Note
Before using this information and the product it supports, read the information in Appendix E, “Notices”, on page 229.
About this guide

The IBM Tivoli Monitoring for Databases: Microsoft SQL Server Reference Guide provides detailed information about the tasks, resource models, and commands for the IBM Monitoring for Databases: Microsoft SQL Server software. Use this guide in conjunction with the IBM Tivoli Monitoring for Databases: Microsoft SQL Server User’s Guide.

Who should read this guide

This guide is intended for system architects (for planning) and system administrators and database administrators (for implementation and operation).

Readers should be familiar with the following:
- Windows NT® or 2000® or UNIX® operating systems
- Tivoli software
- Microsoft SQL Server software

What this guide contains

This guide contains the following sections:

- **Chapter 1, “Introduction”, on page 1** describes how the guide is organized. This chapter describes commands, resource model output, and IBM Tivoli Monitoring logging.
- **Chapter 2, “Resource models”, on page 5** describes how to use the resource models to manage databases.
- **Chapter 3, “Tasks”, on page 135** describes how to run the IBM Tivoli Monitoring for Databases: Microsoft SQL Server tasks in the ITMMicrosoftSQLServerTasks library.
- **Appendix A, “Functionality mapping: resource model to monitor”, on page 201** provides a functionality mapping table to compare monitoring capabilities of similar previous releases with this release.
- **Appendix C, “Resource model CIM class quick reference”, on page 219** displays the resource models and their corresponding CIM classes.
- **Appendix D, “Creating custom resource models”, on page 221** provides information on how you can create your own custom resource model.

Publications

This section lists publications in the IBM Tivoli Monitoring for Databases: Microsoft SQL Server library and any other related guides. It also describes how to access Tivoli publications online, and how to order Tivoli publications.
IBM Tivoli Monitoring for Databases: Microsoft SQL Server library

The following guides are available in the IBM Tivoli Monitoring for Databases: Microsoft SQL Server library:

  Provides information about how to install and setup IBM Tivoli Monitoring for Databases: Microsoft SQL Server.

  Provides information about how to use the resource models and tasks used with IBM Tivoli Monitoring for Databases: Microsoft SQL Server.

- **IBM Tivoli Monitoring for Databases: Microsoft SQL Server Release Notes**, SC23- 4851
  Provides information about product features and provides information about the latest changes to the installation requirements and procedures.

- **IBM Tivoli Monitoring for Databases: Microsoft SQL Server Limitations and Workarounds Supplement**, SC23-4850
  Provides information about the latest information about known product limitations and workarounds. To ensure that the information is the latest available, this guide is provided only on the Web, where it is updated on a regular basis. You can access the Limitations and Workarounds guide through the IBM Tivoli Monitoring for Databases: Microsoft SQL Server link on the Tivoli Information Center Web site:
  

Related publications

The following guides also provide useful information:

- **Tivoli Management Framework User’s Guide**
  Provides information about profiles and profile management.

- **Tivoli Management Framework Planning and Installation Guide**
  Provides information about server and hardware requirements.

- **Tivoli Management Framework Reference Guide**
  Provides information about command line commands, such as the `winstall` command.

- **IBM Tivoli Monitoring User’s Guide**
  Provides information about distributed monitoring.

- **IBM Tivoli Enterprise Console User’s Guide**
  Provides information about using the Tivoli Enterprise Console.

- **Tivoli Software Installation Service (SIS) User’s Guide, Version 4.0**
  Provides information about using SIS to install the IBM Tivoli Monitoring for Databases: Microsoft SQL Server software.

The following guides also provide useful information:

- **Microsoft SQL Server Administrator’s Companion**
  Provides information about administering the Microsoft SQL Server in the relevant version of Microsoft SQL Server.

- **Microsoft SQL Server Books Online**
  Provides information about the Microsoft SQL Server administration.

- **Server Concepts Manual**
Provides information about the Microsoft SQL Server and how it works in the relevant version of Microsoft SQL Server.

- **Server SQL Reference**
  Provides descriptions of Structured Query Language (SQL) used to manage information in Microsoft SQL Server databases, refer to the Microsoft SQL Server Books Online for the relevant version of Microsoft SQL Server.

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

http://publib.boulder.ibm.com/tividd/glossary/termsmst04.htm

**Accessing publications online**

The documentation CD contains the publications that are in the product library. The format of the publications is PDF, HTML, or both. Refer to the readme file on the CD for instructions on how to access the documentation.

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. The Tivoli Software Information Center is located at the following Web address:

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

Click the **IBM Tivoli Monitoring for Databases: Microsoft SQL Server** link to access the product library.

**Note:** If you print PDF documents on other than letter-sized paper, select the **Fit to page** check box in the **Adobe Acrobat Print** window. This option is available when you click **File → Print**. **Fit to page** ensures that the full dimensions of a letter-sized page print on the paper that you are using.

**Ordering publications**

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


You can also order by telephone by calling one of these numbers:

- In the United States: 800-879-2755
- In Canada: 800-426-4968

See the following Web site for a list of telephone numbers in other countries:

http://www.ibm.com/software/tivoli/order-lit/

**Accessibility**

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully. With this product, you can use assistive technologies to hear and navigate the interface. You can also use the keyboard instead of the mouse to operate all features of the graphical user interface.
For additional information, see the Accessibility Appendix in the *IBM Tivoli Monitoring for Databases: Microsoft SQL Server User’s Guide*.

**Contacting software support**

If you have a problem with any Tivoli product, refer to the following IBM Software Support Web site:


If you want to contact customer support, see the *IBM Software Support Guide* at the following Web site:


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

- Registration and eligibility
- Telephone numbers and e-mail addresses, depending on the country in which you are located
- Information you must have before contacting IBM Software Support

**Conventions used in this guide**

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

**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:
- Column headings in a table
- Keywords and parameters in text

**Italic**

- Citations (titles of books, diskettes, and CDs)
- Words defined in text
- Emphasis of words (words as words)
- Letters as letters
- 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.
Chapter 1. Introduction

This book is designed for use with the procedures described in the *IBM Tivoli Monitoring for Databases: Microsoft SQL Server User’s Guide*. It provides alphabetical listings and detailed descriptions of the following:

- Resource models
- Tasks
- Tivoli Enterprise Console classes

Running Tivoli commands

You can perform system operations from a UNIX, Windows NT, or Windows 2000 command line interface (CLI) in addition to using the Tivoli desktop. Operations you run from the command line are referred to as CLI commands.

Consider using the command line interface rather than the graphical user interface to invoke a Tivoli management application operation in the following circumstances:

- You do not have access to a graphical user interface, such as when you dial in over a modem.
- You want to group a number of operations together inside a shell script.
- You want to use accessibility tools that require text-based input of commands.

Command syntax

All Tivoli CLI commands begin with the letter `w` to identify them as Tivoli commands. Command names use a `w+verb+object` syntax. Example: Use the `wdmrm` command to add or remove a resource model.

Bash shells

Most Tivoli commands run within a `bash` shell on a managed node or on a Tivoli management region server. A `shell` is a command interpreter that enables the operating system to process commands. You can run commands from a shell command line or include them in shell scripts on UNIX or Windows operating systems. A `Tivoli management region server` is a Tivoli server and the set of clients that it serves. A `Tivoli management region` addresses the physical connectivity of resources, whereas a `policy region` addresses the logical organization of resources.

Environment variables

Before running Tivoli commands, you must set the Tivoli environment variables for the shell. The managed node or Tivoli management region server installation process supplies the scripts to set the Tivoli environment variables. The following sections of this guide contain descriptions of the procedures to run these scripts:

- “Setting the Tivoli environment on UNIX operating systems” on page 3
- “Setting the Tivoli environment on Windows operating systems” on page 3

Note: A few Tivoli commands can run on an endpoint. To set the Tivoli environment variables on an endpoint, see “Establishing the Tivoli environment on an endpoint” on page 3.
Authorization roles

You must also have the appropriate Tivoli authorization role for running each command.

Running Tivoli commands on Windows NT or 2000 operating systems

When you install a Windows managed node or Windows Tivoli management region server, the installation process copies the bash shell executable file to the machine. The bash shell supports many UNIX commands and UNIX command syntax. An example is the forward slash (/) for the directory separator. The bash shell supports the features of the Bourne shell plus; it also has some extensions applicable only to the bash shell.

Note

You can use the Windows NT MS-DOS shell instead of the bash shell to run most Tivoli commands (after you set the Tivoli environment variables with the %SystemRoot%\system32\drivers\etc\Tivoli\setup_env command). However, some commands and Tivoli tasks may require a bash shell to run successfully. For that reason, all examples of Tivoli commands in Tivoli publications use bash shell syntax.

Running Tivoli commands on UNIX operating systems

The UNIX operating systems contain shells. Tivoli commands can run in the Bourne, Korn, C, and bash shells. The Bourne shell is the standard UNIX shell. Every UNIX system includes the Bourne shell. The Korn shell supports the features of the Bourne shell and has extensions applicable only to the Korn shell. The C shell name comes from the C programming language syntax. The bash shell supports many features of the UNIX shells. Both UNIX and Windows systems use the bash shell.

Where to find additional information about shells

The following resources provide additional information about the various shells. Tivoli does not provide opinions or recommendations about any of these resources.

UNIX shells:

Bash shell:
Establishing the Tivoli environment within a shell
When you install a managed node or Tivoli management region server, the installation process supplies shell setup scripts. You use these scripts to set the environment variables required for running Tivoli commands.

Setting the Tivoli environment on UNIX operating systems
Do the following to set the Tivoli environment within a UNIX shell:
1. Log in to a UNIX managed node or Tivoli management region server.
2. Run the appropriate setup script for the shell.
   • For the Bourne, Korn, or bash shell, run the following command:
     
```
     . /etc/Tivoli/setup_env.sh
     ```
   —OR—
   • For the C shell, run the following command:
     
```
     source /etc/Tivoli/setup_env.csh
     ```

Setting the Tivoli environment on Windows operating systems
Do the following to set the Tivoli environment and start a bash shell on Windows:
1. Log in to a Windows managed node or Tivoli management region server.
2. Open a command window.
3. Run the following command in the command window to set Tivoli environment variables:
   
```
   %SystemRoot%\system32\drivers\etc\Tivoli\setup_env.cmd
   ```
4. Run one of the following commands in the command window to start the bash shell:
   • Type `sh` and press Enter.
   —OR—
   • Type `bash` and press Enter.

Establishing the Tivoli environment on an endpoint
When you install an endpoint, the installation process supplies setup scripts. Use these scripts to set the environment variables required for running Tivoli commands on an endpoint.

Do the following to set the Tivoli environment on an endpoint:
1. Log in to an endpoint.
2. Open a command window.
3. Run the appropriate setup script in the command window. [Table 1 on page 4 contains setup scripts for the different endpoint operating systems.]
Table 1. Setup scripts for endpoint operating systems

<table>
<thead>
<tr>
<th>Operating system</th>
<th>Setup script location</th>
<th>Setup script name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIX, Solaris</td>
<td>/etc/Tivoli/lcf/endpoint_label</td>
<td>lcf_env.sh (for Bourne, Korn, and bash shells)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—OR—</td>
</tr>
<tr>
<td></td>
<td></td>
<td>lcf_env.csh (for C shell)</td>
</tr>
<tr>
<td>Windows</td>
<td>%SystemRoot%\Tivoli\lcf\endpoint_label</td>
<td>lcf_env.cmd (for MS-DOS)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>—OR—lcf_env.sh (for bash shell)</td>
</tr>
</tbody>
</table>

Common attributes for IBM Tivoli Monitoring for Databases: Microsoft SQL Server

This section describes the common attributes used in IBM Tivoli Monitoring for Databases: Microsoft SQL Server.

**Status**  The returned response level on which the resource model triggered. Response levels are defined when adding a resource model. Depending on the resource model, response levels can be a default value or a user-defined value.

- **Normal** is returned only when no severity levels are exceeded and the resource model is set to always return a value.
- **Warning**, **Severe**, or **Critical** is returned when a severity threshold is broken.

**application_label**  The registered object’s Tivoli Management Environment (TME) label.

**application_oid**  The registered object’s Tivoli Management Environment (TME) object identifier.

**application_class**  The registered object’s Tivoli Management Environment (TME) class.

**application_version**  The managed resource’s version; for example, v7, or 2k.

**MSSQL_SID**  The Microsoft SQL Server ID.

**MSSQL_HOME**  The location of the Microsoft SQL Server binaries and libraries.

**HOST**  The Tivoli management agent endpoint on which the Microsoft SQL Server database resource resides.

**ENDPOINT_OID**  The endpoint object ID. It is used internally by Tivoli and can be used for tracing activities with this object.

**INTERP**  The type of operating system on which the resource model executes, such as Windows.
Chapter 2. Resource models

This chapter contains detailed information about the resource models for IBM Tivoli Monitoring for Databases: Microsoft SQL Server. Resource models capture and return information, such as database status and server availability, about a resource or software application in the Tivoli management environment. You can customize many resource model settings to meet your monitoring needs.

In this reference guide, a section for each resource model describes its configuration:

Description
Purpose of the resource model and a table that contains an overview of the resource model. The overview table contains the following information about the resource model:

- Internal name
  Name of the resource model as you use it in the command line.

- Category
  A product, object, or operating system name used by IBM Tivoli Monitoring for organizational purposes. For this product, the category name is "Microsoft SQL Server."

- Indications
  List of indications for the resource model. A resource model generates an indication if certain conditions implied by the resource model settings are not satisfied in a given cycle. The resource model uses an algorithm to determine the combination of settings that generates an indication.

- Tasks and built-in actions
  List of tasks and built-in actions for the resource model, if any. For any event, recovery actions, such as Tivoli Management Framework tasks or built-in actions, can be run automatically. The actions can take positive steps to remedy the situation, and can ensure that information about the event is distributed to the appropriate authorities or entities.

- Default cycle time
  Specifies the default cycle time for the resource model. Cycle time is the duration of the interval within which a resource model gathers data. Cycle time is measured in seconds. Each of the resource models supplied with the software has a default cycle time in seconds, which you can modify according to your needs.

Target managed resource
Name of the managed resource to which this resource model is relevant.

Indications and events
A table that lists the resource model’s indications followed by a section of more specific information about each indication.

The table contains the following information for each indication:

- Event
  An event verifies the persistence of a given indication by eliminating unrepresentative highs and lows for the indication.

- Default severity
Indicates how serious a triggered event is: fatal, critical, warning, harmless, or minor.

- Clearing events
  Specifies whether the resource model has clearing events. A clearing event is a resource model function that, if enabled, allows IBM Tivoli Monitoring to close an error event when the circumstances that caused the event are no longer present. Clearing events can be processed by the Tivoli Enterprise Console server and by Tivoli Business Systems Manager.

- Page
  Page number where the indication is described in this guide.

Each section about a specific indication contains the following information:

- When the resource model sends the indication and why.
- List of the attributes for the indication and notation regarding which attributes are keys.
- Table that describes the following default settings for the indication:
  - Send indications to Tivoli Enterprise Console
    Specifies whether the resource model sends indications to Tivoli Enterprise Console. If YES and Tivoli Enterprise Console is installed, IBM Tivoli Monitoring sends indications to Tivoli Enterprise Console. If Tivoli Business Systems Manager is installed, Tivoli Enterprise Console sends the indications to Tivoli Business Systems Manager.
  - Send indications to Tivoli Business Systems Manager
    Specifies whether IBM Tivoli Monitoring sends indications to Tivoli Business Systems Manager. The default is NO. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.
  - Occurrences
    The number of occurrences refers to how many consecutive times the problem reported by the indication occurs before sending an indication.
  - Holes
    The number of holes refers to the number of cycles during which a given indication does not occur for a given resource model. In other words, the conditions specified for the generation of the indication are not met.
  - Associated tasks and built-in actions
    List of tasks and built-in actions associated with the indication.

**Thresholds**
A table listing the thresholds, if any, for the resource model with a short description and default value for each threshold. A threshold is a named property of the resource with a modifiable default value. Typically, a threshold’s value represents a significant reference level of a performance-related entity. A system administrator might want to know when this level is exceeded or not reached.

**Parameters**
A table listing the resource model’s parameters, if any, with a short description and default value for each parameter. A parameter can take the form of a list of strings, a list of numeric values, a list of predetermined...
Boolean values from which you can make any combination of selections, or a choice list of mutually exclusive alternatives.

**Tasks and built-in actions**
List of tasks and built-in actions including a description of each one.

**Logging**
A table showing the name of the managed resource, context, and properties that the resource model logs, with key properties noted. A resource model does not log data by default; you can enable logging to collect data for any endpoint and write it in a local database. You can also store raw or aggregated data and view it through the IBM Tivoli Monitoring Web Health Console.

**Return codes**
Information that the resource model returns, such as status or availability.

**CLI example**
Example of the syntax for a resource model that illustrates the variable options of the resource model-specific `wdmeditprf` command, such as the name of the resource model, thresholds, parameters, parameter values, and events.

See the "Working with resource models" chapter of the IBM Tivoli Monitoring for Databases: Microsoft SQL Server User’s Guide for more information on resource models and resource model customization.

### Return codes

The resource model returns information about its status and availability. The **Return code** is displayed in the IBM Tivoli Monitoring Web Health Console status field, or by using the `wdmlseng` command.

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Description</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The resource model was distributed to an incorrect target managed resource. The correct target managed resource has a type of &quot;MicrosoftSQLServer.&quot;</td>
<td>Distribute the resource model to the correct resource with a type of &quot;MicrosoftSQLServer.&quot;</td>
</tr>
<tr>
<td>2</td>
<td>The resource model was distributed to the wrong object version. This might be seen in resource models that are specific to a particular version of Microsoft SQL Server.</td>
<td>Distribute the resource model to the proxy object for the supported Microsoft SQL Server version.</td>
</tr>
<tr>
<td>3</td>
<td>The resource model has an invalid parameter configuration. Possibly, the parameter value was out of the supported range, or an invalid type.</td>
<td>Provide valid parameters. The &quot;Parameters&quot; section for each resource model in this guide lists valid parameters for each resource model.</td>
</tr>
<tr>
<td>4</td>
<td>The resource model has an invalid threshold value. Possibly, the threshold value is not within the specified range.</td>
<td>Provide valid thresholds. The &quot;Thresholds&quot; section for each resource model in this guide lists valid thresholds for each resource model.</td>
</tr>
<tr>
<td>5</td>
<td>The resource is unavailable. Possibly, the Microsoft SQL Server database or server is not running.</td>
<td>Consider restarting the unavailable resource. It is possible the resource might be unavailable by intention.</td>
</tr>
</tbody>
</table>
Table 2. Resource model return codes (continued)

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Description</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>The data collected from a resource is determined to be invalid. Possibly, a database column is of a different type than expected.</td>
<td>Check the $Tmw2k.log log file and check the data provider logs in the following directory: $LCF_DATDIR/LCFNEW/CTW/logs</td>
</tr>
<tr>
<td>7</td>
<td>An error occurred while collecting data. This appears when an exception is thrown by the native provider.</td>
<td>Check the $Tmw2k.log log file and check the data provider logs in the following directory: $LCF_DATDIR/LCFNEW/CTW/logs. Additionally, you can contact customer support.</td>
</tr>
<tr>
<td>8</td>
<td>An unknown error has occurred.</td>
<td>Check the $Tmw2k.log log file and check the data provider logs in the following directory: $LCF_DATDIR/LCFNEW/CTW/logs. Additionally, you can contact customer support.</td>
</tr>
</tbody>
</table>

Table 3 contains the internal name and a brief description of each resource model. Following the table are sections of information about each resource model presented in alphabetical order.

Table 3. IBM Tivoli Monitoring for Databases: Microsoft SQL Server resource models

<table>
<thead>
<tr>
<th>Resource model</th>
<th>Internal Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Availability” on page 9</td>
<td>Microsoft_SQL_Server_Availability</td>
<td>Checks the availability of Microsoft SQL servers and databases.</td>
</tr>
<tr>
<td>“Cache/CPU” on page 18</td>
<td>Microsoft_SQL_Server_CacheCpu</td>
<td>Monitors buffer usage in the Microsoft SQL Server, ensures the server is configured properly, and determines its load.</td>
</tr>
<tr>
<td>“Errors” on page 38</td>
<td>Microsoft_SQL_Server_Errors</td>
<td>Monitors error log entries, and enables the customer to ensure that the server is healthy.</td>
</tr>
<tr>
<td>“Jobs” on page 57</td>
<td>Microsoft_SQL_Server_Jobs</td>
<td>Monitors status and run duration for jobs running on Microsoft SQL Server.</td>
</tr>
<tr>
<td>“Locks” on page 67</td>
<td>Microsoft_SQL_Server_Locks</td>
<td>Monitors locks in the Microsoft SQL Server to ensure accessibility of databases, tables, and records.</td>
</tr>
<tr>
<td>“Replication” on page 73</td>
<td>Microsoft_SQL_Server_Replication</td>
<td>Monitors replication in the Microsoft SQL Server to ensure that replication is performing properly and that the server is configured properly.</td>
</tr>
<tr>
<td>“Replication Jobs” on page 97</td>
<td>Microsoft_SQL_Server_Replication_Jobs</td>
<td>Monitors replication jobs to verify completion and returns the status of replication jobs.</td>
</tr>
<tr>
<td>“Space Usage” on page 107</td>
<td>Microsoft_SQL_Server_Space_Usage</td>
<td>Monitors space usage in the Microsoft SQL Server.</td>
</tr>
<tr>
<td>“Users/Transactions” on page 118</td>
<td>Microsoft_SQL_Server_Users_Transactions</td>
<td>Monitors user and transaction activity in the Microsoft SQL Server and ensures the server is appropriately sized and configured.</td>
</tr>
</tbody>
</table>
Availability

Description

Checks the availability of Microsoft SQL services and databases. Use this resource model to monitor when:

- A Microsoft SQL database is suspect
  
  Use the **Microsoft SQL Server Suspect Database** indication to monitor when a database is suspect and cannot be loaded.

- A Microsoft SQL service is stopped
  
  Use the **Microsoft SQL Server Stopped Service** indication to monitor when a service is stopped.

- A Microsoft SQL service is paused or not running
  
  Use the **Microsoft SQL Server Paused Service** indication to monitor when a Microsoft SQL Server service is in a paused state.

By default, this resource model monitors all databases. You can exclude certain databases from monitoring by using the **Exclude Databases** parameter.

By default, this resource model monitors the following Microsoft SQL Server 7.0 and 2000 services:

- MSSQLServer (Database engine)
- SQLServerAgent (SQL Server Agent)
- MSDTC (Microsoft Distributed Transaction Coordinator)
- Microsoft Search (Microsoft search service)
- MSSQLServerOLAPService (Microsoft Analysis Server, available for Microsoft SQL Server 2000 only)

You can customize this resource model by adding additional services or removing unnecessary services with the **Services** parameter.

Note: You can install multiple instances on the same server with Microsoft SQL Server 2000. This impacts the database engine and SQL server agent services. The service names are for the default instance, and might exist on a given implementation. If you implement multiple instances of Microsoft SQL Server 2000 on a system, the instances have unique names. In this situation, there will be multiple database engines and SQL server agent services. The database engine services will be MSSQL$<ServerInstanceName>. The SQL server agent services will be SQLAgent$<ServerInstanceName>. The resource model monitors these database and agent services provided that their names are the Microsoft standard names and the " _ DefaultMSSQLServer_ " and " _ DefaultSQLServerAgent_ " entries are still in the **Services** parameter list.

<table>
<thead>
<tr>
<th>Resource model overview</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal name</td>
<td>Microsoft_SQL_Server_Availability</td>
</tr>
<tr>
<td>Category</td>
<td>Microsoft SQL Server</td>
</tr>
<tr>
<td>Indications</td>
<td><strong>Microsoft SQL Server Paused Service</strong> occurs when a Microsoft SQL Server service is paused.</td>
</tr>
<tr>
<td></td>
<td><strong>Microsoft SQL Server Stopped Service</strong> occurs when a Microsoft SQL Server service is stopped.</td>
</tr>
<tr>
<td></td>
<td><strong>Microsoft SQL Server Suspect Database</strong> occurs when a Microsoft SQL Server database is in a suspect state.</td>
</tr>
</tbody>
</table>
Resource model overview

<table>
<thead>
<tr>
<th>Tasks and built-in actions</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default cycle time</td>
<td>300 seconds (5 minutes)</td>
</tr>
</tbody>
</table>

**Target managed resource**

MicrosoftSQLServer

**Indications and events**

The following table summarizes the indications for this resource model, the events associated with the indications, the default severity of the events, and where to find a detailed description of the indication. The resource model generates an event when the threshold you specify for the indication is triggered.

<table>
<thead>
<tr>
<th>Indication</th>
<th>Generated event</th>
<th>Default severity</th>
<th>Clearing events</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft SQL Server Paused Service</td>
<td>Microsoft_SQL_Server_Paused_Service</td>
<td>Warning</td>
<td>Yes</td>
<td>10</td>
</tr>
<tr>
<td>Microsoft SQL Server Stopped Service</td>
<td>Microsoft_SQL_Server_Stopped_Service</td>
<td>Critical</td>
<td>Yes</td>
<td>12</td>
</tr>
<tr>
<td>Microsoft SQL Server Suspect Database</td>
<td>Microsoft_SQL_Server_Suspect_Database</td>
<td>Critical</td>
<td>Yes</td>
<td>13</td>
</tr>
</tbody>
</table>

**Microsoft SQL Server Paused Service**

Occurs when a Microsoft SQL Server service is paused. Services must be available so that the replication process and scheduled SQL jobs can run. Connection to the Microsoft SQL Server is not necessary for an accurate reporting of the service state. Paused services cannot accept new logons, which causes an availability issue.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

```
<application_label>: The Microsoft SQL Server service <MSSQLServiceName> for server <MSSQLServerName> is <MSSQLServiceState>.
```

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**
  The registered object’s Tivoli Management Environment (TME) class.

- **application_label**
  The registered object’s Tivoli Management Environment (TME) label.

- **application_oid**
  The registered object’s Tivoli Management Environment (TME) object identifier.
application_version
The managed resource’s version; for example, v7, or 2k.

MSSQLServerName
The instance name of Microsoft SQL Server.

MSSQLServiceName
The name of the Microsoft SQL Server service.

MSSQLServiceState
The current state of the Microsoft SQL Server service. States include: paused, running, or stopped.

MSSQLServiceStatus
The current status of the Microsoft SQL Server service. Various operational and non-operational statuses can be defined. The operational status is either OK, Degraded or Pred Fail. Pred Fail indicates that an element might be functioning properly but might fail in the near future.

MSSQLStartMode
The start mode of the Win32 base service. The following list defines the start modes:
- Boot Specifies a device driver started by the operating system loader. This value is valid only for driver services.
- System Specifies a device driver started by the IoInitSystem function. This value is valid only for driver services.
- Automatic Specifies a service to be started automatically by the service control manager during system startup.
- Manual Specifies a service to be started by the service control manager when a process calls the StartService function.
- Disabled Specifies a service that can no longer be started.

This indication has no thresholds.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>1</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.
Microsoft SQL Server Stopped Service

Occurs when a Microsoft SQL Server service is stopped. The following services are critical for normal operations:

- The MSSQLServer service must be running for any operations to be carried out on the server or its databases. The server might have stopped due to a user request or an internal error. It might also have stopped if the Windows computer was rebooted and the service is not automatically set to restart upon reboot. You can configure an automatic response to start the service again. This reduces the amount of time the server is unavailable to users and applications.
- SQLServerAgent must be running for SQL scheduled jobs and replication to operate. This indication helps to identify a replication bottleneck quickly and to avoid a backlog of replicated transactions or an unacceptable delay in subscribers getting published information. It also alerts you when scheduled jobs are unable to run.
- MSDTC must be running in order for distributed transactions to be coordinated correctly.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

```
<application_label>: The Microsoft SQL Server service <MSSQLServiceName> for server <MSSQLServerName> is <MSSQLServiceState>.
```

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**
  The registered object’s Tivoli Management Environment (TME) class.
- **application_label**
  The registered object’s Tivoli Management Environment (TME) label.
- **application_oid**
  The registered object’s Tivoli Management Environment (TME) object identifier.
- **application_version**
  The managed resource’s version; for example, v7, or 2k.
- **MSSQLServerName**
  The instance name of Microsoft SQL Server.
- **MSSQLServiceName**
  The name of the Microsoft SQL Server service.
- **MSSQLServiceState**
  The current state of the Microsoft SQL Server service. States include: paused, running, and stopped.
- **MSSQLServiceStatus**
  The current status of the Microsoft SQL Server service. Various operational and non-operational statuses can be defined. The operational status is either OK, Degraded or Pred Fail. Pred Fail indicates that an element might be functioning properly but predicting a failure in the near future.
MSSQLStartMode
The start mode of the Win32 base service. The following lists and defines
the start modes.

Boot Specifies a device driver started by the operating system loader.
This value is valid only for driver services.

System Specifies a device driver started by the IoInitSystem function. This
value is valid only for driver services.

Automatic Specifies a service to be started automatically by the service control
manager during system startup.

Manual Specifies a service to be started by the service control manager
when a process calls the StartService function.

Disabled Specifies a service that can no longer be started.

This indication has no thresholds.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems</td>
<td>No</td>
</tr>
<tr>
<td>Manager Occurrences</td>
<td>1</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

Note: If you have Tivoli Business Systems Manager configured for your systems,
Tivoli Enterprise Console automatically forwards events to Tivoli Business
Systems Manager. Do not change the configuration of the indication to send
events to Tivoli Business Systems Manager.

Microsoft SQL Server Suspect Database
Occurs when a Microsoft SQL Server database is in a suspect state. Microsoft SQL
Server marks a database as Suspect and fails recovery when problems occur while
attempting to recover the database during Startup. If you can determine the reason
why the database was marked as Suspect, it might be safe to reset it. Before doing
so, check the Microsoft SQL Server Books Online. If you cannot determine whether it
is safe to reset the database, restore it from a known good backup.

This resource model determines if a database is suspect with one of the following
SQL commands:

- Microsoft SQL Server 2000
  Select DatabasePropertyEx (<database name>, 'Status')

  If the database is suspect, the result of the query is SUSPECT.

- Microsoft SQL Server 7
  Select DatabaseProperty (<database name>, 'IsShutDown')
If the database is shutdown, the result of the query is 1. The resource model considers the database to be suspect.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

<application_label>: The Microsoft SQL Server database <MSSQLDatabaseName> on the Microsoft SQL Server <MSSQLServerName> is SUSPECT.

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**
  The registered object’s Tivoli Management Environment (TME) class.

- **application_label**
  The registered object’s Tivoli Management Environment (TME) label.

- **application_oid**
  The registered object’s Tivoli Management Environment (TME) object identifier.

- **application_version**
  The managed resource’s version; for example, v7, or 2k.

- **MSSQLDatabaseName**
  The name of the Microsoft SQL database.

- **MSSQLDatabaseStatus**
  The status of the Microsoft SQL database. Status is either online, or suspect.

- **MSSQLServerName**
  The instance name of Microsoft SQL Server.

This indication has no thresholds.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>1</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.
### Thresholds
None

### Parameters
The following table lists the parameters that can be set for the Availability resource model. The table shows the name, a short description, and the default value for each parameter:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exclude Databases</td>
<td>Type any databases to exclude from monitoring for suspect status.</td>
<td>Blank (monitors every database)</td>
</tr>
<tr>
<td></td>
<td>When this parameter is blank, the resource model monitors every database.</td>
<td></td>
</tr>
<tr>
<td>Refresh Database Frequency</td>
<td>Defines how often the resource model should get a new list of databases from the Microsoft SQL Server. The value represents the number of resource model cycles.</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>For example, if the resource model cycle time is 60 seconds, and this parameter is set to 5, then the resource model gets the list of databases every 5 minutes.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>When a database is added, it is automatically monitored the next time the list of databases is retrieved.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Specify only one entry for this parameter.</td>
<td></td>
</tr>
<tr>
<td>Services</td>
<td>Defines the Microsoft SQL Server services to monitor for stopped status.</td>
<td><em>DefaultMSSQLServer</em> * <em>DefaultSQLServerAgent</em>* <em>MSDT</em> <em>MSSEARCH</em> <em>MSSQLServerOLAPService</em></td>
</tr>
<tr>
<td></td>
<td>You can also add services to monitor by typing in the service name. See the information below for more details about the <em>DefaultMSSQLServer</em> and <em>DefaultSQLServerAgent</em> options.</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** An asterisk (*) denotes a key property.

Additional information about _DefaultMSSQLServer_ and _DefaultSQLServerAgent_:
- **For Microsoft SQL Server 7.0**, the resource model monitors the MSSQLServer and SQLServerAgent services when you select the _DefaultMSSQLServer_ and _DefaultSQLServerAgent_ parameters. If you remove these keywords and do not specify the services, the resource model does not monitor the MSSQLServer and SQLServerAgent services.
- **For Microsoft SQL Server 2000**, the resource model checks to see if the services exist as named instances (MSSQLServer$.servernameFromAppProxyContext). If they do not exist, the resource model checks for the existence of a default instance service (MSSQLServer). If the default instance service exists, it is monitored.

### Tasks and built-in actions
None
Logging

You can log data for the properties of the managed resource listed in the following table. The table shows the context of the managed resource and the properties that the resource model logs for the IBM Tivoli Monitoring Web Health Console.

<table>
<thead>
<tr>
<th>Managed resource</th>
<th>Context</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSSQLDatabase</td>
<td>State</td>
<td>MSSQLServer.MSSQLServerName*&lt;br&gt;The instance name of Microsoft SQL Server.&lt;br&gt;MSSQLDatabase.MSSQLDatabaseName*&lt;br&gt;Specifies the name of the Microsoft SQL Server database.&lt;br&gt;MSSQL_Database_Online&lt;br&gt;Specifies whether the database is online. A returned status of “1” indicates the database is online. A returned status of “0” indicates a status other than online.&lt;br&gt;MSSQL_Database_Suspect&lt;br&gt;Specifies whether the database is suspect. A returned status of “1” indicates a suspect database. A returned status of “0” indicates a status other than suspect.</td>
</tr>
<tr>
<td>MSSQLService</td>
<td>State</td>
<td>MSSQLServer.MSSQLServerName*&lt;br&gt;The instance name of Microsoft SQL Server.&lt;br&gt;MSSQLService.MSSQLServiceName*&lt;br&gt;The name of the Microsoft SQL Server service.&lt;br&gt;MSSQL_Service_Paused&lt;br&gt;Specifies whether the service is paused. A returned status of “1” represents the service is paused. A returned status of “0” indicates a status other than paused.&lt;br&gt;MSSQL_Service_Running&lt;br&gt;Specifies whether the service is running. A returned status of “1” indicates the service is running. A returned status of “0” indicates a status other than running.&lt;br&gt;MSSQL_Service_Stopped&lt;br&gt;Specifies whether the service is stopped or not. A returned status of “1” indicates the service is stopped. A returned status of “0” indicates a status other than stopped.</td>
</tr>
</tbody>
</table>

Note: An asterisk (*) denotes a key property.

Return codes

Table 2 on page 7 contains a listing of resource model return codes, their description, and what action you can take to resolve each code. The return code number is displayed in the IBM Tivoli Monitoring Web Health Console status field, or by using the wdmlseng command.
CLI example

```bash
wdmeditprf -P $Prof1 -edit Microsoft_SQL_Server_Availability \
  -DelPar Services "MSSQLServerOLAPService" \n  -DelPar Services "DefaultMSSQLServer_" \n  -DelPar Services "DefaultSQLServerAgent_" \n  -AddPar Services "MSSQLServer" \n  -AddPar Services "MSSQLServerAgent" \n  -AddPar Services "MSDTC" \n  -AddPar Services "MSSEARCH" \n  -AddPar Refresh_Database_Frequency "5" \n  -e Microsoft_SQL_Server_Stopped_Service \n    -o 1 -h 0 -severity CRITICAL -NoSendTBSM -"$MyTec" \n  -e Microsoft_SQL_Server_Paused_Service \n    -o 1 -h 0 -severity CRITICAL -NoSendTBSM -"$MyTec" \n  -e Microsoft_SQL_Server_Suspect_Database \n    -o 1 -h 0 -severity CRITICAL -NoSendTBSM -"$MyTec"
```
Cache/CPU

Description
Monitors the following Microsoft SQL Server buffer cache and CPU usage metrics:
- Buffer cache hit ratio
- Buffer extended memory cache hit ratio
- CPU busy
- Checkpoint writes per second
- Free buffers
- Lazy writes per second
- Page reads and writes per second
- Read ahead pages per second
- Stolen page count growth

Use this resource model to monitor memory usage and efficiency. Appropriate levels of memory allocation and usage can enhance performance.

Microsoft SQL Server processes only access data and index pages in memory, not directly on disk. When a process needs to access a data or index page, the page must first exist in a buffer pool (memory). If the requested page is not in memory, a physical I/O (disk read or write) must be performed to retrieve the page into memory so that the process can access the page.

Physical I/Os to read or write data are expensive and time-consuming. Strive to have as few physical I/Os as possible. Use the Microsoft SQL Server Page Reads Per Second too high and Microsoft SQL Server Page Writes Per Second too high indications to monitor high physical I/O activity.

<table>
<thead>
<tr>
<th>Resource model overview</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal name</td>
</tr>
<tr>
<td>Category</td>
</tr>
</tbody>
</table>
## Resource model overview

### Indications

<table>
<thead>
<tr>
<th>Indication</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Microsoft SQL Server Buffer Cache Hit Ratio too low</strong></td>
<td>occurs when the buffer cache hit ratio falls below the <strong>Buffer Cache Hit Ratio lower bound</strong> threshold.</td>
</tr>
<tr>
<td><strong>Microsoft SQL Server Buffer Extended Memory Cache Hit Ratio too low</strong></td>
<td>occurs when the buffer extended memory cache hit ratio is less than the <strong>Buffer Extended Memory Cache Hit Ratio lower bound</strong> threshold. Note: Available for Microsoft SQL Server version 7 only.</td>
</tr>
<tr>
<td><strong>Microsoft SQL Server Checkpoint Writes Per Second too high</strong></td>
<td>occurs when the number of checkpoint writes per second exceeds the <strong>Checkpoint Writes Per Second upper bound</strong> threshold.</td>
</tr>
<tr>
<td><strong>Microsoft SQL Server CPU Percent Busy too high</strong></td>
<td>occurs when percentage of CPU busy exceeds the <strong>CPU Percent Busy upper bound</strong> threshold.</td>
</tr>
<tr>
<td><strong>Microsoft SQL Server Free Buffers too low</strong></td>
<td>occurs when the number of cache buffers in the free buffer pool is less than the <strong>Free Buffers lower bound</strong> threshold.            Note: Available for Microsoft SQL Server version 7 only.</td>
</tr>
<tr>
<td><strong>Microsoft SQL Server Lazy Writes Per Second too high</strong></td>
<td>occurs when the number of lazy writes per second exceeds the <strong>Lazy Writes Per Second upper bound</strong> threshold.</td>
</tr>
<tr>
<td><strong>Microsoft SQL Server Page Reads Per Second too high</strong></td>
<td>occurs when the number of page reads per second exceeds the <strong>Page Reads Per Second upper bound</strong> threshold.</td>
</tr>
<tr>
<td><strong>Microsoft SQL Server Page Writes Per Second too high</strong></td>
<td>occurs when the number of page writes per second exceeds the <strong>Page Writes Per Second upper bound</strong> threshold.</td>
</tr>
<tr>
<td><strong>Microsoft SQL Server Read Ahead Pages Per Second too high</strong></td>
<td>occurs when the number of read ahead pages per second exceeds the <strong>Read Ahead Pages Per Second upper bound</strong> threshold.</td>
</tr>
<tr>
<td><strong>Microsoft SQL Server Stolen Page Count Growth too high</strong></td>
<td>occurs when the stolen page count growth increases by an amount equal to or greater than the <strong>Stolen Page Count Growth</strong> threshold for the current monitoring cycle as compared to prior monitoring cycle.</td>
</tr>
</tbody>
</table>

### Tasks and built-in actions

- None

### Default cycle time

- 600 seconds (10 minutes)

---

**Target managed resource**

MicrosoftSQLServer

**Indications and events**

The following table summarizes the indications for this resource model, the events associated with the indications, the default severity of the events, and where to find a detailed description of the indication. The resource model generates an event when the threshold you specify for the indication is triggered.
<table>
<thead>
<tr>
<th>Indication</th>
<th>Generated event</th>
<th>Default severity</th>
<th>Clearing events</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft SQL Server Buffer Cache Hit Ratio too low</td>
<td>Microsoft_SQL_Server_Low_BufferCacheHitRatio</td>
<td>Warning</td>
<td>Yes</td>
<td>20</td>
</tr>
<tr>
<td>Microsoft SQL Server Buffer Extended Memory Cache Hit Ratio too low</td>
<td>Microsoft_SQL_Server_Low_BufferExtMemCacheHitRatio</td>
<td>Warning</td>
<td>Yes</td>
<td>22</td>
</tr>
<tr>
<td>Microsoft SQL Server Checkpoint Writes Per Second too high</td>
<td>Microsoft_SQL_Server_High_CheckpointWritesPerSec</td>
<td>Warning</td>
<td>Yes</td>
<td>23</td>
</tr>
<tr>
<td>Microsoft SQL Server CPU Percent Busy too high</td>
<td>Microsoft_SQL_Server_High_CpuPercentBusy</td>
<td>Minor</td>
<td>Yes</td>
<td>25</td>
</tr>
<tr>
<td>Microsoft SQL Server Free Buffers too low</td>
<td>Microsoft_SQL_Server_Low_FreeBuffers</td>
<td>Warning</td>
<td>Yes</td>
<td>26</td>
</tr>
<tr>
<td>Microsoft SQL Server Lazy Writes Per Second too high</td>
<td>Microsoft_SQL_Server_High_LazyWritesPerSec</td>
<td>Warning</td>
<td>Yes</td>
<td>28</td>
</tr>
<tr>
<td>Microsoft SQL Server Page Reads Per Second too high</td>
<td>Microsoft_SQL_Server_High_Page ReadsPerSec</td>
<td>Warning</td>
<td>Yes</td>
<td>29</td>
</tr>
<tr>
<td>Microsoft SQL Server Page Writes Per Second too high</td>
<td>Microsoft_SQL_Server_High_Page WritesPerSec</td>
<td>Warning</td>
<td>Yes</td>
<td>30</td>
</tr>
<tr>
<td>Microsoft SQL Server Read Ahead Pages Per Second too high</td>
<td>Microsoft_SQL_Server_High_ReadAheadPagesPerSec</td>
<td>Warning</td>
<td>Yes</td>
<td>32</td>
</tr>
<tr>
<td>Microsoft SQL Server Stolen Page Count Growth too high</td>
<td>Microsoft_SQL_Server_High_StolenPageCountGrowth</td>
<td>Warning</td>
<td>Yes</td>
<td>33</td>
</tr>
</tbody>
</table>

**Microsoft SQL Server Buffer Cache Hit Ratio too low**

Occurs when the buffer cache hit ratio falls below the **Buffer Cache Hit Ratio lower bound** threshold. The buffer cache hit ratio is the percentage of pages found in the buffer cache without having to read them from the disk. The percentage is calculated as the total number of successful cache hits divided by the total number of requested cache lookups since the last cycle of the resource model.
Each instance of Microsoft SQL Server has a buffer cache. Data remains in the buffer cache until it has not been referenced for some time and the database needs the buffer area to record more data. Data is written back to disk only if it is modified.

