# Contents

## About these Release Notes
- About these Release Notes ........................................... v
- What is new in this edition of the Release Notes . . v
- Publications ................................................................... v
- Accessing publications online ........................................ vi
- Ordering publications ....................................................... vi
- Tivoli technical training ................................................... vi
- Support information ....................................................... vi
- Conventions used in this book .......................................... vi
- Typeface conventions ....................................................... vii
- Operating system-dependent variables and paths vii

## Chapter 1. About this product release
- Interoperability and compatibility notes  .......................... 1

## Chapter 2. Installation and upgrade notes
- Where to find the resource models ................................. 3
- Supported platforms ....................................................... 3
- System requirements ...................................................... 5
- Software requirements ................................................... 5
- HP-UX 11.23 IA64 requirements ...................................... 5
- AIX requirements .......................................................... 6
- Linux requirements ......................................................... 6
- Tivoli Management Framework requirements .................... 6
- Tivoli Data Warehouse requirements ............................... 6
- Supported Databases ..................................................... 6
- JRE levels ..................................................................... 6
- Supported Configurations ............................................... 7
- Upgrade notes ............................................................... 7
- AMX and AMY post-installation steps ............................... 7

## Chapter 3. Software limitations and problems
- Software limitations ....................................................... 9
- Tivoli Monitoring engine and general functions .................. 9
- Tivoli Enterprise Console Server ..................................... 10
- ETLs ........................................................................ 10
- Resource Models .......................................................... 12
- Software problems and workarounds ............................... 13
- Other considerations .................................................... 21
- Fixed APARs .................................................................. 22

## Chapter 4. Documentation notes
- New information .......................................................... 27
- About the Web Health Console ....................................... 27
- About resource models ................................................ 27
- About Heartbeat process ............................................... 30
- About Tivoli Monitoring functions .................................. 38
- How Tivoli Monitoring is affected by recent changes in Daylight Saving Time ................................................. 56
- Integration between Tivoli Monitoring 5.1.x and Tivoli Monitoring 6.1 .......................................................... 56
- Documentation problems and corrections ....................... 57

## Chapter 5. Internationalization notes
- Software limitations ....................................................... 65
- Enabling language support ............................................. 65
- Updating Web Health Console Files ............................... 66
- Software problems and workarounds ............................... 66
- Rapid Deployment notes ................................................ 67

## Support information
- Searching knowledge bases .......................................... 69
- Searching the information center ................................. 69
- Searching the Internet .................................................. 69
- Obtaining fixes ............................................................ 69
- Receiving weekly support updates ................................. 70
- Contacting IBM Software Support ................................. 71
- Determining the business impact ................................. 71
- Describing problems and gathering information ............... 72
- Submitting problems .................................................. 72

## Notices
- Trademarks ............................................................... 75
About these Release Notes

IBM® Tivoli® Monitoring allows you to monitor availability and performance status of resources on your systems to identify bottlenecks and potential resource problems.

This Release Notes document provides late-breaking information about this product. The information here provided takes precedence over all other documentation. Please review these notes carefully before installing or using this product.

Updated versions of the Release Notes might be placed from time to time on the Tivoli Information Center of the Customer Support Web site at:

http://publib.boulder.ibm.com/tividd/td/link/tdprodlist.html

What is new in this edition of the Release Notes


All changes to this document, with respect to the June edition, are indicated with change bars to the left.

The following sections have been updated:

- Information about JRE levels and fix pack installation steps have been modified in "JRE levels" on page 6.
- New parameters for the wdmepconfig command have been added in Chapter 4, "Documentation notes," on page 27.
- New options for the wdmrm command have been added in Chapter 4, "Documentation notes," on page 27.
- A new parameter for Datacollector is in "Configuring data collection" on page 39.
- New limitations have been added under "Software limitations" on page 9.

Publications

This section describes how to access Tivoli publications online, and how to order Tivoli publications.

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://www.ibm.com/software/tivoli/library/

Click Tivoli product manuals. In the Tivoli Technical Product Documents Alphabetical Listing window, click <Your Product Library Name> to access your product library at the Tivoli software information center.
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.

The Tivoli Software Glossary includes definitions for many of the technical terms related to Tivoli software. The Tivoli Software Glossary is available at the following Web site: [http://publib.boulder.ibm.com/tividd/glossary/tivoliglossarymst.htm](http://publib.boulder.ibm.com/tividd/glossary/tivoliglossarymst.htm)

The IBM Terminology Web site consolidates the terminology from IBM product libraries in one convenient location. You can access the Terminology Web site at the following Web address: [http://www.ibm.com/software/globalization/terminology](http://www.ibm.com/software/globalization/terminology)

**Ordering publications**


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, contact your software account representative to order Tivoli publications.

**Tivoli technical training**

For Tivoli technical training information, refer to the following IBM Tivoli Education Web site:


**Support information**

If you have a problem with your IBM software, you want to resolve it quickly. IBM provides the following ways for you to obtain the support you need:

- **“Searching knowledge bases” on page 69** You can search across a large collection of known problems and workarounds, Technotes, and other information.

- **“Obtaining fixes” on page 69** You can locate the latest fixes that are already available for your product.

- **“Contacting IBM Software Support” on page 71** If you still cannot solve your problem, and you need to work with someone from IBM, you can use a variety of ways to contact IBM Software Support.

For more information about these three ways of resolving problems, see **“Support information” on page 69**

**Conventions used in this book**

This book uses several conventions for special terms and actions, operating system-dependent commands and paths, and margin graphics.
Conventions Used in This Guide

Typeface conventions
This guide uses the following typeface conventions:

**Bold**
- Lowercase commands and mixed case commands that are otherwise difficult to distinguish from surrounding text
- Interface controls (check boxes, push buttons, radio buttons, spin buttons, fields, folders, icons, list boxes, items inside list boxes, multicolumn lists, containers, menu choices, menu names, tabs, property sheets), labels (such as Tip:, and **Operating system considerations:**)
- Keywords and parameters in text

*Italic*
- Words defined in text
- Emphasis of words (words as words)
- New terms in text (except in a definition list)
- Variables and values you must provide

**Monospace**
- Examples and code examples
- File names, programming keywords, and other elements that are difficult to distinguish from surrounding text
- Message text and prompts addressed to the user
- Text that the user must type
- Values for arguments or command options

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.
Conventions Used in This Guide
Chapter 1. About this product release

IBM Tivoli Monitoring, version 5.1.2 (hereafter also referred to as Tivoli Monitoring) is a development of, and upgrades, Tivoli Monitoring Version 5.1.1 and Tivoli Distributed Monitoring (Advanced Edition) Version 4.1.

Using Tivoli Monitoring you can apply pre-configured best practices which encapsulate system administrator expertise to:

- Automate the task of monitoring essential system resources
- Detect bottlenecks and potential problems
- Automatically recover from critical situations without system administrators scanning through extensive performance data

Tivoli Monitoring also provides the foundation for additional automated best practices for monitoring business critical hardware and software, including:

- Middleware
- Applications
- Databases

In addition, Tivoli Monitoring provides seamless integration with other Tivoli availability solutions, including the Tivoli Business System Manager®, the Tivoli Enterprise Console®, and the Tivoli Data Warehouse® to provide a true end-to-end availability solution.

Interoperability and compatibility notes

For information about interoperability and compatibility refer to the installation chapter of the Tivoli Distributed Monitoring: User's Guide, sections "Coexistence with other versions of Tivoli Monitoring" and "Backward Compatibility".

In addition, a matrix is available showing which "IBM Tivoli Monitoring for ..." products are certified to run in an environment with multiple endpoints running concurrently on a single Windows system.

The matrix is available on the following web site:

https://www-111.ibm.com/software/support/ecare/support_login.jsp
Chapter 2. Installation and upgrade notes

Information about product installation is included in the installation chapter of the *IBM Tivoli Monitoring: User’s Guide*, where instructions are provided on how to install the product using any of the following methods:

- Installation wizard
- Tivoli Software Installation Service
- Tivoli Desktop
- Command Line Interface

**Note:** Some HP-UX operating systems might fail to find long file names on the installation CDs for Tivoli Monitoring. These systems are unable to process the Rock Ridge extensions used to burn the Tivoli Monitoring CDs. You must change how the operating system mounts CDs. See the Hewlett-Packard Company Web site for information. The Patch Database in the Hewlett-Packard IT Resource Center has information for some versions and the release notes for HP-UX 10.x systems also provide information.

Where to find the resource models

Tivoli Monitoring provides a set of sample resource models for each supported platform. These resource models (which are loaded as part of the Tivoli Monitoring installation process) are available in the Tivoli Monitoring Tools CD in the RMB directory.

The RMB directory contains resource models in the format used by the Tivoli Monitoring Resource Model Builder. The directory is structured as follows:

- RMB/Samples/OS400 contains the resource models for OS/400
- RMB/Samples/Unix contains the resource models for UNIX
- RMB/Samples/Windows contains the resource models for Windows

Supported platforms

Information about the supported platforms is included in the Installation chapter of the *IBM Tivoli Monitoring: User’s Guide*.

However, Table 5 in that section must be considered revised as follows:

<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>5.1.0 (1) (4), 5.2 (1) (4), 5.3 (9)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Solaris</td>
<td>2.8, 2.9 (3) (4) (9), 2.10 (11)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Solaris x86-64 (AMD 64 and EM64T)</td>
<td>2.10 (12)</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Solaris SPARC</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Windows 2000</td>
<td>Server, Advanced Server, Professional, DataCenter Svr sp3+</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Windows Server 2003</td>
<td>Standard, Enterprise, DataCenter</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Windows XP</td>
<td>Professional</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>System</td>
<td>Versions</td>
<td>Server</td>
<td>Gateway</td>
<td>Endpoint</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>-----------------------------------</td>
<td>--------</td>
<td>---------</td>
<td>----------</td>
</tr>
<tr>
<td>Windows Vista 32-bit(14)</td>
<td>1.0</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SuSE</td>
<td>8.0, 8.1</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>SuSE SLES</td>
<td>7.0</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>SuSE SLES</td>
<td>10.0(12)</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SuSE SLES</td>
<td>7.0 for OS/390 and z/Series (5)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>HP-UX(1)</td>
<td>11, 11i (3), 11.23, and 11.3</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>HP-UX(1)</td>
<td>11.3i</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>OS/400</td>
<td>5.2(6), 5.3(13), 5.4, 6.1 (15)</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>RedHat for iSeries(8)</td>
<td>7.1</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>RedHat Server (IA32)</td>
<td>7.0, 7.1, 7.2, and 7.3</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>RHEL (IA32)</td>
<td>2.1, 3.0</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>RHEL for PowerPC(6)</td>
<td>3.0, 4.0</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>RHEL for z/Series</td>
<td>3.0, 4.0</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>RHEL for Intel(9)</td>
<td>4.0 (10)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>RedHat for Intel(10)</td>
<td>7.3, 8.0</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>UL (SLES 8)</td>
<td>1.0 for IA32 sp2+</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>UL (SLES 8)</td>
<td>1.0 for z/Series sp2+ (3) (4)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>UL (SLES 8)</td>
<td>1.0 for PowerPC</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>UL (SLES 9)</td>
<td>2.0 for IA32 sp2+</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>UL (SLES 9)</td>
<td>2.0 for PowerPC</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>UL (SLES 9)</td>
<td>2.0 for z/Series (3) (4)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Windows 2008(16)</td>
<td>1.0</td>
<td>X</td>
<td>X</td>
<td>X</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 in “toleration” mode.
- (5) Servers and gateways support these platforms only if based on Tivoli Management Framework 4.1.
- (6) Prior to running Tivoli Monitoring on OS/400 5.2, ensure that PTF SI12625 for product 5722S51 is installed. Tivoli Monitoring cannot run without this PTF.
- (7) On endpoints, this version requires Maintenance Level 4.
- (8) Prior to starting the Tivoli Monitoring engine on endpoints running Solaris 9, ensure that libucb.so.1 library is installed.
- (9) Prior to starting the Tivoli Monitoring engine on endpoints running AIX 5.3, ensure that the bos level installed is bos.mp 5.3.0.22, bos.mp64 5.3.0.22, or higher, and that AIX APAR IY70159 is installed. If it is not installed, the engine stops when trying to print the trace, due to an error during libDMXNtvtrace library loading.
- (10) It does not support WebSphere Application Server, and, consequently, it does not support the Web Health Console.
- This zone has its own IP address, which is not shared with another local zone or global zone.
- The file system containing the LCF installation directories is private to the zone and not shared with another local zone or the global zone. All the other parts of the file tree used by the local zone can be either shared or private, including the portion used by endpoint.

- It requires JRE 1.4.2 or higher.
- Tivoli Monitoring engines in localized versions might fail to start when running on OS/400 5.3 (or higher) systems with an lcf higher than 41134. This is due to a Tivoli Management Framework problem, already addressed and fixed with the patch 4.1.1-LCF-0049.
- It does not support TMW_ParamPorts resource model.
- IBM Tivoli Monitoring Fix pack 16 now supports OS/400 version 6.1.
- IBM Tivoli Monitoring Fix pack 16 will support Windows 2008.

For a complete list of supported platforms, refer to the following Web site: 

System requirements

Information about hardware requirements is included in the Installation chapter of the IBM Tivoli Monitoring: User’s Guide.

Note: On endpoints running HP-UX, if an OutOfMemory Java™ exception shows up in the trace_dmxengine.log, and if the OutOfMemory stack trace indicates that the error is related to native thread creation, then the HP kernel max_thread_proc should be increased, and the other thread related parameters should be modified accordingly based on HP’s kernel thread tuning recommendations. When the symptom is a java.lang.OutOfMemory error, check that the kernel parameters: maxdsiz, max_thread_proc, and nkthread are properly set. For more information, refer to HP documentation.

Software requirements

Information about software requirements for Tivoli Monitoring is included in the Installation chapter of the IBM Tivoli Monitoring: User’s Guide. For a complete and updated list of software requirements, see the Readme file provided with the CD.

HP-UX 11.23 IA64 requirements

To run Tivoli Monitoring on HP-UX 11.23 IA64 Itanium endpoints, the following prerequisites are required:
- The operating system must be configured to allow 32-byte applications to run (toleration mode)
- The JRE version must be 1.3.1.16, or higher.

Refer to the following HP Support site to know the minimum operating system patch prerequisites:

After installing the JRE, run the DMLinkJRE task to link and specify the fully qualified path to the JRE location installed.
**AIX requirements**

If you are running IBM Tivoli Monitoring 5.1.2 Fix Pack 11, on a managed node with AIX 5.3 ML 0.5 (oslevel -s result is: 5300-05), you can have problems with middle layer processes. To solve these problems, apply AIX fix for APAR IY89597.

**Linux requirements**

Starting from Tivoli Monitoring 5.1.2 Fix Pack 7, managed nodes and TMR servers running on Linux platforms require Tivoli Management Framework 4.1 or higher.

If you have installed Tivoli Management Framework 4.1.1 Fix Pack 4, the installation of IBM Tivoli Monitoring 5.1.2 Fix Pack 8 might fail. This problem is addressed by Tivoli Management Framework APAR IY77095.

**Tivoli Management Framework requirements**

Tivoli Monitoring 5.1.2 Fix Pack 16 now supports Tivoli Management Framework 4.3.1.

Tivoli Management Framework patch 4.1.1-LCF-0034 or higher is required starting from Tivoli Monitoring 5.1.2 Fix Pack 11. The above patch can be installed on all Framework versions (3.7.1, 4.1, and 4.1.1)

**Tivoli Data Warehouse requirements**

On Tivoli Data Warehouse 1.3, before installing AMX and AMY components of any Tivoli Monitoring fix pack, ensure you have already installed 1.3.0-TIV-TDW-LA0002 patch.

**Supported Databases**

Starting from Tivoli Monitoring 5.1.2 Fix Pack 10, DB2 UDB ESE 9.1 is supported as Server ITM_DB.

**JRE levels**

On the endpoints requiring Java Runtime Environment (JRE), JRE 1.3.1 or 1.4.2 must be installed. In general, you are recommended to install and use the JRE available on the Tivoli Monitoring product CD. For details, refer to the *IBM Tivoli Monitoring: User’s Guide*, chapter "Installing", section "Installing Java Runtime Environment".

In addition, the following JREs are supported on all the Tivoli Monitoring engine platforms, if they are supported by the operating system:

- JRE 1.3.1 SR8 or higher
- JRE 1.4.2 SR6
- JRE 1.5 SR3

Starting from Tivoli Monitoring Fix Pack 14, JRE 1.4.2 SR9 is provided with the product CD for all the platforms, except for AS/400 that still does not support it.

Starting from Tivoli Monitoring Fix Pack 15, JRE 1.3.1, is no longer provided with the product CD.

**Note:** Before upgrading the JRE version, make sure that the operating system and the monitoring applications you are running support it.
The above are the JRE levels supported by Tivoli Monitoring. For more information about the JRE to be used with monitoring applications, refer to the corresponding documentation, and for more information about the JRE level supported by the operating system, refer to specific JRE documentation.

**Supported Configurations**

Starting from Tivoli Monitoring 5.1.2 Fix Pack 12, Data Execution Prevention (DEP) enabled hardware on Windows is supported on Tivoli Monitoring 5.1.2 base product. For information about this support with resource models that have not been provided with the base product, refer to the corresponding documentation.

**Upgrade notes**

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 as 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 Readme file.

Starting from Tivoli Monitoring 5.1.2 Fix Pack 3, AMX and AMY components can be installed on every product level, because Fix Pack 3 and later supersede the AMX and AMY components of all the previous Fix Packs.

To install Tivoli Monitoring 5.1.2 Fix Pack 3 and later, use the UNIX tar command to extract the fix packs. Do not use Winzip to extract them, because this would cause an unsuccessful installation.

Before installing Tivoli Monitoring 5.1.2 Fix Pack 14, it is recommended that you change the TaskLibrary Validation Policy from RestrictedTaskLibrary to NONE in order to avoid possible installation failures.

**AMX and AMY post-installation steps**

If you have a Remote Warehouse Agent in your configuration, you must complete the installation of AMX and AMY fix packs manually on the remote agent site performing the following steps:

1. To run AMX on the remote Agent Site, copy the %TWH_TOP_DIR%/apps/amx tree under the following directory:
   - $TWH_TOP_DIR/apps/amx (on UNIX)
   - %TWH_TOP_DIR%/apps/amx (on Windows)

2. To run AMY on the remote Agent Site, copy the %TWH_TOP_DIR%/apps/amy tree under the following directory:
   - $TWH_TOP_DIR/apps/amy (on UNIX)
   - %TWH_TOP_DIR%/apps/amy (on Windows)

3. Copy from the following directory:
   - $TWH_TOP_DIR/apps/amx/Version/misc (on UNIX)
   - %TWH_TOP_DIR%/apps/amx/Version/misc (on Windows)
   to:
   - $TWH_TOP_DIR/tools/bin (on UNIX)
the following files:

- `amx_extract.sh`
- `amx_rim_prune.sh` (required only if you are installing a fix pack later than Tivoli Monitoring 5.1.2)
- `amx_rim_extract.sh` (required only if you are installing a fix pack later than Tivoli Monitoring 5.1.2)
- `amx_pre_extract.sh`
- `amx_create_comp_msmt.sh`
- `amx_rollup.sh`

4. Set the file permissions to 755.
5. On UNIX systems check that the file mode is set to `UNIX`.
6. Configure the **Processing Option** of AMX and AMY to use the Remote Agent Site, doing as follows:
   a. Start the Tivoli Data Warehouse GUI.
   b. Expand **Administration** --&gt; **Agent Sites** directories.
   c. Right-click on your remote agent site, and select **Properties**.
   d. In the **Warehouse Sources and Targets** tab, move AMX and AMY sources and targets from the **Available warehouse sources and targets** to **Selected warehouse sources and targets**, and click OK.
   e. Repeat steps 6c and 6d for each Agent Site to configure.
7. Configure the AMX and AMY **Processes** to use the Remote Agent Site doing as follows:
   a. From the Tivoli Data Warehouse GUI, browse to: **Subject Area** --&gt;**AMX_IBM_TIVOLI_Monitoring_v5.1.1_Subject_Areas** --&gt; **Processes** --&gt; **AMX_c05_ETL1_Process**.
   b. Demote all the AMX steps into **Development** mode.
   c. Right-click on **AMX_c05_s005_Pre_Extract** step and select **Properties**.
   d. Click on the **Processing Options** tab.
   e. In the **Agent Site** field, select your agent site and click OK.
   f. Repeat 7c and 7d steps for all AMX processes.
   g. Restore all steps into your usual working mode.
   h. Repeat step 7f for each AMY process contained in
      **AMY_Operating_Systems_PAC_v1.1.0_Subject_Area** --&gt; **AMY_m05_ETL2_Process** list.
8. Configure the Warehouse Sources of AMX and AMY to use the Remote Agent Site, doing as follows:
   a. Start the Tivoli Data Warehouse GUI.
   b. Expand **Warehouse Sources**.
   c. Right-click on **AMX_ITM_RIM_Source**, and select **Properties**.
   d. In the **Data Source** tab, fill in the **Data Source Name** and **System Name** fields with the information related to the remote Agent Site, and click **OK**.
   e. Repeat steps 8c and 8d for each Warehouse Source to configure.
9. Configure the Warehouse Targets of AMX and AMY to use the Remote Agent Site, following the procedure described in step 8 applying it to the **Warehouse Targets**, instead of the Sources.
Chapter 3. Software limitations and problems

This section describes the software limitations, problems and workarounds, and list of fixed defects for this product release.

