcmd, uses the MDist 2 service. The -t option is used to specify a timeout that is passed to the MDist 2 service.

wdmcmddistrib stops, starts, or upgrades the Tivoli Monitoring endpoint engine on all the endpoints that are subscribed to the profile manager if the -p option is used, or only on the endpoints listed if the -e option is used. When the start (or stop, or upgrade) is requested using the -m option, all endpoints of a specified managed node (or all endpoints for all managed nodes in the Tivoli region) are started (or stopped, or upgraded). Note that the -start option has effect only if the endpoint engine is not running; if the engine is running, the option has no effect.

The results of the execution of the wdmcmddistrib command are saved in a collection of files as follows:

- for the -stop option:
  - requested_ep_list.txt: provides the list of all target endpoints
  - success_stop_ep_list.txt: provides the list of all endpoints for which the stop was successful
  - unsuccessful_stop_ep_list.txt: if present, provides the list of all endpoints for which the stop was unsuccessful
- for the -start option:
  - requested_ep_list.txt: provides the list of all target endpoints
  - success_start_ep_list.txt: provides the list of all endpoints for which the start was successful
  - unsuccessful_start_ep_list.txt: if present, provides the list of all endpoints for which the start was unsuccessful
- for the -upgrade option:
  - requested_ep_list.txt: provides the list of all target endpoints
  - success_stop_ep_list.txt: provides the list of all endpoints for which the stop was successful
  - unsuccessful_stop_ep_list.txt: if present, provides the list of all endpoints for which the stop was unsuccessful
  - success_start_ep_list.txt: provides the list of all endpoints for which the start was successful
- `unsuccessful_start_ep_list.txt`: if present, provides the list of all endpoints for which the start was unsuccessful
- `success_upgrade_ep_list.txt`: provides the list of all endpoints for which the upgrade of the binaries was successful
- `unsuccessful_upgrade_ep_list.txt`: if present, provides the list of all endpoints for which the upgrade of the binaries was unsuccessful
- `updated_models_list.txt`: provides the list of resource models that have been updated and need to be redistributed. This file is saved only when the `-M` option is used with argument `@models_file`.
- in the directory `ProfilesDiscoveredAtEndpoints` the command saves a set of files associating each profile to the list of target endpoints. These are the profiles discovered at each target endpoint.
- in the directory `ProfilesToBeRedistributed` the command saves a set of files associating each profile to the list of target endpoints. These are the profiles that, according to the contents of `updated_models_list.txt`, need to be redistributed to the target endpoints.
- in the directory `ProfilesRedistributed` the command saves a set of files associating each profile to the list of target endpoints. These are the profiles that, according to the contents of `updated_models_list.txt`, have been redistributed to the target endpoints.

The results of the execution are saved in the directory specified by the `-d` option. If this option is not used, the following default directories are used:

```
$DBDIR/AMW/logs/wdmcmddistrib_mmddyy_hhmss/start
$DBDIR/AMW/logs/wdmcmddistrib_mmddyy_hhmss/stop
$DBDIR/AMW/logs/wdmcmddistrib_mmddyy_hhmss/upgrade
```

### Options

- **-d dir**
  Specifies the directory where the results of the command are saved. If not indicated, the default directories are used.

- **-e { endpoint | @endpoints_file }**
  Specifies the endpoint or list of endpoints (included in a file) that will be started, stopped, or upgraded.

  The options are as follows:

  - `endpoint`
    Specifies one or more endpoints by endpoint label: if more than one endpoint is to be targeted, the endpoint labels should be separated by spaces.

  - `@endpoints_file`
    A file containing endpoint names (one name for each line in the file). The `@` symbol is an indication that a file name follows, but the symbol should not be part of the filename.

- **-m {managed_node | all}**
  Specifies the managed nodes/gateways whose endpoints will be started, stopped, or upgraded.

  The options are as follows:
managed_node

The name of the managed node/gateway on which the command is to be performed.

all

All managed nodes/gateways in the Tivoli region

-M { none | all | @models_file }

Specifies which profiles are distributed during the upgrade. If none is used, no profiles are distributed (only the endpoint binaries are upgraded); if all is used, all profiles are distributed; if @models_file is used, only the profiles containing the resource models specified in @models_file are distributed.

-p profile_manager#region

Specifies the profile manager whose subscribers are the target of the command. The profile manager name must always be fully qualified with the policy region name as shown in the syntax.

If you use this option in conjunction with the –start option, all the profiles that were distributed to the target endpoints are started, including those belonging to other profile managers.

–start

Starts the endpoint engine on the endpoint or endpoints specified. This option has effect only if the engine is not running.

–stop

Stops the endpoint engine on the endpoint or endpoints specified.

–t timeout

Defines the timeout in seconds passed to the MDist 2 service. The distribution of stop/start/upgrade is scheduled with a no retry option and with the defined timeout. If the distribution completes before the timeout expires, the command completes too. The default timeout is 300 seconds (5 minutes). The timeout value must be greater than the notify_interval keyword value that the wmdist command uses (the default value for such keyword is 30 minutes). If the value is smaller, the timeout is automatically adjusted to a value compatible with the value of notify_interval.

–upgrade

Upgrades the endpoint binaries and redistributes the profiles according to the specifications of the –M option.

Authorization

Requires the super, senior or admin roles.

Examples

1. The following example stops all the endpoints that belongs to the profile manager MyProfileManager.

   wdmcmddistrib –stop -p MyProfileManager

2. The following example starts the endpoint engine on endpoints myEP1 and myEP2 using a timeout of 180 seconds.

   wdmcmddistrib –start -e myEP1 myEP2 -t 180

3. The following example upgrades the endpoint engine on endpoint myEP1 and redistributes all profiles to it.

   wdmcmddistrib –upgrade -e myEP1 -M all
See also

wdmcmd
wdmeng
wdmlseng
wdmtrceng
wdmcollect

Starts or stops the collection of data from one or more endpoints. The command can only be used after the Tivoli Enterprise Data Warehouse Support component is installed.

Syntax

\[
\text{wdmcollect} \{[\{-e \{endpoint\_name | @endpoints\_file\} | -s time\_interval | -t | -r \} | -t \} \mid -q] \mid \{-m \text{managed\_node} \} \mid \{-m \text{all}\} \mid -t \mid -q)\]
\]

Options

- `e {endpoint\_name | @endpoints\_file}`
  Specifies the endpoint or list of endpoints (included in a file) from which the data will be collected.

  The options are as follows:

  `endpoint\_name`
  The name of the endpoint on which the command must have effect.

  `@endpoints\_file`
  A file containing endpoint names (one name for each line in the file). The @ symbol is an indication that a file name follows, but the symbol should not be part of the filename.

- `-p`
  Purges all data from the RIM database. This is done according to the value of the `datacollector.db\_purge\_interval` key that you can specify or modify using the `wdmconfig` command.

- `-r`
  Processes all XML files that were archived on the managed node for the specified endpoints (in the corresponding expired directory). The files were archived when the `datacollector.max\_retry\_time` key was exceeded (see the `wdmconfig` command for details).

- `-s` `time\_interval`
  Specifies the time interval (expressed in hours and as an integer number in the range 1 to 24) between two subsequent data collections.
wdmcollect

-t
Stops the data collection from the endpoint or list of endpoints. When no endpoints are specified, it stops the collection on all endpoints. All pending data collection requests are deleted: if you do not want the pending requests to be deleted, use the wdmnn command (with options -stop and -d) to stop the data collection without deleting the pending requests.

-q
Queries the status of the data collection process.

-m managed_node
Specifies the managed node that performs the data collection (using its own data collector). You may choose to start/stop/query the data collection on all managed nodes in the Tivoli region by using -m all, along with either the -t or -q option. If the -m option is not used the data collection is performed on the managed node from which the command is being run.

Authorization
Requires the super, senior, admin, or user roles.

Examples
1. The following example initiates every 2 hours the collection of data from the list of endpoints specified in file my_endpoints.txt.
   wdmcollect -e @my_endpoints.txt -s 2
2. The following example stops the data collection from endpoint myEP1.
   wdmcollect -e myEP1 -t
3. The following example queries the status of all data collector processes on all managed nodes.
   wdmcollect -m all -q
4. The following example initiates every hour the data collection from endpoint myEP1 using the data collector on managed node myMN2
   wdmcollect -e myEP1 -s 1 -m myMN2

See also
wdmcmd
wdmeng
wdmlseng
wdmtrceng
**wdmconfig**

Updates the configuration file of a managed node/gateway.

**Syntax**

```
wdmconfig [-m {managed_node | @managed_nodes_file | all}]
           [-D key=value [-D key=value] ... | -G key]
```

**Description**

The `wdmconfig` command operates on one or more selected managed nodes/gateways, and allows you to change the value of any of the configuration keys to the value you require.

**Options**

```
-D key=value [-D key=value]...
```

Identifies one or more configuration keys to be updated and the new `value` that each key is to take. At least one key/value pair must be supplied. Do not use blanks around the "=" character or, if you use blanks, enclose the key/value pair within quotation marks. The keys that can be modified are as follows (listed in alphabetic order):

- **adapter.trace.enable**
  Set this to `true` if you want to store all trace messages regarding the operations of the adapter. The messages are stored in the file identified in `trace.filename`. The default is `false`.

- **adapter.trace.level**
  If you have enabled adapter trace messages, set this to `low`, `medium` or `high`, according to the level of details you require. The default is `low`.

- **adapter.working.dir**
  Working directory that will be used by the adapter. The default, which you are recommended to use, is the Tivoli Monitoring middle layer directory (`$DBDIR/dmml`).

- **adapter.xml.validation**
  Set this to `true` if you want to enable the xml validation. The default is `false`.

- **core.trace_level**
  Specifies the level of the profile core trace. The minimum value is 0, the maximum is 2, the default is 1.

- **core.trace_size**
  Specifies the size in bytes of the profile core trace. The default is 500000.

- **datacollector.db_purge_interval**
  Specifies the number of days the data is kept on the database: older data is automatically removed from the database. The value can range from 10 to 60. The default is 30 days.

- **datacollector.delay**
  Specifies the time delay (in minutes, compared to the hour) after which the data collector process uploads data from the endpoints. The value can range from 10 to 60 minutes. The default is 30 minutes.
**wdmconfig**

**datacollector.max_retry_time**
Specifies the maximum number of times an XML data file must be processed before being archived when an error occurs. The default is 3 times.

**datacollector.rim_name**
Specifies the name of the RIM object that the data collection process will use to load data to the database. The default is **itm_rim_RIM host name**.

**datacollector.sleep_time**
Specifies the time interval (in minutes) between two consecutive requests of data uploading generated by the data collector processor. The value can range from 10 to 60 minutes. The default is 10 minutes.

**datacollector.trace_level**
Specifies the level of the data collector trace. The minimum value is 0, the maximum is 2, the default is 1.

**datacollector.trace_size**
Specifies the size in bytes of the data collector trace. The default is 500000.

**dmml.trace_level**
Specifies the level of trace (for all components) from 0 (minimal) to 4 (verbose); the default is 1. For more details on traces, refer to the *IBM Tivoli Monitoring: Problem Determination Guide*.

**dmml.trace_size**
Specifies the size of the trace (for all components) in bytes; the default is 500000. For more details on traces, refer to the *IBM Tivoli Monitoring: Problem Determination Guide*.

**gw.trace_level**
Specifies the level of the endpoint upcall trace (tmnt_gtw_eng process). The minimum value is 0, the maximum is 2, the default is 0.

**gw.trace_size**
Specifies the size in bytes of the endpoint upcall trace (tmnt_gtw_eng process). The default is 500000.

**heartbeat.reboot_engine_if_down**
Set this to **true** if you want the heartbeat to restart the engine that was stopped abnormally. A heartbeat event is sent, if you configured sending events. Note that if the engine was stopped using **wmdcmd –stop**, the heartbeat does not restart the engine.

**heartbeat.reg_ep_time**
When this configuration key is used, the engine performs periodic upcalls to register itself in the request manager. This may be useful to, for example, force the migration of an endpoint when a gateway goes down, or to keep the request manager cache up to date. To enable this mechanism, stop the engine, set the key specifying the frequency in seconds (for example, heartbeat.reg_ep_time=1200 enables an upcall every 20 minutes) and restart the engine. To disable or modify the mechanism, it is NOT necessary to stop the engine. If the configuration key is not set, or if it is set to 0, the upcall occurs just once. Also note that this key, in spite of its name, is not related to the way the heartbeat works.

**heartbeat.send_endpoint_migrated_event**
Set this to **true** if you want to enable the generation of **Heartbeat_EndpointMigrated** events; otherwise leave as the default value of **false**.
heartbeat.send_events_to_notice
Set this to false if you do not want to send heartbeat events to the Tivoli Monitoring notice group; otherwise leave as the default value of true.

heartbeat.send_events_to_tbsm
Set this to true if you want to send heartbeat events to the Tivoli Business Systems Manager; otherwise leave as the default value of false.

heartbeat.send_events_to_tec
Set this to true if you want to send heartbeat events to the Tivoli Enterprise Console server; otherwise leave as the default value of false.

heartbeat.tec_server
If you have set heartbeat.send_events_to_tec to true, enter here the name of the Tivoli Enterprise Console server (for example EventServer or EventServer@regionname). The server must be a Tivoli-secure server.

heartbeat.trace_level
Specifies the level of the heartbeat engine trace (tmnt_hb_eng process). The minimum value is 0, the maximum is 2, the default is 1.

heartbeat.trace_size
Specifies the size in bytes of the heartbeat engine trace (tmnt_hb_eng process). The default is 500000.

request_manager.automatic_cancel_frequency
Specifies the time interval (in seconds) after which the Request Manager checks if the applications are using the requests that they have submitted. When an application is not using the requests, the Request Manager cancels the requests submitted by the application. The default value is 600 seconds.

request_manager.check_existance_in_ep_list
If set to false, the Request Manager does not check if the endpoints contained in the cache of the gateway are also contained in the cache of the Endpoint Manager at the framework level. The default is true.

request_manager.request_expiration_period
Specifies the number ("x") of periods allowed for an application to retrieve data. If an application does not receive data during "x" periods, then the Request Manager cancels the request for that application. The default value is 3 periods. This means that, for an application which submitted a request with a refresh time of 10 minutes, if the application is not getting any data for 10*3=30 minutes, then the request gets canceled.

request_manager.threads
Specifies the number of threads that the Request Manager uses to handle internal requests. It is approximately the number of endpoints that can be managed simultaneously. The value can be tuned depending on the workload of the managed node on which the Request Manager runs. The default value is 10.

request_manager.trace_level
Specifies the level of the request manager trace (tmnt_rm_eng process). The minimum value is 0, the maximum is 2, the default is 1.

request_manager.trace_size
Specifies the size in bytes of the request manager trace (tmnt_rm_eng process). The default is 500000.

taskengine.max_threads
Specifies the maximum number of threads for the task engine. The default is 10.
**task.trace_level**
Specifies the level of the task engine trace (tmnt_task_eng process). The minimum value is 0, the maximum is 2, the default is 1.

**task.trace_size**
Specifies the size in bytes of the task engine trace (tmnt_task_eng process). The default is 500000.

**tbsma.listener_timeout**
This parameter is set to define the maximum timeout to create a connection with the Tivoli Business Systems Manager CommonListener. The timeout is expressed in minutes: the minimum value is 1 minute, the default is 30 minutes.

**tbsma.jre_root**
This parameter is set during the installation of the Tivoli Business Systems Manager Adapter (see “Installing the Tivoli Business Systems Manager Adapter” on page 85) and you do not normally need to change it manually. However, if, for example, you want to install the adapter on a group of gateways using one instance of the install action/command, you will then need to change this parameter on any gateways in the group that have JRE installed at a location different to that supplied on the Install Options dialog.

Set this parameter to the complete path of the root directory of Java Runtime Environment, 1.3.0 (excluding the directory /bin).

**Note:** In a Windows NT Workstation, if the install target path contains a directory with spaces in the name, the directory name must be specified between single quotes, as in this example:
D:\'Program Files'\jre

**tbsma.trace_level**
Specifies the level of the Tivoli Business Manager engine traces (tmnt_tbsm_eng and tmnt_tbsm_wrapper processes). The minimum value is 0, the maximum is 2, the default is 1.

**tbsma.trace_size**
Specifies the size in bytes of the Tivoli Business Manager engine traces (tmnt_tbsm_eng and tmnt_tbsm_wrapper processes). The default is 500000.

**trace.filename**
Filename to which the trace messages from the adapter will be written. The default filename is DM.trc.

**transport.local.ip.address**
Specifies the IP address of the node running the Tivoli Business Systems Manager Adapter. The default value is the IP address of the node where the configuration file has been created.

**transport.mqe.usefiller**
Set this to true if the managed node/gateway on which the adapter is installed is running Windows NT, 4.0, Service Pack 5; otherwise leave as the default value of false.

**transport.request.address**
Specifies the address for requests. The default address is local_IP_address.DM.QM+DM.Q (for example, 146.84.112.165.DM.QM+DM.Q)
transport.response.address
Specifies the address for responses. The default address is local_IP_address.DM.QM+DM.Q (for example, 146.84.112.165.DM.QM+DM.Q)

transport.request.port
Specifies the port number used to receive requests. The default value is 6969.

transport.response.port
Specifies the port number user to send responses. The default value is 6969.

transport.mqe.fileregistry
Specifies the file registry for the message queue. The default is com.ibm.mqe.registry.MQeFileSession.

transport.mqe.maxchannels
Specifies the maximum number of channels for the message queue. The default is 1.

transport.mqe.local.queue.store
Specifies how the local message queue stores incoming data. The value can be either of the following (the default is file):

- file Indicates that the data is written to disk until received by the CommonListener
- memory Indicates that the data is kept in memory
- reduced Indicates that some data is kept in memory but, beyond a certain amount, the data is written to disk

transport.mqe.remote.queue.store
Specifies how the remote message queue stores incoming data. The value can be either of the following (the default is file):

- file Indicates that the data is written to disk until received by the CommonListener
- memory Indicates that the data is kept in memory
- reduced Indicates that some data is kept in memory but, beyond a certain amount, the data is written to disk

transport.server.ip.address
IP address or hostname of the CommonListener component of Tivoli Business Systems Manager, which listens out for messages from the systems being managed. For example: transport.server.ip.address=193.202.74.21

transport.server.mqe.address
Address of the CommonListener component. The default address is ServerQM+ServerQ.

transport.server.mqe.port
Port number of the CommonListener component. For example: transport.server.mqe.port=8082

transport.trace.enable
Set this to true if you want to store all messages regarding the transport of
adapter-acquired data to the CommonListener. The messages are stored in the file identified in trace.filename. The default is false

transport.trace.level
If you have enabled adapter trace messages, set this to low, medium or high, according to the level of details you require. The default is low

Note: No validation is performed on the values entered against a given key, so you must ensure yourself that they are valid.

--m {managed_node | @managed_nodes_file | all}
Specifies the managed nodes/gateways on which the product configuration will be updated. If this option is not used the product configuration is updated on the managed node/gateway from which the command is being run.

The options are as follows:

managed_node
The name of the managed node/gateway on which the command is to be performed.

@managed_nodes_file
A file containing managed node/gateway names separated by the CR/LF character; the @ symbol is an indicator that a file name follows but the symbol should not form part of the filename.

all
All managed nodes/gateways in the Tivoli region

--G key
Shows a configuration key. The argument is a key or a pattern for a group of keys. The wild-card ‘*’ is accepted in the pattern. When the wild-card is used, the output of the command shows also configuration keys (other than those documented under the -D option) whose value should not be modified. It is recommended that you do not change the value of such keys without the direction of Tivoli Support.

Authorization
Requires the super or senior roles.

Examples
1. The following example updates the configuration at all managed nodes/gateways identified in the file my_nodes1.txt with the ip address and port of the server where the Tivoli Business Systems Manager’s CommonListener is to be found
   wdmconfig --m @my_nodes1.txt -D transport.server.ip.address=193.202.74.21
   -D transport.server.mqe.port=4068
2. The following example is a query for all the request manager configuration parameters
   wdmconfig --G request_manager.*

See also
wdmdiscovery
wdmheartbeat
wdmmngcache
wdmdiscovery

Runs a bulk or delta discovery on selected endpoints at selected managed nodes/gateways. The command should only be used if the Tivoli Business Systems Manager Adapter component is installed.

Syntax

wdmdiscovery [-m {managed_node | @managed_nodes_file | all}] -b

[-e {endpoint | @endpoints_file} | -a]]

wdmdiscovery [-m {managed_node | @managed_nodes_file | all}] -d

[-e {endpoint | @endpoints_file} | -a]]

Description

The wdmdiscovery command operates on a selected managed node/gateway, a list of managed nodes/gateways contained in a file or all managed nodes/gateways, and sends a bulk or a delta discovery of an endpoint, a list of endpoints contained in a file or all endpoints, to the Tivoli Business Systems Manager CommonListener.

Options

-b
The discovery is to be bulk, in which all information on all endpoints identified by the -e parameter is returned.

-d
The discovery is to be delta, in which information is only returned on endpoints where a change of status has taken place since the previous discovery. The status in this case is the presence or absence of the endpoint in the network. Thus, the delta discovery reports only endpoints that have come alive or have been switched off or are for some other reason unavailable in the network.

[-e {endpoint | @endpoints_file} | -a]
Specifies the endpoints for which information is required. If this option is not used the discovery will be carried out on all endpoints attached to the managed node/gateway from which the command is being run.

The options are as follows:

-e endpoint
An endpoint label.

-e @endpoints_file
A file containing endpoint labels separated by the CR/LF character; the @ symbol is an indicator that a file name follows but the symbol should not form part of the filename.

-a
All endpoints attached to the defined managed nodes/gateways that gave an alive status at the most recent heartbeat monitoring.

-m {managed_node | @managed_nodes_file | all}
Specifies the managed nodes/gateways on which the discovery will be carried out. If this option is not used the discovery is carried out on the managed node/gateway from which the command is being run.

The options are as follows:
**managed_node**

The name of the managed node/gateway on which the command is to be performed.

@**managed_nodes_file**

A file containing managed node/gateway names separated by the CR/LF character; the @ symbol is an indicator that a file name follows but the symbol should not form part of the filename.

**all**

All managed nodes/gateways in the Tivoli region

**Authorization**

Requires the super, senior or admin roles.

**Examples**

1. The following example carries out a bulk discovery on all *alive* endpoints of the managed node/gateway **MyManagedNode**:

   `wdmdiscovery -m MyManagedNode -b -a`

2. The following example carries out a bulk discovery on endpoint **MyEndpoint** of the managed node/gateway **MyManagedNode**:

   `wdmdiscovery -m MyManagedNode -b -e MyEndpoint`

3. The following example carries out a delta discovery on all endpoints of the managed node/gateway **MyManagedNode**:

   `wdmdiscovery -m MyManagedNode -d`

**See also**

wdmconfig

wdmheartbeat

wdmmngcache
wdmdistrib

Distributes a profile to subscribers.

**Syntax**

```
wdmdistrib -p profile_name [-D MDist2_property=value ...] [-M maintain | over_all
| [-d] [-R] [-l] [-s subscribers_file] [subscriber...]
```

```
wdmdistrib -J JRE_location_dir -l [-r] [-D MDist2_property=value ...] [-f] [-n] [-e]
| [-w] [-i] [-d] [-R] [-s subscribers_file] [subscriber...]
```

**Description**

This command distributes a profile to subscribers. Three types of subscribers can be specified, in any of the following formats:

- `@resource:node_name`
- `@Endpoint:node_name`
- `@ProfileManager:node_name` or `/Regions/PolicyRegionName/profile_manager_name`

where `resource` can be Endpoint, ProfileManager, or the application resource type, and where `node_name` is the name of the endpoint, profile manager, or application resource.

Endpoint is the default if no subscriber’s resource type is specified.

If no subscribers are specified then the profile is distributed to all current subscribers of the profile manager to which the profile belongs.

The `-M` option is used to identify the rules to follow when the distribution is for subscribers that already have local copies of the profile.

Tivoli Monitoring uses Multiplexed Distribution (MDist 2) to perform asynchronous profile data transfers. For details on MDist 2, see [Profile distribution using MDist 2" on page 59](#).

**Options**

- `-p profile_name`
  
  Specifies the name of the profile that is to be distributed. The profile name can be fully qualified with the policy region name. If no region is indicated, the local region is assumed.

- `-R`
  
  Specifies that the profile is distributed recursively to all level of subscribers. The default value is `false`, meaning that the profile is delivered only to the next level of subscribers.

- `-M maintain | over_all | over_opts | over_all_no_merge`
  
  If specified, it defines the rule to follow when the distribution is for subscribers that already have local copies of the profile. The `maintain` option keeps the local modifications, `over_all` overwrites the local modifications, `over_opts` merges and distributes all records, `over_all_no_merge` distributes only the specified profile. If `-M` is not specified, the default is `maintain`.

- `-l`
  
  If specified, it allows the distribution to be targeted also to resources which are
not subscribed to the profile manager to which the profile belongs. This option is ignored when no subscribers are specified. Option -J must be specified in conjunction with -l if no profile name is indicated.

-d
   If specified, then the data is disposable, meaning that it can be removed from the repeater depot closer to the endpoint after distribution.

-f
   Specifies that the results should be written to the log file. If neither -n nor -f are specified, then the default is to write the results only to the log file.

-n
   Specifies that the results should be written to the Tivoli Monitoring notice group. If neither -n nor -f are specified, then the default is to write the results only to the log file.

-e
   If specified, then error messages should be written to the log file.

-w
   If specified, then warning messages should be written to the log file.

-i
   If specified, then informational messages should be written to the log file.

-D MDist2_property=value
   Sets one or more properties to configure MDist 2 for the current distribution; a separate -D must be specified for each property to be configured. Only the following MDist 2 properties are supported:

   label   Specifies a description string for the distribution. The default value is the string profile_name(operation), where operation is install.

   pri     Specifies the priority level, which is the order in which distributions are handled by repeaters: h (highest priority), m (medium priority), or l (low priority). The default value is m (medium priority).

   send_timeout
   Specifies the length of time in seconds a repeater will wait for a target system to receive a block of data. This timeout is used to detect network or endpoint failures. The default value is 3600 seconds. This attribute is initially set using the wmdist -s command (the default is 300 seconds). You can override the wmdist -s setting by specifying a different value here.

   result_timeout
   Specifies the length of time in seconds a repeater will wait for the distribution method at the endpoint to return the distribution results.

   deadline
   The date on which a distribution expires, that is, when it fails for unavailable target systems. It is specified in the format "mm/dd/yyyy hh:mm". If this property is not specified, the following default values are assumed: 60 minutes (from the time the distribution is started) for profile installations, 120 minutes (from the time the distribution is started) for JRE distributions.

   -J "JRE_location_dir"
   The complete path of the Tools\JRE directory where Java Runtime Environment (JRE) is located. The path can be included within double quotes. The directory from which the JRE is distributed must be on the local filesystem of the Tivoli
server (not the managed node) and must have the JRE for each operating
system (like in the Tools\JRE directory of the product CD).

–r Indicates that the JRE must be replaced on the target subscribers. This option
has effect only if used in conjunction with the –J option. When –r is not used,
the JRE is distributed only to subscribers that do not have the JRE already
installed or linked.

–s subscribers_file
A file containing the profile’s subscriber names, separated by the CR/LF
character.

subscriber
The name of any subscriber to which the profile is to be distributed. See the
command description for the possible formats. If more than one subscriber is
specified, separate the names by blanks. If no subscribers are specified then the
profile is distributed to all current subscribers of the profile manager to which
the profile belongs. This option must be specified as the last one in the
command syntax.

Authorization
Requires the admin, super, or senior role.

Examples
1. The following example distributes the profile myProfile to the subscriber called
ApacheWebserver1@my_endpoint (whose resource type is ApacheWebServer):

   wmdistrib –p myProfile @ApacheWebserver1@my_endpoint

2. The following example distributes the profile myProfile to the endpoint
myEndpoint. The JRE from directory \Tools\Jre of drive E will be installed at
the endpoint if not present:

   wmdistrib –p myProfile -J "E:\Tools\Jre" myEndpoint

3. The following example distributes the profile myProfile, recursively, to the
subscribers listed in the subscribersFilename file in the directory myDir on
drive D and also to the profile manager myProfileManager:

   wmdistrib –p myProfile -R -l -s "D:\myDir\subscribersFilename"
   @ProfileManager:myProfileManager

See also

wdmconfig

wmdist (see the Tivoli Management Framework Reference manual)

wdepot (see the Tivoli Management Framework Reference manual)
**wdmdumpprf**

Writes the full details of a profile to the standard output.

**Syntax**

```
wdmdumpprf -P profile#region [-x]
```

**Description**

The `wdmdumpprf` command writes the full details of the selected profile to the standard output, in Tivoli Management Framework format or XML format. The output can then be saved and edited and reloaded as a new or amended profile using the `wdmloadprf` command.