Because reading from the cache is less expensive than reading from disk, you want this percentage to be high. After your system maintains a steady state of operation, this metric should achieve rates of 90 or higher. You can increase the buffer cache hit ratio by increasing the amount of memory available to Microsoft SQL Server.

Microsoft SQL Server uses dynamic memory allocation to tune itself. The ideal scenario is for the database server to run on a dedicated machine with dynamic memory allocation in effect. When needed, Microsoft SQL Server increases memory allocation if there is more than 5 megabytes of unallocated memory available. It concedes memory if unallocated memory available falls below 5 megabytes.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

```
<application_label>: Microsoft SQL Server <MSSQLServerName> - Buffer Cache Hit Ratio is <MSSQLBufferCacheHitRatio>, which is below threshold of <LowerBound>.
```

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**: The registered object’s Tivoli Management Environment (TME) class.
- **application_label**: The registered object’s Tivoli Management Environment (TME) label.
- **application_oid**: The registered object’s Tivoli Management Environment (TME) object identifier.
- **application_version**: The managed resource’s version; for example, v7, or 2k.
- **LowerBound**: The value set as the minimum allowable for the specified threshold.
- **MSSQLBufferCacheHitRatio**: The percentage of the buffer cache hits to total requests over the lifetime of an instance. The percentage calculates how often a requested page is found in the buffer cache without requiring disk access. The buffer cache is the portion of the instance that holds pages of data. All user processes that are connected to the instance share access to the buffer cache.
- **MSSQLServerName**: The instance name of Microsoft SQL Server.

This indication has the following threshold:

- Buffer Cache Hit Ratio lower bound

For more information about this threshold, see “Thresholds” on page 35.
The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>4</td>
</tr>
<tr>
<td>Holes</td>
<td>2</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Microsoft SQL Server Buffer Extended Memory Cache Hit Ratio too low**

**Note:** Available for Microsoft SQL Server version 7 only.

Occurs when the buffer extended memory cache hit ratio is less than the Buffer Extended Memory Cache Hit Ratio lower bound threshold. Extended memory is disk memory that is allocated for cache use in addition to the conventional buffer cache. The buffer extended memory cache hit ratio is the percentage of pages accessed in the extended memory. The percentage is calculated as the total number of successful cache hits divided by the total number of requested cache lookups since the last cycle of the resource model.

Reading from the extended memory cache is less expensive than reading from the disk. If this ratio is too low, consider increasing the amount of memory available to Microsoft SQL Server. Appropriate allocation of extended memory can enhance system performance. For example, on a computer with 8 GB of memory, a reasonable value for extended memory size is in the range of 5,000 to 6,000. This allocates 2 GB of buffer memory and most of the rest for use as extended memory cache.

Microsoft SQL Server uses dynamic memory allocation to tune itself. The ideal scenario is for the database server to run on a dedicated machine with dynamic memory allocation in effect. When needed, Microsoft SQL Server increases memory allocation if there is more than 5 megabytes of unallocated memory available. It concedes memory if unallocated memory available falls below 5 megabytes.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

```<application_label>: Microsoft SQL Server <MSSQLServerName> - Buffer Extended Memory Cache Hit Ratio is <MSSQLBufferExtMemCacheHitRatio>, which is below threshold of <LowerBound>.```

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.
The indication has the following attributes:

**application_class**
The registered object’s Tivoli Management Environment (TME) class.

**application_label**
The registered object’s Tivoli Management Environment (TME) label.

**application_oid**
The registered object’s Tivoli Management Environment (TME) object identifier.

**application_version**
The managed resource’s version; for example, v7, or 2k.

**LowerBound**
The value set as the minimum allowable for the specified threshold.

**MSSQLBufferExtMemCacheHitRatio**
(Microsoft SQL Server 2000 only) The percentage of pages accessed in the extended memory. Extended memory is disk memory that is allocated for cache use in addition to the conventional buffer cache. The ratio is calculated as the total number of successful cache hits divided by the total number of requested cache lookups since an instance was started.

**MSSQLServerName**
The instance name of Microsoft SQL Server.

This indication has the following threshold:
- Buffer Extended Memory Cache Hit Ratio lower bound

For more information about this threshold, see “Thresholds” on page 35.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>4</td>
</tr>
<tr>
<td>Holes</td>
<td>2</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Microsoft SQL Server Checkpoint Writes Per Second too high**
Occurs when the number of checkpoint writes per second exceeds the **Checkpoint Writes Per Second upper bound** threshold. A **checkpoint write** is the process of writing modified buffer pages (pages in memory) to disk.

Checkpoints minimize the number of modified pages in memory. A smaller number of modified pages in memory reduces the length of recovery if the server fails.

Checkpoints occur periodically during the following times:
• When a specified number of log records is generated by data modifications
• When requested by a user
• When requested by a system shutdown

By default, this indication occurs when the number of checkpoint writes exceeds 5 per second. Since checkpoints occur periodically, you might see a 0 value for this metric much of the time. During a checkpoint, expect a high I/O rate (perhaps hundreds per second) so that the checkpoint completes as quickly as possible. If you have multiple disks and a fast controller, you can sustain higher I/O rates and shorter checkpoint durations by changing the max async IO option using `sp_configure`.

To limit the amount of consumed resources, Microsoft SQL Server limits checkpoints to a maximum of 100 concurrent write operations. If you want to know how often checkpoints are occurring, and possibly adjust the recovery interval setting, start up Microsoft SQL Server with trace flag 3502. This trace flag writes information to the error log each time a checkpoint occurs.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

```
<application_label>: Microsoft SQL Server <MSSQLServerName> - Checkpoint Writes Per Second is <MSSQLCheckpointWritesPerSec>, which exceeds threshold of <UpperBound>.
```

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- `application_class`
  The registered object’s Tivoli Management Environment (TME) class.
- `application_label`
  The registered object’s Tivoli Management Environment (TME) label.
- `application_oid`
  The registered object’s Tivoli Management Environment (TME) object identifier.
- `application_version`
  The managed resource’s version; for example, v7, or 2k.
- `MSSQLCheckpointWritesPerSec`
  The number of checkpoint writes per second. A checkpoint write is the process of writing modified buffer pages (pages in memory) to disk.
- `MSSQLServerName`
  The instance name of Microsoft SQL Server.
- `UpperBound`
  The value set as the maximum allowable for the specified threshold.

This indication has the following threshold:

- Checkpoint Writes Per Second upper bound

For more information about this threshold, see “Thresholds” on page 35.
The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>5</td>
</tr>
<tr>
<td>Holes</td>
<td>1</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

Note: If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

Microsoft SQL Server CPU Percent Busy too high

Occurs when the percentage of CPU busy exceeds the CPU Percent Busy upper bound threshold. CPU percent busy is the percentage of the total available CPU time that Microsoft SQL Server uses during the current monitoring interval. For example, if the monitoring cycle time is 60 seconds and Microsoft SQL Server uses 6 seconds, then the CPU percent busy is 10 percent. By default, the percentage threshold for CPU busy is 80 percent.

A high CPU usage rate can indicate performance bottlenecks. Consider doing any of the following to lower CPU usage:

• Add multiple processors to the server
• Allocate resources more efficiently (for example, dedicate a computer to run an instance of Microsoft SQL Server)
• Identify resource-intensive applications and optimize those applications
• Reduce the workload on the system by query tuning or improving indexes to reduce I/O and, subsequently, CPU usage
• Upgrade the CPU

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

<application_label>: Microsoft SQL Server <MSSQLServerName> - CPU Percent Busy is <MSSQLCpuPercentBusy>, which exceeds threshold of <UpperBound>.

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

application_class
The registered object’s Tivoli Management Environment (TME) class.

application_label
The registered object’s Tivoli Management Environment (TME) label.
application_oid
The registered object’s Tivoli Management Environment (TME) object identifier.

application_version
The managed resource’s version; for example, v7, or 2k.

MSSQLCpuPercentBusy
The percentage of total CPU time that Microsoft SQL Server has used for the current monitoring interval. For example, if 2500 milliseconds of CPU time has elapsed during a monitoring interval and the Microsoft SQL Server used 500 milliseconds of CPU time, then the total is 20 percent.

MSSQLServerName
The instance name of Microsoft SQL Server.

UpperBound
The value set as the maximum allowable for the specified threshold.

This indication has the following threshold:
• CPU Percent Busy upper bound

For more information about this threshold, see “Thresholds” on page 35.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>3</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

Note: If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

Microsoft SQL Server Free Buffers too low

Note: Available for Microsoft SQL Server version 7 only.

Occurs when the number of cache buffers in the free buffer pool is less than the Free Buffers lower bound threshold. Use this metric to monitor when the number of buffers is getting low. A low number of free buffers can degrade server performance.

If the number of free buffers is low, consider doing one of the following:
• Increase cache size
• Check the disk I/O to ensure buffers are flushing efficiently
• Increase the number in the free buffers configuration option
If the number of free buffers falls below the threshold, you can increase the maximum async io and maximum lazywrite io configuration options, although the lazy writer normally ensures that the value does not fall below the threshold.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

```
<application_label>: Microsoft SQL Server <MSSQLServerName> - Free Buffers is <MSSQLFreeBuffers>, which is below threshold of <LowerBound>.
```

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**
  The registered object’s Tivoli Management Environment (TME) class.

- **application_label**
  The registered object’s Tivoli Management Environment (TME) label.

- **application_oid**
  The registered object’s Tivoli Management Environment (TME) object identifier.

- **application_version**
  The managed resource’s version; for example, v7, or 2k.

- **LowerBound**
  The value set as the minimum allowable for the specified threshold.

- **MSSQLFreeBuffers**
  (Microsoft SQL Server 2000 only) The current number of free buffers available in the free buffer pool.

- **MSSQLServerName**
  The instance name of Microsoft SQL Server.

This indication has the following threshold:

- Free Buffers lower bound

For more information about this threshold, see “Thresholds” on page 35.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>4</td>
</tr>
<tr>
<td>Holes</td>
<td>2</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>
Note: If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

Microsoft SQL Server Lazy Writes Per Second too high
Occurs when the number of lazy writes per second exceeds the Lazy Writes Per Second upper bound threshold. Lazy writes are buffers written by the lazy writer. The lazy writer is a system process that checks the size of the free buffer list. The lazy writer creates available buffers by scanning the buffer cache to reclaim unused pages and write modified pages.

Keeping the number of lazy writes low can enhance performance. A supply of buffers available for immediate use keeps the number of lazy writes low. Before a requested page can be brought into memory, a free buffer must be available in the buffer pool. If no free buffers are available, an existing buffer must be reused. When an existing buffer has to be reused, many buffer pages might have to be searched in order to locate a buffer to reclaim for use. If the buffer found is marked as dirty or modified, the buffer manager must first write the changes to disk before the page can be reused and assigned to the requesting process. This results in a wait for the requesting process. Waiting processes can degrade performance.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

\(<application_label>\): Microsoft SQL Server \(<MSSQLServerName>\) - Lazy Writes Per Second is \(<MSSQLLazyWritesPerSec>\), which exceeds threshold of \(<UpperBound>\).

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- application_class
  The registered object’s Tivoli Management Environment (TME) class.

- application_label
  The registered object’s Tivoli Management Environment (TME) label.

- application_oid
  The registered object’s Tivoli Management Environment (TME) object identifier.

- application_version
  The managed resource’s version; for example, v7, or 2k.

- MSSQLLazyWritesPerSec
  The number of lazy writes per second. Lazy writes are buffers written by the lazy writer. The lazy writer is a system process that checks the size of the free buffer list. The lazy writer creates available buffers by scanning the buffer cache to reclaim unused pages and write modified pages.

- MSSQLServerName
  The instance name of Microsoft SQL Server.
**UpperBound**

The value set as the maximum allowable for the specified threshold.

This indication has the following threshold:

- Lazy Writes Per Second upper bound

For more information about this threshold, see [“Thresholds” on page 35](#).

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>5</td>
</tr>
<tr>
<td>Holes</td>
<td>1</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Microsoft SQL Server Page Reads Per Second too high**

Occurs when the number of page reads per second exceeds the Page Reads Per Second upper bound threshold. A page is a unit of data storage used for allocating database space. In Microsoft SQL Server, a page is 8 kilobytes in size. A page read occurs when a request for a database page results in Microsoft SQL Server transferring the requested page from disk to the buffer pool. Page reads per second are the total number of physical page reads across all databases that are issued per second.

Page reads are slower than cache hits, so they reduce overall performance. When this metric returns high values, attempt to reduce excessive page reads by one or more of the following strategies:

- Increase data cache size
- Revisit the database schema and check for efficiency
- Rewrite queries to be more efficient
- Use intelligent indexes

Another possible solution to reduce excessive page reads is to provide additional memory to Microsoft SQL Server. The more pages that are available in memory, the more likely a page can be found in memory when requested. Microsoft SQL Server allocates most of its virtual memory to a buffer cache and uses the cache to reduce physical I/O.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

```
<application_label>: Microsoft SQL Server <MSSQLServerName> - Page Reads Per Second is <MSSQLPageReadsPerSec>, which exceeds threshold of <UpperBound>.
```

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.
You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

application_class
The registered object’s Tivoli Management Environment (TME) class.

application_label
The registered object’s Tivoli Management Environment (TME) label.

application_oid
The registered object’s Tivoli Management Environment (TME) object identifier.

application_version
The managed resource’s version; for example, v7, or 2k.

MSSQLPageReadsPerSec
The total number of physical page reads across all databases issued per second. A page is a unit of data storage used for allocating database space. In Microsoft SQL Server, a page is 8 kilobytes in size. A page read occurs when a request for a database page results in Microsoft SQL Server having to transfer the requested page from disk to the buffer pool.

MSSQLServerName
The instance name of Microsoft SQL Server.

UpperBound
The value set as the maximum allowable for the specified threshold.

This indication has the following threshold:
• Page Reads Per Second upper bound

For more information about this threshold, see “Thresholds” on page 35.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>2</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

Note: If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

Microsoft SQL Server Page Writes Per Second too high
Occurs when the number of page writes per second exceeds the Page Writes Per Second upper bound threshold. A page is a unit of data storage used for allocating database space. In Microsoft SQL Server, a page is 8 kilobytes in size. A page write occurs when data is modified to a page in memory, and the page is written back to the disk. Page writes per second are the total number of physical writes, including...
checkpoint writes, lazy writer writes, and large block writes done during index creation or bulk copy operations. Use the Microsoft SQL Server Checkpoint Writes Per Second too high and Microsoft SQL Server Lazy Writes Per Second too high indications to specifically monitor checkpoint writes and lazy writes.

Single-page writes are generally expensive, as opposed to batch-wide I/O writes. Reducing single-page-write activity is important for optimal tuning. One way to do this is to ensure the free buffer pool does not run out of free buffers. If it does, single-page writes occur when waiting for an unused cache buffer to flush.

If the number of page writes is high, consider doing any of the following:

- Increase data cache size
- Revisit the database schema and check for efficiency
- Rewrite queries to be more efficient
- Use intelligent indexes

Another possible solution to reduce excessive page reads is to provide additional memory to Microsoft SQL Server. The more pages that are available in memory, the more likely a page can be found in memory when requested. Microsoft SQL Server allocates most of its virtual memory to a buffer cache and uses the cache to reduce physical I/O.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

<application_label>: Microsoft SQL Server <MSSQLServerName> - Page Writes Per Second is <MSSQLPageWritesPerSec>, which exceeds threshold of <UpperBound>.

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**: The registered object’s Tivoli Management Environment (TME) class.
- **application_label**: The registered object’s Tivoli Management Environment (TME) label.
- **application_oid**: The registered object’s Tivoli Management Environment (TME) object identifier.
- **application_version**: The managed resource’s version; for example, v7, or 2k.
- **MSSQLPageWritesPerSec**: The total number of physical writes, including checkpoint writes, lazy writer writes, and large block writes done during index creation or bulk copy operations. A page is a unit of data storage used for allocating database space. In Microsoft SQL Server, a page is 8 kilobytes in size. A page write occurs when data is modified to a page in memory, and the page is written back to the disk.
**MSSQLServerName**

The instance name of Microsoft SQL Server.

**UpperBound**

The value set as the maximum allowable for the specified threshold.

This indication has the following threshold:

- Page Writes Per Second upper bound

For more information about this threshold, see "Thresholds" on page 35.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>2</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Microsoft SQL Server Read Ahead Pages Per Second too high**

Occurs when the number of read ahead pages per second exceeds the **Read Ahead Pages Per Second upper bound** threshold. A page is a unit of data storage used for allocating database space. In Microsoft SQL Server, a page is 8 kilobytes in size. Read ahead pages are pages that are brought into the buffer pool before they are actually requested. **Read ahead pages per second** is the total number of pages read for use per second.

Read-ahead minimizes the number of I/O requests made when data needs to be brought into the buffer pool. If the returned value is high, you might have an opportunity to increase performance by improving disk access speeds. You might also want to review queries to ensure they are not reading more information than required. For example, you might need to rebuild an index to reduce fragmentation, which improves table scans. You also might want to use the metric to see if the read-ahead rate is affected by I/O bottlenecks.

Use this metric to see if the Read Ahead Manager is operating efficiently, or if I/O operations affect its performance. This metric provides a general indication of how busy read-ahead is in terms of actual physical I/O. The Read Ahead Manager reads eight pages per request.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

```
<application_label>: Microsoft SQL Server <MSSQLServerName> - Read Ahead Pages Per Second is <MSSQLReadAheadPagesPerSec>, which exceeds threshold of <UpperBound>.
```

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.
You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**
  The registered object’s Tivoli Management Environment (TME) class.

- **application_label**
  The registered object’s Tivoli Management Environment (TME) label.

- **application_oid**
  The registered object’s Tivoli Management Environment (TME) object identifier.

- **application_version**
  The managed resource’s version; for example, v7, or 2k.

- **MSSQLReadAheadPagesPerSec**
  The total number of pages read for use per second. A page is a unit of data storage used for allocating database space. In Microsoft SQL Server, a page is 8 kilobytes in size. Read ahead pages are pages that are brought into the buffer pool before they are actually requested.

- **MSSQLServerName**
  The instance name of Microsoft SQL Server.

- **UpperBound**
  The value set as the maximum allowable for the specified threshold.

This indication has the following threshold:

- Read Ahead Pages Per Second upper bound

For more information about this threshold, see “Thresholds” on page 35.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>2</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Microsoft SQL Server Stolen Page Count Growth too high**

Occurs when the stolen page count growth increases by an amount equal to or greater than the Stolen Page Count Growth threshold for the current monitoring cycle as compared to prior monitoring cycle. *Stolen pages* are buffer cache pages that are stolen to use for other server memory requests. Stolen pages are used for miscellaneous server purposes such as procedure cache.
When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

<application_label>: Microsoft SQL Server <MSSQLServerName> - Stolen Page Count is <MSSQLStolenPageCount>, an increase of <MSSQLStolenPageCountGrowth>, which equals or exceeds growth threshold of <UpperBound>.

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**: The registered object’s Tivoli Management Environment (TME) class.
- **application_label**: The registered object’s Tivoli Management Environment (TME) label.
- **application_oid**: The registered object’s Tivoli Management Environment (TME) object identifier.
- **application_version**: The managed resource’s version; for example, v7, or 2k.
- **MSSQLServerName**: The instance name of Microsoft SQL Server.
- **MSSQLStolenPageCount**: The number of pages in the buffer cache that were used for server memory requests instead of that for which they were designated.
- **MSSQLStolenPageCountGrowth**: The amount of increase in the number of stolen pages since the last cycle. Stolen page count is the number of pages in the buffer cache that were used for server memory requests instead of that for which they were designated.
- **UpperBound**: The value set as the maximum allowable for the specified threshold.

This indication has the following threshold:

- Stolen Page Count Growth

For more information about this threshold, see “Thresholds” on page 35.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>4</td>
</tr>
<tr>
<td>Holes</td>
<td>2</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>
Note: If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Thresholds**

The following table lists the thresholds that can be set for the Cache/CPU resource model. For each threshold, the table shows the name, a short description, and the default value:

<table>
<thead>
<tr>
<th>Threshold</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffer Cache Hit Ratio lower bound</td>
<td>A Microsoft SQL Server Buffer Cache Hit Ratio too low indication occurs if the buffer cache hit ratio is less than this value.</td>
<td>90</td>
</tr>
<tr>
<td>Buffer Extended Memory Cache Hit Ratio lower bound</td>
<td>A Microsoft SQL Server Buffer Extended Memory Cache Hit Ratio too low indication occurs if the buffer extended memory cache hit ratio is less than this value.</td>
<td>90</td>
</tr>
<tr>
<td>Checkpoint Writes Per Second upper bound</td>
<td>A Microsoft SQL Server Checkpoint Writes Per Second too high indication occurs if the number of checkpoint writes per second exceeds this value.</td>
<td>5</td>
</tr>
<tr>
<td>CPU Percent Busy upper bound</td>
<td>A Microsoft SQL Server CPU Percent Busy too high indication occurs if the percent of busy CPU exceeds this value.</td>
<td>80</td>
</tr>
<tr>
<td>Free Buffers lower bound</td>
<td>A Microsoft SQL Server Free Buffers too low indication occurs if the number of free buffers is less than this value.</td>
<td>75</td>
</tr>
<tr>
<td>Lazy Writes Per Second upper bound</td>
<td>A Microsoft SQL Server Lazy Writes Per Second too high indication occurs if the number of lazy writes per second exceeds this value.</td>
<td>5</td>
</tr>
<tr>
<td>Page Reads Per Second upper bound</td>
<td>A Microsoft SQL Server Page Reads Per Second too high indication occurs if the number of page reads per second exceeds this value.</td>
<td>20</td>
</tr>
<tr>
<td>Page Writes Per Second upper bound</td>
<td>A Microsoft SQL Server Page Writes Per Second too high indication occurs if the number of page writes per second exceeds this value.</td>
<td>20</td>
</tr>
<tr>
<td>Read Ahead Pages Per Second upper bound</td>
<td>A Microsoft SQL Server Read Ahead Pages Per Second too high indication occurs if the number of read ahead pages per second exceeds this value.</td>
<td>50</td>
</tr>
<tr>
<td>Stolen Page Count Growth</td>
<td>A Microsoft SQL Server Stolen Page Count Growth too high indication occurs if the stolen page count growth value increases by this value or more during a monitoring cycle.</td>
<td>1</td>
</tr>
</tbody>
</table>

**Parameters**

None

**Tasks and built-in actions**

None
Logging
You can log data for the properties of the managed resource listed in the following table. The table shows the context of the managed resource and the properties that the resource model logs for the IBM Tivoli Monitoring Web Health Console.

<table>
<thead>
<tr>
<th>Managed resource</th>
<th>Context</th>
<th>Properties</th>
</tr>
</thead>
</table>
| MSSQLServer      | CPU     | MSSQLServer.MSSQLServerName*  
The instance name of Microsoft SQL Server. |
|                  |         | MSSQLVersion  
The version number, v7 or 2k, of the installed Microsoft SQL Server. |
|                  |         | MSSQL_CPU_Percent_Busy  
The percentage of total CPU time that Microsoft SQL Server has used during the monitoring interval. For example, if 2500 milliseconds of CPU time has elapsed during a monitoring interval and the Microsoft SQL Server used 500 milliseconds of CPU time, then the CPU percent busy is 20 percent. |
|                  |         | MSSQLCpuUsageDelta  
The difference (in milliseconds) between MSSQLCpuUsageMS for the current monitoring cycle and MSSQLCpuUsageMS from the prior monitoring cycle. This value is used in calculation of MSSQL_CPU_Percent_Busy. |
|                  |         | MSSQLCpuUsageMS  
The amount of CPU time, in milliseconds, that Microsoft SQL Server has used since Microsoft SQL Server was last started. |
| Memory           |         | MSSQLServer.MSSQLServerName*  
The instance name of Microsoft SQL Server. |
|                  |         | MSSQL_Server_Memory_Used  
The total amount of dynamic memory (in kilobytes) that the Microsoft SQL server is currently using. |

Note: An asterisk (*) denotes a key property.

Return codes
Table 2 on page 7 contains a listing of resource model return codes, their description, and what action you can take to resolve each code. The return code number is displayed in the IBM Tivoli Monitoring Web Health Console status field, or by using the wdmlse command.

CLI example

```
wdmeditprf -P $Prof1 -edit Microsoft_SQL_Server_CacheCpu \
-t Low_MSSQLBufferCacheHitRatio 100.000000 \n-t Low_MSSQLBufferExtMemCacheHitRatio 100.000000 \n-t High_MSSQLCpuPercentBusy 0.000000 \n-t High_MSSQLCheckpointWritesPerSec 0.000000 \n-t Low_MSSQLFreeBuffers 100.000000 \n-t High_MSSQLLazyWritesPerSec 0.000000 \n-t High_MSSQLPageReadsPerSec 0.000000 \n-t High_MSSQLPageWritesPerSec 0.000000 
```
Description

Monitors error log entries and ensures that the server is healthy.

By default, this resource model monitors predefined Errors in the ERRORLOG file for the Microsoft SQL Server being monitored. See Table 4 for a listing of these error numbers. You can customize this resource model by adding or deleting additional error numbers with the Error IDs parameter. Use the Microsoft SQL Server ERRORLOG Event indication to monitor error numbers that you have added to the parameter.

This resource model monitors eight message types in the Microsoft SQL Server ERRORLOG file. The following table details the monitored messages, grouped by the type of message:

Table 4. Error messages monitored by default

<table>
<thead>
<tr>
<th>Error #</th>
<th>Severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allocate Space Failed:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1101</td>
<td>17</td>
<td>Could not allocate new page for database ‘%.*ls’. There are no more pages available in filegroup '%.*ls'. Space can be created by dropping objects, adding additional files, or allowing file growth.</td>
</tr>
<tr>
<td>1105</td>
<td>17</td>
<td>Could not allocate space for object '%.*ls' in database '%.*ls' because the '%.*ls' filegroup is full.</td>
</tr>
<tr>
<td>1703</td>
<td>17</td>
<td>Could not allocate disk space for a work table in database '%.*ls'. You may be able to free up space by using BACKUP LOG, or you may want to extend the size of the database by using ALTER DATABASE.</td>
</tr>
<tr>
<td>1803</td>
<td>17</td>
<td>CREATE DATABASE failed. Could not allocate enough disk space for a new database on the named disks. Total space allocated must be at least %d MB to accommodate a copy of the model database.</td>
</tr>
</tbody>
</table>

Connection Failed:

| 16804   | 11       | SQL Web Assistant: Could not establish a local connection to SQL Server. |
| 18450   | 14       | Login failed for user '%ls'. Reason: Not defined as a valid user of a trusted SQL Server connection. |
| 18452   | 14       | Login failed for user '%ls'. Reason: Not associated with a trusted SQL Server connection. |

Connections High:

| 17809   | 10       | Could not connect. The maximum number of '%1!ld!’ configured user connections are already connected. The system administrator can change the maximum to a higher value using sp_configure. |

Deadlock Alert:

| 1205    | 13       | Transaction (Process ID %d) was deadlocked on [%Z] resources with another process and has been chosen as the deadlock victim. Rerun the transaction. |
| 1211    | 13       | Process ID %d was chosen as the deadlock victim with P_BACKOUT bit set. |
### Table 4. Error messages monitored by default (continued)

<table>
<thead>
<tr>
<th>Error #</th>
<th>Severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3928</td>
<td>16</td>
<td>The marked transaction '%ls' failed. A deadlock was encountered while attempting to place the mark in the log.</td>
</tr>
</tbody>
</table>

**Disk I/O Errors:**

<table>
<thead>
<tr>
<th>Error #</th>
<th>Severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>605</td>
<td>21</td>
<td>Attempt to fetch logical page %PGID in database '%ls' belongs to object '%ls', not to object '%ls'.</td>
</tr>
<tr>
<td>823</td>
<td>24</td>
<td>I/O error %ls detected during %MSG at offset %#016I64x in file '%ls'.</td>
</tr>
</tbody>
</table>

**Errors in Log:**

<table>
<thead>
<tr>
<th>Error #</th>
<th>Severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3159</td>
<td>16</td>
<td>The tail of the log for database '%ls' has not been backed up. Back up the log and rerun the RESTORE statement specifying the FILE clause.</td>
</tr>
<tr>
<td>3163</td>
<td>16</td>
<td>The transaction log was damaged. All data files must be restored before RESTORE LOG can be attempted.</td>
</tr>
<tr>
<td>4329</td>
<td>10</td>
<td>This log file contains records logged before the designated mark. The database is being left in load state so you can apply another log file.</td>
</tr>
<tr>
<td>5032</td>
<td>10</td>
<td>The file cannot be shrunk below page %ud until the log is backed up because it contains bulk logged pages.</td>
</tr>
<tr>
<td>5108</td>
<td>10</td>
<td>Log file '%ls' does not match the primary file. It may be from a different database or the log may have been rebuilt previously.</td>
</tr>
<tr>
<td>9001</td>
<td>10</td>
<td>The log for database '%ls' is not available.</td>
</tr>
<tr>
<td>9006</td>
<td>10</td>
<td>Cannot shrink log file %d (%s) because total number of logical log files cannot be fewer than %d.</td>
</tr>
<tr>
<td>9007</td>
<td>10</td>
<td>Cannot shrink log file %d (%s) because requested size (%dKB) is larger than the start of the last logical log file.</td>
</tr>
<tr>
<td>9008</td>
<td>10</td>
<td>Cannot shrink log file %d (%s) because all logical log files are in use.</td>
</tr>
<tr>
<td>9009</td>
<td>10</td>
<td>Cannot shrink log file %d (%s) because of minimum log space required.</td>
</tr>
</tbody>
</table>

**Full Database:**

<table>
<thead>
<tr>
<th>Error #</th>
<th>Severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5069</td>
<td>16</td>
<td>The transaction log was damaged. All data files must be restored before RESTORE LOG can be attempted.</td>
</tr>
</tbody>
</table>

**Transaction Log Full:**

<table>
<thead>
<tr>
<th>Error #</th>
<th>Severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>9002</td>
<td>19</td>
<td>The log file for database '%ls' is full. Back up the transaction log for the database to free up some log space.</td>
</tr>
</tbody>
</table>

You can specify the ERRORLOG file location with the **ERRORLOG Location** parameter. If you do not specify a location, the resource model checks the registry to find the ERRORLOG file location. The resource model uses the following registry structure to find the location of the errorlog:

- **Microsoft SQL Server version 7 and default instances on Microsoft SQL Server version 2000:**
  
  HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft/MSSQLServer/Parameters

- **Named instances on Microsoft SQL Server version 2000:**
The Parameters directory contains a set of key value pairs with the key being SQLArg0, SQLArg1, SQLArg2, etc. The ERRORLOG setting is preceded with an '-e.'

If the registry entry does not exist for the ERRORLOG file, the resource model uses the default location. The default ERRORLOG file location is the MSSQL_HOME\Log directory of the target installation directory of the Microsoft SQL Server. Typically, this file exists in the following locations:

- For Microsoft SQL Server version 7: C:\MSSQL7\Log\Errorlog
- For Microsoft SQL Server version 2000: C:\Program Files\Microsoft SQL Server\Mssql\Log\Errorlog (for the default instance)
- For a named instance: \Microsoft SQL Server\Mssql$<instancename>\Log\Errorlog

The resource model monitors only the current ERRORLOG file. If the file gets cycled due to the server being stopped and started or by the sp_cycle_errorlog stored procedure, the resource model does not look at the archived version, ERRORLOG.1. If the ERRORLOG file is not found, the resource model is put into an error state.

<table>
<thead>
<tr>
<th>Resource model overview</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal name</td>
</tr>
<tr>
<td>Category</td>
</tr>
<tr>
<td>Indications</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Tasks and built-in actions</td>
</tr>
<tr>
<td>Default cycle time</td>
</tr>
</tbody>
</table>

**Target managed resource**

MicrosoftSQLServer
Indications and events

The following table summarizes the indications for this resource model, the events associated with the indications, the default severity of the events, and where to find a detailed description of the indication. The resource model generates an event when the threshold you specify for the indication is triggered.

<table>
<thead>
<tr>
<th>Indication</th>
<th>Generated event</th>
<th>Default severity</th>
<th>Clearing events</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft SQL Server Allocate Space Failed</td>
<td>Microsoft_SQL_Server_Allocate_Space_Failed</td>
<td>Critical</td>
<td>No</td>
<td>41</td>
</tr>
<tr>
<td>Microsoft SQL Server Connection Failed</td>
<td>Microsoft_SQL_Server_Connection_Failed</td>
<td>Critical</td>
<td>No</td>
<td>43</td>
</tr>
<tr>
<td>Microsoft SQL Server Connections High</td>
<td>Microsoft_SQL_Server_Connections_High</td>
<td>Critical</td>
<td>No</td>
<td>44</td>
</tr>
<tr>
<td>Microsoft SQL Server Deadlock Alert</td>
<td>Microsoft_SQL_Server_Deadlock_Alert</td>
<td>Critical</td>
<td>No</td>
<td>45</td>
</tr>
<tr>
<td>Microsoft SQL Server Disk I/O Error</td>
<td>Microsoft_SQL_Server_Disk_I/O_Error</td>
<td>Critical</td>
<td>No</td>
<td>46</td>
</tr>
<tr>
<td>Microsoft SQL Server ERRORLOG Event</td>
<td>Microsoft_SQL_Server_ERRORLOG_Event</td>
<td>Critical</td>
<td>No</td>
<td>48</td>
</tr>
<tr>
<td>Microsoft SQL Server Errors in Log</td>
<td>Microsoft_SQL_Server_Errors_in_Log</td>
<td>Critical</td>
<td>No</td>
<td>49</td>
</tr>
<tr>
<td>Microsoft SQL Server Full Database</td>
<td>Microsoft_SQL_Server_Full_Database</td>
<td>Critical</td>
<td>No</td>
<td>50</td>
</tr>
<tr>
<td>Microsoft SQL Server Transaction Log Full</td>
<td>Microsoft_SQL_Server_Transaction_Log_Full</td>
<td>Critical</td>
<td>No</td>
<td>52</td>
</tr>
</tbody>
</table>

**Microsoft SQL Server Allocate Space Failed indication**

Occurs when an allocated space error is detected in the monitored error log. Allocated space errors monitored by default include: 1101, 1105, 1703, and 1803. See table 4 on page 38 for a description of these error messages.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

<application_label>: The Microsoft SQL Server <MSSQLServerName> logged error <MSSQLError> in the ERRORLOG at <MSSQLErrorDateTime>. Full message text is - <MSSQLErrorDesc>.

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.
You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**
  - The registered object’s Tivoli Management Environment (TME) class.

- **application_label**
  - The registered object’s Tivoli Management Environment (TME) label.

- **application_oid**
  - The registered object’s Tivoli Management Environment (TME) object identifier.

- **application_version**
  - The managed resource’s version; for example, v7, or 2k.

- **MSSQLServerName**
  - The instance name of Microsoft SQL Server.

- **MSSQLError**
  - The Microsoft SQL Server error number recorded in the error log.

- **MSSQLErrorSev**
  - The number of the Microsoft SQL Server error that indicates the severity of the error condition.

- **MSSQLErrorState**
  - The number that identifies the source from which the error was issued (if the error can be issued from more than one place).

- **MSSQLErrorPID**
  - The system process id of the Microsoft SQL Server process that generated the error. This corresponds to a process id found in the result set of the `sp_who` SQL command.

- **MSSQLErrorDateTime**
  - The date and time that the Microsoft SQL Server error occurred.

- **MSSQLErrorDesc**
  - The description of the problem resulting in the Microsoft SQL Server error.

This indication has no thresholds.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>1</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager.
Microsoft SQL Server Connection Failed indication

Occurs when a failed connection error is detected in the monitored error log. Failed connection errors monitored by default include: 18450, 18452, and 16804. See Table 4 on page 38 for a description of these error messages.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

<application_label>: The Microsoft SQL Server <MSSQLServerName> logged error <MSSQLError> in the ERRORLOG at <MSSQLErrorDateTime>. Full message text is - <MSSQLErrorDesc>.

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- `application_class`
  The registered object’s Tivoli Management Environment (TME) class.

- `application_label`
  The registered object’s Tivoli Management Environment (TME) label.

- `application_oid`
  The registered object’s Tivoli Management Environment (TME) object identifier.

- `application_version`
  The managed resource’s version; for example, v7, or 2k.

- `MSSQLServerName`
  The instance name of Microsoft SQL Server.

- `MSSQLError`
  The Microsoft SQL Server error number recorded in the error log.

- `MSSQLErrorSev`
  The number of the Microsoft SQL Server error that indicates the severity of the error condition.

- `MSSQLErrorState`
  The number that identifies the source from which the error was issued (if the error can be issued from more than one place).

- `MSSQLErrorPID`
  The system process id of the Microsoft SQL Server process that generated the error. This corresponds to a process id found in the result set of the `sp_who` SQL command.

- `MSSQLErrorDateTime`
  The date and time that the Microsoft SQL Server error occurred.

- `MSSQLErrorDesc`
  The description of the problem resulting in the Microsoft SQL Server error.

This indication has no thresholds.
The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>1</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

Note: If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Microsoft SQL Server Connections High indication**

Occurs when a connections high error is detected in the monitored error log. The connections high error monitored by default is error 17809. See [Table 4 on page 38](#) for a description of this error message.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

```
<application_label>: The Microsoft SQL Server <MSSQLServerName> logged error <MSSQLError> in the ERRORLOG at <MSSQLErrorDateTime>. Full message text is - <MSSQLErrorDesc>.
```

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**
  The registered object’s Tivoli Management Environment (TME) class.

- **application_label**
  The registered object’s Tivoli Management Environment (TME) label.

- **application_oid**
  The registered object’s Tivoli Management Environment (TME) object identifier.

- **application_version**
  The managed resource’s version; for example, v7, or 2k.

- **MSSQLServerName**
  The instance name of Microsoft SQL Server.

- **MSSQLError**
  The Microsoft SQL Server error number recorded in the error log.

- **MSSQLErrorDesc**
  The number of the Microsoft SQL Server error that indicates the severity of the error condition.
MSSQLErrorState
The number that identifies the source from which the error was issued (if the error can be issued from more than one place).

MSSQLErrorPID
The system process id of the Microsoft SQL Server process that generated the error. This corresponds to a process id found in the result set of the sp_who SQL command.

MSSQLErrorDateTime
The date and time that the Microsoft SQL Server error occurred.

MSSQLErrorDesc
The description of the problem resulting in the Microsoft SQL Server error.

This indication has no thresholds.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>1</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Microsoft SQL Server Deadlock Alert indication**
Occurs when a deadlock alert error is detected in the monitored error log. Deadlock alert errors monitored by default include errors 1205, 1211, and 3928. See Table 4 on page 38 for a description of these error messages.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

```
<application_label>: The Microsoft SQL Server <MSSQLServerName> logged error <MSSQLError> in the ERRORLOG at <MSSQLErrorDateTime>. Full message text is - <MSSQLErrorDesc>.
```

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**
  The registered object’s Tivoli Management Environment (TME) class.

- **application_label**
  The registered object’s Tivoli Management Environment (TME) label.
application_oid
The registered object’s Tivoli Management Environment (TME) object identifier.

application_version
The managed resource’s version; for example, v7, or 2k.

MSSQLServerName
The instance name of Microsoft SQL Server.

MSSQLError
The Microsoft SQL Server error number recorded in the error log.

MSSQLErrorSev
The number of the Microsoft SQL Server error that indicates the severity of the error condition.

MSSQLErrorState
The number that identifies the source from which the error was issued (if the error can be issued from more than one place).

MSSQLErrorPID
The system process id of the Microsoft SQL Server process that generated the error. This corresponds to a process id found in the result set of the sp_who SQL command.

MSSQLErrorDateTime
The date and time that the Microsoft SQL Server error occurred.

MSSQLErrorDesc
The description of the problem resulting in the Microsoft SQL Server error.

This indication has no thresholds.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>1</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Microsoft SQL Server Disk I/O Error indication**
Occurs when a server disk I/O error is detected in the monitored error log. Disk I/O errors monitored by default include errors 605, and 823. See Table 4 on page 38 for a description of these error messages.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:
<application_label>: The Microsoft SQL Server <MSSQLServerName> logged error <MSSQLError> in the ERRORLOG at <MSSQLErrorDateTime>. Full message text is - <MSSQLErrorDesc>.

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

application_class
The registered object’s Tivoli Management Environment (TME) class.

application_label
The registered object’s Tivoli Management Environment (TME) label.

application_oid
The registered object’s Tivoli Management Environment (TME) object identifier.

application_version
The managed resource’s version; for example, v7, or 2k.

MSSQLServerName
The instance name of Microsoft SQL Server.

MSSQLError
The Microsoft SQL Server error number recorded in the error log.

MSSQLErrorSev
The number of the Microsoft SQL Server error that indicates the severity of the error condition.

MSSQLErrorState
The number that identifies the source from which the error was issued (if the error can be issued from more than one place).

MSSQLErrorPID
The system process id of the Microsoft SQL Server process that generated the error. This corresponds to a process id found in the result set of the sp_who SQL command.

MSSQLErrorDateTime
The date and time that the Microsoft SQL Server error occurred.

MSSQLErrorDesc
The description of the problem resulting in the Microsoft SQL Server error.

This indication has no thresholds.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>1</td>
</tr>
</tbody>
</table>
Setting | Default value
---|---
Holes | 0
Associated tasks and built-in actions | None

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Microsoft SQL Server ERRORLOG Event indication**
Occurs when a user-specified error is detected in the monitored error log. Use the **Error IDs** parameter to add specific error numbers to monitor. This indication occurs if the specified error is detected.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

```
<application_label>: The Microsoft SQL Server <MSSQLServerName> logged error <MSSQLError> in the ERRORLOG at <MSSQLErrorDateTime>. Full message text is - <MSSQLErrorDesc>.
```

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**
  The registered object’s Tivoli Management Environment (TME) class.

- **application_label**
  The registered object’s Tivoli Management Environment (TME) label.

- **application_oid**
  The registered object’s Tivoli Management Environment (TME) object identifier.

- **application_version**
  The managed resource’s version; for example, v7, or 2k.

- **MSSQLServerName**
  The instance name of Microsoft SQL Server.

- **MSSQLError**
  The Microsoft SQL Server error number recorded in the error log.

- **MSSQLErrorSev**
  The number of the Microsoft SQL Server error that indicates the severity of the error condition.

- **MSSQLErrorState**
  The number that identifies the source from which the error was issued (if the error can be issued from more than one place).
**MSSQLErrorPID**
The system process id of the Microsoft SQL Server process that generated the error. This corresponds to a process id found in the result set of the `sp_who` SQL command.

**MSSQLErrorDateTime**
The date and time that the Microsoft SQL Server error occurred.

**MSSQLErrorDesc**
The description of the problem resulting in the Microsoft SQL Server error.

This indication has no thresholds.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems</td>
<td>No</td>
</tr>
<tr>
<td>Manager</td>
<td></td>
</tr>
<tr>
<td>Occurrences</td>
<td>1</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Microsoft SQL Server Errors in Log indication**
Occurs when an Errors in Log error is detected in the monitored error log. Errors in Log errors monitored by default include: 5032, 4329, 5108, 9001, 9006, 9007, 9008, 9009, 3159, and 3163. See Table 4 on page 38 for a description of these error messages.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

```<application_label>: The Microsoft SQL Server <MSSQLServerName> logged error <MSSQLError> in the ERRORLOG at <MSSQLErrorDateTime>. Full message text is - <MSSQLErrorDesc>.```

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**
  The registered object’s Tivoli Management Environment (TME) class.