Software limitations

This section lists the limitations for this release of Tivoli Monitoring, organized by areas.

Tivoli Monitoring engine and general functions

- The state of profiles/resource models of an endpoint (Windows and UNIX) is not preserved after the engine is restarted.
- When you upgrade your endpoints to Tivoli Monitoring 5.1.2, all old historical data that resides on the endpoints is deleted. This means that, at most, you could lose all data collected in the last 24 hours by the old Tivoli Monitoring 5.1.1 engine, 24 hours being the maximum amount of time that data can be stored on the endpoints.
- To display graphics on the Web Health Console on Windows platforms, the number of display colors must be set to 256 or higher.
- When you uninstall Windows endpoints running resource models that use java-based providers, DMEndpointUninstall task can exceed its timeout value (the default timeout is 60 seconds). Under these circumstances it is highly recommended to change the default timeout value for DMEndpointUninstall from 60 seconds to 300 seconds on the Execute Task dialog that can be opened using the Tivoli Desktop. Similarly, and under the same circumstances, if you invoke the task DMEndpointRemoveFiles using the Tivoli Desktop, it is recommended to change the default timeout value from 60 seconds to 300 seconds.
- If endpoints are renamed after data collection requests were submitted, the requests need to be canceled and submitted again using the new endpoint names.
- OS/400 endpoints running on systems with less than 512MB assigned to the Java Virtual Machine’s storage pool might result in slow startup times for the endpoint engine.
- The cr_itm_db.sh script, which is used to create tables for the ITM_DB database, does not work when DB2 has been configured to work with multiple nodes.
- Before starting the Tivoli Monitoring engine, Solaris 9 endpoints require the installation of library libucb.so.1.
- On Solaris platforms, to enable the TCP scan feature of the Heartbeat, ensure that the Network Cache Server daemon is running.
- You can launch wdmcollect -l -m and wdmcollect -r -m commands from a TMR to upload data from a remote managed node. However, if the TMR runs on UNIX or Linux and the managed node runs on Windows (or vice versa), the commands do not work. A message says that the operation is successful, but no data is uploaded.
- Multiple Tivoli Monitoring engines running on the same physical Windows server, must be updated at the same fix pack level. Configurations where there is any discrepancy in the engine code level are not supported.
- On Solaris x64/x86-based systems, DMXSpp resource model is not supported. DMXNetworkInterface resource model is not supported in Solaris10 local zones.
- Starting from Tivoli Monitoring Fix Pack 9 you can start an engine from a local endpoint. However, this must be done from a Korn shell, because if you do it from a Bourne shell, the engine stops when you quit the session.

**Tivoli Enterprise Console Server**
- While checking events on the Tivoli Enterprise Console (for example by using the `wtdumprl` command) you might notice that the parsing of events failed with an indication "PARSING_FAILED~'Line 1: Value does not match type INTEGER!'".
  
  When this is the case, you should re-build, export, and install all BAROC files into the TEC rule base using the Tivoli Monitoring Resource Model Builder.
- In order to run DMCrateRuleAndLoadBaroc task, IBM Tivoli Monitoring must be installed on the Tivoli Enterprise Console server.
- In an environment with AIX 4.3.3 managed nodes, when heartbeat events are configured to be sent to a Tivoli-unsecure Enterprise Console server, the `heartbeat.unsecure_tec_server` configuration key of the `wdmconfig` command must be set to the IP address of the server (not to the fully qualified name).
- APAR IY77309: The Parametric Event Log resource model sends to the Tivoli Enterprise Console server an event whose size is over 8 KB. Because this is too large for the Tivoli Enterprise Console server, it truncates it.

**ETLs**
- When using AMX to load data into the Tivoli Data Warehouse, while you are in the process of defining log keys, do not use the character ";" (semicolon) nor the character "=" (equal) as a separator because the usage of these characters might create problems. You can use the character "," (comma) instead.
- If during the installation of Tivoli Monitoring 5.1.1 Fix Pack 6 on Tivoli Data Warehouse, the installation of AMY component fails, use the following workaround and reinstall AMY component.

```sql
Delete from twg.msmt where msmttyp_id in (select msmttyp_id
from twg.msmttyp where MsmtTyp_Nm = 'ServerErroredOut');

delete from twg.MGRPMBR where msmttyp_id in (select msmttyp_id
from twg.msmttyp where MsmtTyp_Nm = 'ServerErroredOut');

delete from twg.MSMTRUL where msmttyp_id in (select msmttyp_id
from twg.msmttyp where MsmtTyp_Nm = 'ServerErroredOut');

delete from twg.msmttyp where MsmtTyp_Nm = 'ServerErroredOut';
```
- Starting from the installation of Tivoli Monitoring 5.1.1 Fix Pack 6 on Tivoli Monitoring Warehouse Enablement Pack, there is a mismatch between the levels of AMX and AMY. The following table summarizes AMX and AMY levels for each fix pack:

<table>
<thead>
<tr>
<th>Tivoli Monitoring level</th>
<th>AMX level</th>
<th>AMY level</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1.1 FP 6 (5.1.2)</td>
<td>5.1.1.600</td>
<td>1.1.0.600</td>
</tr>
<tr>
<td>5.1.2 FP 1</td>
<td>5.1.1.660</td>
<td>1.1.0.660</td>
</tr>
<tr>
<td>5.1.2 FP 2</td>
<td>5.1.1.670</td>
<td>1.1.0.670</td>
</tr>
<tr>
<td>5.1.2 FP 3</td>
<td>5.1.1.680</td>
<td>1.1.0.680</td>
</tr>
</tbody>
</table>
Table 1. ETL levels (continued)

<table>
<thead>
<tr>
<th>Tivoli Monitoring level</th>
<th>AMX level</th>
<th>AMY level</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1.2 FP 4</td>
<td>5.1.1.690</td>
<td>1.1.0.690</td>
</tr>
<tr>
<td>5.1.2 FP 5</td>
<td>5.1.1.700</td>
<td>1.1.0.700</td>
</tr>
<tr>
<td>5.1.2 FP 6</td>
<td>5.1.1.710</td>
<td>1.1.0.710</td>
</tr>
<tr>
<td>5.1.2 FP 7</td>
<td>5.1.1.810</td>
<td>1.1.0.810</td>
</tr>
<tr>
<td>5.1.2 FP 8</td>
<td>5.1.1.820</td>
<td>1.1.0.820</td>
</tr>
<tr>
<td>5.1.2 FP 9</td>
<td>5.1.1.900</td>
<td>1.1.0.900</td>
</tr>
<tr>
<td>5.1.2 FP 10</td>
<td>5.1.1.910</td>
<td>1.1.0.910</td>
</tr>
<tr>
<td>5.1.2 FP 11</td>
<td>5.1.1.920</td>
<td>1.1.0.920</td>
</tr>
<tr>
<td>5.1.2 FP 12</td>
<td>5.1.1.925</td>
<td>1.1.0.925</td>
</tr>
<tr>
<td>5.1.2 FP 13</td>
<td>5.1.1.930</td>
<td>1.1.0.930</td>
</tr>
<tr>
<td>5.1.2 FP 14</td>
<td>5.1.1.935</td>
<td>1.1.0.935</td>
</tr>
<tr>
<td>5.1.2 FP 15</td>
<td>5.1.1.940</td>
<td>1.1.0.940</td>
</tr>
<tr>
<td>5.1.2 FP 16</td>
<td>5.1.1.945</td>
<td>1.1.1.945</td>
</tr>
</tbody>
</table>

- If an endpoint, which logs data for the Tivoli Data Warehouse, is renamed, the data logged to the RIM database before the renaming maintains the old endpoint label.

- Occasionally, when you run AMY, the last rollup step fails with the following error message:

  ```
  AMY_m05_s040_Mart_Rollup*
  CDW8101E Select from rpi.ss_joincols failed with the following message:
  Database Connection Information
  Database server     = DB2/NT 7.2.8
  SQL authorization ID = DB2ADMIN
  Local database alias = TWH_MD
  RC = 7356   RC2 = 8410
  ```

  The error is originated by DB2 APAR JR19393, which causes the IWH.Attributelink table of TWH_MD to be emptied. Therefore, the view rpi.ss_joincols cannot be populated and, as a result, the command fails. If this happens, use the following workaround:

  1. Connect to TWH_MD
  2. Verify that the table is actually empty, launching the following command:

     ```
     select * from IWH.Attributelink
     ```

     3. If you are confirmed that there are no records in the IWH.Attributelink table, you have to populate it manually.

     The following is an example of how to populate it:

     ```
     DB2 insert into iwh.attributelink
     (iwhid,updated_by,update_ts,name,short_description,type,operatorexp,
     useandsep,orderno,operexp2) (select source_iwhid,'db2admin',
     '2003-11-29-19.13.48.000000', '', ',JOIN', 1, 0, '='
     from iwh.relationship where relation_name='StarSchema_TO_AttLnk'
     and source_iwhid not in (select iwhid from iwh.attributelink))
     ```

- When trying to run the second AMX step with Oracle database, you can receive the following error message:

  ```
  DWC07356E An agent's processing of a command of type "runUDP" failed for edition "0" of step "AMX_c05_s010_Rim_Extract".
  CDWEX0807E A general SQL error occurred:
  ```
SQL_ERROR: 'Fetch Src Data' 2005.06.16 10:32:35.578
sqlState = HY004,
nativeErr = 0,
errorMsg = [Oracle][ODBC]Invalid SQL data type &LT;-25&GT;.

If it happens, you can use the following workaround: do not use the ODBC driver shipped with Oracle, instead use the driver provided by Tivoli Data Warehouse for Oracle.

**Resource Models**

- The provider of the DMXGroup and DMXUser CIM classes has been changed to return the group ID and user ID as strings rather than as numeric values. As a consequence, any custom resource models that use this data must now use the new DMXGroup_02 and DMXUser_02 classes. The corresponding MOF and provider implementation can be found in the DMXSecurity resource model workspace.

- The provider of the DMXFileSystem CIM class has been changed to return the numeric values as doubles rather than integers. As a consequence, any custom resource models that use this data must now use the new DMXFileSystem_11 class. The corresponding MOF and provider implementation can be found in the DMXFileSystem resource model workspace.

- On the Web Health Console with WebSphere Application server 4, when you are running a resource model that has an ampersand character (&) in its parameters, the resource model results can be displayed incorrectly on the Web Health Console. This can happen, for example, if you are running the DMXPROCESS resource model, whose properties are separated by ampersands. The Web Health Console does not display the resource model indication instances. If the resource model triggers an event, the Web Health Console correctly displays the resource model health as zero (0), but if you search for the failing instance, the console shows that no instance is present. This is because the instance is associated to a property specified using the ampersand, which is not correctly recognized.

- On the Windows TMW_Process resource model, square parentheses ([ ]) are not supported characters in the basename parameter.

- If you have installed Tivoli Monitoring 5.1.2 Fix Pack 6, before running TMW_Process resource model, modify your old profiles removing the following default value: BASENAME=CMD&MINCOUNT=3 from the Parameters section.

- On HP-UX systems, if the Gold Application Patches are not applied, the DMXNetworkInterface resource model uses all available CPU.

- DMXCpu resource model, even when used in a shared processor partition, does not provide information about the Entitled Capacity and the percentage of Entitled Capacity used.

- When running TMW_Process resource model, if the same basename process is used on multiple lines, and both maxcount and mincount are specified, Maximum Process Instance event is not created. Whereas, if the same basename process is used on the same line with both maxcount and mincount specified, the event is created correctly.

- TMW_ParamPorts resource model does not run on Windows Vista platforms.

- The input parameters for the DMXFilesystem resource model are *either* the file systems you want to monitor, *or* the file systems you do not want to monitor. You cannot use wildcards in both the settings. To specify that you want some file systems monitored, and others not, use the wildcard (*) in the first input field:

  Included = /*
to specify that you want all the file systems monitored, and then use the file systems absolute paths:

```
excluded =/file_absolute_path/
```

to detail exactly which file systems are to be excluded from the monitoring.

- **DMXPROCESS**
  
  `mincount=value`

parameter does not work as expected. The following description provided in the Tivoli GUI is incorrect: "Minimum count value for ALL instances of processes that match the basename=value"

---

**Software problems and workarounds**

This section describes known problems in this release of Tivoli Monitoring. Where applicable and known, suggested workarounds are identified. Note that this might not be a complete list of defects.

Current defects and workarounds for this release of Tivoli Monitoring include:

1. **CMVC 1172 and 1411 - Web Health Console**: When attempting to view graphs in a browser, you might experience an indefinite wait on AIX or a server exception on other UNIX platforms on Tivoli Monitoring Web Health Console Servers that do not have an X server. The Tivoli Monitoring Web Health Console Server requires an X server to be able to generate graphs.

   Workaround 1: On servers without an X server or display hardware, an X virtual frame buffer (Xvfb) server can be used instead. The instructions below describe how to install and activate Xvfb on the supported UNIX platforms.

   - **RedHat Linux 2.1 on i386:**
     
     a. Insert the CD
     
        - For the Web Health Console based on WebSphere Application Server 4, insert the Web Health Console CD.
     
        - For the Web Health Console based on WebSphere Application Server 5.x, insert the Tivoli Monitoring 5.1.1 CD
     
     b. Mount CD, launching:
        
        `mount -t iso9660 /dev/cdrom /mnt/cdrom`
     
     c. Install Xvfb, launching:
        
        `rpm -i /mnt/cdrom/support/Xvfb/RedHat72/ XFree86-Xvfb-4.1.0-3.i386.rpm`
     
     d. Edit the file
        
        - For the Web Health Console based on WebSphere Application Server 4, edit the file: `/etc/rc.d/init.d/itmwasd` as follows:
          
          Change: `export XVFB_HOME=`
          
          to: `export XVFB_HOME=/usr/X11R6/bin`
        
        - For the Web Health Console based on WebSphere Application Server 5.x, copy the cdom/ITMWHC/itmwasd file from the Tivoli Monitoring 5.1.2 Fix Pack 5 CD into a local directory and edit it as follows:
          
          Change: `export XVFB_HOME=` to: `export XVFB_HOME=/usr/X11R6/bin`.
          
          change: `REPLACE_WAS_DIR` to `export WAS_DIR=WAS_DIR`, and change: `REPLACE_HTTP_DIR` to `export HTTP_DIR=HTTP_DIR`
     
     e. Restart the server:
- For the Web Health Console based on WebSphere Application Server 4, launch the following commands:
  
  
  /etc/rc.d/init.d/itmwasd stop
  /etc/rc.d/init.d/itmwasd start
  
- For the Web Health Console based on WebSphere Application Server 5.x, launch the following commands:
  
  completelpath itmwasd stop servername
  completelpath itmwasd start servername

  - For the Web Health Console based on WebSphere Application Server 4, edit the following file:
    
    /etc/rc.d/init.d/itmwasd
    
    Change:
    
    export XVFB_HOME
    
    to:
    
    export XVFB_HOME=/usr/bin/
    
    X11
    
  - For the Web Health Console based on WebSphere Application Server 5.x, copy the cdrom/ITMWHC/itmwasd file from the Tivoli Monitoring 5.1.2 Fix Pack 5 CD a local directory and edit it as follows:
    
    Change:
    
    export XVFB_HOME=
    
    to:
    
    export XVFB_HOME=/usr/bin/X11,
    
    change:
    
    REPLACE_WAS_DIR
    
    to:
    
    export WAS_DIR=WAS_DIR,
    
    change:
    
    REPLACE_HTTP_DIR
    
    to:
    
    export HTTP_DIR=HTTP_DIR

e. Restart the server:

  - For the Web Health Console based on WebSphere Application Server 4, launch the following commands:
    
    /etc/rc.d/init.d/itmwasd stop
    /etc/rc.d/init.d/itmwasd start

  - For the Web Health Console based on WebSphere Application Server 5.x, launch the following commands:
    
    completelpath itmwasd stop servername
    completelpath itmwasd start servername

  - For the Web Health Console based on WebSphere Application Server 4, insert the Web Health Console CD.
  
  - For the Web Health Console based on WebSphere Application Server 5.x, insert the Tivoli Monitoring 5.1.1 CD

  b. Mount CD, launching:

     mount -F hsfs -o ro /dev/dsk/c0textension /mnt/cdrom

  c. Change directory launching:

     cd /usr

  d. Install Xvfb launching:

     gunzip -c /mnt/cdrom/support/Xvfb/Solaris/solarisxvfb.tar.gz
     tar xf

  e. Edit the file
For the Web Health Console based on WebSphere Application Server 4, edit the following file: `/etc/init.d/itmwasd` as follows:
Change: `export XVFB_HOME` to: `export XVFB_HOME=/usr/XVFB`

For the Web Health Console based on WebSphere Application Server 5.x, copy the `cdrom/ITMWHC/itmwasd` file from the Tivoli Monitoring 5.1.2 Fix Pack 5 CD a local directory and edit it as follows:
Change: `export XVFB_HOME` to: `export XVFB_HOME=/usr/XVFB`, change: `REPLACE_WAS_DIR`, and change: `REPLACE_HTTP_DIR` to `export HTTP_DIR=HTTP_DIR`

f. Restart the server:
For the Web Health Console based on WebSphere Application Server 4 launch the following commands:
`/etc/rc.d/init.d/itmwasd stop`
`/etc/rc.d/init.d/itmwasd start`

For the Web Health Console based on WebSphere Application Server 5.x, launch the following commands:
`completepath_itmwasd stop servername`
`completepath_itmwasd start servername`

• HP-UX 11.11:
  a. Insert CD
    - For the Web Health Console based on WebSphere Application Server 4, insert the Web Health Console CD.
    - For the Web Health Console based on WebSphere Application Server 5.x, insert the Tivoli Monitoring 5.1.1 CD
  b. Mount CD, launching:
    `mount -F cdfs -r /dev/dsk/c0textension /mnt/cdrom`
    **Note:** Depending on which patches are installed, you might need to use `pfs_mount` rather than `mount`. See the `pfs_mount` man page for more information.
  c. Change directory, launching:
    `cd /usr`
  d. Install Xvfb, launching:
    `gunzip -c /mnt/cdrom/support/Xvfb/HPUX11/hp11xvfb.tar.gz`
    `tar xf -g`
  e. Edit the file
    - For the Web Health Console based on WebSphere Application Server 4, edit the following file: `/sbin/init.d/itmwasd` as follows:
      Change: `export XVFB_HOME` to: `export XVFB_HOME=/usr/XVFB`
    - For the Web Health Console based on WebSphere Application Server 5.x, copy the `cdrom/ITMWHC/itmwasd` file from the Tivoli Monitoring 5.1.2 Fix Pack 5 CD a local directory and edit it as follows:
      Change: `export XVFB_HOME` to: `export XVFB_HOME=/usr/XVFB`, change: `REPLACE_WAS_DIR`
to export WAS_DIR=WAS_DIR, and change. REPLACE_HTTP_DIR
to export HTTP_DIR=HTTP_DIR

f. Restart the server:
   - For the Web Health Console based on WebSphere Application Server
     4, launch the following commands
     `/sbin/init.d/itmwasd stop
     `/sbin/init.d/itmwasd start
   - For the Web Health Console based on WebSphere Application Server
     5.x, launch the following commands:
     `completepath_itmwasd stop servername`
     `completepath_itmwasd start servername`

Notes:
a. CD mounting assumes that /mnt/cdrom is the default mount point. You
   might need to change this to your local mount point or create the directory
   before mounting the CD.
b. You might have to use a different path for XVFB_HOME if you installed at
   a different location.