An example output from this command of a profile containing a single resource model (in XML format), is as follows:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Tmw2kProfile [
<!ELEMENT Tmw2kProfile (GenProfileInfo, ResModelInfo*)>
<!ATTLIST Tmw2kProfile Version CDATA #REQUIRED>
<!ELEMENT GenProfileInfo (ResModelInfo*)>
<!ATTLIST GenProfileInfo ProfileName CDATA #REQUIRED>
<!ATTLIST GenProfileInfo EnableTBSMfeeding CDATA #REQUIRED>
<!ATTLIST GenProfileInfo Tec CDATA #REQUIRED>
<!ATTLIST GenProfileInfo TecEventServer CDATA #REQUIRED>
<!ELEMENT ResModelInfo (SchedulingInfo,LoggingInfo,Property*,Parameter*,EventAggregInfo*)>
<!ATTLIST ResModelInfo Enabled CDATA #REQUIRED>
<!ATTLIST ResModelInfo ModelName CDATA #REQUIRED>
<!ATTLIST ResModelInfo Zipfile CDATA #REQUIRED>
<!ATTLIST ResModelInfo Platform CDATA #REQUIRED>
<!ATTLIST ResModelInfo MinorVersion CDATA #REQUIRED>
<!ATTLIST ResModelInfo MajorVersion CDATA #REQUIRED>
<!ATTLIST ResModelInfo CycleTime CDATA #REQUIRED>
<!ELEMENT SchedulingInfo (String)*>
<!ATTLIST SchedulingInfo StartDate CDATA #REQUIRED>
<!ATTLIST SchedulingInfo StopDate CDATA #REQUIRED>
<!ELEMENT String (#PCDATA)>
<!ELEMENT LoggingInfo (DataLoggingInfo)>
<!ELEMENT DataLoggingInfo (String)>
<!ATTLIST DataLoggingInfo EnableLogging CDATA #REQUIRED>
<!ATTLIST DataLoggingInfo TEDW CDATA #REQUIRED>
<!ATTLIST DataLoggingInfo RAW CDATA #REQUIRED>
<!ATTLIST DataLoggingInfo LoggingPeriod CDATA #REQUIRED>
<!ATTLIST DataLoggingInfo AggregateData CDATA #REQUIRED>
<!ATTLIST DataLoggingInfo AggregationPeriod CDATA #REQUIRED>
<!ATTLIST DataLoggingInfo WantMin CDATA #REQUIRED>
<!ATTLIST DataLoggingInfo WantMax CDATA #REQUIRED>
<!ATTLIST DataLoggingInfo WantAvg CDATA #REQUIRED>
<!ELEMENT Property (#PCDATA)>
<!ATTLIST Property Name CDATA #REQUIRED>
<!ATTLIST Property Value CDATA #REQUIRED>
<!ELEMENT Parameter (String)>]
<!ATTLIST Parameter Name CDATA #REQUIRED>
<!ATTLIST Parameter Type CDATA #REQUIRED>
<!ELEMENT EventAggregInfo (Action*,Task*)>
<!ATTLIST EventAggregInfo Name CDATA #REQUIRED>
<!ATTLIST EventAggregInfo Occurrences CDATA #REQUIRED>
<!ATTLIST EventAggregInfo Holes CDATA #REQUIRED>
<!ATTLIST EventAggregInfo SendToTBSM CDATA #REQUIRED>
<!ATTLIST EventAggregInfo Severity CDATA #REQUIRED>
```
Chapter 8. Commands 133

<Tmw2kProfile Version = "51106">
<GenProfileInfo
ProfileName = "prfa#user1-region"
EnableTBSMfeeding = "FALSE"
Tec = "None"
TecEventServer = ""/>
<ResModelInfo
ModelName = "TMW_MemoryModel"
Enabled = "TRUE"
Zipfile = "TMW_MemoryModel.zip"
Platform = "w32-ix86"
MajorVersion = "1"
MinorVersion = "0"
CycleTime = "60">
<SchedulingInfo
StartDate = "0"
StopDate = "0">
</SchedulingInfo>
</ResModelInfo>
<LoggingInfo>
<DataLoggingInfo
EnableLogging = "FALSE"
TEDW = "FALSE"
RAW = "TRUE"
LoggingPeriod = "720"
AggregateData = "FALSE"
AggregationPeriod = "15"
WantMin = "FALSE"
WantMax = "FALSE"
WantAvg = "TRUE">
</DataLoggingInfo>
</LoggingInfo>
<Property
Name = "ExcessivePageFaults"
Value = "350,000000">
</Property>
<Property
Name = "ExcessivePaging"
Value = "60,000000">
</Property>
<Property
Name = "LowCacheHitsPercent"
Value = "70,000000">
</Property>
<Property
Name = "MinimumAvail"
Value = "10485760,000000">
</Property>
<Property
Name = "MinimumCommitted"
Value = "5242880,000000">
</Property>
<EventAggregInfo
Name = "TMW_MemoryLeakInPB"
Occurrences = "40"
Holes = "1"
SendToTBSM = "FALSE"
Severity = "CRITICAL"
Clearing = "YES"

SendTec = "TRUE"
ExecuteAction = "FALSE">
</EventAggregInfo>
<EventAggregInfo
Name = "TMW_LowAvailWithSmallPageFile"
Occurrences = "1"
Holes = "0"
SendToTBSM = "FALSE"
Severity = "WARNING"
Clearing = "NO"
SendTec = "TRUE"
ExecuteAction = "FALSE">
</EventAggregInfo>
<EventAggregInfo
Name = "TMW_LowPinReadHits"
Occurrences = "10"
Holes = "1"
SendToTBSM = "FALSE"
Severity = "MINOR"
Clearing = "NO"
SendTec = "FALSE"
ExecuteAction = "FALSE">
</EventAggregInfo>
<EventAggregInfo
Name = "TMW_LowMDLReadHits"
Occurrences = "10"
Holes = "1"
SendToTBSM = "FALSE"
Severity = "MINOR"
Clearing = "NO"
SendTec = "FALSE"
ExecuteAction = "FALSE">
</EventAggregInfo>
<EventAggregInfo
Name = "TMW_LowAvailCausingSoftPaging"
Occurrences = "10"
Holes = "1"
SendToTBSM = "FALSE"
Severity = "WARNING"
Clearing = "NO"
SendTec = "TRUE"
ExecuteAction = "FALSE">
</EventAggregInfo>
<EventAggregInfo
Name = "TMW_LowAvail"
Occurrences = "6"
Holes = "2"
SendToTBSM = "FALSE"
Severity = "WARNING"
Clearing = "NO"
SendTec = "TRUE"
ExecuteAction = "FALSE">
</EventAggregInfo>
<EventAggregInfo
Name = "TMW_LowDataMapHits"
Occurrences = "10"
Holes = "1"
SendToTBSM = "FALSE"
Severity = "MINOR"
Clearing = "NO"
SendTec = "FALSE"
ExecuteAction = "FALSE">
</EventAggregInfo>
<EventAggregInfo
Name = "TMW_LowAvailCausingHardPaging"
Occurrences = "10"
Holes = "1"
SendToTBSM = "FALSE"
Severity = "WARNING"
Clearing = "NO"
SendTec = "TRUE"
ExecuteAction = "FALSE">
</EventAggregInfo>
<EventAggregInfo
Name = "TMW_LowAvailCausingSoftPagefileResize"
Occurrences = "1"
Holes = "0"
SendToTBSM = "FALSE"
Severity = "WARNING"
Clearing = "NO"
SendTec = "TRUE"
ExecuteAction = "FALSE">
</EventAggregInfo>
<EventAggregInfo
Name = "TMW_LowAvailHighCache"
Occurrences = "6"
Holes = "1"
SendToTBSM = "FALSE"
Severity = "WARNING"
Clearing = "NO"
SendTec = "TRUE"
ExecuteAction = "FALSE">
</EventAggregInfo>
<EventAggregInfo
Name = "TMW_PageFileResizing"
Occurrences = "1"
Holes = "0"
SendToTBSM = "FALSE"
Severity = "WARNING"
Clearing = "NO"
SendTec = "TRUE"
ExecuteAction = "FALSE">
</EventAggregInfo>
<EventAggregInfo
Name = "TMW_MemoryLeakInSC"
Occurrences = "40"
Holes = "10"
SendToTBSM = "FALSE"
Severity = "CRITICAL"
Clearing = "YES"
SendTec = "TRUE"
ExecuteAction = "FALSE">
</EventAggregInfo>
<EventAggregInfo
    Name = "TMW_LowAvailCausingManyProblems"
    Occurrences = "1"
    Holes = "0"
    SendToTBSM = "FALSE"
    Severity = "CRITICAL"
    Clearing = "YES"
SendTec = "TRUE"
ExecuteAction = "FALSE">
</EventAggregInfo>
<EventAggregInfo
    Name = "TMW_HighPaging"
    Occurrences = "15"
    Holes = "1"
    SendToTBSM = "FALSE"
    Severity = "WARNING"
    Clearing = "NO"
SendTec = "TRUE"
ExecuteAction = "FALSE">
</EventAggregInfo>
<EventAggregInfo
    Name = "TMW_MemoryLeakInSD"
    Occurrences = "40"
    Holes = "10"
    SendToTBSM = "FALSE"
    Severity = "CRITICAL"
    Clearing = "YES"
SendTec = "TRUE"
ExecuteAction = "FALSE">
</EventAggregInfo>
<EventAggregInfo
    Name = "TMW_LowCopyReadHits"
    Occurrences = "10"
    Holes = "1"
    SendToTBSM = "FALSE"
    Severity = "MINOR"
    Clearing = "NO"
SendTec = "FALSE"
ExecuteAction = "FALSE">
</EventAggregInfo>
</ResModelInfo>
</Tmw2kProfile>

Options

-P profile#region
 Defines the profile to be written to the standard output. The profile name
must always be fully qualified with the policy region name in the syntax
shown.

Note: To obtain a listing of the complete names of all the profiles that are
on an endpoint, issue the following command:

wdmiseng -e endpoint
Dumps the profile in XML format. The default format is Tivoli Management Framework.

**Authorization**
Requires the super, senior, admin, or user roles.

**Examples**
The following example writes the full details of the profile `MyProfile` in the region `MyRegion` to the standard output.

```plaintext
wdmdumpprf -P MyProfile#MyRegion
```

**See also**

- `wdmeditprf`
- `wdmloadprf`
wdmeditprf

Allows you to edit the definition of a profile, including all resource model details.

**Syntax**

```
wdmeditprf -P profile#region { profile_options | resource_model_actions }
```

where the `profile_options` syntax is:

```
[ -list | -Tec { broadcast -S server_list | failover -S server_list | no } ] [ -TBSM { yes | no } ]
```

the `resource_model_actions` syntax is:

```
–remove resource_model | –print resource_model [ -t ] [ -e ] [ -tasks ] [ -Log ] [ -c ]
[ -par ] [ -schedule ] [ -Tec ] [ -TBSM ] [ -add resource_model [ resource_model_addSpecifier ] ]
[ -edit resource_model resource_model_editSpecifier ]
```

the `resource_model_addSpecifier` syntax is:

```
[ -enable | -disable ] [ -c cycle_time ] [ -Schedule [ -start date ] [ -stop date ] [ -Rule
name day1:day2:... ] [ -interval hour:minute hour:minute ] ] [ -t threshold_name value ]...
[ -e event [ -o occurrences ] [ -h holes ] [ -SendTec | -NoSendTec | -SendTBSM | -NoSendTBSM ] [ -severity severity_type ] [ -clearing { yes | no } ] ] ] ] ] ]... [ -AddPar parameter_name parameter ]... [ -SetAction event action_name
action_retry_value ]... [ -DelAction event action_name ]... [ -AddTask event task_lib
task_name [ -a task_arg ... ] ]... [ -Log [ -LogEnable | -LogDisable | [ -p period | [ [ -RAW { yes | no } ] [ -TEDW { yes | no } ] ] [ -Agg { yes | no } ] ] ]... [ -ap
aggregation_period ] [ -Min | -Min ] [ -Max | -Max ] [ -Avg | -Avg ] ]
```

and the `resource_model_editSpecifier` syntax is:

```
[ -enable | -disable ] [ -c cycle_time ] [ -Schedule [ -start date ] [ -stop date ] [ -Rule
name day1:day2:... ] [ -interval hour:minute hour:minute ] ] [ -t threshold_name value ]...
[ -e event [ -o occurrences ] [ -h holes ] [ -SendTec | -NoSendTec | -SendTBSM | -NoSendTBSM ] [ -severity severity_type ] [ -clearing { yes | no } ] ] ] ] ] ]... [ -AddPar parameter_name parameter ]... [ -DelPar parameter_name parameter ]... [ -AddAction event action_name action_retry_value ]... [ -SetAction event action_name
action_retry_value ]... [ -DelAction event action_name ]... [ -AddTask event task_lib
task_name [ -a task_arg ... ] ]... [ -DelTask event task_lib task_name ]... [ -Log [ -LogEnable | -LogDisable | [ -p period | [ [ -RAW { yes | no } ] [ -TEDW { yes | no } ] ] [ -Agg { yes | no } ] ] ]... [ -ap aggregation_period ] [ -Min | -Min ] [ -Max | -Max ]
[ -NoMax ] [ -Avg | -Avg ] ]
```

**Description**

The `wdmeditprf` command lets you change various attributes of a profile:

- Resource models can be added with default values or with value supplied by you
- Resource model details can be edited
- Profiles or selected resource model details can be printed
- The destination monitors for events generated by the resource model can be defined
- The tasks triggered by events, generated by the resource model, can be defined.
Resource models can be added using all of the default values supplied, which are documented in *IBM Tivoli Monitoring: Resource Model Reference Guide*. Alternatively you can add a model with one or more values modified to suit your circumstances. You can also edit any of the details of an existing resource model. The various options shown in the *Syntax* section, above, and defined in the *Options* section, below, are non-exclusive; thus you could issue a single command to add a resource model modifying the default values of, for example, the cycle time and an event. You could then issue a separate command to edit, for example, the logging details and two thresholds.

**Options**

The options to the **wdmeditprf** command must be specified in the correct sequence:

1. `–P profile#region`
2. `–add, –edit, –list, –print, –remove, –TBSM or –Tec`
3. `resource_model` (if appropriate)
4. Arguments to the `–add` or `–edit` options: these can be entered in any order

The option details are as follows:

`–P profile#region`

The profile on which the actions of this command are to be performed.

`–add resource_model`

Adds the named resource model to the profile. Use the resource model internal name, as specified for each resource model in the *IBM Tivoli Monitoring: Resource Model Reference Guide*.

The command may optionally be followed by one or more of the resource model arguments, that will modify the default values of the named resource model, only the values that need changing need to be specified. Details of these arguments are given following the `–Tec` option.

`–edit resource_model`

Specifies that you want to edit the named resource model. Use the resource model internal name, as specified for each resource model in the *IBM Tivoli Monitoring: Resource Model Reference Guide*.

The command may optionally be followed by one or more of the resource model arguments, that will modify the default values of the named resource model, only the values that need changing need to be specified. Details of these arguments are given following the `–Tec` option.

`–list`

Lists a summary of the profile details to the standard output. The output format is as follows:

<table>
<thead>
<tr>
<th>Resource Model</th>
<th>Enable</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMXFileSystem</td>
<td>YES</td>
</tr>
<tr>
<td>DMXMemory</td>
<td>YES</td>
</tr>
<tr>
<td>DMXProcess</td>
<td>YES</td>
</tr>
<tr>
<td>TMW_ParamEventLog</td>
<td>NO</td>
</tr>
</tbody>
</table>

`–print resource_model`

Prints basic information about the named resource model. Use the resource model internal name, as specified for each resource model in the *IBM Tivoli Monitoring: Resource Model Reference Guide*. Additional information can be obtained by specifying (at least) one or more of the following arguments:

`–c` For printing cycle time information.
For printing event information.

-`tasks` For printing information associated with tasks.

-`Log` For printing data logging information.

-`par` For printing parameter information (for parametric resource models).

-`schedule` For printing schedule information.

-`t` For printing threshold information.

-`Tec` For printing information related to the Tivoli Enterprise Console.

-`TBSM` For printing information related to the Tivoli Business System Manager.

-`remove resource_model` Removes the named resource model from the profile. Use the resource model internal name, as specified for each resource model in the *IBM Tivoli Monitoring: Resource Model Reference Guide*.

-`TBSM` Specifies whether events for this profile are to be sent to the Tivoli Business Systems Manager. The options are:

  - `yes` Specifies that events will be sent to the Tivoli Business Systems Manager.

  - `no` Specifies that no events will be sent to the Tivoli Business Systems Manager. If this option is selected it will ignore any requests to “Send to TBSM” that you specify or may have specified for individual indications.

-`Tec` Specifies whether events for this profile are to be sent to the Tivoli Enterprise Console server, and identifies the server. The options are:

  - `broadcast` `–S server_list` Specifies that the events will be sent to all Tivoli Enterprise Console servers in the list. To indicate multiple servers, use a comma (`,`) as a separator. If you specify a server with no region name, the events will be sent to the local Tivoli Enterprise Console server. If the server is an unsecure server, specify the server location followed by the plus `+` sign and the server port.

  - `failover` `–S server_list` Specifies that the events will be sent to the first available unsecure Tivoli Enterprise Console server in the list. Servers are specified by the server location followed by the plus `+` sign and the server port.

  - `no` Specifies that no events will be sent to the Tivoli Enterprise Console server. If this option is selected it will ignore any requests to “Send to TEC” that you specify or may have specified for individual indications.

**Arguments to the –add and –edit options**

The `–add` and `–edit` options can take the following arguments, in any order, and according to the specifications provided with the `resource_model_add_specifier` syntax and `resource_model_edit_specifier` syntax:

-`AddPar parameter_name parameter_value` Defines the values of the parameters for parametric resource models. Each parameter is of one of four types. Details of the parameter names
and types for each parametric resource model can be found in the IBM Tivoli Monitoring: Resource Model Reference Guide. The parameter values are added using the `–AddPar` option according to the parameter type, as follows:

- **Numeric:**
  A list of numeric values. The `–AddPar` option appends the value supplied in `parameter_value` to the existing values in the numeric parameter.

- **String:**
  A list of strings. The `–AddPar` option appends the string supplied in `parameter_value` to the existing values in the string parameter.

- **Boolean list:**
  A boolean list of pre-defined non-exclusive values. The `–AddPar` option switches the boolean list value supplied in `parameter_value` to the `true` state. To switch it to the `false` state use the `–DelPar` option.

- **Choice list:**
  A choice list of pre-defined exclusive alternatives. The `–AddPar` option selects the choice list value supplied in `parameter_value`; the previously selected value will automatically be deselected when the command is performed.

```
–AddAction event action_name action_retry_value
  Adds an action with the specified values to the resource model:
  • `event`
    The event which triggers the action.
  • `action_name`
    The name of the built-in action.
  • `action_retry_value`
    The retry value for the action.
  
  Note: Built-in actions can be added only if they were previously removed from the resource model.

–SetAction event action_name action_retry_value
  Defines an action with the specified values in the resource model:
  • `event`
    The event which triggers the action.
  • `action_name`
    The name of the built-in action.
  • `action_retry_value`
    The retry value for the action.

–DelAction event action_name
  Removes an action with the specified values from the resource model:
  • `event`
    The event which triggers the action.
  • `action_name`
    The name of the built-in action.

–AddTask event task_library task_name [ –a task_arg ... ]
  Defines the values associated to a task:
  • `event`
    The event which triggers the task.
  • `task_library`
The task library for the specified task.

- **task_name**
  The name of the task.

- **task_arg**
  Any task arguments.

--cycle_time
Changes the default value of the cycle time of a resource model. The value is supplied as a number of seconds.

--DelPar parameter_name parameter_value
Changes the values of the parameters for parametric resource models. Each parameter is of one of four types. Details of the parameter names and types for each parametric resource model can be found in the *IBM Tivoli Monitoring: Resource Model Reference Guide*. The parameter values are changed using the --DelPar option according to the parameter type, as follows:

- **Numeric:**
  A list of numeric values. The --DelPar option deletes the value supplied in `parameter_value` from the existing values in the numeric parameter.

- **String:**
  A list of strings. The --DelPar option deletes the string supplied in `parameter_value` from the existing values in the string parameter.

- **Boolean list:**
  A boolean list of pre-defined non-exclusive values. The --DelPar option switches the boolean list value supplied in `parameter_value` to the false state. To switch it to the true state use the --AddPar option.

- **Choice list:**
  This type of parameter cannot be modified with the --DelPar option; use the --AddPar instead.

--DelTask event task_library task_name
Deletes a task with the specified values from the resource model:

- **event**
  The event which triggers the task.

- **task_library**
  The task library for the specified task.

- **task_name**
  The name of the task.

--disable
Disables the resource model for the defined profile.

--e event
Changes the default value for a named event. Any number of events can be defined. The specific values that can be changed are the following:

- **–h holes:**
  Defines the number of consecutive holes (cycles where an indication does not happen) that can interrupt the count of consecutive occurrences without zeroing that count.

- **–o occurrences**
  Defines the number of consecutive occurrences (cycles where an indication happens) that are needed to trigger the event.

- **–NoSendTBSM**
Defines that the event is not to be sent to Tivoli Business Systems Manager.

- **–NoSendTec**
  The event is not to be sent to the Tivoli Enterprise Console server.

- **–SendTBSM**
  Defines that the event is to be sent to Tivoli Business Systems Manager.

- **–SendTec**
  Defines that the event is to be sent to the Tivoli Enterprise Console server.

- **–severity severity_type**
  The severity type of the event must be one of the following: FATAL, CRITICAL, WARNING, HARMLESS, MINOR.

- **–clearing**
  Specifies whether a clearing event is to be sent when the circumstances that generated the event are no longer present. The options are:
  - yes  A clearing event is sent.
  - no   A clearing event is not sent.

- **–enable**
  Enables the resource model for the defined profile.

- **–Log**
  Changes the default values for the data logging details. The specific values that can be changed are the following:
  - **–Agg** (yes | no)
    Determines whether logging data will be aggregated (yes) or not (no).
  - **–ap** aggregation_period
    Defines the period for which data is aggregated, in the format HH:MM; maximum 24:00 hours. The allowed period is in the range: 00:05 - 24:00. The minimum aggregation unit is 5 minutes.
  - **–Avg**
    Defines that the average of the values encountered during the aggregation period will be logged.
  - **–LogDisable**
    Defines that the data logging feature is to be disabled.
  - **–LogEnable**
    Defines that the data logging feature is to be enabled.
  - **–Max**
    Defines that the highest value encountered during the aggregation period will be logged.
  - **–Min**
    Defines that the lowest value encountered during the aggregation period will be logged.
  - **–NoAvg**
    Defines that the average of the values encountered during the aggregation period will not be logged (see option -Avg)
  - **–NoMax**
    Defines that the highest value encountered during the aggregation period will not be logged (see option -Max)
  - **–NoMin**
Defines that the lowest value encountered during the aggregation period will not be logged (see option -Min)

- \( v \) \( -p \) \( \text{period} \)
  Defines the historical duration of data in the data logging database, in the format HH:MM; maximum 24:00 hours. The allowed period is in the range: 00:05 - 24:00. The minimum aggregation unit is 5 minutes.

- \( v \) \( -\text{RAW} \) \{yes | no\}
  Determines whether the logging of raw data is enabled (yes) or disabled (no).

- \( v \) \( -\text{TEDW} \) \{yes | no\}
  Determines whether the logging of TEDW data is enabled (yes) or disabled (no).

- \( t \) \text{threshold threshold_value}
  Defines the value for a named threshold; any number of thresholds can be defined.

- \( \text{Schedule} \ [ \ -\text{start date} \] \ -\text{stop date} \)
  Defines the schedule for the resource model. The start and stop dates are in the format year-month-day (for example: to indicate July 16 2003, specify 2003-7-16). The default start date is today. At least one schedule rule must be specified in the format:

- \( \text{Rule name day1:day2:...} \[ \ -\text{interval hour:minute hour:minute} \]

**Authorization**

Requires the super, senior or admin roles.

**Examples**

1. The following example adds the resource model \textbf{MyResourceModel} to the profile \textbf{MyProfile} in the region \textbf{MyRegion}, using all of the default values:
   
   \texttt{wdmeditprf -P MyProfile#MyRegion -add MyResourceModel}

2. The following example adds the resource model \textbf{MyResourceModel} to the profile \textbf{MyProfile} in the region \textbf{MyRegion}, changing a selection of the defaults (the changed values are shown on separate lines for clarity, but would normally be concatenated in a single string):

   \texttt{wdmeditprf -P MyProfile#MyRegion -add MyResourceModel}
   
   -t MyThreshold 60
   -e MyEvent1 -o 5 -h 2 -SendTec -severity WARNING -NoSendTBSM
   -Log -LogEnable -Agg yes -ap 00:30 -Min -NoMax -Avg -p 12:00
   -c 120
   -enable

3. The following example edits the parametric resource model \textbf{MyParamEventLog} in the profile \textbf{MyProfile} in the region \textbf{MyRegion}, changing a selection of the defaults:

   \texttt{wdmeditprf -P MyProfile#MyRegion -edit MyParamEventLog}
   
   -AddPar Eids 2034 -AddPar Source Win2K -DelPar Source WinNT
   -AddPar Severity warning -DelPar Severity Information
   -AddPar FilterType and

4. The following example lists a summary of the profile \textbf{MyProfile} in the region \textbf{MyRegion} on the standard output:
5. The following example prints the event information about the resource model `MyResourceModel` in the profile `MyProfile` in the region `MyRegion`:

   ```
   wmeditprf -P MyProfile#MyRegion -list
   ```

6. The following example prints information about the resource model `MyResourceModel` in the profile `MyProfile` in the region `MyRegion`, including information on the cycle time, the events, the data logging and the parameters:

   ```
   wmeditprf -P MyProfile#MyRegion -print MyResourceModel -c -e -Log -par
   ```

7. The following example removes the resource model `MyResourceModel` from the profile `MyProfile` in the region `MyRegion`:

   ```
   wmeditprf -P MyProfile#MyRegion -remove MyResourceModel
   ```

8. The following example enables the sending of events to Tivoli Business Systems Manager for the profile `MyProfile` in the region `MyRegion`:

   ```
   wmeditprf -P MyProfile#MyRegion -TBSM yes
   ```

9. The following example disables the sending of events to Tivoli Business Systems Manager for the profile `MyProfile` in the region `MyRegion`:

   ```
   wmeditprf -P MyProfile#MyRegion -TBSM no
   ```

10. The following example defines the secure Tivoli Enterprise Console server `EventServer` for the profile `MyProfile` in the region `MyRegion`:

    ```
    wmeditprf -P MyProfile#MyRegion -Tec secure -S EventServer
    ```

11. The following example defines the insecure Tivoli Enterprise Console server `tecinserv` at port `8080` for the profile `MyProfile` in the region `MyRegion`:

    ```
    wmeditprf -P MyProfile#MyRegion -Tec insecure
    -S tecinserv.rome.tivoli.com -p 8080
    ```

12. The following example disables the sending of events to the Tivoli Enterprise Console server for the profile `MyProfile` in the region `MyRegion`:

    ```
    wmeditprf -P MyProfile#MyRegion -Tec no
    ```

See also

- `wdmdumpprf`
- `wdmloadprf`
**wdmeng**

Stops or starts profiles or resource models at endpoints; also deletes profiles at endpoints.

**Syntax**

```
wdmeng -e endpoint [-p profile#region] resource_model -start
wdmeng -e endpoint [-p profile#region] resource_model -stop
wdmeng -e endpoint -p profile#region -start
wdmeng -e endpoint -p profile#region -stop
wdmeng -e endpoint -p profile#region -delete
```

**Description**

The **wdmeng** command allows you to stop and start the Tivoli Monitoring engine for a specific profile or resource model at a specified endpoint; it also allows you to delete a profile at a specific endpoint.

**Options**

- **-e endpoint**
  Defines the endpoint on which the command is to be performed; only one endpoint can be specified.

- **-delete**
  Deletes the named profile.

  **Note:** To delete a resource model at an endpoint, you should disable the resource model in the profile at the Tivoli server using the **wdmeditprf** command and distribute the new profile to the endpoint, using this command to delete the old profile.

- **-p profile#region**
  Defines the profile on which the command’s actions will be carried out. If the profile is not specified, the command is performed on all profiles at the named endpoint. The profile name must always be fully qualified with the policy region name in the syntax shown.

  **Note:** To obtain a listing of the complete names of all the profiles that are on an endpoint, issue the following command:

  ```
  wdm1seng -e endpoint
  ```

- **resource model**
  Defines the resource model to be started or stopped. Use the resource model internal name, as specified for each resource model in the *IBM Tivoli Monitoring: Resource Model Reference Guide*.

- **-start**
  Starts the named resource model or profile.

- **-stop**
  Stops the named resource model or profile.
Authorization

Requires the super, senior or admin roles.

Examples

1. The following example starts the resource model MyResourceModel on the endpoint MyEndpoint in all profiles.
   
   wdmeng -e MyEndpoint MyResourceModel -start

2. The following example stops the resource model MyResourceModel on the endpoint MyEndpoint in all profiles.
   
   wdmeng -e MyEndpoint MyResourceModel -stop

3. The following example starts the resource model MyResourceModel in the profile MyProfile in policy region MyRegion on the endpoint MyEndpoint.
   
   wdmeng -e MyEndpoint -p MyProfile#MyRegion MyResourceModel -start

4. The following example stops the resource model MyResourceModel in the profile MyProfile in policy region MyRegion on the endpoint MyEndpoint.
   
   wdmeng -e MyEndpoint -p MyProfile#MyRegion MyResourceModel -stop

5. The following example starts the profile MyProfile in policy region MyRegion on the endpoint MyEndpoint.
   
   wdmeng -e MyEndpoint -p MyProfile#MyRegion -start

6. The following example stops the profile MyProfile in policy region MyRegion on the endpoint MyEndpoint.
   
   wdmeng -e MyEndpoint -p MyProfile#MyRegion -stop

7. The following example deletes the profile MyProfile in policy region MyRegion on the endpoint MyEndpoint.
   
   wdmeng -e MyEndpoint -p MyProfile#MyRegion -delete

See also

wdmcmd
wdmlseng
wdmtrceng
wdmheartbeat

Stops or starts the ‘heartbeat’ monitoring of endpoints, changes its frequency and queries the status of the heartbeat processor.

Syntax

wdmheartbeat [–m {managed_node | @managed_nodes_file | all}] [–s frequency | –t | –q ]

Description

The wdmheartbeat command operates on selected managed nodes/gateways and starts or stops the ‘heartbeat’, which is a periodic monitoring at a managed node/gateway of all attached endpoints to determine certain basic status information. In addition, this command can also be used to change the frequency of the heartbeat, in other words the period elapsing between two heartbeat requests.

Options

–m {managed_node | @managed_nodes_file | all}
   Specifies the managed nodes/gateways on which the heartbeat command is to be implemented. If this option is not used the heartbeat command is implemented on the managed node/gateway from which the command is being run.

   The options are as follows:

   managed_node
   The name of the managed node/gateway on which the command is to be performed.

   @managed_nodes_file
   A file containing managed node/gateway names separated by the CR/LF character; the @ symbol is an indicator that a file name follows but the symbol should not form part of the filename.

   all
   All managed nodes/gateways in the Tivoli region

–q
   Queries the status of the heartbeat processor. An example of the output is as follows:

   Processing ManagedNode mcrudele...
   HeartBeat processor status: STARTED, frequency: 60
   Processing ManagedNode dmw2k2...
   HeartBeat processor status: STARTED, frequency: 60

–s frequency
   This option starts the heartbeat on the defined managed nodes/gateways with the frequency indicated in seconds (rounded to the nearest following minute: for example, if the frequency is set to 59 then the actual frequency will be 60 seconds, and if the frequency is set to 61 then the actual frequency will be 120 seconds). If the heartbeat is already running, this option changes the heartbeat frequency to the one specified.

–t
   This option stops the heartbeat with immediate effect on the defined managed nodes/gateways. The heartbeat must be running for this command to take effect. Although the heartbeat is stopped, the heartbeat process is not
terminated. In order to terminate the process, you need to issue the `wdmnn` command (together with the `-stop` and `-h` options).

**Authorization**

Requires the super, senior or admin roles.

**Examples**

1. The following example stops the heartbeat on the managed node/gateway `MyManagedNode`:
   
   ```
   wdmheartbeat -m MyManagedNode -t
   ```

2. The following example starts the heartbeat on the managed node/gateway `MyManagedNode`, the monitoring taking place every 180 seconds:
   
   ```
   wdmheartbeat -m MyManagedNode -s 180
   ```

3. The following example queries the status of the heartbeat on all managed node/gateways in the Tivoli management region:
   
   ```
   wdmheartbeat -m all -q
   ```

**See also**

`wdmconfig`

`wdmdiscovery`

`wdmmngcache`
**wdmloadprf**

Loads new and updates old profiles at a Tivoli management region server (Tivoli server).

**Syntax**

```
wdmloadprf -f profile_filename -g profile_manager#region [-P profile]
            [(-k | -m | -s) | -x]
```

**Description**

The `wdmloadprf` command makes new profiles available on the Tivoli server. If the command identifies an existing profile the command will update the profile, requiring you to choose whether to merge the new resource models with the existing ones, substitute the existing resource models for the new ones or keep the existing resource models.

If you use this option to move profiles between Tivoli servers, you should make sure that the resource models defined in the profile have already been loaded onto the receiving Tivoli server (using the `wdmrm` command), before loading the profile.

The input is a file created by using the `wdmdumpprf` command.

**Options**

- `–f profile_filename`
  Identifies the file containing the new profile. The name must include the full path. The file should be in Tivoli Management Framework format, unless the `–x` option is used to load a file in XML format (this option available only in Windows systems).

- `–g profile_manager#region`
  Identifies the profile manager to which the new profile is to belong or to which the profile to be updated belongs. The profile manager name must always be fully qualified with the policy region name in the syntax shown.

- `–P profile`
  Defines the profile to be added or updated. If the profile is not specified, the profile name contained in the XML file will be used.