- **application_label**
  The registered object’s Tivoli Management Environment (TME) label.
application_oid
The registered object’s Tivoli Management Environment (TME) object identifier.

application_version
The managed resource’s version; for example, v7, or 2k.

MSSQLServerName
The instance name of Microsoft SQL Server.

MSSQLErrorCode
The Microsoft SQL Server error number recorded in the error log.

MSSQLErrorCodeSev
The number of the Microsoft SQL Server error that indicates the severity of the error condition.

MSSQLErrorCodeState
The number that identifies the source from which the error was issued (if the error can be issued from more than one place).

MSSQLErrorCodePID
The system process id of the Microsoft SQL Server process that generated the error. This corresponds to a process id found in the result set of the sp_who SQL command.

MSSQLErrorCodeDateTime
The date and time that the Microsoft SQL Server error occurred.

MSSQLErrorCodeDesc
The description of the problem resulting in the Microsoft SQL Server error.

This indication has no thresholds.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>1</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Microsoft SQL Server Full Database indication**
Occurs when a Full Database error is detected in the monitored error log. The Full Database error monitored by default is 5069. See Table 4 on page 38 for a description of this error message.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:
The Microsoft SQL Server `<MSSQLServerName>` logged error `<MSSQLError>` in the ERRORLOG at `<MSSQLErrorDateTime>`. Full message text is - `<MSSQLErrorDesc>`.

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**: The registered object’s Tivoli Management Environment (TME) class.
- **application_label**: The registered object’s Tivoli Management Environment (TME) label.
- **application_oid**: The registered object’s Tivoli Management Environment (TME) object identifier.
- **application_version**: The managed resource’s version; for example, v7, or 2k.
- **MSSQLServerName**: The instance name of Microsoft SQL Server.
- **MSSQLError**: The Microsoft SQL Server error number recorded in the error log.
- **MSSQLErrorSev**: The number of the Microsoft SQL Server error that indicates the severity of the error condition.
- **MSSQLErrorState**: The number that identifies the source from which the error was issued (if the error can be issued from more than one place).
- **MSSQLErrorPID**: The system process id of the Microsoft SQL Server process that generated the error. This corresponds to a process id found in the result set of the `sp_who` SQL command.
- **MSSQLErrorDateTime**: The date and time that the Microsoft SQL Server error occurred.
- **MSSQLErrorDesc**: The description of the problem resulting in the Microsoft SQL Server error.

This indication has no thresholds.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>1</td>
</tr>
</tbody>
</table>
### Microsoft SQL Server Transaction Log Full indication

Occurs when a Transaction Log Full error is detected in the monitored error log. The Transaction Log Full error monitored by default is 9002. See Table 4 on page 38 for a description of this error message.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

```<application_label>': The Microsoft SQL Server `<MSSQLServerName> logged error `<MSSQLError>` in the ERRORLOG at `<MSSQLerrorDateTime>`. Full message text is - `<MSSQLErrorDesc>`.

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**
  The registered object’s Tivoli Management Environment (TME) class.

- **application_label**
  The registered object’s Tivoli Management Environment (TME) label.

- **application_oid**
  The registered object’s Tivoli Management Environment (TME) object identifier.

- **application_version**
  The managed resource’s version; for example, v7, or 2k.

- **MSSQLServerName**
  The instance name of Microsoft SQL Server.

- **MSSQLError**
  The Microsoft SQL Server error number recorded in the error log.

- **MSSQLErrorSev**
  The number of the Microsoft SQL Server error that indicates the severity of the error condition.

- **MSSQLErrorState**
  The number that identifies the source from which the error was issued (if the error can be issued from more than one place).
**MSSQLErrorPID**

The system process id of the Microsoft SQL Server process that generated the error. This corresponds to a process id found in the result set of the `sp_who` SQL command.

**MSSQLErrorDateTime**

The date and time that the Microsoft SQL Server error occurred.

**MSSQLErrorDesc**

The description of the problem resulting in the Microsoft SQL Server error.

This indication has no thresholds.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>1</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Thresholds**

None

**Parameters**

The following table lists the parameters that can be set for the Errors resource model. The table shows the name, a short description, and the default value for each parameter:
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error IDs</td>
<td>Lists the error IDs to be monitored in the ERRORLOG. See Table 4 on page 38 for a description of these default error messages. You can add additional error IDs to this parameter list. User-specified error IDs generate the Microsoft SQL Server ERRORLOG Event indication when the specified error is detected. This indication provides all the related detail information regarding the error. You can also delete error IDs to exclude them from monitoring.</td>
<td>605, 823, 1101, 1105, 1205, 1211, 1703, 1803, 3159, 3163, 3928, 4329, 5032, 5069, 5108, 9001, 9002, 9006, 9007, 9008, 9009, 16804, 17809, 18450, 18452</td>
</tr>
<tr>
<td>ERRORLOG Location</td>
<td>Specifies the location of the ERRORLOG file relative to the Microsoft SQL Server’s home directory. By default, the location is blank. You can specify the fully-qualified path to the ERRORLOG file. If you do not specify a location, the resource model searches to determine the default location of the ERRORLOG file.</td>
<td>Blank</td>
</tr>
</tbody>
</table>

**Tasks and built-in actions**

None

**Logging**

You can log data for the properties of the managed resource listed in the following table. The table shows the context of the managed resource and the properties that the resource model logs for the IBM Tivoli Monitoring Web Health Console.
<table>
<thead>
<tr>
<th>Managed resource</th>
<th>Context</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSSQLServer</td>
<td>State</td>
<td>MSSQLServer.MSSQLServerName*&lt;br&gt;The instance name of Microsoft SQL Server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_Error_Count_Allocate_Space_Failed&lt;br&gt;The number of Allocate Space Failed errors in the ERRORLOG for the monitoring interval.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_Error_Count_Connections_High&lt;br&gt;The number of Connections High errors in the ERRORLOG for the monitoring interval.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_Error_Count_Connections_Failed&lt;br&gt;The number of Connection Failed errors in the ERRORLOG for the monitoring interval.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_Error_Count_Customer_Defined&lt;br&gt;The number of Customer Defined errors in the ERRORLOG for the monitoring interval.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_Error_Count_Database_Full&lt;br&gt;The number of Database Full errors in the ERRORLOG for the monitoring interval.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_Error_Count_Deadlock_Alert&lt;br&gt;The number of Deadlock Alert errors in the ERRORLOG for the monitoring interval.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_Error_Count_Disk_IO_Errors&lt;br&gt;The number of Disk IO errors in the ERRORLOG for the monitoring interval.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_Error_Count_Transaction_Log_Errors&lt;br&gt;The number of Transaction Log errors in the ERRORLOG for the monitoring interval.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_Error_Count_Transaction_Log_Full&lt;br&gt;The number of Transaction Log Full errors in the ERRORLOG for the monitoring interval.</td>
</tr>
</tbody>
</table>

Note: An asterisk (*) denotes a key property.

**Return codes**

Table 2 on page 7 contains a listing of resource model return codes, their description, and what action you can take to resolve each code. The return code number is displayed in the IBM Tivoli Monitoring Web Health Console status field, or by using the `wdmlseng` command.

**CLI example**

```
wdmeditprf -P $Prof1 -edit Microsoft_SQL_Server_Errors
- e Microsoft_SQL_Server_Connections_High
 - o 1 - h 0 - severity CRITICAL - SendTBSM "$MyTec"
 - e Microsoft_SQL_Server_Full_Database
 - o 1 - h 0 - severity CRITICAL - SendTBSM "$MyTec"
 - e Microsoft_SQL_Server_Disk_IO_Error
 - o 1 - h 0 - severity CRITICAL - SendTBSM "$MyTec"
 - e Microsoft_SQL_Server_Errs_in_Log
 - o 1 - h 0 - severity CRITICAL - SendTBSM "$MyTec"
 - e Microsoft_SQL_Server_Transaction_Log_Full
 - o 1 - h 0 - severity CRITICAL - SendTBSM "$MyTec"
 - e Microsoft_SQL_Server_Cnctn_Failed
 - o 1 - h 0 - severity CRITICAL - SendTBSM "$MyTec"
```
-e Microsoft_SQL_Server_Deadlock_Alert \
  -o 1 -h 0 -severity CRITICAL -SendTBSM -"$MyTec" \
-e Microsoft_SQL_Server_ERRORLOG_Event \
  -o 1 -h 0 -severity CRITICAL -SendTBSM -"$MyTec" \
-e Microsoft_SQL_Server_Allocate_Space_Failed \
  -o 1 -h 0 -severity CRITICAL -SendTBSM -"$MyTec"
-AddPar Error_IDs "testdb"
-AddPar ERRORLOG_Location "c:\ErrorLog.txt"
**Description**

Monitors status and run duration for non-replication jobs running on Microsoft SQL Server. Use this resource model to ensure that scheduled jobs are running properly. This resource model monitors the following job statistics:

- Cancelled jobs
- Failed jobs
- Not run jobs
- Excessive run durations

A job is a series of operations, or steps, that are performed automatically by a SQL Server Agent. You can use jobs to automate administrative tasks and run them on a recurring basis or in response to schedules and alerts. Jobs can be run on the local instance of Microsoft SQL Server or on multiple servers. To run jobs on multiple servers, you must set up at least one master server and one or more target servers. Anyone can create a job, but a job can be edited only by its owner or by those assigned the System Administrator role.

By default, this resource model monitors all non-replication jobs. You can customize this resource model to exclude specific jobs from monitoring by using the **Ignore Jobs** parameter.

All jobs are scheduled to run by default. To prevent a scheduled job from running, you must disable the schedule. The job can still execute in response to an alert or when a user runs the job manually. SQL Server Agent automatically disables schedules that are no longer current. When this occurs, schedules must be reenabled to run.

You can customize this resource model to monitor a specific job for a specified duration time by using the **Jobs to Monitor for Specific Threshold Values** parameter. Job duration times specified with this parameter overwrite the **Default Duration Threshold** value.

**Resource model overview**

<table>
<thead>
<tr>
<th>Internal name</th>
<th>Microsoft_SQL_Server_Jobs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>Microsoft SQL Server</td>
</tr>
<tr>
<td>Indications</td>
<td><strong>Microsoft SQL Server Cancelled Job</strong> occurs when the resource model detects a cancelled status for a monitored job.</td>
</tr>
<tr>
<td></td>
<td><strong>Microsoft SQL Server Failed Job</strong> occurs when the resource model detects a failed job status.</td>
</tr>
<tr>
<td></td>
<td><strong>Microsoft SQL Server Job Duration Threshold</strong> occurs when a job duration exceeds the specified threshold.</td>
</tr>
<tr>
<td></td>
<td><strong>Microsoft SQL Server Job Not Run</strong> occurs when the resource model detects a job Not Run status for three consecutive cycles of the resource model.</td>
</tr>
<tr>
<td>Tasks and built-in actions</td>
<td>None</td>
</tr>
<tr>
<td>Default cycle time</td>
<td>300 seconds (5 minutes)</td>
</tr>
</tbody>
</table>
Target managed resource
MicrosoftSQLServer

Indications and events

The following table summarizes the indications for this resource model, the events associated with the indications, the default severity of the events, and where to find a detailed description of the indication. The resource model generates an event when the threshold you specify for the indication is triggered.

<table>
<thead>
<tr>
<th>Indication</th>
<th>Generated event</th>
<th>Default severity</th>
<th>Clearing events</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft SQL Server Cancelled Job</td>
<td>Microsoft_SQL_Server_Cancelled_Job</td>
<td>Critical</td>
<td>Yes</td>
<td>58</td>
</tr>
<tr>
<td>Microsoft SQL ServerFailed Job</td>
<td>Microsoft_SQL_Server_Failed_Job</td>
<td>Critical</td>
<td>Yes</td>
<td>59</td>
</tr>
<tr>
<td>Microsoft SQL Server Job Duration Threshold</td>
<td>Microsoft_SQL_Server_Job_Duration_Threshold</td>
<td>Minor</td>
<td>No</td>
<td>60</td>
</tr>
<tr>
<td>Microsoft SQL Server Job Not Run</td>
<td>Microsoft_SQL_Server_Job_Not_Run</td>
<td>Warning</td>
<td>Yes</td>
<td>61</td>
</tr>
</tbody>
</table>

**Microsoft SQL Server Cancelled Job indication**

Occurs when the resource model detects a cancelled status for a monitored job. A cancelled job is a job that has been cancelled by a Microsoft SQL Server administrator. Use this indication to alert you when a job has been cancelled.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

\<application label\>: Microsoft SQL Server Cancelled Job: Job Name: \<MSSQLJobName\> Server Name: \<MSSQLServerName\> Job Duration: \<MSSQLJobDuration\>

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**
  The registered object’s Tivoli Management Environment (TME) class.

- **application_label**
  The registered object’s Tivoli Management Environment (TME) label.

- **application_oid**
  The registered object’s Tivoli Management Environment (TME) object identifier.

- **application_version**
  The managed resource’s version; for example, v7, or 2k.
**MSSQLJobDuration**  
Numeric value indicating the length of time in seconds that the job ran before cancellation.

**MSSQLJobName**  
The name of the Microsoft SQL Server job currently monitored by the resource model.

**MSSQLJobStatus**  
The status of the job. In this case, the status is Cancelled.

**MSSQLServerName**  
The name of the Microsoft SQL Server containing the monitored job.

This indication has no thresholds.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>1</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Microsoft SQL Server Failed Job indication**  
Occurs when the resource model detects a failed job status. A failed job is a job that generates an error when Microsoft SQL Server attempts execution. These jobs include user-defined jobs, such as maintenance or backup tasks. Check the alert log or trace files for error information about a failed job.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

```
<application_label>: Microsoft SQL Server Failed Job: Job Name: <MSSQLJobName>  
MSSQL Server Name: <MSSQLServerName> Job Duration: <MSSQLJobDuration>
```

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**  
The registered object’s Tivoli Management Environment (TME) class.

- **application_label**  
The registered object’s Tivoli Management Environment (TME) label.
application_oid
The registered object’s Tivoli Management Environment (TME) object identifier.

application_version
The managed resource’s version; for example, v7, or 2k.

MSSQLJobDuration
Numeric value indicating the length of time in seconds that the job ran before failing.

MSSQLJobName
The name of the Microsoft SQL Server job currently monitored by the resource model.

MSSQLJobStatus
The status of the job. In this case, the status is Failed.

MSSQLServerName
The name of the Microsoft SQL Server containing the monitored job.

This indication has no thresholds.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>1</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Microsoft SQL Server Job Duration Threshold indication**
Occurs when a job duration exceeds the specified threshold. Use this indication to monitor the efficiency of job executions. If jobs are exceeding the duration threshold, review your system performance. You can set specific duration threshold times for specific jobs with the Jobs To Monitor for Specific Threshold Values parameter. See “Parameters” on page 63 for more information about using this parameter to meet your monitoring needs.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

<application_label> : Microsoft SQL Server Job Duration Threshold Exceeded:
Job Name - <MSSQLJobName>, MSSQL Server Name - <MSSQLServerName>,
Job Duration - <MSSQLJobDuration>, Threshold - <MSSQLJobDurationThreshold>

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.
You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**
  The registered object’s Tivoli Management Environment (TME) class.

- **application_label**
  The registered object’s Tivoli Management Environment (TME) label.

- **application_oid**
  The registered object’s Tivoli Management Environment (TME) object identifier.

- **application_version**
  The managed resource’s version; for example, v7, or 2k.

- **JobStatNum**
  The numeric status of the job as returned by the provider.

- **MSSQLJobDuration**
  Numeric value indicating the length of time in seconds that the job ran.

- **MSSQLJobDurationThreshold**
  Numeric value indicating the defined length of time in seconds set for the threshold.

- **MSSQLJobName**
  The name of the Microsoft SQL Server job currently monitored by the resource model.

- **MSSQLServerName**
  The name of the Microsoft SQL Server containing the monitored job.

This indication has the following threshold:

- **Default Duration Threshold**
  For more information about this threshold, see “Thresholds” on page 63.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>1</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Microsoft SQL Server Job Not Run indication**

Occurs when the resource model detects a job Not Run status for three consecutive cycles of the resource model. A Not Run job is a job for which execution was not
attempted by Microsoft SQL Server. Use this indication to alert you when a job has had a Not Run status for three consecutive cycles of the resource model. You can change the Not Run notification default (three consecutive cycles) by changing the occurrences value for this indication.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

\[<application_label>: Microsoft SQL Server Job Not Run: <MSSQLJobName> on Server: <MSSQLServerName> has not run during multiple cycles of this resource model.\]

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**: The managed resource’s version; for example, v7, or 2k.
- **application_label**: The registered object’s Tivoli Management Environment (TME) label.
- **application_oid**: The registered object’s Tivoli Management Environment (TME) object identifier.
- **application_version**: The managed resource’s version; for example, v7, or 2k.
- **MSSQLJobName**: The name of the Microsoft SQL Server job currently monitored by the resource model.
- **MSSQLJobStatus**: The status of the job. In this case, the status is Not Run.
- **MSSQLServerName**: The name of the Microsoft SQL Server containing the monitored job.

This indication has no thresholds.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>3</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note**: If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager.
Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Thresholds**

The following table lists the thresholds that can be set for the Jobs resource model. For each threshold it shows the name, a short description, and the default value:

<table>
<thead>
<tr>
<th>Threshold</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Duration Threshold</td>
<td>A Microsoft SQL Server Job Duration Threshold indication occurs when the job duration time exceeds this value.</td>
<td>1000 (units = seconds)</td>
</tr>
</tbody>
</table>

**Parameters**

The following table lists the parameter that can be set for the Jobs resource model. The table shows the name, a short description, and the default value for each parameter:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ignored Jobs</td>
<td>Type any non-replication job names to exclude from monitoring. When this parameter is blank, all jobs are monitored. The wildcard character (<em>) is supported. The following examples illustrate how to code your parameter: • To match a single job such as &quot;XYZ,&quot; use the following expression: XYZ • To match a job that begins with &quot;ABC,&quot; use the following expression: ABC</em> • To match a job that contains &quot;DEF,&quot; use the following expression: <em>DEF</em> • To match a job that begins with &quot;A&quot; and ends with &quot;Z,&quot; use the following expression: A*Z</td>
<td>Blank (monitors all non-replication jobs)</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Default value</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Jobs To Monitor for Specific Threshold Values</td>
<td>Use this parameter to associate a job duration threshold for a specific job or groups of jobs. This parameter overwrites the Default Duration Threshold for a specific job. The wildcard (<em>) character is supported. Syntax: (&lt;job \text{ name}&gt;=&lt;duration \text{ threshold}&gt;) where: (&lt;job \text{ name}&gt;) is the name of a job for which you want to change the default duration threshold. (&lt;duration \text{ threshold}&gt;) is the numeric value in seconds specifying the maximum allowable duration time for the job. Example: Myjob=125 The following examples illustrate how to code your parameter: • To match a single job such as &quot;XYZ,&quot; use the following expression: XYZ • To match a job that begins with &quot;ABC,&quot; use the following expression: ABC</em> • To match a job that contains &quot;DEF,&quot; use the following expression: <em>DEF</em> • To match a job that begins with &quot;A&quot; and ends with &quot;Z,&quot; use the following expression: A*Z Note: The &quot;</td>
<td>&quot; character is not supported in these expressions.</td>
</tr>
<tr>
<td>Refresh Monitored Jobs Frequency</td>
<td>Defines how often the resource model should get a new list of job names from the Microsoft SQL Server. The value represents the number of resource model cycles. Consider the cycle time of the resource model when setting this parameter. For example, if the resource model cycle time is 15 minutes, setting this parameter to a value of 5 tells the resource model to obtain a new list of job names from the Microsoft SQL Server every 75 minutes.</td>
<td>5</td>
</tr>
</tbody>
</table>

**Tasks and built-in actions**

None
Logging

You can log data for the properties of the managed resource listed in the following table. The following table shows the context of the managed resource and the properties that the resource model logs for the IBM Tivoli Monitoring Web Health Console.

<table>
<thead>
<tr>
<th>Managed resource</th>
<th>Context</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSSQLJobCategory</td>
<td>Status</td>
<td>MSSQLServer.MSSQLServerName* The name of the server that contains the monitored job.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQLJobCategory.MSSQLJobCategory* The job category of the specified Microsoft SQL Server job.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_Job_Cancelled Specifies whether the specified job is cancelled. A returned status of &quot;1&quot; indicates the job is Cancelled. A returned status of &quot;0&quot; indicates a status other than Cancelled.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_Job_Duration Numeric value indicating the length of time in seconds that the job ran.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_Job_Failed Specifies whether the specified job failed. A returned status of &quot;1&quot; indicates the job failed. A returned status of &quot;0&quot; indicates a status other than failed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_Job_In_Progress Specifies whether the specified job is in progress. A returned status of &quot;1&quot; indicates the job is In Progress. A returned status of &quot;0&quot; indicates a status other than In Progress.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_Job_Not_Run Specifies whether the specified job has a status of Not Run. A returned status of &quot;1&quot; indicates the job is Not Run. A returned status of &quot;0&quot; indicates a status other than Not Run.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_Job_Retry Specifies whether the specified job has a status of Retry. A returned status of &quot;1&quot; indicates the job is in a Retry state. A returned status of &quot;0&quot; indicates a status other than Retry.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_Job_Succeeded Specifies whether the specified job has succeeded. A returned status of &quot;1&quot; indicates the job status is Succeeded. A returned status of &quot;0&quot; indicates a status other than Succeeded.</td>
</tr>
</tbody>
</table>

Note: An asterisk (*) denotes a key property.
Return codes

Table 2 on page 7 contains a listing of resource model return codes, their description, and what action you can take to resolve each code. The return code number is displayed in the IBM Tivoli Monitoring Web Health Console status field, or by using the `wdmlsend` command.

CLI example

```
wdmeditprf -P $Prof1 -edit Microsoft_SQL_Server_Jobs \
  -t defaultDurationThresh 1000.000000 \
  -AddPar JobsToMonitorDurThresh "T1Test" \
  -AddPar JobsToMonitorDurThresh "SleepJob=10" \
  -e Microsoft_SQL_Server_Technical_Job \
    -o 1 -h 0 -severity CRITICAL -NoSendTBSM -"SendTec" \
  -e Microsoft_SQL_Server_Cancelled_Job \
    -o 1 -h 0 -severity CRITICAL -NoSendTBSM -"SendTec" \
  -e Microsoft_SQL_Server_Job_Duration_Threshold \
    -o 1 -h 0 -severity CRITICAL -NoSendTBSM -"SendTec" \
  -e Microsoft_SQL_Server_Job_Not_Run \
    -o 3 -h 0 -severity CRITICAL -NoSendTBSM -"SendTec"
```
Locks

Description

Monitors locks in the Microsoft SQL Server to ensure accessibility of databases, tables, and records.

A lock is the means by which integrity of data is protected by preventing more than one user from accessing or changing the same data or object at the same time. Locks prevent users from reading data undergoing changes by other users, and prevent multiple users from changing the same data at the same time. Microsoft SQL Server enforces locking automatically.

Use this resource model to monitor high wait times for resource access and large number of deadlock occurrences. High occurrences of lock wait time and number of deadlocks can indicate poor performance of Microsoft SQL Server.

This resource model monitors the following lock resources:

- **Row ID (RID)**
  - Protects a single row in a table by specifying the row identifier.

- **Key (KEY)**
  - Protects a row within an index. Protects key ranges for records in a table and ensures serializable transactions.

- **Page (PAG)**
  - Protects an eight-kilobyte (KB) data page or index page.

- **Extent (EXT)**
  - Protects a contiguous group of eight data pages or index pages. An extent is an allocated unit of space. Extent locks are used when space needs to be allocated. SQL statements based on CREATE, DROP, and INSERT require this type of lock to be taken out.

- **Table (TAB)**
  - Protects an entire table, including all data and indexes.

- **Database (DB)**
  - Protects entire databases.

You can use the `sp_lock` system stored procedure to view a report of current active locks.

<table>
<thead>
<tr>
<th>Resource model overview</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal name</td>
</tr>
<tr>
<td>Category</td>
</tr>
<tr>
<td>Indications</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Tasks and built-in actions</td>
</tr>
<tr>
<td>Default cycle time</td>
</tr>
</tbody>
</table>
**Target managed resource**

Microsoft SQL Server

**Indications and events**

The following table summarizes the indications for this resource model, the events associated with the indications, the default severity of the events, and where to find a detailed description of the indication. The resource model generates an event when the threshold you specify for the indication is triggered.

<table>
<thead>
<tr>
<th>Indication</th>
<th>Generated event</th>
<th>Default severity</th>
<th>Clearing events</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft SQL Server Lock Wait Time Per Second too high</td>
<td>Microsoft_SQL_Server_High_LockWaitTimePerSec</td>
<td>Warning</td>
<td>Yes</td>
<td>68</td>
</tr>
<tr>
<td>Microsoft SQL Server Number Deadlocks Per Second too high</td>
<td>Microsoft_SQL_Server_High_NumberDeadlocksPerSec</td>
<td>Critical</td>
<td>Yes</td>
<td>69</td>
</tr>
</tbody>
</table>

**Microsoft SQL Server Lock Wait Time Per Second too high indication**

Occurs when the lock wait time per second exceeds the Lock Wait Time (ms) Per Second upper bound threshold. The rate is expressed as milliseconds per second. Lock wait time is the time that a process spends waiting for another process to release a lock. This indication focuses on the average total wait time (in milliseconds) for locks per second during the monitoring interval.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

`<application_label>`: Microsoft SQL Server `<MSSQLServerName>` - `<MSSQLResourceType>`: Lock Wait Time is `<MSSQLLockWaitTime>` milliseconds per second, which exceeds threshold of `<UpperBound>`.

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- `application_class`
  - The registered object’s Tivoli Management Environment (TME) class.

- `application_label`
  - The registered object’s Tivoli Management Environment (TME) label.

- `application_oid`
  - The registered object’s Tivoli Management Environment (TME) object identifier.

- `application_version`
  - The managed resource’s version; for example, v7, or 2k.
MSSQLLockWaitTime
The total wait time in milliseconds that a process spends waiting for another process to release a lock.

MSSQLResourceType
The type of resource with which the lock is associated (Page, Table, etc.).

MSSQLServerName
The instance name of Microsoft SQL Server.

UpperBound
The value set as the maximum allowable for the specified threshold.

This indication has the following threshold:
• Lock Wait Time (ms) Per Second upper bound

For more information about this threshold, see “Thresholds” on page 71

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>1</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Microsoft SQL Server Number Deadlocks Per Second too high indication**

Occurs when the number of deadlocks per second exceeds the **Number Deadlocks Per Second upper bound** threshold. A deadlock occurs when two processes, each having locked a resource, request access to put a lock on the other’s locked resource. Since each process is waiting for the other to release their lock, the processes wait indefinitely until one of the processes is killed. Microsoft SQL Server automatically resolves deadlocks by rolling back one of the processes to release the lock. The process that is rolled back is the “victim” since the system kills it to resolve the deadlock.

The resource model monitors the number of deadlocks that have occurred during the current monitoring interval. This indication focuses on the number of lock requests per second that resulted in a deadlock. Use this indication to alert you to a high number of deadlocks occurring, which ensures that you do not lose any transactions.

**Note:** Microsoft SQL Server error 1205 is a deadlock error. Because any process can be chosen as the deadlock victim, use an error handler to trap error message 1205. If the error is not captured, processes can proceed unaware that the transaction was rolled back. This can result in errors. You can use the **NT Event Logging** task to log this error in the Windows NT event log. You can...
also use the [“Microsoft SQL Server Deadlock Alert indication” on page 45](#) in the Errors resource model to monitor occurrences of deadlocks.

A traditional wait for a lock is not a deadlock. Typically, when a process that is holding a lock completes a transaction, it releases the lock. The next process waiting for the resource receives the lock. This type of wait is normal and required for multiple-use systems.

Consider doing one of the following to minimize the number of deadlocks:
- Stagger competing processes.
- Alter the order in which competing processes run.
- Rewrite queries that cause deadlocks.
- Set the deadlock priority. Encourage SQL Server to choose specific processes to kill by setting the DEADLOCK_PRIORITY of a session to LOW, using the SET statement. If a session’s setting is LOW, that session is the preferred victim when involved in a deadlock situation.

You can also code applications to resubmit processes that are killed as a result of a deadlock.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

```
<application_label>: Microsoft SQL Server <MSSQLServerName> - <MSSQLResourceType>: Number Deadlocks Per Second is <MSSQLNumberDeadlocksPerSec>, which exceeds threshold of <UpperBound>.
```

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**
  The registered object’s Tivoli Management Environment (TME) class.

- **application_label**
  The registered object’s Tivoli Management Environment (TME) label.

- **application_oid**
  The registered object’s Tivoli Management Environment (TME) object identifier.

- **application_version**
  The managed resource’s version; for example, v7, or 2k.

- **MSSQLNumberDeadlocksPerSec**
  The number of lock requests per second that resulted in a deadlock. A deadlock occurs when two processes, each having locked a resource, request access to put a lock on the other’s locked resource.

- **MSSQLResourceType**
  The type of resource with which the lock is associated (Page, Table, etc.).

- **MSSQLServerName**
  The instance name of Microsoft SQL Server.
UpperBound

The value set as the maximum allowable for the specified threshold.

This indication has the following threshold:

- Number Deadlocks Per Second upper bound

For more information about this threshold, see “Thresholds” on page 71.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>1</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Thresholds**

The following table lists the thresholds that can be set for the Locks resource model. For each threshold it shows the name, a short description, and the default value:

<table>
<thead>
<tr>
<th>Threshold</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lock Wait Time (ms) Per Second upper bound</td>
<td>A Microsoft SQL Server Lock Wait Time Per Second too high indication occurs if the number of milliseconds of lock wait time per second exceeds this value. The threshold is expressed as milliseconds per second.</td>
<td>10</td>
</tr>
<tr>
<td>Number Deadlocks Per Second upper bound</td>
<td>A Microsoft SQL Server Number Deadlocks Per Second too high indication occurs if the number of deadlocks per second exceeds this value.</td>
<td>50</td>
</tr>
</tbody>
</table>

**Parameters**

None

**Tasks and built-in actions**

None

**Logging**

You can log data for the properties of the managed resource listed in the following table. The table shows the context of the managed resource and the properties that the resource model logs for the IBM Tivoli Monitoring Web Health Console.
### Managed resource Context Properties

<table>
<thead>
<tr>
<th>Managed resource</th>
<th>Context</th>
<th>Properties</th>
</tr>
</thead>
</table>
| MSSQLServer      | Locks Performance                | **MSSQLServer.MSSQLServerName**
|                  |                                  | The instance name of Microsoft SQL Server.                                 |
|                  |                                  | **MSSQL_Lock_Requests**
|                  |                                  | The number of new locks per second requested from the lock manager.       |
|                  |                                  | **MSSQL_Lock_Timeouts**
|                  |                                  | The number of lock requests per second that timed out before the lock was granted. |
|                  |                                  | **MSSQL_Lock_Wait_Time**
|                  |                                  | The total wait time in milliseconds that a process spends waiting for another process to release a lock. |
|                  |                                  | **MSSQL_Number_Deadlocks**
|                  |                                  | The number of lock requests per second that resulted in a deadlock. A deadlock occurs when two processes, each having locked a resource, request access to put a lock on the other’s locked resource. |
|                  |                                  | **MSSQLVersion**
|                  |                                  | The version number, v7 or 2k, of the installed Microsoft SQL Server.       |
|                  |                                  | **MSSQLResourceType**
|                  |                                  | The type of resource with which the lock is associated (Page, Table, etc.). |

| MSSQLServer      | TableLock Escalations            | **MSSQLServer.MSSQLServerName**
|                  |                                  | The instance name of Microsoft SQL Server.                                 |
|                  |                                  | **MSSQL_Table_Lock_Escalations**
|                  |                                  | The number of Table Lock Escalations occurring per second.                |
|                  |                                  | **MSSQLVersion**
|                  |                                  | The version number, v7 or 2k, of the installed Microsoft SQL Server.       |

**Note:** An asterisk (*) denotes a key property.

---

### Return codes

Table 2 on page 7 contains a listing of resource model return codes, their description, and what action you can take to resolve each code. The return code number is displayed in the IBM Tivoli Monitoring Web Health Console status field, or by using the `wdmlseng` command.

### CLI example