Workaround 2: Set the DISPLAY variable to point to an X server on another
machine prior to starting the Web Health Console.
a. On the X server, add the Web Health Console server to the access control
   list of machine with a running X-windows graphical environment:
   `[root@myhost]# xhost +itmhcserver`
b. On the console machine, stop the WebSphere Application Server by
   running the `stopServer.sh` command:
   `[root@itmconsole]# WAS_DIR/bin/stopServer.sh`
c. On the console machine, set the DISPLAY variable to the X server:
   `DISPLAY=myhost; export DISPLAY`
d. Start the WebSphere Application Server by running the `startServer.sh`
   command:
   `[root@itmconsole]# WAS_DIR/bin/startServer.sh`

2. CMVC 49542 - Endpoints that were renamed might not be visible on the Web
   Health Console.
   Workaround: Do the following:
a. Remove the endpoint from the endpoint list of the Web Health Console
      before renaming it.
b. Stop the endpoint engine.
c. Run the command:
   `wep old_endpoint_label -s new_endpoint_label`
d. Restart the endpoint engine.
e. Add the renamed endpoint to the endpoint list of the Web Health Console.

3. CMVC 49710 - The `cr_itm_db.sh` script, which is used to automate the creation
   of the Tivoli Monitoring database, might fail on some Oracle 9.2 environments
   with the following Oracle error:
   `ORA-12536: TNS:operation would block`
   Workaround: This is a random problem that seems to be related to the Oracle
   network library SQL Net 2. Usually, the re-execution of the `cr_itm_db.sh` is
   sufficient to resolve the problem but, if you are still unable to successfully
   complete the automated procedure, you can manually create the Tivoli
   Monitoring database, by executing the SQL script (cr_db.ora.sql) that is
created into the WarehouseCfg directory by the cr_itm_db.sh script, using your favorite Oracle client (Sqlplus or SQL Worksheet) and after logging in as SYS user with SYSDBA authority.

4. APAR IY60275 - The fresh install of AMX on a distributed system fails with error code CDWIC5066E. The twh_ibm_db2_runlog.log shows an SQL0647N error. There are two possible workarounds to this problem.
   Workaround 1:
   a. Stop the Warehouse Server and Warehouse Logger services from: Start --> Settings --> Control Panel --> Administrative Tools --> Services
   b. Stop the database instance on the system hosting the TWH_CDW, by issuing a db2stop force.
   c. Restart the database instance, by issuing a db2start.
   d. Restart the Warehouse Server and Warehouse Logger.
   e. Retry the installation.
   Workaround 2:
   a. Prior to the installation, create a buf8k buffer pool executing the instructions in the amx_script.sh (connect to TWH_CDW user userid using passwd, create bufferpool buf8k size 500 pagesize 8 K).
   b. Stop and start the database instance under which the TWH_CDW is defined (do so on the system hosting the TWH_CDW).
   c. Proceed with the installation.

5. CMVC 51623 - Occasionally, on endpoints running Windows platforms the endpoint database might become corrupted.
   Workaround:
   a. Stop the Tivoli Monitoring engine.
   b. Delete the files Tmw2kdb.mdb and Tmw2kdp.ldb located in the LCF_DATDIR/LCFNEW/Tmw2k/db directory.
   c. Restart the Tivoli Monitoring engine.

6. APAR IY59210 (for TMF 3.7 and 4.1) and APAR IY45239 (for TMF 4.1.1) - When the jre.tar.gz files have read-only attributes, wdmdistrib is not able to distribute JRE to the endpoint and the wmdist -I output shows that the distribution was interrupted. To bypass the problem, change the file attributes to include write permissions.

7. If you attempt to remove a profile manager with the command: wdel@ProfileManager, you might encounter the following error:

   FRW:CCMS:0245 Cannot remove profile manager 'act.dm.unix.daemon' because it still has original profiles

   If you do, resolve the problem as follows:
   a. Remove all subscribers to each profile still existing under the Profile Manager
   b. Delete all profiles from the problematic profile manager.
   c. Run wchkdb to clean out the database
   d. Run the following command to clean out the databases attribute of the profile manager:

   ```
   idlattr -ts <PM_OID> databases
   imp_TMF_CCMS::ProfileOrganizer::database_list
   '[]' -I
   ```

   Note: PM_OID is the object id of the profile manager being removed.
e. Delete the profile manager again with the following command:

   wdel @ProfileManager:profilemgrname

8. CMVC 53151 - If you change the Tivoli Monitoring database used as a RIM source from which you extract populated data, the Tivoli Data Warehouse might be unable to find it and retrieve data. If it happens, use the following workaround:

   insert into iwh.attributelink (iwhid,updated_by,update_ts,name,short_description,type,operator_exp,use_and_sep,order_no,operator_exp2)
   (select source_iwhid,'db2admin','2003-11-20-19.13.48.000000',''','JOIN','1',0,'=' from iwh.relationship where
    relation_name='StarSchema_TO_AttLnk' and source_iwhid not in
    (select iwhid from iwh.attributelink))

9. Occasionally, you might find some duplicate entries in the Endpoint table of ITM_DB database. You might find the same endpoint associated to two hostnames, one of which is wrong. To resolve this problem the following scripts are released with Fix Pack 3 and later:

   • rm_ep_label.db2
   • rm_ep_label.mssql
   • rm_ep_label.syb
   • rm_ep_label.inf
   • rm_ep_label.ora

   You can find the scripts under the following path: $BINDIR/TME/Tmw2k/
   WarehouseCfg/cleanup_db. To launch these scripts, use
   the itm_run_db_query.sh shell script. You can use this shell to run also any
   sql script file you have customized for your own queries on the ITM_DB database. You can find the shell script under the following path:
   $BINDIR/TME/Tmw2k/WarehouseCfg/cleanup_db.

   The usage of this script is:

   Usage: sh itm_db_run_query.sh query_file
   NOTE: for DB2 the query_file must use $ as statement terminator character.

   Where query_file is the sql script rm_ep_label without the rdbms extension or
   can be any of your filenames, without the rdbms extension.

   The following is an example with ITM_DB on oracle rdbms:
   itm_db_run_query.sh rm_ep_label

   Looking up TME 10 system information...

   This script runs rm_ep_label script on ITM DB

   THIS OPERATION COULD TAKE SEVERAL MINUTES

   Do you want to continue? [yes/no]

   y

   The RIM object within this TMR is configured to define the following database:

   Database Vendor: Oracle
   Database Server ID: itmfp
   Database Home: /space/oracle/OraHome1
   Database ID: itm_db
   Database User ID: oracle

   If this is correct, enter 'Y' to continue. Otherwise, check the configuration of the local RIM object or run this script from the RIM host within the TMR you wish to configure.
10. CMVC - 53235 If you have installed IBM Tivoli Monitoring for Web Infrastructure WebSphere® Application Server 5.1.2 on Framework 4.1.1, some Tivoli Monitoring tasks launched on OS/400 systems can fail. The reason for this is that IBM Tivoli Monitoring for Web Infrastructure WebSphere Application Server changes the OS/400 dependencies. Therefore, Tivoli Monitoring tasks on OS/400, requiring GAWK, fail because the right path to GAWK cannot be found. This problem has been notified (defect 139936) and partially fixed with 5.1.2-WAS-0002. To run failed tasks, use the following workaround:
   a. Install the 5.1.2-WAS-0002 interim fix (contact the support)
   b. Run $BINDIR/../generic_unix/TME/WSAPPSVR/itmwas_tool_dep.sh

11. CMVC 53490: Occasionally, when running the AMX_c05_s030_Exception step, the command fails and the following error occurs:
   CDWEX8087E A general SQL error occurred: SQL_ERROR: 
   sqlState = 57011, nativeErr = -964, 
   errorMsg = [IBM][CLI Driver][DB2/6000] 
   SQL0964C The transaction log for the database is full. 
   SQLSTATE=57011

   This might happen because the transaction log is full. Therefore, if you receive this error, increase the transaction log size (to 500 MB) and run the AMX_c05_s030_Exception step again. Alternatively, you can increase the number of log files. Refer to DB2 documentation for guidelines on sizing log files.

12. APAR IY69317: On Solaris, a process running without parameters returns a maximum of 80 characters, including the fully qualified path name for the process. This can limit the name filters. This limitation does not apply to processes that contain arguments. On Solaris, the basename parameter is the first 15 characters of the process name. The ps -e command only displays the first 8 characters so you should determine any applicable remaining characters from the ps -ef command.

13. Occasionally, when you launch wdmcmddistrib -upgrade on a running endpoint, some resource models fail. The reason for the error is a possible mismatch in the ILT classes. If this occurs, restart the engine on the failing endpoint and the mismatch is solved.

14. If you remove any of the engine update files while the engine of the endpoint is running, the Heartbeat and the Web Health Console show that the endpoint is connected but it is not possible to contact the engine. This is due to the dependency check update implemented with Tivoli Monitoring 5.1.2 Fix Pack 4. If this occurs, restart the Tivoli Monitoring engine to recover from this situation.

15. Starting from Tivoli Monitoring 5.1.2 Fix Pack 2, if you experience problems with logging or trace enablement of any of the monitoring applications you use with Tivoli Monitoring, refer to Component Services IF0005 patch.

16. When you create a new Tivoli Monitoring database using the cr_itm_db.sh utility script, the new database schema is not updated. To update it, launch the up_itm_db.sh update script. For more details about updating the Tivoli Monitoring database, refer to the Readme file supplied with the product.

17. When you run the DMXPhysicalDisk resource model on AIX 5.1 systems that have two or more disk drives, you might experience a memory leak. To solve this problem, apply the AIX 5.1 maintenance level 7 fixes.

18. APAR IY76573: On Solaris platforms, when the JRE is distributed with the wmdistrib command, the user ID and group ID of the JRE/DMAE directory are set respectively to 1114 and 150, instead of being set to root. To solve this
problem, log on to system and set the user ID and group ID to the IDs for the root account using the chown command.

19. RHEL 3 and RHEL 4 on Linux for z/OS systems cannot run Tivoli Monitoring using the JRE provided with the product CD. They require JRE 1.3.1 SR 9 (or later) or JRE 1.4.2 SR 3 (or later).

20. On TMR and managed nodes running SLES 9 z/OS, when you launch Tivoli Monitoring commands, you could receive the following error message:
   symbol errno, version GLIBC_2.0 not defined in file libc.so.6
   with link time reference
   This happens because the glibc version is too old for your environment. In this case, export the LD_ASSUME_KERNEL environment variable and set it to glibc, version 2.4.1.

21. CMVC 56343: On managed nodes running Linux (Intel or z/OS), and having the heartbeat active and enabled to send events to the Tivoli Enterprise Console server, the following problem can occur: when the heartbeat stops, some of the HeartbeatOff events are sent to the Tivoli Enterprise Console server (as expected). Other HeartbeatOff events are either stored in the cache, or both, cached and sent to the Tivoli Enterprise Console server. This problem is caused by the Tivoli Enterprise Console server and has been addressed with defect number 184725.

22. On Tivoli Data Warehouse, if you cannot find the amy_set_extract_control.sh script in the TWH_INST_DIR/apps/amy/v110820/misc/tools directory, perform the following steps:
   a. From the installable images, browse to the following directory:
      5.1.2-ITM-FP08/tedw_apps_etl/amy/misc/tools.
   b. Copy the amy_set_extract_control.sh script and paste it in the following directory: TWH_INST_DIR/apps/amy/v110820/misc/tools.

23. APAR IY84280: Occasionally, on Windows 2003 Service Pack 1, and on Windows XP Service Pack 2 systems, the heartbeat process could fail when you launch wdmmngcache command. This behavior is due to a Tivoli Management Framework problem, already addressed by APAR IY74447. However, as a workaround, when the problem occurs, start the Internet Connection Service (ICS).

24. The waitTime metric for the DMXCpu resource model is not available on several Linux OS flavours. Therefore, the value -100% means that the metric does not exist.

25. APAR IY87355: Tivoli Monitoring engine experiences a core dump on HP-UX Itanium 64bit Java systems. As a workaround, set the java option -Xint for Tivoli Monitoring Java engine using the wdmepconfig -e endpoint_name -D java_GENERICOPTS="-Xint".

26. APAR IY87514: Occasionally, with Tivoli Monitoring Fix Pack 9 on Sun Solaris, launching wmdistrib -J JRE images-1r endpoint can corrupt the JRE directory deleting its content. The JRE directory can then be deleted manually only. Should the problem occur, run the DMRemoveLinkJRE task before launching the wmdistrib-J JRE images command.

27. Occasionally, DMXPhysicalDisk resource model on HP can cause a core dump when JRE 1.3.1 is used. In this case, JRE 1.4.2 is required in order to solve the problem.

28. CMVC 58149, and 58150: Occasionally, after you run odadmin reexec disp
command, the Heartbeat may not restart, or may cause a core dump. To avoid this problem, it is recommended that you run separately the two commands: odadmin shutdown disp and odadmin start disp.

29. Occasionally, it can occur that an endpoint is not registered into the heartbeat cache after a migration. If the destination heartbeat is stopped, an error occurs due to the misalignment between the endpoint cache and the internal endpoint label cache. If this happens, use the following workaround: keep all the heartbeats running on all the gateways, keeping all the endpoints disabled.

30. APAR IY93378: Occasionally, on AIX platforms, DMCompressDB task terminates abnormally with a core. To prevent this from happening, you can modify the script of the DMCompressDB task by inserting the following command before calling Java in the script:

```bash
export IBM_JAVA_MMAP_JAVA_HEAP=true
export LDR_CNTRL=MAXDATA=0x80000000@USERREGS
```

31. When you run wdmcmd command against a list of endpoints, if one or more of the listed endpoints does not exist, the command completes successfully on the existing endpoints, and fails on the non-existing ones. However, its return code is non zero.

32. On Windows endpoints, if you distribute JRE 1.5 SR3, using wmdistrib command, the wdmcheckprereq command does not find it, and incorrectly returns a FAIL status when checking the prerequisites. If this occurs, run DMLinkJRE task to solve the problem.

33. APAR IZ04648: Occasionally, on AIX operating systems, the tmnt_rimh_eng process can cause a core dump. This is because, by default, processes in AIX have a process memory size limitation of 256 MB. This process might try to allocate more memory, due to the big size of the XML files. If this happens, before starting the parent process, set and export the LDR_CNTRL environmental variable that controls the maximum number of data segments for a process. For example: export LDR_CNTRL=MAXDATA=0x20000000 and then restart middle layer process.

34. On UNIX platforms, when running a custom resource model that uses Java Script Split function with a JRE 1.4, the function can fail. If this happens, change the pattern according to the specifications provided with JRE 1.4.2.

35. On the AIX 6.1 platform, the DMXNetworkInterface resource model does not work unless you uncomment the following line:

```bash
VACM_VIEW defaultView internet - included -
```

in the snmpdv3.conf file, located in ./etc directory.

36. To install the IBM Tivoli Monitoring on a Linux-ix86 TMRs or Managed Nodes on Framework 4.3.1, export the LD_ASSUME_KERNEL environment variable and set it to version 2.4.1.

**Other considerations**

This section describes special considerations that apply to this release of Tivoli Monitoring:

- If you are planning to install the AMX and AMY Warehouse Enablement Packs in a Tivoli Data Warehouse 1.2 fresh environment (not upgraded from Tivoli Enterprise Data Warehouse 1.1), you should use the installation package provided with Tivoli Monitoring 5.1.2 (instead of the package provided with Tivoli Monitoring 5.1.1 plus Fix Pack 6).
Tivoli Monitoring 5.1.2 includes the same functions and features of Tivoli Monitoring 5.1.1 plus Fix Pack 6, but publishes only the latest sample reports into Crystal Enterprise ePortfolio.

If you do not use the Tivoli Monitoring 5.1.2 installation package, but use Tivoli Monitoring 5.1.1 plus Fix Pack 6, you will have four additional reports published in Crystal Enterprise ePortfolio (NET_STATS, UNIX_CPU_STATS, WIN_CPU_STATS, BUSIEST_SYS). These reports are no longer valid and are replaced by the new reports prefixed by the "Operating System:" label.

- Starting from Tivoli Monitoring 5.1.2 Fix Pack 4, when you run AMX Warehouse Enablement Pack for the first time, you should perform the following steps, to avoid a decrease in performance:
  
  From DB2 Command Window, after running amx_c05_s20_parsing and before running amx_c05_s30_exception step, launch the following commands:
  
  1. db2 connect to twh_cdw user <...> using <...>
  2. db2 runstats on table amx.stage_metricsdata with distribution and indexes all
  3. db2 runstats on table amx.stage_inst_no_ex with distribution and indexes all
  4. db2 runstats on table amx.met_no_ex with distribution and indexes all
  5. db2 runstats on table amx.stage_metrics with distribution and indexes all
  6. db2 terminate.

**Fixed APARs**

The following APARs have been corrected with Tivoli Monitoring 5.1.2 (the APARs corrected by subsequent product fix packs are listed in the corresponding Readme file).

The APAR number precedes the description of each problem.

APARs fixed for this release of Tivoli Monitoring include:

1. IY44585: Tivoli Monitoring creates *.jar and *.class files with UID=1114 and GID=105 in %LCF_DATDIR%/LCFNEW subdirectories.
2. IY49647: Setting a custom schedule within a IBM Tivoli Monitoring profile, it does not work on AS400. Even if the schedule correctly set, the profile never fires.
3. IY48646: AS/400® History Log RM not sending any indications.
4. IY48797: The Tivoli Monitoring e-mail response task does not work on Windows platform if more than one e-mail address has been specified in the task configuration.
5. IY48940: Any Tivoli Monitoring Automated Tasks (dm_mn_send_email or dm_mn_send_notice) fails in inter-connected TMR because of ambiguous reference to the TaskLibrary name.
6. IY49151: Resource Models scheduling does not work properly if a RM is restarted with the command wdmemg.
7. IY49366: wdmeditprf does not have any support for setting or editing the built-in actions of an Indication Event in an Tivoli Monitoring profile.
8. IY49515: The tmtt_gtw process of the middle layer does not write any traces into its trace file (trace_gtw_eng#.log).
9. IY49697: The distribution of a profile with a profile name longer than 124 characters fails on AIX endpoints.

10. IY49751: After Tivoli Monitoring engine sends its first tec event, it starts to spawn an upcall every 60 seconds.

11. IY49976: Environment variables created as part of an event are inconsistent between different Operating Systems.

12. IY50143: The `wdmcollect` is distributed also with the FP03 and FP04 base product component.

13. IY50205: Null environment variables passed in resource models created from DM Classic cause a NullPointerException.

14. IY50340: Unix Engine: logging logic is incorrect in case of profiles containing a RMs with logging enabled and a RMs with logging disabled. In this case the Engine consider all RMs belonging the profiles with the logging enabled.

15. IY50347: The `wdmngcache -m all -l` command should show the Endpoints and HB Status, grouped by the Managed Node they are assigned to, but instead it simply shows all the Endpoint and HB Status entries for all Managed Nodes at the end of the list of Managed Node names.

16. IY50377: The UNIX Network Interface resource model did not correctly ignore logical network interfaces. This caused confusing error messages in the log files. This problem occurs on Solaris.

17. IY50415: M12JavaProvider does not correctly handle the method invocation for the methods implemented in an ILT file.

18. IY50454: When the task DMEndpointRemoveFiles for windows platform performs the Windows Management Instrumentation stopping and there are other services depending on it, the service will not be stopped and some IBM Tivoli Monitoring files loaded by this service could be left on the endpoint.

19. IY50499: Custom Resource Models on Unix fails if they try to spawn a script using perl.

20. IY50811: For AIX, the UNIX CPU resource model calculates idle and system time in a way that is inconsistent with the other UNIX platforms.

21. IY50822: `wdmcmdistrib` fails to start if the Endpoint names in the Endpoints list are followed by blank spaces.

22. IY50826: `wdmcmdistrib` command line with -upgrade option does not work properly under certain conditions: It does not perform the operations of stopping engine, update binaries and start engine properly as expected.