  **Note:** To obtain a listing of the complete names of all the profiles that are on an endpoint, issue the following command:

  ```
  wdm1seng -e endpoint
  ```

  `[-k | -m | -s]`
  Defines how the model is updated. If none of the following options is defined, a new profile is created from the input information, provided that the supplied profile name does not already exist, in which case the command fails. The mutually exclusive updating options are as follows:

  - `–k`
    If a profile already exists with the supplied name, the `–k` option leaves all resource models in the existing profile as they are, only changing the profile parameters that are external to the resource models.

  - `–m`
    If a profile already exists with the supplied name, the `–m` option
performs the following actions on the resource models in the existing profile, by comparing the names of the resource models contained therein:

- A resource model in the XML file that does not match a resource model in the existing profile, is added to the existing profile.
- A resource model in the XML file that matches a resource model in the existing profile, overwrites the resource model in the existing profile.
- A resource model in the existing profile that does not match a resource model in the XML file is not changed.

\(-s\) If a profile already exists with the supplied name, the \(-s\) option substitutes the entire profile with that defined in the XML file.

\(-x\) This option can only be used on Windows systems. It determines that the file to be loaded is in XML format, instead of the default Tivoli Management Framework format.

**Authorization**

Requires the super, senior or admin roles.

**Examples**

1. The following example loads the profile stored in the file `MyXMLFile` in the `MyProfiles` directory and adds it to the profile manager `MyProfileManager` in the region `MyRegion`. The profile name will be determined from the contents of the XML file:

   ```shell
   wdmloadprf -f MyProfiles\MyXMLFile -g MyProfileManager#MyRegion
   ```

2. The following example loads the profile stored in the file `MyXMLFile` in the directory `MyProfiles` and adds it to the profile manager `MyProfileManager` in the region `MyRegion`, naming it `MyProfileName`:

   ```shell
   wdmloadprf -f MyProfiles\MyXMLFile -g MyProfileManager#MyRegion
            -P MyProfileName
   ```

3. The following example updates an existing profile identified in the file `MyXMLFile` in the `MyProfiles` directory and belonging to the profile manager `MyProfileManager` in the region `MyRegion`, and merges its resource models with those contained in the XML file:

   ```shell
   wdmloadprf -f MyProfiles\MyXMLFile -g MyProfileManager#MyRegion -m
   ```

4. The following example updates the profile `MyProfileName` in the region `MyRegion`, and belonging to the profile manager `MyProfileManager` by substituting its contents with those stored in the file `MyXMLFile` in the directory `MyProfiles`:

   ```shell
   wdmloadprf -f MyProfiles\MyXMLFile -g MyProfileManager#MyRegion
            -P MyProfileName -s
   ```

5. The following example updates the profile `MyProfileName` in the region `MyRegion`, and belonging to the profile manager `MyProfileManager` by keeping its resource models, only updating the profile parameters external to the resource models with those contained in the file `MyXMLFile` in the directory `MyProfiles`:

   ```shell
   wdmloadprf -f MyProfiles\MyXMLFile -g MyProfileManager#MyRegion
            -P MyProfileName -k
   ```
wdmloadprf

See also

wdmdumpprf

wdmeditprf
wdmlseng

Returns a list and the status of all resource models that have been distributed on a specified endpoint.

Syntax

wdmlseng -e endpoint [-verbose] [-xml]

wdmlseng -e endpoint -p qualified_profile#region [-verbose] [-xml]

wdmlseng -e endpoint -p qualified_profile#region -r resource_model [-verbose] [-xml]

wdmlseng -e endpoint -p qualified_profile#region -r resource_model -i indication [-verbose] [-xml]

Description

This command returns a list of all the resource models that have been distributed on the specified endpoint. The status of each resource model is also displayed. Some status codes are standard; others are defined by the user in the Tivoli Monitoring Resource Model Builder, in which case they are accompanied by a user-defined reason code.

The status can be one of the following:

Disabled
The resource model has been distributed but not started.

Error
An error has occurred with the Tivoli Monitoring engine, possibly due to a failure in the resource monitoring agent.

Failed reason code
A resource model has failed.

Failed after recovery reason code
A resource model has failed after the recovery operation has tried running it three times (see “Recovering”).

Failing reason code
The endpoint engine tries to run the resource model once every three minutes, indefinitely.

Missed Prereq
One or more prerequisites of the resource model is not being met. For example, if you distribute the Print resource model to a Windows NT system, this status is generated because the resource model works only with Windows 2000.

Not compiled
The resource model has not been compiled.

Recovering reason code
The endpoint engine tries to run the resource model once every cycle, for three cycles only, after which the status is changed to failed after recovery.

Retrying reason code
The endpoint engine tries to run the resource model three times in each cycle, indefinitely.

Running
The resource model is running.
Scheduled
The resource model has been scheduled to be started.

Stopped
The resource model has been stopped. It is compiled and enabled and is not in an error state but it is not running and is not scheduled.

Unable to start reason code
A resource model has been unable to start.

Several examples of the output are given
1. The first shows the output received when sending a request at endpoint level (with just the endpoint specified) and without the verbose option:
   Forwarding the request to the engine...
The following profiles are running:
   myprofile#ucaselli-region
   TMW_Processor :Running
   [ucaselli][y:/w32-ix86/cdrom]
2. The second output example shows the same endpoint with the verbose setting:
   Forwarding the request to the engine...
The following profiles are running:
   myprofile#ucaselli-region
   TMW_Processor :Running
   TMW_ProcessorBusy 100 %
   TMW_HighProcesses 100 %
   TMW_CPUCanKeepUpWithHW 100 %
   TMW_HWKeepingCPUBusy 100 %
   TMW_BusyHardware 100 %
   TMW_HighPercentUsageDelta 100 %
   [ucaselli][y:/w32-ix86/cdrom]
3. The third output example shows the same endpoint with the verbose setting in XML format:
   Forwarding the request to the engine...
The following profiles are running:
   <Profiles>
     <Profile Name="myprofile#ucaselli-region">
       <RMs>
         <RM Name="TMW_Processor" Status="Running">
           <Indications>
             <Indication Name="TMW_ProcessorBusy" Health="100"> </Indication>
             <Indication Name="TMW_HighProcesses" Health="100"> </Indication>
             <Indication Name="TMW_CPUCanKeepUpWithHW" Health="100"> </Indication>
             <Indication Name="TMW_HWKeepingCPUBusy" Health="100"> </Indication>
             <Indication Name="TMW_BusyHardware" Health="100"> </Indication>
             <Indication Name="TMW_HighPercentUsageDelta" Health="100"> </Indication>
           </Indications>
         </RM>
       </RMs>
     </Profile>
   </Profiles>

Options
-e endpoint
Specifies the endpoint on which the command is to be performed.
-i indication
Selects the indication on which the command is to be performed. Use the event internal name, as specified for each resource model in the IBM Tivoli Monitoring: Resource Model Reference Guide. If the indication is not specified the command is performed on all indications in the specified resource model.

-p qualified_profile#region
Defines the profile on which the command is to be performed. If the profile is not specified, the command will be performed on all profiles at the endpoint. The profile name must always be fully qualified with the policy region name in the syntax shown. The profile name can have either of the following formats:

profile#region
object_label.profile#region
in the case of application management objects

Note: To obtain a listing of the complete names of all the profiles that are on an endpoint, issue the following command:

wdmlseng -e endpoint

-r resource_model
Selects the resource model on which the command is to be performed. Use the resource model internal name, as specified for each resource model in the IBM Tivoli Monitoring: Resource Model Reference Guide. If the resource model is not specified the command is performed on all resource models in the specified profile.

-verGate
Displays all indications for each resource model. The worst instance key (if it exists) and the status (consolidation level) of each indication are also displayed. The consolidation level indicates the completion status of the generated occurrences that are required to trigger the particular event.

If a profile is specified, all indications are displayed for each resource model of that profile.

If a resource model is also specified, all instances are displayed for each indication of that resource model.

If an indication is also specified, all categories (the indication’s string property value) and metrics (the indication’s numeric property value) data are displayed for each instance.

-xml
Displays the output in XML format. In case of a command error, the corresponding error message is displayed and no XML is generated.

If the output contains DBCS characters, it should be redirected to a file and viewed using an editor that supports UTF-8 encoding.

Authorization
Requires the super, senior, admin or user roles.

Examples
1. This example shows the command necessary to list all the resource models at the endpoint MyEndpoint with full details of all indications in XML format:

wdmlseng -e MyEndpoint -verbose -xml
wdmlseng

2. This example shows the command necessary to list all the resource models in the profile **MyProfile** at the endpoint **MyEndpoint** with full details of all indications in XML format:

```
wdmlseng -e MyEndpoint -p MyProfile -verbose -xml
```

3. This example shows the command necessary to list the resource model **MyResourceModel** in the profile **MyProfile** at the endpoint **MyEndpoint** with full details of all indications in XML format:

```
wdmlseng -e MyEndpoint -p MyProfile -r MyResourceModel -verbose -xml
```

4. This example shows the command necessary to list the indication **MyIndication** in the resource model **MyResourceModel** in the profile **MyProfile** at the endpoint **MyEndpoint** with full details of all indications in XML format:

```
wdmlseng -e MyEndpoint -p MyProfile -r MyResourceModel -i MyIndication -verbose -xml
```

**See also**

wdmcmd

wdmeng

wdmtrceng
wdmmn

Stops or starts selected Tivoli Monitoring processes on one or all managed nodes/gateways.

Syntax

wdmmn –start [–m {managed_node | all}] [–d] [–t] [–r]
wdmmn –stop [–m {managed_node | all}] [–b] [–d] [–h] [–t] [–r]

Description

This command starts or stops the various Tivoli Monitoring processes running on managed nodes/gateways. The processes that can be started and stopped are:

- Task engine
- Tivoli Business Systems Manager Adapter
- Heartbeat engine (to start it, use wdmheartbeat)
- The request manager processor
- The data collector processor for the Tivoli Enterprise Data Warehouse Support component

You can either specify individual managed nodes/gateways or all managed nodes/gateways on which Tivoli Monitoring is installed.

Options

- **–b**
  Stops the Tivoli Business Systems Manager Adapter. The process is automatically restarted when a wdmdiscovery command is issued.

- **–d**
  Starts or stops the data collector processor used by the Tivoli Enterprise Data Warehouse Support component. When –stop is used, any pending data collection requests are suspended. When –start is used, any pending data collection requests are resumed.

- **–h**
  Stops the heartbeat engine.

- **–m {managed_node | all}**
  Identifies the managed node/gateway on which the command is to be run. If it is not supplied, the command will be performed on the managed node/gateway from which it is issued. If it is supplied you must identify a specific managed node/gateway, or all managed nodes/gateways using the all argument.

- **–start**
  Starts the defined process or processes. If no –d, –r, or –t argument is supplied, the task engine is started (–t).

- **–stop**
  Stops the defined process or processes. If no –b, –d, –h, –r, or –t argument is supplied, the task engine is stopped (–t).

- **–t**
  Starts or stops the task engine. This is the default value.

- **–r**
  Starts or stops the request manager.
Authorization

Requires the super, senior or admin roles.

Examples

1. The following example stops the Tivoli Monitoring heartbeat engine on the managed node/gateway from which it is issued.
   `wdmmn -stop -h`
2. The following example starts the Tivoli Monitoring task engine on the managed node/gateway from which it is issued.
   `wdmmn -start`
3. The following example starts the task engine on managed node/gateway MyManagedNode.
   `wdmmn -start -m MyManagedNode -t`
4. The following example stops the Tivoli Business Systems Manager Adapter on all the managed nodes/gateways.
   `wdmmn -stop -m all -b`
5. The following example stops the heartbeat engine on all the managed nodes/gateways.
   `wdmmn -stop -m all -h`
6. The following example stops the Tivoli Monitoring task engine and the Tivoli Business Systems Manager Adapter on all the managed nodes/gateways.
   `wdmmn -stop -m all -t -b`
7. The following example stops the heartbeat engine and the Tivoli Business Systems Manager Adapter on all the managed nodes/gateways.
   `wdmmn -stop -m all -h -b`
8. The following example stops the Tivoli Monitoring task engine and the heartbeat engine on all the managed nodes/gateways.
   `wdmmn -stop -m all -t -h`
9. The following example stops all processes on all the managed nodes/gateways.
   `wdmmn -stop -m all -t -h -b -r`

See also

`wdmheartbeat`
wdmmngcache

Deletes or lists part or all of the contents of the gateway endpoint cache.

Syntax

wdmmngcache [-m {managed_node | @managed_nodes_file | all}] -l [-v]

wdmmngcache [-m {managed_node | @managed_nodes_file | all}]
   -d [all | discovered | dead | endpoint | @endpoints_file]

Description

The wdmmngcache command operates on a selected managed node/gateway, a list of managed nodes/gateways contained in a file or all managed nodes/gateways, and either lists or deletes all or part of the contents of the cache maintained by the Tivoli Business Systems Manager Adapter.

If the delete option is used, endpoint details are marked for deletion but are not deleted until the next discovery action. Thus, you are recommended to carry out a discovery prior to using the list option of this command, and after using the delete option.

The output of the list option is available in two formats. A sample of the output without using the -v argument is as follows, showing details of the heartbeat status of the endpoints in the cache:

```
Processing ManagedNode mpulp...
Processing ManagedNode dmw2k2...
Endpoint | Status
------------------------------+-------
mpulp-ep                                     DMAgentAlive
dmw2k2-ep                                     DMAgentAlive
```

A sample of the output using the -v argument is as follows, showing the Tivoli Business Systems Manager status of the endpoints in the cache:

```
Processing ManagedNode mcrudele...
Processing ManagedNode dmw2k2...
Warning: DM_Advanced_Edition feature not installed on the Managed Node 'boccaccio'. Skipping...
Endpoint | HB status | TBSM status
---------------------+-----------------------
dmw2k2-ep         DMEngineOff   Not discovered
mcrudele-ep       DMEngineOff   Not discovered
```

The possible heartbeat statuses of the endpoints in the cache are the following:

Unreachable

The endpoint cannot be reached

DMEngineOff

The Tivoli Monitoring engine is down

RMsInError

The Tivoli Monitoring engine is running but some resource models are in error

DMAgentAlive

No problems discovered

HBOff

The heartbeat is not active
The possible **Tivoli Business Systems Manager statuses** of the endpoints in the cache are the following:

**Not discovered**
A discovery was not carried out

**Discovered**
A discovery was carried out

**Modified**
The endpoint information were modified. A new discovery should be carried out.

**Removed**
The endpoint was removed from cache, and put in a "removed" state. A new discovery should be carried out to remove the resource from Tivoli Business System Manager. At that point, the resource is not shown anymore on the Tivoli Business System Manager console until the endpoint registers itself again in the endpoint cache.

### Options

```
–m {managed_node | @managed_nodes_file | all}
```

Specifies the managed nodes/gateways on which the command is to be carried out. If this option is not used the command is carried out on the managed node/gateway from which it is being run.

The options are as follows:

- `managed_node`
  The name of the managed node/gateway on which the command is to be performed.

- `@managed_nodes_file`
  A file containing managed node/gateway names separated by the CR/LF character; the @ symbol is an indicator that a file name follows but the symbol should not form part of the filename.

- `all`
  All managed nodes/gateways in the Tivoli region

```
–l
```

This option lists the contents of the cache at the defined managed nodes/gateways. It has one optional argument:

- `–v`
  Provides details of the Tivoli Business Systems Manager status of the endpoints in the cache. If this argument is not supplied, the command provides details of the heartbeat status of the endpoints in the cache.

```
–d {all | discovered | dead | endpoint | @endpoints_file}
```

Deletes the cache appertaining to the defined endpoints. You should note that the deleted endpoint details remain visible using the –l option until the next discovery request is made to the managed node/gateway to which the endpoint is attached. The endpoint definition can be one of the following:

- `all`
  All endpoints attached to the managed nodes/gateways

- `discovered`
  All endpoints revealed as active by any discovery commands

- `dead`
  All endpoints revealed as inactive by any discovery commands

- `endpoint`
  An endpoint label.
@endpoints_file
A file containing endpoint labels separated by the CR/LF character; the @ symbol is an indicator that a file name follows but the symbol should not form part of the filename.

Authorization
Requires the super, senior or admin roles.

Examples
1. The following example lists the cache at the managed node/gateway MyManagedNode:
   wdmmngcache -m My ManagedNode -l
2. The following example lists the status of the Tivoli Business Systems Manager Adapter at all gateways/managed nodes in the Tivoli management region:
   wdmmngcache -m all -l -v
3. The following example deletes the cached details of all endpoints at all managed nodes/gateways in the Tivoli region:
   wdmmngcache -m all -d all
4. The following example deletes the cached details of endpoints revealed by a discovery at all managed nodes/gateways in the Tivoli region:
   wdmmngcache -m all -d discovered
5. The following example deletes the cached details of endpoints not revealed by a discovery at all managed nodes/gateways in the Tivoli region:
   wdmmngcache -m all -d dead
6. The following example deletes the cached details of endpoint MyEndpoint1 at the managed node/gateway from which the command is given:
   wdmmngcache -d MyEndpoint1
7. The following example deletes the cached details of all endpoints listed in the file MyEndpointList at the managed node/gateway form which the command is given:
   wdmmngcache -d @MyEndpointList

See also
wdmconfig
wdmdiscovery
wdmheartbeat
**wdmrm**

Adds, lists or removes a specified default resource model at the Tivoli management region server or managed node/gateway from where it is issued. It also adds the NLS catalog to an already installed default resource model.

The command works only for the server and managed nodes of the local region. In the case of multiple interconnected regions, the command must be issued at each specific region.

**Syntax**

```plaintext
wdmrm -add resource_model_tarfile
wdmrm -addcat resource_model [-f catalog_file -l locale]
wdmrm -list
wdmrm -remove resource_model
```

**Description**

This command adds, lists or removes the specified resource model at the Tivoli server. In the case of the remove option it removes the resource model from the Tivoli Name Registry (TNR) on the server and from all profiles to which the resource model has previously been added. It does not remove the resource model from the copies of the profiles at the endpoints.

The command also adds the NLS catalog to the resource model, so that when it is subsequently distributed to endpoints, the resource model has all the necessary NLS information to support the display of localized events on the Tivoli Enterprise Console server.

The list option produces output similar to the following for each resource model:

```
Resource -> DMXCpu
  NLS name     : CPU
  product_id   : none
  major_version: 1
  minor_version: 0
  platform     : aix4-r1\hpux10\linux-ix86\linux-s390\solaris2
  message catalog: DMXCpu
  zip file     : DMXCpu.zip
```

**Options**

- **-add**
  Adds the resource model in the specified `resource_model_tarfile` to the Tivoli server. If the resource model already exists, it gets replaced by the new version, in which case, you must stop and restart the endpoint engine.

- **-addcat**
  Adds the specified NLS catalog to the specified resource model at the Tivoli server; if the resource model does not exist the command fails. If no NLS catalog is specified, the command searches the `msg_cat` directory and updates the specified resource model with the details of any NLS catalogs that it finds, that are not already added to the resource model (locales that are absent are added, those that are present are upgraded).
[-f catalog_file -l locale]
Defines a specific NLS catalog that is to be added to the resource model. If this option is not specified, the resource model will be updated with the details of all NLS catalogs in the msg_cat directory.

--list
Lists all resource models (by internal name) defined at the Tivoli server.

--remove
Removes the specified resource model from the Tivoli server, removing it also from all profile definitions at the server that contain the resource model. Note that the removal does not apply to profile definitions at the endpoints.

resource_model
Specifies the resource model to remove, or to which you want to add NLS catalog details. Use the resource model internal name, as specified for each resource model in the IBM Tivoli Monitoring: Resource Model Reference Guide.

resource_model_tarfile
Specifies the name of the file that is produced by Tivoli Resource Model Builder when you build a resource model package.

Authorization
On a Windows system, this command can only be run from the bash Shell. It requires the super, admin or senior roles.

Examples
1. The following example adds a resource model created by the user, and saved in a file called MyResModel.tar, to the list of available resource models.
   ```shell
   wdmrm -add MyResModel.tar
   ```
2. The following example updates the resource model called MyResModel, with all the NLS catalog details in the msg_cat directory:
   ```shell
   wdmrm -addcat MyResModel
   ```
3. The following example adds the locale MyLocale in the NLS catalog MyCatalog to the resource model called MyResModel:
   ```shell
   wdmrm -addcat MyResModel -f MyCatalog -l MyLocale
   ```
4. The following example returns a list of all the resource models that are defined at the Tivoli management region.
   ```shell
   wdmrm -list
   ```
5. The following example removes the resource model MyResModel from the resource model registry and all profiles in the Tivoli region (except copies of profiles at endpoints):
   ```shell
   wdmrm -remove MyResModel
   ```

See also
None.
wdmtrceng

Sets the trace parameters of the Tivoli Monitoring engine at the endpoint.

Syntax

wdmtrceng –e endpoint logfile_name trace_level size

Description

This command sets the trace parameters of the engine on the endpoint. For more
details about the traces, see "Logs and Traces" in the IBM Tivoli Monitoring: Problem
Determination Guide.

Options

–e endpoint

Specifies the endpoint where the command is to be executed.

trace_level

The value set for the trace level of the endpoint’s engine. The level can be one
of the following:

0 Only errors are stored in the log file. This is the default value. For
performance reasons, Tivoli recommends that you do not change this value
unless you are experiencing problems that you want to report to IBM
Software Support.

1 Only warnings and errors are stored in the log file.

2 All steps of the monitoring process are stored in the log file.

3 Verbose mode: all operations performed during monitoring are stored in the
log file.

logfile_name

• For Windows endpoints, the name, with its complete path, of the logfile
where the output of the trace is to be saved.

• For UNIX, Linux, or OS/400 endpoints, this option to set the log name
cannot be used; the log names are as described in section "Non-Windows
Endpoint Logs" in the IBM Tivoli Monitoring: Problem Determination Guide.

size

The maximum size of the logfile, specified in bytes. Specify –1 to reset the size
to the default value (5 MB for Windows, 2.5MB for UNIX/Linux).

Authorization

Requires the super, senior or admin roles.

Examples

1. The following example sets the logfile MyLogFile to trace level verbose mode
and a size of 1 MB on the endpoint MyEP1.

   wdmtrceng -e MyEP1 C:\Log\MyLogFile 3 1000000

2. The following example resets the default trace level on the endpoint MyEP1.

   wdmtrceng -e MyEP1 "" -1 -1
See also
  wdmcmd
  wdmeng
  wdmlseng
wtmntaddrm

This is an obsolete command that has now been merged into the new command \texttt{wdmrm}; the command has been retained in this version of the product to provide continuity for customers with scripts that used the old command. It will not be included in any future release, so customers are advised to convert scripts to use the new command.

For details of the new command \texttt{wdmrm} see “\texttt{wdmrm}” on page 162.

\textbf{See also}

- \texttt{wtmntdefrm}
- \texttt{wtmntrrmrm}
**wtmntdefrm**

This is an obsolete command that has now been merged into the new command `wdmrm`; the command has been retained in this version of the product to provide continuity for customers with scripts that used the old command. It will not be included in any future release, so customers are advised to convert scripts to use the new command.

For details of the new command `wdmrm` see "wdmrm" on page 162.

**See also**

- `wtmntaddrm`
- `wtmntrmrm`
**wtmntrmrm**

This is an obsolete command that has now been merged into the new command **wdmrm**; the command has been retained in this version of the product to provide continuity for customers with scripts that used the old command. It will not be included in any future release, so customers are advised to convert scripts to use the new command.

For details of the new command **wdmrm** see "wdmrm" on page 162.

**See also**

wtmntaddrm

wtmntdefrm
Return codes from commands

Commands may return various codes, as indicated in the following table. The first column shows the returned numeric value, the second column shows the symbolic code, and the third column indicates the corresponding message identifier.

Table 16. Return codes from commands

<table>
<thead>
<tr>
<th>Returned code</th>
<th>Symbolic code</th>
<th>Message identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>-20</td>
<td>EX_DATACOLLECTOR_STOP_REMOVE_ERROR</td>
<td></td>
</tr>
<tr>
<td>-19</td>
<td>EX_DATACOLLECTOR_QUERY_ERROR</td>
<td></td>
</tr>
<tr>
<td>-18</td>
<td>EX_COULDNOT_STOP_ENGINE</td>
<td></td>
</tr>
<tr>
<td>-17</td>
<td>EX_TBSMENGINE_STOP_FAIL</td>
<td></td>
</tr>
<tr>
<td>-16</td>
<td>EX_TBSMENGINE_START_FAIL</td>
<td></td>
</tr>
<tr>
<td>-15</td>
<td>EX_CANNOT_STOP_HB_ENGINE</td>
<td></td>
</tr>
<tr>
<td>-14</td>
<td>EX_CANNOT_START_RM_ENGINE</td>
<td></td>
</tr>
<tr>
<td>-13</td>
<td>EX_CANNOT_STOP_RM_ENGINE</td>
<td></td>
</tr>
<tr>
<td>-12</td>
<td>EX_CANNOT_STOP_DATACOLLECTORENGINE</td>
<td></td>
</tr>
<tr>
<td>-11</td>
<td>EX_CANNOT_START_DATACOLLECTORENGINE</td>
<td></td>
</tr>
<tr>
<td>-10</td>
<td>EX_CANNOT_TERMINATE_TASKENGINE</td>
<td></td>
</tr>
<tr>
<td>-9</td>
<td>CANNOT_CREATE_PROFILE</td>
<td></td>
</tr>
<tr>
<td>-8</td>
<td>EX_WRONG_XML_VERSION</td>
<td></td>
</tr>
<tr>
<td>-6</td>
<td>EX_CLI_USAGE_TBSM_SETTINGS_RC</td>
<td></td>
</tr>
<tr>
<td>-5</td>
<td>EX_TEC_SERVER_LIST_CHECK_FAILED</td>
<td></td>
</tr>
<tr>
<td>-4</td>
<td>EX_CLI_USAGE_TEC_SETTINGS_RC</td>
<td></td>
</tr>
<tr>
<td>-2</td>
<td>EX_CLI_GENERIC_RC</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>EX_CLI_USAGE_RC</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>EX_CLI_SUCCESS_RC</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>EX_CLI_USAGE_RC</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>EX_CANNOT_OPEN_FILE</td>
<td>AMW0036E</td>
</tr>
<tr>
<td>46</td>
<td>EX_UNABLE_TOEXECUTE_OPERATION</td>
<td>AMW0046E</td>
</tr>
<tr>
<td>47</td>
<td>EX_OPERATION_ALREADY_DONE</td>
<td>AMW0047E</td>
</tr>
<tr>
<td>48</td>
<td>EX_RESMODEL_NOT_PRESENT</td>
<td>AMW0048E</td>
</tr>
<tr>
<td>49</td>
<td>EX_PROFILE_NOT_PRESENT</td>
<td>AMW0049E</td>
</tr>
<tr>
<td>52</td>
<td>EX_ENGINE_DOWN</td>
<td>AMW0052E</td>
</tr>
<tr>
<td>64</td>
<td>EXPROFILE_NOT_EXISTENT</td>
<td>AMW0064E</td>
</tr>
<tr>
<td>65</td>
<td>EX_RM_NOT_EXISTENT</td>
<td>AMW0065E</td>
</tr>
<tr>
<td>66</td>
<td>EX_RM_NOT_IN_PROFILE</td>
<td>AMW0066E</td>
</tr>
<tr>
<td>67</td>
<td>EX_RM_MISSING_CYCLE_TIME</td>
<td>AMW0067E</td>
</tr>
<tr>
<td>69</td>
<td>EX_RM_MISSING_LOGGING_PERIOD</td>
<td>AMW0069E</td>
</tr>
<tr>
<td>70</td>
<td>EX_RM_WRONG_LOGGING_PERIOD</td>
<td>AMW0070E</td>
</tr>
<tr>
<td>71</td>
<td>EX_RM_MISSING_AGGR_SET</td>
<td>AMW0071E</td>
</tr>
<tr>
<td>72</td>
<td>EX_RM_MISSING_AGGR_PERIOD</td>
<td>AMW0072E</td>
</tr>
<tr>
<td>73</td>
<td>EX_RM_WRONG_AGGR_PERIOD</td>
<td>AMW0073E</td>
</tr>
<tr>
<td>Returned code</td>
<td>Symbolic code</td>
<td>Message identifier</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>74</td>
<td>EX_RM_MISSING_EVENT</td>
<td>AMW0074E</td>
</tr>
<tr>
<td>75</td>
<td>EX_RM_MISSING_OCCURRENCIES</td>
<td>AMW0075E</td>
</tr>
<tr>
<td>76</td>
<td>EX_RM_MISSING_HOLES</td>
<td>AMW0076E</td>
</tr>
<tr>
<td>77</td>
<td>EX_RM_EVENT_NOT_EXISTING</td>
<td>AMW0077E</td>
</tr>
<tr>
<td>78</td>
<td>EX_RM_MISSING_THRESHOLD</td>
<td>AMW0078E</td>
</tr>
<tr>
<td>79</td>
<td>EX_RM_MISSING_THRESHOLD_VALUE</td>
<td>AMW0079E</td>
</tr>
<tr>
<td>80</td>
<td>EX_RM_MISSING_PROPERTY</td>
<td>AMW0080E</td>
</tr>
<tr>
<td>82</td>
<td>EX_TEC_MISSING_SERVER_NAME</td>
<td>AMW0082E</td>
</tr>
<tr>
<td>83</td>
<td>EX_TEC_MISSING_TEC_PORT</td>
<td>AMW0083E</td>
</tr>
<tr>
<td>84</td>
<td>EX_TEC_WRONG_DELIVERY</td>
<td>AMW0084E</td>
</tr>
<tr>
<td>85</td>
<td>EX_RM_PARAM_WRONG</td>
<td>AMW0085E</td>
</tr>
<tr>
<td>86</td>
<td>EX_COULD_NOT_START_TRANSACTION</td>
<td>AMW0086E</td>
</tr>
<tr>
<td>87</td>
<td>EX_DUPLICATE_RM_IN_PROFILE</td>
<td>AMW0087E</td>
</tr>
<tr>
<td>88</td>
<td>EX_RM_MISSING_SEVERITY</td>
<td>AMW0088E</td>
</tr>
<tr>
<td>91</td>
<td>EX_XERCESS_INIT_FAILED</td>
<td>AMW0091E</td>
</tr>
<tr>
<td>92</td>
<td>EX_XERCESSPARSE_FAILED</td>
<td>AMW0092E</td>
</tr>
<tr>
<td>97</td>
<td>EX_NO_XML_DECLARATION</td>
<td>AMW0097E</td>
</tr>
<tr>
<td>98</td>
<td>EX_NO_DTD_FOUND</td>
<td>AMW0098E</td>
</tr>
<tr>
<td>99</td>
<td>EX_NO_ELEMENT_FOUND</td>
<td>AMW0099E</td>
</tr>
<tr>
<td>100</td>
<td>EX_DTD_NOT_SUPPORTED</td>
<td>AMW0100E</td>
</tr>
<tr>
<td>101</td>
<td>EX_NEED_VALID_NUMBER</td>
<td>AMW0101E</td>
</tr>
<tr>
<td>102</td>
<td>EX_PROFILE_TOO_OLD</td>
<td>AMW0102E</td>
</tr>
<tr>
<td>103</td>
<td>EX_PROFILE_TOO_NEW</td>
<td>AMW0103E</td>
</tr>
<tr>
<td>104</td>
<td>EX_PROFILE_NAME_MISSED</td>
<td>AMW0104E</td>
</tr>
<tr>
<td>105</td>
<td>EX_TRUEORFALSE_REQUIRED</td>
<td>AMW0105E</td>
</tr>
<tr>
<td>106</td>
<td>EX_WRONG_TEC_ATTRIBUTE</td>
<td>AMW0106E</td>
</tr>
<tr>
<td>109</td>
<td>EX_RM_NOT_INSTALLED</td>
<td>AMW0109E</td>
</tr>
<tr>
<td>110</td>
<td>EX_BAD_ZIP_FILE</td>
<td>AMW0110E</td>
</tr>
<tr>
<td>111</td>
<td>EX_BAD_PLATFORMS</td>
<td>AMW0111E</td>
</tr>
<tr>
<td>112</td>
<td>EX_WRONG_VERSION</td>
<td>AMW0112E</td>
</tr>
<tr>
<td>113</td>
<td>EX_RM_LOGGING_NOT_ALLOWED</td>
<td>AMW0113E</td>
</tr>
<tr>
<td>114</td>
<td>EX_AGGREGATION_PERIOD_ERROR</td>
<td>AMW0114E</td>
</tr>
<tr>
<td>115</td>
<td>EX_THRESHOLD_NOT_DEFINED</td>
<td>AMW0115E</td>
</tr>
<tr>
<td>116</td>
<td>EX_THRESHOLD_MISSED</td>
<td>AMW0116E</td>
</tr>
<tr>
<td>117</td>
<td>EX_PARAM_NOT_DEFINED</td>
<td>AMW0117E</td>
</tr>
<tr>
<td>118</td>
<td>EX_RM_PARAMETER_MISSED</td>
<td>AMW0118E</td>
</tr>
<tr>
<td>119</td>
<td>EX_BAD_PARAM_TYPE</td>
<td>AMW0119E</td>
</tr>
<tr>
<td>120</td>
<td>EX_VALUE_MUSTBE_SPECIFIED</td>
<td>AMW0120E</td>
</tr>
<tr>
<td>121</td>
<td>EX_VALUE_NOT_ALLOWED</td>
<td>AMW0121E</td>
</tr>
</tbody>
</table>
Table 16. Return codes from commands (continued)