```
wdmeditprf -P $Prof1 -edit Microsoft_SQL_Server_Locks
-t High_MSSQL_LockWaitTimePerSec 0.000000
-t High_MSSQL_NumberDeadlocksPerSec 0.000000
-e Microsoft_SQL_Server_High_LockWaitTimePerSec
-o 1 -h 0 -severity CRITICAL -SendTBSM -"$MyTec"
-e Microsoft_SQL_Server_High_NumberDeadlocksPerSec
-o 1 -h 0 -severity CRITICAL -SendTBSM -"$MyTec"
```
Replication

Description

Monitors replication performance on Microsoft SQL Server databases. Ensures that replication is performing properly and that the server is configured properly. By default, this resource model monitors all databases. You can exclude certain databases from monitoring by using the Exclude Databases parameter.

Replication is the process of copying and distributing data or database objects from one server or database to another server or database and then synchronizes servers or databases for consistency. Replication offers the following benefits:

- Enhances application performance
- Physically separates data based upon how the data is used
- Distributes database processing across multiple servers

There are three types of replication: Merge, Snapshot, and Transactional.

Merge replication distributes data from Publisher to Subscribers. Publishers and Subscribers can both make changes while connected or disconnected. Merge replication then merges the changes between them after they are connected.

Snapshot replication distributes data exactly as it appears at a specific moment in time and does not refresh for any changes made to the data. During replication, the complete snapshot is generated and sent to Subscribers.

Transactional replication applies an initial snapshot of data to Subscribers, and then captures and propagates data changes made at the Publisher to the same Subscribers.

Replication terminology follows a publishing-industry model composed of the following terms:

Publisher

A server where you specify the data you want to replicate. This server also detects which data has changed during transactional replication and maintains information about all publications at that site.

Distributor

A server that hosts the distribution database and stores history data, transactions, and meta data. The role of the Distributor varies depending on which type of replication you implement. A remote Distributor is a server that is separate from the Publisher and is configured as a Distributor of replication. A local Distributor is a server that is configured to be both a Publisher and a Distributor of replication.

Subscriber

A server that receives replicated data. Subscribers subscribe to publications, not to individual articles within a publication, and they subscribe only to the publications that they need, not all of the publications available on a Publisher. Depending on the type of replication and replication options you choose, the Subscriber can also propagate data changes back to the Publisher or republish the data to other Subscribers.

Article

A table of data, a partition of data, or a database object that is specified for replication.
Subscription

A request for a copy of a database object or data. The subscription order also defines the Subscriber and when the publication is received. Publishers can request a subscription (a push subscription), as well as Subscribers (a pull subscription).

Some replication processes copy and move data between the Publisher and Subscriber. These processes are the Snapshot Agent, Distribution Agent, Log Reader Agent, and Merge Agent.

The Log Reader Agent monitors the transaction log of each database configured for replication, and copies the transactions marked for replication from the transaction log into the distribution database.

The Log Reader Agent is used with transactional replication. It moves transactions marked for replication from the transaction log on the Publisher to the distribution database. Each database that is marked for transactional replication has one Log Reader Agent that runs on the Distributor and connects to the Publisher.

Use the following indications to check the flow of replication transactions from the distribution to the subscriber database:

- Microsoft SQL Server Dist Delivered Cmds Per Second too low indication
- Microsoft SQL Server Dist Delivered Trans Per Second too low indication
- Microsoft SQL Server Distribution Delivery Latency too high indication

Use the following indications to check the flow of replication transactions from the publisher to the distribution database:

- Microsoft SQL Server Replication Pending Transactions too high indication
- Microsoft SQL Server Replication Transaction Rate too low indication

<p>| Resource model overview |      |
|-------------------------|--|--|</p>
<table>
<thead>
<tr>
<th>Resource model overview</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Indications</strong></td>
</tr>
<tr>
<td><strong>Microsoft SQL Server Dist Delivered Cmnds Per Second too low</strong> occurs when the number of distribution commands delivered per second to the Subscriber is less than the Dist Delivered Cmnds Per Second lower bound threshold.</td>
</tr>
<tr>
<td><strong>Microsoft SQL Server Dist Delivered Trans Per Second too low</strong> occurs when the number of distribution transactions delivered per second to the Subscriber is less than the Dist Delivered Trans Per Second lower bound threshold.</td>
</tr>
<tr>
<td><strong>Microsoft SQL Server Distribution Delivery Latency too high</strong> occurs when the duration time for transactions to be delivered to the Distributor and applied at the Subscriber exceeds the Distribution Delivery Latency upper bound threshold.</td>
</tr>
<tr>
<td><strong>Microsoft SQL Server Downloaded Changes Per Second too high</strong> occurs when the number of rows per second merged (inserted, updated, and deleted) from the Publisher to the Subscriber exceeds the Downloaded Changes Per Second upper bound threshold.</td>
</tr>
<tr>
<td><strong>Microsoft SQL Server Logreader Delivered Cmnds Per Second too low</strong> occurs when the number of Log Reader Agent commands delivered per second to the Distributor exceeds the Logreader Delivered Cmnd Per Second lower bound threshold.</td>
</tr>
<tr>
<td><strong>Microsoft SQL Server Logreader Delivered Trans Per Second too low</strong> occurs when the number of Log Reader Agent transactions delivered per second to the Distributor is less than the Logreader Delivered Trans Per Second lower bound threshold.</td>
</tr>
<tr>
<td><strong>Microsoft SQL Server Logreader Delivery Latency too high</strong> occurs when the duration time, in seconds, elapsed from when transactions are applied at the Publisher to when they are delivered to the Distributor exceeds the Logreader Delivery Latency upper bound threshold.</td>
</tr>
<tr>
<td><strong>Microsoft SQL Server Merge Conflicts Per Second too high</strong> occurs when the number of conflicts per second that occurred in the Publisher/Subscriber upload and download exceeds the Merge Conflicts Per Second upper bound threshold.</td>
</tr>
<tr>
<td><strong>Microsoft SQL Server Replication Pending Transactions too high</strong> occurs when the number of pending replication transactions exceeds the Replication Pending Transactions upper bound threshold.</td>
</tr>
<tr>
<td><strong>Microsoft SQL Server Replication Transaction Rate too low</strong> occurs when the replication transaction rate is less than the Replication Transaction Rate lower bound threshold.</td>
</tr>
<tr>
<td><strong>Microsoft SQL Server Uploaded Changes Per Second too high</strong> occurs when the number of rows per second merged (inserted, updated, and deleted) from the Subscriber to the Publisher is less than the Uploaded Changes Per Second upper bound threshold.</td>
</tr>
<tr>
<td><strong>Tasks and built-in actions</strong></td>
</tr>
<tr>
<td><strong>Default cycle time</strong></td>
</tr>
</tbody>
</table>
**Target managed resource**

MicrosoftSQLServer

**Indications and events**

The following table summarizes the indications for this resource model, the events associated with the indications, the default severity of the events, and where to find a detailed description of the indication. The resource model generates an event when the threshold you specify for the indication is triggered.

<table>
<thead>
<tr>
<th>Indication</th>
<th>Generated event</th>
<th>Default severity</th>
<th>Clearing events</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft SQL Server Dist Delivered Cmds Per Second too low</td>
<td>Microsoft_SQL_Server_Low_DistDeliveredCmdsPerSec</td>
<td>Warning</td>
<td>Yes</td>
<td>77</td>
</tr>
<tr>
<td>Microsoft SQL Server Dist Delivered Trans Per Second too low</td>
<td>Microsoft_SQL_Server_Low_DistDeliveredTransPerSec</td>
<td>Warning</td>
<td>Yes</td>
<td>79</td>
</tr>
<tr>
<td>Microsoft SQL Server Distribution Delivery Latency too high</td>
<td>Microsoft_SQL_Server_High_DistDeliveryLatency</td>
<td>Warning</td>
<td>Yes</td>
<td>80</td>
</tr>
<tr>
<td>Microsoft SQL Server Downloaded Changes Per Second too high</td>
<td>Microsoft_SQL_Server_High_DownloadedChangesPerSec</td>
<td>Warning</td>
<td>Yes</td>
<td>82</td>
</tr>
<tr>
<td>Microsoft SQL Server Logreader Delivered Cmds Per Second too low</td>
<td>Microsoft_SQL_Server_Low_LogreaderDeliveredCmdsPerSec</td>
<td>Warning</td>
<td>Yes</td>
<td>83</td>
</tr>
<tr>
<td>Microsoft SQL Server Logreader Delivered Trans Per Second too low</td>
<td>Microsoft_SQL_Server_Low_LogreaderDeliveredTransPerSec</td>
<td>Warning</td>
<td>Yes</td>
<td>84</td>
</tr>
<tr>
<td>Indication</td>
<td>Generated event</td>
<td>Default severity</td>
<td>Clearing events</td>
<td>Page</td>
</tr>
<tr>
<td>----------------------------------------------------------------</td>
<td>-----------------------------------------------</td>
<td>------------------</td>
<td>-----------------</td>
<td>------</td>
</tr>
<tr>
<td>Microsoft SQL Server Logreader Delivery Latency too high</td>
<td>Microsoft_SQL_Server_High_LogreaderDeliveryLatency</td>
<td>Warning</td>
<td>Yes</td>
<td>85</td>
</tr>
<tr>
<td>Microsoft SQL Server Merge Conflicts Per Second too high</td>
<td>Microsoft_SQL_Server_High_MergeConflictsPerSec</td>
<td>Warning</td>
<td>Yes</td>
<td>87</td>
</tr>
<tr>
<td>Microsoft SQL Server Replication Pending Transactions too high</td>
<td>Microsoft_SQL_Server_High_ReplPendingTransactions</td>
<td>Warning</td>
<td>Yes</td>
<td>88</td>
</tr>
<tr>
<td>Microsoft SQL Server Replication Transaction Rate too low</td>
<td>Microsoft_SQL_Server_Low_ReplTransactionRate</td>
<td>Warning</td>
<td>Yes</td>
<td>89</td>
</tr>
<tr>
<td>Microsoft SQL Server Uploaded Changes Per Second too high</td>
<td>Microsoft_SQL_Server_High_UploadedChangesPerSec</td>
<td>Warning</td>
<td>Yes</td>
<td>91</td>
</tr>
</tbody>
</table>

**Microsoft SQL Server Dist Delivered Cmds Per Second too low indication**

Occurs when the number of distribution commands delivered per second to the Subscriber is less than the **Dist Delivered Cmds Per Second lower bound** threshold. This indication monitors the number of commands and transactions read from the distribution database and delivered to the subscriber databases by the SQL Server Distribution Agent. If this indication is triggered, there might be bottlenecks in the replication process of a database. Bottlenecks cause delays in the published transactions reaching the subscriber database.

If the number of distribution commands delivered per second to the Subscriber is too low, consider doing the following:

- Check the network connection between the publication database and the distribution database if they are on separate server machines
- Check the workload on one or both servers to determine if either is overloaded (use the **Microsoft SQL Server CPU Percent Busy too high** indication in the Cache/CPU resource model)
- Determine if the distribution database is available (use the **Suspect Database** indication in the Availability resource model)

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:
<application_label>: Microsoft SQL Server <MSSQLServerName> -
Dist Delivered Cmds Per Second is <MSSQLDistDeliveredCmdsPerSec>, which is below threshold of <LowerBound>.

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

application_class
The registered object’s Tivoli Management Environment (TME) class.

application_label
The registered object’s Tivoli Management Environment (TME) label.

application_oid
The registered object’s Tivoli Management Environment (TME) object identifier.

application_version
The managed resource’s version; for example, v7, or 2k.

LowerBound
The value set as the minimum allowable for the specified threshold.

MSSQLDistDeliveredCmdsPerSec
The number of distribution commands delivered per second to the Subscriber.

MSSQLServerName
The instance name of Microsoft SQL Server.

This indication has the following threshold:
• Dist Delivered Cmds Per Second lower bound

For more information about this threshold, see “Thresholds” on page 92

The following table describes the default settings for this indication:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>1</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.
Microsoft SQL Server Dist Delivered Trans Per Second too low indication

Occurs when the number of distribution transactions delivered per second to the Subscriber is less than the Dist Delivered Trans Per Second lower bound threshold. Use this indication to monitor the number of transactions that have been successfully executed against the destination database during a specified period. If this indication is triggered, there might be bottlenecks in the replication process of a database. Bottlenecks cause delays in the published transactions reaching the subscriber database.

A delivered transaction is a transaction that has been executed against the destination database, but is still residing in the distribution database. Transaction rate can be affected by the following:

- Distribution schedule, which determines the interval between transactions being distributed to the subscriber
- The number of transactions that are configured to be sent in a batch

If the number of distribution transactions delivered per second to the Subscriber is too low, consider doing the following:

- Check the network connection between the distribution database and subscriber database if they are on different servers
- Check the workload on one or both servers to determine if either is overloaded (use the Microsoft SQL Server CPU Percent Busy too high indication in the Cache/CPU resource model)
- Determine if the subscriber database is available (use the Suspect Database indication in the Availability resource model)

If this value is high, you might be unnecessarily filling up the distribution database. Check whether the amount of time that delivered transactions remain in the distribution database is too long. If it is, reduce the value for the period of time.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

\(<\text{application_label}: \text{Microsoft SQL Server } <\text{MSSQLServerName}> - \text{ Dist Delivered Trans Per Second is } <\text{MSSQLDistDeliveredTransPerSec}>, \text{ which is below threshold of } <\text{LowerBound}>.\)  

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**
  The registered object’s Tivoli Management Environment (TME) class.
- **application_label**
  The registered object’s Tivoli Management Environment (TME) label.
- **application_oid**
  The registered object’s Tivoli Management Environment (TME) object identifier.
application_version
The managed resource's version; for example, v7, or 2k.

LowerBound
The value set as the minimum allowable for the specified threshold.

MSSQLDistDeliveredTransPerSec
The number of distribution transactions delivered per second to the Subscriber.

MSSQLServerName
The instance name of Microsoft SQL Server.

This indication has the following threshold:
• Dist Delivered Trans Per Second lower bound

For more information about this threshold, see “Thresholds” on page 92

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>1</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

Note: If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

Microsoft SQL Server Distribution Delivery Latency too high indication
Occurs when the duration time for transactions to be delivered to the Distributor and applied at the Subscriber exceeds the Distribution Delivery Latency upper bound threshold. This indication monitors the number of seconds a replicated transaction remains in the distribution database before it is delivered to the subscription server. The subscription server is the server with the subscriber database on it. The distribution server is the server that has the distribution database on it.

If this indication is triggered, there might be bottlenecks in the replication process of a database. Bottlenecks cause delays in the published transactions reaching the subscriber database. Transaction rate can be affected by the following:
• Distribution schedule, which determines the interval between transactions being distributed to the subscriber
• The number of transactions that are configured to be sent in a batch

Distribution delivery duration time is measured in seconds. If this value is high, consider doing the following:
• Check the network connection between the publication database and the distribution database if they are on separate server machines
• Check the workload on one or both servers to determine if either is overloaded
  (use the Microsoft SQL Server CPU Percent Busy too high indication in the
  Cache/CPU resource model)
• Determine if the distribution database is available (use the Suspect Database
  indication in the Availability resource model)

When the indication occurs often enough to trigger an event, the event delivers a
message to the Tivoli Enterprise Console in the following format:
<application_label>: Microsoft SQL Server <MSSQLServerName> -
Distribution Delivery Latency is <MSSQLDistDeliveryLatency> seconds, which exceeds
threshold of <UpperBound>.

If you have Tivoli Business Systems Manager configured for your system, Tivoli
Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web
Health Console. For more information, see the IBM Tivoli Monitoring Web Health
Console documentation.

The indication has the following attributes:

application_class
  The registered object’s Tivoli Management Environment (TME) class.

application_label
  The registered object’s Tivoli Management Environment (TME) label.

application_oid
  The registered object’s Tivoli Management Environment (TME) object
  identifier.

application_version
  The managed resource’s version; for example, v7, or 2k.

MSSQLDistDeliveryLatency
  The number of seconds a replicated transaction remains in the distribution
database before it is delivered to the subscription server. This is the time
between a job being inserted in the distribution database and being
executed against the destination database.

MSSQLServerName
  The instance name of Microsoft SQL Server.

UpperBound
  The value set as the maximum allowable for the specified threshold.

This indication has the following threshold:
• Distribution Delivery Latency upper bound

For more information about this threshold, see “Thresholds” on page 92

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>2</td>
</tr>
</tbody>
</table>
Holes | 0
---|---
Associated tasks and built-in actions | None

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Microsoft SQL Server Downloaded Changes Per Second too high indication**

Occurs when the number of rows per second merged (inserted, updated, and deleted) from the Publisher to the Subscriber exceeds the **Downloaded Changes Per Second upper bound** threshold. Merge executions move data changes up from a merge replication Subscriber to the Publisher, and down from the Publisher to the Subscriber.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

<application_label>: Microsoft SQL Server <MSSQLServerName> - Downloaded Changes Per Second is <MSSQLDownloadedChangesPerSec>, which exceeds threshold of <UpperBound>.

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**
  The registered object’s Tivoli Management Environment (TME) class.

- **application_label**
  The registered object’s Tivoli Management Environment (TME) label.

- **application_oid**
  The registered object’s Tivoli Management Environment (TME) object identifier.

- **application_version**
  The managed resource’s version; for example, v7, or 2k.

- **MSSQLDownloadedChangesPerSec**
  The number of rows per second merged (inserted, updated, and deleted) from the Publisher to the Subscriber. Merge executions move data changes up from a merge replication Subscriber to the Publisher, and down from the Publisher to the Subscriber.

- **MSSQLServerName**
  The instance name of Microsoft SQL Server.

- **UpperBound**
  The value set as the maximum allowable for the specified threshold.

This indication has the following threshold:
• Downloaded Changes Per Second upper bound

For more information about this threshold, see "Thresholds" on page 92.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>2</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Microsoft SQL Server Logreader Delivered Cmds Per Second too low indication**

Occurs when the number of Log Reader Agent commands delivered per second to the Distributor exceeds the Logreader Delivered Cmds Per Second lower bound threshold. If this indication is triggered, there might be possible bottlenecks in the replication process of a database. Bottlenecks cause delays in the published transactions reaching the subscriber database.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

```<application_label>: Microsoft SQL Server <MSSQLServerName> - Logreader Delivered Cmds Per Second is <MSSQLLogreaderDeliveredCmdsPerSec>, which is below threshold of <LowerBound>.```

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**
  The registered object’s Tivoli Management Environment (TME) class.

- **application_label**
  The registered object’s Tivoli Management Environment (TME) label.

- **application_oid**
  The registered object’s Tivoli Management Environment (TME) object identifier.

- **application_version**
  The managed resource’s version; for example, v7, or 2k.

- **LowerBound**
  The value set as the minimum allowable for the specified threshold.
**MSSQLLogreaderDeliveredCmdsPerSec**

The number of Log Reader Agent commands delivered per second to the Distributor.

**MSSQLServerName**

The instance name of Microsoft SQL Server.

This indication has the following threshold:

- Logreader Delivered Cmds Per Second lower bound

For more information about this threshold, see “Thresholds” on page 92.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>1</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Microsoft SQL Server Logreader Delivered Trans Per Second too low indication**

Occurs when the number of Log Reader Agent transactions delivered per second to the Distributor is less than the Logreader Delivered Trans Per Second lower bound threshold. If this indication is triggered, there might be possible bottlenecks in the replication process of a database. Bottlenecks cause delays in the published transactions reaching the subscriber database.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

\(<application_label\>: Microsoft SQL Server \(<MSSQLServerName>\) - Logreader Delivered Trans Per Second is \(<MSSQLLogreaderDeliveredTransPerSec>\), which is below threshold of \(<LowerBound>\).\n
If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**
  The registered object’s Tivoli Management Environment (TME) class.

- **application_label**
  The registered object’s Tivoli Management Environment (TME) label.
The registered object’s Tivoli Management Environment (TME) object identifier.

The managed resource’s version; for example, v7, or 2k.

The value set as the minimum allowable for the specified threshold.

The number of Log Reader Agent transactions delivered per second to the Distributor.

The instance name of Microsoft SQL Server.

This indication has the following threshold:
- Logreader Delivered Trans Per Second lower bound

For more information about this threshold, see “Thresholds” on page 92

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>1</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

Note: If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Microsoft SQL Server Logreader Delivery Latency too high indication**

Occurs when the duration time, in seconds, elapsed from when transactions are applied at the Publisher to when they are delivered to the Distributor exceeds the Logreader Delivery Latency upper bound threshold. If this indication is triggered, there might be possible bottlenecks in the replication process of a database. Bottlenecks cause delays in the published transactions reaching the subscriber database.

Distribution delivery duration time is measured in seconds. If this value is high, consider doing the following:
- Check the network connection between the publication database and the distribution database if they are on separate server machines
- Check the workload on one or both servers to determine if either is overloaded (use the Microsoft SQL Server CPU Percent Busy too high indication in the Cache/CPU resource model)
- Determine if the distribution database is available (use the Suspect Database indication in the Availability resource model)
When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

\(<application\_label>\): Microsoft SQL Server \(<MSSQLServerName>\) - Logreader

Delivery Latency is \(<MSSQLLogreaderDeliveryLatency>\) seconds, which exceeds threshold of \(<UpperBound>\).

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application\_class**
  The registered object’s Tivoli Management Environment (TME) class.

- **application\_label**
  The registered object’s Tivoli Management Environment (TME) label.

- **application\_oid**
  The registered object’s Tivoli Management Environment (TME) object identifier.

- **application\_version**
  The managed resource’s version; for example, v7, or 2k.

- **MSSQLLogreaderDeliveryLatency**
  The duration time, in seconds, elapsed from when transactions are applied at the Publisher to when they are delivered to the Distributor.

- **MSSQLServerName**
  The instance name of Microsoft SQL Server.

- **UpperBound**
  The value set as the maximum allowable for the specified threshold.

This indication has the following threshold:
- Logreader Delivery Latency upper bound

For more information about this threshold, see “Thresholds” on page 92.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>2</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.
Microsoft SQL Server Merge Conflicts Per Second too high indication

Occurs when the number of conflicts per second that occurred in the Publisher/Subscriber upload and download exceeds the Merge Conflicts Per Second upper bound threshold. This indication monitors each SQL Server merge execution that moves data changes up from a merge replication Subscriber to the Publisher, and down from the Publisher to the Subscriber.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

```
<application_label>: Microsoft SQL Server <MSSQLServerName> - Merge Conflicts Per Second is <MSSQLMergeConflictsPerSec>, which exceeds threshold of <UpperBound>.
```

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**
  The registered object’s Tivoli Management Environment (TME) class.

- **application_label**
  The registered object’s Tivoli Management Environment (TME) label.

- **application_oid**
  The registered object’s Tivoli Management Environment (TME) object identifier.

- **application_version**
  The managed resource’s version; for example, v7, or 2k.

- **MSSQLMergeConflictsPerSec**
  The number of conflicts per second that occurred in the Publisher/Subscriber upload and download.

- **MSSQLServerName**
  The instance name of Microsoft SQL Server.

- **UpperBound**
  The value set as the maximum allowable for the specified threshold.

This indication has the following threshold:

- **Merge Conflicts Per Second upper bound**

For more information about this threshold, see “Thresholds” on page 92.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>4</td>
</tr>
</tbody>
</table>
Setting | Default value
---|---
Holes | 0
Associated tasks and built-in actions | None

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Microsoft SQL Server Replication Pending Transactions too high indication**

Occurs when the number of pending replication transactions exceeds the **Replication Pending Transactions upper bound** threshold. If this indication is triggered, there might be possible bottlenecks in the replication process of a database. Bottlenecks cause delays in the published transactions reaching the subscriber database.

If the number of Replication Transactions pending is high, consider doing the following:

- Check the network connection between the publication database and the distribution database if they are on separate server machines.
- Check the workload on one or both servers to determine if either is overloaded (use the **Microsoft SQL Server CPU Percent Busy too high** indication in the **Cache/CPU** resource model).
- Determine if the distribution database is available (use the **Suspect Database** indication in the **Availability** resource model).

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

```xml
<application_label>: Microsoft SQL Server <MSSQLServerName>, Database <MSSQLDatabaseName> - Number of Replication Pending Transactions is <MSSQLReplPendingTransactions>, which exceeds threshold of <UpperBound>.
```

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**
  The registered object’s Tivoli Management Environment (TME) class.

- **application_label**
  The registered object’s Tivoli Management Environment (TME) label.

- **application_oid**
  The registered object’s Tivoli Management Environment (TME) object identifier.

- **application_version**
  The managed resource’s version; for example, v7, or 2k.
**MSSQLDatabaseName**
The name of the database on the specified Microsoft SQL Server.

**MSSQLReplPendingTransactions**
The number of transactions in the transaction log of the publication database that are marked for replication, but have not been delivered to the distribution database.

**MSSQLServerName**
The instance name of Microsoft SQL Server.

**UpperBound**
The value set as the maximum allowable for the specified threshold.

This indication has the following threshold:
- Replication Pending Transactions upper bound

For more information about this threshold, see “Thresholds” on page 92.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems</td>
<td>No</td>
</tr>
<tr>
<td>Manager</td>
<td></td>
</tr>
<tr>
<td>Occurrences</td>
<td>1</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Microsoft SQL Server Replication Transaction Rate too low indication**
Occurs when the replication transaction rate is less than the **Replication Transaction Rate lower bound** threshold. If this indication is triggered, there might be possible bottlenecks in the replication process of a database. Bottlenecks cause delays in the published transactions reaching the subscriber database.

If the Replication Transaction Rate is low, consider doing the following:
- Check the network connection between the publication database and the distribution database if they are on separate server machines
- Check the workload on one or both servers to determine if either is overloaded (use the **Microsoft SQL Server CPU Percent Busy too high** indication in the Cache/CPU resource model)
- Determine if the distribution database is available (use the **Suspect Database** indication in the Availability resource model)

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

```
<application_label>: Microsoft SQL Server <MSSQLServerName>, Database <MSSQLDatabaseName> - Replication Transaction Rate is <MSSQLReplTransactionRate>, which is below threshold of <LowerBound>.
```
If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**: The registered object’s Tivoli Management Environment (TME) class.
- **application_label**: The registered object’s Tivoli Management Environment (TME) label.
- **application_oid**: The registered object’s Tivoli Management Environment (TME) object identifier.
- **application_version**: The managed resource’s version; for example, v7, or 2k.
- **LowerBound**: The value set as the minimum allowable for the specified threshold.
- **MSSQLDatabaseName**: The name of the database on the specified Microsoft SQL Server.
- **MSSQLReplTransactionRate**: The number of transactions per second read out of the transaction log of the publication database and delivered to the distribution database.
- **MSSQLServerName**: The instance name of Microsoft SQL Server.

This indication has the following threshold:

- Replication Transaction Rate lower bound

For more information about this threshold, see “Thresholds” on page 92.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>1</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.
Microsoft SQL Server Uploaded Changes Per Second too high indication

Occurs when the number of rows per second merged (inserted, updated, and deleted) from the Subscriber to the Publisher is less than the **Uploaded Changes Per Second upper bound** threshold. When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

```<application_label>: Microsoft SQL Server <MSSQLServerName> - Uploaded Changes Per Second is <MSSQLUploadedChangesPerSec>, which exceeds threshold of <UpperBound>.```

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**
  The registered object’s Tivoli Management Environment (TME) class.

- **application_label**
  The registered object’s Tivoli Management Environment (TME) label.

- **application_oid**
  The registered object’s Tivoli Management Environment (TME) object identifier.

- **application_version**
  The managed resource’s version; for example, v7, or 2k.

- **MSSQLServerName**
  The instance name of Microsoft SQL Server.

- **MSSQLUploadedChangesPerSec**
  The number of rows per second merged (inserted, updated, and deleted) from the Subscriber to the Publisher.

- **UpperBound**
  The value set as the maximum allowable for the specified threshold.

This indication has the following threshold:

- Uploaded Changes Per Second upper bound

For more information about this threshold, see “Thresholds” on page 92.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>2</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>
**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Thresholds**

The following table lists the thresholds that can be set for the Replication resource model. For each threshold it shows the name, a short description, and the default value:

<table>
<thead>
<tr>
<th>Threshold</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dist Delivered Cmds Per Second lower bound</td>
<td>A <strong>Microsoft SQL Server Dist Delivered Cmds Per Second too low</strong> indication occurs when the number of distribution commands delivered per second to the Subscriber is less than this value. The default threshold setting is based on a target rate of 30 transactions per minute and assumes the distribution schedule is set to run every minute.</td>
<td>0</td>
</tr>
<tr>
<td>Dist Delivered Trans Per Second lower bound</td>
<td>A <strong>Microsoft SQL Server Dist Delivered Trans Per Second too low</strong> indication occurs if the number of distribution transactions delivered per second to the Subscriber is less than this value.</td>
<td>5</td>
</tr>
<tr>
<td>Distribution Delivery Latency upper bound</td>
<td>A <strong>Microsoft SQL Server Distribution Delivery Latency too high</strong> indication occurs if the duration time for transactions to be delivered to the Distributor and applied at the Subscriber exceeds this value. The unit of measure is in seconds. The default threshold setting is based on a target rate of 30 transactions per minute and assumes distribution schedule is set to run every minute.</td>
<td>5</td>
</tr>
<tr>
<td>Downloaded Changes Per Second upper bound</td>
<td>A <strong>Microsoft SQL Server Downloaded Changes Per Second too high</strong> indication occurs if number of rows per second merged (inserted, updated, and deleted) from the Publisher to the Subscriber exceeds this value.</td>
<td>99</td>
</tr>
<tr>
<td>Logreader Delivered Cmds Per Second lower bound</td>
<td>A <strong>Microsoft SQL Server Logreader Delivered Cmds Per Second too low</strong> indication occurs if the number of Log Reader Agent commands delivered per second to the Distributor is less than this value.</td>
<td>0</td>
</tr>
<tr>
<td>Logreader Delivered Trans Per Second lower bound</td>
<td>A <strong>Microsoft SQL Server Logreader Delivered Trans Per Second too low</strong> indication occurs if the number of Log Reader Agent transactions delivered per second to the Distributor is less than this value.</td>
<td>5</td>
</tr>
<tr>
<td>Logreader Delivery Latency upper bound</td>
<td>A <strong>Microsoft SQL Server Logreader Delivery Latency too high</strong> indication occurs if the duration time, in seconds, elapsed from when transactions are applied at the Publisher to when they are delivered to the Distributor exceeds this value. The unit of measure is in seconds.</td>
<td>5</td>
</tr>
</tbody>
</table>
Threshold Description Default value

<table>
<thead>
<tr>
<th>Threshold</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Merge Conflicts Per Second upper bound</td>
<td>A Microsoft SQL Server Merge Conflicts Per Second too high indication occurs if the number of conflicts per second that occurred in the Publisher/Subscriber upload and download exceeds this value.</td>
<td>1</td>
</tr>
<tr>
<td>Replication Pending Transactions upper bound</td>
<td>A Microsoft SQL Server Replication Pending Transactions too high indication occurs if the number of pending replication transactions exceeds this value.</td>
<td>90</td>
</tr>
</tbody>
</table>
| Replication Transaction Rate lower bound      | A Microsoft SQL Server Replication Transaction Rate too low indication occurs if the replication transaction rate is less than this value.  
The default threshold setting is based on a target rate of 30 transactions per minute and the log reader set to run every minute. | 0             |
| Uploaded Changes Per Second upper bound       | A Microsoft SQL Server Uploaded Changes Per Second too high indication occurs if the number of rows per second merged (inserted, updated, and deleted) from the Subscriber to the Publisher exceeds this value. | 99            |

Parameters

The following table lists the parameter that can be set for the Replication resource model. The table shows the name, a short description, and the default value for each parameter:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
</table>
| Exclude Databases          | Type any database names you want to exclude from monitoring for the "Replication Pending Transactions" and "Replication Transaction Rate" metrics. When this parameter is blank, the resource model monitors all databases associated with the Microsoft SQL Server for transaction activity.  
You can type more than one database name. Separate multiple database names with spaces.  
Wildcard characters are not recognized.                                                                 | Blank (monitors all databases) |

Tasks and built-in actions

None

Logging

You can log data for the properties of the managed resource listed in the following table. The table shows the context of the managed resource and the properties that the resource model logs for the IBM Tivoli Monitoring Web Health Console.
<table>
<thead>
<tr>
<th>Managed resource</th>
<th>Context</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSSQLDatabase/Replication</td>
<td>Transaction Performance</td>
<td>MSSQLServer.MSSQLServerName*&lt;br&gt;The instance name of Microsoft SQL Server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQLDatabaseReplication.MSSQLDatabaseName*&lt;br&gt;The name of the database on the specified Microsoft SQL Server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_Replication_Pending_Transactions&lt;br&gt;The number of transactions in the transaction log of the publication database that are marked for replication, but have not been delivered to the distribution database.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_Replication_Transaction_Rate&lt;br&gt;The number of transactions per second read out of the transaction log of the publication database and delivered to the distribution database.</td>
</tr>
<tr>
<td>MSSQLServer\Replication</td>
<td>Merge Performance</td>
<td>MSSQLServer.MSSQLServerName*&lt;br&gt;The instance name of Microsoft SQL Server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQLServerReplication.MSSQLAgentType*&lt;brSpecifies the type of Replication Agent: Distribution, Logreader or Merge. This attribute is part of a composite key for logging metrics with respect to the specific replication agent.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_Conflicts&lt;br&gt;The number of conflicts per second that occurred in the Publisher/Subscriber upload and download.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_Downloaded_Changes&lt;br&gt;The number of rows per second merged (inserted, updated, and deleted) from the Publisher to the Subscriber. Merge executions move data changes up from a merge replication Subscriber to the Publisher, and down from the Publisher to the Subscriber.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_Uploaded_Changes&lt;br&gt;The number of rows per second merged (inserted, updated, and deleted) from the Subscriber to the Publisher.</td>
</tr>
</tbody>
</table>
Managed resource Context Properties

<table>
<thead>
<tr>
<th>Managed resource</th>
<th>Context</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSSQLServer\Replication</td>
<td>Performance</td>
<td>MSSQLServer.MSSQLServerName* The instance name of Microsoft SQL Server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQLServerReplication.MSSQLAgentType* Specifies the type of Replication Agent: Distribution, Logreader or Merge. This attribute is part of a composite key for logging metrics with respect to the specific replication agent.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_Delivery_Latency Specifies the average transaction latency.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_Delivered_Commands Specifies the number of commands per second.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_Delivered_Transactions Specifies the number of transactions per second.</td>
</tr>
</tbody>
</table>

Note: An asterisk (*) denotes a key property.

Return codes

Table 2 on page 7 contains a listing of resource model return codes, their description, and what action you can take to resolve each code. The return code number is displayed in the IBM Tivoli Monitoring Web Health Console status field, or by using the `wdmlseng` command.

CLI example

```
wdmeditprf -P $Prof1 -edit Microsoft_SQL_Server_Replication -t Low_MSSQL_DistDeliveredCmdsPerSec 99.000000 -t Low_MSSQL_DistDeliveredTransPerSec 99.000000 -t High_MSSQL_DistDeliveryLatency 0.000000 -t High_MSSQL_DownloadedChangesPerSec 0.000000 -t Low_MSSQL_LogreaderDeliveredCmdsPerSec 99.000000 -t Low_MSSQL_LogreaderDeliveredTransPerSec 99.000000 -t High_MSSQL_LogreaderDeliveryLatency 0.000000 -t High_MSSQL_MergeConflictsPerSec 0.000000 -t High_MSSQL_ReplPendingTransactions 0.000000 -t Low_MSSQL_ReplTransactionRate 100.000000 -t High_MSSQL_UploadedChangesPerSec 0.000000 -e Microsoft_SQL_Server_High_DownloadedChangesPerSec -o 1 -h 0 -severity CRITICAL -SendTBSM -"$MyTec" -e Microsoft_SQL_Server_High_LogreaderDeliveryLatency -o 1 -h 0 -severity CRITICAL -SendTBSM -"$MyTec" -e Microsoft_SQL_Server_Low_LogreaderDeliveredCmdsPerSec -o 1 -h 0 -severity CRITICAL -SendTBSM -"$MyTec" -e Microsoft_SQL_Server_Low_LogreaderDeliveredTransPerSec -o 1 -h 0 -severity CRITICAL -SendTBSM -"$MyTec" -e Microsoft_SQL_Server_High_DistDeliveryLatency -o 1 -h 0 -severity CRITICAL -SendTBSM -"$MyTec" -e Microsoft_SQL_Server_Low_ReplTransactionRate -o 1 -h 0 -severity CRITICAL -SendTBSM -"$MyTec" -e Microsoft_SQL_Server_High_MergeConflictsPerSec -o 1 -h 0 -severity CRITICAL -SendTBSM -"$MyTec" -e Microsoft_SQL_Server_Low_LogreaderDeliveredTransPerSec -o 1 -h 0 -severity CRITICAL -SendTBSM -"$MyTec" -e Microsoft_SQL_Server_High_DistDeliveredTransPerSec -o 1 -h 0 -severity CRITICAL -SendTBSM -"$MyTec" -e Microsoft_SQL_Server_High_DistDeliveredCmdsPerSec -o 1 -h 0 -severity CRITICAL -SendTBSM -"$MyTec" -e Microsoft_SQL_Server_High_UploadedChangesPerSec
```
-o 1 -h 0 -severity CRITICAL -SendTBSM "$MyTec"
-e Microsoft_SQL_Server_High_ReplPendingTransactions \
-o 1 -h 0 -severity CRITICAL -SendTBSM "$MyTec"
-AddPar Exclude_Databases "testdb"
Replication Jobs

Description
Monitors replication jobs to verify completion and returns replication job status. A replication job is the automated process of duplicating data from one or more tables or databases and synchronizing the information between databases to maintain consistency.

By default, this resource model monitors all replication jobs. You can customize this resource model to exclude specific jobs from monitoring by using the Ignored Jobs parameter.

An indication occurs when a replication job’s current or completion status is one of the following:
• Cancelled
• Failed
• Idle
• Suspended
• Waiting for thread

<table>
<thead>
<tr>
<th>Resource model overview</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal name</td>
</tr>
<tr>
<td>Category</td>
</tr>
<tr>
<td>Indications</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Tasks and built-in actions</td>
</tr>
<tr>
<td>Default cycle time</td>
</tr>
</tbody>
</table>

Target managed resource
MicrosoftSQLServer

Indications and events
The following table summarizes the indications for this resource model, the events associated with the indications, the default severity of the events, and where to find a detailed description of the indication. The resource model generates an event when the threshold you specify for the indication is triggered.
# Microsoft SQL Server Cancelled Replication Job indication

Occurs when a replication job status is cancelled. When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

```
<application_label>: The Microsoft SQL Server Replication Job <MSSQLReplicationJobName> for server <MSSQLServerName> was cancelled.
```

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**
  The registered object’s Tivoli Management Environment (TME) class.

- **application_label**
  The registered object’s Tivoli Management Environment (TME) label.

- **application_oid**
  The registered object’s Tivoli Management Environment (TME) object identifier.

- **application_version**
  The managed resource’s version; for example, v7, or 2k.

- **MSSQLReplicationJobName**
  The name of the selected Microsoft SQL Server replication job.

- **MSSQLServerName**
  The Microsoft SQL Server server name on which the replication job is running.

This indication has no thresholds.
The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>1</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Microsoft SQL Server Failed Replication Job indication**

Occurs when a replication job status fails. When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

```xml
<application_label>: The Microsoft SQL Server Replication Job
<MSSQLReplicationJobName> for server <MSSQLServerName> has failed.
```

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**
  - The registered object’s Tivoli Management Environment (TME) class.

- **application_label**
  - The registered object’s Tivoli Management Environment (TME) label.

- **application_oid**
  - The registered object’s Tivoli Management Environment (TME) object identifier.

- **application_version**
  - The managed resource’s version; for example, v7, or 2k.

- **MSSQLFailedReplicationJobErrorCode**
  - The error code from a failed replication job.

- **MSSQLReplicationJobName**
  - The name of the selected Microsoft SQL Server replication job.

- **MSSQLServerName**
  - The Microsoft SQL Server server name on which the replication job is running.

This indication has no thresholds.
The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>1</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Microsoft SQL Server Idle Replication Job indication**

Occurs when a replication job status is idle. When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

```xml
<application_label>: Microsoft SQL Server Idle Replication Job
<MSSQLReplicationJobName> for server <MSSQLServerName> is idle.
```

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- `application_class`
  The registered object’s Tivoli Management Environment (TME) class.

- `application_label`
  The registered object’s Tivoli Management Environment (TME) label.

- `application_oid`
  The registered object’s Tivoli Management Environment (TME) object identifier.

- `application_version`
  The managed resource’s version; for example, v7, or 2k.

- `MSSQLReplicationJobName`
  The name of the selected Microsoft SQL Server replication job.

- `MSSQLServerName`
  The Microsoft SQL Server server name on which the replication job is running.

This indication has no thresholds.
The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>1</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Microsoft SQL Server Suspended Replication Job indication**

Occurs when a replication job status is suspended. When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

```
<application_label>: The Microsoft SQL Server Replication Job
<MSSQLReplicationJobName> for server <MSSQLServerName> is suspended.
```

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**
  The registered object’s Tivoli Management Environment (TME) class.

- **application_label**
  The registered object’s Tivoli Management Environment (TME) label.

- **application_oid**
  The registered object’s Tivoli Management Environment (TME) object identifier.

- **application_version**
  The managed resource’s version; for example, v7, or 2k.

- **MSSQLReplicationJobName**
  The name of the selected Microsoft SQL Server replication job.

- **MSSQLServerName**
  The Microsoft SQL Server server name on which the replication job is running.

This indication has no thresholds.
The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems</td>
<td>No</td>
</tr>
<tr>
<td>Manager</td>
<td></td>
</tr>
<tr>
<td>Occurrences</td>
<td>1</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Microsoft SQL Server Waiting For Thread Replication Job indication**

Occurs when a replication job status is Waiting for Thread. The database server uses threads to identify and store information for each individual user or a particular service request. The server uses the stored information to complete user and service requests. Threads help ensure that work is being performed for some user connections even when other connections are blocked (for example, when waiting for a disk read or write operation to complete). A replication job cannot complete when its status is Waiting for Thread.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

```text
<application_label>: The Microsoft SQL Server Replication Job <MSSQLReplicationJobName> for server <MSSQLServerName> is waiting for a thread.
```

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- `application_class`
  - The registered object’s Tivoli Management Environment (TME) class.

- `application_label`
  - The registered object’s Tivoli Management Environment (TME) label.

- `application_oid`
  - The registered object’s Tivoli Management Environment (TME) object identifier.

- `application_version`
  - The managed resource’s version; for example, v7, or 2k.

- `MSSQLReplicationJobName`
  - The name of the selected Microsoft SQL Server replication job.
**MSSQLServerName**

The Microsoft SQL Server server name on which the replication job is running.

This indication has no thresholds.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>1</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Thresholds**

None

**Parameters**

The following table lists the parameter that can be set for the Replication Jobs resource model. The table shows the name, a short description, and the default value for each parameter:
### Parameter Description Default value

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ignored Jobs</td>
<td>Type any job names you want to exclude from monitoring.</td>
<td>Blank (monitors all replication jobs)</td>
</tr>
<tr>
<td></td>
<td>When this parameter is blank, the resource model monitors all replication jobs.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The wildcard character (*) is supported.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The following examples illustrate how to code your parameter:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• To match a single job such as &quot;XYZ,&quot; use the following expression:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>XYZ</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• To match a job that begins with &quot;ABC,&quot; use the following expression:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC*</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• To match a job that contains &quot;DEF,&quot; use the following expression:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><em>DEF</em></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• To match a job that begins with &quot;A&quot; and ends with &quot;Z,&quot; use the following expression:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A*Z</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The &quot;</td>
<td>&quot; character is not supported in these expressions.</td>
</tr>
<tr>
<td>Refresh Monitored Jobs Frequency</td>
<td>Defines how often the resource model receives a new list of monitored jobs from the Microsoft SQL Server. The value represents the number of resource model cycles between retrieving a new list of replication jobs to monitor.</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>For example, if the cycle time is 60 seconds and this parameter is set to 5, the resource model receives the list of replication jobs every 5 minutes.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>When a replication job is added, it is monitored the next time the list of replication jobs is retrieved.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Specify only one entry for this parameter.</td>
<td></td>
</tr>
</tbody>
</table>

### Tasks and built-in actions

None

### Logging

You can log data for the properties of the managed resource listed in the following table. The table shows the context of the managed resource and the properties that the resource model logs for the IBM Tivoli Monitoring Web Health Console.
<table>
<thead>
<tr>
<th>Managed resource</th>
<th>Context</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSSQLJobCategory</td>
<td>Status</td>
<td>MSSQLJobCategory.MSSQLJobCategory*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The job category of the specified Microsoft SQL Server job.</td>
</tr>
<tr>
<td>MSSQLServerReplication.MSSQLServerName*</td>
<td></td>
<td>The name of the Microsoft SQL Server server.</td>
</tr>
<tr>
<td>MSSQLReplicationJobName</td>
<td></td>
<td>The name of the selected Microsoft SQL Server replication job.</td>
</tr>
<tr>
<td>MSSQL_Replication_Job_Between_Retries</td>
<td></td>
<td>The time duration in seconds between retries of a successful execution for the specified replication job.</td>
</tr>
<tr>
<td>MSSQL_Replication_Job_Canceled</td>
<td></td>
<td>Specifies whether the specified replication job is canceled. A returned status of “1” indicates the replication job is canceled. A returned status of “0” indicates a status other than canceled.</td>
</tr>
<tr>
<td>MSSQL_Replication_Job_Executing</td>
<td></td>
<td>Specifies whether the specified replication job is executing. A returned status of “1” indicates the replication job is executing. A returned status of “0” indicates a status other than executing.</td>
</tr>
<tr>
<td>MSSQL_Replication_Job_Failed</td>
<td></td>
<td>Specifies whether the specified replication job failed. A returned status of “1” the replication job failed. A returned status of “0” indicates a status other than failed.</td>
</tr>
<tr>
<td>MSSQL_Replication_Job_Idle</td>
<td></td>
<td>Specifies whether the specified replication job is idle. A returned status of “1” the replication job is idle. A returned status of “0” indicates a status other than idle.</td>
</tr>
<tr>
<td>MSSQL_Replication_Job_In_Progress</td>
<td></td>
<td>Specifies whether the specified replication job is in progress. A returned status of “1” the replication job is in progress. A returned status of “0” indicates a status other than in progress.</td>
</tr>
<tr>
<td>MSSQL_Replication_Job_Performing_Completion_Actions</td>
<td></td>
<td>Specifies whether the specified replication job is performing completion actions. A returned status of “1” indicates the replication job is performing completion actions. A returned status of “0” indicates a status other than performing completion actions.</td>
</tr>
</tbody>
</table>

**Note:** An asterisk (*) denotes a key property.
### Managed resource, Context, Properties

<table>
<thead>
<tr>
<th>Managed resource</th>
<th>Context</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSSQLJobCategory</td>
<td>Status</td>
<td><strong>MSSQL_Replication_Job_Retry</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Specifies whether the specified replication job is retrying. A returned status of “1” indicates the replication job is retrying. A returned status of “0” indicates a status other than retrying.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>MSSQL_Replication_Job_Succeeded</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Specifies whether the specified replication job succeeded. A returned status of “1” indicates the replication job succeeded. A returned status of “0” indicates a status other than success.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>MSSQL_Replication_Job_Suspended</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Specifies whether the specified replication job is suspended. A returned status of “1” indicates the replication job is suspended. A returned status of “0” indicates a status other than suspended.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>MSSQL_Replication_Job_Waiting _For _Thread</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Specifies whether the specified replication job is waiting for a thread. A returned status of “1” indicates the replication job is waiting for a thread. A returned status of “0” indicates a status other than waiting for a thread.</td>
</tr>
</tbody>
</table>

**Note:** An asterisk (*) denotes a key property.

---

### Return codes

Table 2 on page 7 contains a listing of resource model return codes, their description, and what action you can take to resolve each code. The return code number is displayed in the IBM Tivoli Monitoring Web Health Console status field, or by using the `wdmlseng` command.

### CLI example