23. IY50827: output of `wdmcmdistrib -upgrade` does not report correct status of the distribution.

24. IY50843: The command `wdmcmdistrib` with -m option does not work if launched on HUB TMR specifying the managed node sitting on the SPOKE TMR.

25. IY50953: The Windows engine stops working properly when a trace log level greater than 0 is set; under certain conditions the semaphore might remain locked blocking the calling threads. The engine itself might stop behaving correctly (the events are not sent anymore to TEC and XML files are not created).

26. IY50959: Windows engine can get DrWatson error from the Logger component calling the CLogger::NotifyModelWasStopped method.

27. IY51004: `wdmdistrib` does not update profile copies correctly when Resource Models have been deleted from master profile.
28. IY51006: In an interconnected TMR environment (two-way interconnect), the Web Health Console cannot manage endpoints in the remote TMR after Tivoli Monitoring 5.1.1 FP05 is installed.

29. IY51034: After Tivoli Monitoring 5.1.1 Fix Pack 5 the wdmseg output is not translated on a Japanese system, it is in English.

30. IY51054: On HP-UX, the name and arguments filters of the UNIX Process resource model were limited to a string of no more than 64 characters. Similarly, the basename filter was limited to 15 characters. These restrictions are removed, but, on HP-UX 11.00, the HP patch PHKL_26008 is required.

31. IY51081: There is an inconsistency between the User Guide and the wdmconfig behavior.

32. IY51191: HEARTBEAT_EndpointMigrated is not generated after endpoint migration to another gateway.

33. IY51260: The arguments filter for the UNIX Process resource model was limited to a length of 250 characters. This was changed to 4096 characters.

34. IY51335: the script dmae_tec_inst.sh used to update the rule will not complete, if current rulebase includes any baroc files other than basic TEC or Tivoli Monitoring files, as Websphere, Tivoli Monitoring for DB2. It will delete all the baroc files correctly but it will fail to import them back, with error message "Unable to import baroc file".

35. IY51374: Prior to Tivoli Monitoring FP05, installation of an ODBC driver on windows endpoints was optional, as detailed in the 5.11 User’s guide. The only side affect of not having ODBC present on the EP was the lack of functionality in respect of data logging. However, post FP05, it appears that an ODBC driver has now become mandatory. Windows NT 4.0 does not come with any ODBC support - and some customers will not install ODBC if they do not require it. The Tivoli Monitoring engine does not work.

36. IY51401: Malformed XML files are generated by Windows Engine under certain conditions.

37. IY51517: Cannot use 24:00 as end of schedule.

38. IY51572: Unable run init script from 5.1.1-Tivoli Monitoring-FP05 on interconnected TMRS. The wdmrm command fails with the following error message: Unable to create entries in the name registry.

39. IY51584: The wdmeditprf -Schedule option does not work correctly.

40. IY51760: The UNIX Process resource model generated ProcessKilledOrNotExisting indications on computers with more than 2000 running processes. The 2000 process limit was removed.

41. IY51761: The Tivoli Monitoring 5.1 commands wdmdumpprf and wdmloadprf work successfully only if the dump is taken at the same patch level as the load.

42. IY51843: Endpoint migrations in an environment with firewalls between the Gateways results in a large number of _get_label methods in "rwait".

43. IY51859: Tivoli Monitoring 5.1.1 FP05 wdmeditprf edit schedule change is not reflected correctly through the desktop.

44. IY51879: The DMEndpointUninstall command removes not only Tivoli Monitoring dll’s.

45. IY51910: A Managed Node that is not a Gateway can create problems with endpoint displaying using WHC.

46. IY52097: The wmdistrib -o delete option (as seen in ITMUnsubscribe policy method script) does not work on application specific subscribers.
47. IY52098: ITMUnsubscribe validation policy script lacks quotes for variable storing profile manager name, causing problems for profile manager with spaces.
48. IY52224: On AIX, the UNIX Security resource model could send a "HighLoggingNumber" indication even though the number of logins for a specific user ID did not exceed the threshold.
49. IY52235: The UNIX Process resource model does not send a "process killed" indication if the filter includes a user or group name that does not exist on the computer.
50. IY52262: Heartbeat uses incorrect IP address in HA environment to determine health.
51. IY52296: The built-in task "Delete Registry KEY Shares" does not work, it does not actually delete the entry in the registry.
52. IY52500: Using the ITMUnsubscribe val policy from 5.1.1-Tivoli Monitoring-FP05 fails to delete local copies when used on a ProfileManager with a subscribed DataLess-ProfileManager.
53. IY52573: The UNIX CPU resource model would not trigger a LowIdleCpuUsage indication even though the sar command shows the CPU completely busy. The problem occurred on HP-UX.
54. IY52584: Documented needs to be added to user guide for troubleshooting missed prorate errors.
55. IY52687: On HP-UX, the UNIX Process resource model could fail to gather process data. The problem occurred on 64-bit machines which also contained large amounts of memory (gigabytes).
56. IY52706: wdmconfig -m MN -D trace.filename=filename return "unknown property".
57. IY52925: Tmw2k log indicates the error: "DBManager 1004 Something failed, removing XML file. An error has occurred retrieving warehouse data Warehouse data logging is not enabled on any of the resource models on the endpoint" when logging is not enabled for the resource model.
58. IY52968- IY52430: After several days of running the ITM_engine on the UNIX EP there are a lot of CLOSE_WAIT sessions from "random" ports to the EP listening port.
59. IY52999: In the DMXProcess Resource Model, if the provider tries to return a long string, this string results corrupted.
60. IY53037: Custom Resource Models on Unix might fail after applying patch 5.1.1-Tivoli Monitoring-0090LA.
61. IY53142: On HP-UX, the UNIX Process resource model would not send ProcessHighCPU indication.
62. IY53395: On HP-UX, the UNIX FileSystem resource model reports a status of "Error" in response to a wdmseng command. The trace log shows that a NullPointerException has occurred when the resource model attempts to get a list of supported file system names.
63. IY53365: Downloading of a Custom Resource Model containing a class file as a dependency on a Unix endpoint, modifies the file size.
64. IY53409: Tivoli Monitoring 5.1.1 Fix Pack 5 causes wdmeditprf to generate superfluous error message ‘AMW0252E missing argument’ ... followed by usage page. Prior to FP05 the following syntax produced the correct output: wdmeditprf -P <tmw2kprofile> -print <resource model> -Tec

Now after FP05 is installed, the same syntax produced the valid TEC settings output followed by the error message ‘AMW0252E missing argument’
65. IY53418: When installing the Java Runtime Environment 1.4 with task DMLinkJre on a Windows XP endpoint, the task completes successfully, but the registry entry JreType is set incorrectly to server instead of client.

66. IY53611: wdmcollect -m <MN> -e <EP> -s <time-interval> fails and the following error is displayed:
   
   exception generated:The endpoint <EP> is not valid
   
   Note: The root cause of the problem is that the epcache might get corrupted when removing endpoints.

67. IY54070: The datacollector does not work with Oracle 9.2

68. IY54335: Get <not a string> message for fqhostname - CorrelatedEvent. A <not a string> message could be got in the fqhostname tec event slot on our TMW_CorrelatedEvents or custom RMs.

69. IY52958: The RIM Handler is not able to load XML file created before fix pack 5.1.1-Tivoli Monitoring-FP05 was applied on the endpoint. Under this conditions, the tmnt_rimh_eng process can core dump.

70. IY53183, IY51401: When using Data logging in Tivoli Monitoring TDW Data collection extracts Metrics Data stored locally at the Endpoint keyed to IIDs missing from the Instances Table.

71. IY46180: AMX SQL file names wrong case when running AMX on UNIX system with TEDW CDW component and Remote Agent.

72. IY49091: Running the AMY_m15_s040_fact Step in the Data Warehouse Center GUI for the AMY Warehouse Pack sometimes causes the following error:
   
   "CDW8087E: IBM CLI Driver DB2/NT SQL0803N One or more values in the INSERT statement, UPDATE statement, or foreign key update caused by a DELETE statement are not valid because the primary key, unique constraint or unique index identified by "I" constrains table "AMY.F_CPU_HOUR" from having duplicate rows for those columns. SQLSTATE=23505"

73. IY51291: AMX does not pull any from the RIM database source. The problem was due to the fact that the twh_run.sh that should be sourced at the beginning of any AMX shell script was not called, so perl was not found and thus not invoked in the rim_extract.sh script; by so the amx_c05_s020_rim_extract scripts were not complete and executing them did not extract any data.

74. IY53012: AMX_c05_s040_Comp_Msmt.db2 sql script declares a temporary table stage_AMY_PROCESS_AMY_PROCNM with a size for the Attr_val column that is too short. The column we select from (AMX.Stage_instkey_no_ex's Property_val) is 254 bytes wide while the column we insert into is only 120 bytes wide. When running this step, the following error occurs:
   
   CDW8087E (22001) : (Err:-433) IBM CLI Driver DB2/6000 SQL0433N

   Value "<character string greater than120 bytes>" is too long SQLSTATE=22001
Chapter 4. Documentation notes

This section describes the changes and upgrades that have been implemented after Tivoli Monitoring 5.1.2 release, and could not be included in the documentation supplied with the above release.

New information

This section documents late-breaking information that could not be included in the library documents.

About the Web Health Console

Starting from Tivoli Monitoring 5.1.2 Fix Pack 5, the Web Health Console based on WebSphere Application Server 5.0 is supported on the following platforms:

- Linux RH 2.1
- AIX 5.1 and 5.2
- Windows 2000 and 2003
- Solaris 2.8
- HP 11.11i

Note: Starting from Tivoli Monitoring 5.1.2 Fix Pack 5, the Web Health Console based on WebSphere Application Server 5.0 is released as a web application in EAR format.

About resource models

The following information describes resource models and their related functions.

About OS/400 resource models

Starting from Tivoli Monitoring 5.1.2 Fix Pack 8, the following two new resource models have been added for OS/400 systems:

- JobPerformance Resource Model
- MessageQueue Resource Model

See the IBM Tivoli Monitoring: Resource Model Reference Guide for more details about the resource models.

About the Process resource model on UNIX, Linux, and Windows

Process resource models on all the platforms have been enhanced. Starting from Tivoli Monitoring 5.1.2 Fix Pack 6, these resource models have the Process parameter that contains new property filters, consisting of name=value pairs. In addition, some of these filters support the use of wild cards. For more information about this enhancement, see IBM Tivoli Monitoring: Resource Model Reference Guide.

About the UNIX Process resource model

The ILT interface of the DMXProcess resource model has been changed in IBM Tivoli Monitoring 5.1.2 Fix Pack 1. If you use the DMXProcess ILT in your custom resource model, you can add the following lines in the "Init" function:

Svc.AssociateParameterToClass("cpuThr","DMXProcess");
Svc.AssociateParameterToClass("processes","DMXProcess");
"processes" is a string parameter with the list of processes to monitor, and "cpuThr" is a numeric parameter used as threshold for the percentage of CPU used by a single process.

The ILT returns data from all processes that are in the "processes" list or that exceed the "cpuThr" threshold. If you do not specify the processes parameter, then the ILT returns data from all processes. To use the threshold that you have set in the resource model as the threshold to filter the processes, you can add the following lines in the "Init" function:

```java
thresHighCPUUsed = Svc.GetThreshold("HighCPUUsed");
Svc.DefineNumParameter("cpuThr", thresHighCPUUsed);
```

For an example, refer to the RMB workspace of the DMXProcess resource model.

Starting from Tivoli Monitoring 5.1.2 Fix Pack 13, wildcards are supported in the name parameter (addressed by APAR IY97125).

**DMXCpu resource model on UNIX**

Starting from Tivoli Monitoring 5.1.2 Fix Pack 10, DMXCpu resource model reports the same CPU values as the AIX performance commands on a eServer P5 Logical Partition (LPAR) running AIX 5.3, regardless of LPAR being a Shared Processor Partition (SIPAR) (Micro-Partitioning technology) or not. However, even when used in a shared processor partition, DMXCpu resource model does not provide any information about the Entitled Capacity and the percentage of Entitled Capacity used.

**DMXFileSystem resource model**

On AIX platforms, DMXFileSystem resource model is enabled to monitor General Parallel filesystems (GPFs). The AIX GPFS supported versions are 2.3 and 3.1.

On AIX platforms, DMXFileSystem resource model is not enabled to monitor Veritas filesystems (VxFS).

On Solaris platforms, starting from Tivoli Monitoring 5.1.2 Fix Pack 14, DMXFileSystem is enabled to monitor also ZFS file systems.

On Linux platforms, starting from Tivoli Monitoring 5.1.2 Fix Pack 16, journaled file systems percentage metrics will not consider the amount of space used by journals. Resource model percentage metrics could differ from `df -k` ones.

**Solaris zone support**

Starting from Tivoli Monitoring 5.1.2 Fix Pack 11, Solaris global and local zone support have been introduced with the following behavior and impact on the specific resource models:

**DMXProcess**

It allows filtering processes in the global zone so that processes in the local zones do not appear in the list of processes for the global zone. The following parameter and property have been added to support the new feature:

**Processes Filter**

This parameter controls whether the global zone process filtering is active or inactive. This parameter is Boolean and the default is for the filtering to be inactive, which means that running DMXProcess in the global zone returns processes for all zones.
**DMXProcess CIM Class property**

This class contains the value of the zone ID associated with a process. This ID is collected by the Solaris resource model provider, and any resource model can then use the zone ID to obtain the zone name.

Because the DMXProcess resource model is supported on systems other than Solaris with zones, in those environments the zone ID will always be set to 0. If the monitored system does not support zones, the parameter is ignored.

**Note:**
The DMXProcess resource model does not use the new zoneID parameter as part of its calculations. The zone ID parameter is available in the .mof file specifically for anyone writing custom resource models.

**DMXNetworkInterface**

It can monitor physical interfaces only. If the local zone has only logical interfaces the resource model excludes them and the status will be Failing(1)

**DMXCpu**

Because this resource model uses the Solaris kstat APIs to accumulate the data about each CPU, overall processor usage can be determined. Running DMXCpu in the global zone will gather metrics about all CPUs. Running DMXCpu in a local zone will gather metrics about all CPUs that the local zone can see.

**Note:**
Local zones cannot determine how much CPU is spent specifically on each zone, unless CPUs are dedicated to the local zone.

**DMXFile**

It monitors a user-defined list of files. Running this resource model in the global zone allows you to specify any file visible to the global zone (which should be any file). Running this resource model in the local zone requires that you specify files visible to the local zone.

**Note:**
Some files can be shared between zones depending on how the zones are configured.

**DMXFileSystem**

This resource model monitors all UFS and VXFS filesystems by default. Therefore, you can run this resource model in the global zone to monitor all filesystems or you can specify a list of filesystems to ignore so that the local zone filesystems are ignored when running in the global zone. Running this resource model in the local zone will only monitor the filesystems defined for the local zone.

**DMXMemory**

The memory data collected for local and global zones comes from the kstat API and is the same whether you are in a global or local zone.

**DMXNetworkRPCNFS**

The network RPC NFS data collected from kstat in a local zone is specific to the local zone. In a global zone, you get global data.
DMXPhysicalDisk
The same statistics about the physical disks are reported in the local and global zones.

DMXSecurity
Monitors critical files and user logins. The monitored user logins are specific to the zone.

About Heartbeat process
With Tivoli Monitoring 5.1.2 Fix Pack 4, heartbeat performance has been improved. Refresh time of data collected on endpoints has significantly reduced.

The following sections document various changes that have been implemented in the heartbeat processor since the Tivoli Monitoring 5.1.2 release. They list the main changes and corresponding new configuration keys following the upgraded heartbeat features.

Event notification
The heartbeat process allows you to send automatic event notifications to the Tivoli Enterprise Console server and to the Notice Groups. This is a short summary of the heartbeat new features, implemented after the Tivoli Monitoring 5.1.2 release:

• You can customize the severity of the events sent by the heartbeat. In particular, you can activate a mechanism to increase the severity of events at each heartbeat interval for as long as the problem that generated the event persists. When this mechanism is active, it overrides any customization applied to the severity of events.
• You can send events to multiple secure and unsecure Tivoli Enterprise Console servers, in either broadcast or fail-over mode, and cache events when none of the servers is available.

For event notification, use the following keys:
• heartbeat.engine_alive_event_severity
• heartbeat.engine_down_event_severity
• heartbeat.engine_restarted_event_severity
• heartbeat.ep_migrated_event_severity
• heartbeat.ep_not_updated_event_severity
• heartbeat.ep_unreachable_event_severity
• heartbeat.gateway_down_event_severity
• heartbeat.hboff_event_severity
• heartbeat.no_data_event_severity
• heartbeat.rms_in_error_event_severity
• heartbeat.severity_incremental_step
• heartbeat.severity_escalation

For more details about the configuration keys, see “New configuration keys” on page 34.

When multiple Tivoli Enterprise Console servers are defined in the heartbeat configuration file, the default behavior is that the heartbeat sends an event to all servers (broadcast mode). So, for example:
• There are two servers (server1 and server2), both of which are active: the event is sent to both servers.
• There are two servers (server1 and server2), one of which is down (server2): the event is sent to server1 and is cached in the tec\cache\server2.cache file (under the directory defined by the adapter.working.dir key).

• There are two servers (server1 and server2), both of which are down: the event is cached in both cache files (tec\cache\server1.cache and tec\cache\server2.cache).

You can enable the failover mode by setting the heartbeat.tec_fail_over configuration key to true. If you do, the event is sent only to the first active server. So, for example:

• There are two servers (server1 and server2), both of which are active: the event is sent only to the first server (server1).

• There are two servers (server1 and server2), one of which is down (server1): the event is sent only to server2.

• There are two servers (server1 and server2), both of which are down: the event is cached in the cache file of server1 (tec\cache\server1.cache).

For the configuration broadcast mode, use the following keys:

• heartbeat.tec_fail_over
• heartbeat.tec_server#
• heartbeat.usecure_tec_server
• heartbeat.unsecure_tec_server#

More keys that are not new but are related to broadcast configuration and must be used in connection with the above, are the following:

• heartbeat.send_events_to_notice
• heartbeat.send_events_to_tbsm
• heartbeat.send_events_to_tec
• heartbeat.tec_server

See the IBM Tivoli Monitoring: User’s Guide for more details about these keys.

**Enhanced failure detection capability:** The following new events are sent by the heartbeat after detecting failures. These events are not related to a specific endpoint but are associated to the heartbeat process and related to all the endpoints.

**GatewayDown’ and ‘Heartbeat_EndpointGatewayDown**

Sent when the heartbeat detects failures on the gateway, if the gateway is down.

**Heartbeat_EndpointStatusNotUpdated**

Sent when the heartbeat detects delays in the polling mechanism through which the heartbeat monitors the endpoints. If a delay is detected, an event is sent and the endpoint status is set to ’NotUpdated’. This capability is deployed through the heartbeat.check_delays configuration key.

**HeartBeat_RequestManagerNoData**

Sent when the heartbeat does not receive any data from the Request Manager, due to problems with the Request Manager.

**HeartBeat_FrameworkFailure**

Sent when the heartbeat does not receive any data from the Request
Manager, due to problems with Tivoli Management Framework. This event might not be sent if any of the Framework processes responsible for handling TEC events is not working.

**Heartbeat_Clearing**
Sent when either one of the two previous problems has been solved and the connection with the managed node has been restored.

**Heartbeat_EndpointStatusNotUpdated**
The event is generated when the heartbeat is not able to monitor the endpoints with the specified frequency or when the Request Manager fails to perform a request on a specific endpoint.

**Heartbeat_DMAgentRestarted**
Sent when the agent has been restarted

For the enhanced failure detection capability, use the following keys:
- heartbeat.check_delays
- heartbeat.no_data_event
- heartbeat.skip_cycle

**TCP/IP scan function**
When the heartbeat cannot retrieve data from an endpoint, it marks it as unreachable. To find out whether the system is actually down, you can try to establish a TCP/IP connection, enabling the `heartbeat.tcp_ports_to_scan` configuration key of the `wdmconfig` command by specifying one or more port numbers. When this key is enabled, the heartbeat tries all the ports you have specified looking for the first TCP/IP successful connection. It waits a timeout period on each port before trying to connect again, and stops only when it establishes a successful connection or when it has tried all the specified ports. The timeout period can be set using the `heartbeat.tcp_timeout` configuration key. You can also specify the number of attempts the heartbeat must make to establish a TCP/IP connection after the first failure (`heartbeat.tcp_retries` configuration key) and the wait time between TCP/IP connection attempts (`heartbeat.tcp_retry_interval` configuration key).