<table>
<thead>
<tr>
<th>Returned code</th>
<th>Symbolic code</th>
<th>Message identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>122</td>
<td>EX_EVENT_NOT_DEFINED</td>
<td>AMW0122E</td>
</tr>
<tr>
<td>123</td>
<td>EX_RM_EVENT_MISSED</td>
<td>AMW0123E</td>
</tr>
<tr>
<td>124</td>
<td>EX_RM_INVALID_EVENT_SEVERITY</td>
<td>AMW0124E</td>
</tr>
<tr>
<td>125</td>
<td>EX_ACTION_NOT_ALLOWED</td>
<td>AMW0125E</td>
</tr>
<tr>
<td>127</td>
<td>EX_PROFILE_ALREADY_EXIST</td>
<td>AMW0127E</td>
</tr>
<tr>
<td>131</td>
<td>EX_RM_TASK_WRONG_NUMBER_ARGS</td>
<td>AMW0131W</td>
</tr>
<tr>
<td>133</td>
<td>EX_RM_MISSING_PARAM_NAME</td>
<td>AMW0133E</td>
</tr>
<tr>
<td>134</td>
<td>EX_RM_MISSING_PARAM_VALUE</td>
<td>AMW0134E</td>
</tr>
<tr>
<td>135</td>
<td>EX_RM_PARAM_VALUE_NOT_A_NUMBER</td>
<td>AMW0135E</td>
</tr>
<tr>
<td>136</td>
<td>EX_RM_PARAMETER_NOT_EXISTING</td>
<td>AMW0136E</td>
</tr>
<tr>
<td>139</td>
<td>EX_RM_PARAMETER_INVALID_VALUE</td>
<td>AMW0139E</td>
</tr>
<tr>
<td>140</td>
<td>EX_RM_PARAMETER_CANNOT_DELETE</td>
<td>AMW0140E</td>
</tr>
<tr>
<td>141</td>
<td>EX_XML_NOT_SUPPORTED</td>
<td>AMW0141E</td>
</tr>
<tr>
<td>170</td>
<td>EX_INVALID_TARGET</td>
<td>AMW0170E</td>
</tr>
<tr>
<td>172</td>
<td>EX_INVALID_SUBSCRIBERS</td>
<td>AMW0172E</td>
</tr>
<tr>
<td>173</td>
<td>EX_PROFILE_WITHOUT_SUBSCRIBERS</td>
<td>AMW0173E</td>
</tr>
<tr>
<td>175</td>
<td>EX_RM_TASK_MISSING_EVENT_NAME</td>
<td>AMW0175E</td>
</tr>
<tr>
<td>176</td>
<td>EX_RM_TASK_MISSING_LIB_NAME</td>
<td>AMW0176E</td>
</tr>
<tr>
<td>177</td>
<td>EX_RM_TASK_MISSING_NAME</td>
<td>AMW0177E</td>
</tr>
<tr>
<td>178</td>
<td>EX_RM_TASK_NOT_FOUND</td>
<td>AMW0178E</td>
</tr>
<tr>
<td>179</td>
<td>EX_RM_TASK_NOT_BOUND_TO_EVENT</td>
<td>AMW0179E</td>
</tr>
<tr>
<td>204</td>
<td>EX_CYCLETIME_OUT_OF_RANGE</td>
<td>AMW0204E</td>
</tr>
<tr>
<td></td>
<td>EX_OCCURRENCIES_OUT_OF_RANGE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>EX_HOLES_OUT_OF_RANGE</td>
<td></td>
</tr>
<tr>
<td>214</td>
<td>EX_RM_MISSING_TEDW_SETTING</td>
<td>AMW0214E</td>
</tr>
<tr>
<td>215</td>
<td>EX_RM_MISSING_RAW_SETTING</td>
<td>AMW0215E</td>
</tr>
<tr>
<td>216</td>
<td>EX_INTERP_NOT_FOUND</td>
<td>AMW0216E</td>
</tr>
<tr>
<td>218</td>
<td>EX_UNABLE_TODELETE_PROFILE</td>
<td>AMW0218E</td>
</tr>
<tr>
<td>219</td>
<td>EX_CLI_OPTION_NOT_ALLOWED</td>
<td>AMW0219E</td>
</tr>
<tr>
<td>227</td>
<td>EX_RESPONSE_LEVEL_NOT_EXISTENT</td>
<td>AMW0227E</td>
</tr>
<tr>
<td>233</td>
<td>EX_EDITPROFILE_WRCNGL_WOPERAND</td>
<td>AMW0233E</td>
</tr>
<tr>
<td>234</td>
<td>EX_NEED_AVALID_CLEARING</td>
<td>AMW0234E</td>
</tr>
<tr>
<td>235</td>
<td>EX_NEED_AVALID_POLICY</td>
<td>AMW0235E</td>
</tr>
<tr>
<td>250</td>
<td>EX_NO_PROFILE_RUNNING</td>
<td>AMW0250E</td>
</tr>
<tr>
<td>251</td>
<td>EX_ONLY_PROFILE_REQUIRED</td>
<td>AMW0251E</td>
</tr>
<tr>
<td>258</td>
<td>EX_RM_MISSING_SCHEDULE</td>
<td>AMW0258E</td>
</tr>
<tr>
<td>259</td>
<td>EX_RM_MISSING_START</td>
<td>AMW0259E</td>
</tr>
<tr>
<td>260</td>
<td>EX_RM_MISSING_STOP</td>
<td>AMW0260E</td>
</tr>
<tr>
<td>261</td>
<td>EX_RM_MISSING_RULE</td>
<td>AMW0261E</td>
</tr>
<tr>
<td>262</td>
<td>EX_RM_MISSING_RULENAME</td>
<td>AMW0262E</td>
</tr>
<tr>
<td>Returned code</td>
<td>Symbolic code</td>
<td>Message identifier</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>263</td>
<td>EX_RM_MISSING_RULECYCLE</td>
<td>AMW0263E</td>
</tr>
<tr>
<td>264</td>
<td>EX_RM_MISSING_RULEINTERVAL</td>
<td>AMW0264E</td>
</tr>
<tr>
<td>265</td>
<td>EX_RM_INVALID_RULETIMES</td>
<td>AMW0265E</td>
</tr>
<tr>
<td>266</td>
<td>EX_RM_INVALID_STARTDAY</td>
<td>AMW0266E</td>
</tr>
<tr>
<td>267</td>
<td>EX_RM_INVALID_STARTMONTH</td>
<td>AMW0267E</td>
</tr>
<tr>
<td>268</td>
<td>EX_RM_INVALID_STARTYEAR</td>
<td>AMW0268E</td>
</tr>
<tr>
<td>269</td>
<td>EX_RM_INVALID_STOPDAY</td>
<td>AMW0269E</td>
</tr>
<tr>
<td>270</td>
<td>EX_RM_INVALID_STOPMONTH</td>
<td>AMW0270E</td>
</tr>
<tr>
<td>271</td>
<td>EX_RM_INVALID_STOPYEAR</td>
<td>AMW0271E</td>
</tr>
<tr>
<td>272</td>
<td>EX_RM_INVALID_RULECYCLE</td>
<td>AMW0272E</td>
</tr>
<tr>
<td>273</td>
<td>EX_ENGINE_INIT_FAILED</td>
<td>AMW0273E</td>
</tr>
<tr>
<td>274</td>
<td>EX_CANNOT_PARSE_LABEL_FILE_RC</td>
<td>AMW0274E</td>
</tr>
<tr>
<td>275</td>
<td>EX_COULD_NOT_EXECUTE_COMMAND</td>
<td>AMW0275E</td>
</tr>
<tr>
<td>278</td>
<td>EX_PURGING_DATA_ERROR</td>
<td>AMW0278E</td>
</tr>
<tr>
<td>279</td>
<td>EX_WRONG_TASK_ARGS_NUMBER</td>
<td>AMW0279E</td>
</tr>
<tr>
<td>282</td>
<td>EX_NO_VALID_ENDPOINTS</td>
<td>AMW0282E</td>
</tr>
</tbody>
</table>
Part 2. Web Health Console

Chapter 9. Introduction to the Web Health Console
Overview ........................................ 175
  Understanding resource health ................ 175
  Connecting the Web Health Console .......... 176
  The request manager processor ............... 176
System requirements ........................... 177
Software requirements .......................... 177
Installing the Web Health Console ............ 178
  Installing on Windows .......................... 178
  Installing on UNIX .............................. 179
  Troubleshooting the installation .............. 180
  Enabling the Web Health Console to use your
  resource models ............................... 183
  Enabling Secure Socket Layer support ........ 183
Uninstalling the Web Health Console .......... 185

Chapter 10. Using the Web Health Console .... 187
Logging on to the Web Health Console ....... 187
  Logging on to the Web Health Console for the
  first time ..................................... 188
Setting your preferences ........................ 188
  Managing endpoint lists ........................ 188
  Managing general preferences .................. 190
  Managing chart preferences .................... 190
Common Web Health Console features ......... 191
Using the endpoint list view .................. 192
  Working with endpoints ........................ 193
Using the resource model list view ............ 193
Using the endpoints by resource model view .. 194
  Working with endpoints ........................ 195
Working with the endpoint health view ....... 195
  Working with the resource models frame ....... 196
  Working with resource models ................. 196
  Indications frame ................................ 197
  Historical data frame ......................... 197
Working with chart views ..................... 198
  Graph controls .................................. 198
  Online data graph .............................. 199
  Historical data graph ......................... 200
Chapter 9. Introduction to the Web Health Console

This chapter provides an overview of the IBM Tivoli Monitoring Web Health Console (Web Health Console) and the information to install it.

The Web Health Console runs on Netscape 6.2 (or later) and Internet Explorer 6.x. You can use the Web Health Console for the following purposes:

- Checking, displaying, and analyzing the status and health of endpoints that have distributed resource monitors
- Displaying an endpoint’s real-time and historical data logged to the IBM Tivoli Monitoring database
- Viewing online and historical data on endpoints as a follow-up to specific problems
- Starting and stopping the IBM Tivoli Monitoring engine and individual resource models on selected endpoints
- Removing a profile from the selected endpoint

This chapter contains the following information:

- A conceptual overview of the Web Health Console
- An explanation of how health is determined for a resource
- Connection information for the Web Health Console

Overview

You can use the Web Health Console to check, display, and analyze the status and health of any endpoint with profiles and resource models. Status reflects the state of the endpoint displayed on the Web Health Console, such as running or stopped. Health is a numeric value determined by resource model settings. You can also use the Web Health Console to work with real-time or historical data from an endpoint that is logged to the IBM Tivoli Monitoring database.

You can use the diagnostic and monitoring capabilities of the Web Health Console to perform targeted analysis of problems associated with individual endpoints when an event is sent to the Tivoli Enterprise Console. Use the online and historical data to follow up specific problems with single endpoints.

Understanding resource health

The Web Health Console obtains events and indications from endpoints. The Web Health Console displays the health of each potential problem as a numeric value between 100 (perfect health) and zero (with zero meaning that the conditions for the corresponding event are met). Intermediate values show the percentage of occurrences currently registered with respect to the total number of occurrences needed to trigger an event. See Table 17.

Table 17. Health determination example

<table>
<thead>
<tr>
<th>Cycle</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU%</td>
<td>55</td>
<td>73</td>
<td>54</td>
<td>63</td>
<td>68</td>
</tr>
<tr>
<td>Occurrences or holes</td>
<td>H</td>
<td>O</td>
<td>H</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Occurrence count</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>
Table 17. Health determination example (continued)

<table>
<thead>
<tr>
<th>Health %</th>
<th>100</th>
<th>75</th>
<th>75</th>
<th>50</th>
<th>25</th>
</tr>
</thead>
</table>

Table 17 on page 175 gives the health percentage changes in steps of 25% because four occurrences were required to trigger an event. If the indication required five occurrences, the health percentage would have changed by steps of 20%. Resource health is determined at the indication level and passed up to the endpoint. The lowest health of any indication in a resource model is shown as the health of that resource model, and the lowest health of any resource model installed on an endpoint is shown as the health of that endpoint. For example, if one indication on one resource model that is installed on an endpoint has a health of zero, the health of the endpoint is shown as zero. The required occurrences, cycle times, thresholds, and parameters for indications are defined when the resource model is created in the IBM Tivoli Monitoring Resource Model Builder. For information about this, see Part 1, “Installing, configuring, and using,” on page 1.

Connecting the Web Health Console

You can connect the Web Health Console to any Tivoli management region server or managed node and configure it to monitor any or all of the endpoints that are found in that region (assuming that all gateways are interconnected).

To connect to the Web Health Console you need access to the server on which the Web Health Console server is installed and the IBM Tivoli Managed Region on which you want to monitor health. All user management and security is handled through the IBM Tivoli management environment. This includes creating users and passwords as well as assigning authority. For more information about how to log on to the Web Health Console, refer to “Logging on to the Web Health Console” on page 187.

The request manager processor

The Web Health Console uses the request manager processor of Tivoli Monitoring. The request manager collects, stores, and manages all endpoint requests created and used by Tivoli Monitoring components (such as the Web Health Console and the heartbeat) and applications.

The request manager acts like a down-call concentrator: it receives all requests (from the various Web Health Console monitors and from the heartbeat processor) and concentrates them into one request to the endpoint. Once the endpoint sends data back, the request manager stores the data in a centralized cache on the gateway. At that point, the Web Health Console monitors and the heartbeat can retrieve the data from the gateway cache.

The following figure shows how the request manager works:
To customize and use the request manager processor, see the corresponding parameters described in the `wdmconfig` command, Chapter 8, “Commands,” on page 109.

**System requirements**

The Web Health Console has the following system requirements:

- The target machine for installing the Web Health Console has minimum memory requirements of 384 MB, but 512 MB is recommended.
- The disk space requirements within the temp directory and the selected install directory for the single computer selected are:
  - 500 MB for the install directory during the installation
  - 230 MB for the install directory after the installation
  - 100 MB for the temporary directory

**Software requirements**

The Web Health Console runs on Netscape 6.2 (or later) and Internet Explorer 6.x and supports the following platforms:

- AIX 5.1
- HP-UX 11.0
- Red Hat Linux for Intel 7.1
- Solaris 2.8
- Windows 2000 Server and Advanced Server SP3
- Windows NT Server Version 4.0 SP6

### Installing the Web Health Console

The following sections describe how to install the Web Health Console on the different operating systems.

The Web Health Console can be installed on any system (Tivoli Management Region Server, Tivoli Managed Node/Gateway, or Endpoint) or on a system outside of the Tivoli Management Region.

The Web Health Console installation installs three software components:
- WebSphere® Application Server, Advanced Edition, Single Server, 4.0.2
- IBM HTTP Server
- Web Health Console

In order to install WebSphere Application Server, the Web Health Console installation invokes the WebSphere Application Server silent install.

The WebSphere Application Server code installed is a single server edition that is installed and configured for use by the Web Health Console. As such it does not provide a secured access to port 9090 (no password is required).

If you are upgrading an existing Web Health Console, it is necessary for the patch installation to stop the Web Health Console application server while performing these operations. This means that any user, logged into the Web Health Console when the installation is started, is unable to continue working until the server is restarted. Once the Web Health Console starts again, the user needs to reconnect to the Web Health Console server.

### Installing on Windows

To install the Web Health Console on Windows:

1. Insert Disk 2 in the CD-ROM drive and double-click the setupwin32.exe file.

   **Note:** If the temporary directory does not have enough space, you could get an error. To avoid this, enter the following command:

   ```
   setupwin32 -is:tempdir TMPdir
   ```

   where `TMPdir` is the name of the temporary directory.

2. Follow the directions presented in the install dialogs. In particular:
   - provide the Directory Name for the location on which you want to install the Web Health Console Server. The Directory Name must contain no spaces.
   - provide the User name under which the Web Server runs. This user must have the following Advanced User Rights: "Act as part of the operating system" and "Logon as a service".
   - provide the Password for the User.

To install in "silent" mode using the command line, you must provide the following arguments:

-silent
-P base_installinstallLocation="Directory Name"

-W user_input.user="UserName"

-W user_input.password="Password"

You can also include the above arguments into a file and pass the file to the launcher using the options switch:

-options options file

Installing on UNIX

To install the Web Health Console on UNIX:

1. Install Disk 1 in the CD-ROM drive and run one of the following files depending on the UNIX platform you are running:
   • setupaix.bin on AIX
   • setuphp11x.bin on HP-UX11.x
   • setupsolarisSparc.bin on Sun Solaris
   • setuplinux.bin on Linux

   **Note:** If the temporary directory does not have enough space, you could get an error. To avoid this, enter the command (corresponding to your UNIX platform):

   ```
   setupaix.bin -is:tempdir TMPdir
   ```

   where TMPdir is the name of the temporary directory.

2. Follow the directions presented in the install dialogs. In particular, provide the Directory Name for the location on which you want to install the Web Health Console Server.

To install in "silent" mode using the command line, you must provide the following arguments to the setup command indicated before:

-silent

-P base_installinstallLocation="Directory Name"

You can also include the above arguments into a file and pass the file to the launcher using the options switch:

-options options file

**Note:** The UNIX installation installs the IBM HTTP Server into a standard location regardless of the directory that you specify for the install. The rest of the installation goes to the directory that you have specified. The standard locations are:

- AIX: /usr/HTTPServer
- Sun Solaris: /opt/IBMHTTPD
- Linux: /opt/IBMHTTPServer
- HP: /opt/HTTPServer
Troubleshooting the installation

The Web Health Console installation installs three software components:
- WebSphere Application Server, Advanced Edition, Single Server, 4.0.2
- IBM HTTP Server
- Web Health Console

In order to install WebSphere Application Server, the Web Health Console installation invokes the WebSphere Application Server silent install. This means that, to debug the install, you may need to review several log files.

The primary log for the install is AMW_WAS_Install.log, which is stored in the temporary directory used by InstallShield. In addition, several other logs have useful information. The WebSphere Application Server install, which is executed by the Web Health Console, creates log files at the following location:
- on UNIX: /tmp/install.log
- on Windows:
  - during the install:
    - install_dir/setup.log: this is the IBM HTTP Server install log
  - after the WebSphere Application Server installation completes:
    - install_dir/logs/wssetup.log

If the installation is not successful, see the following troubleshooting information:
- Make sure that WebSphere Application Server is running:
  - Access http://hostname:9090/admin
  - If this loads the WebSphere Application Server Admin console, then WebSphere Application Server is running correctly
  - If this does not load the WebSphere Application Server Admin console:
    - Check the WebSphere Application Server install log previously indicated
    - If a prior installation of the Web Health Console failed before the WebSphere Application Server component install completed, the ITM_WHC_Uninstall.sh script cannot be used to uninstall the product. The subsequent reinstallation fails indicating that the Web Health Console is already installed. To correct this condition:
      - If Java 1.3 or later is installed on the machine, run:
        `install_dir/_uninst/java -jar uninstall -silent`
      - If you do not have Java 1.3 or later installed, run:
        `/install_temporary_directory/AMWInstallCheck.sh` where `install_temporary_directory` is the temporary directory specified during the failed installation attempt, otherwise `/tmp`. InstallCheck.sh outputs a list of packages that are blocking the install. Use the OS specific software maintenance tools to remove these packages. Also, remove these entries from the installer database, vpd.properties. This file is located as follows:
        - on Windows: %SYSTEMDIR%/vpd.properties
        - on Linux: $HOME/vpd.properties
        - on UNIX: /usr/lib/objrepos/vpd.properties
  - Make sure that the WebSphere Application Server application is running:
• **on Windows**: check the Services Panel. Make sure that the service "WebSphere Application Server V4 - WebSphere for ITM" is running. If not, attempt to start it using the controls in the Services Panel.

• **on UNIX**: check for the WebSphere Application Server process using the `ps` command on the java process under the WebSphere Application Server install directory. If WebSphere Application Server is not running, try to start it using `install_dir/bin/startServer.sh`

- Check the WebSphere Application Server run time logs at: `install_dir/logs`

• Make sure that the IBM HTTP Server is running:
  - Check the installation logs:
    - **on Windows**: look in `install_dir/setup.log`
    - **on UNIX**: look in `/tmp/install.log`. The IBM HTTP Server install occurs approximately half way through the install.log. The command line used to install IBM HTTP Server is in this log. To find it, look for "IHS install command". Copy that command and attempt to run it manually on the command line.

- Check to see if a previous installation of Apache or IHS has interfered on Windows 2000 and NT: if a previous installation has left an `httpd.conf` file or a shortcut to a non-existent `httpd.conf`, the installer reuses this file instead of writing a new one. Any files that the installer attempted to reuse is logged in `%TEMP%/IHSInst.err` and must be removed before reattempting the installation.

  - Check if the process is running:
    - **on Windows**: check the Services Panel. Make sure that the service "IBM HTTP Server" is running. If not, attempt to start it using the controls in the Services Panel. If the service does not start, try to start the Server from the command line by running: `install_dir/HTTP_Server/Apache`

  • If the Server runs on the command line but not in a service, check the parameters of the service:
    - Ensure the Login and Password are correct.
    - Ensure the path is correct.
    - If you are using Windows NT 4.0, attempt the following:
      1. Run `regedit`.
      2. Go to:

         ```
         HKEY_LOCAL_MACHINE/System/CurrentControlSet/Services/
         IBMHTTPService/Parameters
         ```

      3. Set the value of `ConfigArgs` to be blank.

**Note**: Editing the Windows registry can corrupt the Windows operating system. Attempt the above instructions only if you are knowledgeable about the Windows Registry.

• If you see errors referring to the Security Library, check the following:
  - Ensure that the IBM Key Management Utility is installed. If it is installed, it shows in `Program Files/ibm/gsk5` on the drive where you installed the Web Health Console.

  - Make sure that the environment variable `PATH` contains the path to the IBM Key Management Utility. For example: `PATH=c:\Program Files\ibm\gsk5\lib`.

- **on UNIX**: Check for the IBM HTTP Server process using the `ps` command on the http process. If IBM HTTP Server is not running, try to start it using
IBM HTTP Server install_dir/bin/apachectl start. If the IBM HTTP Server does not start, look at the log files in IBM HTTP Server install_dir/logs.

- Check that the Web Health Console application is installed correctly:
  - If both WebSphere Application Server and IBM HTTP Server are running correctly but the Web Health Console is still not available, then the Web Health Console is not functioning. At this point do the following:
    1. Launch the WebSphere Application Server Admin Console:
       http://hostname:9090/admin
    2. Go to Nodes → yourNodeName → Enterprise Applications.
       The application "ITM Web Health Console" should be present and it should be running.
       - If the application is present but it is not running, try to start it, by selecting its entry on the WebSphere Application Server Admin Console. If it does not start, check the following:
         • Look in the WebSphere logs at install_dir/logs.
         • Look in the Web Health Console logs at:
           c:\Tivoli\AMW\logs\traceDMWebConsole.log or
           /opt/Tivoli/AMW/logs\traceDMWebConsole.log
       - If the application is not present, check the following:
         • Check (for WebSphere Application Server) install_dir/healthConsole.
           It should contain the file dm.ear. If the file is not there, then the installation has failed completely. Check the AMW_WAS_Install.log indicated before.
         • Check install_dir/installedApps. It should contain a directory named dm.ear. If it does not contain that directory, then the configuration command failed to run:
           • Check the AMW_WAS_Install.log and install_dir/appInstall.out or
             install_dir/appInstall.err for details.
           • Attempt to run the following command manually:
             install_dir/bin/SEAppInstall -install
             install_dir/healthConsole/dm.ear -interactive false
         • Check install_dir/config/server-cfg.xml. It should contain a reference to the Web Health Console (that you can find by expanding first the nodes tag and then the installedApps tags until an element DM Web Health Console is found). If it does not contain a reference, but the previous step worked, you probably overwrote the file when you logged into the Admin Console. To correct this, stop the WebSphere Application Server by running:
             install_dir/bin/stopServer[.sh] and execute again the first step in this ordered list.
         • Check the install_dir/config/plugin-cfg.xml. It should contain a reference to the string "dmwhc". If it does not contain a reference, then GenPluginCfg command failed. Look in install_dir/appInstall12.out or install_dir/appInstall12.err for details.
           Run the following command manually:
           install_dir/bin/GenPlugIn -configFile install_dir/config/server-cfg.xml
         • Restart the server, if you stopped it, by running:
           install_dir/bin/startServer[.sh]
Enabling the Web Health Console to use your resource models

To enable the Web Health Console to use any resource model that you created with Tivoli Monitoring Resource Model Builder, perform the following steps.

**Note:** To follow this procedure, you must have Java Development Kit (JDK) 1.3 installed on the system on which you need to compile the resource model (usually the system where Tivoli Monitoring Resource Model Builder resides).

1. In the main window of Tivoli Monitoring Resource Model Builder, from the Build menu select Export Java Message Catalog.
2. Save the java file with the internal name of the resource model (for example, DMXCPU.java).
3. From a command prompt, compile the java file by entering the following command:
   - On Windows
     ```
     JDKdir\bin\javac filename.java
     ```
   - On UNIX or Linux
     ```
     JDKdir/bin/javac filename.java
     ```
     Where:
     - **JDKdir** The fully qualified path where JDK 1.3 is installed.
     - **filename** The name of the java file.
     A class file with the same name of the java file is created.
4. Copy the class file to the Web Health Console application server at:
   ```
   install_dir/installedApps/dm.ear/dm.war/WEB-INF/classes/com/tivoli/DmForNT/resources
   ```

Enabling Secure Socket Layer support

The Web Health Console is preconfigured for Secure Socket Layer (SSL) communication. Before SSL can be used, however, a key database and site certificate must be generated and the SSL preconfiguration enabled.

This section shows you the steps to enable SSL.

**Note:** In the following directions, replace IBM HTTP Server Location with:
- AIX: /usr/HTTPServer
- Sun Solaris: /opt/IBMHTTPD
- Linux: /opt/IBMHTTPServer
- HP: /opt/HTTPServer
- UNIX: /opt/HTTPServer
- Windows: Web Health Console Server Location/HTTP_Server

Follow these steps to enable SSL:

1. Create a directory on the server to hold your SSL key database files and certificates. In the following steps, this directory is referred to as key_db_dir.
2. **For Windows NT:** start the IBM Key Management utility by invoking `gsk5ikm.exe` from `web_health_console_install_drive\Program Files\ibm\gsk5\bin` or by selecting **Start>Programs>IBM HTTP Server>Start Key Management Utility.**
For AIX, Sun Solaris, HP/UX, and Linux: start the IBM Key Management utility by invoking gsk5ikm from:

- AIX: /usr/ibm/gsk5/bin
- Linux: /usr/local/ibm/gsk5/bin

To enable SSL on Linux, you must enable the use of libstdc++-6.1-2 by downloading it, if necessary, and running the following command:

```bash
export LD_PRELOAD=/usr/lib/libstdc++-libc6.1-2.so.3
```

- Sun Solaris: /opt/ibm/gsk5/bin
- HP/UX: /opt/ibm/gsk5/bin

3. Within IBM Key Management, select Key Database File>New.

4. Complete the fields in the New dialog box, making the following selection entries:

- **Key Database Type**: Select the CMS Key database file
- **File Name**: Enter a base filename for the new key database files. Several files are created using this base filename, including:
  - `key_database_name.kdb`, `key_database_name.rdb`,
  - `key_database_name.sth`, and `key_database_name.crl`.

To use the default Web Health Console configuration, use IBM HTTP SERVER LOCATION/key_db for the Location.

5. Click OK.

6. Complete the fields in the Password Prompt dialog box, making the following selection and entries:

- Enter the password which will be used to encrypt and decrypt the key database
- Reenter the password to confirm that it is typed correctly
- Select Stash the password to a file

7. If you have a server certificate from a Certificate Authority (for example, Verisign), you can click Import to import this certificate into your SSL key file. If not, create a new one by selecting Create>New Self-Signed Certificate.

8. Complete the fields in the Create New Self-Signed Certificate dialog box, making the following selection and entries:

- **Key Label**: A label by which the key and certificate in the database will be identified.
- **Common Name**: The fully qualified hostname of the server on which the Web Health Console is installed.
- **Organization**: Enter an organization name. This information is presented to the client as part of the credentials of the server.

Leave the remaining parameters at their default value, and click OK.

9. Using a text editor, open the file IBM HTTP SERVER LOCATION/conf/httpd.conf and modify the lines described in the detailed platform instructions below. If you did not use the default location and name for the key database, insert the appropriate information.

**For Windows NT**: Uncomment the lines

```plaintext
#LoadModule ibm_ssl_module modules/IBMMODULESSL128.dll
#Listen 443
#SSLEnable
#Keyfile "IBM HTTP SERVER LOCATION/key_db/key.db"
```

**For AIX, Sun Solaris, and Linux**: Uncomment the lines
#LoadModule ibm_ssl_module libexec/mod_ibm_ssl_128.so
#Listen 443
#SSLEnable
#Keyfile "IBM HTTP SERVER LOCATION/key_db/key.db"

For HP:
Comment the Port and Listen lines for port 80:
Port 80
Listen 80
Uncomment the lines:
#LoadModule ibm_ssl_module libexec/mod_ibm_ssl_128.so
#Listen 443
#SSLEnable
#Keyfile "IBM HTTP SERVER LOCATION/key_db/key.db"

By convention, SSL communication is run on port 443 and the Web Health Console comes preconfigured accordingly. To change the SSL port to a non-standard port, change the "Listen" directive in httpd.conf to reflect the preferred port.