```
wdmeditprf -P $Prof1 -edit Microsoft_SQL_Server_Replication_Jobs \\
-AddPar Replication_Job\_Names "Agent history clean up: distribution" \\
-AddPar Replication_Job\_Names "Distribution clean up: distribution" \\
-AddPar Replication_Job\_Names "Reinitialize subscriptions having data validation failures" \\
-AddPar Replication_Job\_Names "Replication agents checkup" \\
-e Microsoft\_SQL\_Server\_Waiting\_For\_Thread\_Replication\_Job \\
  -o 1 -h 0 -severity CRITICAL -NoSendTBSM -"$MyTec" \\
-e Microsoft\_SQL\_Server\_Idle\_Replication\_Job \\
  -o 1 -h 0 -severity CRITICAL -NoSendTBSM -"$MyTec" \\
-e Microsoft\_SQL\_Server\_Failed\_Replication\_Job \\
  -o 1 -h 0 -severity CRITICAL -NoSendTBSM -"$MyTec" \\
-e Microsoft\_SQL\_Server\_Cancelled\_Replication\_Job \\
  -o 1 -h 0 -severity CRITICAL -NoSendTBSM -"$MyTec" \\
-e Microsoft\_SQL\_Server\_Suspended\_Replication\_Job \\
  -o 1 -h 0 -severity CRITICAL -NoSendTBSM -"$MyTec"
```
Space Usage

Description

Monitors space usage in the Microsoft SQL Server. Use this resource model to monitor whether a specific database server has adequate space for operation. By default, this resource model monitors all databases. You can exclude certain databases from monitoring by using the Exclude Databases parameter.

Adequate space allocation is critical for optimal performance. Managing space usage minimizes the downtime and performance degradations that result from inadequate space allocation. This resource model monitors the following space usage elements:
- Database growth
- Database space percent used (database size)
- Filegroup size
- Log size

Database growth is how much the database has grown since the last resource model cycle. The resource model checks the database size from the previous resource model cycle. The `sp_spaceused` SQL command returns the database size in MB and the unallocated space in MB. The actual database size is determined by subtracting the unallocated space from the database size. If this is the first cycle, the resource model saves the current database size in the registry. On subsequent cycles, the resource model retrieves the last database size from the registry and subtracts it from the current database size to get the difference, which constitutes database growth. Also, the current database size is stored in the registry for the next cycle. The database growth is calculated as a percentage by dividing the last database size by the difference in size.

The Database Space Percent Used percentage is determined by dividing the database size by the maximum growth size for the database. The maximum growth size is determined by summing the maximum size of each physical database file that makes up the database.

A filegroup consists of one or more physical database files. In Microsoft SQL Server, every physical database file is assigned to a filegroup. A filegroup’s maximum size is defined as the maximum size of each physical database file assigned to the filegroup. However, if any database file has a maximum size of "unlimited," then the filegroup is constrained only by the size of the disk drives that make up the filegroup. When a database file is configured, it allocates the maximum size required. Non-database files on the physical disk also take space away from the filegroup. As a result, the space available to a filegroup is the free disk space on the disk drives that make up the filegroup.

Log Space Percent Used is determined by dividing the actual log size by the maximum growth size of the physical log files.

<table>
<thead>
<tr>
<th>Resource model overview</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal name</td>
<td>Microsoft_SQL_Server_Space_Usage</td>
</tr>
<tr>
<td>Category</td>
<td>Microsoft SQL Server</td>
</tr>
</tbody>
</table>
Resource model overview

<table>
<thead>
<tr>
<th>Indications</th>
<th>Microsoft SQL Server High DatabaseGrowth occurs when the growth in used database space exceeds the specified threshold.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indications</td>
<td>Microsoft SQL Server High DatabaseSpacePercentUsed occurs when the used database space percent exceeds the specified threshold.</td>
</tr>
<tr>
<td>Indications</td>
<td>Microsoft SQL Server High FilegroupPercentUsed occurs when the used filegroup space percent exceeds the specified threshold.</td>
</tr>
<tr>
<td>Indications</td>
<td>Microsoft SQL Server High LogSpacePercentUsed occurs when the log space percent used exceeds the specified threshold.</td>
</tr>
</tbody>
</table>

Tasks and built-in actions  None

Default cycle time  600 seconds (10 minutes)

Target managed resource

Microsoft SQL Server

Indications and events

The following table summarizes the indications for this resource model, the events associated with the indications, the default severity of the events, and where to find a detailed description of the indication. The resource model generates an event when the threshold you specify for the indication is triggered.

<table>
<thead>
<tr>
<th>Indication</th>
<th>Generated event</th>
<th>Default severity</th>
<th>Clearing events</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft SQL Server High DatabaseGrowth</td>
<td>Microsoft_SQL_Server_High_DatabaseGrowth</td>
<td>Critical</td>
<td>Yes</td>
<td>108</td>
</tr>
<tr>
<td>Microsoft SQL Server High DatabaseSpacePercentUsed</td>
<td>Microsoft_SQL_Server_High_DatabaseSpacePercentUsed</td>
<td>Critical</td>
<td>Yes</td>
<td>110</td>
</tr>
<tr>
<td>Microsoft SQL Server High FilegroupPercentUsed</td>
<td>Microsoft_SQL_Server_High_FilegroupPercentUsed</td>
<td>Critical</td>
<td>Yes</td>
<td>111</td>
</tr>
<tr>
<td>Microsoft SQL Server High LogSpacePercentUsed</td>
<td>Microsoft_SQL_Server_High_LogSpacePercentUsed</td>
<td>Critical</td>
<td>Yes</td>
<td>112</td>
</tr>
</tbody>
</table>

**Microsoft SQL Server High DatabaseGrowth indication**

Occurs when the growth in used database space exceeds the specified threshold. Use this indication to monitor the health of the database growth for each database for Microsoft SQL Server.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

```
<application_label>: The growth of the Microsoft SQL Server database <MSSQLDatabaseName> on server <MSSQLServerName> has exceeded the predefined threshold of <MSSQLDatabaseGrowthPercentThreshold> percent and is currently at <MSSQLDatabaseGrowthPercentActual> percent.
```

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.
You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**
  - The registered object’s Tivoli Management Environment (TME) class.

- **application_label**
  - The registered object’s Tivoli Management Environment (TME) label.

- **application_oid**
  - The registered object’s Tivoli Management Environment (TME) object identifier.

- **application_version**
  - The managed resource’s version; for example, v7, or 2k.

- **MSSQLCurrentDatabaseSize**
  - The current size of the selected Microsoft SQL Server database in MB.

- **MSSQLDatabaseGrowthPercentActual**
  - The actual database growth percent for the selected Microsoft SQL Server database.

- **MSSQLDatabaseGrowthPercentThreshold**
  - The specified threshold amount set for Database Growth percentage for the Microsoft SQL Server database.

- **MSSQLDatabaseName**
  - The name of the Microsoft SQL Server database.

- **MSSQLLastDatabaseSize**
  - The size of the selected Microsoft SQL Server database as determined by the last resource model cycle.

- **MSSQLServerName**
  - The name of the Microsoft SQL Server server.

This indication has the following threshold:

- Microsoft SQL Server High DatabaseGrowth

For more information about this threshold, see “Thresholds” on page 114.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>4</td>
</tr>
<tr>
<td>Holes</td>
<td>1</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.
Microsoft SQL Server High DatabaseSpacePercentUsed indication

Occurs when the used database space percent exceeds the specified threshold. Use this indication to monitor the health of the database space for each Microsoft SQL Server database.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

<application_label>: The percentage of used database space for the Microsoft SQL Server database <MSSQLDatabaseName> on server <MSSQLServerName> has exceeded the predefined threshold of <MSSQLDatabaseSpacePercentUsedThreshold> percent and is currently at <MSSQLDatabaseSpacePercentUsedActual> percent.

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

application_class
The registered object’s Tivoli Management Environment (TME) class.

application_label
The registered object’s Tivoli Management Environment (TME) label.

application_oid
The registered object’s Tivoli Management Environment (TME) object identifier.

application_version
The managed resource’s version; for example, v7, or 2k.

MSSQLDatabaseFileNames
The names of the files, semicolon-separated, that make up the selected Microsoft SQL Server database.

MSSQLDatabaseMaxGrowthSize
The maximum size, including growth, for the selected Microsoft SQL Server database.

MSSQLDatabaseName
The name of the Microsoft SQL Server database.

MSSQL_Database_Size
The current size of the selected Microsoft SQL Server database.

MSSQLDatabaseSpacePercentUsedActual
The percentage of space used based on the current size and maximum size for the selected Microsoft SQL Server database.

MSSQLDatabaseSpacePercentUsedThreshold
The specified threshold set for the percentage of database space used for the selected Microsoft SQL Server database.

MSSQLServerName
The instance name of Microsoft SQL Server.

This indication has the following threshold:

- Microsoft SQL Server High DatabaseSpacePercent
For more information about this threshold, see “Thresholds” on page 114.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>5</td>
</tr>
<tr>
<td>Holes</td>
<td>2</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Microsoft SQL Server High FilegroupPercentUsed indication**

Occurs when the used filegroup space percent exceeds the specified threshold. Use this indication to monitor the health of the filegroup space for each filegroup for Microsoft SQL Server.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

\[<\text{application\_label}>]: The percentage of used filegroup space for the Microsoft SQL Server filegroup \(<\text{MSSQLFilegroupName}>\) on server \(<\text{MSSQLServerName}>\) has exceeded the predefined threshold of \(<\text{MSSQLFilegroupSpacePercentUsedThreshold}>\) percent and is currently at \(<\text{MSSQLFilegroupSpacePercentUsedActual}>\) percent.

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application\_class**
  The registered object’s Tivoli Management Environment (TME) class.

- **application\_label**
  The registered object’s Tivoli Management Environment (TME) label.

- **application\_oid**
  The registered object’s Tivoli Management Environment (TME) object identifier.

- **application\_version**
  The managed resource’s version; for example, v7, or 2k.

- **MSSQLDatabaseName**
  The name of the Microsoft SQL Server database.

- **MSSQLFilegroupID**
  The identification number of the Microsoft SQL Server database filegroup.
**MSSQLFilegroupMaximumGrowthSize**
The maximum size of the specified Microsoft SQL Server database filegroup.

**MSSQLFilegroupName**
The name of the selected Microsoft SQL Server database filegroup.

**MSSQL_Filegroup_Size**
The current size of the selected Microsoft SQL Server database filegroup.

**MSSQLFilegroupSpacePercentUsedActual**
The actual percentage of space used based on the current size and maximum size of the selected Microsoft SQL Server database filegroup.

**MSSQLFilegroupSpacePercentUsedThreshold**
The specified threshold set for the percentage of filegroup space used for the selected Microsoft SQL Server database filegroup.

**MSSQLServerName**
The instance name of Microsoft SQL Server.

This indication has the following threshold:

- Microsoft SQL Server High FilegroupPercent

For more information about this threshold, see “Thresholds” on page 114.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>5</td>
</tr>
<tr>
<td>Holes</td>
<td>2</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Microsoft SQL Server High LogSpacePercentUsed indication**
Occurs when the used log space percentage exceeds the specified threshold. Use this indication to monitor the health of the log space for each database for Microsoft SQL Server.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

<application_label>: The percentage of used log space for the Microsoft SQL Server database <MSSQLDatabaseName> on server <MSSQLServerName> has exceeded the predefined threshold of <MSSQLLogSpacePercentUsedThreshold> and is currently at <MSSQLLogSpacePercentUsedActual>

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.
You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**
  - The registered object’s Tivoli Management Environment (TME) class.

- **application_label**
  - The registered object’s Tivoli Management Environment (TME) label.

- **application_oid**
  - The registered object’s Tivoli Management Environment (TME) object identifier.

- **application_version**
  - The managed resource’s version; for example, v7, or 2k.

- **MSSQLDatabaseName**
  - The name of the Microsoft SQL Server database.

- **MSSQLLogFileName**
  - The name of the files, semicolon-separated, that make up the selected Microsoft SQL Server database log.

- **MSSQLLogMaximumGrowthSize**
  - The maximum size, including growth, of the selected Microsoft SQL Server log.

- **MSSQL_Log_File_Size**
  - The current size of the Microsoft SQL Server database log.

- **MSSQLLogSpacePercentUsedActual**
  - The percentage of space used based on the current log size and maximum size of the Microsoft SQL Server database log.

- **MSSQLLogSpacePercentUsedThreshold**
  - The specified threshold for the log space percentage of the selected Microsoft SQL Server database log file.

- **MSSQLServerName**
  - The instance name of Microsoft SQL Server.

This indication has the following threshold:

- Microsoft SQL Server High LogSpacePercent

For more information about this threshold, see “Thresholds” on page 114.

The following table describes the default settings for this indication:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>5</td>
</tr>
<tr>
<td>Holes</td>
<td>2</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>
**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Thresholds**

The following table lists the thresholds that can be set for the Space Usage resource model. For each threshold it shows the name, a short description, and the default value:

<table>
<thead>
<tr>
<th>Threshold</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft_SQL_Server High</td>
<td>A Microsoft SQL Server High DatabaseGrowth indication occurs if the percentage of growth for a Microsoft SQL Server database exceeds this value.</td>
<td>10</td>
</tr>
<tr>
<td>DatabaseGrowth</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Microsoft_SQL_Server High</td>
<td>A Microsoft SQL Server High DatabaseSpacePercentUsed indication occurs if the used database space percentage for a Microsoft SQL Server database exceeds this value.</td>
<td>85</td>
</tr>
<tr>
<td>DatabaseSpacePercentUsed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Microsoft_SQL_Server High</td>
<td>A Microsoft SQL Server High FilegroupPercentUsed indication occurs if the used filegroup space percentage for a Microsoft SQL Server filegroup exceeds this value.</td>
<td>85</td>
</tr>
<tr>
<td>FilegroupPercentUsed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Microsoft_SQL_Server High</td>
<td>A Microsoft SQL Server High LogSpacePercentUsed indication occurs if the used log space percentage for a Microsoft SQL Server database exceeds this value.</td>
<td>85</td>
</tr>
<tr>
<td>LogSpacePercentUsed</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Parameters**

The following table lists the parameters that can be set for the Space Usage resource model. The table shows the name, a short description, and the default value for each parameter:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exclude Databases</td>
<td>Type the names of any databases to exclude from monitoring.</td>
<td>Blank (monitors all databases)</td>
</tr>
<tr>
<td></td>
<td>When this parameter is blank, the resource model monitors all databases.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>You can type more than one database name. Separate multiple database names with spaces.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Wildcard characters are not recognized.</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Default value</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Refresh Database Frequency</td>
<td>Defines how often the resource model receives a new list of databases from the Microsoft SQL Server. This value represents the number of resource model cycles. For example, if the resource model cycle time is 60 seconds and the refresh frequency is set to 5, the resource model gets the list of databases every 5 minutes. When a database is added, it is monitored the next time the list of databases is retrieved.</td>
<td>5</td>
</tr>
<tr>
<td>Unlimited File Monitoring Flag</td>
<td>Specifies whether unlimited space usage monitoring for filegroups is enabled or not. If enabled, the resource model uses the total capacity of the disk that the files reside on to determine space usage and health. If not enabled and any of the database files have a maximum size of &quot;Unlimited,&quot; then the filegroup space usage is &quot;healthy.&quot;</td>
<td>Not Enabled</td>
</tr>
</tbody>
</table>

**Tasks and built-in actions**
None

**Logging**
You can log data for the properties of the managed resource listed in the following table. The table shows the context of the managed resource and the properties that the resource model logs for the IBM Tivoli Monitoring Web Health Console.
<table>
<thead>
<tr>
<th>Managed resource</th>
<th>Context</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSSQLDatabase</td>
<td>Storage</td>
<td><strong>MSSQLServer.MSSQLServerName</strong>&lt;sup&gt;*&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The name of the Microsoft SQL Server server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>MSSQLDatabase.MSSQLDatabaseName</strong>&lt;sup&gt;*&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The name of the Microsoft SQL Server database.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>MSSQL_Database_Growth_Percent_Actual</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The actual database growth percent for the selected Microsoft SQL Server database.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>MSSQLDatabaseMaximumGrowthSize</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The maximum size, including growth, for the selected Microsoft SQL Server database.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>MSSQL_Database_Size</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The current size of the selected Microsoft SQL Server database.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>MSSQL_Database_Space_Percent_Used_Actual</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The percentage of space used based on the current size and maximum size for the selected Microsoft SQL Server database.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>MSSQLLogFileName</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The names of the files that make up the selected Microsoft SQL Server database log. Log file names are separated by semicolons.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>MSSQL_Log_File_Size</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The current size of the Microsoft SQL Server database log.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>MSSQLLogMaximumGrowthSize</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The maximum size, including growth, of the selected Microsoft SQL Server log.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>MSSQL_Log_Space_Percent_Used_Actual</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The percentage of space used based on the current log size and maximum size of the Microsoft SQL Server database log.</td>
</tr>
</tbody>
</table>

**Note:** An asterisk (*) denotes a key property.
### Managed resource Context Properties

<table>
<thead>
<tr>
<th>Managed resource</th>
<th>Context</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSSQLFilegroup</td>
<td>Storage</td>
<td><strong>MSSQLServer.MSSQLServerName</strong>&lt;sup&gt;*&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The instance name of Microsoft SQL Server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>MSSQLDatabase.MSSQLDatabaseName</strong>&lt;sup&gt;*&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The name of the Microsoft SQL Server database.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>MSSQLFilegroup.MSSQLFilegroupName</strong>&lt;sup&gt;*&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The name of the selected Microsoft SQL Server database filegroup.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>MSSQLFilegroupMaximumGrowthSize</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The maximum size of the specified Microsoft SQL Server database filegroup.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>MSSQL_Filegroup_Size</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The current size of the selected Microsoft SQL Server database filegroup.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>MSSQL_Filegroup_Space_Percent_Used_Actual</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The actual percentage of space used based on the current size and maximum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>size of the selected Microsoft SQL Server database filegroup.</td>
</tr>
</tbody>
</table>

**Note:** An asterisk (*) denotes a key property.

### Return codes

Table 2 on page 7 contains a listing of resource model return codes, their description, and what action you can take to resolve each code. The return code number is displayed in the IBM Tivoli Monitoring Web Health Console status field, or by using the **wdmlseng** command.

### CLI example

```bash
wdmeditprf -P $Prof1 -edit Microsoft_SQL_Server_Space_Usage \
-t Microsoft_SQL_Server_High_FilegroupPercent 85.000000 \
-t Microsoft_SQL_Server_High_DatabaseSpacePercent 85.000000 \
-t Microsoft_SQL_Server_High_LogSpacePercent 85.000000 \
-t Microsoft_SQL_Server_High_DatabaseGrowth 10.000000 \
-AddPar Refresh_Database_Frequency "5" \ 
-AddPar Unlimited_File_Monitoring_Flag 0 \ 
-AddPar Exclude_Databases "0" \ 
-e Microsoft_SQL_Server_High_LogSpacePercentUsed \ 
  -o 5 -h 2 -severity CRITICAL -NoSendTBSM -"$MyTec" \ 
-e Microsoft_SQL_Server_High_DatabaseSpacePercentUsed \ 
  -o 5 -h 2 -severity CRITICAL -NoSendTBSM -"$MyTec" \ 
-e Microsoft_SQL_Server_High_FilegroupPercentUsed \ 
  -o 5 -h 2 -severity CRITICAL -NoSendTBSM -"$MyTec" \ 
-e Microsoft_SQL_Server_High_DatabaseGrowth \ 
  -o 4 -h 1 -severity CRITICAL -NoSendTBSM -"$MyTec"
```
Users/Transactions

Description
Monitors user and transaction activity in the Microsoft SQL Server. Ensures the server is appropriately sized and configured.

The resource model monitors the following activities:
- Blocked and long-running processes
- Transaction performance
- User connection load

By default, this resource model monitors all databases associated with the Microsoft SQL Server for transaction activity. You can exclude databases from monitoring by using the Exclude Databases parameter.

User, process and transaction management is an important component of your environment. Proper management can have a large impact on the performance of SQL Server and the consistency of your databases.

Resource model overview

<table>
<thead>
<tr>
<th>Internal name</th>
<th>Microsoft_SQL_Server_Users_Transactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>Microsoft SQL Server</td>
</tr>
<tr>
<td>Indications</td>
<td></td>
</tr>
<tr>
<td><strong>Microsoft SQL Server Active Transactions too high</strong> occurs when the number of active transactions exceeds the <strong>Active Transactions upper bound</strong> threshold.</td>
<td></td>
</tr>
<tr>
<td><strong>Microsoft SQL Server Blocked Processes too high</strong> occurs when the number of blocked processes exceeds the <strong>Blocked Processes upper bound</strong> threshold.</td>
<td></td>
</tr>
<tr>
<td><strong>Microsoft SQL Server Logins Per Second too high</strong> occurs when the number of user logins per second to the Microsoft SQL Server exceeds the <strong>Logins Per Second upper bound</strong> threshold.</td>
<td></td>
</tr>
<tr>
<td><strong>Microsoft SQL Server Logouts Per Second too high</strong> occurs when the number of logouts per second exceeds the <strong>Logouts Per Second upper bound</strong> threshold.</td>
<td></td>
</tr>
<tr>
<td><strong>Microsoft SQL Server Long Running Processes too high</strong> occurs when the number of long running processes exceeds the <strong>Long Running Process Count upper bound</strong> threshold.</td>
<td></td>
</tr>
<tr>
<td><strong>Microsoft SQL Server Percent User Connections Used too high</strong> occurs when the percentage of user connections exceeds the <strong>Percent User Connections Used upper bound</strong> threshold.</td>
<td></td>
</tr>
<tr>
<td><strong>Microsoft SQL Server Transactions Per Second too high</strong> occurs when the number of transactions per second exceeds the <strong>Transactions Per Second upper bound</strong> threshold.</td>
<td></td>
</tr>
<tr>
<td>Tasks and built-in actions</td>
<td>None</td>
</tr>
<tr>
<td>Default cycle time</td>
<td>600 seconds</td>
</tr>
</tbody>
</table>

Target managed resource
MicrosoftSQLServer
Indications and events

The following table summarizes the indications for this resource model, the events associated with the indications, the default severity of the events, and where to find a detailed description of the indication.

<table>
<thead>
<tr>
<th>Indication Generated event</th>
<th>Default severity</th>
<th>Clearing events</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft SQL Server Active Transactions too high</td>
<td>Microsoft_SQL_Server_High_ActiveTransactions</td>
<td>Critical</td>
<td>Yes</td>
</tr>
<tr>
<td>Microsoft SQL Server Blocked Processes too high</td>
<td>Microsoft_SQL_Server_High_BlockedProcesses</td>
<td>Warning</td>
<td>Yes</td>
</tr>
<tr>
<td>Microsoft SQL Server Logins Per Second too high</td>
<td>Microsoft_SQL_Server_High_LoginsPerSec</td>
<td>Critical</td>
<td>Yes</td>
</tr>
<tr>
<td>Microsoft SQL Server Logouts Per Second too high</td>
<td>Microsoft_SQL_Server_High_LogoutsPerSec</td>
<td>Critical</td>
<td>Yes</td>
</tr>
<tr>
<td>Microsoft SQL Server Long Running Processes too high</td>
<td>Microsoft_SQL_Server_High_LongRunningProcesses</td>
<td>Warning</td>
<td>Yes</td>
</tr>
<tr>
<td>Microsoft SQL Server Percent User Connections Used too high</td>
<td>Microsoft_SQL_Server_High_PctUserConnectionsUsed</td>
<td>Critical</td>
<td>Yes</td>
</tr>
<tr>
<td>Microsoft SQL Server Transactions Per Second too high</td>
<td>Microsoft_SQL_Server_High_TransactionsPerSec</td>
<td>Critical</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Microsoft SQL Server Active Transactions too high indication**

Occurs when the number of active transactions exceeds the Active Transactions upper bound threshold.

Active transactions are transactions that are currently executing. Monitoring active transactions is useful to determine the efficiency of the database. A high number of active transactions can indicate that certain transactions are never completing.

During an active transaction, no other transaction can modify the data held by the active transaction. While you are modifying data, you hold an exclusive lock on that data. Conversely, you cannot read data held by another transaction because another transaction holds an exclusive lock on the data that prevents you from reading it. Keeping a low level of active transactions assists in keeping requested data accessible.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:
If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**: The registered object’s Tivoli Management Environment (TME) class.
- **application_label**: The registered object’s Tivoli Management Environment (TME) label.
- **application_oid**: The registered object’s Tivoli Management Environment (TME) object identifier.
- **application_version**: The managed resource’s version; for example, v7, or 2k.
- **MSSQLActiveTransactions**: The current number of active transactions in the Microsoft SQL Server. Active transactions are transactions that are currently executing.
- **MSSQLDatabaseName**: The name of the selected Microsoft SQL Server database.
- **MSSQLServerName**: The instance name of Microsoft SQL Server.
- **UpperBound**: The value set as the maximum allowable for the specified threshold.

This indication has the following threshold:

- **Active Transactions upper bound**

For more information about this threshold, see "Thresholds" on page 129.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>1</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.
**Microsoft SQL Server Blocked Processes too high indication**

Occurs when the number of blocked processes exceeds the **Blocked Processes upper bound** threshold.

A blocked process is a process that requests a lock, but waits for a competing lock to be released by another process. A high number of blocked processes can cause poor performance due to transaction delays. Use this indication to check how many processes are being held up through lock contention rather than hardware performance issues.

Blocked processes are different from deadlocks. A deadlock cannot be resolved without killing one of the processes. A blocked process is simply waiting for a lock to be released by another process. Blocked processes still require attention, since performance can suffer, and every system on the network can appear to hang.

Use the following guidelines to avoid or resolve blocking problems:

1. Keep transactions as small as possible. The quicker the transaction, the shorter the time that locks are held.
2. Keep the entire transaction within one batch whenever possible.
3. Stagger competing transactions or alter the order in which the transactions run.
4. Avoid pauses in transactions for use input.
5. Use the LOCK_TIMEOUT session option to configure SQL Server to automatically cancel out of a locking situation.
6. Process all rows as quickly as possible when processing result sets. An application that stops processing results can prevent the server from sending more results. This stalls the scanning process and requires locks to be held much longer.
7. Use cursors with optimum concurrency control for browsing applications.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

```
<application_label>: Microsoft SQL Server <MSSQLServerName> - Number of Blocked Processes is <MSSQLBlockedProcesses>, which exceeds threshold of <UpperBound>.
```

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**
  The registered object’s Tivoli Management Environment (TME) class.

- **application_label**
  The registered object’s Tivoli Management Environment (TME) label.

- **application_oid**
  The registered object’s Tivoli Management Environment (TME) object identifier.

- **application_version**
  The managed resource’s version; for example, v7, or 2k.
**MSSQLBlockedProcesses**
The total number of processes in the Microsoft SQL Server that are currently blocked. A blocked process is a process that requests a lock, but waits for a competing lock to be released by another process.

**MSSQLServerName**
The instance name of Microsoft SQL Server.

**UpperBound**
The value set as the maximum allowable for the specified threshold.

This indication has the following threshold:
- Blocked Processes upper bound

For more information about this threshold, see “Thresholds” on page 129.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>2</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Microsoft SQL Server Logins Per Second too high indication**
Occurs when the number of user logins per second to the Microsoft SQL Server exceeds the Logins Per Second upper bound threshold.

A login is a new connection event since a trace was started. A high number of logins per second can indicate a need for configuration changes.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

<application_label>: Microsoft SQL Server <MSSQLServerName> - Logins Per Second is <MSSQLLoginsPerSec>, which exceeds threshold of <UpperBound>.

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

*application_class*

The registered object’s Tivoli Management Environment (TME) class.
**application_label**
The registered object’s Tivoli Management Environment (TME) label.

**application_oid**
The registered object’s Tivoli Management Environment (TME) object identifier.

**application_version**
The managed resource’s version; for example, v7, or 2k.

**MSSQLLoginsPerSec**
The number of user logins to the Microsoft SQL Server per second. A login is a new connection event since a trace was started.

**MSSQLServerName**
The instance name of Microsoft SQL Server.

**UpperBound**
The value set as the maximum allowable for the specified threshold.

This indication has the following thresholds:

- Logins Per Second upper bound

For more information about this threshold, see "Thresholds" on page 129.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>1</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Microsoft SQL Server Logouts Per Second too high indication**
Occurs when the number of logouts per second from the Microsoft SQL Server exceeds the Logouts Per Second upper bound threshold.

A logout is a user disconnect from the server since a trace was started. A high number of logouts per second can indicate a need for configuration changes.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

```
<application_label>: Microsoft SQL Server <MSSQLServerName> - Logouts Per Second is <MSSQLLogoutsPerSec>, which exceeds threshold of <UpperBound>.
```

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.
You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**
  The registered object’s Tivoli Management Environment (TME) class.

- **application_label**
  The registered object’s Tivoli Management Environment (TME) label.

- **application_oid**
  The registered object’s Tivoli Management Environment (TME) object identifier.

- **application_version**
  The managed resource’s version; for example, v7, or 2k.

- **MSSQLLogoutsPerSec**
  The number of user logouts from the Microsoft SQL Server per second. A logout is a user disconnect from the server since a trace was started.

- **MSSQLServerName**
  The instance name of Microsoft SQL Server.

- **UpperBound**
  The value set as the maximum allowable for the specified threshold.

This indication has the following threshold:

- Logouts Per Second upper bound

For more information about this threshold, see "Thresholds" on page 129.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems</td>
<td>No</td>
</tr>
<tr>
<td>Manager</td>
<td></td>
</tr>
<tr>
<td>Occurrences</td>
<td>1</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Microsoft SQL Server Long Running Processes too high indication**

Occurs when the number of long-running processes for the Microsoft SQL Server exceeds the **Long Running Process Count upper bound** threshold.

A high number of long-running processes can indicate issues and bottlenecks in the SQL or database design. Long-running processes can degrade performance by holding locks for an extended period of time, preventing other processes from accessing the locked data. Transaction information for each process is recorded in
each database’s transaction log. Long-running transactions cause problems for the system administrator who is attempting to back up and prune the transaction log since the transactions do not complete.

You can determine the identity of long-running processes and how long their transactions are running with the following commands:

- **DBCC OPENTRAN** provides information about the oldest active transaction in a database, such as the spid of the process that initiated the transaction, the user ID, the name of the transaction, the Log Sequence Number (LSN) of the page containing the initial `BEGIN TRAN` statement for the transaction, and the transaction start time.

- **sp_who** provides information about the current Microsoft SQL Server users and processes. You can filter returned data to display only active processes by specifying `ACTIVE`.

- The **sysprocesses** table holds information about processes running on Microsoft SQL Server. Processes include client and system processes. The **sysprocesses** table is stored in the master database.

After identifying long-running processes, you can evaluate the necessity of these processes. If a process is blocking other important processes with locks, or is executing a query that uses necessary system resources, you can consider killing the process. The **KILL** command terminates a process. Use **KILL** very carefully, especially when critical processes are running. System processes and processes running an extended stored procedure cannot be terminated, and you cannot kill your own process. Other processes that you do not want to kill are:

- **AWAITING COMMAND**
- **CHECKPOINT SLEEP**
- **LAZY WRITER**
- **LOCK MONITOR**
- **SELECT**
- **SIGNAL HANDLER**

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

```<application_label>: Microsoft SQL Server <MSSQLServerName> - Number of Long Running Processes is <MSSQLLongRunningProcessCount>, which exceeds threshold of <UpperBound>.``` 

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**
  The registered object’s Tivoli Management Environment (TME) class.

- **application_label**
  The registered object’s Tivoli Management Environment (TME) label.
application_oid
The registered object’s Tivoli Management Environment (TME) object identifier.

application_version
The managed resource’s version; for example, v7, or 2k.

MSSQLLongRunningProcessCount
The total number of long running processes in the selected Microsoft SQL Server.

MSSQLServerName
The instance name of Microsoft SQL Server.

UpperBound
The value set as the maximum allowable for the specified threshold.

This indication has the following thresholds:
- Long Running Process Count upper bound
- Long Running Process Duration

Additional Information: The Long Running Process Duration threshold does not trigger an indication. Use this threshold to define how many seconds a process runs before it is considered to be a long-running process.

For more information about these thresholds, see “Thresholds” on page 129.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>1</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

Note: If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

Microsoft SQL Server Percent User Connections Used too high indication
Occurs when the current number of user connections as a percentage of the available number of user connections exceeds the Percent User Connections Used upper bound threshold.

User connections specifies the number of concurrent users that are allowed on Microsoft SQL Server. Since each user connection consumes 40 KB of memory space, a high number of user connections can impact throughput and cause a performance slow-down. Use this metric to gain an overview of high access periods and to warn you of impending availability problems.

A high percentage of user connections can also indicate a potential shortage of allocated user connections. If a maximum number of user connections is set and
reached, Microsoft SQL Server rejects login requests. This competition for connection results in contention between users.

If the number of user connections is high, consider doing any of the following:

- Control the number of user connections by doing one of the following:
  - Enable Microsoft SQL Server to automatically add user connections as needed. Do this through the Microsoft SQL Server SQL Server Properties window by setting the user connections configuration option to 0. Tivoli recommends using this option. If you specify a value other than 0, Microsoft SQL Server allocates memory for those connections at startup, which burns up portions of the memory pool.
  - OR —
    - Specify a maximum number of user connections. Set this amount to the expected maximum number of concurrent users. Microsoft SQL Server rejects new login requests after reaching the maximum number of user connections when this value is set. After changing the maximum number of user connections, you must restart Microsoft SQL Server.

- Ensure that applications are not using multiple connections when using only one is sufficient.
- Stagger access requirements.
- Increase the available memory.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

```
<application_label>: Microsoft SQL Server <MSSQLServerName> - Percent User Connections Used is <MSSQLPctUserConnectionsUsed>, which exceeds threshold of <UpperBound>. There are <MSSQLUserConnections> actual User Connections.
```

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:

- **application_class**
  The registered object’s Tivoli Management Environment (TME) class.

- **application_label**
  The registered object’s Tivoli Management Environment (TME) label.

- **application_oid**
  The registered object’s Tivoli Management Environment (TME) object identifier.

- **application_version**
  The managed resource’s version; for example, v7, or 2k.

- **MSSQLPctUserConnectionsUsed**
  The percentage of user connections currently used. The percentage is calculated as the number of used user connections divided by the total number of available user connections.

- **MSSQLServerName**
  The instance name of Microsoft SQL Server.
MSSQLUserConnections
The number of concurrent users that are on Microsoft SQL Server.

UpperBound
The value set as the maximum allowable for the specified threshold.

This indication has the following threshold:
• Percent User Connections Used upper bound

For more information about this threshold, see “Thresholds” on page 129.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>1</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

Note: If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

Microsoft SQL Server Transactions Per Second too high indication
Occurs when the number of transactions per second exceeds the Transactions Per Second upper bound threshold. A transaction is one or more database operations combined into a single operation action that is either fully committed or not performed at all.

Transaction rate is affected by general system performance and resource constraints, such as I/O, number of users, cache size, and complexity of requests. A high rate of transactions can indicate that some transactions are not completing.

Using stored procedures in transactions speeds transaction processing because SQL code in stored procedures runs locally on the server. The transactions can finish much faster without the network traffic required by the transactions.

When the indication occurs often enough to trigger an event, the event delivers a message to the Tivoli Enterprise Console in the following format:

<application_label>: Microsoft SQL Server <MSSQLServerName>, Database <MSSQLDatabaseName> - Transactions Per Second is <MSSQLTransactionsPerSec>, which exceeds threshold of <UpperBound>.

If you have Tivoli Business Systems Manager configured for your system, Tivoli Enterprise Console forwards the message to Tivoli Business Systems Manager.

You can check the health of this resource model in the IBM Tivoli Monitoring Web Health Console. For more information, see the IBM Tivoli Monitoring Web Health Console documentation.

The indication has the following attributes:
application_class
The registered object’s Tivoli Management Environment (TME) class.

application_label
The registered object’s Tivoli Management Environment (TME) label.

application_oid
The registered object’s Tivoli Management Environment (TME) object identifier.

application_version
The managed resource’s version; for example, v7, or 2k.

MSSQLDatabaseName
The name of the selected Microsoft SQL Server database.

MSSQLServerName
The instance name of Microsoft SQL Server.

MSSQLTransactionsPerSec
The number of user transactions in the Microsoft SQL Server per second.

UpperBound
The value set as the maximum allowable for the specified threshold.

This indication has the following threshold:
• Transactions Per Second upper bound

For more information about this threshold, see “Thresholds” on page 129.

The following table describes the default settings for this indication.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send indications to Tivoli Enterprise Console</td>
<td>Yes</td>
</tr>
<tr>
<td>Send indications to Tivoli Business Systems Manager</td>
<td>No</td>
</tr>
<tr>
<td>Occurrences</td>
<td>1</td>
</tr>
<tr>
<td>Holes</td>
<td>0</td>
</tr>
<tr>
<td>Associated tasks and built-in actions</td>
<td>None</td>
</tr>
</tbody>
</table>

**Note:** If you have Tivoli Business Systems Manager configured for your systems, Tivoli Enterprise Console automatically forwards events to Tivoli Business Systems Manager. Do not change the configuration of the indication to send events to Tivoli Business Systems Manager.

**Thresholds**

The following table lists the thresholds that can be set for the Users/Transactions resource model. For each threshold it shows the name, a short description, and the default value:

<table>
<thead>
<tr>
<th>Threshold</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Transactions upper bound</td>
<td>A Microsoft SQL Server Active Transactions too high indication occurs if the number of active transactions exceeds this value.</td>
<td>200</td>
</tr>
<tr>
<td>Threshold</td>
<td>Description</td>
<td>Default value</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Blocked Processes upper bound</td>
<td>A Microsoft SQL Server Blocked Processes too high indication occurs if the number of blocked processes exceeds this value.</td>
<td>2</td>
</tr>
<tr>
<td>Logins Per Second upper bound</td>
<td>A Microsoft SQL Server Logins Per Second too high indication occurs if the number of logins per second exceeds this value.</td>
<td>10</td>
</tr>
<tr>
<td>Logouts Per Second upper bound</td>
<td>A Microsoft SQL Server Logouts Per Second too high indication occurs if the number of logouts per second exceeds this value.</td>
<td>100</td>
</tr>
<tr>
<td>Long Running Process Count upper bound</td>
<td>A Microsoft SQL Server Long Running Processes too high indication occurs if the number of long-running processes exceeds this value.</td>
<td>3</td>
</tr>
<tr>
<td>Long Running Process Duration</td>
<td>This threshold defines a long-running process. The value represents the number of seconds a process can run before it is considered to be a long-running process. Use this threshold in combination with the Long Running Process Count upper bound threshold. These two thresholds work in combination to define the criteria to trigger a Microsoft SQL Server Long Running Processes too high indication. First, clarify the long-running process duration with this threshold. Then use the Long Running Process Count upper bound threshold to specify how many long-running processes trigger an indication. All long-running processes are logged, even if an indication is not triggered.</td>
<td>5</td>
</tr>
<tr>
<td>Percent User Connections Used upper bound</td>
<td>A Microsoft SQL Server Percent User Connections Used too high indication occurs if the percentage of user connections in use exceeds this value.</td>
<td>80</td>
</tr>
<tr>
<td>Transactions Per Second upper bound</td>
<td>A Microsoft SQL Server Transactions Per Second too high indication occurs if the number of transactions per second exceeds this value.</td>
<td>200</td>
</tr>
</tbody>
</table>

**Parameters**

The following table lists the parameters that can be set for the Users/Transactions resource model. The table shows the name, a short description, and the default value for each parameter.
Parameter Description Default value

Exclude Databases Type any database names that you want to exclude from monitoring active transactions and transactions per second.

When this parameter is blank, the resource model monitors all databases associated with the Microsoft SQL Server for transaction activity.

You can type more than one database name. Separate multiple database names with spaces.

Wildcard characters are not recognized. Blank (monitors all databases)

Tasks and built-in actions

None

Logging

You can log data for the properties of the managed resource listed in the following table. The table shows the context of the managed resource and the properties that the resource model logs for the IBM Tivoli Monitoring Web Health Console.