The ports you specify must be open and a service (or daemon) listening on them. If a firewall exists between the gateway and the endpoint, make sure that the ports on the firewall are open. Each time you change the value of any of the above configuration keys, you have to restart the request manager to enable them.

The TCP/IP scan function returns these endpoint states:

**OK** A TCP/IP connection with the endpoint has been successfully established.

**FAIL** A TCP/IP connection has been tried through all the specified ports, but no connection could be established.

**N/A** It was not possible to perform a TCP/IP scan.

Note: When the Heartbeat_EndpointStatusNotUpdated event occurs, the TCP/IP scan results relate to the latest successful endpoint update.

To retrieve the TCP/IP address of an endpoint to be scanned, the heartbeat uses one of the following methods (according to the value you set for the `heartbeat.endpoint_address` parameter):
- It queries the system to resolve the hostname of the endpoint (`dns`)
- It uses the address configured locally on the endpoint (`cache`)
• It tries to query the system, if it fails, it uses the local address (both)
• It uses the address used by Tivoli Framework and stored in the Endpoint Manager (epmgr)

If you specify the latter, all the endpoint hostnames and the corresponding TCP/IP addresses are retrieved and stored in a cache internal to the Request Manager. The cache is refreshed periodically, according to the value you specify for the parameter **heartbeat.address_refresh_interval**. You can specify an interval (in seconds) or leave the default value (zero). If the parameter is set to zero, the cache is initialized when you start the heartbeat, and its content remains unchanged until the heartbeat or the Request Manager is restarted. You might prefer this choice from a performance point of view. However, consider that if an endpoint changes its address after the heartbeat has started, the heartbeat will not handle this change and continue to attempt TCP/IP connections using its previous (and invalid) address until it is restarted.

For this function, use the following keys:
• **heartbeat.address_refresh_interval**
• **heartbeat.endpoint_address**
• **heartbeat.tcp_ports_to_scan**
• **heartbeat.tcp_retries**
• **heartbeat.tcp_retry_interval**
• **heartbeat.tcp_timeout**
• **heartbeat.use_address_cache**
• **heartbeat.use_port_range**

See "New configuration keys" on page 34 for details.

**New slots in the Heartbeat_Event class**

New slots have been defined in the Heartbeat_Event class:

<table>
<thead>
<tr>
<th>Slot</th>
<th>Slot description</th>
<th>Slot value</th>
</tr>
</thead>
<tbody>
<tr>
<td>interp</td>
<td>The operating system platform of the endpoint where the event occurred</td>
<td>none</td>
</tr>
<tr>
<td>tcp_status</td>
<td>The result of the TCP connection test</td>
<td>none</td>
</tr>
</tbody>
</table>

**Heartbeat miscellaneous**

The following list summarizes various heartbeat functions that have been added or improved after the Tivoli Monitoring 5.1.2 release.

• You can change the number of threads that the heartbeat uses to send events to the Tivoli Enterprise Console server and to run recovery actions. See the **heartbeat.max_threads** configuration key for details.
• **wdmmngcache** now also reports detailed information about the endpoint status.
• The response time of Windows endpoints is improved.
• The heartbeat unsecure tec events definition key-value pair has changed. In previous releases, the syntax to define an unsecure tec server, was the following:
  
  ```
  heartbeat.unsecure_tec_server#1 = ip_address,port
  ```

  Now, the comma (,) between the ip_address and the port number has been replaced by a plus (+). It has changed as follows:
  
  ```
  heartbeat.unsecure_tec_server#1 = ip_address+port
  ```
The old syntax is still supported but there is a message in the Heartbeat Trace stating that the use of a comma (,) as separator is deprecated, and it is suggested to use the plus (+) as separator.

**New configuration keys**
The following list summarizes in alphabetical order all the new heartbeat configuration keys implemented after the Tivoli Monitoring 5.1.2 release.

The new configuration keys can be set for the heartbeat function using the **wdmconfig** command. The new keys are:

**heartbeat.address_refresh_interval**
Defines the time interval (in seconds) after which the cache containing the IP addresses retrieved from the EndpointManager is refreshed. The default value is zero, which means that the cache is initialized when the heartbeat request is created and its content remains unchanged until you create a new request or until the Request Manager is restarted. To use this key, you must have selected the **epmgr** option of the heartbeat.endpoint_address.

**heartbeat.check_delays**
Set this to false if you want to disable the capability to send the ‘Heartbeat_EndpointStatusNotUpdated’ event. The default value is true.

**heartbeat.endpoint_address**
Defines the address of the endpoint. It can have the following values:

- **Dns**
  - It is the default value. It is the IP address, as it results from gethostbyname.

- **Cache**
  - It is the IP address stored in the endpoint cache, which is the IP address you have configured locally on the endpoint.

- **Both**
  - It tries to use the dns default, but if this fails, it uses the IP address stored in the cache.

  **Note:** This value is deprecated. Starting from Tivoli Monitoring 5.1.2 Fix Pack 9, it will no longer be supported, but will be replaced with a new string.

- **epmgr**
  - It retrieves the IP address stored in the EndpointManager. It is the same address used by Tivoli Framework. If you select this option, the addresses are stored in an internal cache. You can define the cache refresh time using the heartbeat.address_refresh_interval parameter.

**heartbeat.engine_alive_event_severity**
Defines the severity for the ‘Hearbeat_DMAgentAlive’ event. The possible values are harmless, warning, minor, critical, or fatal. The default value is harmless.

**heartbeat.engine_down_event_severity**
Defines the severity for the ‘Hearbeat_DMAgentDown’ event. The possible values are harmless, warning, minor, critical, or fatal. The default value is critical.

**heartbeat.engine_restarted_event_severity**
Defines the severity for the ‘Hearbeat_DMAgentRestarted’ event. The possible values are harmless, warning, minor, critical, or fatal. The default value is warning.
heartbeat.ep_migrated_event_severity
Defines the severity for the ‘Heartbeat_EndpointMigrated’ event. The possible values are harmless, warning, critical, or fatal. The default value is warning.

heartbeat.ep_not_updated_event_severity
Defines the severity for the ‘Heartbeat_EndpointStatusNotUpdated’ event. The possible values are harmless, warning, minor, critical, or fatal. The default value is warning.

heartbeat.ep_unreachable_event_severity
Defines the severity for the ‘Heartbeat_EndpointUnreachable’ event. The possible values are harmless, warning, minor, critical, or fatal. The default value is critical.

heartbeat.severity_escalation
Allows the definition of a dynamic custom severity escalation sequence (only if the heartbeat.severity_incremental_step key is enabled). When the Heartbeat starts up, it loads the TEC event message as usual, and this mapping represents the default configuration for the new key. To modify it, run the wdmconfig -B command to set a new value for this key. The value of this key must be a sequence of the allowed TEC event severities separated by commas. Possible values are harmless, warning, minor, critical, and fatal. The key values must be in escalation order. The Heartbeat must be restarted for the newly configured key to take effect. If this key does not exist, or is left blank, the default sequence is used.

heartbeat.gateway_down_event_severity
Defines the severity for the ‘Heartbeat_EndpointGatewayDown’ event. The possible values are harmless, warning, minor, critical, or fatal. The default value is warning.

heartbeat.hboff_event_severity
Defines the severity for the ‘Heartbeat_Off’ event. The possible values are harmless, warning, minor, critical, or fatal. The default value is harmless.

heartbeat.max_threads
Specifies the number of threads that the heartbeat uses to send events to the Tivoli Enterprise Console server and to run recovery actions. The value can be tuned depending on the workload of the managed node on which the heartbeat runs. Important factors are:
- the number of endpoints
- the frequency by which the endpoints change status
- the actions (events, notices) that are run when the status changes

The default value of this key is 10. Values between 10 and 100 should be adequate for all environments.

heartbeat.no_data_event
The default value is false. If you set it to true, the heartbeat process sends an event to the Tivoli Enterprise Console server specifying that it cannot retrieve any data from a managed node (and, consequently, from all the endpoints connected to it) due to some connection problems with the Request Manager or to a Tivoli Management Framework Failure.

heartbeat.no_data_event_severity
Defines the severity for the ‘HeartBeat_RequestManagerNoData’ and ‘HeartBeat_FrameworkFailure’ events. The possible values are harmless, warning, minor, critical, or fatal. The default value is warning.
heartbeat.rms_in_error_event_severity
Defines the severity for the ‘Heartbeat_ResourceModelsInError’ event. The possible values are harmless, warning, minor, critical, or fatal. The default value is critical.

heartbeat.skip_cycle
This configuration key is enabled only when the heartbeat.no_data_event key is set to true, and its default value is equal to the value set for the heartbeat.no_data_event key. After a heartbeat error condition has recovered, if you have set this key to true, the heartbeat sends a ‘Heartbeat_Clearing’ event and waits a monitoring cycle before updating the cache and sending possible further events.

heartbeat.severity_incremental_step
For the following events:
- ‘Heartbeat_EndpointUnreachable’
- ‘Heartbeat_EndpointGatewayDown’
- ‘Heartbeat_EndpointStatusNotUpdated’

it defines the number of monitoring cycles after which the severity of the event is incremented. When this key is set, the initial value of the severity is always warning. The default value of this key is 0 and, when set to this value, it disables the mechanism. The default severity sequence is: WARNING, CRITICAL, and FATAL.

heartbeat.tec_fail_over
Set this to true if you want the set of Tivoli-secure Enterprise Console servers (that you have defined using the heartbeat.tec_server#n key) to work in fail-over mode. If a server fails to receive an event, the event is routed to the next available server (following the order established by the n value). If no server is available, the event is cached in the tec\cache\servername.cache file under the directory defined by the adapter.working.dir key. The default value of heartbeat.tec_fail_over is false.

heartbeat.tcp_ports_to_scan
Specify the port numbers you want to be used to establish a TCP/IP connection. The port numbers must be separated by a comma. By default, this key does not contain any value. For performance reasons, you should not define too many ports: keep the number small (two or three well-known and open ports). If a firewall exists between the gateway and the endpoint, make sure that the ports on the firewall are open. Each time you change the value of this configuration key, restart the request manager to enable it. The result of the TCP/IP connection test is reported in the tcp_status slot of the Heartbeat_Event class.

heartbeat.tcp_retries
Specify the number of TCP/IP connection attempts the heartbeat must make after the first failure. The default value is zero.

heartbeat.tcp_retry_interval
Specify the number of seconds the heartbeat must wait between a TCP/IP connection attempt and the following one. The default value is zero.

heartbeat.tcp_timeout
It sets the timeout period expressed in seconds (overwriting the system default settings) for the TCP/IP scan process. It is the time the process waits before considering a port unavailable and trying with the following port. The default value is five seconds.
heartbeat.tec_server
If you have set heartbeat.send_events_to_tec to true, enter here the name of a Tivoli-secure Enterprise Console server that will receive the heartbeat events. You can define more than one server. The integer \( n \) allows to define the order through which servers are contacted when working in fail-over mode.

heartbeat.unsecure_tec_fail_over
Set this to true if you want the set of Tivoli-unsecure Enterprise Console servers (that you have defined using the heartbeat.unsecure_tec_server\( n \) key) to work in fail-over mode. If a server fails to receive an event, the event is routed to the next available server (following the order established by the \( n \) value). If no server is available, the event is cached in the tec\cache\servername.cache file under the directory defined by the adapter.working.dir key. The default value of heartbeat.unsecure_tec_fail_over is false.

heartbeat.unsecure_tec_server
If you have set heartbeat.send_events_to_tec to true, enter here the IP server name (IP address or fully qualified name) and IP port of the Tivoli-unsecure Enterprise Console server to receive the heartbeat events. Use a plus sign (+) to separate the IP server name and the IP port. If you want the events to be sent to more than one unsecure server, use the heartbeat.unsecure_tec_server\( n \) key.

heartbeat.unsecure_tec_server\( n \)
If you have set heartbeat.send_events_to_tec to true, enter here the name of a Tivoli-unsecure Enterprise Console server that will receive the heartbeat events. You can define more than one server. Use integer \( n \) to define the order in which servers are contacted when working in fail-over mode.

heartbeat.use_address_cache
The default value is false. Set it to true if you want to store the endpoint IP addresses in the heartbeat cache. In this way, you can improve TCP/IP scan performance. However, it is recommended that you only enable this key in environments where IP address resolution is very slow. When this key is enabled, each time an endpoint changes its address, the request manager must be restarted to update the heartbeat cache.

heartbeat.use_port_range
Default value is false. Set it to true if you want the TCP/IP scan process to use only the limited port range configured with Tivoli Management Framework, launching the odadmin set_port_range port1-port2 command.

Note: The Event Integration Facility (EIF) libraries are used to send heartbeat events to secure and unsecure Tivoli Enterprise Console servers. A configuration file (servername.conf, located in the tec\conf directory under the directory defined by the adapter.working.dir key) exists for each Tivoli Enterprise Console server. You cannot modify the entries in this configuration file.

Excluding endpoints from the heartbeat process
Starting from IBM Tivoli Monitoring 5.1.2 Fix Pack 9, you can exclude one or more endpoints from the heartbeat process.

To do this, specify a local setting on the endpoint, using the wdmepconfig command with the following new parameter:
HeartbeatEnabled
When set to false, it excludes the endpoint from the heartbeat process. The default value is true.

Because this is a local setting, to make it effective, you must restart the engine, so that the new configuration is applied.

The endpoints that you exclude from the heartbeat process are displayed in the wdmncache output as HBOff, under the HB Status column, and as Disabled under the HB Enabled new column.

About Tivoli Monitoring functions
This section describes improvements and changes in Tivoli Monitoring functions.

Enabling IBM Tivoli License Manager
Starting from IBM Tivoli Monitoring 5.1.2 Fix Pack 2, Tivoli Monitoring is enabled to work with Tivoli License Manager so that this latter can detect the installation and use of Tivoli Monitoring and of its applications.

A new command (wdmlsitlm) is available to provide information on which "IBM Tivoli Monitoring for ..." applications are in use at an endpoint and if there is an associated license.

To use this feature, download the IBM Tivoli License Manager information to the endpoints using the wdmcmddistrib -upgrade command against the endpoints. After the distribution, wait some time (normally minutes) and restart the Tivoli Monitoring engine to enable the wdmlsitlm command to generate the correct output. Occasionally, a second restart is needed, if the first one was not effective (in this case the wdmlsitlm command would return ITLM_UNKONW_PRODUCT status for licensed products).

wdmlsitlm
Provides a list of all resource models and corresponding applications in use at an endpoint. In addition, the output indicates if an associated license exists.

Note: After distributing the IBM Tivoli License Manager information to the endpoints, restart the engine to make wdmlsitlm command generate the correct output and a certain time (normally minutes) must be waited between the completion of the wdmcmddistrib -upgrade and the Tivoli Monitoring engine restart. Occasionally, a second restart is needed, if the first one was not effective (in this case the wdmlsitlm command would return ITLM_UNKONW_PRODUCT status for licensed products).

Usage
wdmlsitlm –e endpoint [ –CC chargeable_component | –rm resource_model ]

Purpose
The wdmlsitlm command tells which resource models and corresponding applications (chargeable components) are in use at an endpoint, and whether a license exists to use those applications. The output provides the following information:
• The resource models in use
• The name of the corresponding profile
• The corresponding application (chargeable component)
• The availability of a license
• The license status

Parameters

-e endpoint
   Specifies the name of the endpoint for which the information has to be provided.

-CC chargeable_component
   Provides a list of the resource models in use for a specified chargeable component and tells whether an associated license exists.

-rm resource_model
   Provides a list of the chargeable component associated to a specified resource model and tells whether a license exists.

Authorization
   Requires the super, senior, admin, or user roles.

Examples

1. The following example provides the list of all resource models in use at endpoint MyEndpoint, the corresponding chargeable components, and whether an associated license exists:
   wdmlsitlm -e MyEndpoint

2. The following example tells which chargeable component is associated to resource model MyResourceModel, and whether a license exists:
   wdmlsitlm -e MyEndpoint -rm MyResourceModel

Configuring data collection

Starting from Tivoli Monitoring 5.1.2 Fix Pack 4 you can collect data from the endpoints without having to write it immediately to the database. The data collection and data writing steps are now separate. This means that, if you have to disconnect the database for any reason, you can still perform data collection. You need only to configure the data collector accordingly, and the automatic upload of collected data into the ITM_DB is disabled. Data is collected at the endpoint database (in XML files) and stored in ZIP files located under $DBDIR/dmml/tedw/endpoint_name, on the managed node you specified with -m option when you launched wdmcollect command, as shown in the following command usage:

Then, you can upload data stored in the ZIP files at a later time using -l option of wdmcollect command, as shown in the following command usage:

To configure the data collector, the following new key-value pairs have been added to the wdmconfig command:

datacollector.db_deep_purge
   The default value is false. If you set it to true, you enable a new ITM_DB purge method that performs a deep purge, cleaning unnecessary data from all the ITM_DB tables (the standard method only cleans CATEGORIESDATA and METRICSDATA tables).

datacollector.insert_data
   Set this to false, if you do not want to load collected data into the database, but do want it on the managed nodes. The parameter default value is true. If you leave the default, then you can set the datacollector.enable_table_loader parameter to decide where to store the metricsdata (see the corresponding option on page 40).
**datacollector.is_master**

The default value is **true**. When set to true, the data collector is enabled to update the control table and lock the database for the time required for this update. Within a Tivoli management region, this parameter must be set to **false** for all the data collectors except one, and the master datacollector must have at least one endpoint configured for data collection.

**datacolletor.prune_successful**

The default value is **false**. If you set it to **true** you enable Tivoli Monitoring to use the Tivoli Management Framework parameter for MDist2. Use this parameter to handle communication failures during .zip file uploads.

After updating the database, the data collector temporarily locks it and inserts in the control table the last record it wrote in the database. The AMX checks the control table and, using this record, retrieves from the database only the data written between the last AMX reading and the last data collector writing. In an environment with more than one data collector, you must configure only one of them to have permission to lock the database and write in the control table, otherwise this process does not work properly.

**Using a table loader**

Starting from Tivoli Monitoring 5.1.2 Fix Pack 5, you can enable the data collector to download metrics data to a file, instead of writing it to the database. The data collector produces a dump file for each XML provided by the endpoint. You can then use a table loader program (such as SQL Loader or DB2 import) to upload the data of the dump files into the database.

To configure the data collector to write to a file, change its configuration using the wdmconfig -D command and modifying the following parameter:

**datacollector.enable_table_loader**

The default value is **false**. If you set this parameter to **true**, the data collector, instead of inserting metricsdata into the database, dumps it into a file. The configuration of this parameter is only considered if data collector.insert_data is set to true, otherwise it is ignored.

To send metricsdata to a file, the datacollector does the following:

   
   The directory content is the following:
   
   - `$FILENAME.xml`
   - `$FILENAME.xml.md.itmdmp`

2. Dumps the metricsdata table into the file: `$FILENAME.xml.md.itmdmp`.

   The format of the dump file is a list of fields separated by a separator. You can specify the separator character by configuring the following parameter of the wdmconfig -D command:

**datacollector.field_separator**

The default value is the comma. However, you can set whatever separator you like, provided it is compliant with the table loader you use and it is not a symbol that the system uses as the decimal separator. The separator must be specified between single quotes ('), as shown in the following example:

```
wdmconfig -D datacollector=';'
```
When this feature is enabled, the measurement data is written into a dump file, one file for each XML. You then, have to load data manually into the database. To optimize the advantages of this process and the table loader capacity, collect several dump files and load them altogether into the database. Loading them one by one might affect performance. You can also do this by creating a custom script, such as the following:

```sh
#!/bin/sh
find . -type f -name 'ITM511_WH@*' |
while read file ;
do if [-f "file.md.itmdmp"] ;
then cat "file.md.itmdmp" >> metricsdata.dump ;
echo $file >> metricsdata.list ;
fi ;
done;
```

**Uploading data on DB2 database:** To upload the content of dump file on a DB2 database, launch the following command:

D:\work>db2 import from metricsdata.dump of del insert into db2.metricsdata(metricvalue,timekey,eid,mid,iid)

where metricsdata.dump is a single file that includes the content of all the separate dump files.