10. **On Windows:**
Restart the IBM HTTP Server by selecting **IBM HTTP Server** in the Windows NT Services control panel, clicking **Stop**, and then clicking **Start**

**On UNIX:**
Restart the IBM HTTP Server by running **IBM HTTP SERVER LOCATION/bin/apachectl restart**.

The IBM HTTP Web Server is now configured for SSL.

---

**Uninstalling the Web Health Console**

To uninstall the Web Health Console, do the following:

1. Run the uninstall procedure:
   - **on Windows**: on the Control Panel, select Add/Remove Programs and remove IBM Tivoli Monitoring 5.1.1 Web Health Console.
   - **on UNIX**: `install_dir/ITM_WHC_Uninstall.sh`

2. Clean up the remaining files:
   - The uninstall procedure does not remove the directories and some of the files that were created during the Web Health Console installation. This is done to avoid deleting configuration information that you may want to save. To complete the uninstall, remove the following directories after ensuring that they do not contain any critical data:
     - `install_dir`
     - `IBM_HTTP_Server_install_dir` (refer to the installation procedure to see the directory name specific for each platform)
     - `log_dir`:
       - **on Windows**: `log_dir` is `c:\Tivoli\AMW\logs`.
       - **on UNIX**: `log_dir` is `/opt/Tivoli/AMW/logs`. 
Chapter 10. Using the Web Health Console

This chapter describes how to use the Web Health Console, navigate the components, and interpret component contents.

You can use the Web Health Console to display real-time or historical data about resources on a specified system. The Web Health Console displays 24 hours of historical data.

The data in the Web Health Console is automatically refreshed every 5 minutes, by default. You can change the refresh rate of the Web Health Console on the General tab of the Preferences view. See "Managing general preferences" on page 190 for more information.

Note: To use the Web Health Console you need the following authorizations:
- To access the Web Health Console you need one of these roles (on the Tivoli region): itm_whc_user, user, super, admin, or senior.
- To view an endpoint you need one of the following roles (on the endpoint): user, super, admin, or senior.
- To work with an endpoint you need one of the following roles (on the endpoint): super, admin, or senior.

Logging on to the Web Health Console

Perform the following steps to log onto the Web Health Console:

1. Connect to http://server_name/dmwhc, where server_name is the server on which you installed the Web Health Console server.

2. Supply the following information to display the Login View:
   - User: Tivoli user ID
   - Password: The Password associated with the Tivoli user ID
Host  The Name of the managed node to which you want to connect

3. Click OK to connect to the Web Health Console.

A message is displayed above the text entry fields if the login was unsuccessful, or you were redirected to this view because of a different error.

Logging on to the Web Health Console for the first time
Use the following steps if you are logging on to the Web Health Console for the first time:

1. Log on to the Web Health Console.
   The first time you log on to the Web Health Console, the Preferences view is displayed. You must populate the Selected Endpoint list before you can access any other Web Health Console views. When you log on subsequently, the endpoint list is automatically loaded.

2. Select the endpoints that you want to see. See “Managing endpoint lists” for more information.

3. Customize the Web Health Console to your specification. See “Managing general preferences” on page 190 and “Managing chart preferences” on page 190 for more information.

Setting your preferences
You can customize the Web Health Console views to display your preferred settings. This section explains how to manage the following preferences:

- “Managing endpoint lists” on page 188
- “Managing general preferences” on page 190
- “Managing chart preferences” on page 190

Managing endpoint lists
To maximize performance you should limit the number of endpoints that the Web Health Console monitors. Use Preferences view to specify only those endpoints in
which you are interested.

The first time you log into the Web Health Console the **Preferences** view is displayed. You must populate the **Selected Endpoint** list before you can access any other Web Health Console views. When you log on subsequently, the endpoint list is automatically loaded.

To populate the endpoint list:

1. Type the name of an endpoint filter in the **Filter** field.
   For example, type abc* to display all endpoints in the Tivoli managed region with names that start with abc and that have resource models installed. The filtering capability support is limited to the (*) regular expression character.

2. Click **Go**.

3. Click on the endpoint to select it from the **Available endpoints** list.
   Use Control-click or Shift-click to select multiple endpoints.

4. Click **Add >>**.
   Repeat these steps to add more endpoints to the list.

   **Note:** You can apply a new filter at any time to update the **Available endpoints** list without affecting the **Selected endpoints**. You can remove endpoints from the **Selected endpoints** list by using the **<< Remove** button.

5. Click **Save** to save the **Selected endpoint** list or **Cancel** to cancel these changes.

After you populate your views, you can configure other preferences. See “**Managing general preferences**” on page 190 and “**Managing chart preferences**” on page 190.
Managing general preferences

Perform the following steps to manage the General Preferences view:

1. Log on to display the Preferences view.
2. Click the General tab.
3. Select and define the following:
   - **Refresh Interval**
     The number of minutes between automatic updates of the data views. The default interval is 5 minutes. The interval you specify here does not control the rate at which the chart views refresh. Refer to “Managing chart preferences” on page 190 for more information.
   - **Default view**
     The view you want displayed whenever you log in to the Web Health Console. The default is the Endpoint View.
4. Click Save to implement changes you make to the Preferences, General view.

Managing chart preferences

Use the Chart page of the Preferences view to select the type of chart that is displayed when you click Graph on the Endpoint view, the refresh interval for
Perform the following steps to manage the **Preferences, Chart** view:

1. Log on to display the **Preferences** view.
2. Click the **Chart** tab.
3. Select a chart from the **View Type** drop-down list.
4. Select the refresh interval from the **Online chart refresh interval** drop-down list.
5. The color series represents the order in which colors appear in all of the charts. To change a color:
   a. Click the color you want to change to display a color palette containing all of the possible choices.
   b. Specify the color with which you want to replace the selected color.
   c. Click **Save**.
6. Click **Save** to implement the changes.

---

**Common Web Health Console features**

The top portion of every view of the Web Health Console, except the **Login** view, contains a common menu banner.

This menu banner contains the following information:

**Table 18. Navigation icons**

| User: Administrator | The user name with which you logged onto this session of the Web Health Console |

---
Table 18. Navigation icons (continued)

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![refresh]</td>
<td>The refresh icon button to manually refresh the current view</td>
</tr>
<tr>
<td>![home]</td>
<td>The home icon button to return to the view that is set on the General tab of the Preferences view. The options are the Endpoint List View or the Resource Model List View</td>
</tr>
<tr>
<td>![endpoint list]</td>
<td>The endpoint list icon button to open the Endpoint List View</td>
</tr>
<tr>
<td>![resource model]</td>
<td>The resource model icon button to open the Resource Model List View</td>
</tr>
<tr>
<td>![view/edit]</td>
<td>The view/edit preferences icon button to open the Preferences view</td>
</tr>
<tr>
<td>![sign off]</td>
<td>The sign off button to return to the Login view</td>
</tr>
</tbody>
</table>

The common navigation bar has fly-over help associated with each button.

As you drill down through the views, the Web Health Console displays a dynamic path in the title portion of the window. This path shows the views you went through to arrive at the currently displayed view. It also provides links to any of the previous views. You can return to a previous view by clicking on the title of that view.

Health is displayed in the views as an exact health percentage and as an iconic representation of possible states of alert. See Table 19 for a description of the health alert icons:

Table 19. Health alert icons

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![health]</td>
<td>The health of all of the resource models installed on the endpoint is at 70-100%.</td>
</tr>
<tr>
<td>![health low]</td>
<td>The health of at least one of the resource models installed on the endpoint is less than 70% but greater than 10%.</td>
</tr>
<tr>
<td>![health low 0-10]</td>
<td>The health of at least one of the resource models installed on the endpoint is at 0-10%.</td>
</tr>
<tr>
<td>![health unavailable]</td>
<td>The health of at least one of the resource models installed on the endpoint is unavailable.</td>
</tr>
</tbody>
</table>

Using the endpoint list view

The Endpoint List View shows the current health of all the endpoints specified in the Endpoints tab of the Preferences view. (See “Managing endpoint lists” on page 188 for more information.) The Web Health Console sorts the endpoints by health order, with the lowest health displayed first. For example, an endpoint with a health percentage of 20 is listed before an endpoint with a health percentage of 80. If there is a problem contacting an endpoint, the Web Health Console displays a message indicating the problem.  

1. To display the health of a specific endpoint, or group of endpoints, type the appropriate information in the Filter field and click Submit.
For example, type abc* to display all endpoints in the Selected endpoints list with names that start with "abc" and that have resource models installed. The filtering capability support is limited to the (*) regular expression character.

The **Endpoint List View** table provides information on the following:

**Status**  
The status of the specific endpoint being monitored.

**Health**  
The lowest health of all the resource models installed on the endpoint. For example, if the endpoint has two resource models installed and one is at 20% health and the other is at 80% health, this column displays 20. The health is displayed as an exact health percentage and as an iconic representation of possible states of alert.

2. Click the endpoint name to display the **Endpoint** view for detailed information about each resource model running on that endpoint. Refer to "Working with the endpoint health view" on page 195 for detailed information.

—OR—

Click the radio button to select an endpoint and take action on that endpoint. See "Working with endpoints."

### Working with endpoints

**Note:** These operations require the super, admin, or senior role.

To start or stop the IBM Tivoli Monitoring engine from the **Endpoint List View**:
1. Click the radio button to select the endpoint.
2. Select **Start Engine** or **Stop Engine** from the **Select an action** drop-down list.

### Using the resource model list view

The **Resource Model List View** shows all of the resource models installed on the endpoints specified in the **Endpoint List** page of the **Preferences** view. (See "Managing endpoint lists" on page 188 for more information.) The Web Health Console sorts the resource models by health order, with the lowest health displayed first. For example, a resource model with a health percentage of 20 is...
listed before a resource model with a health percentage of 80.

The Resource Model List View table provides information on:

**Status**  The status of the resource model that is being displayed.

**Health**  The lowest health of that resource model out of all the endpoints. For example, if the resource model is installed on two endpoints, and the resource model has a health percentage of 20 on one endpoint, and a health percentage of 80 on the other endpoint, then this column displays 20. Health is displayed as an exact percentage and as an iconic representation of possible states of alert. See Table 19 on page 192 for information on the icons.

Click a resource model name to display the Endpoint by Resource Model List View. See “Using the endpoints by resource model view” for more information.

**Using the endpoints by resource model view**

This view can be displayed only by clicking a resource model name in the Resource Model List View. Use the navigation bar at the top of this view to return to the Resource Model List View.
This view shows all of the endpoints specified in the Endpoint List page of the Preferences view that have the selected resource model installed. See "Managing endpoint lists" on page 188 for more information.

To display the health of the resource model on a specific endpoint, or group of endpoints, type the appropriate information in the Filter field and click Submit.

For example, type abc* to display all endpoints in the Selected endpoints list with names that start with "abc" and that have resource models installed. The filtering capability support is limited to the (*) regular expression character.

**Working with endpoints**

*Note:* These operations require the super, admin, or senior role.

You can start or stop the IBM Tivoli Monitoring engine from the Endpoint List View:

1. Click the radio button to select the endpoint.
2. Select **Start Engine** or **Stop Engine** from the **Select an action** drop-down list.

---

**Working with the endpoint health view**

The **Endpoint Health** view is the most detailed view of the health of an endpoint. In this view, the following information is displayed:

- The health and status of all resource models installed on the endpoint
- The health of the indications that make up the resource model and historical data

The view is divided into three frames. The upper frame contains the common Web Health Console information. (See "Common Web Health Console features" on page 191). The middle frame (Resource Models frame) displays a table of the resource models installed on this endpoint listed according to lowest health. The lower frame displays a table of the indications for the selected resource model (Indications frame) or the historical data controls for the selected resource model.
(Historical Data frame).

![Image](image.png)

**Working with the resource models frame**

The middle frame of the **Endpoint Health** view displays a table that includes the following information:

- **Name**  
  Resource model name

- **Profile**  
  Name of the profile in which the resource model resides

- **Status**  
  Status of the resource model

- **Health**  
  Health percentage of the resource model. The resource model health is the lowest health of any indication in the resource model. The Web Health Console sorts the endpoints by health order, with the lowest health displayed first.

Select **Indications** or **Historical Data** to determine the information displayed in the lower frame for the selected resource model. See “Indications frame” on page 197 and “Historical data frame” on page 197 for more information.

**Working with resource models**

**Note:** These operations require the super, admin, or senior role.

From the **Endpoint Health** view you can perform the following actions on resource models:

- **Start Resource Model**  
  To start the resource model selected.

- **Stop Resource Model**  
  To stop the resource model selected.

- **Start all Resource Models of the Same Profile**  
  To start all the resource models belonging to the profile selected.
Stop all Resource Models of the Same Profile
To stop all the resource models belonging to the profile selected.

Remove Profile
To delete the profile selected from the endpoint.

Indications frame
This frame of the Endpoint Health view displays a table listing the indications for the selected resource model sorted by health.

From this table you can launch an online data graph for any indication with a health less than 100%. Select the desired indication and click Graph. This launches an online data graph in a separate window. See “Online data graph” on page 199 for more information.

Historical data frame
Using the historical data selection controls, you select instances and metrics from the selected resource model to create a chart of recent historical data from the endpoint. To create historical data, logging must be enabled for the resource model. With logged data, you can use the Historical Data Graph to identify specific instances of resource problems over the past one, six, twelve, or twenty-four hours.

To create a Historical Data Graph from the Endpoint Health view:
1. Select a resource model from the Resource Model list in the middle frame.
2. Click the Historical Data radio button to display the Historical Data selection information in the lower frame.
3. Click the Resource drop-down list and select a resource. The Resource drop-down list is the only option that is active when the frame opens.
4. Click the Contexts drop-down list and select a context. Each context identifies a logical grouping of problems related to the specified resource.
5. Select one or more instance from the Instances list. These identify specific instances of the selected indication.
6. Select one or more metric from the Metrics list. These are the metrics used to measure the selected indication.
7. Click Graph.
For detailed description of the Historical Graph, see “Historical data graph” on page 200.

---

**Working with chart views**

The following sections provide you with a description of the chart views used by the Web Health Console.

**Graph controls**

- **Select a Graph Format**
  Allows you to change the style of graph displayed for the current data. Use the Table format to display non-numeric metrics. To change the default graph format, use the View Type preference in the Graph View tab of the Preferences view.

- **Show History View of**
  Allows you to define the time period for the history view. Select the appropriate time period from the drop-down list.

- **Zoom**
  For non-tabular formats, the graph image x-axis is divided into ten regions of equal size. To zoom in on a graph, click the area to be magnified and the graph is redrawn, narrowing the visual focus to the selected region. This means that you do not see new data points being added if they are outside of the zoomed area.

- **Zoom Out**
  The Zoom Out button reduces the magnification up to a maximum of 100%. For example, if you have zoomed in three times, clicking Zoom Out three times returns you to the original graph magnification.

- **Close**
  Use the Close button positioned above the graph image to ensure that all server-side resources associated with the graph are freed in a timely fashion. Using the Close control in the window title bar has the same effect, however, the resources take longer to reclaim.
Online data graph

The Online Data Graph displays the current value of the metrics that make up the indication on a given instance of a resource; for example, Processor 0 on a single processor system. One data point is shown for each metric when the graph is launched.

As long as the health remains below 100 percent, data is added at each refresh interval to the right side of the graph or the bottom of the table. Data older than 24 hours falls off the left side of the graph or the top of the table. When the health of the instance returns to 100, the graph displays a text message indicating this condition. If, later, the health drops below 100 percent, new data points are added to the right side of the graph.
Historical data graph

The **Historical Graph** displays recent historical data logged by a resource model on an endpoint. Metrics logged for multiple instances of a resource can be plotted on the same graph. The metrics displayed on the graph are specified in the **Historical Data** frame of the **Endpoint Health View**. See "Historical data frame" on page 197 for more information.

With logged data, you can use the **Historical Graph** to identify specific instances of resource problems over the past one, six, twelve, or twenty-four hours. The default history view displays the last hour of data. You can change the time period from the **Show History View of** drop-down list. Use Tivoli Data Warehouse to view data older than 24 hours.
Part 3. Appendixes
Appendix A. Environment variables

This appendix describes the environment variables used by Tivoli Monitoring.

The following table shows a list of environment variables set by events (implemented through tasks, program responses, and Common Information Model (CIM) methods).

<table>
<thead>
<tr>
<th>Variable</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>EVENT_NAME</td>
<td>The Aggregated Event Name</td>
</tr>
<tr>
<td>EVENT_SEVERITY</td>
<td>The event severity.</td>
</tr>
<tr>
<td>TMW_EVENT</td>
<td>Same as EVENT_NAME. It is deprecated and remains for backward compatibility only.</td>
</tr>
<tr>
<td>TMW_PROFILE</td>
<td>Same as PROFILE. It is deprecated and remains for backward compatibility only.</td>
</tr>
<tr>
<td>$(EVENT_NAME)_Propertyxxx</td>
<td>Local event attributes.</td>
</tr>
<tr>
<td>$(EVENT_NAME)_alreadysent</td>
<td>Boolean value that indicates if the specified event had already been sent.</td>
</tr>
<tr>
<td>$(EVENT_NAME)_event_key</td>
<td>A key property of the specified event.</td>
</tr>
<tr>
<td>$(EVENT_NAME)_eventid</td>
<td>The identifier of the specified event.</td>
</tr>
<tr>
<td>$(EVENT_NAME)_severity</td>
<td>The severity of the specified event.</td>
</tr>
</tbody>
</table>

The following table shows the list of environment variables that can be set through tasks, program responses, CIM methods, custom script or monitoring collections started by resource model scripts.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADMIN</td>
<td>The name of the administrator responsible for the most recent distribution of the profile containing the monitor.</td>
</tr>
<tr>
<td>ENDPOINT</td>
<td>The name of the endpoint object through which the profile containing the monitor was distributed.</td>
</tr>
<tr>
<td>ENDPOINT_CLASS</td>
<td>The Tivoli class (or &quot;type&quot;) name of the endpoint object.</td>
</tr>
<tr>
<td>ENDPOINT_OID</td>
<td>The Tivoli object ID of the endpoint object.</td>
</tr>
<tr>
<td>HOST</td>
<td>The Hostname of the endpoint to which the resource model was distributed.</td>
</tr>
<tr>
<td>ITM_ENDPOINT</td>
<td>The endpoint label.</td>
</tr>
<tr>
<td>OPERATOR</td>
<td>The identity through which the administrator gained access to the Tivoli environment.</td>
</tr>
<tr>
<td>PROFILE</td>
<td>Full name of the profile.</td>
</tr>
<tr>
<td>Variable</td>
<td>Definition</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PROFILEOID</td>
<td>The Tivoli object ID of the profile that contains the model.</td>
</tr>
<tr>
<td>RESOURCE_MODEL</td>
<td>Resource Model internal name.</td>
</tr>
</tbody>
</table>
Appendix B. Installing using the installation wizard

This appendix explains how to install Tivoli Monitoring using the installation wizard (hereafter also referred to as the installer).

The installer creates a complete Tivoli environment, if none exists, and installs the product and, if needed, install also supplemental software. Use the installer if you have no previous Tivoli software installations.

If you use the installer, you must choose between an Evaluation or Typical installation method. Use the following criteria to determine which method you want to follow:

*Use the Evaluation installation method* if you want to install a simple Tivoli environment and Tivoli Monitoring on one computer so that you can test and evaluate the product. This installation method is not intended for use in a production environment.

*Use the Typical installation method when the following conditions are true:*

- You want to install the product on a target computer that has no software installed for the Tivoli Management Framework.
- You are a first-time Tivoli user.

This appendix describes both installation methods.

It is recommended that you close all other applications before an installation.

A proxy agent must exist in a Windows domain to enable the installer to create Windows endpoints. When you are installing a Tivoli server on a Windows operating systems, the installer automatically creates the Windows proxy agent so that you can create Windows endpoints in the domain of the Tivoli server. When you are installing the Tivoli server on a UNIX-based operating system and you want to create Windows endpoints, refer to section "Creating a Windows proxy endpoint" on page 213.

If you get problems using the installer, refer to the IBM Tivoli Monitoring: Problem Determination Guide for troubleshooting information.

Performing an Evaluation installation

When you use the Evaluation installation method you can install a simple Tivoli environment and Tivoli Monitoring on one computer so that you can test and evaluate the product.

The Evaluation installation works only if the target computer does not have any Tivoli software installed. During an Evaluation installation, the installer creates one local endpoint on the computer. No software is installed on any remote computers.

When you perform an Evaluation installation, the following Tivoli software is installed:

- Tivoli Management Framework, Version 4.1.1
- IBM Tivoli Monitoring, Version 5.1.2
IBM Tivoli Monitoring Web Health Console, Version 5.1.1 (plus Fix Pack 6)

Refer to Chapter 9, “Introduction to the Web Health Console,” on page 175 for details on the system and software requirements of the Web Health Console.

The required authorization to use this procedure is:

- **on Windows**: the user who installs the product must have membership in the Administrators group
- **on UNIX**: the user who installs the product must have root privileges

It is recommended that you close all other applications before an installation.

To perform an Evaluation installation, follow these steps:

1. Log on to the computer where you want to install an evaluation version of the product.
2. Run the setup executable in the root directory of the installation CD labeled IBM Tivoli Monitoring Version 5.1.2: Installation to launch the installer:
   - **on Windows**: run setupITM.bat
   - **on UNIX**: run setupITM.sh
3. Click Next to display the license agreement. Accept the terms of the license agreement and click Next again.
4. The installer may display a warning about DHCP. Click Next.
5. Specify the directory path where you want to install the product. Click Next.
6. Specify the directory path where you want to create a depot (temporary directory) for product installation. The depot directory contains all files required for the installation. If you click Cancel in an installer panel, this action always deletes the temporary depot directory. You can begin installation again by running the setup executable on the installation CD. After the installation is complete, the installer provides an option to automatically delete the depot directory.

   **Note**: On Windows, the depot must reside on the target computer. You cannot perform an installation of the product on Windows using a depot that is located on a remote system. On UNIX, provide a directory path that is accessible from the installation computer to the installation depot.
7. Click Next to copy the files to the installation directories.
8. Click Next when the installer prompts you to restart the application and continue the installation. The installation options window is displayed.

9. Select the Evaluation/Product Demonstration option and click Next to display the User Password window.

10. Type the password corresponding to the displayed User ID in the Password and Verify Password fields.

11. Click Next to display the language option window.

12. Select the check box next to any language you want to install.

13. Click Next to display the first of a series of windows, each of which prompts you for the directory location of a specified CD or CD image that contains the software to be installed.

14. For each CD prompt window that is displayed, perform the following steps:
   a. Insert the specified CD into the drive.
   b. Type a directory path or click Browse to specify the path to the CD or CD image. Click Next.
   c. Click Yes in response to the message that asks if you want to copy the files to the depot directory.

15. Click Next on the final CD prompt window to display the Step List window. When the Step List window is displayed, the installer begins installation of the software from the depot to the install location. The Step List shows the components that the installer installs, and it displays status updates for the installation process. See the IBM Tivoli Monitoring: Problem Determination Guide for a descriptions of the status icons and options in the Step List window.

If you want to cancel the installation while the Step List is running, perform the following steps:
   a. Click Stop and allow the currently running process to finish. If you click Cancel without allowing the currently running process to complete, the system might remain in an unpredictable state.
   b. Click Cancel to stop the installation and close the installer.
c. Delete the temporary depot directory created by the installer.

16. *(Optional)* Double-click any item in the Step List to see details and to set processing instructions for that step in the Details window.

17. Click OK in the final prompt of the Step List.

This final prompt is displayed when all steps in the Step List show a Passed status. If all steps do not show a Passed status, see the IBM Tivoli Monitoring: Problem Determination Guide for troubleshooting information. Click Next.

18. Select **Delete the temporary files** to delete the directory of temporary installation files that you specified at the beginning of the installation process.

19. Click Next and then click Finish.

### Performing a Typical installation

When you use the Typical installation method with predefined default values you automatically create a Tivoli environment and install Tivoli Monitoring. The installer optionally creates managed resources in the Tivoli environment, as described in this procedure.

When you perform a Typical installation, the following Tivoli software is installed:

- IBM Tivoli Monitoring, Version 5.1.2
- *(Optional)* IBM Tivoli Monitoring Web Health Console, Version 5.1.1 (plus Fix Pack 6)

During installation, you can choose whether or not to install the Web Health Console on target computers that you specify. Refer to Chapter 9, “Introduction to the Web Health Console,” on page 175 for details on the system and software requirements of the Web Health Console.

During the installation procedure, you can optionally create managed nodes (other than the Tivoli server), endpoints, and Web Health Consoles.

To create Windows endpoints using the installer, at least one Windows endpoint must exist in the domain of the Tivoli server (the computer from which you start the installer). If you are installing from a Windows system, and no Windows endpoints exist, the installer automatically creates a Windows proxy endpoint, which enables you to create additional Windows endpoints. If you are installing from a UNIX-based system, you must create the Windows proxy endpoint by following a separate procedure. Table 22 summarizes what you need to do under various conditions.

<table>
<thead>
<tr>
<th>OS at the Tivoli server</th>
<th>Do Windows endpoints already exist in the domain of the Tivoli server?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Yes</strong></td>
</tr>
<tr>
<td>Windows</td>
<td>Complete this procedure (installing the product using the Typical option). The installer automatically creates one Windows proxy endpoint, which enables you to create Windows endpoints using the installer.</td>
</tr>
</tbody>
</table>
To perform a Typical installation, follow these steps:

**Note:** In the following steps, the term *Tivoli server* refers to a managed node. In addition, the term *a computer to monitor* refers to an endpoint.

1. Log on to the computer that you want to be the Tivoli server.
2. Run the setup executable in the root directory of the installation CD labeled *IBM Tivoli Monitoring Version 5.2: Installation* to launch the installer:
   - **on Windows:** run setupITM.bat
   - **on UNIX:** run setupITM.sh
3. Click **Next** to display the license agreement. Accept the terms of the license agreement and click **Next** again.
4. The installer may display a warning about DHCP. Click **Next**.
5. Specify the directory path where you want to install the product. Click **Next**.
6. Specify the directory path where you want to create a depot (temporary directory) for product installation. The depot directory contains all files required for the installation. If you click **Cancel** in an installer panel, this action always deletes the temporary depot directory. You can begin installation again by running the setup executable on the installation CD. After the installation is complete, the installer provides an option to automatically delete the depot directory.

**Note:** On Windows, the depot must reside on the target computer. You cannot perform an installation of the product on Windows using a depot that
is located on a remote system. On UNIX, provide a directory path that
is accessible from the installation computer to the installation depot.

7. Click Next to copy the files to the installation directories.
8. Click Next when the installer prompts you to restart the application and
continue the installation. The installation options window is displayed.

9. Select Typical and click Next.
10. (Optional) Select the languages that you want to enable for users of the
product. English is always enabled for this product. The installer installs the
language enablement packages for additional languages that you choose.
11. Click Next. A list of existing managed nodes is displayed. At least one
managed node is displayed in the list. This managed node is the Tivoli server
(referred to on the installation windows as the principal Tivoli server). The
installer creates the Tivoli server on the computer on which you are installing
this product. The installer also discovers and lists any managed nodes that
were created in a previous installation of Tivoli Management Framework.

12. (Optional) Click Add to create additional managed nodes.
   • If you are creating a Windows managed node, the target computer must
     have the Tivoli Remote Execution Service (TRIP) process running. If you are
     creating a UNIX managed node, the target computer must have the `rexecd`
daemon running.
   • After adding a managed node to the list, you can use the Edit and Remove
     buttons to edit it or remove it from the list. You can edit or remove only the
     managed nodes that you added. You cannot edit or remove any of the
     managed nodes that were originally displayed in the list.
When you click OK, you save settings for both the Basic tab and the Advanced tab.

13. Follow these steps to add information for the managed node:
   
a. Specify the values for the managed node in the fields of the Basic tab:
   
      • **Host Name** — Type the fully qualified host name for the computer on which you create the managed node. For example, the fully qualified host name for **tokyo** could be **tokyo.sales.mycompany.com**.
      
      • **Platform** — Click the drop-down list and select the name of the operating system on the managed node.
      
      • **User** — Type the login user ID of the person who performs the installation on the managed node. This user ID must have full administrative privileges (Administrator on Windows, and root on UNIX) on the target computer.
      
      • **Password** and **Verify Password** — Type the login password of the user specified in the User field.

b. Click the Advanced tab.

   Specify values in the fields of the Advanced tab.
   
   • **Destination** — *(Optional)* Type a customer path on the target machine where you want to install the Tivoli software. Otherwise, Tivoli uses the default path specification.
   
   • **Reboot** — You must select this check box when you add Windows managed nodes to the Tivoli region. The managed node software is activated after you restart the Windows computer.
   
   • **Gateway** — Specify if this computer is a gateway.

c. Click OK to save your settings and return to the installer.

14. Click Next to display a list of Tivoli servers.

15. Select each Tivoli server that you are going to use to communicate with endpoints that will use the product. When the product is already installed on a managed node, the row is not in edit mode. The check box is selected and you cannot modify it.

16. Click Next to display the endpoint option panel.

17. Select the endpoints that you want to monitor.

18. *(Optional)* You can create endpoints for Tivoli to monitor and manage. By default the installer assigns endpoints to managed nodes whenever possible. The installer assigns endpoints to the Tivoli server only when no managed node is available.

   Follow these steps to specify endpoints for the installer to create:
a. Click Add.

b. Specify the values in the fields of the Basic tab:
   - **Host Name** — Type the fully qualified host name for the computer on which you create the endpoint. For example, the fully qualified host name for tokyo could be tokyo.sales.mycompany.com.
   - **Label** — Type a unique name for the endpoint. This name can include alphanumeric characters, underscores (_), hyphens (-), and periods (.). The names are case-sensitive. For example, the name EPName is different from epname. This label should be different than the name of any managed node in the Tivoli region.
   - **Platform** — Click the drop-down list and select the name of the operating system on the endpoint.
   - **User** — Type the login user ID of the person who performs the installation on the endpoint. It is recommended that you use Administrator on Windows, and root on UNIX.
   - **Password** and **Verify Password** — Type the login password of the user specified in the User field.

c. Click the Advanced tab.

Specify values in the fields of the Advanced tab, if necessary.
   - **Port** — Type a port number dedicated to communication with the Tivoli server. The default is 9495.
   - **Destination** — Type the path on the target computer where you want to install the Tivoli software.
   - **Reboot** — You must select this check box when you add Windows endpoints. The endpoint software is activated after you restart the target Windows computer.

d. Click OK to save your settings and continue installation in the installer panels.

19. Click Next to display the Web Health Console panel.

20. (Optional) Install the Web Health Console as follows:
   a. The panel displays the list of available managed nodes.
   b. Place a check mark in the Web Health Console column beside the name of each managed node on which you want to install the console. When you specify the installation of the Web Health Console on a pre-existing managed node, you must supply the user ID and password. The installer prompts you for this information when you click Next. If the Web Health
Console is already installed, do not place a check mark here. Otherwise, the console is installed again because the installer cannot detect previous installations of the Web Health Console.

21. Click Next to display the first of a series of windows, each of which prompts you for the directory location of a specified CD or CD image that contains the software to be installed.

22. For each CD prompt window that is displayed, perform the following steps:
   a. Insert the specified CD into the drive.
   b. Type a directory path or click Browse to specify the path to the CD or CD image.
   c. Click Next.
   d. Click Yes in response to the message that asks if you want to copy the files to the depot directory.