<table>
<thead>
<tr>
<th>Managed resource</th>
<th>Context</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSSQLDatabase</td>
<td>Transaction Performance</td>
<td>MSSQLServer.MSSQLServerName* The instance name of Microsoft SQL Server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQLDatabase.MSSQLDatabaseName* The name of the selected Microsoft SQL Server database.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQLVersion The version number, v7 or 2k, of the installed Microsoft SQL Server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_Active_Transactions The current number of active transactions in the Microsoft SQL Server. Active transactions are transactions that are currently executing.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_Transactions_Rate The number of user transactions in the Microsoft SQL Server per second.</td>
</tr>
<tr>
<td>MSSQLServer</td>
<td>Blocked Process Performance</td>
<td>MSSQLServer.MSSQLServerName* The instance name of Microsoft SQL Server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQLVersion The version number, v7 or 2k, of the installed Microsoft SQL Server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_BlockedProcesses The total number of processes in the Microsoft SQL Server that are currently blocked. A blocked process is a process that requests a lock, but waits for a competing lock to be released by another process.</td>
</tr>
<tr>
<td>Managed resource</td>
<td>Context</td>
<td>Properties</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td>MSSQLServer</td>
<td>Long Running Process</td>
<td>MSSQLServer.MSSQLServerName*</td>
</tr>
<tr>
<td></td>
<td>Performance</td>
<td>The instance name of Microsoft SQL Server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQLVersion</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The version number, v7 or 2k, of the installed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Microsoft SQL Server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_LRP Blocked</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The Microsoft SQL Server process ID of the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Microsoft SQL Server process that is blocking</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the long-running process.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_LRP_Duration</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The time duration (in seconds) that a long-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>running process has been executing.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_LRP_Spid</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The system process ID for a long-running process</td>
</tr>
<tr>
<td></td>
<td></td>
<td>on the Microsoft SQL Server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_LRP_HostName</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The host name of the workstation on which the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>long-running process exists.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_LRP_Status</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The current status of this long-running process</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(sleeping, background, runnable, etc.).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_LRP_Command</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The command currently being executed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_LRP_ProgramName</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The name of the application program that issued</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the command.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_LRP_LoginName</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The login name as defined by a Microsoft SQL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Server user account.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSSQL_LRP_Nt_User</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The Windows user name for the process (if using</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Windows Authentication) or a trusted connection.</td>
</tr>
</tbody>
</table>

* MSSQLServerName is a property in IBM Tivoli Monitoring for Databases: Microsoft SQL Server: Reference Guide.
### Managed resource Context Properties

<table>
<thead>
<tr>
<th>Managed resource</th>
<th>Context</th>
<th>Properties</th>
</tr>
</thead>
</table>
| MSSQLServer      | User Performance| **MSSQLServer.MSSQLServerName**<sup>*</sup>  
  The instance name of Microsoft SQL Server. |
|                  |                 | **MSSQLVersion**  
  The version number, v7 or 2k, of the installed Microsoft SQL Server. |
|                  |                 | **MSSQL_Logins**  
  The number of user logins to the Microsoft SQL Server per second. |
|                  |                 | **MSSQL_Logouts**  
  The number of user logouts from the Microsoft SQL Server per second. |
|                  |                 | **MSSQL_Percent_User_Connections_Used**  
  The percentage of user connections currently used. The percentage is calculated as the number of "used" user connections divided by the total number of available user connections. |
|                  |                 | **MSSQL_User_Connections**  
  The number of concurrent users that are connected to a Microsoft SQL Server. |

**Note:** An asterisk (*) denotes a key property.

### Return codes

Table 2 on page 7 contains a listing of resource model return codes, their description, and what action you can take to resolve each code. The return code number is displayed in the IBM Tivoli Monitoring Web Health Console status field, or by using the `wdmlseng` command.

### CLI example

```
wdmeditprf -P $Prof1 -edit Microsoft_SQL_Server_Users_Transactions \
  -t High_MSSQL_ActiveTransactions 0.000000 \n  -t High_MSSQL_BlockedProcesses 0.000000 \n  -t High_MSSQL_LoginsPerSec 0.000000 \n  -t High_MSSQL_LogoutsPerSec 0.000000 \n  -t High_MSSQL_LongRunningProcess_Count 0.000000 \n  -t High_MSSQL_LongRunningProcess_Duration 0.000000 \n  -t High_MSSQL_PctUserConnectionsUsed 0.000000 \n  -t High_MSSQL_TransactionsPerSec 0.000000 \n  -e Microsoft_SQL_Server_High_LoginsPerSec \n    -o 1 -h 0 -severity CRITICAL -SendTBSM "$MyTec" \n  -e Microsoft_SQL_Server_High_ActiveTransactions \n    -o 1 -h 0 -severity CRITICAL -SendTBSM "$MyTec" \n  -e Microsoft_SQL_Server_High_LogoutsPerSec \n    -o 1 -h 0 -severity CRITICAL -SendTBSM "$MyTec" \n  -e Microsoft_SQL_Server_High_BlockedProcesses \n    -o 1 -h 0 -severity CRITICAL -SendTBSM "$MyTec" \n  -e Microsoft_SQL_Server_High_TransactionsPerSec \n    -o 1 -h 0 -severity CRITICAL -SendTBSM "$MyTec" \n  -e Microsoft_SQL_Server_High_PctUserConnectionsUsed \n    -o 1 -h 0 -severity CRITICAL -SendTBSM "$MyTec" \n  -e Microsoft_SQL_Server_High_LongRunningProcesses \n    -o 1 -h 0 -severity CRITICAL -SendTBSM "$MyTec" \n  -AddPar Exclude_Databases "testdb"
```
Chapter 3. Tasks

IBM Tivoli Monitoring for Databases: Microsoft SQL Server tasks are operations or sets of operations that perform database operations routinely. This chapter contains information about the tasks for IBM Tivoli Monitoring for Databases: Microsoft SQL Server. The section for each task describes how to configure and run each task by covering the following information:

**Description**
Purpose of the task.

**Authorization role**
Role required to run the task.

**Target endpoint**
Identifies the target on which the task is run, such as the managed node where the server is installed.

**GUI data entry fields**
List of the options for the task that are available in the Tivoli desktop graphical user interfaces. These options correspond to the parameters in the command syntax.

**CLI syntax**
Syntax for the `wruntask` command that you use to run every task from the command line. All tasks contain the following syntax:

```
wruntask -t <Task Name>
-l <Task Library Name>
-h <"Object Name"> \
-a <"Additional Parameters">
```

where:

**<Task Name>**
Name of the task to run.

**<Task Library Name>**
Name of the task library that contains the task.
ITMMicrosoftSQLServerTasks is the name of the task library for IBM Tivoli Monitoring for Databases: Microsoft SQL Server.

**<Object Name>**
Name of the object about which the task gathers data.

**<Additional Parameters>**
Additional parameters for the task, such as Date or Database Name. Some of the additional parameters might be required and some might be optional. The parameters must be typed in the order in which they appear in the syntax. If you do not want to specify a parameter, use empty quotation marks like the following:

```
-a ""
```

Use quotation marks around the name of a variable that contains spaces.

**CLI example**
A task example contains a brief description with sample syntax. The task in the following example shows how to run the `DiscoverMSSQLServers` task on the endpoint `v1172k-lcf`:
wruntask -t DiscoverMSSQLServers
-l ITMMicrosoftSQLServerTasks
-h @ManagedNode:hebergin@abc
-a v172k=1cf

Usage notes
Additional notes relevant to using the task.

See also
Additional commands relevant for using the task.

Using a Tivoli task offers the following benefits:
• You can set up tasks to run on a schedule within a Tivoli job.
• You can set up tasks to run in response to the indications for a resource model.
• You can set arguments for a task in the Tivoli desktop graphical user interface and save them to run at another time (In contrast, you must type all arguments every time you run a task in the command line.).
• Tasks appear as icons that you can click and run in the Tivoli desktop graphical user interface.

See the "Working with tasks and jobs" chapter of the IBM Tivoli Monitoring for Databases: Microsoft SQL Server User's Guide for more information on working with tasks and jobs.

The following table lists the tasks described in this chapter:

<table>
<thead>
<tr>
<th>Goal</th>
<th>Refer to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Understanding the ConfigureTECMSSQL task</td>
<td>&quot;ConfigureTECMSSQL” on page 138</td>
</tr>
<tr>
<td>Understanding the CustomizeTECNTEventAdapterMSSQL task</td>
<td>&quot;CustomizeTECNTEventAdapterMSSQL” on page 141</td>
</tr>
<tr>
<td>Understanding the DiscoverMSSQLServers task</td>
<td>&quot;DiscoverMSSQLServers” on page 144</td>
</tr>
<tr>
<td>Understanding the DumpDatabase task</td>
<td>&quot;DumpDatabase” on page 147</td>
</tr>
<tr>
<td>Understanding the DumpTransactionLog task</td>
<td>&quot;DumpTransactionLog” on page 152</td>
</tr>
<tr>
<td>Understanding the ExecuteSQL task</td>
<td>&quot;ExecuteSQL” on page 158</td>
</tr>
<tr>
<td>Understanding the MSSQLAuthentication task</td>
<td>&quot;MSSQLAuthentication” on page 161</td>
</tr>
<tr>
<td>Understanding the MSSQLTBSMDiscovery task</td>
<td>&quot;MSSQLTBSMDiscovery” on page 164</td>
</tr>
<tr>
<td>Understanding the NTEventLogging task</td>
<td>&quot;NTEventLogging” on page 166</td>
</tr>
<tr>
<td>Understanding the RebuildFragmentedIndexes task</td>
<td>&quot;RebuildFragmentedIndexes” on page 169</td>
</tr>
<tr>
<td>Understanding the SetRetrieveDatabaseOptions task</td>
<td>&quot;SetRetrieveDatabaseOptions” on page 173</td>
</tr>
<tr>
<td>Understanding the StartMSSQLServerServices task</td>
<td>&quot;StartMSSQLServerServices” on page 183</td>
</tr>
<tr>
<td>Understanding the StopMSSQLServerServices task</td>
<td>&quot;StopMSSQLServerServices” on page 187</td>
</tr>
<tr>
<td>Understanding the TDPFullBackup task</td>
<td>&quot;TDPFullBackup” on page 191</td>
</tr>
<tr>
<td>Understanding the TDPIncBackup task</td>
<td>&quot;TDPIncBackup” on page 196</td>
</tr>
</tbody>
</table>
Refer to the *IBM Tivoli Monitoring for Databases: Microsoft SQL Server User’s Guide* for information on running and customizing tasks.

Refer to the *Tivoli Management Framework Reference Manual* for information on the `wruntask` command.
**ConfigureTECMSSQL**

**Description**

Configures Tivoli Enterprise Console to handle Microsoft SQL Server events. This is done by importing new classes and new rules, installing new event sources, compiling the new rule base, then stopping and restarting the event server. The task can be executed against a Tivoli Enterprise Console server.

After entering the name of the new rule base, you can decide whether to:

- Clone a currently active rule base
  - OR -
- Create a new rule base without cloning
  - OR -
- Specify a rule base to clone

This task also provides Tivoli Enterprise Console configuration options. You can configure Tivoli Enterprise Console to receive events from the following:

- IBM Tivoli Monitoring resource models
- Tasks from the **ITMMicrosoftSQLServerTasks** task library

The task adds the class and rule set definitions of the selected configuration option to a valid rule base if the class and rule set definitions are not already defined in the specified rule base. For more information, see the *IBM Tivoli Monitoring for Databases: Microsoft SQL Server User’s Guide*.

Before you run this task, the following software must be installed:

- Tivoli Enterprise Console Event Server
- IBM Tivoli Monitoring for Databases: Microsoft SQL Server on the Tivoli management region server of the Tivoli management region on which you want to run this task

ConfigureTECMSSQL does the following, depending on the options you choose:

- Creates a rule base in the directory where Tivoli Enterprise Console is installed
- Provides Tivoli Enterprise Console configuration for the specified options
- Copies a specified rule base into the newly created rule base
- Loads the rule base
- Restarts the event server

**Authorization role**

mssqlserver_dba, senior, or super

**Target endpoint**

Managed node (where the event server is installed)
GUI data entry fields

The following shows the Configure TEC for Microsoft SQL window, followed by a description of each field.

Enter the name for the Microsoft SQL rule base

Specifies the name of the new or existing rule base to configure. If you specify an existing rule base, the task verifies that both its class and rule set files are defined correctly. It is preferable not to modify the Default rule base, so do not type Default. The Tivoli wlsrb command provides a list of all rule bases. Type only the name because a full path is ignored.

A rule base consists of a set of expressions (rule set files) used by the event server to determine if an event meets the rule conditions. The rules can also define a set of actions taken when an event meets the specified rule conditions.

Clone currently ACTIVE rule base

Clones a new rule base from the currently active rule base.

Create new rule base without cloning

Creates a new rule base that is not based on an existing rule base.

Specify a rule base to clone

Copies an existing rule base as the basis for the rule base you are setting up for this configuration option. Specify the name of the rule base in the Rule base to clone field.

Rule base to clone:

Specifies the name of the existing rule base to copy. If you have not previously created a new rule base, you might want to clone an existing rule base, such as the Default rule base. A task error occurs if you select the Clone currently ACTIVE rule base button and type a name in this field.

Restart Event Server

Restarts the event server immediately following configuration. When you create a new rule base or modify an existing one, you must restart the event server for rule base configurations to take effect.

If the event server is not running, this task starts it. If the event server is running, this task stops and restarts it. If you do not use the task to restart the event server, you must load the rule base and restart the event server manually. See the IBM Tivoli Enterprise Console User’s Guide for instructions.
CLI syntax

Using the \texttt{wruntask} command:

\texttt{–t ConfigureTECMSSQL}
\texttt{–l ITMMicrosoftSQLServerTasks}
\texttt{–h \langle TEC Server Object \rangle}
\texttt{–a \langle newrulebasename \rangle}
\texttt{[–a \"\langle ACTIVE \mid NOCLONE \mid EXISTING\rangle \langle rulebasetocopy \rangle \mid \_unchanged\_\"]}
\texttt{[–a \langle Y \mid N \rangle]}

where:

\texttt{\langle newrulebasename \rangle}
\quad \langle newrulebasename \rangle \text{is the name to give to the new rule base.}

GUI Name: \textbf{Rule Base Name}

\textbf{The second parameter defines which rule base to clone.}

\quad \texttt{ACTIVE} uses the active rule base as the basis for the rule base you are setting up. \texttt{NOCLONE} does not use an existing rule base. \texttt{EXISTING} uses an existing rule base as the basis for the rule base you are setting up. If you specify \texttt{EXISTING}, define the name of the existing rule base (\texttt{\langle rulebasetocopy \rangle}). If you specify \texttt{ACTIVE}, use \texttt{\_unchanged\_} to use the currently active rule base.

GUI Name: \textbf{Copy Rule Base and Rule Base to Copy}

\textbf{The third parameter specifies event server restart.}

\quad \texttt{Y} restarts the event server. \texttt{N} does not restart the event server.

GUI Name: \textbf{Restart Event Server}

CLI example

\texttt{wruntask}
\texttt{–t ConfigureTECMSSQL}
\texttt{–l ITMMicrosoftSQLServerTasks}
\texttt{–h @ManagedNode:vision9}
\texttt{–m 600}
\texttt{–a ITMSSQLBase}
\texttt{–a "ACTIVE \_unchanged\_"}
\texttt{–a Y}

Usage notes

You must restart the event server after defining new classes so that Tivoli Enterprise Console can recognize and display instances of classes. Because each command defines new classes, you must stop and restart the event server for each instance of these defined classes.

See also

See also the following commands in the \textit{Tivoli Management Framework Reference Manual}: \texttt{wruntask}, \texttt{wcrttask}, \texttt{wcrtjob} and \texttt{wgettask}.
CustomizeTECNTEventAdapterMSSQL

Description
Customizes the Tivoli Enterprise Console NT Event Adapter to detect and forward Microsoft SQL Server events from the Windows NT event log to the Tivoli Enterprise Console server. You can define classes for different types of events and write rules for actions you want taken.

This task performs the following actions:
- Checks if the Tivoli Enterprise Console NT Event Adapter is configured to accept customization.
- Makes a backup of the existing Tivoli Enterprise Console NT Event configuration. See “Usage notes” on page 143 for information on how and why this is done.
- Customizes the Adapter to receive different types of events.

Authorization role
mssqlserver_dba, and senior

Target endpoint
MicrosoftSQLServer

GUI data entry fields
The following shows the CustomizeTECNTEventAdapterMSSQL window, followed by a description of each field.

Select any event class to exclude events. By default, this task does not exclude any events. All events in the following event classes are sent to the Tivoli Enterprise Console:

**Microsoft SQL Server Error Events**
Select the check box to exclude Microsoft SQL Server error events.

**Microsoft SQL Server Message Events**
Select the check box to exclude message events. Messages can include warnings that a backup record was not written to the
SYSBACKUPHISTORY table. Microsoft defines which types of events are considered messages or information. This task uses the Microsoft classification.

Microsoft SQL Server Information Events
Select the check box to exclude information events. Information events can include information on backups being performed and the server being stopped and started. Microsoft defines which types of events are considered messages or information. This task uses the Microsoft classification.

Microsoft SQL Server Agent Events
Select the check box to exclude events associated with the SQLServerAgent. SQLServerAgent events include information on Microsoft SQL Server tasks and replication.

Microsoft Distributed Transaction Coordination Events
Select the check box to exclude events associated with the MSDTC service. MSDTC information relates to Microsoft Distributed Transaction Coordination.

**CLI syntax**

Using the `wruntask` command:

```
-t CustomizeTECNTEventAdapterMSSQL
-l ITMMicrosoftSQLServerTasks
-h <ObjectName>
[-a FilterErrors={Y | N}]
[-a FilterMessages={Y | N}]
[-a FilterInformation={Y | N}]
[-a FilterSQLServerAgent={Y | N}]
[-a FilterMSDTC={Y | N}]
```

where:

**FilterErrors**

Y filters out events associated with Microsoft SQL Server errors. N does not filter.

GUI Name: Microsoft SQL Server Error Events

**FilterMessages**

Y filters out events associated with Microsoft SQL Server messages. N does not filter.

GUI Name: Microsoft SQL Server Message Events

**FilterInformation**

Y filters out events associated with Microsoft SQL Server information. N does not filter.

GUI Name: Microsoft SQL Server Information Events

**FilterSQLServerAgent**

Y filters out events associated with the SQLExecutive service. N does not filter.

GUI Name: Microsoft SQL Server Agent Events

**FilterMSDTC**

Y filters out events associated with the MSDTC service. N does not filter.
GUI Name: **Microsoft Distributed Transaction Coordination Events**

**CLI example**

```
wruntask -t CustomizeTECEventAdapterMSSQL
-1 ITMMicrosoftSQLServerTasks
-h @MicrosoftSQLServer:hbergin@abc
-a FilterInformation=Y
-a FilterSQLServerAgent=Y
-a FilterMSDTC=Y
```

**Usage notes**

This task searches the registry of the Tivoli endpoint for the directory that contains the Event Adapter service executable. The task confirms the existence of the required Adapter Files before attempting to modify the Tivoli Enterprise Console Configuration file by appending to it the MSSQL Configuration File. This customized configuration file is used as the source for the generation of a Class Definition Statement (CDS) file.

The CDS file is used by the adapter to map incoming raw events to a particular class and to define the events attributes before forwarding the event to the Event Server. The event filtering that you specify alters the source of the Tivoli Enterprise Console Adapter Configuration file. The configuration file is read by the Adapter when it is started and specifies the event filters for events that are not sent to the event server. The Tivoli Enterprise Console Adapter is restarted to institute the newly created customization.

This task backs up the `tecad_nt.conf` and `tecad_nt.fmt` files before customizing them.

The naming convention for the backup files is:

```
<filename>.before.mssql.<timestamp>
```

The `<timestamp>` convention is `year.month.day.hour.minutes`. For example, the following file names show that backups occurred at 10:16 a.m. on February 9, 2003, before you customized the Adapter.

```
tecad_nt.conf.before.mssql.2003.02.09.10.16

ntecad_nt.fmt.before.mssql.2003.02.09.10.16
```

Only one set of backup files is maintained at a time.

**See also**

See also the following commands in the *Tivoli Management Framework Reference Manual*: `wruntask`, `wcrtask`, `wcrtjob` and `wgettask`. 
DiscoverMSSQLServers

Description

Finds a Microsoft SQL Server installation on an endpoint and registers it on the Tivoli desktop. Run this task serially when you run it against multiple nodes. Running this task in parallel can cause contention issues on the Tivoli management region server.

Note: Running Discovery in large-scale environments might not return the task screen for some time. You can verify the task is working in the Tivoli management environment where database icons display as they are discovered.

See the IBM Tivoli Monitoring for Databases: Microsoft SQL Server User’s Guide for additional information on setting up IBM Tivoli Monitoring for Databases: Microsoft SQL Server.

Authorization role

mssqlserver_dba

Target endpoint

MicrosoftSQLServer

GUI data entry fields

The following shows the DiscoverMSSQLServers window, followed by a description of each field.

Select Endpoints

Designates the endpoint on which to discover Microsoft SQL Servers.
Optionally, you can click **Endpoints** to display the following window:

![Select Endpoints Window](image)

**Choose one or more**
Designates the endpoint on which to discover Microsoft SQL Servers.

**Select a Proxy Managed Node (optional)**
Specifies whether to select a managed node to contain the MicrosoftSQLServer object for the Microsoft SQL Servers being discovered. If no proxy managed node is specified, the task uses the managed node (gateway) currently defined for the endpoints. Click **Proxy Managed Nodes** to display the following window:

![Select a Proxy Managed Node (optional) Window](image)

**Choose one**
Designates the proxy managed node on which to contain the MicrosoftSQLServer object for the Microsoft SQL Servers being discovered.

**CLI syntax**
Using the `wruntask` command:

```
-t DiscoverMSSQLServers
-l ITMMicrosoftSQLServerTasks
-h <managed.node>
-a {<end_point_name> | <end_point_name> end_point_name>
[-a "<proxy Managed_node>"]
```

where:

`<end_point_name>`

`<end_point_name>` is the name of the endpoint in which to register the database servers. Use spaces to separate multiple endpoint names.

GUI Name: **Endpoints**
<proxy_managed_node>

<proxy_managed_node> is the name of the managed node to contain the MicrosoftSQLServer object for the Microsoft SQL Servers being discovered. If no proxy managed node is specified, the task uses the managed node (gateway) currently defined for the endpoints.

**CLI example**

wruntask -t DiscoverMSSQLServers
-l ITM\MicrosoftSQLServerTasks
-h @ManagedNode:vision9
-a v1172k-1cf

**Usage notes**

Microsoft SQL Server Versions 7.0 and 2000 are supported.

See also

See also the following commands in the *Tivoli Management Framework Reference Manual*: wruntask, wcrttask, wcrtjob, and wgettask.
DumpDatabase

Description
Performs a full database backup of one or more databases. The DumpDatabase task performs the following actions:

- Provides the option for checking the database integrity before backing up to ensure the backup is viable.
- Enables you to designate an existing backup device, a temporary backup device, or the previous backup device when possible.
- Enables you to specify a list of backup options, for example, whether to append to a previous backup or to overwrite it.

Authorization role
mssqlserver_dba

Microsoft SQL Server Permissions: System Administrator, db_owner, or db_backupoperator

Target endpoint
MicrosoftSQLServer

GUI data entry fields
The following shows the DumpDatabase window, followed by a description of each field.

Database name(s)
Specify one or more databases on which you want to run this task. Separate multiple databases with commas. Use an asterisk (*) to run this task on all databases.
Validate database integrity

Runs the DBCC CHECKDB command to determine if the database is corrupted. If the database fails the integrity checks, the backup is not performed.

Specify backup device (logical name or temporary device)

Specifies the logical name for an existing backup device or a temporary device. When you specify a temporary device, the system creates one if it does not exist.

For striped disk backups, type a comma-separated list of either logical names or temporary devices or a mixture of both. You can also make striped disk backups to a mix of device types (disk, tape, floppy, or pipe).

Dump devices used in a stripe set must always be used in a striped set (unless you reinitialize it at some point). When a device is defined as part of a striped set, you cannot use it for a single device dump, unless you overwrite it. You cannot use a device that contains non-striped dumps in a striped set, unless you overwrite it. Do not dump to a diskette. See the "Usage Notes" for additional information.

The following example shows how logical and temporary device names are typed:

Logical Device:<logical_name>

Temporary Device:

<DISK|TAPE|FLOPPY|DISKETTE=<physical path and file name>

Backup device options

Type a comma-separated list of options. For all options that require a parameter, specify a Microsoft SQL Server variable or literal text. For more information on these options, see the description of the BACKUP command in Microsoft SQL Server Books Online.

Choose from the following options:

BLOCKSIZE=number
Sets the block size to use when writing the backup. When writing to DISK or TAPE, the size is set automatically unless overridden. When writing to PIPE, the default size is 65,536 unless overridden.

DESCRIPTION=text
Specifies a description for the backup set. The maximum length is 255 characters. The default is blank.

DIFFERENTIAL
Specifies a differential backup rather than a full backup. The default is full backup.

EXPIREDATE=date or RETAINDAYS=number
Specifies when the backup file can be overwritten. Only one of these options can be used. EXPIREDATE specifies the expiration date. RETAINDAYS specifies how many days to retain the backup.

FORMAT or NOFORMAT
Writes the media header on all volumes used for this backup and rewrites the backup device. Any existing media header is overwritten. All media contents are invalidated. Existing passwords are ignored. Use only on complete backup sets. Using it on a single tape belonging to an existing striped backup set renders the entire
backup set useless. Using FORMAT implicitly sets INIT and SKIP. You can set the INIT and SKIP options separately.

INIT or NOINIT

INIT specifies the backup overwrites any previous backup on the backup device. The media header is preserved. NOINIT specifies the backup is appended to the previous backup, if any. Default: NOINIT. The backup does not overwrite if it encounters any of the following conditions:

- The media contains unexpired backup sets. You can override this condition by specifying SKIP.
- An explicitly given backup set name does not match the name on the backup media. You can override this condition by specifying SKIP.
- The backup media is password-protected.
- The backup media is encrypted.

MEDIADESCRIPTION=text
Specifies a description for the media set. The maximum length is 128 characters. The default is blank.

MEDIANAME=text
Specifies the media name for the backup. When specified, if the existing MEDIANAME on the backup media does not match, the backup fails.

NAME=text
Specifies the name of the backup set. The maximum length is 128 characters. The default is blank.

RESTART
Specifies to restart a backup operation that was interrupted. Microsoft SQL Server attempts to restart the backup where it was interrupted, possibly saving time.

SKIP or NOSKIP
Specify SKIP to disable checking for name match and expiration date of all backup sets on the media. You can use SKIP with INIT (see description). Specify NOSKIP with FORMAT to override its default behavior (see the description for FORMAT). The default is NOSKIP except where otherwise noted.

STATS=percent
Specifies how often to write a progress message, and is expressed in percent complete. The default is 10.

UNLOAD or NOUNLOAD
Use only for tape devices. UNLOAD specifies that the media is rewound and unloaded (ejected) when the backup is complete. The default is NOUNLOAD.

**CLI syntax**

Using the wrundtask command:

```
-t DumpDatabase
-l ITMMicrosoftSQLServerTasks
-h <managed.node>
[-a database=<database_name>]
```
[–a ValidateDB=\{Y | N\}]
–a DumpTo=\{devicename1\}, \{devicename2\}, \{devicename3\}, \{devicename\...\}]
[–a WithOptions=\"<options>\"]

where:

**database**

\<database_name\> specifies the name of the database on which you want to run this task. Separate multiple databases with commas. You can use an asterisk (*) to run this task on all databases.

GUI Name: **Database name(s)**

**ValidateDB**

Y runs DBCC CHECKDB to determine if the database is corrupted. N does not validate the integrity of the database. This is the default.

GUI Name: **Validate database integrity**

**DumpTo**

Specifies the logical name for an existing backup device or a temporary device.

GUI Name: **Specify backup device (logical name or temporary device)**

**WithOptions**

Specifies the backup options to use. Options must be a comma-separated list and enclosed in double quotes. The options are specified the same way as in the **Backup device options** GUI field.

GUI Name: **Backup device options**

**CLI example**

wruntask -t DumpDatabase
-1 ITMMicrosoftSQLServerTasks
-h @MicrosoftSQLServer\:hbergin@abc
-a database=EastCoast
-a ValidateDB=N
-a DumpTo=mylogical2

**Usage notes**

You must have **DBO** (database owner) or the **sa** (Microsoft SQL Server System Administrator) authorization to run this task.

Before backing up a database, back up a **TRANSACTION LOG** using the **TRUNCATE_ONLY** option to clear out the log and decrease the total time required to back up the database. However, if the database backup is not done immediately or fails, you can lose all the logged transactions. This task does not truncate the log before backing up the database.

For backups to remote machines, Microsoft SQL Server service must have rights on the remote machine and cannot be running under the local system account.

Do not make master backups to striped devices because that causes the task to fail and return a warning.

Do not dump backups to diskette. You should back up to a disk device and copy (or use the **file backup** command) to a diskette instead. There is no diskette option
available when creating a new device. If you specify an existing diskette device, you must ensure that the operation is successful.

CONSOLE.EXE must be running in an MS-DOS window to make a backup to a floppy disk. You must also manually load and remove diskettes. The dump cannot continue if CONSOLE.EXE is not running. A series of prompts guides you through the backup. If it is running, the dump can continue only if you respond to the prompts. See the Windows NT Help or the Microsoft SQL Server Books Online system for additional information on the CONSOLE command.

The system creates the DISKDUMP device when you install Microsoft SQL Server. Backing up to DISKDUMP sends the data to the NULL device, which dumps data without saving it. You can use the DISKDUMP device name as a bit bucket in conjunction with database or TRANSACTION LOG backups. You cannot retrieve dumps sent to the NULL (DISKDUMP) device.

Use the sp_addumpdevice stored procedure to add a backup device to the master..dbo.sysdevices table. It can then be referred to logically in this task. See the Microsoft Transact-SQL Reference Guide for details.

See also

See also the following commands in the Tivoli Management Framework Reference Manual: wruntask, wcrtjob, and wgettask.
DumpTransactionLog

Description

Performs a backup or truncation of a database’s TRANSACTION LOG.

In a production environment, many DBAs schedule TRANSACTION LOG dumps to manage normal server activity so that the TRANSACTION LOG never fills up. Full database backups do not truncate the log. During exceptionally high database usage, the TRANSACTION LOG can become full. To alleviate this problem, configure the task Scheduler to run this task to perform periodic dumps.

This task does the following:

• Provides the option for checking the database integrity before backing up to ensure the backup is viable.
• Enables you to designate an existing backup device, a temporary backup device, or the previous backup device when possible.
• Enables you to specify whether to append to or overwrite the previous backup on the chosen device. (Appending is the default.)
• Enables you to choose to eject the tape after backing up or to ignore header information when backing up to tape devices.
• Checks the readability of the backup (by reading header information) after the backup is complete.
• Reports some statistics on the backup procedure and header information on the backup device.

Authorization role

mssqlserver_dba

Microsoft SQL Server permissions: System Administrator or db_owner or db_backupoperator

Target endpoint

MicrosoftSQLServer
GUI data entry fields

The following shows the **DumpTransactionLog** window, followed by a description of each field.

![DumpTransactionLog Window](image)

**Database name(s)**
Specify the name of the database on which you want to run this task. Separate multiple databases with commas. You can use an asterisk (*) to run this task on all databases.

**Specify backup device (logical name or temporary device)**
Specify the logical name for an existing backup device or a temporary device. When you specify a temporary device, the system creates it, if it does not exist.

For striped disk backups, type a comma-separated list of either logical names or temporary devices or a mixture of both. You can also make striped disk backups to a mix of device types (disk, tape, floppy, or pipe).

You cannot use a device that is defined as part of a striped set for a single device dump, unless you overwrite it. You cannot use a device that contains non-striped dumps in a striped set, unless you overwrite it. Tivoli does not recommend dumping to diskette. See the "Usage Notes" for additional information.

The following example shows how logical and temporary device names are entered:

**Logical Device**: `<logical_name>`

**Temporary Device**: `<DISK|TAPE|FLOPPY|DISKETTE>=<physical path and file name>`
Dump options
Specifies how to backup and or truncate a TRANSACTION LOG. Your choices include the following:

- **Backup and truncate** backs up the TRANSACTION LOG and truncates it. *Truncating* the log clears it of all non-active transactions. You can use this option only when the TRANSACTION LOG is a separate device fragment than the data.

- **Backup** backs up the TRANSACTION LOG, but does not truncate it. The SYSBACKUPHISTORY table does not record this operation.

- **Truncate** truncates the TRANSACTION LOG, but does not back it up. This operation is logged.

- **Truncate (use this option when log is full)** truncates the TRANSACTION LOG, but does not back it up. It is the only choice when the log is full. This operation is not logged.

Backup options
Type a comma-separated list of options. For all options that require a parameter, specify a Microsoft SQL Server variable or literal text. For more information on these options, see the description of the BACKUP command in Microsoft SQL Server Books Online.

Choose from the following options:

**BLOCKSIZE=number**
Sets the block size to use when writing the backup. When writing to DISK or TAPE, the size is set automatically unless overridden. When writing to PIPE, the default size is 65,536 unless overridden.

**DESCRIPTION=text**
Specifies a description for the backup set. The maximum length is 255 characters. The default is blank.

**DIFFERENTIAL**
Specifies a differential backup rather than a full backup. The default is full backup.

**EXPIREDATE=** or **RETAINDAYS=number**
Specifies when the backup file can be overwritten. Only one of these options can be used. EXPIREDATE specifies the expiration date. RETAINDAYS specifies how many days to retain the backup.

**FORMAT** or **NOFORMAT**
Writes the media header on all volumes used for this backup and rewrites the backup device. Any existing media header is overwritten. All media contents are invalidated. Existing passwords are ignored. Use only on complete backup sets. Using FORMAT implicitly sets INIT and SKIP. You can specify the INIT and SKIP options separately.

**INIT** or **NOINIT**
INIT specifies the backup overwrites any previous backup on the backup device. The media header is preserved. NOINIT specifies the backup is appended to the previous backup, if any. The default is NOINIT. The backup does not overwrite if it encounters any of the following conditions:

- The media contains unexpired backup sets. You can override this condition by specifying SKIP.
• An explicitly given backup set name does not match the name on the backup media. You can override this condition by specifying SKIP.
• The backup media is password-protected.
• The backup media is encrypted.

**MEDIADESCRIPTION=text**
Specifies a description for the media set. The maximum length is 128 characters. The default is blank.

**MEDIANAME=text**
Specifies the media name for the backup. When specified, if the existing MEDIANAME on the backup media does not match, the backup fails.

**NAME=text**
Specifies the name of the backup set. The maximum length is 128 characters. The default is blank.

**NO_TRUNCATE**
Backs up the log without truncating it. Otherwise, the log is truncated as per the setting in the Dump Options. It also backs up the log if the database becomes damaged, if the database has been marked suspect, or if the database has not been recovered.

**RESTART**
Specifies to restart a backup operation that was interrupted. Microsoft SQL Server attempts to restart the backup where it was interrupted, possibly saving time.

**SKIP** or **NOSKIP**
Specify **SKIP** to disable the checking for name match and expiration date of all backup sets on the media. You can use **SKIP** with **INIT** (see description). Specify **NOSKIP** with **FORMAT** to override its default behavior (see the description for **FORMAT**). Default: **NOSKIP** except where otherwise noted.

**STATS=percent**
Specifies how often to write a progress message, and is expressed in percent complete. Default: 10.

**UNLOAD** or **NOUNLOAD**
Used only for tape devices. **UNLOAD** specifies that the media is rewound and unloaded (ejected) when the backup is complete. Default: **NOUNLOAD**.

**CLI syntax**
Using the `wruntask` command:

```
-t DumpTransactionLog
-l ITMMicrosoftSQLServerTasks
-h <managed.node>
[-a database=<database_name>]
[-a DumpOption=[Default | NoTruncate | TruncateOnly | NoLog]]
-a DumpTo=<devicename1>, [<devicename2>, <devicename3>, <devicename...>]
[-a WithOptions="<options>"]
```

where:
database
<database_name> specifies the database on which you want this task to run.

DumpOption
Choose one of the following:

- **Default**: backs up the TRANSACTION LOG and truncates it. This is the default.
  GUI Name: **Backup and truncate**
- **NoTruncate**: backs up the TRANSACTION LOG, but does not truncate it.
  GUI Name: **Backup**
- **TruncateOnly**: truncates the TRANSACTION LOG, but does not back it up.
  GUI Name: **Truncate**
- **NoLog**: truncates the TRANSACTION LOG, but does not back it up. It is the only choice you have when the TRANSACTION LOG is full.
  GUI Name: **Truncate (use this option when log is full)**

DumpTo
Specifies the task the logical name for an existing backup device or a temporary device.

GUI Name: **Specify backup device (logical name or temporary device)**

WithOptions
Specifies the backup options to use. Options must be a comma-separated list and enclosed in double quotes. The options are specified the same way as in the **Backup options** GUI field.

GUI Name: **Backup options**

CLI example
wruntask -t DumpTransactionLog
-l ITM\MicrosoftSQLServerTasks
-h \@MicrosoftSQLServer:hbergin@abc
-a database=EastCoast
-a DumpOption=Default
-a DumpTo=mylogical2

Usage notes
Use the **NoTruncate** option when media fails. The **NoTruncate** option enables you to dump a TRANSACTION LOG even when the database is inaccessible because it uses a pointer to the TRANSACTION LOG in the master database. The **NoTruncate** option provides media recovery by the minute when the master database and the log portion of the user database reside on undamaged database devices and when only the data portion of the user database is damaged. This backup is not recorded in the SYSBACKUPHISTORY table. The only way to tell if it was successful is to read the header information on the backup device (LOAD HEADERONLY). You can backup the master database log separately if you use the **NoTruncate** option.

The **DUMP DATABASE** command backs up both the data and the log. The **DUMP TRANSACTION** command backs up only the TRANSACTION LOG and not the data. To back up only the TRANSACTION LOG log, it must exist in a different storage file than the data. When both the data and the TRANSACTION LOG are stored in the same file (or device), this is known as sharing a device.
Dumping a database always means the same as backing up the database. Dumping a TRANSACTION LOG means that you are backing it up, truncating it, or both.

**Note:** The restrictions in these notes are Microsoft product restrictions. It is physically impossible, for example, to dump a TRANSACTION LOG when the database was never dumped. You should back up the TRANSACTION LOG when it is 50 to 75 percent full.

See the [DumpDatabase](#) task for additional information.

Use the `sp_addumpdevice` stored procedure to add a backup device to the `master.dbo.sysdevices` table. It can then be referred to logically in this task. See the Microsoft *Transact-SQL Reference Guide* for details.

**See also**

See also the following commands in the *Tivoli Management Framework Reference Manual*: `wruntask`, `wcrtjob`, and `wgettask`. 
ExecuteSQL

Description
Runs a user-defined TSQL (Transact-SQL) statement on a database.

Note: Some actions resulting from this task can have far-reaching repercussions on data and database availability. The performance of your environment might be affected if you specify a Transact-SQL statement that requires processing a large number of objects.

All applications that communicate with Microsoft SQL Server use Transact-SQL. The applications send Transact-SQL statements to the server, regardless of the application’s user interface. Use this task to execute a customized Transact-SQL statement specific to your operational needs.

Authorization role
mssqlserver_dba

Microsoft SQL Server Permissions: depends on SQL statement

Target endpoint
MicrosoftSQLServer

GUI data entry fields
The following shows the ExecuteSQL window, followed by a description of each field.

TSQL Command
Specifies the Transact-SQL string you want the task to execute. Type the exact string in the space provided. You do not need to enclose the string in quotes.

The use command is not supported by this task. Instead of the use command, prepend the database for any tablenames in your SQL statement. For example, the following string is not valid:

use Northwind; select * from Employees
Instead, use a valid string such as the following:

```
select * from Northwind.dbo.Employees
```

You have the option to use a customized Transact-SQL command, a stored procedure, or a dbcc command. Although it is not required, you can precede the name of the stored procedure with `execute` or `exec` when calling a stored procedure, for example:

```
sp_catalog
```

**CLI syntax**

Using the `wruntask` command:

```
–t ExecuteSQL
–l ITMMicrosoftSQLServerTasks
–h <target_name>
–a TSQL="<sql command>"
```

where:

- **TSQL "<sql command>"** specifies the Transact-SQL string you want the task to execute. Type the exact string and enclose the string in quotes.

  The `use` command is not supported by this task. Instead of the `use` command, prepend the database for any tablenames in your SQL statement. For example, the following string is *not* valid:

  ```
  use Northwind; select * from Employees
  ```

  Instead, use a valid string such as the following:

  ```
  select * from Northwind.dbo.Employees
  ```

  You have the option to use a customized Transact-SQL command, a stored procedure, or a dbcc command. Although it is not required, you can precede the name of the stored procedure with `execute` or `exec` when calling a stored procedure, for example:

  ```
  "exec sp_catalog"
  ```

**GUI Name:** **TSQL Command**

**CLI example**

```
wruntask -t ExecuteSQL
–l ITMMicrosoftSQLServerTasks
–h @MicrosoftSQLServer:VISION117@abc
–a TSQL="select count(*) from pubs.dbo.authors"
```

**Usage notes**

Transact-SQL statements must be valid on the targeted version of Microsoft SQL Server. Refer to the Microsoft SQL Server Books Online for more information about Transact-SQL statements in combination with Microsoft SQL Server versions.

Resource-intensive queries can hamper the availability of the server to other users.

SQL commands issued with the **ExecuteSQL** task should only produce one result set. If the SQL command produces multiple result sets, only the first set is returned.
Exercise caution when using this task with a DDL (Data Definition Language) or DML (Data Manipulation Language) command, such as DROP TABLE, DROP DATABASE, DELETE FROM TABLE, or DELETE FROM DATABASE. Dropping a table can have serious consequences for your database. When a table is dropped, the table’s data is not recoverable. Do not drop a table unless you fully understand the implications of this command. Immediately before and after dropping a table, back up the database completely.

See also
See also the following commands in the Tivoli Management Framework Reference Manual: wruntask, wcrttask, wcrtjob, and wgettask.
MSSQLAuthentication

Description
Specifies a user name and password for login access to a Microsoft SQL Server and its databases. The task also specifies whether authentication of the user name is performed by Microsoft Windows or Microsoft SQL Server.

Note: This task changes the authentication for Microsoft SQL Server objects on the managed node on which the proxy object exists. Even though a managed node appears in the target endpoint list, that does not mean that the managed node controls the Microsoft SQL server objects. Only the managed node on which the proxy object exists defines which Microsoft SQL server objects are updated with this task.

If you used the DiscoverMSSQLServers task and selected a proxy managed node, then the selected proxy managed node controls the Microsoft SQL server objects that are updated with this task. Review your environment carefully before executing this task to change authentication.

Authorization role
mssqlserver_dba

Target endpoint
MicrosoftSQLServer

GUI data entry fields
The following shows the MSSQLAuthentication window, followed by a description of each field.