**Uploading data on Oracle database:** To upload the content of dump file on an Oracle database, launch the following command, customizing it with the options you require (for more details, refer to Oracle documentation):

sqlldr control=metricsdata.sqlldr.ctl data=metricsdata.dump

The following is an example of how a control file can be structured, if you use SQL Loader:

```
LOAD DATA
INFILE 'metricsdata.dump'
BADFILE 'bad_metricsdata.dump'
DISCARDFILE 'discard_metricsdata.dump'
APPEND
INTO TABLE METRICSDATA
FIELDS TERMINATED by ','
TRAILING NULLCOLS
{
  METRICVALUE float EXTERNAL NULLIF (METRICVALUE=BLANKS)
  , TIMEKEY INTEGER EXTERNAL NULLIF (TIMEKEY=BLANKS)
  , EID INTEGER EXTERNAL NULLIF (EID=BLANKS)
  , MID INTEGER EXTERNAL NULLIF (MID=BLANKS)
  , IID INTEGER EXTERNAL NULLIF (IID=BLANKS)
}
```

After loading data on the database, ensure that the process completed successfully, and manually delete the files contained in the table_loader directory. The following is a simple script you can use to do this:

```sh
#!/bin/sh
cat metricsdata.list | while read file ;
do rm $file $file.md.itmdmp ;
done ;
```

**Improving Resource Model responsiveness**

You can monitor the status of the engine, and of the resource models that have been distributed to it, by using one of the following:
**wdmlseng**
A command that returns a text stream with a detailed description (status and health) of every resource model that runs in the engine.

**Heartbeat**
A process that keeps track of the status of the engine and of the resource models that it contains.

**Web Health Console**
That displays the same information as the `wdmlseng` command, but in a web based GUI.

A new state of the resource models has been introduced to provide a time indication to the data collection:

**Delaying(T)**
This state means that a resource model is in delay, and T is the delay, expressed in seconds.

If a resource model does not provide fresh data for more than a specified tolerance period, the output of the `wdmlseng` and Web Health Console show the name of the delaying resource model and the number of seconds of delay. Also the Heartbeat process shows that the resource model is in error, and specifies the seconds of delay. This new status does not necessarily mean that the resource model is not running, but only that it is taking more time than expected to complete its operations.

You can set a tolerance period after which the resource model is considered to be Delaying. To configure this value, edit the following parameter of the engine configuration file (see “wdmepconfig” on page 44 for more details):

**DelayTolerance**
Factor used to calculate the maximum number of seconds the engine tolerates before changing the resource model state from **Running** to **Delaying**. The default value is 5. The value you specify here must be a decimal number that is then combined with the cycle time, using the following formula:

```
Maximum Delay = CycleTime + (DelayTolerance x CycleTime)
```

So, for example:
- DelayTolerance set to 0: the engine does not tolerate any delay.
- DelayTolerance set to 0.5: the engine tolerates a cycle time increase of 50%.
- DelayTolerance set to 1.5: the engine tolerates a cycle time increase of 150%.
- DelayTolerance set to 5: the engine tolerates a cycle time increase of 500%.

**Collecting raw data information**
Starting from Tivoli Monitoring 5.1.2 Fix Pack 8, you can have the information collected by the resource model locally available on the endpoint, saved in text files with XML format. You can activate this feature by enabling the corresponding parameters of the `wdmepconfig` command.

If you set the LogRawDataToFile parameter to true, the Tivoli Monitoring engine stores raw data information in a buffer cache and then writes the cache content on a text file. Every five minutes (default value that can be configured using the
RawDataFileGenerationTime parameter) the engine creates a new text file, where it writes the updated content. Then, periodically, all the raw data files older than a specified time (which you can set using the RawDataFilePurgeTime parameter) are deleted from the endpoint.

Moreover, setting the DisableDatabaseLogging parameter to true, you can disable the raw data logging to the IBM Tivoli Monitoring database on the endpoint, so that the database is not populated with this information.

If both the DisableDatabaseLogging and LogRawDataToFile parameters are set to true, the engine stores on the endpoint only the raw data and does not create any XML files.

For more information about all the parameters to configure this new feature, see "wdmepconfig" on page 44.

Text file location and format: The file name is:

ITM511_RAW@YEAR#MONTH#DAY#HOUR#MINUTE.txt

The file has an entry for each item of raw data collected, as shown in the following example:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<RawDataValues hostName="host1" ipAddress="9.48.67.88" epLabel="ep_1"
   gmtOffset="offset">
   <RawData>
   <sampletimestamp time="1135964698"/>
   <profile name="test_profile#my-region"/>
   <category name="Windows"/>
   <rm name="TMW_Processor"/>
   <context name="Availability"/>
   <rsc name="RealSystem"/>
   <instance name="Name=System"/>
   <sAttr name="Name">System</sAttr>
   <nAttr name="SystemUpTime">22012.0000</nAttr>
   </RawData>
   <RawData>
   <sampletimestamp time="1135964698"/>
   <profile name="rawprof2#host1-region"/>
   <category name="Windows"/>
   <rm name="TMW_Processor"/>
   <context name="Interrupt Time"/>
   <rsc name="Processor"/>
   <instance name="Processor=0"/>
   <nAttr name="Processor">0</nAttr>
   <nAttr name="InterruptsSec"> 113.0000</nAttr>
   </RawData>
   .......
</RawDataValues>
```

The file location is the following:

For UNIX platforms

$LCF_DATDIR/LCFNEW/Tmw2k/Unix/data/raw

For Windows platforms

$LCF_DATDIR/LCFNEW/Tmw2k/raw

About commands

This section describes commands that have been added or changed since the Tivoli Monitoring 5.1.2 release.
**wdmepconfig**: This command updates the configuration parameters of the Tivoli Monitoring engine. Retrieves or modifies the values of any of the configuration keys, setting the values you specify or resetting the defaults. The new values become effective after the engine is rebooted. The command operates on one or more specified endpoints.

**Usage:**

```
wdmepconfig [-e {endpoint |@endpoint_list_file}] [-D key=value [-D key=value]...] [-G key] [-R key]
```

**Parameters:**

- **-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 and value pair must be specified. Do not use blanks around the equal character (=) or, if you need to use blanks, enclose the key and value pair within quotation marks. You can only modify the following parameters:

  - **engine_tec_conf_persistence**
    Default value is false. If you set it to true you can make persistent any change in the Tivoli Enterprise Console configuration file. In this way it will not be overwritten at the engine restart or during a profile distribution. Only the ServerLocation and ServerPort configuration keys are allowed to be changed.

  - **DataSeeding**
    Enables the Tivoli Monitoring 5.1.x engine to send monitoring data also to Tivoli Monitoring 6.1. Possible values are:
    - **ITM5** Interoperability is not enabled. The Tivoli Monitoring 5.1.2 engine does not send monitoring data (either real time or historical data) to the Tivoli Monitoring 6.1 management environment.
    - **ITM6** Interoperability is enabled. The Tivoli Monitoring 5.1.2 engine sends monitoring data (both real time and historical data) to the Tivoli Monitoring 6.1 management environment. Historical data is no longer loaded into the Tivoli Monitoring 5.1.2 engine local database.
    - **BOTH** Interoperability is enabled. The Tivoli Monitoring 5.1.2 engine sends monitoring data (both real time and historical data) to the Tivoli Monitoring 6.1 management environment and historical data is also loaded into the Tivoli Monitoring 5.1.2 engine local database.

    The default value is ITM5.

- **db.removeOnCorruption**
  Default value is False. If you set it to True, at engine startup the system automatically detects and removes the endpoint database if it finds it corrupted, and writes a message in the msg_dmxengine.log trace file.

- **DelayTolerance**
  Factor used to calculate the maximum number of seconds the engine tolerates before changing the resource model status from Running to Delaying. The default value is 5. The value you specify here must be a decimal number that is then combined with the cycle time, using the following formula:
  
  Maximum Delay = CycleTime + (DelayTolerance x CycleTime)
So, for example:
- DelayTolerance set to 0: the engine does not tolerate any delay.
- DelayTolerance set to 0.5: the engine tolerates a cycle time increased by 50%.
- DelayTolerance set to 1.5: the engine tolerates a cycle time increased by 150%.
- DelayTolerance set to 5: the engine tolerates a cycle time increased by 500%.

**DisableDatabaseLogging**
Default value is false. If you set it to true, and the LogRawDataToFile is enabled, the engine stores on the endpoint only the raw data and does not create any XML files for warehouse.

**eif_BufferFlushRate**
Sets and changes the Tivoli Enterprise Integration Facility (EIF library) BufferFlushRate parameter for the UNIX Tivoli Monitoring engine. The default value is 0, however, it is suggested to set it to 10. For more information, refer to Tivoli Enterprise Console documentation.

**eif_MaxPacketSize**
Sets and changes the Tivoli Enterprise Integration Facility (EIF library) MaxPacketSize parameter for the UNIX Tivoli Monitoring engine. The default value is 0, and it is suggested to leave the default. For more information, refer to Tivoli Enterprise Integration Facility (Tec EIF) documentation.

**engine_ulimitdatasize**
Supported on AIX platforms, you use it to customize the process data size for the engine process. It sets the data seg size parameter as it would be set by launching the ulimit -d command. Do not change the default value unless absolutely necessary.

**engine_LDR_CNTRL**
Supported on AIX platforms, you use it to customize the process data size for the engine process. It sets the MAXDATA parameter. Do not change the default value unless absolutely necessary.

**EnumInstWMITimeOut**
Supported only on Windows platforms. It is the period of time after which a data collection request sent to WMI service expires. It is expressed in milliseconds and the default value is 600000. Do not change the default value unless absolutely necessary.

**HeartbeatEnabled**
When set to false, it excludes the endpoint from the heartbeat process. The default value is True.

**java_endorsed_dir**
Supported only on Windows platforms, you use it to set the endorsed directory for the Java used by some Tivoli Monitoring applications. The default value is false. For more information, refer to the documentation of the relevant Tivoli Monitoring applications.

**java_GCOPTS**
Sets and changes specific Garbage Collector java options for the Tivoli Monitoring UNIX engine. For more information about using this parameter, refer to Java documentation.
java_GENERICOPTS
Sets and changes generic Java options for the Tivoli Monitoring UNIX engine, such as the Garbage Collector options, as shown in the following example:

```
wdmepconfig -e endpoint -Djava_GENERICOPTS=
"-verbose:gc -XX:+PrintGCDetails
-XX:+PrintGCTimeStamps"
```

For more information about using this parameter, refer to Java documentation.

java_INITIALHEAPSIZE
Supported only on UNIX platforms. It is the initial size of the heap memory of the itm java process. The default value is 32 MB. Do not change the default value if not absolutely necessary. Setting this parameter too high or too low, can prevent the itm process from working properly. If you change the heap memory size, you have to restart the engine for the change to take effect.

java_LD_ASSUME_KERNEL
Supported only on Linux endpoints. Use this parameter if a different version of Linux kernel needs to be used for the Tivoli Monitoring java engine.

java_MAXHEAPSIZE
Supported only on UNIX platforms. It is the maximum size of the heap memory of the itm java process. The default value is 256 MB. Do not change the default value if not absolutely necessary. Setting this parameter too high or too low, can prevent the itm process from working properly. If you change the heap memory size, you have to restart the engine for the change to take effect.

LogRawDataToFile
Default is false. If you set it to true, and the profile is enabled to log raw data on the endpoint the engine writes data on a text file.

RawDataFileGenerationTime
Defines the time interval (in minutes) after which a new raw data file is created. The default value is 5 minutes. The minimum value is 1.

RawDataFilePurgeTime
Defines the time period (in minutes) after which raw data files are removed from the endpoint. The default value is 300 minutes. This means that all raw data files are deleted 5 hours after their generation. The minimum value is 1.

StatePersistence
If set to true, it enables the engine to keep track of the events sent before stopping. The default value is false.

StatePersistenceTime
Active only when the StatePersistence parameter is enabled. It specifies the time interval (expressed in minutes) after which the engine takes a new picture of the event situation and replaces the older one (if present). The default is 0, which means that the engine does not take any periodical snapshot of the event situation.

StatePersistenceValidityTime
Active only when the StatePersistence parameter is enabled. It specifies the maximum period of time (expressed in hours) before the picture of the event situation, taken by the engine, expires. If the engine restarts
after this period has elapsed, it does not take the stored event situation into account. The default is 0, which means that no time check is performed on the old event situation.

\texttt{-e \{endpoint |@endpoint_list_file\}}

Specifies the name of the endpoint or the file that contains a list of the endpoints (one name per line) for which you want to change the value of the specified configuration keys. The @ symbol indicates that a filename follows.

\texttt{-G key}

Identifies a configuration key. The argument is a key or a pattern for a group of keys. The wildcard character (*) is accepted in the pattern. Use the wildcard to show all the configuration keys for the specified endpoint.

\texttt{-R key}

Resets the default value for the specified key. The argument must be a key.

Authorization: Requires the super or senior roles.

Examples:

1. The following example sets the DelayTolerance key of dmhp3_hp10 endpoint to 1. This means that, assuming that the cycle time of this resource model is 50 seconds, the resource model is delaying if it does not provide fresh data for more than 100 seconds, according to the formula:

\[
\text{Maximum Delay} = \text{CycleTime} + (\text{DelayTolerance} \times \text{CycleTime})
\]

\texttt{wdmepconfig -e dmhp3_hp10 -D DelayTolerance=1}

2. The following example resets the DelayTolerance key of dmhp3_hp11 endpoint to the default value.

\texttt{wdmepconfig -e dmhp3_hp11 -R DelayTolerance}
wdmlseng:

Starting from Tivoli Monitoring 5.1.2 Fix Pack 9, this command has the following new optional parameter: -dump [-p|-t|-pt].

Usage: The usage of the command now is:

wdmlseng -e endpoint -p profile#region [-dump [-p|-t|-pt]]

Note: For more details about the usage of this command, see: IBM Tivoli Monitoring: User’s Guide.

The new dump option provides information, in XML format, about the resource models running on the endpoint for the specified profile. The new option can be specified only against a single profile.

The parameters are used as follows:

Parameters:

dump A dump of some information and settings about the resource models running on the endpoint. In particular, it provides information about indications, thresholds, parameters, and some logging information.

p A partial dump providing information only about the parameters of the resource models running on the endpoint.

pt A partial dump providing information only about thresholds and parameters of the resource models running on the endpoint.

t A partial dump providing information only about the thresholds of the resource models running on the endpoint.

Starting from Tivoli Monitoring 5.1.2 Fix Pack 16, the wdmlseng command has the following new optional parameter: -list. The new list option shows on the screen a list of all the profiles on the endpoint, both Loaded (distributed on the endpoint and activated) or Not Loaded (distributed on the endpoint but not activated), each with the list of its Resource Models.

The command usage has changed as follows:

- On TMR:
  wdmlseng -e <endpoint> [-list]

- On Endpoint:
  wdmlseng [-list]

You should get the following results when the -list wdmlseng option is used:

wdmlseng -e <endpoint> -list
testprofile_1#test-region : Loaded
  ResourceModel_1
  ResourceModel_2
  ...
  ResourceModel_n
testprofile_2#test-region : Not Loaded
  ResourceModel_1
  ResourceModel_2
  ...
  ResourceModel_n
**wdmngcache**: In addition to providing the endpoint label, and the heartbeat and Tivoli Business Systems statuses of the endpoints in the cache, the --v argument of **wdmngcache** now reports also the time of the last status change, the previous status, the time stamp of the last successful downcall, and the result of the TCP connection test.

The --v argument must be used with the --l argument.

The following resource model statuses, reported in the output of the **wdmngcache -I** command, lead to a resource model error state in the heartbeat:

- Error
- Missed Prereq
- Not Compiled
- Reserved (number)
- Retrying (number)
- Failing (number)
- Failed (number)
- Unable to start (number)
- Recovering (number)
- Failed after recovery (number)
- Delaying (number)

Instead, the following statuses are not considered in error:

- Running
- StoppedDisabled
- Scheduled
Starting from Tivoli Monitoring 5.1.2 Fix Pack 15, APAR IZ17302, to avoid duplicated entries, a new option "-O" has been introduced to wdmdistrib command, and needs to be used when distributing the same object (such as a JRE) many times to the same endpoint. The new command usage is as following:


**Purpose**

- **-O**
  This option makes it possible to distribute the same object to the same depot multiple times avoiding duplicated objects.

**Note:** For more details about the usage of this command, see: *IBM Tivoli Monitoring: User’s Guide.*

Starting from Tivoli Monitoring 5.1.2 Fix Pack 11, this command has the following new options: -v, and -I. The new options are used as follows:

**Usage:**

wdmdistrib -J /JRE_SOURCE_PATH/ -v 14 -l -r -s subscribers_file

and:

wdmdistrib -J /JRE_SOURCE_PATH/ -l w32-ix86,linux-ppc -l -r -s subscribers_file

**Purpose:**

**-v**
Allows you to specify the JRE version that must be distributed to the endpoints. Possible values for this option are 13 (for JRE 1.3), 14 (for JRE 1.4), and 15 (for JRE 1.5). To be distributed, JRE 14, must be in the following form: jre14.tar.gz. Which means that you must produce first the tar and then the .zip of the JRE to distribute it.

**-I**
Allows you to specify the interps where you want to distribute the JRE. Using this option, it is no longer necessary to have all the interp directories under /JRE_SOURCE_PATH/ directory. Possible values for this option are: w32-ix86, solaris2, aix4-r1, linux-ix86, linux-s390, hpux10, solaris2-ix86, linux-ppc.
wdmrm:

Starting from Tivoli Monitoring 5.1.2 Fix Pack 14, the following new options have been added to this command to improve profile handling and serviceability:

- **-report**
  Creates a directory under $DBDIR/AMW/logs/wdmrm_logs/
  resourcemodel_date_time path, in which to store information about the resource model specified in the wdmrm command.

- **-trace**
  Creates, under $DBDIR/AMW/logs/wdmrm_logs/resourcemodel_date_time path, a verbose trace file named wdmrmfilenumber.log, which contains all tracing information. If you select this option, also specify the trace level (values from 0 to 2) and, optionally the trace size.

- **-size**
  Allows you to specify the file size of your trace. Available values are from 500000 to 5000000 bytes. Default value is: 500000 bytes.

Using these options, you can keep track of possible errors, if, for example, a profile update fails or if there are corrupted profiles. The following is an example of the files created using the new options:

**wdmrm0.log**
It is the trace file.

**resource_model_name_corruptedProfiles.txt**
It is the file containing the profiles that cannot be updated because they are corrupted.

**resource_model_name_failProfiles.txt**
It is the file containing the profiles that cannot be updated for other reasons.

**resource_model_name_noUpdateRequiredProfiles.txt**
It is the file containing the profiles that do not need to be updated.

**resource_model_name_successUpdatedProfiles.txt**
It is the file containing the profiles that have been successfully updated.

Using these options, it is possible to understand whether or not a profile was correctly updated with the new resource model. The command usage has changed as follows:

```
wdmrm --add resource_model_tarfile [-report] [-trace trace_level] [-size trace_size]
wdmrm --addcat resource_model [-f catalog_file -l locale] [-trace trace_level] [-size trace_size]
wdmrm --remove resource_model [-trace trace_level] [-size trace_size]
wdmrm -list
```

**Commands launched from endpoints:** Starting from Tivoli Monitoring 5.1.2 Fix Pack 9, from UNIX endpoints you can run the following commands:

- wdmcmd
- wdmlseng

These two commands allow you to start, stop, and query the engine locally from an endpoint, just as you could do from the TMR server.

They follow the same rules and usage as the wdmcmd and wdmlseng commands launched from the server, except for the following:
Before launching any of these commands from an endpoint, you must import all the required environment variables, launching the `itm_c1i_set_env.sh` scripts from the following path: `$LCF_DATDIR/LCFNEW/Tmw2k/Unix/bin`.

**wdmlseng**
When launching this command from an endpoint, skip the `-e` option. This option is used only from the server, to specify the endpoints on which you want to run the commands.

**wdmcmd**
When launching this command from an endpoint, use only one of the two mandatory parameters (`restart` or `stop`) skipping all the optional ones.

**Note:** For more details about the rules and usage of these commands, see: *IBM Tivoli Monitoring: User's Guide*.

**Changes to the endpoint logs and traces**
The maximum and default file size for the endpoint logs and traces is changed as follows:

- On Windows endpoints the default file size is 500 KB, while the maximum file size is 100 MB
- On UNIX endpoints the default file size is 5000 KB, while the maximum file size is 500 MB

**About ETLs**
The following sections contain information about AMY and AMX processes that are not included in the official documentation released with Tivoli Monitoring 5.1.2.

Some of the information is not new but was previously documented in the Readme file. It has now been removed from there and included in this document.

**AMX extraction process:** Starting with Tivoli Monitoring 5.1.2 Fix Pack 1, the RIMHandler process also updates the CONTROLTABLE, created on the IBM Tivoli Monitoring Database, inserting the last value reached by the INSERTSEQ field, used to enumerate the records inserted in the METRICSDATA and CATEGORIESDATA tables. This allows the AMX extraction process to skip the computation of such values to determine the proper boundaries of the extraction window. Before running the new AMX extraction process, complete the installation of IBM Tivoli Monitoring Tivoli Enterprise Data Warehouse Support, 5.1.2 Fix Pack 3.

To avoid problems related to transaction log growth, the extraction window has been divided into multiple smaller sub-windows, each belonging to a different transaction. The default maximum extension of a single transaction is 10000 records, but you can change this value if it is not appropriate for your environment. You can increase or reduce the commit period parameter (COMMITPERIOD) so that a commit is performed only after the extraction of the specified number of records. The commit period parameter must be manually updated in the `amx_rim_extract.sh` wrapper script located in the `%TWH_TOPDIR%/tools/bin` directory.

**AMX prune process:** The pruning activity removes data on a timely basis. However, starting with Tivoli Monitoring 5.1.2 Fix Pack 1, you can indirectly control the number of records to remove during each transaction by defining the temporal extension (PRUNE_PERIOD) of each slice of data to be pruned during each transaction.
For example: each hour an average of 10,000 records is inserted into the Tivoli Monitoring Database, if you set the PRUNE_PERIOD=3600 (1 hour), an average of 10,000 records will be pruned from the Tivoli Monitoring Database during each pruning transaction.

To define the PRUNE_PERIOD parameter (whose default value is 3600 seconds), update the AMX.PRUNE_CONTROL table defined in the TWH_CDW database:

```
UPDATE AMX.PRUNE_CONTROL SET PRUNE_PERIOD = number_of_seconds
```

**Cleanup scripts:** Tivoli Monitoring 5.1.2 Fix Pack 1 introduces maintenance scripts to perform the cleanup of the Tivoli Monitoring database. This maintenance activity is not critical for the functions of the Data Collection process, but it should be performed on a regular basis, especially when some resource models are no longer distributed. To launch the cleanup script, go to the `$BINDIR/TME/Tmw2k/WarehouseCfg/cleanup_db` directory and run the `cleanup_itm_db.sh` wrapper script from a Tivoli environment.

**Note:** If you are using DB2, define the DB2 environment before exporting the Tivoli environment and running the `cleanup_itm_db.sh` wrapper script. Only for DB2, the `cleanup_itm_db.sh` script automatically invokes the update statistics administration command before and after the pruning.

**Changes to AMY and star schemas:** With Tivoli Monitoring 5.1.2 Fix Pack 5, the AMY Memory star schemas have been modified. The AMY.D_CONFIG dimension table has been added to include information about physical memory. Data is collected by the TMW_Memory resource model, under the new category `TotalPhysicalMemorySize`. This change is documented in the ITM_OS_TEDW.pdf provided with AMY.

**Manual data extraction:** The following procedure is to be considered as a recovery action in case of failure in the data extraction normal process. If you have a very large amount of data that must be extracted from the central data warehouse and loaded into the data mart, the whole process might take a very long time to complete or it might fail due to the transaction log being full. As a recovery action from this situation, you can use the provided scripts to update and extract a limited amount of data at a time, and repeat the procedure more than once, for an incremental extraction of all the required records into the data mart, as described in the following procedure:

1. Open a command window on the central data warehouse system only (remote database client usage is not supported) and change directories to:
   ```
   %TWH_TOPDIR%/apps/am\Version\misc\tools
   ```
2. Type `bash` and press Enter to open a bash shell.
3. From the bash$ prompt, enter the following command:
   ```
   ./AMY_set_extract_control.sh twh_cdw DB2user DB2password Number_of_records
   ```
4. Check the log file to ensure that the script completed successfully. The log file is named `amy_set_extract_control.log` and is located in the DB2 installation directory, under the SQLLIB\logging directory. If the script runs successfully, you will find a corresponding message in the log.
5. If the script completed successfully, run the ETL2 as usual for the records that need to be transferred into the data mart.
6. Repeat this procedure as many times as required to incrementally transfer all the necessary records.
7. When you have transferred all the required records, run AMY a final time so that the extract control table correctly reflects the sequence values for each of the components.

The update of the extract control table together with the updated SQL script amy_m05_s020_mart_extract.db2 process only the number of records indicated by the command line argument passed to ./AMY_set_extract_control.sh. It is important to note that the number of records to process is limited to AMY measurement table entries only.

To determine how many times you need to repeat the whole process, perform the following procedure:

1. From the central data warehouse, run the following statement to retrieve the number of records in the measurement table:
   
   ```sql
   select max(msmt_id) from twg.msmt;
   ```

2. Run the following statement to retrieve the current record count inserted, for example, into the data mart for AMY.STG_CPU_HR:

   ```sql
   select extctl_from_intseq from twg.extract_control
   where extctl_source= 'TWG.MSMT'
   and extctl_target = 'AMY.STG_CPU_HR';
   ```

3. Subtract the results of the second statement from the first, to calculate the number of records that need to be processed by the data mart.

**ETL2 process additional step:** Starting from Tivoli Monitoring 5.1.2 Fix Pack 7, AMY_m05_s035_mart_post_load step has been added to perform the extract_control logging after the data mart extract and load has completed successfully. This ensures that the data has been completely populated into the mart database before the log is updated. This allows recovery of records from the central data warehouse, should any of the steps prior to the mart_post_load step fail.

**About State Persistence**

Starting from IBM Tivoli Monitoring 5.1.2 Fix Pack 9, you can configure the engine so that, when restarted, it remembers the events it had sent before stopping.

As a result, when the engine restarts, if it finds that a previously sent event is still present, it does not send it again. On the contrary, if it finds that a previously sent event is no longer present, it sends a clearing event (if enabled).

If you enable the persistence feature, when you stop the engine, it takes a picture of the open events just before stopping, and uses it as a later reference when it restarts.

The picture of the event situation is saved in the aggreg_state.dmp binary file, located, respectively:

- On Windows under: LCFNEW/Tmw2k/bin/status directory
- On UNIX and AS/400 under: LCFNEW/Tmw2k/Unix/data directory.

**Note:** This feature enables the engine to remember the sent events only. It does not keep any track of the monitoring information it has collected before stopping.

When the engine restarts, the event information is reloaded and the file is deleted.

To enable the engine state persistence feature, specify the following key with `wdmepconfig -D` command:
**StatePersistence**

If set to true, it enables the engine to keep track of the events sent before stopping. The default value is false.

However, this feature works only if the engine is regularly stopped and restarted using the `wdcmd -restart` command. It is not activated by performing a profile distribution when the engine is off. To avoid the risk of losing the event information if the engine crashes, you can configure the engine to take a periodical picture of the event situation, and use it when restarting after crash. To configure the periodical picture feature, specify the following key with `wdmepconfig -D` command:

**StatePersistenceTime**

It specifies the time interval (expressed in minutes) after which the engine takes a new picture of the event situation and replaces the older one (if present). The default is 0, which means that the engine does not take any periodical snapshot of the event situation.

If the period of time that elapses between when the engines stop and restarts is too long, the event situation picture that the engine took might be too old, and no longer reliable. In this case, it might be better that the engine restarts without taking it into account, just as if it was started for the first time. Therefore, you can specify a period of time, after which the event situation picture is no longer valid, because it is out of date. To specify the validity period of the picture, use the following key with `wdmepconfig -D` command:

**StatePersistenceValidityTime**

It specifies the maximum period of time (expressed in hours) before the picture of the event situation, taken by the engine, expires. If the engine restarts after this period has elapsed, it does not take the stored event situation into account. The default is 0, which means that no check is performed about how old the event situation is.

**New messages**

Starting from Tivoli Monitoring 5.1.2 Fix Pack 9, on non-Windows endpoint, the following new messages can be written on the `aggreg_state.dmp` file, as a result of the State Persistence feature enablement:

- **AMW2047I**: The engine state has been saved.
- **AMW2048I**: The `aggreg_state.dmp` file has been removed and the engine state has not been reloaded.
- **AMW2049I**: Loading engine state from file `aggreg_state.dmp`.
- **AMW2350I**: The profile dump for `{profile}` has been rewritten.
- **AMW2351E**: Error while restarting the profile `{profile_dump}`.

**Explanation**

The file containing the profile dump is not readable and therefore no profile settings will be reloaded.

**Operator Response**

Perform a profile distribution to overwrite the corrupted dump file, or delete the file from the endpoint.

**About UNIX engine**

On AIX platforms, if you modify the path variable in the `.cshrc` file, ensure you write `$PATH` at the end of the modified path. This is necessary to avoid the new path overwriting the path variable set and used by the Tivoli Monitoring engine.
About Linux platforms
Starting from Tivoli Monitoring 5.1.2 Fix Pack 7 the Single Port Bulk Data Transfer feature provided by Tivoli Management Framework is enabled on Linux platforms. From this fix pack, Tivoli Management Framework 4.1 or higher is required as a prerequisite on Linux TMR servers and managed nodes.

About Tivoli Mobile Console on Windows endpoints
On Windows endpoints with the Tivoli Mobile Console installed, each time a profile is distributed, or a command using the MDist 2 service is run, a pop-up window appears asking to confirm the distribution. Starting from Tivoli Monitoring 5.1.2 Fix Pack 7, you can enable the automatic distribution, preventing the Tivoli Mobile Console from displaying pop-up confirmation windows. To enable the automatic distribution, set the distribution.is_hidden key of wdmconfig command to true.

How Tivoli Monitoring is affected by recent changes in Daylight Saving Time
A provision of the United States Energy Policy Act of 2005 extends Daylight Saving Time by four weeks, beginning in 2007. Several countries (Canada, Bermuda) are also implementing the same DST change; other countries may decide to make similar changes to DST rules. Beginning in 2007, DST start and end dates will be:
• Start date: Second Sunday of March (11 March 2007)
• End date: First Sunday of November (4 November 2007)
For further information on the new DST policy, please refer to the main IBM DST site:
http://www.ibm.com/support/alerts/daylightsavingstimealert.html
Tivoli Monitoring is not affected by this change, provided that Java and the operating system contain the required patches. For information, you can refer to the Tivoli Monitoring web page:
Starting from Tivoli Monitoring Fix Pack 11 the JREs distributed with the product already contain the patches necessary for the new DST rules for all platforms, except for Windows..
The JRE 131 SR8 for Windows with the DST enablement is provided with the 5.1.2-ITM-0183LA patch.

Integration between Tivoli Monitoring 5.1.x and Tivoli Monitoring 6.1
Starting from Tivoli Monitoring 5.1.2 Fix Pack 6, you can send data asynchronously, from Tivoli Monitoring 5.1.x to Tivoli Monitoring 6.1 and insert it directly into Tivoli Data Warehouse 2.1. The monitoring engine sends data to a new integration agent, and this data can then be inserted into the Tivoli Data Warehouse 2.1 directly, without previous aggregation.
To enable the integration, ensure that you have installed on the Tivoli Monitoring 5.1.x TMR server, the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint supplied with the Tivoli Monitoring 6.1 CD, and that you have distributed it on all
To enable this feature, configure the endpoint monitoring engine, setting the 
**DataSeeding** option of the **wdmepconfig** command as required. For more details about these option, see: [“wdmepconfig” on page 44](#).

### Documentation problems and corrections

This section documents known documentation defects for this release of Tivoli Monitoring. The following list provides the revised information for each documentation problem:

1. On page 15 of the *Tivoli Decision Support for Server Performance Prediction: Release Notes*, within section "Remove the TDS Configuration Database Structure", step 3 needs to be modified as follows:

   **Note:** Starting with Tivoli Monitoring 5.1, the TDS Configuration Component is now called Gathering Historical Data Component.

   "If the database server or client on which you need to run the procedure does not have TDS Configuration installed on it, you should proceed as follows, depending on the operating system of the server or client:

   - **UNIX:** Mount the `$BINDIR/Tmw2k/TDS/rdbcfg` directory of any system with TDS Configuration installed (for example, the Tivoli server) as an NFS mount on the database server or client, and run the procedure described in step 2 on page 15 (applied to `rm_rollup_db.sh` script) from within that directory.

   - **Windows:** follow these steps on a system where Tivoli is installed (or cut and paste the content of file `rm_db.<database_extension>`, see below, and use it from an SQL session of the database):
     a. Copy the following files from the `$BINDIR/TME/Tmw2k/TDS/rdbcfg` directory to any system where the TDS Configuration has been installed:
        - `rm_rollup_db.sh`
        - `rm_db.<database_extension>`, where `database_extension` is one of the following, depending on your database vendor:
          - Db2: DB2®
          - Inf: Informix®
          - Mssql: Microsoft®-SQL 6.x
          - Mssql7: Microsoft-SQL 7.0
          - Ora: Oracle
          - Syb: Sybase
     b. Run the `rm_rollup_db.sh` script to remove the database structure.

2. On page 4 of the *Tivoli Decision Support for Server Performance Prediction: Release Notes*, within section "Supported Databases", the following information must be changed:

   - DB2 Version 7.2 is now supported instead of DB2 Version 6.1
   - Oracle Version 8.1.6 is now supported instead of Oracle Version 8.0.5

3. On page 17 and 18 of the *Tivoli Decision Support for Server Performance Prediction: Release Notes*, within section "Normal Running", the following information must be changed:
• In the second bullet, the words "Each day, at 1:30 a.m." must change into "Each day, at 1:00 a.m."
• In the third bullet, the words "At 2:30 a.m., a second job is run" must change into "At 2:00 a.m., a second job is run"

4. On page 5 of the *Tivoli Decision Support for Server Performance Prediction: Release Notes*, within section "Install the TDS Configuration Component", the following changes apply:
   • The Install Options dialog (and screen capture) no longer contains a Database User Password field
   • The Install Options dialog (and screen capture) now shows the default value dm_db for field Database ID
   • The information about Database User Password (in Table 1 - TDS Configuration Install Options) must be extracted from the table and placed within a Note.
   • Under bullet number 4, the first unordered bullet before Table 1 now says:
     - If your RDBMS server or client is installed on the Tivoli management region server, you should fill out the fields as indicated in Table 1, and click Set. If you click Set without filling out the install options, the Rim object is created with the default parameters (for DB2).
   • Under bullet number 4, the second unordered bullet right after Table 1 now says:
     - If neither your RDBMS server nor client is installed on the Tivoli management region server, you should leave all fields empty, click Close, and set up the Rim object as described in "Creating the RIM Object Using a Shell Script" on page 56.
   • Under bullet number 4, a third unordered bullet must be added to say:
     - If you leave all fields blank and click Close, then the Rim object is not created.

5. On page 9 of the *Tivoli Decision Support for Server Performance Prediction: Release Notes*, chapter 2. Installation and Customization, within section "Step 4. Creating the Tivoli DM Database", at the end of the second bullet the following new paragraph is added:
   To run the `cr_db.db2` SQL script, use the command:
   ```
db2 -vf cr_db.db2.sql
   ```

6. On page 169 of the *IBM Tivoli Monitoring: User's Guide*, Chapter 8. Commands, section “Return codes from commands”, the following changes apply to Table 16:
   • The symbolic code EX_CLI_USAGE_RC is only associated to return code -1.
   • Messages AMW0175E, AMW0176E, and AMW0177E are associated to the following new symbolic codes:

<table>
<thead>
<tr>
<th>Symbolic code</th>
<th>Message identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>EX_TASK_MISSING_EVENT_NAME</td>
<td>AMW0175E</td>
</tr>
<tr>
<td>EX_TASK_MISSING_LIB_NAME</td>
<td>AMW0176E</td>
</tr>
<tr>
<td>EX_TASK_MISSING_NAME</td>
<td>AMW0177E</td>
</tr>
</tbody>
</table>

7. The HTML version of the *IBM Tivoli Monitoring: User's Guide* contains an incomplete table in Chapter 2. Installing, section "Supported platforms". The correct table is documented in these Release Notes, see "Supported platforms" on page 3 for details.
8. On page 19 of the *IBM Tivoli Monitoring: User's Guide*, Chapter 2. Installing, Table 8. Prerequisite software, the row that defines the prerequisites for the integration with Tivoli Data Warehouse is changed to state the following required software:

   Tivoli Enterprise Data Warehouse 1.1 Fix Pack 3 (plus latest available patch), or a later version, must be available.

9. On page 29 of the *IBM Tivoli Monitoring: Problem Determination Guide*, Chapter 2. Built-in serviceability, section "Serviceability tasks", the following additional file is collected on UNIX and Windows platforms by the DMCollectMNLog task:

   $DBDIR/TMP/mn_time_info.log

10. On page 69 of the *IBM Tivoli Monitoring: User's Guide*, Chapter 4. Heartbeat function, section "Installing and configuring the heartbeat", the description for configuration key `heartbeat.send_events_to_notice` is revised as follows:

    `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.

11. On page 69 of the *IBM Tivoli Monitoring: User's Guide*, Chapter 4. Heartbeat function, section "Installing and configuring the heartbeat", in the example for the configuration key `heartbeat.tec_server`, replace @ with #, as follows:

    EventServer or EventServer#regionname

12. On page 162 of the *IBM Tivoli Monitoring: User's Guide*, Chapter 8. Commands, command `wdmrm`, the description for the `–add` option is changed in the following way:

    `–add` Adds the resource model in the specified `resource_model_tarfile` to the Tivoli server. If the resource model already exists, to have it replaced by the new version you must:
    • stop the endpoint engine and redistribute the profile.
    • or run the command `wdmcmdistrib` with the `–upgrade` option.

13. On page 162 of the *IBM Tivoli Monitoring: User's Guide*, Chapter 8. Commands, command `wdmrm`, two options have been added, see : "`wdmrm` on page 51".

14. On page 110 and 162 of the *IBM Tivoli Monitoring: User's Guide*, Chapter 8. Commands, command `wdmrm`, cannot be issued from any managed node and gateway as stated, but can only be issued from a Tivoli management region server (TMR).

15. On page 117 of the *IBM Tivoli Monitoring: User's Guide*, Chapter 8. Commands, command `wdmcmdistrib`, the description for option `–t` now includes the following:

    **Note**: If the timeout you specify is too small, the command might fail to complete some operations on several endpoints. Differences in the time settings of separate machines can increase the impact of a too small timeout.

16. On page 117 of the *IBM Tivoli Monitoring: User's Guide*, Chapter 8. Commands, command `wdmcmdistrib`, the description for option `–upgrade` has changed as follows:

    `–upgrade`
    Performs the following operations:
    • Stops all endpoint engines
    • Distributes the updated engine binaries
- Distributes the updated profiles to the endpoints
- Restarts all endpoint engines.

Use `-upgrade` with the `-t` option, to set a timeout for all the above operations (except for the distribution of discovered profiles to the endpoints, which uses the timeout set by the MDist 2 service.) The response time for all the operations is set using the `notify_interval` parameter of MDist 2 service. So, if you set `notify_interval` to three minutes, every three minutes you are notified of partial operation results (number of endpoints where operation succeeded, failed, and is still pending) until either the operation completes successfully, or timeout elapses.

17. On page 116 of the *IBM Tivoli Monitoring: User’s Guide*, Chapter 8. Commands, command `wdmcddistrib`, the list of files and directories affected by the `-upgrade` option includes also the following directories:
   - In the directory `InvalidProfilesDiscoveredAtEndpoints` 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 removed from the TMR server, but are still present on the endpoints.
   - In the directory `ProfilesDistributionFailed` the command saves a set of files associating each profile to the list of target endpoints. These are the profiles whose distribution failed, according to the contents of `updated_models_list.txt` file.

18. On page 116 of the *IBM Tivoli Monitoring: User’s Guide*, Chapter 8. Commands, command `wdmcddistrib`, under `-m` option description, the following information is missing: For this option to work properly, all the involved managed nodes must be aligned at the same date and time.

19. On page 131 of the *IBM Tivoli Monitoring: User’s Guide*, Chapter 8. Commands, command `wdmdistrib`, the description for option `-r` is now the following:

   `-r` Indicates that the JRE must be replaced on the target subscribers (and the link to the existing JRE removed). 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.

20. On page 121 of the *IBM Tivoli Monitoring: User’s Guide*, Chapter 8. Commands, command `wdmconfig`, `datacollector.db_purge_interval` is not between 10 - 60 days, as stated. Instead, it is from 2 to 60.

21. On page 121 of the *IBM Tivoli Monitoring: User’s Guide*, Chapter 8. Commands, command `wdmconfig`, `datacollector.delay` parameter is not between 10 - 60 minutes, as stated. Instead, it is from 2 to 60.

22. On page 122 of the *IBM Tivoli Monitoring: User’s Guide*, Chapter 8. Commands, command `wdmconfig`, `datacollector.sleep_time` parameter states that the time interval (in minutes) between two consecutive requests of data uploading can range from 10 to 60 minutes, and that the default is 10 minutes. Instead, the value can range from 1 to 60, and the default is 1 minute.

23. On page 122 of the *IBM Tivoli Monitoring: User’s Guide*, Chapter 8. Commands, command `wdmconfig`, `dmml.trace_level` parameter specifies that the level of trace (for all components) can be set from 0 (minimal) to 4 (verbose). Instead, the trace level can range from 0 to 2.

24. On pages 122 and 123 of the *IBM Tivoli Monitoring: User’s Guide*, Chapter 8. Commands, command `wdmconfig`, the following new `key-value` pairs have been added:
datacollector.insert_data
Set this to false, if you do not want to load collected data into the database. The default is true.

request_manager.verify_epcache
When the request manager starts, it checks that all the endpoints present in the endpoint cache are also present in the endpoint manager. Set this key to false, if you do not want the request manager to perform this check. The default is true.

distribution.is_hidden
Set this to true, if you want the distribution of profiles and commands that use the MDist 2 service to be performed automatically, without asking for your confirmation. The default value is false.

25. On page 119 of the IBM Tivoli Monitoring: User’s Guide, Chapter 8. Commands, command wdmcollect, has the following new option:

  wdmcollect -e [endpoint_name 1 @endpoint_list_file] -l [-m managednode]

You can use the -l option to collect data directly from the endpoints and store it on the Tivoli Monitoring RIM database. This option allows you to retrieve data more quickly, because it by-passes the request manager and processes the data request immediately without queueing it.

26. On page 119 of the IBM Tivoli Monitoring: User’s Guide, Chapter 8. Commands, the authorization levels required for command wdmcollect, are incorrectly stated. The correct authorization levels required are: super, senior or admin role.


If you are modifying an existing profile, you can specify always (or Always) as a value for the schedule option. If you do, all the defined rules are ignored and the profile schedule is set to always.

28. On page 148 of the IBM Tivoli Monitoring: User’s Guide, Chapter 8. Commands, command wdmheartbeat, the description for option -t is now the following:

  -t This option stops the heartbeat with immediate effect on the defined managed nodes and gateways. The heartbeat must be running for this command to take effect. Using this option, the heartbeat is stopped and the heartbeat process is terminated.

29. On page 162 of the IBM Tivoli Monitoring: User’s Guide, Chapter 8. Commands, command wdmrm, the description for option –add is incomplete, replace with the following:

  –add Adds the resource model in the specified resource_model_tarfile to the Tivoli server. If the resource model already exists, it is replaced by the new version, in which case, you must stop and restart the endpoint engine. To distribute the new resource model on the endpoint, run the wdmcmddistrib-upgrade command.

30. On page 163 of the IBM Tivoli Monitoring: User’s Guide, Chapter 8. Commands, command wdmrm, the minimum authorization role required to use the option –add is senior or super. (The administrator role, as incorrectly stated in the User’s Guide, is not enough)

31. On page 20 of the IBM Tivoli Monitoring: User’s Guide, Chapter 2.Installing, Table 8, the following new information must be imbedded in the row for endpoints running UNIX or Linux:
32. On page 48 of the IBM Tivoli Monitoring: Problem Determination Guide, Chapter 3. Problem resolution, at the bottom of Table 4. Problems and their workarounds, the following problem must be added:

   Problem: The Tivoli Monitoring engine does not start when a profile is distributed to an endpoint running RedHat Enterprise Linux 3.0 AS 64bit pSeries.

   Workaround: Install the compat-libstdc++-7.3-2.95.4.4m rpm package on the endpoint.

33. APAR IY63470 - On page 122 of the IBM Tivoli Resource Model Builder: User’s Guide, Appendix B, Service object method library, the description for the GetShellStdOut method states that the method is used to get the full standard output of a spawned process. Instead, it returns only the first 4 KB of standard output. The reason for this is to prevent the engine from having a performance decrease of CPU and memory due to the overhead of reading a big sized file. However, usually resource models are intended to manage small sized output returning from scripts or providers (such as return codes). If you need to manage an output longer than 4 KB, you should specify in the resource model that the output must be segmented in separate 4 KB pieces.

34. On page 129 of the IBM Tivoli Resource Model Builder: User’s Guide, Appendix B, Service object method library, DefineClassAsync the description for the method is missing the following information:

   After creating a new resource model that uses this class, to have the resource model working properly, you have to add manually the RepeatCount field in the SetDefaultConfiguration, as shown in the example:

   ```
   Svc.DefineClass ("CIM", "Win32_NTLogEvent",
   "root\cimv2:Win32_NTLogEvent", "WHERE(Logfile="Application" AND EventCode="1221")",
   "RepeatCount,RecordNumber,EventCode,EventIdentifier",
   "Logfile,Message,TimeWritten", "None", ",", 0, -1);
   ```

35. On page 122 of the IBM Tivoli Resource Model Builder: User’s Guide, Appendix B, Service object method library, Method Shell, the description of the method says incorrectly: "The launched process must return within 60 seconds, otherwise an error generates." Instead, it should state: "The process timeout depends on the engine operating system. For Windows engines (tmw2k.exe), the process has a two-minute timeout. For UNIX engines, there is no timeout.

36. APAR IY91663: On page 32 of the IBM Tivoli Monitoring: User’s Guide, Chapter 2, Installing Java Runtime Environments, the sentence that says: "You can use SIS to install JRE. The JRE product name in SIS is ITM 5.1.2 - JRE 1.3.0." is no longer correct. The JRE cannot be installed using SIS.

37. On page 58 of IBM Tivoli Monitoring: Resource Model Reference Guide, under the Logging section for DMXNetworkInterface resource model, the table shows BytesReceivedSec; BytesSentSec in the Properties column. These are not the correct property names (as defined in the raw data text file) The correct names are: InBytes and OutBytes.

38. On page 120 of IBM Tivoli Monitoring: Resource Model Reference Guide, in the Thresholds section of File System resource model, the Percentage of Used Space threshold description incorrectly say: "The threshold is the maximum percentage of used space in a file system." Instead, it should say: "The threshold is the minimum percentage of used space in a file system."

incomplete saying: "...all file systems of the supported types (UFS and VxFS)...". Instead, it should say: "...all file systems of the supported types (UFS, VxFS on Solaris, and GPFS on AIX)..."
Chapter 5. Internationalization notes

This section documents the specific requirements, notes, problems, defects, and corrections for the international versions of Tivoli Monitoring.

Software limitations

The following limitations applies:

- If you use Tivoli Monitoring Workbench in a Japanese environment for debugging, you have to change the font settings. The default font is set to MS Sans Serif (SBCS font); you need to set it to a DBCS font to read the information in the debugging folder properly.

- In a Windows environment, it is strongly recommended not to use DBCS characters in profile names or in policy names. For example, if you use DBCS characters in a profile name and distribute the profile, the distribution completes successfully. However, when you use the DMCollect MnLog serviceability task to gather log files, the log file for that profile is not collected.

Enabling language support

Localization has been supported since Tivoli Monitoring 5.1.1 Fix Pack 6. The following procedure was documented only in the readme file, and has now been added to the Release Notes:

To apply the language support from the Tivoli desktop, perform the following steps:

1. From the Desktop menu, select the Install -> Install Product, to display the Install Product dialog.
2. Click Select Media and the File Browser dialog opens.
3. In the Path Name field, type the path to the directory that contains the fix pack.
4. Click Set Media & Close to return to the Install Product dialog.

The product install list contains the names of the language support components to be installed: IBM Tivoli Monitoring, Version 5.1.1 fix pack 6 - Language Support

5. Select the IBM Tivoli Monitoring, Version 5.1.1.- Language Support
6. Select the language to be installed from the list.
7. Select the managed nodes on which to install the language support.
8. Click Install.

Each component can be installed using the command line with the command wpatch, as follows:

\texttt{winstall -c <CDROM-PATH> -i AMW_<LOCATE>.IND [NODE1 NODE2 ...]}

where: NODE1 NODE2 ... are the managed nodes where the language support is installed.

If no managed node is specified, language support will be installed on all managed nodes of the TMR.
Updating Web Health Console Files

Note: This feature is supported only on WebSphere Application Server 4.
To update the Web Health Console files, you can copy them manually, by doing as the following:

Copy the contents of the HCONSOLE directory of catalog files into the following directory on the target computer that hosts the Web Health Console:

- INSTALL_DIR\installedApps\dm.ear\dm.war\WEB-INF\classes\com\tivoli\DmForNt\resources
- INSTALL_DIR is the directory where the product is installed

Note: For UNIX systems, use / instead of \ in the above path name

To update or re-distribute the resource model files, issue the following command in the bash shell for the resource model you wish to add:

wdrrm -addcat resource_model_files

Alternatively, you can update the Web Health Console files by doing as the following:

From the IBM Tivoli Monitoring Language Support CD (/WHC Subdirectory), run one of the following provided batch files:

- LPinstall.bat
- LPinstall.sh

This installs all the IBM Tivoli Monitoring Web Health Console NLS files required on existing Web Health Console Server.

Software problems and workarounds

This section describes known defects which apply to the international versions of Tivoli Monitoring. Where applicable and known, suggested workarounds are identified.

1. CMVC 33449 - In a DBCS environment, when a profile name contains DBCS data, the endpoint log file named trace_dmxengine.log and the message log file named msg_dmxengine.log might show a corrupted profile name.
2. CMVC 33297 - In a DBCS environment, events containing DBCS data might show corrupted on the Tivoli Enterprise Console.
3. CMVC 34339 - In a DBCS non-English environment, on W2k Professional Server (Windows and UNIX) machines, the installation of the Gathering Historical Data Component fails.

Workaround: Before installing the Gathering Historical Data component on a DBCS W2k Professional Server machine, perform the following steps to define two environment variables (LANG and LC_ALL):

- On a Windows machine:
  b. Add two system variables for local Administrator: one is LANG, the other is LC_ALL, both are set to C.
  c. Restart the oserv through ‘service’.
- On a UNIX machine:
a. Open the Command Line Interface.
b. Set the LANG and LC_ALL system variables to C.
c. Restart the oserv by issuing the command `odadmin reexec all`.
d. Issue the command `wsetlang -o -l C`.

Now you can successfully install the Gathering Historical Data component.

After the installation completes, you must remove the two variables and restart
the oserv to go back to the normal DBCS non-English locale environment.

4. **CMVC 34884** - In a DBCS environment, distributing all OS resource models
with defaults, or any subset of them but including the `Parametric Event Log`
(TMW_ParamEventLog) with defaults, when a windows event containing
mixed english-chinese text occurs, like the one from the WFPS (source
'Windows File Protection' subsystem), then this causes the engine to lock on a
write operation. Every subsequent request arriving at the engine is enqueued,
causing the queue of instances of the tmw2k_ep processes. The engine is still
running, but it is not able to respond to them unless you use wdmcmd to stop or
restart the engine.

5. **APAR IY69172** - On AS/400 localized endpoints, thresholds are interpreted
incorrectly because of a mismatch in interpreting punctuation. The `dot` character
(·) is interpreted as a `comma` (,). For example, a threshold set to 1 (stored as
1.000000 in the CONF file) is returned as 1000000. So, when the resource
model runs, it does not trigger any event, because it does not interpret the
threshold value correctly. The problem is due to a locale mismatch. If this
problem occurs, ensure you have the operating system values correctly
configured, as follows:

```
QLOCALE=/QSYS.LIB/LOCALE.LOCALE
```

For example, for an Italian locale, the following is the correct configuration for
the operating system values:

```
QCCSID=280
QCHRID=697/280
QCNTRYID=IT
QLANGID=ITA
QLOCALE=/QSYS.LIB/IT_IT.LOCALE
```

6. On SLES 9 systems, if you install a locale that uses the `,` (comma) as a decimal
separator, threshold numeric values could be wrongly interpreted. If this
problem occurs, set both the following variables to the same locale value and
restart the engine:

- `LC_CTYPE`
- `LC_NUMERIC`

---

**Rapid Deployment notes**

In non-English environments, note the following:

- On some non-English locales, usually DBCS, truncation of text in panels is fixed
  by setting the environment variable `JAVA2D_USEAWTFONTS=0`. 
Support information

If you have a problem with your IBM software, you want to resolve it quickly. This section describes the following options for obtaining support for IBM software products:

- Searching knowledge bases
- Obtaining fixes
- Receiving weekly support updates on page 70
- Contacting IBM Software Support on page 71

Searching knowledge bases

You can search the available knowledge bases to determine whether your problem was already encountered and is already documented.

Searching the information center

IBM provides extensive documentation that can be installed on your local computer or on an intranet server. You can use the search function of this information center to query conceptual information, instructions for completing tasks, and reference information.

Searching the Internet

If you cannot find an answer to your question in the information center, search the Internet for the latest, most complete information that might help you resolve your problem.

To search multiple Internet resources for your product, use the Web search topic in your information center. In the navigation frame, click Troubleshooting and support ▶ Searching knowledge bases and select Web search. From this topic, you can search a variety of resources, including the following:

- IBM technotes
- IBM downloads
- IBM Redbooks®
- IBM developerWorks®
- Forums and newsgroups
- Google

Obtaining fixes

A product fix might be available to resolve your problem. To determine what fixes are available for your IBM software product, follow these steps:

2. Click Downloads and drivers in the Support topics section.
3. Select the Software category.
4. Select a product in the Sub-category list.
5. In the Find downloads and drivers by product section, select one software category from the Category list.
6. Select one product from the Sub-category list.
7. Type more search terms in the Search within results if you want to refine your search.
8. Click Search.
9. From the list of downloads returned by your search, click the name of a fix to read the description of the fix and to optionally download the fix.

For more information about the types of fixes that are available, see the IBM Software Support Handbook at [http://techsupport.services.ibm.com/guides/handbook.html](http://techsupport.services.ibm.com/guides/handbook.html)

**Receiving weekly support updates**

To receive weekly e-mail notifications about fixes and other software support news, follow these steps:

2. Click My support in the upper right corner of the page.
3. If you have already registered for My support, sign in and skip to the next step. If you have not registered, click register now. Complete the registration form using your e-mail address as your IBM ID and click Submit.
4. Click Edit profile.
5. In the Products list, select Software. A second list is displayed.
6. In the second list, select a product segment, for example, Application servers. A third list is displayed.
7. In the third list, select a product sub-segment, for example, Distributed Application & Web Servers. A list of applicable products is displayed.
8. Select the products for which you want to receive updates, for example, IBM HTTP Server and WebSphere Application Server.
9. Click Add products.
10. After selecting all products that are of interest to you, click Subscribe to email on the Edit profile tab.
11. Select Please send these documents by weekly email.
12. Update your e-mail address as needed.
13. In the Documents list, select Software.
14. Select the types of documents that you want to receive information about.
15. Click Update.

If you experience problems with the My support feature, you can obtain help in one of the following ways:

**Online**
Send an e-mail message to erchelp@ca.ibm.com, describing your problem.

**By phone**
Call 1-800-IBM-4You (1-800-426-4968).
Contacting IBM Software Support

IBM Software Support provides assistance with product defects.

Before contacting IBM Software Support, your company must have an active IBM software maintenance contract, and you must be authorized to submit problems to IBM. The type of software maintenance contract that you need depends on the type of product you have:

• For IBM distributed software products (including, but not limited to, Tivoli, Lotus®, and Rational® products, as well as DB2 and WebSphere products that run on Windows, or UNIX operating systems), enroll in Passport Advantage® in one of the following ways:
  
  **Online**
  Go to the Passport Advantage Web site at [http://www.lotus.com/services/passport.nsf/WebDocs/Passport_Advantage_Home](http://www.lotus.com/services/passport.nsf/WebDocs/Passport_Advantage_Home) and click How to Enroll.
  
  **By phone**
  For the phone number to call in your country, go to the IBM Software Support Web site at [http://techsupport.services.ibm.com/guides/contacts.html](http://techsupport.services.ibm.com/guides/contacts.html) and click the name of your geographic region.

• For customers with Subscription and Support (S & S) contracts, go to the Software Service Request Web site at [https://techsupport.services.ibm.com/ssr/login](https://techsupport.services.ibm.com/ssr/login).


• For IBM eServer™ software products (including, but not limited to, DB2 and WebSphere products that run in zSeries, pSeries, and iSeries environments), you can purchase a software maintenance agreement by working directly with an IBM sales representative or an IBM Business Partner. For more information about support for eServer software products, go to the IBM Technical Support Advantage Web site at [http://www.ibm.com/servers/eserver/techsupport.html](http://www.ibm.com/servers/eserver/techsupport.html).

If you are not sure what type of software maintenance contract you need, call 1-800-IBMSERV (1-800-426-7378) in the United States. From other countries, go to the contacts page of the IBM Software Support Handbook on the Web at [http://techsupport.services.ibm.com/guides/contacts.html](http://techsupport.services.ibm.com/guides/contacts.html) and click the name of your geographic region for phone numbers of people who provide support for your location.

To contact IBM Software support, follow these steps:
1. “Determining the business impact”
2. “Describing problems and gathering information” on page 72
3. “Submitting problems” on page 72

Determining the business impact

When you report a problem to IBM, you are asked to supply a severity level. Therefore, you need to understand and assess the business impact of the problem that you are reporting. Use the following criteria:
Severity 1
The problem has a critical business impact. You are unable to use the program, resulting in a critical impact on operations. This condition requires an immediate solution.

Severity 2
The problem has a significant business impact. The program is usable, but it is severely limited.

Severity 3
The problem has some business impact. The program is usable, but less significant features (not critical to operations) are unavailable.

Severity 4
The problem has minimal business impact. The problem causes little impact on operations, or a reasonable circumvention to the problem was implemented.

Describing problems and gathering information
When describing a problem to IBM, be as specific as possible. Include all relevant background information so that IBM Software Support specialists can help you solve the problem efficiently. To save time, know the answers to these questions:

• What software versions were you running when the problem occurred?
• Do you have logs, traces, and messages that are related to the problem symptoms? IBM Software Support is likely to ask for this information.
• Can you re-create the problem? If so, what steps were performed to re-create the problem?
• Did you make any changes to the system? For example, did you make changes to the hardware, operating system, networking software, and so on.
• Are you currently using a workaround for the problem? If so, be prepared to explain the workaround when you report the problem.

Submitting problems
You can submit your problem to IBM Software Support in one of two ways:

Online
Click Submit and track problems on the IBM Software Support site at [http://www.ibm.com/software/support/probsub.html](http://www.ibm.com/software/support/probsub.html). Type your information into the appropriate problem submission form.

By phone
For the phone number to call in your country, go to the contacts page of the IBM Software Support Handbook at [http://techsupport.services.ibm.com/guides/contacts.html](http://techsupport.services.ibm.com/guides/contacts.html) and click the name of your geographic region.

If the problem you submit is for a software defect or for missing or inaccurate documentation, IBM Software Support creates an Authorized Program Analysis Report (APAR). The APAR describes the problem in detail. Whenever possible, IBM Software Support provides a workaround that you can implement until the APAR is resolved and a fix is delivered. IBM publishes resolved APARs on the Software Support Web site daily, so that other users who experience the same problem can benefit from the same resolution.
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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM’s future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM’s suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

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.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information in softcopy form, the photographs and color illustrations might not display.

### Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (© or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo 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, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SET and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.
Other company, product, and service names may be trademarks or service marks of others.