23. Click Next on the final CD prompt window to display the Step List window. When the Step List window is displayed, the installer begins installation of the software from the depot to the install location. The Step List shows the components that the installer installs, and it displays status updates for the installation process. See the IBM Tivoli Monitoring: Problem Determination Guide for a description of the status icons and options in the Step List window.

If you want to cancel the installation while the Step List is running, perform the following steps:
   a. Click Stop and allow the currently running process to finish. If you click Cancel without allowing the currently running process to complete, the system might remain in an unpredictable state.
   b. Click Cancel to stop the installation and close the installer.
   c. Delete the temporary depot directory created by the installer.

24. (Optional) Double-click any item in the Step List to see details and to set processing instructions for that step in the Details window.

25. (UNIX-based Tivoli servers only) If you specified Windows endpoints to create in Step 18 on page 21 and no Windows endpoints currently exist in the domain of the Tivoli server, perform the following steps:
   a. Click Stop at the top of the Step List
   b. Perform the procedure described in “Creating a Windows proxy endpoint”

26. Click OK in the final prompt of the Step List.

This final prompt is displayed when all steps in the Step List show a Passed status. If all steps do not show a Passed status, see the IBM Tivoli Monitoring: Problem Determination Guide for troubleshooting information. Click Next.

27. Select Delete the temporary files to delete the directory of temporary installation files that you specified at the beginning of the installation process.

28. Click Next and then click Finish.

### Creating a Windows proxy endpoint

This section describes how to create a Windows proxy endpoint if you are installing from a UNIX-based Tivoli server when there are no existing Windows endpoints in the domain of the server. This procedure is not necessary if you are installing from a Windows system.

The installer can create Windows endpoints for the Tivoli server that you are installing. However, this capability exists only when you are installing the Tivoli server on a Windows operating system.
A proxy endpoint must exist in a Windows domain to enable the installer to create Windows endpoints. When the Tivoli server runs on Windows and you want to create Windows endpoints, the installer automatically creates the proxy endpoint in the domain in which that server resides. This proxy should share the same domain as the Tivoli domain because it uses Windows Universal Naming Convention (UNC) methods to provide remote access to newly created Window endpoints. The proxy endpoint enables creation of Windows endpoints in any domain that has a trusted relationship with the domain that contains the proxy. The Tivoli server uses the user ID and password for the proxy endpoint to obtain access to the file system of the other Windows servers.

The installer also associates the proxy endpoint with a gateway in the Tivoli region. Otherwise, the Tivoli server cannot use the proxy endpoint. You perform this procedure at a specific point during the installation process when a gateway has been created to associate with the proxy endpoint.

With this proxy access, the Tivoli server is able to install endpoints on Windows systems in that domain and in other Windows domains in the network that have a trusted relationship. Without a proxy endpoint, the Tivoli server cannot create Windows endpoints in that remote domain.

In UNIX installations, the installer does not automatically create the proxy endpoint that Windows requires. When the Tivoli server you are installing runs a UNIX operating system, you must use this procedure. The procedure sets up the required proxy endpoint in the Windows domain that contains target Windows servers that you want to monitor.

The required authorization to use this procedure is:

- **on Windows**: the user who installs the product must have membership in the **Administrators** group
- **on UNIX**: the user who installs the product must have **root** privileges

Before you initiate this procedure, you must have followed the Typical installation procedure up to Step 25 on page 213.

You also need to identify the Windows computer that will serve as the proxy endpoint: you need the endpoint name as well as the host name for the machine. For example, if the host name is **vision35.mycompany.com**, you may want to name the endpoint **vision35_ep**.

You perform this procedure in the installer and in the Tivoli Management Framework Endpoint Setup wizard.

To perform the procedure, follow these steps:

1. Click **Stop** at the top of the Step List when the installer starts running **Step 1**.
The system displays a message indicating that it will stop at the end of the current step.

2. Click OK and wait for the step to finish processing.
3. In the Step List scroll to the **TMA installation** step, which installs any endpoints that you have configured.
4. Double-click on the **TMA installation** step to access the Detail window for the step.

5. Perform the following steps:
   a. Click **Toggle Breakpoint**.
   b. Click **OK**.
   This action sets a breakpoint for when the installer begins to install endpoints. A stop sign icon indicates the step at which you inserted a breakpoint. While this step is in breakpoint mode, you enable the endpoint proxy as described in Step 7.
6. Click **Run All** at the top of the Step List window.
   Processing of the steps in the Step List resumes, until the installer reaches the breakpoint and stops. Prior to the breakpoint you set in Step 5 the installer completes the creation of a gateway. To install correctly, the proxy endpoint must refer to this gateway. Otherwise the Tivoli server cannot recognize the proxy endpoint.
7. When the installer reaches the breakpoint you created in Step 5 install the proxy endpoint as follows. Perform these steps on the computer where you are going to install the proxy endpoint, not on the Tivoli server.
   a. Insert the CD labeled **Tivoli Management Framework 2 of 2, Version 4.1.1** into the CD drive of the computer that you want to be the endpoint.
   b. Run the setup executable located in the following subdirectory path of the Tivoli installation directory:
   \lcf\winnt
   c. Click **Next** in the Tivoli Management Framework Endpoint Setup window.
d. Read the license agreement.

e. Click Next to display the Endpoint installation options window.

f. (Optional) Click Browse to change the destination directory for the software.

g. Click Next to display the Tivoli Remote Access Account window.

h. Type the user name and password of the account through which Tivoli Management Framework accesses remote file systems. If you do not require access to remote domains, click Next to skip this step.

i. Click Next to access the Advanced Configuration window.

j. Specify the startup and configuration options information listed in the following table:

<table>
<thead>
<tr>
<th>Field name</th>
<th>Use when ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gateway</td>
<td>The gateway uses a port number other than the default 9494.</td>
</tr>
<tr>
<td>Endpoint</td>
<td>You want to specify a port for the endpoint to use other than the default. The default is 9495.</td>
</tr>
<tr>
<td>Other</td>
<td>Use the (-g host_name_of_the_gateway) option to specify the intercepting gateway. Use the (-n host_name_of_the_endpoint) option to specify the name of the endpoint that you are installing as the proxy.</td>
</tr>
</tbody>
</table>

k. Click Next to continue the installation process.

The installer reports that communication of the proxy endpoint and the gateway was successful. The endpoint is installed and logged into a gateway. The installation process configures the endpoint system to automatically start the endpoint service when the system starts.

If the system does not start, there might be a problem with the communication between the proxy endpoint and the gateway. If this is the case, you should remove and then reinstall the endpoint.

l. Click Next to display the Setup Complete window.

m. Click Finish to display the Restart Windows window.

n. Select No to prevent the restarting of the operating system.

You can restart the system after the installation of Tivoli Monitoring is complete. When you restart the computer, the Tivoli Endpoint service is created, with the startup type set to automatic.

8. Press OK to complete the endpoint installation.

9. On the Tivoli server, click Run All in the installer and complete the Evaluation installation steps that follow Step 25 on page 213.
Appendix C. Migration considerations

This appendix provides guidelines for:

- Tivoli Distributed Monitoring (Classic Edition) users who are migrating to Tivoli Monitoring 5.1.2
- Tivoli Web Component Manager users who are migrating to Tivoli Monitoring 5.1.2

It describes some considerations you should make when planning your migration to make it as easy and efficient as possible.

Migrating from Tivoli Distributed Monitoring (Classic Edition) to Tivoli Monitoring

Tivoli Monitoring allows you to monitor availability and performance status of resources on your systems to identify bottlenecks and potential resource problems.

Users of Tivoli Distributed Monitoring (Classic Edition) can easily migrate to using Tivoli Monitoring.

Overview

Tivoli Monitoring 5.1 applies monitoring in a different way from Tivoli Distributed Monitoring (Classic Edition). Whereas the Tivoli Distributed Monitoring (Classic Edition) is based on the concept of monitoring capability collections and monitors, Tivoli Monitoring applies preconfigured, automated best practices to the automated monitoring of essential system and application resources. Basically, Tivoli Distributed Monitoring (Classic Edition) provides the means for retrieving data through scripts or commands, verifies that the retrieved values do not exceed given thresholds, and finally, based on the user’s customization, triggers some response (Tivoli Enterprise Console® events, e-mail, notices, and so on). Tivoli Distributed Monitoring (Classic Edition) leaves the correlation and the problem’s root cause analysis to the user.

Tivoli Monitoring is based on the concept of resource models, implemented through best practice scripts, or reference models, and the definition in Common Information Model (CIM) in terms of monitored resources. For more details, see the Web site:

http://www.dmtf.org/standards/cim_spec_v22/

Moreover, Tivoli Monitoring provides a set of resource models designed to detect run time bottlenecks and other potential problems and to automatically recover from critical situations, eliminating the need for system administrators to manually scan through extensive performance data.

Tivoli Monitoring may use processes that form part of the endpoints’ operating systems to obtain resource data.

On Windows systems it uses the Windows Management Instrumentation (WMI), which is Microsoft’s implementation of CIM. For more details, see the Web site:

WMI allows applications to retrieve information about the current status of a system.

On UNIX®, Linux, and OS/400 platforms the information collection agent is incorporated in the product based on CIM specifications.

The IBM Tivoli Monitoring Version 5.1.2 product bundle provides both Tivoli Monitoring 5.1.2 and Tivoli Distributed Monitoring (Classic Edition) 3.7, so that users can use both and plan their migration to Tivoli Monitoring 5.1.2 in the smoothest possible way. Users should take the following two aspects into consideration:

**Coexistence**

The two products can coexist as they have two different implementations at the server, gateways, and endpoints. The installation of Tivoli Monitoring does not change the Tivoli Distributed Monitoring (Classic Edition) environment or configuration. Even when working in Compatibility Mode, see "Compatibility mode" on page 226, the two products can coexist and run entirely independently.

**Migration**

To make the migration from Tivoli Distributed Monitoring (Classic Edition) to Tivoli Monitoring 5.1.2 as easy as possible and to save the user’s investment on custom scripts and home-developed monitoring collections, Tivoli Monitoring 5.1.2 runs in "Compatibility Mode", the new working mode that allows Tivoli Monitoring users to use Tivoli Distributed Monitoring (Classic Edition) monitor collections and custom scripts within resource models.

Due to the different nature of the two versions of Tivoli Distributed Monitoring, the migration can only be semi-automated. In addition to the Compatibility Mode which gives users the possibility of using their scripts or monitors by leveraging Tivoli Monitoring features, Tivoli Monitoring also provides a migration helper script, dmae_sentryprofile_analyser.sh, which provides information on how and where the metric monitored by the current Tivoli Distributed Monitoring (Classic Edition) profiles can be replaced using Tivoli Monitoring. The script is able to automatically create Tivoli Monitoring profiles containing resource models that try to address the same problems as the Tivoli Distributed Monitoring (Classic Edition) monitors. More details about this script can be found later in the chapter.

To take full advantage of the Tivoli Monitoring key features such as instance autodiscovery, metric correlation, aggregation, data logging, and so on, may require some Visual Basic or JavaScript coding skills that might not be immediately available, so to help to introduce Tivoli Distributed Monitoring (Classic Edition) users to the Tivoli Monitoring monitoring approach, Tivoli Monitoring provides a wizard-driven process within the Resource Model Builder to create resource models and automatically generate all the required code. The wizard allows the user to follow the simple monitoring paradigm in Tivoli Distributed Monitoring (Classic Edition) by leveraging the new functionality offered by Tivoli Monitoring. The wizard approach relieves the user from writing additional code by generating code that applies simple Tivoli Distributed Monitoring (Classic Edition) logic:

```plaintext
if (metric value - comparison - threshold) then response action
```

Using the wizard together with the compatibility mode allows users to replicate the same monitoring they already have in place with Tivoli

Guidelines for migrating to Tivoli Monitoring 5.1.2

To move monitoring solutions from Tivoli Distributed Monitoring (Classic Edition) to Tivoli Monitoring 5.1.2 implementation, it is important to better understand the key features that Tivoli Monitoring offers at the different phases of its use.

Design time

Tivoli Monitoring provides the Resource Model Builder, an integrated development environment (IDE) to design, create, test, and debug resource models. See the IBM Tivoli Monitoring: Resource Model Builder User’s Guide for more information.

Basically, new Tivoli Monitoring users should consider the Resource Model Builder as the place where they can build their own monitoring best practice, where they can modify the current out-of-the-box resource model best practices, and where they can customize and configure the resource models in a more detailed way. In Tivoli Monitoring users should see the Resource Model Builder as an additional and more complete GUI where they can customize resource models in addition to the Tivoli-based GUI. The importance of the Resource Model Builder is mainly related to the possibility of creating and debugging new resource models.

Data sources

Tivoli Monitoring is able to gather data not only from CIM data sources but also from Tivoli Distributed Monitoring (Classic Edition) monitors and custom scripts (Compatibility Mode). In this way Tivoli Monitoring enhances the set of data sources with those already supported by Tivoli Distributed Monitoring (Classic Edition). However, it is still preferable to work in native mode, where the monitored data is collected from the underlying CIM implementation. While on UNIX, Linux, and OS/400 platforms the CIMOM implementation is embedded on the engine, on Windows platforms the engine is based on WMI implementation. See the Web site:


Microsoft operating systems and Backoffice applications are delivered with their own WMI providers so that the monitoring of such resources can be implemented very quickly.

Aggregation

Differently from Tivoli Distributed Monitoring (Classic Edition), in Tivoli Monitoring all events can be consolidated locally at the endpoint before flowing to the upper layers such as Tivoli Enterprise Console or Tivoli Business Systems Manager. This feature is very important from a scalability point of view. Also, the best practice can leverage this functionality to make the troubleshooting analysis smarter and more realistic (for example, whether a process exceeds its CPU usage just once or many times). New Tivoli Monitoring users should always take this feature into account when generating an event.

Clearing

As the starting point for Tivoli Monitoring is providing information for problems that are critical to operating system or application resources, Tivoli Monitoring notifies users only when the problem arises through Tivoli Enterprise Console or Tivoli Business Systems Manager, and only when the problem has been solved, does it send a warning with a Clearing Event.
Correlation

The most important feature is definitely having the possibility to write best practices to troubleshoot problems before they happen. That is, the root cause analysis can be performed by the resource model script that, using the Tivoli Monitoring engine API, defines the data to be collected and, once it has been collected by the engine, retrieves all the metric values and correlates them to establish the final cause of arising problems. The correlation is then achieved by following a programming model that, in simple cases can be implemented by using the wizard-driven process. Metric values collected through the compatibility mode can of course be correlated too.

Multimetric

When starting to write new resource models, new Tivoli Monitoring users should consider that in each resource model it is possible to collect an undefined number of resources and for each resource an undefined number of metrics. Moreover it gives the possibility to have autodiscovery for all the instances of a resource.

Note: Implementing a resource model that looks just at one metric of a resource may be expensive as Tivoli Monitoring creates a new thread and a set of objects for each script. Better results can be obtained by designing the resource model to address a specific set of problems (for example, memory and CPU bottlenecks).

Automation

When building new resource models, or while modifying existing ones, Tivoli Monitoring users can set recovery actions called built-in actions to be triggered whenever an indication is consolidated. The built-in actions are the running of CIM methods (see IBM Tivoli Monitoring: Workbench User’s Guide) against a CIM class or CIM class instance.

Logging

Tivoli Monitoring allows users to log data to a local database. Data stored in such a database can be retrieved and viewed through the Web Health Console. Data retrieved through the compatibility mode can also be logged.

Deployment time

Tivoli Monitoring resource models are configured and deployed through Tivoli Management Framework just as they are in Tivoli Distributed Monitoring (Classic Edition). Instead of using a Monitoring Collection and monitors, Tivoli Monitoring users use resource models, and instead of creating a SentryProfile create a Tivoli Monitoring profile, Tnw2kProfile. As the configuration and deployment procedures of Tivoli Monitoring follow the paradigm followed by all profile-based Tivoli monitoring applications, we describe only the things that differ from Tivoli Distributed Monitoring (Classic Edition) usage.

Resource model installation

Once Tivoli Monitoring users have built their own resource models they can build the package with the Resource Model Builder (see IBM Tivoli Monitoring: Resource Model Builder User’s Guide) and then install it on the Tivoli management region, using the wdmrm command.

Tivoli Enterprise Console and Tivoli Business Systems Manager Customization

Tivoli Monitoring allows users to set a unique destination Tivoli Enterprise Console server per profile. This means that all resource models belonging to the same profile send events to the Tivoli Enterprise Console server. The target Tivoli Enterprise Console can be specified from the properties dialog.
Task configuration
Tivoli tasks can be run as recovery actions. Differently from Tivoli Distributed Monitoring (Classic Edition) they can be triggered only when an indication has been consolidated.

Note: In the Task environment there are the Event properties in the form of environment variables.

Profile distribution
Tivoli Monitoring supports endpoints only. This means that there is no managed node, and implies some rework to be done when replacing Tivoli Distributed Monitoring (Classic Edition) profiles with Tivoli Monitoring profiles because Tivoli Distributed Monitoring (Advanced Edition) cannot be distributed on managed nodes.

Run time
Once a Tivoli Monitoring profile and its resource models have been distributed to endpoints, the following considerations must be made for Tivoli Monitoring:

Multithread
Tivoli Monitoring operates in a multithread environment. This means that each script runs in a separate thread and no processes are generated except those needed to run in compatibility mode. Tivoli Monitoring users should take this into account when creating a resource model that looks at only one metric.

Web Health Console

Tivoli Enterprise Console rule for clearing event
Tivoli Monitoring provides a Tivoli Enterprise Console rule to automatically close events for which a Clearing event is generated.

Migration process
As described above, Tivoli Monitoring provides a set of facilities to help and facilitate Tivoli Distributed Monitoring (Classic Edition) users to migrate their monitoring solutions into a Tivoli Monitoring environment. There is no tool to completely automate the migration process because, as described above, the two products are really different.

So what should new Tivoli Monitoring users do? How can they migrate to the new monitoring infrastructure? Here is a description of the facilities that, used together, help users in their migration task. Tivoli Monitoring users who do not plan to use the Sentry Profile analyzer script should still read the following section because it describes the thought process to be followed when planning to migrate.

Sentry Profile analyzer
The first facility is the Sentry Profile analyzer script that analyses the contents of all the Sentry Profiles and, based on a mapping table provided with Tivoli Monitoring, produces a report suggesting how the monitors can be replaced with resource models or how new resource models can be created to collect the same data.
Installation Path

The installation of Tivoli Monitoring stores the script named dmae_sentryanalyser.sh in the $BINDIR/TME/Tmw2k/migration_helper directory. The mapping table file named monitors_rm_table is also stored in the same directory.

Mapping Table Content

The mapping table defines:

- If the metric values returned by each monitor provided by Tivoli Distributed Monitoring (Classic Edition) are collected by a Tivoli Monitoring resource model, for example, the monitor AvailBytes of the NT_Memory monitoring collection is collected by the Memory resource model.
- Which CIM class has as property the metric defined in a Tivoli Distributed (Classic Edition) monitor.

Note: many metrics usually belong to the same class, for example, the monitor AvailBytes of NT_Memory can be found as a property of the CIM class TMW_Memory, and the monitor Committed Bytes can be found as a property of the same TMW_Memory class.

- For the metrics defined in a CIM class, there could be a Tivoli Monitoring MOF file where they are defined. See the Web site:
  

for more details. For example, the class TMW_Memory is defined in the file TMW_Resources10.mof.

Note: the TMW_Resources10.mof file is automatically installed on WMI with the endpoint engine when the first push is performed.

All the other mof files are only installed in the related CIM implementation when the resource model containing these files as dependencies gets downloaded. This means that when creating a resource model that uses a resource defined in TMW_Resource10.mof there is no need to add that file to the dependency, while for all other cases the user must add the mof file to the dependency of the resource model.

- If a metric is made available by any CIM provider, independently of whether or not it is a property of a CIM class, for example, the TransitionFaults metric is made available by the PerfProv WMI provider, but there is not any CIM class already available on Window NT that indirectly uses that provider to collect that counter.

At the Web site:


users can find more information on how to use performance data providers and information on all other WMI providers.

All providers used on Windows platforms are provided by default with WMI core, so there is no need to install or download these on the endpoints.

- There could be the case of a TMW_NetworkPortInfo whose provider has been developed by Tivoli and then, in order to be used on an endpoint, the provider must be added to the dependencies of a resource model. All providers used for UNIX, Linux, and OS/400 platforms, as there is no native CIM implementation, have been developed by Tivoli and their
entire implementation must be added to the resource model dependencies. The implementations of all UNIX, Linux, and OS/400 providers consist of:

- A tar file containing a set of Java class files
- For each supported platform, a native shared library responsible for gathering data.

Based on the above considerations, in order to reuse a UNIX, Linux, or OS/400 CIM class inside a resource model, users must add to the dependencies the related mof file and the tar file (they are common to all platforms) and the related shared library for each platform to be supported.

The Sentry Profile analyzer script must be run from the Tivoli environment on any Server or Managed Node. Based on the content of the mapping table described above, the Sentry Profile analyzer analyzes all Sentry Profiles present in the Tivoli management region and produces a report that suggests the way to proceed in the migration process.

Optionally, using `-p` option can also create Tivoli Monitoring profiles with the resource models covering as much as possible the resources monitored by the Tivoli Distributed Monitoring (Classic Edition) monitors.

**The Analyzer Script process**

The script proceeds with the following paradigm.

- It verifies whether the metrics of the monitors set in a SentryProfile are collected by any Tivoli Monitoring. If they are, the script generates a section in the report describing which monitor can be replaced with which resource model. Optionally a Tmw2kProfile with the same name as that containing the monitor is created. Other resource models can also be added to that profile. Note that the configuration of a monitor in terms of arguments, responses, schedules, and so on, cannot be replicated because of the different product object models (see the section above). Usually the arguments in Tivoli Distributed Monitoring (Classic Edition) monitors are used to identify a specific resource instance. This is in conflict with the format of the out-of-the-box Tivoli Monitoring resource models which try to autodiscover the failing instances at run time. Current resource models in fact do not accept resource instances as input. Nevertheless they can easily be modified with the Resource Model Builder by using the parameters that only look and perform their analyses on specific instances. Also, other customization related to responses and schedules cannot be automatically recreated within a resource model but instead existing resource models can be modified in order to simulate the Tivoli Distributed Monitoring (Classic Edition) configuration.

- If no resource models that collect a specific metric are found in the mapping table then the script tries to identify whether a CIM class exists which provides that value. If one does then a section is added to the report. This only happens for Windows Platforms because there are several providers and CIM classes already made available by Microsoft. This information is really important because users, using the Resource Model Builder (with or without the wizard), can easily take advantage of these classes to create a more sophisticated resource model running in native mode.
• Sometimes there might be the case where the script finds a metric that is not implemented by any CIM class, but that is provided by a provider such as PerfProv. Microsoft provides with WMI core the WMI performance counter provider that can be used to define CIM classes representing the objects and their counters as they appear in the Windows performance monitor. Instructions on how to use the provider can be found at the Web site:


Good examples can be seen in the TMW_Resources10.mof that is stored in $BINDIR/..//cf_bundle40/Tmw2k/Mof. Most Tivoli Monitoring Windows resource models are based on CIM classes using the PerfProv WMI provider. Here is an example:

On Windows platforms the provider for instances or properties of a given class is specified through the "Provider" CIM qualifier.

For example the provider for the following class NTProcesses is the "PerfProv".

[dynamic, provider("PerfProv"), ClassContext("local\Process")]
class NTProcesses
{
[key]
String Process;
[PropertyContext("ID Process")]
uint32 ID;
[PropertyContext("Working Set")]
uint32 WorkingSet;
);

The script adds a section to the report suggesting that the user considers creating a new resource model based on new CIM classes that can be created using the indicated provider.

• If the monitor examined cannot be remapped in any CIM class without the creation of a new provider (see the Web site


for how to create WMI providers), the script suggests using the Compatibility Mode in conjunction with the Wizard-driven process and choosing the Tivoli Distributed Monitoring Classic Monitoring Collection as the data source. Note that in order to optimize the use of monitors users should consider using more than one monitor of the same Monitoring collection inside the same resource model wherever possible.

• If the monitor is a custom script (string or numeric) and cannot be remapped in any CIM class without the creation of a new provider (see the Web site


for how to create WMI providers), the script suggests using the Compatibility Mode in conjunction with the Wizard-driven process and choosing Custom Script as the data source.

• If the monitor is an async monitor then users should ask Tivoli services to replace that functionality as it has not yet been implemented.

Examples: Here is an example of report generated by the Sentry Profile analyzer script:
The metrics collected by the following monitors are collected by the by the following RESOURCES MODELS:

<table>
<thead>
<tr>
<th>COLLECTION</th>
<th>MONITOR</th>
<th>RESOURCE MODEL</th>
<th>CIM CLASSES(PROVIDER)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UnixProfile#MyRegion</td>
<td>Unix_Sentry</td>
<td>File size</td>
<td>DMXFile(DMXFileIlit)</td>
</tr>
<tr>
<td>UnixProfile#MyRegion</td>
<td>Unix_Sentry</td>
<td>File size</td>
<td>DMXSecurity(DMXFileSecIlit)</td>
</tr>
<tr>
<td>UnixProfile#MyRegion</td>
<td>Unix_Sentry</td>
<td>Space free</td>
<td>DMXFileSystem(DMXFileSystemIlit)</td>
</tr>
<tr>
<td>WinNTProfile#MyRegion</td>
<td>NT_NetworkMonitor</td>
<td>Network utilization</td>
<td>TMW_NetworkIntCard(TMW_NetworkSegment(PerfProv))</td>
</tr>
<tr>
<td>WinNTProfile#MyRegion</td>
<td>NT_LogicalDisk</td>
<td>Avg Disk sec/Trans</td>
<td>TMW_LogicalDisk(TMW_LogicalDisk(PerfProv))</td>
</tr>
</tbody>
</table>

The metrics collected by the following monitors can be found in the following CIM CLASSES:

<table>
<thead>
<tr>
<th>COLLECTION</th>
<th>MONITOR</th>
<th>CIM CLASSES(PROVIDER)</th>
</tr>
</thead>
</table>

Please consider to create Cim class to use the performance provider.
For more information visit:

The metrics collected by the following monitors are not collected by any resource model and can not be found in any CIM class.
Please consider to use WORKBENCH WIZARD choosing 'DM CLASSIC MONITORING COLLECTION' to import the monitor in a resource model

<table>
<thead>
<tr>
<th>COLLECTION</th>
<th>MONITOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>DummyProfile#NewPolicy</td>
<td>Unix_Sentry</td>
</tr>
<tr>
<td>DummyProfile#NewPolicy</td>
<td>Unix_Sentry</td>
</tr>
<tr>
<td>DummyProfile#NewPolicy</td>
<td>Unix_Sentry</td>
</tr>
<tr>
<td>WinNTProfile#MyRegion</td>
<td>NT_Cache</td>
</tr>
<tr>
<td>WinNTProfile#MyRegion</td>
<td>NT_Cache</td>
</tr>
<tr>
<td>WinNTProfile#MyRegion</td>
<td>NT_Cache</td>
</tr>
</tbody>
</table>

The metrics collected by the following monitors are not collected by any resource model and can not be found in any CIM class.
Please consider to use WORKBENCH WIZARD choosing 'CUSTOM SCRIPTS' to import the monitor in a resource model

<table>
<thead>
<tr>
<th>COLLECTION</th>
<th>MONITOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>DummyProfile#NewPolicy</td>
<td>Unix_Sentry</td>
</tr>
<tr>
<td>DummyProfile#NewPolicy</td>
<td>Unix_Sentry</td>
</tr>
<tr>
<td>DummyProfile#NewPolicy</td>
<td>Unix_Sentry</td>
</tr>
<tr>
<td>OS400Profile#EMEA-region</td>
<td>OS/400 Job</td>
</tr>
<tr>
<td>OS400Profile#EMEA-region</td>
<td>OS/400 Object</td>
</tr>
<tr>
<td>UnixProfile#MyRegion</td>
<td>Unix_Sentry</td>
</tr>
</tbody>
</table>

Appendix C. Migration considerations 225
Please consider to ask Tivoli services for replacing the following monitors:

<table>
<thead>
<tr>
<th>COLLECTION</th>
<th>MONITOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>UnixProfile#MyRegion</td>
<td>Unix_Sentry</td>
</tr>
<tr>
<td></td>
<td>sasync</td>
</tr>
</tbody>
</table>

**Compatibility mode**

The compatibility mode is a new working mode allowing Tivoli Monitoring users to use Tivoli Distributed Monitoring (Classic Edition) monitors inside a Tivoli Monitoring resource model. In this way, Tivoli Monitoring can collect data to analyze, not only from the CIM data source, but also from Tivoli Distributed Monitoring (Classic Edition) monitoring sources (also known as probes). This means that Tivoli Distributed Monitoring (Classic Edition) users can recycle their existing customized Tivoli Distributed Monitoring (Classic Edition) monitoring collections into the new Tivoli Monitoring resource models.

The compatibility mode can be used in addition to the native mode, so it is possible to have resource models that use both these technologies in their implementation. The compatibility mode provides an easy way to import Tivoli Distributed Monitoring (Classic Edition) probe implementation into a resource model without the user having to write any additional code, by supplying the Resource Model Builder with a new wizard that imports Tivoli Distributed Monitoring (Classic Edition) monitoring sources. The Wizard process is described in *IBM Tivoli Monitoring: Workbench User’s Guide*.

The following picture shows the Tivoli Monitoring endpoint engine architecture.
The Service Object API enables the resource model scripts to use CIM objects, monitoring collections, and custom scripts in any combination. WMI (on Windows) and the CIM Object Manager (CIMOM, on the other supported platforms) are responsible for loading the providers that, in turn, get performance and availability data from system and application resources.

**Wizard process**

The Resource Model Builder provides a wizard, that is a GUI-driven process, to create new resource models. Basically, the wizard, starting from a selected CIM class taken from the WMI repository, from a monitoring collection, or from a custom script, displays a sequence of dialogs with default values already filled in to drive the user to create simple resource models. At the end of the wizard process all the needed Visual Basic or JavaScript code is automatically generated. This code implements a monitoring logic very similar to that used by Tivoli Distributed Monitoring (Classic Edition), that is:

\[
\text{if (metric value - comparison - threshold) then indication}
\]

Obviously the generated code takes into account the Tivoli Monitoring features described in earlier sections so it is optimized to fit into the new monitoring paradigm. The wizard process is described in detail in *IBM Tivoli Monitoring: Workbench User’s Guide*. 