User name
Specifies a user name with access to log in to the Microsoft SQL Server and its databases. This field is a mandatory field. The following text shows the correct format for User name:

`[<domainname>]\<username>`

where:

`<domainname>` is the name of a Windows Network Domain and is only specified if the user name is a member of a Windows Network Domain Account.
<username> is a unique name identifying a user account to Windows.

**Password**
Specifies the corresponding password for **Username**. The password must match the value stored in the user’s account. If there is no password associated with the user name, you can leave this field blank.

**SQL Server account?**
Specifies whether Microsoft SQL Server, rather than the Windows operating system, authentication is applied to the user name. Select the check box to use Microsoft SQL Server to validate the user name. Clear the check box to use the local Windows operating system or Windows Network Domain Server to validate the user name.

**CLI syntax**
Using the `wruntask` command:

```bash
-t MSSQLAuthentication
-l ITMMicrosoftSQLServerTasks
-h <target_name>
-a Username="<username>"
[-a Password="<password>"]
[-a SQLServerAccount=[Y | N]]
```

where:

- **target_name**
  `<target_name>` is the target instance name.

- **UserName**
  `<username>` specifies the login name that has access to the database server and its databases. The login name format is:
  ```
  [<domainname>]\<username>
  ```

  where `<domainname>` is the name of a Windows Network Domain and `<username>` is a unique name identifying a user account to Windows. The value, `<domainname\>`, is specified only if the user name is a member of a Windows Network Domain Account.

  **GUI Name:** **User name**

- **Password**
  `<password>` specifies the password for the login name specified by **UserName**. The password must match the value stored in the user’s account.

  **GUI Name:** **Password**

- **SQLServerAccount**
  Specifies whether Microsoft SQL Server, rather than the Windows operating system, authentication will be applied to the user name. Y specifies Microsoft SQL Server validates the user name. N specifies either the local Windows operating system or Windows Network Domain Server validates the user name.

  **GUI Name:** **SQL Server account?**
CLI example

`wruntask -t MSSQLAuthentication
-l ITMMicrosoftSQLServerTasks
-h @ManagedNode:acclaim
-a UserName="sa"
-a Password="Xt3eWxq"
-a SQLServerAccount=Y`

Usage notes

The `MSSQLAuthentication` task changes the user name, password, and server account type for all of the Microsoft SQL Server objects in the Tivoli management region.

See also

See also the following commands in the `Tivoli Management Framework Reference Manual`: `wruntask`, `wcrttask`, `wcrtjob`, and `wgettask`. 
MSSQLTBSMDiscovery

Description
Checks for recently added or removed active instances and notifies Tivoli Business Systems Manager of the changes. Tivoli Business Systems Manager makes the appropriate changes to its display.

MSSQLTBSMDiscovery sends a DISCOVER event to Tivoli Business Systems Manager for each Microsoft SQL Server that IBM Tivoli Monitoring for Databases: Microsoft SQL Server discovers (or is currently managing).

For resources that were removed since the last time this task was run, this task sends a GONE event to Tivoli Business Systems Manager. If the task completes successfully, the Tivoli Business Systems Manager server receives a list of resources for which DISCOVER and GONE events were sent.

Authorization role
mssqlserver_dba

Target endpoint
MicrosoftSQLServer

GUI data entry fields
None

CLI syntax
Using the wruntask command:

```
-t MSSQLTBSMDiscovery
-l ITMMicrosoftSQLServerTasks
-h <$TMR_server_name>
```

where:

$TMR_server_name

\(<$TMR_server_name>\) is the name of the machine where the Tivoli management region server is installed.

CLI example

```
wruntask -t MSSQLTBSMDiscovery
-l ITMMicrosoftSQLServerTasks
-h vision-20
```

Usage notes

This task uses the Tivoli Business Systems Manager generic interface installed on a Tivoli Enterprise Console server to send events to Tivoli Business Systems Manager. You must run the task on a managed node that has a Tivoli Enterprise Console server and you must configure Tivoli Business Systems Manager to interface with that Tivoli Enterprise Console server.

Set the timeout value to at least 600 seconds to avoid premature task failure.
See also

See also the following commands in the *Tivoli Management Framework Reference Manual*: wruntask, wcrttask, wcrtjob, and wgettask.
NTEventLogging

Description
Specifies the non-fatal errors and user defined errors to log in the Windows event log. This error logging is performed in addition to the logging performed by the database server Microsoft SQL Server error log. The specified Microsoft SQL Server errors are logged to the Application log section of the event log.

This task operates only on non-fatal and user-defined errors. All fatal errors are logged to the Windows event log and the Microsoft SQL Server error log.

Authorization role
mssqlserver_dba

Target endpoint
MicrosoftSQLServer

GUI data entry fields
The following shows the NTEventLogging window, followed by a description of each field.

Change current settings
Choose one of the following:

Add adds the error numbers (that you designate with Non-fatal error numbers) to the errors already included in the Windows event log. This changes the logging status for the errors entered and does not affect any other existing error configurations. If you designate a number that is already included, the system does not include it twice or cause an error.

Remove changes the logging status for the errors (that you designate with Non-fatal error numbers) by removing them. It does not affect any other existing error configurations. If an error is already configured not to be logged to the Windows event log, this does not cause an error.
**Overwrite** overwrites any existing error numbers with the errors specified in the **Non-fatal error numbers** field in the Windows event log. Any errors not specified in **Non-fatal error numbers** field are sent only to the SQL ERROR LOG.

If you leave the **Non-fatal error numbers** field blank and select **Overwrite**, the task sends non-fatal or user-defined errors to the Windows event log. The task overwrites any previously selected error numbers.

**Non-fatal error numbers**
Identifies the errors to log in the Windows event log. Type the Microsoft SQL Server error numbers, separated by commas. If you this field blank, no errors are written to the Windows event log.

**CLI syntax**

Using the `wruntask` command:

```
-t NTEventLogging
-l ITMMicrosoftSQLServerTasks
-h <target_name>
-a EditMode=[Add | Remove | Overwrite]
-a ErrorNumbers=[<number> | <number>,...,<number>]
```

where:

**EditMode**
You can specify one of the following three mutually exclusive arguments.

**Add** adds the error numbers (that you designate with **ErrorNumbers**) to the current settings.

GUI Name: **Add**

**Remove** removes the errors (that you designate with **ErrorNumbers**), but it does not affect any other existing error configurations.

GUI Name: **Remove**

**Overwrite** logs only the error numbers (that you designate with **ErrorNumbers**) to the Windows event log.

GUI Name: **Overwrite**

**ErrorNumbers**

**<number>** identifies the Microsoft SQL Server errors numbers to include in the Windows event log. Separate error numbers with commas. GUI Name: **Non-fatal error numbers**

**CLI example**

```
wruntask -t NTEventLogging
-l ITMMicrosoftSQLServerTasks
-h @MicrosoftSQLServer:hbergin@abc
-a EditMode=Add
-a ErrorNumbers=50125,56752
```

**Usage notes**
This task uses the Microsoft SQL Server `sp_altermessage` stored procedure. See the Microsoft SQL Server Books Online for more information about `sp_altermessage`. 
Non-fatal error messages can be added programatically or manually by using the `sp_addmessage` stored procedure. Refer to the Microsoft *Transact-SQL Reference Guide* for details.

Fatal errors (severity 19 and above) that are not marked for logging in the `sysmessages` table are automatically logged to the Windows NT event log. You cannot exclude fatal errors from the log.

**See also**

See also the following commands in the *Tivoli Management Framework Reference Manual*: `wruntask`, `wcrttask`, `wcrjob`, and `wgettask`. 
**RebuildFragmentedIndexes**

**Description**

Rebuilds one or more indexes for a table in the specified database. You can rebuild clustered and non-clustered indexes.

*Note:* Rebuilding indexes in large tables can be time-consuming and I/O intensive. Consider the size of a table before you run this task. You can increase the job duration time for this task to ensure the job displays completion information. See the section on working with tasks and jobs in the IBM Tivoli Monitoring for Databases: Microsoft SQL Server User’s Guide for instructions on increasing the job duration time.

An *index* is a database object that enables you easy access to data in the rows of a table without scanning the entire table. Microsoft SQL Server supports clustered and non-clustered indexes.

A *clustered index* is an index that physically sorts the records in order. Clustered indexes have an index identifier of $\text{INDID} = 1$. A *non-clustered index* is an index that does not physically sort the records in order. Non-clustered indexes have an index identifier of $\text{INDID} > 1$.

Rebuilding is the process of reorganizing the index to combine records. You can rebuild a highly fragmented index. Rebuilding an index can improve disk performance by reducing the number of page reads required to retrieve requested data.

**Authorization role**

mssqlserver_dba

**Microsoft SQL Server Permissions:**

- Microsoft SQL Server 7.0: System Administrator, or db_owner
- Microsoft SQL Server 2000: System Administrator server role, db_owner, db_ddladmin, or owner of the table

**Target endpoint**

MicrosoftSQLServer
GUI data entry fields

The following shows the RebuildFragmentedIndexes window, followed by a description of each field.

Table name
Specifies the table or tables in which to rebuild the indexes.

Index name
Rebuilds the indexes with a particular name, or all indexes. An asterisk (*) updates all indexes and is the default.

The following chart compares how entries in the Table name and Index name fields determine which indexes and tables are updated:

<table>
<thead>
<tr>
<th>Table name/Table exists</th>
<th>Index name/Index exists</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entered/Exists</td>
<td>Entered/Exists</td>
<td>Updates only the specified index on the specified table.</td>
</tr>
<tr>
<td>Entered/Exists</td>
<td>Entered/Does not exist</td>
<td>Error. Specified index does not exist.</td>
</tr>
<tr>
<td>Entered/Exists</td>
<td>Blank</td>
<td>Updates all indexes on the specified table.</td>
</tr>
<tr>
<td>Entered/Does not exist</td>
<td>Exists</td>
<td>Error. Specified table does not exist.</td>
</tr>
<tr>
<td>Entered/Does not exist</td>
<td>Entered/Does not exist</td>
<td>Error. Specified table does not exist.</td>
</tr>
<tr>
<td>Entered/Does not exist</td>
<td>Blank</td>
<td>Error. Specified table does not exist.</td>
</tr>
<tr>
<td>Blank</td>
<td>Entered/Exists</td>
<td>Updates all indexes that share the specified name.</td>
</tr>
<tr>
<td>Blank</td>
<td>Entered/Does not exist</td>
<td>Error. Specified index does not exist.</td>
</tr>
<tr>
<td>Blank</td>
<td>Blank</td>
<td>Updates all indexes.</td>
</tr>
</tbody>
</table>

Database name(s)
Specifies the name of the database that contains the index to rebuild.
**CLI syntax**

This task can be run with no arguments. Default values are used for the arguments.

Using the *wruntask* command:

```
-t RebuildFragmentedIndexes
-l ITMMicrosoftSQLServerTasks
-h <target_name>
[-a <tablename>]
[-a <indexname>]
[-a {<databasename>|<databasename>,...,<databasename>}}]
```

**Note:** See Table 6 on page 170 for acceptable combinations of the **Table name** and **Index name** entries.

where:

**TableName**

`<tablename>` specifies the task to rebuild the indexes on the specified table. If you do not specify `<tablename>`, the task rebuilds indexes for all tables.

GUI Name: **Table name**

**IndexName**

`<indexname>` specifies the task to rebuild the specified index. If you do not specify `<indexname>`, the task updates all indexes.

GUI Name: **Index name**

**Databasename**

`<databasename>` specifies the name of the database that contains the index to rebuild.

GUI Name: **Database name(s)**

**CLI example**

```
wruntask -t RebuildFragmentedIndexes
-l ITMMicrosoftSQLServerTasks
-h @MicrosoftSQLServer:VISION118"v118nt-lcf"
-a "Employees"
-a "*
-a "Northwind"
```

**Usage notes**

You can improve table fragmentation when the table has a clustered index that you can rebuild. Table data is directly related to the clustered index since it is the leaf level of the index. This task first checks the clustered index on a table, then rebuilds the clustered index by reordering the table’s data pages.

If a table has no clustered index or the clustered index is not fragmented (called non-clustered index fragmentation), the task checks the non-clustered indexes for fragmentation and automatically rebuilds on an individual basis, if necessary.

When you create an index in the database, the index information used by queries is stored in index pages. The sequential index pages are chained together by pointers from one page to the next. When changes are made to the data that affect the index, the information in the index can become scattered in the database.
Rebuilding an index reorganizes the storage of the index data (and table data in the case of a clustered index) to remove fragmentation.

Rebuilding an index has benefits over recreating an index. Deleting and then recreating an index is expensive because all of the secondary indexes use the clustering key to point to the data rows. Deleting and recreating a clustered index causes all the non-clustered indexes to be deleted and recreated twice. This happens first when you delete the clustered index, and again when you recreate it. Rebuilding the index avoids this expense by recreating the index in a single step.

The RebuildFragmentedIndexes task tells Microsoft SQL Server that you are reorganizing an existing index and avoids the unnecessary work of deleting and recreating non-clustered indexes. This task can also use the sorted order of the data in the existing index. You do not need to re-sort the data if you use the sorted order. Using the sorted order for both clustered and non-clustered indexes can significantly reduce the cost of rebuilding an index.

See also

See also the following commands in the Tivoli Management Framework Reference Manual: wruntask, wcrtjob, and wgettask.
SetRetrieveDatabaseOptions

Description
Changes the configuration of the database and reports on its current configuration.

You can run the task with each database option set to one of three values: True, False, or No Change. Using True and False sets the database option to True or False, respectively. Using No Change leaves the database configuration as it is currently.

To determine the current configuration settings, run the task with No Change set for all options.

Authorization role
mssqlserver_dba

Microsoft SQL Server permissions: System Administrator or db_owner

Target endpoint
MicrosoftSQLServer

GUI data entry fields
The following shows the SetRetrieveDatabaseOptions window, followed by a description of each field.

Database name
Specifies the name of the database for which you want to set or retrieve
options. Separate multiple database names with commas. You can use an asterisk (*) to run this task on all the databases on the selected Microsoft SQL Servers from the task’s Endpoint Target list.

ANSI Null Default
Specifies whether or not the default attribute of new columns in a database are nullable.

No change leaves the database configuration as it is currently.

True sets the default attribute of new columns in a database to be nullable. New columns are created to accept null values on input.

False sets the default attribute of new columns in a database to reject null values on input (non-nullable). This option does not change the attributes of existing columns.

ANSI Nulls
Specifies whether a return of NULL displays if either side of a comparison operation contains a null value.

No change leaves the database configuration as it is currently.

True returns NULL if either side of a comparison operation (=, <>) contains a null value. Otherwise, a true comparison is made, returning TRUE or FALSE.

False does not return NULL if either side of a comparison operation (=, <>) contains a null value.

ANSI Warnings
Specifies ANSI-92 standard behavior for several error conditions.

No change leaves the database configuration as it is currently.

True generates a message if null values appear in aggregate functions. In addition, the statement is rolled back if divide-by-zero or arithmetic overflow errors are encountered.

False does not generate a message for null values in aggregate functions. Null values are returned when divide-by-zero and arithmetic overflow errors are encountered.

Auto create statistics
Specifies whether or not statistical information about the distribution of values in a column is automatically created and stored. The query processor and query optimizer use these statistics.

No change leaves the database configuration as it is currently.

True automatically creates and stores statistical information about the distribution of values in a column.

False does not create and store statistical information about the distribution of values in a column.

Auto update statistics
Specifies whether or not statistics are updated automatically when changes to the table make them obsolete.

No change leaves the database configuration as it is currently.

True automatically updates statistics when changes to the table make them obsolete (based on a statistical sampling algorithm).
**False** does not update statistics. You must update the statistics manually when needed.

**Autoclose**
Specifies whether or not the database closes and frees its resources when no user connections access the database.

**No change** leaves the database configuration as it is currently.

**True** closes the database and cleanly shuts down after the last database user exits and all processes complete.

**False** leaves the database open even if no user connection to the database is active.

**Autoshrink**
Specifies whether or not to use automatic shrinking on database and log files.

**No change** leaves the database configuration as it is currently.

**True** automatically shrinks database and log files. Files are shrunk if they contain more than 25% unused space.

**False** does not automatically shrink any databases or log files.

**Concat null yields null**
Specifies whether or not concatenating a null value to any string causes the string to be set to NULL. **Concatenation** is taking two or more character strings or expressions and placing them adjacent to each other to become one binary string or expression.

**No change** leaves the database configuration as it is currently.

**True** returns a value of NULL if a string is concatenated with a null value.

**False** does not return a NULL value if a string is concatenated with a null value. Concatenating a NULL to a string does not change the string.

**Close on commit**
Specifies whether or not open cursors are closed when a transaction is committed.

**No change** leaves the database configuration as it is currently.

**True** closes any open cursors after a transaction is committed.

**False** does not close any open cursors after a transaction is committed.

**dbo use only**
Specifies that only the database owner (DBO) or logins aliased to the DBO can access a database.

**No change** leaves the database configuration as it is currently.

**True** limits database access to the DBO or logins aliased to the DBO.

**False** does not limit database access to the DBO or logins aliased to the DBO.

**Default to local cursor**
Specifies whether to create the cursor scope as local unless specified as global when created.

**No change** leaves the database configuration as it is currently.
True sets the cursor scope default to local unless global is specified when it is created.

False does not set the cursor scope default to local.

**Merge publish**
Specifies whether or not the database can be used for merge replication publications.

No change leaves the database configuration as it is currently.

True uses the database for merge replication publications.

False restricts the database from merge replication publications.

**Offline**
Specifies whether or not to take the database offline.

No change leaves the database configuration as it is currently.

True takes the database offline.

False does not take the database offline.

**Published**
Specifies whether or not to make the database available for replication publications.

No change leaves the database configuration as it is currently.

True uses the database for replication publications.

False restricts the database from replication publications.

**Quoted identifier**
Specifies whether or not to delimit identifiers with double quotes and literal with single quotes.

No change leaves the database configuration as it is currently.

True specifies that identifiers can be delimited by double quotes and literals must be delimited by single quotes.

False specifies that identifiers cannot be in quotes and must follow all Transact-SQL rules for identifiers.

**Read only**
Specifies whether or not the database is read-only.

No change leaves the database configuration as it is currently.

True makes the database read-only. If it is read-only, you can read from the database, but they cannot make changes to it.

False does not make the database read-only. You can make changes to the database.

**Recursive triggers**
Specifies whether or not to allow triggers to fire recursively.

No change leaves the database configuration as it is currently.

True allows triggers to fire recursively. Note that there is a risk of causing a race condition in table updates that are initiated by triggers.

False does not allow triggers to fire recursively.
Select into/bulk copy
Specifies whether to allow you to use the SELECT INTO statement and BCP (Bulk Copy) command.

No change leaves the database configuration as it is currently.
True allows you to use the SELECT INTO statement and BCP command.
False does not allow you to use the SELECT INTO statement and BCP command.

Single user
Specifies whether to put a database into single-user mode. Setting single-user mode does not disconnect current user connections. It simply prevents any new connections to the database.

No change leaves the database configuration as it is currently.
True sets the database to single-user mode. You can set this option only if there are no other users currently on the database.
False does not set the database to single-user mode.

Subscribed
Specifies whether to make the database available to be subscribed for publication.

No change leaves the database configuration as it is currently.
True makes the database available to be subscribed for publication.
False make the database unavailable to be subscribed for publication.

Torn page detection
Specifies whether the server can detect incomplete I/O operations caused by system or power failures. Detection normally occurs during recovery. Detection of a torn page results in the database being marked suspect, an I/O error, and termination of the connection.

No change leaves the database configuration as it is currently.
True enables Microsoft SQL Server to detect incomplete I/O operations.
False does not enable Microsoft SQL Server to detect incomplete I/O operations.

Truncate log on checkpoint
Specifies whether the transaction log is truncated each time a checkpoint is issued.

No change leaves the database configuration as it is currently.
True truncates the transaction log each time a checkpoint is issued.
False does not truncate the transaction log each time a checkpoint is issued.

CLI syntax
Using the wruntask command:

–t SetRetrieveDatabaseOptions
–l ITMMicrosoftSQLServerTasks
–h <target for task>
–a database={<database_name> | <databasename>,...,<databasename>}
[–a ANSI_Null_Default={NOCHANGE | TRUE | FALSE}]
[-a ANSI_Nulls=[NOCHANGE | TRUE | FALSE]]
[-a ANSI_Warnings=[NOCHANGE | TRUE | FALSE]]
[-a Auto_Create_Statistics=[NOCHANGE | TRUE | FALSE]]
[-a Auto_Update_Statistics=[NOCHANGE | TRUE | FALSE]]
[-a Autoclose=[NOCHANGE | TRUE | FALSE]]
[-a Autoshrink=[NOCHANGE | TRUE | FALSE]]
[-a Concat_Null_Yields_Null=[NOCHANGE | TRUE | FALSE]]
[-a Cursor_Close_On_Commit=[NOCHANGE | TRUE | FALSE]]
[-a Dbo_Use_Only=[NOCHANGE | TRUE | FALSE]]
[-a Default_To_Local_Cursor=[NOCHANGE | TRUE | FALSE]]
[-a merge_publish=[NOCHANGE | TRUE | FALSE]]
[-a Offline=[NOCHANGE | TRUE | FALSE]]
[-a Published=[NOCHANGE | TRUE | FALSE]]
[-a Quoted_Identifier=[NOCHANGE | TRUE | FALSE]]
[-a Read_Only=[NOCHANGE | TRUE | FALSE]]
[-a Recursive_Triggers=[NOCHANGE | TRUE | FALSE]]
[-a Select_Into=[NOCHANGE | TRUE | FALSE]]
[-a single_user=[NOCHANGE | TRUE | FALSE]]
[-a Subscribed=[NOCHANGE | TRUE | FALSE]]
[-a Torn_Page_Detection=[NOCHANGE | TRUE | FALSE]]
[-a Trunc=[NOCHANGE | TRUE | FALSE]]

where:

database
<database_name> specifies the name of the database on which this task is run. If this is left blank, it refers to system database. You can use an asterisk (*) to run this task on all the databases on the selected Microsoft SQL Servers from the task’s Endpoint Target list.

GUI Name: Database name

ANSI_Null_Default
Specifies whether the default attribute of new columns in a database are nullable. NOCHANGE leaves the database configuration as it is currently. TRUE sets the default attribute of new columns in a database to be nullable. New columns are created to accept null values on input. FALSE sets the default attribute of new columns in a database to reject null values on input (non-nullable). This option does not change the attributes of existing columns.

GUI Name: ANSI Null Default

ANSI_Nulls
Specifies whether a return of NULL displays if either side of a comparison operation contains a null value. NOCHANGE leaves the database configuration as it is currently. TRUE returns NULL if either side of a comparison operation (=, <>) contains a null value. Otherwise, a true comparison is made, returning TRUE or FALSE. FALSE does not return NULL if either side of a comparison operation (=, <>) contains a null value.

GUI Name: ANSI Nulls

ANSI_Warnings
Specifies ANSI-92 standard behavior for several error conditions. NOCHANGE leaves the database configuration as it is currently. TRUE generates a message if null values appear in aggregate functions. In addition, the statement is rolled back if divide-by-zero or arithmetic...
overflow errors are encountered. **FALSE** does not generate a message for null values in aggregate functions. Null values are returned when divide-by-zero and arithmetic overflow errors are encountered.

**GUI Name:** **ANSI Warnings**

**Auto_Create_Statistics**
Specifies whether statistical information about the distribution of values in a column is automatically created and stored. The query processor and query optimizer use these statistics. **NOCHANGE** leaves the database configuration as it is currently. **TRUE** automatically creates and stores statistical information about the distribution of values in a column. **FALSE** does not create and store statistical information about the distribution of values in a column.

**GUI Name:** **Auto create statistics**

**Auto_Update_Statistics**
Specifies whether statistics are updated automatically when changes to the table make them obsolete. **NOCHANGE** leaves the database configuration as it is currently. **TRUE** automatically updates statistics when changes to the table make them obsolete (based on a statistical sampling algorithm). **FALSE** does not update statistics. You must update the statistics manually when needed.

**GUI Name:** **Auto update statistics**

**Autoclose**
Specifies whether the database closes and frees its resources when no user connections access the database. **NOCHANGE** leaves the database configuration as it is currently. **TRUE** closes the database and cleanly shuts down after the last database user exits and all processes complete. **FALSE** leaves the database open even if no user connection to the database is active.

**GUI Name:** **Autoclose**

**Autoshrink**
Specifies whether to use automatic shrinking on database and log files. **NOCHANGE** leaves the database configuration as it is currently. **TRUE** automatically shrink database and log files. Files are shrunk if they contain more than 25% unused space. **FALSE** does not automatically shrink any databases or log files.

**GUI Name:** **Autoshrink**

**Concat_Null_Yields_Null**
Specifies whether concatenating a null value to any string causes the string to be set to NULL. **Concatenation** is taking two or more character strings or expressions and placing them side-by-side next to each other to become one binary string or expression. **NOCHANGE** leaves the database configuration as it is currently. **TRUE** returns a value of NULL if a string is concatenated with a null value. **FALSE** does not return a NULL value if a string is concatenated with a null value. Concatenating a NULL to a string does not change the string.

**GUI Name:** **Concat null yields null**

**Cursor_Close_On_Commit**
Specifies whether open cursors are closed when a transaction is committed. **NOCHANGE** leaves the database configuration as it is currently. **TRUE**
closes any open cursors after a transaction is committed. **FALSE** does not close any open cursors after a transaction is committed.

GUI Name: Close on commit

**Dbo_Use_Only**

Specifies that only the database owner (DBO) or logins aliased to the DBO can access a database. **NOCHANGE** leaves the database configuration as it is currently. **TRUE** limits database access to the DBO or logins aliased to the DBO. **FALSE** does not limit database access to the DBO or logins aliased to the DBO.

GUI Name: dbo use only

**Default_To_Local_Cursor**

Specifies whether to create the cursor scope as local unless specified as global when created. **NOCHANGE** leaves the database configuration as it is currently. **TRUE** sets the cursor scope default to local unless global is specified when it is created. **FALSE** does not set the cursor scope default to local.

GUI Name: Default to local cursor

**merge_publish**

Specifies whether the database can be used for merge replication publications. **NOCHANGE** leaves the database configuration as it is currently. **TRUE** uses the database for merge replication publications. **FALSE** restricts the database from merge replication publications.

GUI Name: Merge publish

**Offline**

Specifies whether to take the database offline. **NOCHANGE** leaves the database configuration as it is currently. **TRUE** takes the database offline. **FALSE** does not take the database offline.

GUI Name: Offline

**Published**

Specifies whether to make the database available for replication publications. **NOCHANGE** leaves the database configuration as it is currently. **TRUE** uses the database for replication publications. **FALSE** restricts the database from replication publications.

GUI Name: Published

**Quoted_Identifier**

Specifies whether to delimit identifiers with double quotes and literal with single quotes. **NOCHANGE** leaves the database configuration as it is currently. **TRUE** specifies that identifiers can be delimited by double quotes and literals must be delimited by single quotes. **FALSE** specifies that identifiers cannot be in quotes and must follow all Transact-SQL rules for identifiers.

GUI Name: Quoted identifier

**Read_Only**

Specifies whether the database is read-only. **NOCHANGE** leaves the database configuration as it is currently. **TRUE** makes the database read-only. If it is read-only, you can read from the database, but you cannot make changes to it. **FALSE** does not make the database read-only. You can make changes to the database.
GUI Name: Read only

Recursive Triggers
Specifies whether to allow triggers to fire recursively. NOCHANGE leaves the database configuration as it is currently. TRUE allows triggers to fire recursively. Note that there is a risk of causing a race condition in table updates that are initiated by triggers. FALSE does not allow triggers to fire recursively.

GUI Name: Recursive triggers

Select Into
Specifies whether to allow you to use the SELECT INTO statement and BCP (Bulk Copy) command. NOCHANGE leaves the database configuration as it is currently. TRUE allows you to use the SELECT INTO statement and BCP command. FALSE does not allow you to use the SELECT INTO statement and BCP command.

GUI Name: Select into/bulk copy

Single_user
Specifies whether to put a database into single-user mode. Setting single-user mode does not disconnect current user connections. It simply prevents any new connections to the database. NOCHANGE leaves the database configuration as it is currently. TRUE sets the database to single-user mode. You can set this option only if there are no other users currently on the database. FALSE does not set the database to single-user mode.

GUI Name: Single user

Subscribed
Specifies whether to make the database available to be subscribed for publication. NOCHANGE leaves the database configuration as it is currently. TRUE makes the database available to be subscribed for publication. FALSE makes the database unavailable to be subscribed for publication.

GUI Name: Subscribed

Torn Page Detection
Specifies whether the server can detect incomplete I/O operations caused by system or power failures. Detection normally occurs during recovery. Detection of a torn page results in the database being marked suspect, an I/O error, and termination of the connection. NOCHANGE leaves the database configuration as it is currently. TRUE enables Microsoft SQL Server to detect incomplete I/O operations. FALSE does not enable Microsoft SQL Server to detect incomplete I/O operations.

GUI Name: Torn page detection

Trunc
Specifies whether or not the transaction log is truncated each time a checkpoint is issued. NOCHANGE leaves the database configuration as it is currently. TRUE truncates the transaction log each time a checkpoint is issued. FALSE does not truncate the transaction log each time a checkpoint is issued.

GUI Name: Truncate log on checkpoint
CLI example

    wruntask -t SetRetrieveDatabaseOptions
    -l ITM\MicrosoftSQLServerTasks
    -h @MicrosoftSQLServer:hbergin@abc
    -a database=EastCoast
    -a SelectIntoBulkCopy=TRUE

Usage notes

    This task collects invocation parameters and issues an sp_dboption command.

See also

    See also the following commands in the Tivoli Management Framework Reference Manual: wruntask, wcrtjob, and wgettask.
**StartMSSQLServerServices**

**Description**

Starts or continues any of the following Microsoft SQL Server services on a specified endpoint:

- Microsoft SQL Server Service (Database engine)
- ServerAgent (SQL server agent)
- MSDTC (Microsoft Distributed Transaction Coordinator)
- Microsoft Search (Microsoft search service — for Microsoft SQL Server 2000 only)

The **ServerAgent** is dependent upon the **Microsoft SQL Server Service**. If **ServerAgent** is started, **Microsoft SQL Server Service** also starts, if it is not running.

You can start the **ServerAgent** service and **Microsoft SQL Server Service** independently. **Continue** restarts a paused service. See "Usage Notes" for more information about these services.

**Authorization role**

mssqlserver_dba

**Microsoft SQL Server Permissions:** System Administrator

**Target endpoint**

MicrosoftSQLServer

**GUI data entry fields**

The following shows the **StartMSSQLServerServices** window, followed by a description of each field.

![StartMSSQLServerServices window](image)

**Start Microsoft SQL Server Service?**

Click **New State** to display a window with options for starting the Microsoft SQL Server service. Choose one of the following from the **Start Microsoft SQL Server Service** window:

- **NO CHANGE** maintains the current state setting.
START starts the database server in multi-user mode with full access.

CONTINUE continues the service, if it is paused.

**Start Microsoft SQL Server ServerAgent?**

Click New State to display a window with options for starting the Microsoft SQL Server ServerAgent. Choose one of the following from the Start Microsoft SQL Server ServerAgent window:

- **NO CHANGE** maintains the current state setting.
- **START** starts the service required for running Microsoft SQL Server tasks and replication.

**Start Microsoft SQL Server MSDTC?**

Click New State to display a window with options for starting the Microsoft SQL Server MSDTC. Choose one of the following from the **Start Microsoft SQL Server MSDTC** window:

- **NO CHANGE** maintains the current state setting.
- **START** starts the service required for distributed transaction control.

**Start Microsoft Search Service?**

(Microsoft SQL Server 2000 only) Click New State to display a window with options for starting the Microsoft Search Service. Choose one of the following from the **Start Microsoft Search Service?** window:

- **NO CHANGE** maintains the current state setting.
- **START** starts the Microsoft Search Service so you can use the search service.

### CLI syntax

Using the `wruntask` command:

```
-t StartMSSQLServerServices
-l ITMMicrosoftSQLServerTasks
-h <target for task>
[-a {START | NO_CHANGE | CONTINUE}]
[-a {START | NO_CHANGE}]
[-a {START | NO_CHANGE}]
[-a {START | NO_CHANGE}]
```

where:

**The first parameter refers to the Microsoft SQL Server Service**

- **START** starts the service required for any operations to be carried out on the server or its databases. **NO_CHANGE** does not alter the current state of the service. **CONTINUE** restarts the service after it has been paused.

**GUI Name:** Start Microsoft SQL Server Service?

**The second parameter refers to the Microsoft SQL Server ServerAgent**

- **START** starts the service required for running Microsoft SQL Server tasks and replication. **NO_CHANGE** does not alter the current state of the service.

**GUI Name:** Start Microsoft SQL Server ServerAgent?

**The third parameter refers to the Microsoft SQL Server MSDTC**

- **START** starts the service required for distributed transaction control. **NO_CHANGE** does not alter the current state of the service.
GUI Name: **Start Microsoft SQL Server MSDTC?**

The fourth parameter refers to the Microsoft Search Service  
(Microsoft SQL Server 2000 only) **START** starts the service that assists in complex search queries. **NO_CHANGE** does not alter the current state of the service.

GUI Name: **Start Microsoft Search Service?**

**CLI example**

```bash
wruntask -t StartMSSQLServerServices
-l ITM\MicrosoftSQLServerTasks
-h @MicrosoftSQLServer:hbergin@abc
-a START
-a START
-a START
-a START
```

**Usage notes**

This task starts the following services:

- **Microsoft SQL Server Service (Database engine)**
  The Microsoft SQL Server Service must be running for any operations to be carried out on the server or its databases. A user request or an internal error might stop the server. The server might also stop if the Windows computer was rebooted and the service is not automatically set to restart at boot time. You can configure an automatic response to try to start the service again. This reduces the amount of time the server is unavailable to users and applications.  
  The Microsoft SQL Server Service manages all the database files owned by an instance of SQL Server and processes all Transact-SQL statements sent from Microsoft SQL Server client applications. The service effectively allocates computer resources between multiple concurrent users. It also enforces business rules defined in stored procedures and triggers, ensures the consistency of the data, and prevents logical problems such as having two people trying to update the same data at the same time.

- **ServerAgent (SQL Server Agent)**
  Microsoft SQL Server Agent must be running for SQL scheduled jobs and replication to operate. If this service is stopped, your environment could experience replication bottleneck, a backlog of replicated transactions, or a delay in the transfer of published information to subscribers.  
  The Microsoft SQL Server Agent is a service that executes jobs, monitors Microsoft SQL Server, and fires alerts. This service enables you to automate certain administrative tasks. Ensure this service is running so that configured administrative tasks can run automatically.

- **MSDTC (Microsoft Distributed Transaction Coordinator)**
  The Microsoft Distributed Transaction Coordinator service must be running in order for distributed transactions to be coordinated correctly. A distributed transaction is a transaction within a single Microsoft SQL Server that spans two or more databases.  
  The Microsoft Distributed Transaction Coordinator is a transaction manager that enables client applications to include several different sources of data in one transaction. This service coordinates committing the distributed transaction across all the servers that are enlisted in the transaction. Microsoft SQL Server assists in a distributed transaction by:
  - Calling stored procedures on remote servers running Microsoft SQL Server
– Promoting a local transaction to a distributed transaction and enlisting remote servers in the transaction
– Making distributed updates that update data on multiple OLE DB data sources

Microsoft Distributed Transaction Coordinator assists in the successful completion of a distributed transaction. The service ensures that all specified updates on all the servers are made permanent. In the case of errors, the service ensures that all updates are erased.

• **Microsoft Search (Microsoft search service)**

  (Microsoft SQL Server 2000 only) The Microsoft Search service is a full-text indexing and search engine. The service provides indexing and querying support. Using the service enables Microsoft SQL Server to support more complex searches on character string columns.

**See also**

See also the following commands in the *Tivoli Management Framework Reference Manual*: `wruntask`, `wcrtjob`, and `wgettask`. 
StopMSSQLServerServices

Description
Stops or pauses any of the following Microsoft SQL Server services on a specified endpoint:

- Microsoft SQL Server Service (Database engine)
- ServerAgent (SQL server agent)
- MSDTC (Microsoft Distributed Transaction Coordinator)
- Microsoft Search (Microsoft search service — for Microsoft SQL Server 2000 only)

The ServerAgent is dependent on the Microsoft SQL Server Service. If the ServerAgent and the Microsoft SQL Server Service services are both running and Microsoft SQL Server Service stops, ServerAgent stops as well. See “Usage Notes” for more information about these services.

Authorization role
mssqlserver_dba

Microsoft SQL Server Permissions: System Administrator

Target endpoint
MicrosoftSQLServer

GUI data entry fields
The following shows the StopMSSQLServerServices window, followed by a description of each field.

Stop or Pause Microsoft SQL Server Service?
Click New State to display a window with options for stopping the Microsoft SQL Server Service. Choose one of the following from the Stop or Pause Microsoft SQL Server Service? window:

NO CHANGE maintains the current state setting.

PAUSE stops any new users from logging onto the server.

STOP stops the server without waiting for transactions to finish.
Stop or Pause Microsoft SQL Server ServerAgent?
Click New State to display a window with options for stopping the Microsoft SQL Server ServerAgent. Choose one of the following from the Stop or Pause Microsoft SQL Server ServerAgent? window:

NO CHANGE maintains the current state setting.

STOP stops the SQLServerAgent service. The service is required for running Microsoft SQL Server tasks and replication.

Stop Microsoft SQL Server MSDTC?
Click New State to display a window with options for stopping the Microsoft SQL Server MSDTC. Choose one of the following from the Stop Microsoft SQL Server MSDTC? window:

NO CHANGE maintains the current state setting.

STOP stops the service required for distributed transaction control.

Stop Microsoft Search Service?
(Microsoft SQL Server 2000 only) Click New State to display a window with options for stopping the Microsoft Search Service. Choose one of the following from the Stop Microsoft Search Service? window:

NO CHANGE maintains the current state setting.

STOP stops the Microsoft Search Service.

CLI syntax
Using the wruntask command:

```bash
–t StopMSSQLServerServices
–l ITMMicrosoftSQLServerTasks
–h <ObjectName>
[-a {STOP | NO_CHANGE | PAUSE}]
[-a {STOP | NO_CHANGE}]
[-a {STOP | NO_CHANGE}]
[-a {STOP | NO_CHANGE}]
```

where:

The first parameter refers to the Microsoft SQL Server Service

STOP stops the SQLExecutive service. NO_CHANGE does not alter the current state of the service. PAUSE pauses the service.

GUI Name: Stop or Pause Microsoft SQL Server Service?

The second parameter refers to the Microsoft SQL Server ServerAgent

STOP stops the SQLServerAgent service. NO_CHANGE does not alter the current state of the service.

GUI Name: Stop or Pause Microsoft SQL Server ServerAgent?

The third parameter refers to the Microsoft SQL Server MSDTC

STOP stops the service required for distributed transaction control.

NO_CHANGE does not alter the current state of the service.

GUI Name: Stop Microsoft SQL Server MSDTC?

The fourth parameter refers to the Microsoft Search Service

(Microsoft SQL Server 2000 only) STOP stops the service used for complex searches. NO_CHANGE does not alter the current state of the service.
GUI Name: **Stop Microsoft Search Service?**

### CLI example

```
wruntask -t StopMSSQLServerServices
-l ITM\MicrosoftSQLServerTasks
-h @MicrosoftSQLServer:VISION117@abc
-a STOP
-a STOP
-a STOP
-a STOP
```

### Usage notes

This task stops the following services:

- **Microsoft SQL Server Service (Database engine)**
  
The Microsoft SQL Server Service is required for any operations to be carried out on the server or its databases.

  The *Microsoft SQL Server Service* manages all the database files owned by an instance of SQL Server, and processes all Transact-SQL statements sent from Microsoft SQL Server client applications. The service effectively allocates computer resources between multiple concurrent users. It also enforces business rules defined in stored procedures and triggers, ensures the consistency of the data, and prevents logical problems such as having two people trying to update the same data at the same time.

- **ServerAgent (SQL Server Agent)**

  Microsoft SQL Server Agent is required for SQL scheduled jobs and replication to operate. If this service is stopped, your environment might experience replication bottleneck, a backlog of replicated transactions, or a delay in the transfer of published information to subscribers.

  The *Microsoft SQL Server Agent* is a service that runs jobs, monitors Microsoft SQL Server, and fires alerts. This service enables you to automate certain administrative tasks. This service must run for configured administrative tasks to run automatically.

- **MSDTC (Microsoft Distributed Transaction Coordinator)**

  The Microsoft Distributed Transaction Coordinator service is required for distributed transactions to be coordinated correctly. A distributed transaction is a transaction within a single Microsoft SQL Server that spans two or more databases.

  The *Microsoft Distributed Transaction Coordinator* is a transaction manager that enables client applications to include several different sources of data in one transaction. This service coordinates committing the distributed transaction across all the servers that are enlisted in the transaction. Microsoft SQL Server assists in a distributed transaction by:

  - Calling stored procedures on remote servers running Microsoft SQL Server
  - Promoting a local transaction to a distributed transaction and enlisting remote servers in the transaction
  - Making distributed data updates on multiple OLE DB data sources

  Microsoft Distributed Transaction Coordinator assists in the successful completion of a distributed transaction. The service ensures that all specified updates on all the servers are made permanent. In the case of errors, the service ensures that all updates are erased.

- **Microsoft Search (Microsoft search service)**
(Microsoft SQL Server 2000 only) The Microsoft Search service is a full-text indexing and search engine. The service provides indexing and querying support. Using the service enables Microsoft SQL Server to support more complex searches on character string columns.

The SQLServerAgent service is dependent upon the MSSQLServer service. If both are running and MSSQLServer is stops, SQLServerAgent also stops.