---

**Figure 13. Tivoli Monitoring endpoint engine architecture.**
How everything fits together

The following picture summarizes the migration approach that new Tivoli Monitoring users should follow after taking into consideration what has been described above:

1. The Sentry Analyzer script analyzes the Sentry Profile based on a monitor-resource model mapping table
2. The Sentry Analyzer script generates a report
3. For each monitor in the report the user can:
   a. If a resource model collecting the related metric exists:
      1) If the resource model addresses the monitoring problems controlled by the monitor, customize the resource model and deploy it.
      2) If the resource model does not address the monitoring problems controlled by the monitor, go to step 4.
   b. If a resource model collecting the related metric does not exist, but a CIM class and its provider do, then go to step 4.
   c. If a resource model collecting the related metric does not exist and neither does a CIM class, but a provider does (usually this happens for WMI performance data providers):
      1) Write the MOF.
      2) Then go to step 4.
   d. If a resource model collecting the related metric does not exist, and nor do a CIM class nor a provider:
      1) Write the provider.
      2) Write the MOF as indicated.
      —OR—
         Use the compatibility mode.
      3) Then go to step 4.
4. If the monitoring problem is complex and needs correlation and aggregation:
   a. Create a new resource model or modify an existing one using all features provided by Tivoli Monitoring.
   b. Go to step 6.
5. If the monitoring problem is a simple one that follows the Tivoli Distributed Monitoring (Classic Edition) monitoring paradigm, then use the wizard process in the Resource Model Builder to create a new resource model.
6. Build the resource model package and install it on the Tivoli management region using the wdmrm command.
7. Deploy the resource model.
Migrating from Tivoli Web Component Manager

Tivoli Monitoring provides some tools to facilitate the migration of Tivoli Web Component Manager users to Tivoli Monitoring:

- A Java tool (called Tims2XML) which parses the database on the Tivoli Web Component Manager server (also referred to as TIMS server) and retrieves information about all TIMS tasks.

- A set of Tivoli Monitoring resource models that have a behavior very similar to Tivoli Web Component manager tasks.
**Tims2XML**

Tims2XML is a Java application that collects data related to Tivoli Web Component Manager tasks which are running on a TIMS server. Once collected, the data is automatically formatted into XML.

Tims2XML is available on the Tivoli Monitoring CD.

**Installation and customization**

To be used, Tims2XML must be installed on the TIMS server whose tasks needs to be monitored.

Before using the tool, the launch.bat file (or launch.sh on UNIX) need to be customized. The CLASSPATH must be set to point to specific .jar files. These are:

- `fw-common.jar`
- `fw-svr.jar`
- `properties.jar`

In addition, the `lib` and `lib/properties` directories must be included in the CLASSPATH.

Another required component is the JDBC library that is provided with your Oracle/DB2 installation, for example `classes111.zip` for Oracle.

The following is an example of CLASSPATH definition:

```sh
set TIMS_HOME=D:\Tivoli\Internet\ManagementServer\TIMS
set CLASSPATH=%TIMS_HOME%\lib\fw-common.jar
set CLASSPATH=%CLASSPATH%;%TIMS_HOME%\lib\fw-svr.jar
set CLASSPATH=%CLASSPATH%;%TIMS_HOME%\lib\properties.jar
set CLASSPATH=%CLASSPATH%;%TIMS_HOME%\lib
set CLASSPATH=%CLASSPATH%;%TIMS_HOME%\lib\properties
set CLASSPATH=%CLASSPATH%;D:\Oracle\Ora81\jdbc\lib\classes111.zip
set CLASSPATH=%CLASSPATH%;.
```

Once you have set the CLASSPATH, the Tims2XML can be launched by double-clicking the `launch` file or by issuing the following command:

```sh
java Tims2XML > TIMS_tasks.xml
```

where `TIMS_tasks.xml` is the name of the XML file that is generated.
Structure of the XML file
A batch file and some HTML files are provided on the Tivoli Monitoring CD to show the contents of the XML file on Internet Explorer. The following is a sample:

The XML contains the following information for each task:

- **taskname**: the name of the task, as a unique identifier
- **type**: the type of task, which can be either of the following values:
  - APACHEMONITORINGTASK
  - IISMONITORINGTASK
  - SYSTEMMONITORINGTASK
- **description**: a description of the task
- **reporttype**: the type of report, which can be either of the following values:
  - DAILY
  - WEEKLY
  - MONTHLY
  - None
- <schedule>
  - <startnow>1.0</startnow>
  - <runforever />
  - <startdatetime>/
    - <enddatetime>2002-01-09 12:00:00</enddatetime>
    - <starttime>01:15:00</starttime>
    - <endtime>02:15:00</endtime>
    - <timezone>endpoint_tz</timezone>
  </schedule>
- <endpoint>
  - <epname>trinity</epname>
  - <ipaddr>trinity</ipaddr>
</endpoint>

- parameters: a set of parameter tags that define the requested monitor activities
- normalization: the normalization value for the task. The normalization process occurs after a threshold has exceeded (see the constraint tag), and when the observed value is above or below a specified normalization value (for example, 10%)

Each single parameter tag describes an entity to be monitored. The entity is always associated to a name, but it may or may not have a value (some monitors do not need to be configured). For example, the CPU resource has a name cpu but not a value; however, a file monitoring activity must be initialized with a path/file name value.
A parameter can be associated with several monitors, or no monitor at all (for example, a configuration parameter may not have a monitor associated). However, if a monitor is activated, then the XML output contains a monitor section.

The generic structure of a monitor tag is the following:
- *name* (mandatory)
- *rate* (not mandatory)
- *constraint* (not mandatory or empty)

The *rate* tag may not be applicable to a monitor (for example, it is not applicable to the Windows Event Log monitor).

The *constraint* tag is not applicable when the monitored attribute has a report only type (for example, the changedStatus monitor for a FILE resource), otherwise the *constraint* tag applies.

**System monitoring tasks**
This section describes the XML output for the following System Monitoring tasks:
- Process Monitoring
- CPU Monitoring
- UNIX File System Monitoring

**Process Monitor task:** For the Process Monitor task, the *parameter* section shows the following tags:
- The *name* tag always contains the same identifier (processName)
- The name of the process is inside the *value* tag
- If the N_Threads Monitor is active, then its monitor section contains:
  - The numThreads identifier into the *name* tag
  - The sampling interval (in seconds) into the *rate* tag
  - The threads threshold into the *constraint* tag
- If the % CPU Monitor is active, then its monitor section contains:
  - The percCPU identifier into the *name* tag
  - The sampling interval (in seconds) into the *rate* tag
  - The CPU occupation threshold (%) into the *constraint* tag
- If the % Memory Monitor is active, then its monitor section contains:
  - The percMem identifier into the *name* tag
  - The sampling interval (in seconds) into the *rate* tag
  - The memory occupation threshold (%) into the *constraint* tag
As an example, consider the Tivoli Web Component Manager configuration panel for the Process Monitor task:

<table>
<thead>
<tr>
<th>Process Name</th>
<th>Sampling Interval (in seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>java</td>
<td>50</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Enable</th>
<th>Monitor Name</th>
<th>Event Violation</th>
<th>Event Recovery</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️</td>
<td>N Threads</td>
<td>Warning</td>
<td>Warning</td>
</tr>
<tr>
<td>✔️</td>
<td>CPU</td>
<td>Warning</td>
<td>Warning</td>
</tr>
<tr>
<td>✔️</td>
<td>Memory</td>
<td>Warning</td>
<td>Warning</td>
</tr>
</tbody>
</table>

The following picture shows the corresponding section in the XML file:

```
- <parameter>
  - <name>processName</name>
  - <value>java</value>
  - <monitor>
    - <name>numThreads</name>
    - <rate>50000.0</rate>
    - <constraint>30.0</constraint>
  - <monitor>
    - <name>percCPU</name>
    - <rate>50000.0</rate>
    - <constraint>50.0</constraint>
  - <monitor>
    - <name>percMem</name>
    - <rate>50000.0</rate>
    - constraint />
  </monitor>
</parameter>
```

**CPU Monitor task:** For the CPU Monitor task, the parameter section shows the following tags:

- The name tag always contains the same identifier (cpu)
- The name of the process is inside the value tag
- If the **System CPU Monitor** is active, then its monitor section contains:
  - The sysCPU identifier into the name tag
  - The sampling interval (in seconds) into the rate tag
  - The system CPU occupation threshold into the constraint tag
- If the **% User CPU Monitor** is active, then its monitor section contains:
  - The userCPU identifier into the name tag
  - The sampling interval (in seconds) into the rate tag
  - The user CPU occupation threshold (%) into the constraint tag
If the % Total CPU Monitor is active, then its monitor section contains:
  – The totCPU identifier into the name tag
  – The sampling interval (in seconds) into the rate tag
  – The total CPU occupation threshold (%) into the constraint tag

As an example, consider the Tivoli Web Component Manager configuration panel for the CPU Monitor task:

<table>
<thead>
<tr>
<th>CPU</th>
<th>Enable</th>
<th>Monitor Name</th>
<th>Sampling Interval (in seconds)</th>
<th>Event Violation</th>
<th>Event Recovery</th>
</tr>
</thead>
<tbody>
<tr>
<td>☑</td>
<td>System CPU</td>
<td>%</td>
<td>60</td>
<td>Warning</td>
<td>Warning</td>
</tr>
<tr>
<td>☑</td>
<td>User CPU</td>
<td>%</td>
<td>60</td>
<td>Warning</td>
<td>Warning</td>
</tr>
<tr>
<td>☑</td>
<td>Total CPU</td>
<td>%</td>
<td>60</td>
<td>Warning</td>
<td>Warning</td>
</tr>
</tbody>
</table>

The following picture shows the corresponding section in the XML file:

```xml
- <parameter>
  <name>cpu</name>
  - <monitor>
    <name>sysCPU</name>
    <rate>50000.0</rate>
    <constraint>40.0</constraint>
  </monitor>
  - <monitor>
    <name>userCPU</name>
    <rate>50000.0</rate>
    <constraint/>
  </monitor>
  - <monitor>
    <name>totCPU</name>
    <rate>50000.0</rate>
    <constraint>80.0</constraint>
  </monitor>
</parameter>
```

UNIX File System Monitor task: Within a single task it is possible to monitor up to 10 UNIX File Systems simultaneously. Therefore we univocally associate each file system to a parameter section:

- The name tag always contains the same identifier (fileSystemName)
- The file system mount point is inside the value tag
- The name of the process is inside the value tag
- If the Percentage Used Monitor is active, then its monitor section contains:
  – The fsUsed identifier into the name tag
  – The sampling interval (in seconds) into the rate tag
  – The FS used threshold into the constraint tag
- If the % MB Used Monitor is active, then its monitor section contains:
  – The mbFSUUsed identifier into the name tag
- The sampling interval (in seconds) into the *rate* tag
- The FS used threshold (in MBytes) into the *constraint* tag

- If the **INodes Free Monitor** is active, then its monitor section contains:
  - The iNodesFSFree identifier into the *name* tag
  - The sampling interval (in seconds) into the *rate* tag
  - The INodes Free threshold (%) into the *constraint* tag

- If the **Percentage INodes Free Monitor** is active, then its monitor section contains:
  - The percINodesFSFree identifier into the *name* tag
  - The sampling interval (in seconds) into the *rate* tag
  - The INodes Free threshold (%) into the *constraint* tag

As an example, consider the Tivoli Web Component Manager configuration panel for the File System Monitor task:

<table>
<thead>
<tr>
<th>Monitor Name</th>
<th>Sampling Interval (in seconds)</th>
<th>Event Violation</th>
<th>Event Recovery</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percentage Used Monitor</td>
<td>50 s</td>
<td>Unknown</td>
<td>Unknown</td>
</tr>
<tr>
<td>Mb Used Monitor</td>
<td>40 s</td>
<td>Unknown</td>
<td>Unknown</td>
</tr>
<tr>
<td>INodes Free Monitor</td>
<td>20 s</td>
<td>Unknown</td>
<td>Unknown</td>
</tr>
<tr>
<td>Percentage INodes Free Monitor</td>
<td>10 s</td>
<td>Unknown</td>
<td>Unknown</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>File System Name</th>
<th>Percentage Used</th>
<th>Mb Used</th>
<th>INodes Free</th>
<th>Percentage INodes Free</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>Enabled</td>
<td>50 %</td>
<td>1024 MBytes</td>
<td>70 %</td>
</tr>
</tbody>
</table>
The following picture shows the corresponding section in the XML file:

```
- <parameter>
  - <name>fileSystemName</name>
  - <value>/data</value>
  - <monitor>
    - <name>fsUsed</name>
    - <rate>50000.0</rate>
    - <constraint>50.0</constraint>
  </monitor>
  - <monitor>
    - <name>mbFSUsed</name>
    - <rate>40000.0</rate>
    - <constraint>1024.0</constraint>
  </monitor>
  - <monitor>
    - <name>iNodesFSFree</name>
    - <rate>20000.0</rate>
    - <constraint>5000.0</constraint>
  </monitor>
  - <monitor>
    - <name>percINodesFSFree</name>
    - <rate>10000.0</rate>
    - <constraint>70.0</constraint>
  </monitor>
</parameter>
```

Resource models

To assist with the migration, Tivoli Monitoring provides a set of resource models that have a behavior very similar to Tivoli Web Component Manager tasks.

The Tivoli Monitoring CD contains six resource models (three for Windows and three for UNIX):

- Process Monitor resource model (Windows and UNIX)
- Memory Monitor resource model (Windows and UNIX)
- Logical Disk Monitor resource model (Windows)
- File System Monitor (UNIX)

To use these resource models, you need to add them to the Tivoli Monitoring server by running the `wdrmrm -add` command.

Logical Disk resource model

The Logical Disk resource model monitors logical disks on a Windows system.

To use this resource model, create a new Tivoli Monitoring profile, then select the TWCM Migration Resource Model category, and the TWCM Migration Logical
**Disk Monitor** resource model, as shown in the following dialog:

![Disk Monitor resource model](image)

This resource model can monitor disk space and disk space percentage.

The thresholds value is by default set to zero, so that no event is generated by the resource model, but you can modify this value by clicking the **Threshold Name** and entering a new value.

You can select the logical disks that needs to be monitored by clicking the **Parameter** button and by modifying the default values.

By default, data logging is not enabled. To have a log, click **Logging**.

**Processor resource model**

The Memory resource model is available for both Windows and UNIX systems.

To use this resource model, create a new Tivoli Monitoring profile, then select the **TWCM Migration Resource Model** category, and the **TWCM Migration Processor Model** (or **TWCM Migration Unix Processor Model**) resource model, as shown in
the following screen:

Memory resource model
The Memory resource model is available for both Windows and UNIX systems.

To use this resource model, create a new Tivoli Monitoring profile, then select the TWCM Migration Resource Model category, and the TWCM Migration Memory Model (or TWCM Migration Unix Memory Model) resource model.

File System resource model
The File System resource model is available only for UNIX systems.

To use this resource model, create a new Tivoli Monitoring profile, then select the TWCM Migration Resource Model category, and the TWCM Migration Memory Model (or TWCM Migration Unix File System Model) resource model.

This resource model can monitor Free INode Percentage and Free Space Percentage. By default, all thresholds are set to zero so that the resource model does not generate any event.

To select the file systems to monitor, click Parameter and set the appropriate mount point values (if no values are set, the resource model gives an error).

Migration sample: Memory Monitor task
This section describes how a Memory Monitor task on a UNIX system can be migrated to Tivoli Monitoring.

The following screen shows the Tivoli Web Component Manager configuration panel for the Memory Monitor task:
Once the Tims2XML tool has retrieved the information about the tasks running on the TIMS server, we get the following XML output for the Memory Monitor task:

```
- <parameter>
  <name>memory</name>
- <monitor>
  <name>memUsed</name>
  <rate>50000.0</rate>
  <constraint>49.0</constraint>
</monitor>
</parameter>
```

To replicate this behavior in Tivoli Monitoring, a new profile needs to be created with the following data:

- Category: TWCM Migration Model
- Resource Model: TWCM Migration Unix Memory Model
- Cycle Time: 50
- Memory Percentage Lower Bound: 49

as shown in the following dialog:
Appendix D. Supporting clusters

A cluster (as it pertains to computers) refers to a group of servers and other resources that are connected through hardware, networks, and software to behave as if they were a single system: the cluster of computers appears as if it were one totally integrated system.

The support provided for clusters by Tivoli Monitoring is a limited solution and is similar to the support that was provided by Tivoli Distributed Monitoring (Classic Edition).

The solution is based on having a Tivoli endpoint running on each node, and on having one or more additional "virtual" endpoints floating among all nodes in the cluster, and not tied to a particular physical endpoint. More specifically, the solution requires multiple instances of the Tivoli Monitoring Agent running on a single system: one of the agents monitors the system and its resources, while the other agents run in a different resource group (or cluster) and monitor the shared resources contained in that group.

It must be pointed out that, with this solution, Tivoli Monitoring is not aware of the cluster and is not capable of recognizing the shared resources, nor is capable of managing the cluster manager application.

The following sections describe two specific clustering configurations and how they can be supported by Tivoli Monitoring on UNIX or Windows nodes.

True fail over cluster on UNIX

On UNIX, the true fail over cluster includes a production node (node A) and a stand-by node (node B). The applications are running on the production node, while the stand-by node is running only a copy of the operating system with the necessary components of clustering software.

In case of fail over, all shared cluster resources (applications, and file systems) are moved to the stand-by node that becomes the production node.

To support this configuration, you must have two instances of Tivoli endpoint code running in parallel at each given moment on the system working as a production node: a local instance of that system and a shared instance for handling resources of the cluster.

To enable this configuration, you create two filesystems on a shared DASD (Disk Array SCSI Drive), for example:

```
/opt/Tivoli/lcf_s
/etc/Tivoli/lcf_s
```

These filesystems must be included in the fail over plan for the cluster configuration, so that they can be unmounted, moved, and mounted on the stand-by node when the fail over occurs.
After that, you install a shared instance (_SHARED_) of Tivoli endpoint in each filesystem on the shared DASD, and you install local instances (_LOCAL_) of Tivoli endpoint on each node of the cluster. To do this, you must create a local file on each node:

on Node A (production):

/opt/Tivoli/lcf1
/etc/Tivoli/lcf1

on Node B (stand-by):

/opt/Tivoli/lcf2
/etc/Tivoli/lcf2

Then you install local instances of Tivoli endpoint on each node to those file systems using any method (for example `winstlcf`), with the following lcf options (supplied using the option `-D` in the command line or in the last.cfg file in $LCF_DATDIR):

Local Tivoli Endpoint for Node A:

`lcfd_port=9497`
`lcfd_preferred_port=9497`
`gateway_port=9494`
`lcfd_alternate_port=9498`

Local Tivoli Endpoint for Node B:

`lcfd_port=9499`
`lcfd_preferred_port=9499`
`gateway_port=9494`
`lcfd_alternate_port=9500`

By default, the shared instance will use the following settings:

`lcfd_port=9495`
`lcfd_preferred_port=9495`
`gateway_port=9494`
`lcfd_alternate_port=9496`

Therefore, for example, in a normal situation we have:

on Node A (production):
local instance of Tivoli endpoint running on port 9497 with alternate 9498
shared instance of Tivoli endpoint running on port 9495 with alternate 9496
on Node B (stand-by):
local instance of Tivoli Endpoint running on port 9499 with alternate 9500
After the fail over has occurred, the configuration will be:
on Node A (that failed, but then restored and running as stand-by):
local instance of Tivoli endpoint running on port 9497 with alternate 9498
on Node B (that became the production node):
local instance of Tivoli endpoint running on port 9499 with alternate 9500
shared instance of Tivoli endpoint running on port 9495 with alternate 9496

The next step you need to work on is modifying the inittab script (or init rc scripts) on each node to allow both the local and shared instance of Tivoli endpoint to start once the node is rebooted. You will need to have two entries in inittab or two start/stop rc scripts: one for the local instance and one for the shared instance.

One more step is to create a start/stop shell script for the _shared_ instance of Tivoli endpoint, or to modify $LCF_DATDIR\lcfd.sh script to start and stop the endpoint when the fail over occurs. The start/stop script for the Tivoli endpoint must be included in the fail over plan for the clustering software being used. Usually, the fail over plan is made of one or more shell scripts: you will need to include the start/stop scripts for Tivoli endpoint to the plan.

The final step is to configure the logical environment in the Tivoli region to correctly place and subscribe these three endpoints. If there are any mechanisms in use to restrict the allow_login policy in the Tivoli region (for example, having endpoint databases allowed to login), then these will need to be configured too.

## True fail over cluster on Windows

As on UNIX, on Windows nodes the true fail over cluster includes a production node (node A) and a stand-by node (node B). The applications are running on the production node, while the stand-by node is running only a copy of the operating system with the necessary components of clustering software.

The cluster configuration assumes that there is a shared Logical Disk (for example Disk "S:" ) and that all cluster applications are installed on that drive. The production and stand-by nodes have the operating system installed on Disk "C:".

In case of fail over, all shared cluster resources (applications and shared Disk "S:" ) are moved to the stand-by node that becomes the production node.

To support this configuration, you must install the Tivoli endpoint code on each node to local Disk "C:" and then reboot both nodes. For example, install the endpoint into: C:\admin\Tivoli\lcf.
Then you stop the Tivoli endpoint on both nodes and copy all files and directories from C:\admin\Tivoli\lcf from one of the nodes to shared Disk "S:" into S:\admin\Tivoli\lcf.

The next step is to remove or rename on each node the directory C:\admin\Tivoli\lcf (with all files in it).

Then, on each node, you edit all registry keys that refer to C:\admin\Tivoli\lcf and change them to point to the new directory on Disk "S:" (S:\admin\Tivoli\lcf).

After that, you edit on each node some files in the %LCF_DATDIR% directory (S:\admin\Tivoli\lcf\dat\1) on shared Disk "S:" and in the %SystemRoot%\Tivoli\lcf\1 directory (C:\WINNT\Tivoli\lcf\1) on disk "C:". The files that need to be edited are lcf_env.cmd, lcf_env.sh, and last.cfg. You need to change all references from C:\admin\Tivoli\lcf to S:\admin\Tivoli\lcf.

Then, using command net start lcfd or using the GUI, you start the Tivoli endpoint on the production node (where shared Disk "S:" and all cluster resources are loaded).

The final step is to configure the clustering software (for example Microsoft Cluster Services) to include the Tivoli endpoint NT service into the cluster configuration, so that the clustering software will be able to stop the endpoint, move all resources to the stand-by node, and start the endpoint on it when a fail over event occurs.

---

**Load balancing cluster on UNIX**

On UNIX, the load-balanced cluster includes two nodes, both of which are production nodes. Applications are running on both nodes in cluster "packages": package A runs on node A, package B runs on node B. Each "package" usually includes filesystems on a shared DASD and a set of applications.

If a fail over occurs (for example, node A fails) all resources (applications and filesystems) defined in package A are moved to node B and node B runs both packages at the same time until node A is fixed and brought back online. The same happens when node B fails and node A gets both packages to run.

To enable this configuration, you must install two _SHARED_ instances of Tivoli endpoint, include them into the appropriate cluster package and configure them to run on different ports, so that they will not conflict with each other when both packages are running on one node of the cluster. For example:

Node A

file systems on shared DASD:

/opt/Tivoli/lcf_a

/etc/Tivoli/lcf_a

These file systems must be included in package A and in the fail over plan for that package in order to be unmounted, moved and mounted on another node when the fail over occurs.

Node B
file systems on shared DASD:

/private/Tivoli/lcf_b
/etc/Tivoli/lcf_b

These file systems must be included in package B and in the fail over plan for that package in order to be unmounted, moved and mounted on another node when the fail over occurs.

After this, you install instances of Tivoli endpoint on each node to those file systems using any method (for example winstlcf ) with the following lcf options (supplied using the option -D in the command line or in the last.cfg file in $LCF_DATDIR):

Tivoli Endpoint for package A:

1cfd_port=9495
   1cfd_preferred_port=9495
   gateway_port=9494
   1cfd_alternate_port=9496

Tivoli Endpoint for package B:

1cfd_port=9497
   1cfd_preferred_port=9497
   gateway_port=9494
   1cfd_alternate_port=9498

The next step is to create a start/stop shell script for both instances of Tivoli endpoint in each package, or to modify the $LCF_DATDIR\1cfd.sh scripts to start and stop the endpoints when the fail over occurs. The start/stop script for the Tivoli endpoint must be included in the package A and B fail over plans.

One more step you need to work on is modifying the init script (or init rc scripts) on each node to reflect changes made on the production node during the installation of the Tivoli endpoint. This is needed, for example, in the event you will need to reboot the stand-by node (when it is acting as a production node after a fail over) in order to enable the automatic startup of the Tivoli endpoint during boot time.

The final step is modifying the init script (or init rc scripts) on each node to allow both instances of Tivoli endpoint to start upon reboot of any node of the cluster. This is needed, for example, in the event you will need to reboot a node (that is running package A and B until the failed node is repaired) in order to enable the automatic startup of both Tivoli endpoints during boot time.
Limitations

The solution provided by Tivoli Monitoring to support clusters has several limitations. This section describes the major disadvantages:

- The solution requires a complicated setup because the clustering policies must be manually configured.
- The solution may create uncontrolled processes (in UNIX literature typically referred to as zombie processes) because it is difficult to migrate all processes generated on the Tivoli endpoint.
- Having multiple instances of the Tivoli Monitoring Agent running on a single system causes an additional overhead in terms of system resources.
- Both nodes (production and stand-by) may generate events for the Tivoli Enterprise Console, notifying problems on resources hosted by the cluster. Such events carry in their slots references (like the IP address, hostname, and so forth) of the node on which the endpoint engine runs. This means that writing Tivoli Enterprise Console rules becomes more complicated: this is true also for any decision process based on the equivalent information.
- Tivoli Monitoring providers creating long running external processes may not be able to support a migration scenario. To support migration scenarios, these providers must be defined in such a way that they stop themselves when the engine that started them goes away.
- Specifically to data collections for the Tivoli Data Warehouse:
  - A resource model cannot be customized to log data for Tivoli Data Warehouse at the same time on multiple engines sharing the same hostname on the same node.
  - If a virtual endpoint is running a resource model customized to log data for Tivoli Data Warehouse and, in the migration from a node of the cluster to another node, the endpoint changes its hostname, then the data for Tivoli Data Warehouse is stored under different sections in the Tivoli Monitoring RIM database.
  - When an endpoint is migrated to a different node of the cluster, there is usually a loss of data for one or two hours.
Appendix E. Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in Tivoli Monitoring enable users to:

- Use assistive technologies such as screen-reader software and a digital speech synthesizer to hear what is displayed on the screen
- Operate specific or equivalent features using only the keyboard
- Magnify what is displayed on the screen

In addition, the product documentation has been modified to include features to aid accessibility:

- All documentation available in both HTML and convertible PDF formats to give the maximum opportunity for users to apply screen-reader software.
- All images provided with alternative text so that users of the documentation with vision impairments can understand the contents of the images.

Using assistive technologies

Assistive technology products such as screen-readers, function with both the text-based and graphical user interfaces found in Tivoli Monitoring. Consult the assistive technology product documentation for specific information about using it to access command line or graphical interfaces.

Additional accessibility features might be included as part of the user interface of a particular Tivoli Monitoring component. Check with the individual component’s documentation for any additional information about accessibility.

Magnifying what is displayed on the screen

In all components of Tivoli Monitoring other than the Web Health Console, users can magnify the screens used by the product’s user interfaces using facilities provided by the operating systems on which the product is run. For example, in a Windows environment you can change the screen settings to a lower resolution to enlarge the font sizes of the text on the screen. Information about these facilities is provided in the relevant operating system documentation.

Documentation in accessible formats

All user documentation is provided in HTML format, which can be read directly by assistive tools such as screen readers, or in convertible PDF format. Convertible PDF files are those that can be converted from PDF to HTML by the Adobe PDF to HTML converter. For information about converting PDF documents to HTML, refer to the Adobe book *Optimizing Adobe PDF Files for Accessibility*.

Using alternative text

All documentation images are provided with an alternative text that can be read by assistive tools such as screen readers.
Appendix F. Use of Rhino: JavaScript for Java

Rhino is an open-source implementation of JavaScript written entirely in Java. It is typically embedded into Java applications to provide scripting to end users. Its use in Tivoli Monitoring is governed by The Mozilla Public License Version 1.1, given below, as amended by The Netscape Public License Version 1.1, given on page 256.

The Mozilla Public License Version 1.1

1. Definitions.

1.0.1. “Commercial Use” means distribution or otherwise making the Covered Code available to a third party.
1.1. “Contributor” means each entity that creates or contributes to the creation of Modifications.
1.2. “Contributor Version” means the combination of the Original Code, prior Modifications used by a Contributor, and the Modifications made by that particular Contributor.
1.3. “Covered Code” means the Original Code or Modifications or the combination of the Original Code and Modifications, in each case including portions thereof.
1.4. “Electronic Distribution Mechanism” means a mechanism generally accepted in the software development community for the electronic transfer of data.
1.6. “Initial Developer” means the individual or entity identified as the Initial Developer in the Source Code notice required by Exhibit A.
1.7. “Larger Work” means a work which combines Covered Code or portions thereof with code not governed by the terms of this License.
1.8. “License” means this document.
1.8.1. “Licensable” means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.
1.9. “Modifications” means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is:
   A. Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.
   B. Any new file that contains any part of the Original Code or previous Modifications.
1.10. “Original Code” means Source Code of computer software code which is described in the Source Code notice required by Exhibit A as Original Code, and which, at the time of its release under this License is not already Covered Code governed by this License.
1.10.1. “Patent Claims” means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.
1.11. “Source Code” means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control compilation and installation of an Executable, or source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor’s choice. The Source Code can be in a compressed or archival form, provided the appropriate decompression or de-archiving software is widely available for no charge.

© Copyright IBM Corp. 2004
1.12. “You” (or “Your”) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 6.1. For legal entities, “You” includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, “control” means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. Source Code License.

2.1. The Initial Developer Grant.

The Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer to use, reproduce, modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, and/or as part of a Larger Work; and
(b) under Patents Claims infringed by the making, using or selling of Original Code, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Code (or portions thereof).
(c) the licenses granted in this Section 2.1(a) and (b) are effective on the date Initial Developer first distributes Original Code under the terms of this License.
(d) Notwithstanding Section 2.1(b) above, no patent license is granted: 1) for code that You delete from the Original Code; 2) separate from the Original Code; or 3) for infringements caused by: i) the modification of the Original Code or ii) the combination of the Original Code with other software or devices.

2.2. Contributor Grant.

Subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor, to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof) either on an unmodified basis, with other Modifications, as Covered Code and/or as part of a Larger Work; and
(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: 1) Modifications made by that Contributor (or portions thereof); and 2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).
(c) the licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first makes Commercial Use of the Covered Code.
(d) Notwithstanding Section 2.2(b) above, no patent license is granted: 1) for any code that Contributor has deleted from the Contributor Version; 2) separate from the Contributor Version; 3) for infringements caused by: i) third party modifications of Contributor Version or ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or 4) under Patent Claims infringed by Covered Code in the absence of Modifications made by that Contributor.

3. Distribution Obligations.
3.1. Application of License.

The Modifications which You create or to which You contribute are governed by the terms of this License, including without limitation Section 2.2. The Source Code version of Covered Code may be distributed only under the terms of this License or a future version of this License released under Section 6.1, and You must include a copy of this License with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this License or the recipients’ rights hereunder. However, You may include an additional document offering the additional rights described in Section 3.5.

3.2. Availability of Source Code.

Any Modification which You create or to which You contribute must be made available in Source Code form under the terms of this License either on the same media as an Executable version or via an accepted Electronic Distribution Mechanism to anyone to whom you made an Executable version available; and if made available via Electronic Distribution Mechanism, must remain available for at least twelve (12) months after the date it initially became available, or at least six (6) months after a subsequent version of that particular Modification has been made available to such recipients. You are responsible for ensuring that the Source Code version remains available even if the Electronic Distribution Mechanism is maintained by a third party.

3.3. Description of Modifications.

You must cause all Covered Code to which You contribute to contain a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent statement that the Modification is derived, directly or indirectly, from Original Code provided by the Initial Developer and including the name of the Initial Developer in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.

3.4. Intellectual Property Matters

(a) Third Party Claims.
If Contributor has knowledge that a license under a third party’s intellectual property rights is required to exercise the rights granted by such Contributor under Sections 2.1 or 2.2, Contributor must include a text file with the Source Code distribution titled “LEGAL” which describes the claim and the party making the claim in sufficient detail that a recipient will know whom to contact. If Contributor obtains such knowledge after the Modification is made available as described in Section 3.2, Contributor shall promptly modify the LEGAL file in all copies Contributor makes available thereafter and shall take other steps (such as notifying appropriate mailing lists or newsgroups) reasonably calculated to inform those who received the Covered Code that new knowledge has been obtained.

(b) Contributor APIs.
If Contributor’s Modifications include an application programming interface and Contributor has knowledge of patent licenses which are reasonably necessary to implement that API, Contributor must also include this information in the LEGAL file.

(c) Representations.
Contributor represents that, except as disclosed pursuant to Section 3.4(a) above, Contributor believes that Contributor’s Modifications are Contributor’s original creation(s) and/or Contributor has sufficient rights to grant the rights conveyed by this License.
3.5. Required Notices.

You must duplicate the notice in Exhibit A in each file of the Source Code. If it is not possible to put such notice in a particular Source Code file due to its structure, then You must include such notice in a location (such as a relevant directory) where a user would be likely to look for such a notice. If You created one or more Modification(s) You may add your name as a Contributor to the notice described in Exhibit A. You must also duplicate this License in any documentation for the Source Code where You describe recipients’ rights or ownership rights relating to Covered Code. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Code. However, You may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear than any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.6. Distribution of Executable Versions.

You may distribute Covered Code in Executable form only if the requirements of Section 3.1-3.5 have been met for that Covered Code, and if You include a notice stating that the Source Code version of the Covered Code is available under the terms of this License, including a description of how and where You have fulfilled the obligations of Section 3.2. The notice must be conspicuously included in any notice in an Executable version, related documentation or collateral in which You describe recipients’ rights relating to the Covered Code. You may distribute the Executable version of Covered Code or ownership rights under a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable version does not attempt to limit or alter the recipient’s rights in the Source Code version from the rights set forth in this License. If You distribute the Executable version under a different license You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or any Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.7. Larger Works.

You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Code.

4. Inability to Comply Due to Statute or Regulation.

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Code due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be included in the LEGAL file described in Section 3.4 and must be included with all distributions of the Source Code. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Application of this License.

This License applies to code to which the Initial Developer has attached the notice in Exhibit A and to related Covered Code.

6. Versions of the License.
6.1. New Versions.

Netscape Communications Corporation ("Netscape") may publish revised and/or new versions of the License from time to time. Each version will be given a distinguishing version number.

6.2. Effect of New Versions.

Once Covered Code has been published under a particular version of the License, You may always continue to use it under the terms of that version. You may also choose to use such Covered Code under the terms of any subsequent version of the License published by Netscape. No one other than Netscape has the right to modify the terms applicable to Covered Code created under this License.

6.3. Derivative Works.

If You create or use a modified version of this License (which you may only do in order to apply it to code which is not already Covered Code governed by this License), You must (a) rename Your license so that the phrases “Mozilla”, “MOZILLAFL”, “MOZPL”, “Netscape”, “MPL”, “NPL” or any confusingly similar phrase do not appear in your license (except to note that your license differs from this License) and (b) otherwise make it clear that Your version of the license contains terms which differ from the Mozilla Public License and Netscape Public License. (Filling in the name of the Initial Developer, Original Code or Contributor in the notice described in Exhibit A shall not of themselves be deemed to be modifications of this License.)

7. DISCLAIMER OF WARRANTY.

COVERED CODE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED CODE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED CODE IS WITH YOU. SHOULD ANY COVERED CODE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

8. TERMINATION.

8.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. All sublicenses to the Covered Code which are properly granted shall survive any termination of this License. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

8.2. If You initiate litigation by asserting a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You file such action is referred to as "Participant") alleging that:
(a) such Participant’s Contributor Version directly or indirectly infringes any patent, then any and all rights granted by such Participant to You under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively, unless if within 60 days after receipt of notice You either: (i) agree in writing to pay Participant a mutually agreeable reasonable royalty for Your past and future use of Modifications made by such Participant, or (ii) withdraw Your litigation claim with respect to the Contributor Version against such Participant. If within 60 days of notice, a reasonable royalty and payment arrangement are not mutually agreed upon in writing by the parties or the litigation claim is not withdrawn, the rights granted by Participant to You under Sections 2.1 and/or 2.2 automatically terminate at the expiration of the 60 day notice period specified above.

(b) any software, hardware, or device, other than such Participant’s Contributor Version, directly or indirectly infringes any patent, then any rights granted to You by such Participant under Sections 2.1(b) and 2.2(b) are revoked effective as of the date You first made, used, sold, distributed, or had made, Modifications made by that Participant.

8.3. If You assert a patent infringement claim against Participant alleging that such Participant’s Contributor Version directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

8.4. In the event of termination under Sections 8.1 or 8.2 above, all end user license agreements (excluding distributors and releasers) which have been validly granted by You or any distributor hereunder prior to termination shall survive termination.

9. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED CODE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY’S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

10. U.S. GOVERNMENT END USERS.

11. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by California law provisions (except to the extent applicable law, if any, provides otherwise), excluding its conflict-of-law provisions. With respect to disputes in which at least one party is a citizen of, or an entity chartered or registered to do business in the United States of America, any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California, with venue lying in Santa Clara County, California, with the losing party responsible for costs, including without limitation, court costs and reasonable attorneys’ fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License.

12. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

13. MULTIPLE-LICENSED CODE.

Initial Developer may designate portions of the Covered Code as “Multiple-Licensed”. “Multiple-Licensed” means that the Initial Developer permits you to utilize portions of the Covered Code under Your choice of the NPL or the alternative licenses, if any, specified by the Initial Developer in the file described in Exhibit A.

EXHIBIT A -Mozilla Public License.

“The contents of this file are subject to the Mozilla Public License Version 1.1 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.mozilla.org/MPL/
Software distributed under the License is distributed on an “AS IS” basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.
The Original Code is _____________________________.
The Initial Developer of the Original Code is ___________________________. Portions created by _____________ are Copyright (C) _____________ ____________. All Rights Reserved.
Contributor(s): ___________________________.
Alternatively, the contents of this file may be used under the terms of the _____ license (the “[ ] License”), in which case the provisions of [_____] License are applicable instead of those above. If you wish to allow use of your version of this file only under the terms of the [_____] License and not to allow others to use your version of this file under the MPL, indicate your decision by deleting the provisions above and replace them with the notice and other provisions required by the [_____] License. If you do not delete the provisions above, a recipient may use your version of this file under either the MPL or the [_____] License.”

[NOTE: The text of this Exhibit A may differ slightly from the text of the notices in the Source Code files of the Original Code. You should use the text of this Exhibit A rather than the text found in the Original Code Source Code for Your Modifications.]
The Netscape Public License Version 1.1

AMENDMENTS

The Netscape Public License Version 1.1 ("NPL") consists of the Mozilla Public License Version 1.1 with the following Amendments, including Exhibit A-Netscape Public License. Files identified with "Exhibit A-Netscape Public License" are governed by the Netscape Public License Version 1.1.

Additional Terms applicable to the Netscape Public License.

I. Effect.

These additional terms described in this Netscape Public License -- Amendments shall apply to the Mozilla Communicator client code and to all Covered Code under this License.

II. "Netscape’s Branded Code"

Means Covered Code that Netscape distributes and/or permits others to distribute under one or more trademark(s) which are controlled by Netscape but which are not licensed for use under this License.

III. Netscape and logo.

This License does not grant any rights to use the trademarks "Netscape", the "Netscape N and horizon" logo or the "Netscape lighthouse" logo, "Netcenter", "Gecko", "Java" or "JavaScript", "Smart Browsing" even if such marks are included in the Original Code or Modifications.

IV. Inability to Comply Due to Contractual Obligation.

Prior to licensing the Original Code under this License, Netscape has licensed third party code for use in Netscape’s Branded Code. To the extent that Netscape is limited contractually from making such third party code available under this License, Netscape may choose to reintegrate such code into Covered Code without being required to distribute such code in Source Code form, even if such code would otherwise be considered “Modifications” under this License.

V. Use of Modifications and Covered Code by Initial Developer.

V.1. In General.

The obligations of Section 3 apply to Netscape, except to the extent specified in this Amendment, Section V.2 and V.3.

V.2. Other Products.

Netscape may include Covered Code in products other than the Netscape’s Branded Code which are released by Netscape during the two (2) years following the release date of the Original Code, without such additional products becoming subject to the terms of this License, and may license such additional products on different terms from those contained in this License.

V.3. Alternative Licensing.

Netscape may license the Source Code of Netscape’s Branded Code, including Modifications incorporated therein, without such Netscape Branded Code becoming subject to the terms of this License, and may license such Netscape Branded Code on different terms from those contained in this License.
VI. Litigation.

Notwithstanding the limitations of Section 11 above, the provisions regarding litigation in Section 11(a), (b) and (c) of the License shall apply to all disputes relating to this License.

EXHIBIT A-Netscape Public License.

“The contents of this file are subject to the Netscape Public License Version 1.1 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.mozilla.org/NPL/

Software distributed under the License is distributed on an “AS IS” basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is Mozilla Communicator client code, released March 31, 1998.
The Initial Developer of the Original Code is Netscape Communications Corporation.
Portions created by Netscape are Copyright (C) 1998-1999 Netscape Communications Corporation. All Rights Reserved.

Contributor(s): _______________________________________.

Alternatively, the contents of this file may be used under the terms of the ______ license (the “[___] License”), in which case the provisions of [_____] License are applicable instead of those above. If you wish to allow use of your version of this file only under the terms of the [_____] License and not to allow others to use your version of this file under the NPL, indicate your decision by deleting the provisions above and replace them with the notice and other provisions required by the [___] License. If you do not delete the provisions above, a recipient may use your version of this file under either the NPL or the [___] License.”
Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user’s responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.
IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM’s application programming interfaces.

If you are viewing this information in softcopy form, the photographs and color illustrations might not appear.
IBM, the IBM logo, Tivoli, the Tivoli logo, AIX, Cloudscape, DB2, OS/390, S/390, OS/2, OS/400, iSeries, Informix, IBMLink, NetView, WebSphere, Tivoli Management Environment, Tivoli Enterprise, Tivoli Enterprise Console, and Tivoli Enterprise Data Warehouse are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Microsoft, Windows, and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.
Glossary

A

Adapter Configuration Facility. In the Tivoli Enterprise Console, a graphical user interface that enables a Tivoli administrator to easily configure and customize event adapters.

authorization. (1) In computer security, the right granted to a user to communicate with or make use of a computer system. (2) The process of granting a user either complete or restricted access to an object, resource, or function.

authorization role. A role that is assigned to administrators that enables them to perform their assigned systems management tasks. In a Tivoli environment, a role can be granted over the entire Tivoli management region or over a specific set of resources, such as those resources that are contained in a policy region. Examples of authorization roles include super, senior, admin, and user.

B

Basic Recorder of Objects in C (BAROC). The internal representation of the defined event classes at the event server.

C

cache. A buffer storage that contains frequently accessed instructions and data; it is used to reduce access time.

configuration file. A file that specifies the characteristics of a system device or network.

D

data collection. The process of obtaining performance and availability monitoring data and providing that data to a metric evaluator. Examples of data collectors include DNS probes, Web page analyzers, or database analyzers.

data mart. A subset of a data warehouse that contains a data that is tailored and optimized for the specific reporting needs of a department or team. A data mart can be a subset of a warehouse for an entire organization, such as data contained in online analytical processing (OLAP) tools.

data warehouse. (1) A subject-oriented nonvolatile collection of data that is used to support strategic decision making. The warehouse is the central point of data integration for business intelligence. It is the source of data for data marts within an enterprise and delivers a common view of enterprise data. (2) A central repository for all or significant parts of the data that an organization’s business system collects. Also known as an information warehouse. See also data mart.

default policy. In a Tivoli environment, a set of resource property values that are assigned to a resource when the resource is created.

depot. A directory that enables the temporary or permanent storage of data segments.

discovery. The automatic detection of a topology change, such as finding new and deleted nodes or links within a network topology, or such as finding storage resources and devices within a network that are not yet being monitored.

E

endpoint. In a Tivoli environment, a Tivoli client that is the ultimate recipient for any type of Tivoli operation.

endpoint list. In a Tivoli environment, a list of all endpoints in a Tivoli region with their assigned gateways.

ETL. See extract, transform, and load.

event. Any significant change in the state of a system resource, network resource, or network application. An event can be generated for a problem, for the resolution of a problem, or for the successful completion of a task. See also indication.

event class. A classification for an event that indicates the type of information that the event adapter can send to the event server. See also Basic Recorder of Objects in C.

event correlation. The process of analyzing event data to identify patterns, common causes, and root causes. Event correlation analyzes the incoming events for predefined states, using predefined rules, and against predefined relationships.

extract, transform, and load (ETL). The process of collecting data from one or more sources, cleansing and transforming the data, and then loading the data into a database.
F

fragmentation. An operating system's process of writing different parts of a file to discontiguous sectors on a computer storage medium when contiguous space that is large enough to contain the entire file is not available. When data is thus fragmented, the time that it takes to access the data may increase because the operating system must search different tracks for information that should be in one location.

G

gateway. In a Tivoli environment, software running on a managed node that provides all communication services between a group of endpoints and the rest of the Tivoli environment. This gateway includes the multiplexed distribution (MDist) function, enabling it to act as the fanout point for distributions to many endpoints.

I

IBM Tivoli Monitoring. A Tivoli application that applies preconfigured, automated best practices to the automated monitoring of essential system resources. The application detects bottlenecks and other potential problems and provides for the automatic recovery from critical situations, which eliminates the need for system administrators to manually scan through extensive performance data. The application also integrates seamlessly with other Tivoli Availability solutions, including the Tivoli Business Systems Manager and the Tivoli Enterprise Console. Previously called Tivoli Distributed Monitoring for Windows.

indication. A problem on an endpoint that involves one or more resources. Indications are consolidated into events on the endpoint that is being monitored. See also event.

installation repository (IR). In Tivoli Software Installation Service (SIS), the directory that contains reusable installation images and other data that is used by SIS.

M

managed node. In a Tivoli environment, any managed resource on which the Tivoli Management Framework is installed.

Management Console. In IBM Tivoli Monitoring, a component that displays real-time and historical data for any resource model at any endpoint. Using the graphical user interface, users can locate individual problems associated with one or more resources. The status is displayed as a value between 0 (representing an identified problem, that is, an event) and 100 (representing no recent indications). Users can select views of resource problems as tabular data, different types of charts, and so on.

monitoring collection. In Tivoli Distributed Monitoring (Classic Edition), a collection of predefined monitors. Several monitoring collections are packaged with Tivoli Distributed Monitoring, but Tivoli administrators can also use custom-developed and third-party monitoring collections.

N

notice. In a Tivoli environment, a message that is generated by a systems management operation that contains information about an event or the status of an application. Notices are stored in notice groups.

notice group. In a Tivoli environment, an application- or operation-specific container that stores and displays notices that pertain to specific Tivoli functions. The Tivoli bulletin board is comprised of notice groups.

O

object. In object-oriented design or programming, a concrete realization of a class that consists of data and the operations associated with that data.

oserv. The Tivoli service that is used as the object request broker (ORB). This service runs on the Tivoli server and each managed node.

P

policy. A set of rules that are applied to managed resources.

policy region. In a Tivoli environment, a group of managed resources that share one or more common policies. Tivoli administrators use policy regions to model the management and organizational structure of a network computing environment. The administrators can group similar resources, define access to and control the resources, and associate rules for governing the resources.

policy subregion. In a Tivoli environment, a policy region created or residing in another policy region. When a policy subregion is created, it initially uses the resource and policy properties of the parent policy region. The Tivoli administrator can later change or customize these properties to reflect the specific needs and differences of the subregion.

portal. An integrated Web site that dynamically produces a customized list of Web resources, such as link, content, or services, available to a specific user, based on the access permissions for the particular user.
portlet. An area of content on a portal Web page that has a predefined role, such as retrieving the latest news headlines, driving a search engine, searching a database, viewing stock quotes, serving HTML files, or displaying a calendar.

profile. In a Tivoli environment, a container for application-specific information about a particular type of resource. A Tivoli application specifies the template for its profiles, which includes information about the resources that the Tivoli application can manage.

A profile is created in the context of a profile manager; the profile manager links a profile to the Tivoli resource (for example, a managed node) that uses the information contained in the profile. A profile does not have any direct subscribers.

profile manager. In a Tivoli environment, a container for profiles that links the profiles to a set of resources, called subscribers. Tivoli administrators use profile managers to organize and distribute profiles. A profile manager is created in the context of a policy region and is a managed resource in a policy region.

proxy endpoint. In a Tivoli environment, a representation for an entity (such as a network device or a host) that functions as a subscriber for profiles. The proxy endpoint is created on a managed node, which performs the proxy role during profile distribution. Multiple proxy endpoints can be created on the same managed node.

R

RDBMS. See relational database management system.

RDBMS interface model (RIM). The module in the distributed object database that contains information about the installation of the relational database management system (RDBMS).

relational database management system (RDBMS). A collection of hardware and software that organizes and provides access to a relational database.

reference model. In the context of Tivoli software, the model configuration for a system or set of systems that is used to maintain consistent configurations in a distributed environment.

repeater. In a Tivoli environment, a managed node that receives a single copy of data and distributes it to the next tier of clients.

resource. A hardware, software, or data entity that is managed by Tivoli software. See also managed resource.

resource model. In IBM Tivoli Monitoring, the logical modeling of one or more resources, along with the logic on which cyclical data collection, data analysis, and monitoring are based. Related events and actions are triggered, if required. For any resource model, users can specify individual thresholds and event aggregation rules. See also event.

resource model engine (RME). In IBM Tivoli Monitoring, an analysis engine that is used to identify, notify, and cure performance and availability problems. The RME analyzes performance data that is collected from physical resources and uses that data to identify a problem, then triggers corrective actions to cure the discovered problem, and finally escalates problem notification to management tools.

RIM. See RDBMS Interface Module.

RIM host. In a Tivoli environment, the managed node on which one or more RIM objects are installed. See also RIM object.

RIM object. An object that provides the attributes and methods that enable applications to access an RDBMS.

RIM repository. In a Tivoli environment, a relational database that contains information that is collected or generated by Tivoli applications. Examples of a RIM repository include the configuration repository and the event database.

RME. See resource model engine.

rule. A set of logical statements that enable the event server to recognize relationships among events and to execute automated responses accordingly.

rule base. In the Tivoli Enterprise Console, one or more rule sets and the event class definitions for which the rules are written. The Tivoli Enterprise Console uses the rule base in managing events. An organization can create many rule bases, with each rule base fulfilling a different set of needs for network computing management.

S

schema. The set of statements, expressed in a data definition language, that completely describes the structure of a database. In a relational database, the schema defines the tables, the fields in each table, and the relationships between fields and tables.

Software Installation Service (SIS). A Tivoli product that provides an easy-to-use, efficient interface for installing Tivoli Enterprise software. SIS uses Tivoli’s MDist technology and provides automated checking for prerequisite software, a reusable repository of installation images, and both graphical and command line interfaces for deploying Tivoli products to a large number of computers.

subscriber. In a Tivoli environment, a managed node, a profile manager, an endpoint, or another Tivoli client that is subscribed to a profile manager. Although
profiles are distributed to a subscriber, the subscriber may or may not be the final destination of the profile distribution.

**T**

**task library.** In a Tivoli environment, a container in which a Tivoli administrator can create and store tasks and jobs.

**threshold.** A customizable value for defining the acceptable tolerance limits (maximum, minimum, or reference limit) for an application resource or system resource. When the measured value of the resource is greater than the maximum value, less than the minimum value, or equal to the reference value, an exception is raised.

**Tivoli desktop.** In the Tivoli environment, the desktop that system administrators use to manage their network computing environments.

**Tivoli Distributed Monitoring.** Previous name of Tivoli Distributed Monitoring (Classic Edition).

**Tivoli Distributed Monitoring (Advanced Edition).** Previous name of IBM Tivoli Monitoring.

**Tivoli Distributed Monitoring (Classic Edition).** A Tivoli application that provides distributed monitors for monitoring system resources. The application initiates necessary corrective actions and informs system administrators of potential problems. These monitors can be centrally configured and deployed to monitor individual machines. The application also integrates seamlessly with other Tivoli Availability solutions, including the Tivoli Enterprise Console. Previously called Tivoli Distributed Monitoring.

**Tivoli Distributed Monitoring for Windows.** Previous name of Tivoli Distributed Monitoring (Advanced Edition).

**Tivoli Enterprise Console.** A Tivoli product that collects, processes, and automatically initiates corrective actions for system, application, network, and database events; it is the central control point for events from all sources. The Tivoli Enterprise Console provides a centralized, global view of the network computing environment; it uses distributed event monitors to collect information, a central event server to process information, and distributed event consoles to present information to system administrators.

**Tivoli environment.** The Tivoli applications, based upon the Tivoli Management Framework, that are installed at a specific customer location and that address network computing management issues across many platforms. In a Tivoli environment, a system administrator can distribute software, manage user configurations, change access privileges, automate operations, monitor resources, and schedule jobs.

**Tivoli management agent.** In the Tivoli environment, an agent that securely performs administrative operations.

**Tivoli Management Framework.** The base software that is required to run the applications in the Tivoli product suite. This software infrastructure enables the integration of systems management applications from Tivoli Systems Inc. and the Tivoli Partners.

In a Tivoli environment, the Tivoli Management Framework is installed on every client and server; however, the Tivoli server of the region is the only server that holds the full object database.

**Tivoli region.** In a Tivoli environment, a Tivoli server and the set of clients (managed nodes and endpoints) that it serves. An organization can have more than one Tivoli region. A Tivoli region addresses the physical connectivity of resources whereas a policy region addresses the logical organization of resources.

**Tivoli server.** The server for a specific Tivoli region that holds or references the complete set of Tivoli software, including the full object database.

**W**

**warehouse enablement pack.** A separately installable part of a Tivoli software product that provides Tivoli Data Warehouse functionality. The warehouse enablement pack provides extract, transform, and load programs to populate the central data warehouse and to create data marts as well as customizable reports to answer specific business questions. Often referred to as a warehouse pack.
Index

A
accessibility 247
actions (recovery), concept 10
actions, built-in
adding 50
editing 49
removing 50
Adapter Configuration Facility, as prerequisite for Tivoli Enterprise Console server 22
assured delivery, MDist 2 60
asynchronous delivery, MDist 2 60

B
backward compatibility 23
books
  online xii
ordering xii
built-in actions
adding 50
concept 10
editing 49
removing 50
bulk data transfer 15

C
cache, endpoint 74
check-point and restart, MDist 2 60
clearing events, concept 8
clusters
  limitations of the Tivoli Monitoring solution 246
load balancing on UNIX nodes 244
ture fail over on UNIX nodes 241
ture fail over on Windows nodes 243
coexistence 22
collecting the data 14
commands
  commands (continued)
    return codes from 169
    upgraded names 109
    wdmcheckprereq 112
    wdmcmd 113
    wdmcmddistrib 115
    wdmcollect 119
    wdmconfig 121
    wdmdiscovery 127
    wdmddistrib 129
    wdmddumpprf 132
    wdmeditprf 138
    wdmeng 146
    wdmheartbeat 148
    wdmloadprf 150
    wdmmseng 153
    wdmnn 157
    wdmnngcache 159
    wdmrm 162
    wdmrcreng 164
    winstall 29
    wtntaddr 166
    wtntdefm 167
    wtntmrmm 168
  compatibility 23
  components
    diagram 11
    endpoint 13
    Gathering Historical Data 13
    IBM Tivoli Monitoring Base
    Component 12
    installation 26
    introduction 11
    Tivoli Business Systems Manager
    Adapter 13
    Tivoli Enterprise Data Warehouse
    Support 13
    upgrading 30
    Web Health Console 13
  concepts
    actions (recovery) 10
    built-in actions 16
    clearing events 8
    correlated events 8
    cycles 4
    data logging 11
    event monitors 9
    events 6
    heartbeat function 11
    holes 6
    indications 5
    logging (data) 11
    monitors 9
    occurrences 6
    parameters 5
    profiles 11
    recovery actions 10
    resource models 4
    resources 4
    scheduling 10
  concepts (continued)
    tasks (Tivoli Management Framework) 10
    thresholds 4
    configuration
      heartbeat function 69
      Tivoli Business Systems Manager
      Adapter 86
    customizing profiles and resource models 39
    cycles, concept 4

d
data collection 14
data depotting, MDist 2 60
data flow
  resource models 14
  Tivoli Business Systems Manager
  CommonListener, events 15
  Tivoli Enterprise Console server, events 15
  Web Health Console, events 15
data logging
  concept 11
  customizing 56
desktop, installing product from 26
directory names, notation xv
distribution 58

e
endpoint
  cache viewing 74
  collecting the data at 14
  component 13
  data collection 14
  obtaining resource data 15
  profile management 63
  resource model management 13
  uninstalling product 34
  environment variables 203
  environment variables, notation xv
  event monitors
    concept 9
    Tivoli Business Systems Manager 10
    Tivoli Enterprise Console server 9
    Web Health Console 9
  events
    clearing 8
    concept 6
    correlated 8
    customizing 47
data flow
    Tivoli Business Systems Manager
    CommonListener 15
    Tivoli Enterprise Console
    server 15
    Web Health Console 15
events (continued)
sending a notice in response 53
sending an e-mail in response 52
events (clearing), concept 8

F
firewalls 15

G
gateway
managing 63
uninstalling product from 37
Gathering Historical Data Component description 13

H
hardware requirements
IBM Tivoli Monitoring 18
heartbeat function
concept 11
control 70
data flows 71
description 69
endpoint registration 71
installation 69
managing 70
monitoring 73
statuses 69
viewing the endpoint cache 74
holes
concept 6
customizing 47, 48

I
IBM Software Support xv
IBM Tivoli Monitoring
Base component 12
coeexistence 22
hardware requirements 18
installation 26
notice group addition 31
overview 3
software requirements 19
uninstalling 34
indications
concept 5
customizing 47
installation
before beginning 25
creating a Windows proxy endpoint 213
default resource models 30
heartbeat function 69
Java Runtime Environment 31
of product from the Tivoli Desktop 26
of product using commands 29
of product using installation wizard 205
evaluation installation 205
typical installation 208
installtion (continued)
of product with installation wizard 25
of product with Software Installation Service 25
Tivoli Business Systems Manager Adapter 85
Tivoli Enterprise Data Warehouse Support 99
installation wizard
installing product with 25

J
Java Runtime Environment
installation 31
supported versions 20

L
logging (data)
concept 11
customizing 56

M
managed resources, setting up 40
manuals
online xii
ordering xii
MDist 2
assured delivery 60
asynchronous delivery 60
check-point and restart 60
data deploting 60
using 59
migration
from Tivoli Distributed Monitoring (Classic Edition) 217
from Tivoli Web Component Manager 229
monitors
customizing 39
diagram 11
Tivoli Business Manager 10
Tivoli Enterprise Console server 9
Web Health Console 9

N
notation
environment variables xv
path names xv
style notation xv
notice group
IBM Tivoli Monitoring
adding 31

O
occurrences
concept 6
customizing 48
online publications xiv

P
parameters
concepts 5
customizing 54
path names, notation xv
platforms supported 17
port numbers for endpoints 216
preinstallation notes 25
prerequisites
hardware 18
software 19
profile manager
creating 41
for rerunning failed distributions 60
profiles
adding customized resource models 45
adding default resource models 43
concept 11
creating 41
customizing 39
deleting profile copies during an unsubscribe 61
description 40
distributing using MDist 2 59
key points 40
managing at endpoint 63
using commands to manage at the server 43
publications
online xii
ordering xii

R
recovery actions, concept 10
removing
default resource models 30
request manager processor 176
Resource Model Builder 16
resource models
concept 4
creating with Resource Model Builder 16
customized, adding to profiles 45
customizing 39, 46
data flow 14
default
adding to profiles 43
installing 30
removing 30
using commands to manage at the server 43
installing default 30
managing at endpoint 63
modifying with Resource Model Builder 16
removing default 30
resources
count 4
obtaining data at endpoint 15
rules, scheduling 55

S
scheduling
count 10
customizing 54
rules 55
Secure Socket Layer 3 15
security considerations 15
server
uninstalling product from 37
Software Installation Service
installing product with 25
software requirements
IBM Tivoli Monitoring 19
SSL 15
statuses, heartbeat function 69
supported platforms 17

T
tasks (Tivoli Management Framework)
count 10
specifying 50
tasks (Tivoli Monitoring)
DMCleanDB 64
DMCollectEpEnv 64
DMCollectEpLog 64
DMCollectMnLog 64
DMCompressDB 65
DMCreateRuleAndLoadBaroc 65
DMEndpointRemoveFiles 63
DMEndpointUninstall 64
DMGetCloudscapeNetworkServerProperties 65
DMLinkJRE 64
DMPurge_ITM_RIM_DB 65
DMRebootUninstall 64
DMRemoveLinkJRE 64
DMRemoveQuadcapDB 64
DMSetCloudscapeNetworkServerProperties
running from the CLI 66
running from the Tivoli Desktop 66
using 63
thresholds
count 4
customizing 47
timing rules 55
Tivoli Business Systems Manager
as monitor, concept 10
description 85
integration
disabling 92
enabling 92
overview 85
viewing data 90
Tivoli Business Systems Manager Adapter
count 13
configuration 86
hardware requirements 19
installation 26, 85
managing 93
using 89
Tivoli Business Systems Manager
CommonListener, data flow for
events 15
Tivoli Data Warehouse
component description 13
customer scenario using Tivoli
Monitoring 97
enabling data logging for 97
enabling Tivoli Monitoring to work
with 34
integration with 95
overview 95
Tivoli Desktop, installing product
from 26
Tivoli Distributed Monitoring (Classic Edition)
coexistence 22
Tivoli Distributed Monitoring for
Windows
coexistence 22
Tivoli Distributed Monitoring Resource
Model Builder 16
Tivoli Enterprise Console server
as monitor, concept 9
data flow for events 15
enabling of events 78
enabling of heartbeat messages 78
identifying to the product 77
integration 77
Tivoli Enterprise Data Warehouse
Support component
installation and configuration 99
Tivoli Management Framework
firewalls 15
software requirements 19
tasks, concept 10
Tivoli Monitoring, coexistence 22

U
uninstalling
from endpoint 34
from gateway 37
from server 37
product, overview 34
upgrading
from Workbench to Resource Model
Builder 31
instructions 30

V
variables, notation for xv

W
wdmcheckprereq 112
wmdcmd 113
wmdcmddistrib 115
wmdcollect 119
wmdconfig 121
wmdiscovery 127
wmdistrib 129
wmdumpprf 132
wmeditprf 138
wmdeng 146
wdmheartbeat 148
wmdloadprf 150
wmdlseng 153
wmmm 157
wddmngcache 159
wddmrm 162
wdmtrceng 164
Web Health Console
as monitor, concept 9
component 13
data flow, events 15
enabling it to use your resource
models 183
installing 178
introduction to 175
Secure Socket Layer 183
software requirements 177
system requirements 177
troubleshooting the installation 180
using it 187
who should read this guide xi
winstall 29
wmtntaddrm 166
wmtntdefrm 167
wmtntrmrm 168
wmtnttrrmm 168