See also

See also the following commands in the Tivoli Management Framework Reference Manual: wruntask, wcrtjob, and wgettask.
TDPFullBackup

Description
Performs a backup of an entire database and its associated transaction log. You have the option to truncate the transaction log after the backup.

The backup is performed correctly regardless of whether the transaction log is on the same device fragment as the database or not.

Note: Verify the following before using TDPFullBackup:
- The database server installed on the endpoint is registered by IBM Tivoli Monitoring for Databases: Microsoft SQL Server.
- The Storage Agent is installed and configured to back up the database server on the endpoint.

Authorization role
mssqlserver_dba

Target endpoint
MicrosoftSQLServer

GUI data entry fields
The following shows the TDPFullBackup window, followed by a description of each field.

Database name(s)
Specify the database on which you want to run this task. Separate database names with commas. You can use an asterisk (*) to run this task on all databases.
**Truncate log after backup**
Check to clear out the inactive part of the transaction log. The default is not to truncate the log after a backup.

**Mount wait**
Check to make the SQL Agent wait for the Tivoli Storage Manager server to complete mount requests.

**Quiet**
Check to prevent status information from displaying.

**TSM Node**

*Note:* If you specify **TSM Node**, you must change the **PASSWORDAccess** value in the Tivoli Data Protection options file to "PROMPT."

Name that the Tivoli Data Protection Agent uses to logon to the Tivoli Storage Manager server. If you do not specify a value for **TSM Node**, the task uses the default value, which is **client**. You can store a password in the options file, but this parameter overrides the value in the options file.

If you type a value for **TSM Node**, you also must type a value for **TSM Password**. If the value for **TSM Password** does not match the password assigned to **TSM Node**, the task fails.

**TSM Password**
Password the Tivoli Data Protection Agent uses to logon to the Tivoli Storage Manager server. The default password is blank. You can store a password in the options file, but this parameter overrides the value in the options file.

**TSM Options File**
Specifies the name of the Tivoli Storage Manager options file that the SQL Agent uses. Type only the filename with the `.opt` extension. The file must already exist in the install path of the SQL Agent. If you type a full path name, the task fails.

**Activity Log File**
Specifies the name of the activity log. Statistics about backed up and restored database objects are written to the specified log file. This option requires only the file name. If you do not specify a path, Tivoli Storage Manager stores the file in a default location in its install path. You can force the file to be saved to a specific directory with forward slashes (/). The task ignores a back slash (\) and cannot interpret double (escape) back slashes (\\).

If you specify only the filename without a path name, Tivoli Storage Manager stores the file in a default location in its install path.

**Number of buffers**
Number of buffers the SQL Agent uses to improve asynchronous communications with the Tivoli Storage Manager and database servers.

**CLI syntax**
This task can be run with no arguments; it uses default values.

Using the `wruntask` command:

```
-t TDPFullBackup
-l ITMMicrosoftSQLServerTasks
-h <Object Name>
[-a Databases=<{database_name} | <databasename>,...,<databasename>]]
```
[–a Truncate=Y]
[–a Mountwait=Y]
[–a Quiet=Y]
[–a TSMNode=<nodename>]
[–a TSMPwd=<node password>]
[–a TSMOptFile=<options file>.opt]
[–a LogFile=<tdp logfile>]
[–a Buffers=<buffernumber>]

where:

Databases

<database_name> specifies the name of the database on which you want this
task to run. You can use an asterisk (*) to run this task on all databases. If
no databases are listed, all databases are backed up.

Truncate

Y truncates the transaction log. Use this option only when you want to
truncate the log.

GUI Name: Truncate log after backup

MountWait

Y tells the SQL Agent to wait for the Tivoli Data Protection server to
complete mount requests. Use this option only when you want to activate
the MountWait function.

GUI Name: Mount wait

Quiet

Y does not display status information. Use this option only when you do
not want status displayed.

GUI Name: Quiet

TSMNode

Note: If you specify TSMNode, you must change the PASSWORDAccess
value in the Tivoli Data Protection options file to "PROMPT."
<nodename> designates the name the SQL Agent uses to logon to the Tivoli
Data Protection server. The default nodename is client.

GUI Name: TSM Node

TSMPwd

<node password> designates the password the SQL Agent uses to logon to
the Tivoli Data Protection server. The default is blank.

GUI Name: TSM Password

TSMOptFile

<options file>.opt designates the options file the SQL Agent uses. Type only
the filename with the .opt extension.

GUI Name: TSM Options File

LogFile

<tdp logfile> designates the name of the activity log.

GUI Name: Activity Log File

Buffers

<buffernumber> designates the number of buffers the SQL Agent uses to
improve asynchronous communications.
GUI Name: **Number of buffers**

**CLI example**

```
wruntask -t TDPFullBackup
-1 ITM\MicrosoftSQLServerTasks
-h @MicrosoftSQLServer:hbergin@abc
-a Databases=EastCoast
-a Truncate=Y
-a MountWait=Y
-a TSMNode=client
-a TSMPwd=client
```

**Usage notes**

The **TDPFullBackup** task output window does not return success or failure status. To verify task completion status, see the task activity log for information. The default name of the activity log file is **tdpsql.log**. This log file might be under a different name if you changed the name using the **Activity Log** option in the Desktop or the **Logfile** option in the command line.

**TDPFullBackup** makes online backups of one or more databases to Tivoli Data Protector (TDP) storage.

**Note:** Database back up can take a large amount of time, depending on the size of the database. If you want to back up only the transaction log, use the **TDPIncBackup** task.

This task prompts for values and then connects to the Tivoli Storage Manager (TSM) server through the Tivoli Data Protection Client. The task uses the values to create the **tdpsqlc** command.

The following steps occur when you run the **TDPFullBackup**:

1. **TDPFullBackup** connects to and prompts the Storage Agent to perform a backup with the arguments provided.
2. The Storage Agent starts a session with a Tivoli Storage Manager server using the Tivoli Storage Manager API and the information contained in a client options file.
3. The Storage Agent instructs the database server to begin a dump of the selected database objects.
4. The Storage Agent receives data from the database server and forwards it to the Tivoli Storage Manager server.
5. The Storage Agent ends the Tivoli Storage Manager and database server sessions.

Microsoft SQL Server 2000 and Microsoft SQL Server 7.0 are supported by Storage Agent version 2.2.

When the Tivoli Storage Manager server receives database objects, it stores the objects’ attributes in its own database and writes the database objects’ data to its storage hierarchy. Tivoli Storage Manager policies control all aspects of how and where the data is actually stored on the Tivoli Storage Manager server. A backup creates a copy of a database or transaction log on storage media. The Tivoli Storage Manager server is effectively the dump device when backing up data. The Tivoli Storage Manager server hides the details of the actual dump devices.
See also

See also the following commands in the *Tivoli Management Framework Reference Manual*: `wruntask`, `wcrtask`, `wcrjob`, and `wgettask`. 
**TDPlncBackup**

**Description**
Performs a backup of only transaction logs. Incremental backups reduce network usage, server storage, and the time required for backups. You can back up a transaction log only when it is on a separate device than the database. You can choose not to truncate the log.

**Authorization role**
mssqlserver_dba

**Target endpoint**
MicrosoftSQLServer

**GUI data entry fields**
The following shows the **TDPlncBackup** window, followed by a description of each field.

![TDPlncBackup window](image)

**Database name(s)**
Specify the database on which you want to run this task. Separate database names with commas. You can use an asterisk (*) to run this task on all databases.

**Truncate log after backup**
Clears out the inactive part of the transaction log. The default is not to truncate the log after a backup.

**Mount wait**
SQL Agent should wait for the Tivoli Storage Manager server to complete mount requests.

**Quiet**
Prevents status information from displaying.

**TSM Node**
Note: If you specify **TSM Node**, you must change the **PASSWORDAccess** value in the Tivoli Data Protection options file to "PROMPT."

Name the Tivoli Data ProtectionAgent uses to logon to the Tivoli Storage Manager server. If you do not specify a value for **TSM Node**, the task uses the default value, which is **client**. You can store a password in the options file, but this parameter overrides the value in the options file. If you type a value for **TSM Node** you also must type a value for **TSM Password**. If the value for **TSM Password** does not match the password assigned to **TSM Node**, the task fails.

**TSM Password**

Password the Tivoli Data Protection Agent uses to logon to the Tivoli Storage Manager server. The default password is blank. You can store a password in the options file, but this parameter overrides the value in the options file.

**TSM Options File**

Name of the Tivoli Storage Manager options file the SQL Agent uses. Type only the filename with the .opt extension. The file must already exist in the install path of the SQL Agent. If you type a full path name, the task fails.

**Activity Log File**

Name of the activity log. Statistics about backed up and restored database objects are written to the specified log file. This option requires only the filename. If you do not specify a path, Tivoli Storage Manager stores the file in a default location in its install path. You can force the file to be saved to a specific directory with forward slashes (/). The task ignores a back slash (\) and cannot interpret double (escape) back slashes (\\). If you specify only the filename without a path name, Tivoli Storage Manager stores the file in a default location in its install path.

**Number of buffers**

Number of buffers the SQL Agent uses to improve asynchronous communications with the Tivoli Storage Manager and database servers.

**CLI syntax**

This task can be run with no arguments; it uses default values.

Using the **wruntask** command:

```bash
-t TDPIncBackup
-l ITMMicrosoftSQLServerTasks
-h <Object Name>
[-a Databases=\{<database_name> | <databasename>,..,<databasename>\}]
[-a Truncate=Y]
[-a Mountwait=Y]
[-a Quiet=Y]
[-a TSMNode=<nodename>]
[-a TSMPwd=<node password>]
[-a TSMOptFile=<options file>.opt]
[-a LogFile=<tdp logfile>]
[-a Buffers=<buffernumber>]
```

where:

**Databases**

\(<database_name>\) specifies the name of the database on which you want this
task to run. You can use an asterisk (*) to run this task on all databases. If no databases are listed, all databases are backed up.

**Truncate**

Y truncates the transaction log. Use this option only when you want to truncate the log.

GUI Name: **Truncate log after backup**

**MountWait**

Y tells the SQL Agent to wait for the Tivoli Data Protection server to complete mount requests. Use this option only when you want to activate the MountWait function.

GUI Name: **Mount wait**

**Quiet**

Y does not display status information. Use this option only when you do not want status displayed.

GUI Name: **Quiet**

**TSMNode**

*Note:* If you specify **TSMNode**, you must change the **PASSWORDAccess** value in the Tivoli Data Protection options file to "PROMPT."

*<nodename>* designates the name the SQL Agent uses to logon to the Tivoli Data Protection server. The default nodename is **client**.

GUI Name: **TSM Node**

**TSMPwd**

*<node password>* designates the password the SQL Agent uses to logon to the Tivoli Data Protection server. The default is blank.

GUI Name: **TSM Password**

**TSMOptFile**

*<options file>.opt* designates the options file the SQL Agent uses. Type only the filename with the .opt extension.

GUI Name: **TSM Options File**

**LogFile**

*<tdp logfile>* designates the name of the activity log.

GUI Name: **Activity Log File**

**Buffers**

*<buffernumber>* designates the number of buffers the SQL Agent uses to improve asynchronous communications.

GUI Name: **Number of buffers**

**CLI example**

```
wruntask -t TDPIncBackup
-l ITMMicrosoftSQLServerTasks
-h @MicrosoftSQLServer:hebergin@abc
-a Databases=EastCoast
-a Truncate=Y
```

**Usage notes**

The **TDPincBackup** task output window does not return success or failure status. To verify task completion status, see the task activity log for information. The
default name of the activity log file is `tdpsql.log`. This log file might be under a different name if you changed the name using the Activity Log option in the Desktop or the Logfile option in the command line.

TDPincBackup makes online backups of one or more transaction logs to the Tivoli Data Protector (TDP) storage.

This task prompts for values and then connects to the Tivoli Storage Manager (TSM) Server through the Tivoli Data Protection Client.

The following steps occur when you run the TDPincBackup:

1. TDPincBackup connects and prompts the Storage Agent to perform a backup with the arguments provided.
2. The Storage Agent starts a session with the database server using DB-LIBRARY.
3. The Storage Agent starts a session with a Tivoli Storage Manager server using the Tivoli Storage Manager APU and the information contained in a client options file.
4. The Storage Agent instructs the database server to begin a dump of the selected database transaction log.
5. The Storage Agent receives data from the database server and forwards it to the Tivoli Storage Manager server.
6. The Storage Agent ends the session between the Tivoli Storage Manager and database server.

When the Tivoli Storage Manager server receives database objects, it stores the objects’ attributes in its own database and writes the database objects’ data to its storage hierarchy. Tivoli Storage Manager policies control all aspects of how and where the data is actually stored on the Tivoli Storage Manager server. A backup creates a copy of the transaction log on storage media. The Tivoli Storage Manager server is effectively the dump device when backing up data using this task. The Tivoli Storage Manager server hides the details of the actual dump devices.

**See also**

See also the following commands in the *Tivoli Management Framework Reference Manual*: wruntask, wcrtjob, and wgettask.
Appendix A. Functionality mapping: resource model to monitor

Although the release of IBM Tivoli Monitoring for Databases: Microsoft SQL Server is a new product offering, some functionality comparisons to previous Tivoli software offerings might be made.

IBM Tivoli Monitoring for Databases: Microsoft SQL Server offers many new features and monitoring capability, but it also captures many of the quality monitoring capability of previous Tivoli software releases. Use the following tables for functionality mapping:

- Table 7 compares the Tivoli Manager for Microsoft SQL Server monitors with the IBM Tivoli Monitoring for Databases: Microsoft SQL Server resource models.
- Table 8 on page 204 compares the IBM Tivoli Monitoring for Databases: Microsoft SQL Server resource models with the Tivoli Manager for Microsoft SQL Server monitors.
- Table 9 on page 204 compares the Microsoft SQL Server version 7 rules in Tivoli Management Solution for Microsoft SQL with the IBM Tivoli Monitoring for Databases: Microsoft SQL Server resource models.
- Table 10 on page 205 compares the Microsoft SQL Server version 2000 rules in Tivoli Management Solution for Microsoft SQL with the IBM Tivoli Monitoring for Databases: Microsoft SQL Server resource models.

For complete details on the IBM Tivoli Monitoring for Databases: Microsoft SQL Server resource models, see Chapter 2, “Resource models”, on page 5.

<table>
<thead>
<tr>
<th>Table 7. Monitor and resource model comparison</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tivoli Manager for Microsoft SQL Server, Version 1.3 monitor name</td>
</tr>
<tr>
<td><strong>Server monitoring</strong></td>
</tr>
<tr>
<td>Cache Average Free Page Scan</td>
</tr>
<tr>
<td>Cache Configured Free Buffers Percent Used</td>
</tr>
<tr>
<td>Cache Hit Ratio</td>
</tr>
<tr>
<td>Cache Maximum Free Page Scan</td>
</tr>
<tr>
<td>Cache Number of Free Buffers</td>
</tr>
<tr>
<td>Client Count</td>
</tr>
<tr>
<td>Client Count Percent Used</td>
</tr>
<tr>
<td>CPU Percent Busy</td>
</tr>
<tr>
<td>CPU Percent Busy Doing I/O</td>
</tr>
<tr>
<td>CPU Percent Idle</td>
</tr>
<tr>
<td>Database Status (Server)</td>
</tr>
<tr>
<td>Data Space Percent Used (Server)</td>
</tr>
<tr>
<td>Delivered Latency</td>
</tr>
<tr>
<td>Delivered Transaction Rate</td>
</tr>
<tr>
<td>Delivered Transactions</td>
</tr>
</tbody>
</table>
Table 7. Monitor and resource model comparison (continued)

<table>
<thead>
<tr>
<th>Tivoli Manager for Microsoft SQL Server, Version 1.3 monitor name</th>
<th>IBM Tivoli Monitoring for Databases: Microsoft SQL Server, Version 5.1.1 resource model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extent Locks Exclusive</td>
<td>None</td>
</tr>
<tr>
<td>Extent Locks Total</td>
<td>None</td>
</tr>
<tr>
<td>Extent Locks Update</td>
<td>None</td>
</tr>
<tr>
<td>Freeform SQL Numeric</td>
<td>None</td>
</tr>
<tr>
<td>Freeform SQL String</td>
<td>None</td>
</tr>
<tr>
<td>I/O Batch Average Size</td>
<td>None</td>
</tr>
<tr>
<td>I/O Batch Maximum Size</td>
<td>None</td>
</tr>
<tr>
<td>I/O Batch Write Rate</td>
<td>None</td>
</tr>
<tr>
<td>I/O Disk Errors</td>
<td>Errors</td>
</tr>
<tr>
<td>I/O Disk Reads</td>
<td>None</td>
</tr>
<tr>
<td>I/O Disk Writes</td>
<td>None</td>
</tr>
<tr>
<td>I/O Log Write Rate</td>
<td>None</td>
</tr>
<tr>
<td>I/O Outstanding Reads</td>
<td>None</td>
</tr>
<tr>
<td>I/O Outstanding Writes</td>
<td>None</td>
</tr>
<tr>
<td>I/O Page Read Rate</td>
<td>Cache/CPU</td>
</tr>
<tr>
<td>I/O Single Page Write Rate</td>
<td>None</td>
</tr>
<tr>
<td>I/O Transaction Rate</td>
<td>Cache/CPU</td>
</tr>
<tr>
<td>I/O Transactions Per Log Record</td>
<td>None</td>
</tr>
<tr>
<td>Intent Locks Exclusive</td>
<td>None</td>
</tr>
<tr>
<td>Intent Locks Shared</td>
<td>None</td>
</tr>
<tr>
<td>Intent Locks Total</td>
<td>None</td>
</tr>
<tr>
<td>Log Space Percent Used (Server)</td>
<td>Space Usage</td>
</tr>
<tr>
<td>Network Read Rate</td>
<td>None</td>
</tr>
<tr>
<td>Network Write Rate</td>
<td>None</td>
</tr>
<tr>
<td>NT Application Log</td>
<td>None</td>
</tr>
<tr>
<td>NT Application Log Age</td>
<td>None</td>
</tr>
<tr>
<td>NT Application Log Percent Full</td>
<td>None</td>
</tr>
<tr>
<td>Number of Blocked Processes</td>
<td>Users/Transactions</td>
</tr>
<tr>
<td>Number of Deadlocks</td>
<td>Locks</td>
</tr>
<tr>
<td>Oldest Open Transaction (Server)</td>
<td>None</td>
</tr>
<tr>
<td>Page Locks Exclusive</td>
<td>None</td>
</tr>
<tr>
<td>Page Locks Shared</td>
<td>None</td>
</tr>
<tr>
<td>Page Locks Update</td>
<td>None</td>
</tr>
<tr>
<td>Procedure Buffers Active</td>
<td>None</td>
</tr>
<tr>
<td>Procedure Buffers Total</td>
<td>None</td>
</tr>
<tr>
<td>Procedure Buffers Used</td>
<td>None</td>
</tr>
<tr>
<td>Procedure Cache Active</td>
<td>None</td>
</tr>
<tr>
<td>Procedure Cache Size</td>
<td>None</td>
</tr>
<tr>
<td>Procedure Cache Used</td>
<td>None</td>
</tr>
</tbody>
</table>
Table 7. Monitor and resource model comparison (continued)

<table>
<thead>
<tr>
<th>Tivoli Manager for Microsoft SQL Server, Version 1.3 monitor name</th>
<th>IBM Tivoli Monitoring for Databases: Microsoft SQL Server, Version 5.1.1 resource model</th>
</tr>
</thead>
<tbody>
<tr>
<td>RA Effectiveness</td>
<td>None</td>
</tr>
<tr>
<td>RA Pages Fetched into Cache Per Second</td>
<td>Cache/CPU</td>
</tr>
<tr>
<td>RA Pages Found in Cache Per Second</td>
<td>None</td>
</tr>
<tr>
<td>RA Physical Read Rate</td>
<td>None</td>
</tr>
<tr>
<td>RA Slots Used</td>
<td>None</td>
</tr>
<tr>
<td>Service State</td>
<td>Availability</td>
</tr>
<tr>
<td>SQL Server State</td>
<td>Availability</td>
</tr>
<tr>
<td>SQLServerAgent Failed Jobs</td>
<td>Jobs</td>
</tr>
<tr>
<td>Table Locks Exclusive</td>
<td>None</td>
</tr>
<tr>
<td>Table Locks Shared</td>
<td>None</td>
</tr>
<tr>
<td>Table Locks Total</td>
<td>None</td>
</tr>
<tr>
<td>Total Blocking Locks</td>
<td>None</td>
</tr>
<tr>
<td>Total Exclusive Locks</td>
<td>None</td>
</tr>
<tr>
<td>Total Locks</td>
<td>None</td>
</tr>
<tr>
<td>Total Locks Percent Used</td>
<td>None</td>
</tr>
<tr>
<td>Total Locks Remaining</td>
<td>None</td>
</tr>
<tr>
<td>Total Shared Locks</td>
<td>None</td>
</tr>
<tr>
<td>Undelivered Transactions</td>
<td>None</td>
</tr>
<tr>
<td>User Connections</td>
<td>Users/Transactions (Logged metric only)</td>
</tr>
<tr>
<td>User Connections Percent Used</td>
<td>Users/Transactions</td>
</tr>
<tr>
<td>User Connections Remaining</td>
<td>None</td>
</tr>
</tbody>
</table>

**Database Monitoring**

| Data Space Percent Used (Database)                           | Space Usage                                                             |
| Data Space Used                                              | Space Usage                                                             |
| Database Status (Database)                                   | Availability                                                            |
| Fragmentation                                                | None                                                                     |
| Log Space Percent Used (Database)                            | Space Usage                                                             |
| Log Space Used                                               | Space Usage                                                             |
| Oldest Open Transaction (Database)                           | None                                                                     |
| Optimizer Statistics Age                                     | None                                                                     |
| PeopleSoft Fragmentation                                      | None                                                                     |
| PeopleSoft Optimizer Statistics Age                          | None                                                                     |
| PeopleSoft Space Used Table                                  | None                                                                     |
| Replicated Transaction Rate                                  | Replication                                                             |
| Replicated Transactions                                      | None                                                                     |
| Replication Latency                                          | Replication                                                             |
| Space Used (Table)                                            | Space Usage                                                             |
| Suspect Database                                             | Availability                                                            |
### Table 8. Resource model and monitor comparison

<table>
<thead>
<tr>
<th>IBM Tivoli Monitoring for Databases: Microsoft SQL Server, Version 5.1.1 resource model</th>
<th>Tivoli Manager for Microsoft SQL Server, Version 1.3 monitor name</th>
</tr>
</thead>
</table>
| **Availability** | Database Status  
Service State  
SQL Server State  
Suspect Database |
| **Cache/CPU** | Cache Hit Ratio  
Cache Number of Free Buffers  
CPU Percent Busy  
I/O Page Read Rate  
I/O Transaction Rate  
I/O Transactions Per Log Record  
RA Pages Fetched into Cache Per Second |
| **Errors** | I/O Disk Errors |
| **Jobs** | SQLServerAgent Failed Jobs |
| **Locks** | Number of Blocked Processes  
Number of Deadlocks |
| **Replication** | Delivered Latency  
Delivered Transaction Rate  
Replicated Transaction Rate  
Replication Latency |
| **Replication Jobs** | **New** |
| **Space Usage** | Data Space Percent Used (Database)  
Data Space Percent Used (Server)  
Data Space Used  
Log Space Percent Used (Database)  
Log Space Percent Used (Server)  
Log Space Used  
Space Used (Table) |
| **Users/Transactions** | Number of Blocked Processes  
I/O Transaction Rate  
User Connections  
User Connections Percent Used |

### Table 9. Tivoli Management Solution for Microsoft SQL 7.0 rules to resource model comparison

<table>
<thead>
<tr>
<th>SQL Server 7.0 Rule</th>
<th>IBM Tivoli Monitoring for Databases: Microsoft SQL Server, Version 5.1.1 resource model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allocate_Space_Failed</td>
<td>Errors</td>
</tr>
<tr>
<td>Blocked_Processes_Alert</td>
<td>Users/Transactions</td>
</tr>
<tr>
<td>Blocked_Processes_Details</td>
<td>None</td>
</tr>
<tr>
<td>Connection_Failed</td>
<td>Errors</td>
</tr>
<tr>
<td>Connections_High</td>
<td>Users/Transactions</td>
</tr>
<tr>
<td>Corrupted_Database</td>
<td>None</td>
</tr>
<tr>
<td>Data_Throughput</td>
<td>None</td>
</tr>
<tr>
<td>Database_Growth</td>
<td>Space Usage</td>
</tr>
<tr>
<td>Databases_Present</td>
<td>None</td>
</tr>
</tbody>
</table>
Table 9. Tivoli Management Solution for Microsoft SQL 7.0 rules to resource model comparison (continued)

<table>
<thead>
<tr>
<th>SQL Server 7.0 Rule</th>
<th>IBM Tivoli Monitoring for Databases: Microsoft SQL Server, Version 5.1.1 resource model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deadlock_Alert</td>
<td>Errors</td>
</tr>
<tr>
<td>Errors_Detected</td>
<td>Errors</td>
</tr>
<tr>
<td>Full_Databases</td>
<td>Space Usage</td>
</tr>
<tr>
<td>Full_Logs</td>
<td>None</td>
</tr>
<tr>
<td>Generate_DB_Used_SQL</td>
<td>Space Usage</td>
</tr>
<tr>
<td>Generate_DB_Used_Statistics</td>
<td>None</td>
</tr>
<tr>
<td>Insufficient_Locks</td>
<td>None</td>
</tr>
<tr>
<td>Log_Growth</td>
<td>None</td>
</tr>
<tr>
<td>Replication_Latency</td>
<td>Replication</td>
</tr>
<tr>
<td>Server_Memory</td>
<td>Space Usage</td>
</tr>
<tr>
<td>Set_SQL7_Errorlog</td>
<td>Errors</td>
</tr>
<tr>
<td>SQL7_Buffer_Cache_Efficiency</td>
<td>Cache/CPU</td>
</tr>
<tr>
<td>SQL7_Disks</td>
<td>None</td>
</tr>
<tr>
<td>SQL7_IO_Errors</td>
<td>Errors</td>
</tr>
<tr>
<td>SQL7_Process_Busy</td>
<td>None</td>
</tr>
<tr>
<td>SQL7_Process_CPU</td>
<td>None</td>
</tr>
<tr>
<td>SQL7_Recompiles</td>
<td>None</td>
</tr>
<tr>
<td>SQL7_Services</td>
<td>Availability</td>
</tr>
<tr>
<td>SQL7_Transaction_Log_Full</td>
<td>None</td>
</tr>
<tr>
<td>Total_Database_Growth</td>
<td>None</td>
</tr>
<tr>
<td>Total_Log_Growth</td>
<td>None</td>
</tr>
<tr>
<td>Unsafe_Auto_Params</td>
<td>None</td>
</tr>
<tr>
<td>Warnings_Detected</td>
<td>None</td>
</tr>
<tr>
<td>Workspace_Memory</td>
<td>None</td>
</tr>
<tr>
<td>Workspace_Memory_Grants</td>
<td>None</td>
</tr>
</tbody>
</table>

Table 10. Tivoli Management Solution for Microsoft SQL 2000 rules to resource model comparison

<table>
<thead>
<tr>
<th>SQL Server 2000 Rule</th>
<th>IBM Tivoli Monitoring for Databases: Microsoft SQL Server, Version 5.1.1 resource model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blocked_SS2K_Processes_Alert</td>
<td>Users/Transactions</td>
</tr>
<tr>
<td>Blocked_SS2K_Processes_Details</td>
<td>None</td>
</tr>
<tr>
<td>SS2K_Allocate_Space_Failed</td>
<td>Errors</td>
</tr>
<tr>
<td>SS2K_Buffer_Cache_Efficiency</td>
<td>Cache/CPU</td>
</tr>
<tr>
<td>SS2K_Connection_Failed</td>
<td>Errors</td>
</tr>
<tr>
<td>SS2K_Connections_High</td>
<td>Users/Transactions</td>
</tr>
<tr>
<td>SS2K_Corrupted_Database</td>
<td>Errors</td>
</tr>
<tr>
<td>SS2K_Data_Throughput</td>
<td>None</td>
</tr>
<tr>
<td>SQL Server 2000 Rule</td>
<td>IBM Tivoli Monitoring for Databases: Microsoft SQL Server, Version 5.1.1 resource model</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>----------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SS2K_Database_Growth</td>
<td>Space Usage</td>
</tr>
<tr>
<td>SS2K_Deadlock_Alert</td>
<td>Errors</td>
</tr>
<tr>
<td>SS2K_Disk_Space</td>
<td>None</td>
</tr>
<tr>
<td>SS2K_Errors_Detected</td>
<td>Errors</td>
</tr>
<tr>
<td>SS2K_Full_Databases</td>
<td>Space Usage</td>
</tr>
<tr>
<td>SS2K_Full_Logs</td>
<td>None</td>
</tr>
<tr>
<td>SS2K_Insufficient_Locks</td>
<td>None</td>
</tr>
<tr>
<td>SS2K_IO_Errors</td>
<td>Errors</td>
</tr>
<tr>
<td>SS2K_Log_Growth</td>
<td>None</td>
</tr>
<tr>
<td>SS2K_Performance_Checks</td>
<td>Cache/CPU</td>
</tr>
<tr>
<td>SS2K_Process_Load</td>
<td>None</td>
</tr>
<tr>
<td>SS2K_Processes</td>
<td>None</td>
</tr>
<tr>
<td>SS2K_Recompiles</td>
<td>None</td>
</tr>
<tr>
<td>SS2K_Replication_Latency</td>
<td>Replication</td>
</tr>
<tr>
<td>SS2K_Server_Memory</td>
<td>None</td>
</tr>
<tr>
<td>SS2K_Services</td>
<td>Availability</td>
</tr>
<tr>
<td>SS2K_Total_Database_Growth</td>
<td>Space Usage</td>
</tr>
<tr>
<td>SS2K_Total_Log_Growth</td>
<td>None</td>
</tr>
<tr>
<td>SS2K_Transaction_Log_Full</td>
<td>None</td>
</tr>
<tr>
<td>SS2K_Unsafe_Auto_Params</td>
<td>None</td>
</tr>
<tr>
<td>SS2K_Warnings_Detected</td>
<td>None</td>
</tr>
<tr>
<td>SS2K_Workspace_Memory</td>
<td>None</td>
</tr>
<tr>
<td>SS2K_Workspace_Memory_Grants</td>
<td>None</td>
</tr>
</tbody>
</table>
Appendix B. Tivoli Enterprise Console classes

This appendix contains the Tivoli Enterprise Console classes delivered with IBM Tivoli Monitoring for Databases: Microsoft SQL Server. The classes contain the new slot values for the event.

 Slots are the fields in a Tivoli event. All Tivoli events have a base set of slots as described in the Tivoli Enterprise Console User's Guide. In addition to these slots, IBM Tivoli Monitoring for Databases: Microsoft SQL Server adds slots for additional information. You can reference these classes and slots to assist you in writing your own business rules.

Resource models

Collectively, the resource model files implement the following class hierarchy:

```
TMW_Event
  ITMSystem_Base
  ITMDatabase_Base
  ITMMSSQL_Base
    MSSQL_Server
      individual resource model classes
```

**ITMAApplications.baroc**

```
TEC_CLASS :
  ITMSystem_Base ISA TMW_Event;
END

TEC_CLASS :
  ITMDatabase_Base ISA TMW_Event;
END
```

**ITMMSSQL.baroc**

```
TEC_CLASS :
  ITMMSSQL_Base ISA ITMDatabase_Base
    DEFINES {
      interp   : STRING;
      context  : STRING;
      mssqlservername : STRING;
      application_oid : STRING;
      application_version : STRING;
      application_class : STRING;
      application_label : STRING;
    };
END

TEC_CLASS :
  MSSQL_Server ISA ITMMSSQL_Base;
END

TEC_CLASS :
  MSSQL_Database ISA MSSQL_Server;
END
```
Availability resource model

**Microsoft_SQL_Server_Stopped_Service**

TEC_CLASS :
Microsoft_SQL_Server_Stopped_Service ISA MSSQL_Server
DEFINES {
  mssqlservicename : STRING;
  mssqlservicestate : STRING;
  mssqlservicestatus : STRING;
  mssqlstartmode : STRING;
  severity: default = CRITICAL;
};
END

**Microsoft_SQL_Server_Paused_Service**

TEC_CLASS :
Microsoft_SQL_Server_Paused_Service ISA MSSQL_Server
DEFINES {
  mssqlservicename : STRING;
  mssqlservicestate : STRING;
  mssqlservicestatus : STRING;
  mssqlstartmode : STRING;
  severity: default = WARNING;
};
EN

**Microsoft_SQL_Server_Suspect_Database**

TEC_CLASS :
Microsoft_SQL_Server_Suspect_Database ISA MSSQL_Server
DEFINES {
  mssqldatabasename : STRING;
  mssqldatabasestatus : STRING;
  severity: default = CRITICAL;
};
END

CacheCpu resource model

**Microsoft_SQL_Server_Low_BufferCacheHitRatio**

TEC_CLASS :
Microsoft_SQL_Server_Low_BufferCacheHitRatio ISA MSSQL_Server
DEFINES {
  mssqlbuffercachehitratio : REAL;
  lowerbound : REAL;
  severity: default = WARNING;
};
END

**Microsoft_SQL_Server_High_LazyWritesPerSec**

TEC_CLASS :
Microsoft_SQL_Server_High_LazyWritesPerSec ISA MSSQL_Server
DEFINES {
  mssqllazywritespersec : REAL;
  upperbound : REAL;
  severity: default = WARNING;
};
END

**Microsoft_SQL_Server_High_StolenPageCountGrowth**

TEC_CLASS :
Microsoft_SQL_Server_High_StolenPageCountGrowth ISA MSSQL_Server
DEFINES {
  mssqlstolenpagecount : REAL;
upperbound : REAL;
mssqlstolenpagecountgrowth : REAL;
severity: default = WARNING;
};
END

Microsoft_SQL_Server_High_PageReadsPerSec
TEC_CLASS :
Microsoft_SQL_Server_High_PageReadsPerSec ISA MSSQL_Server
DEFINES {
  mssqlpagereadspersec : REAL;
  upperbound : REAL;
  severity: default = WARNING;
};
END

Microsoft_SQL_Server_High_ReadAheadPagesPerSec
TEC_CLASS :
Microsoft_SQL_Server_High_ReadAheadPagesPerSec ISA MSSQL_Server
DEFINES {
  mssqlreadaheadpagespersec : REAL;
  upperbound : REAL;
  severity: default = WARNING;
};
END

Microsoft_SQL_Server_Low_BufferExtMemCacheHitRatio
TEC_CLASS :
Microsoft_SQL_Server_Low_BufferExtMemCacheHitRatio ISA MSSQL_Server
DEFINES {
  mssqlbufferextmemcachehitratio : REAL;
  lowerbound : REAL;
  severity: default = WARNING;
};
END

Microsoft_SQL_Server_High_PageWritesPerSec
TEC_CLASS :
Microsoft_SQL_Server_High_PageWritesPerSec ISA MSSQL_Server
DEFINES {
  mssqlpagewritespersec : REAL;
  upperbound : REAL;
  severity: default = WARNING;
};
END

Microsoft_SQL_Server_High_CpuPercentBusy
TEC_CLASS :
Microsoft_SQL_Server_High_CpuPercentBusy ISA MSSQL_Server
DEFINES {
  upperbound : REAL;
  mssqlcpupercentbusy : REAL;
  severity: default = MINOR;
};
END

Microsoft_SQL_Server_Low_FreeBuffers
TEC_CLASS :
Microsoft_SQL_Server_Low_FreeBuffers ISA MSSQL_Server
DEFINES {
  mssqlfreebuffers : REAL;
lowerbound : REAL;
severity: default = WARNING;
);
END

Microsoft_SQL_Server_High_CheckpointWritesPerSec

TEC_CLASS :
Microsoft_SQL_Server_High_CheckpointWritesPerSec ISA MSSQL_Server
DEFINES {
mssqlcheckpointwritespersec : REAL;
upperbound : REAL;
severity: default = WARNING;
}
END

Errors resource model

Microsoft_SQL_Server_Connections_High

TEC_CLASS :
Microsoft_SQL_Server_Connections_High ISA MSSQL_Server
DEFINES {
mssqlerror : STRING;
mssqlerrorsev : STRING;
mssqlerrorstate : STRING;
mssqlerrorpid : STRING;
mssqlerrordatetime : STRING;
mssqlerrordesc : STRING;
severity: default = CRITICAL;
}
END

Microsoft_SQL_Server_Full_Database

TEC_CLASS :
Microsoft_SQL_Server_Full_Database ISA MSSQL_Server
DEFINES {
mssqlerror : STRING;
mssqlerrorsev : STRING;
mssqlerrorstate : STRING;
mssqlerrorpid : STRING;
mssqlerrordatetime : STRING;
mssqlerrordesc : STRING;
severity: default = CRITICAL;
}
END

Microsoft_SQL_Server_Disk_IO_Error

TEC_CLASS :
Microsoft_SQL_Server_Disk_IO_Error ISA MSSQL_Server
DEFINES {
mssqlerror : STRING;
mssqlerrorsev : STRING;
mssqlerrorstate : STRING;
mssqlerrorpid : STRING;
mssqlerrordatetime : STRING;
mssqlerrordesc : STRING;
severity: default = CRITICAL;
}
END

Microsoft_SQL_Server_Errors_in_Log

TEC_CLASS :
Microsoft_SQL_Server_Errors_in_Log ISA MSSQL_Server
DEFINES {
mssqlerror : STRING;
210 IBM Tivoli Monitoring for Databases: Microsoft SQL Server: Reference Guide
mssqlerrorsev : STRING;
mssqlerrorstate : STRING;
mssqlerrorpid : STRING;
mssqlerrordatetime : STRING;
mssqlerrordesc : STRING;
severity: default = CRITICAL;
);
END

Microsoft_SQL_Server_Transaction_Log_Full

TEC_CLASS :
Microsoft_SQL_Server_Transaction_Log_Full ISA MSSQL_Server
DEFINES {
  mssqlerror : STRING;
mssqlerrorsev : STRING;
mssqlerrorstate : STRING;
mssqlerrorpid : STRING;
mssqlerrordatetime : STRING;
mssqlerrordesc : STRING;
  severity: default = CRITICAL;
};
END

Microsoft_SQL_Server_Connection_Failed

TEC_CLASS :
Microsoft_SQL_Server_Connection_Failed ISA MSSQL_Server
DEFINES {
  mssqlerror : STRING;
mssqlerrorsev : STRING;
mssqlerrorstate : STRING;
mssqlerrorpid : STRING;
mssqlerrordatetime : STRING;
mssqlerrordesc : STRING;
  severity: default = CRITICAL;
};
END

Microsoft_SQL_Server_Deadlock_Alert

TEC_CLASS :
Microsoft_SQL_Server_Deadlock_Alert ISA MSSQL_Server
DEFINES {
  mssqlerror : STRING;
mssqlerrorsev : STRING;
mssqlerrorstate : STRING;
mssqlerrorpid : STRING;
mssqlerrordatetime : STRING;
mssqlerrordesc : STRING;
  severity: default = CRITICAL;
};
END

Microsoft_SQL_Server_ERRORLOG_Event

TEC_CLASS :
Microsoft_SQL_Server_ERRORLOG_Event ISA MSSQL_Server
DEFINES {
  mssqlerror : STRING;
mssqlerrorsev : STRING;
mssqlerrorstate : STRING;
mssqlerrorpid : STRING;
mssqlerrordatetime : STRING;
mssqlerrordesc : STRING;
  severity: default = CRITICAL;
};
END
Microsoft_SQL_Server_Allocate_Space_Failed

TEC_CLASS:
Microsoft_SQL_Server_Allocate_Space_Failed ISA MSSQL_Server
DEFINES {
    mssqlerror : STRING;
    mssqlerrorsev : STRING;
    mssqlerrorstate : STRING;
    mssqlerrorpid : STRING;
    mssqlerrordatetime : STRING;
    mssqlerrordesc : STRING;
    severity: default = CRITICAL;
}
END

Jobs resource model

Microsoft_SQL_Server_Failed_Job

TEC_CLASS:
Microsoft_SQL_Server_Failed_Job ISA MSSQL_Server
DEFINES {
    mssqljobname : STRING;
    mssqljobstatus : STRING;
    mssqljobduration : REAL;
    severity: default = CRITICAL;
}
END

Microsoft_SQL_Server_Cancelled_Job

TEC_CLASS:
Microsoft_SQL_Server_Cancelled_Job ISA MSSQL_Server
DEFINES {
    mssqljobname : STRING;
    mssqljobstatus : STRING;
    mssqljobduration : REAL;
    severity: default = CRITICAL;
}
END

Microsoft_SQL_Server_Job_Not_Run

TEC_CLASS:
Microsoft_SQL_Server_Job_Not_Run ISA MSSQL_Server
DEFINES {
    mssqljobname : STRING;
    mssqljobstatus : STRING;
    severity: default = WARNING;
}
END

Microsoft_SQL_Server_Job_Duration_Threshold

TEC_CLASS:
Microsoft_SQL_Server_Job_Duration_Threshold ISA MSSQL_Server
DEFINES {
    mssqljobname : STRING;
    mssqljobduration : REAL;
    jobstatnum : REAL;
    mssqljobdurationthreshold : REAL;
    severity: default = MINOR;
}
END
Locks resource model

Microsoft_SQL_Server_High_LockWaitTimePerSec
TEC_CLASS :
  Microsoft_SQL_Server_High_LockWaitTimePerSec ISA MSSQL_Server
  DEFINES {
    mssqlresourcetype : STRING;
    mssqllockwaittime : REAL;
    upperbound : REAL;
    severity: default = WARNING;
  }
END

Microsoft_SQL_Server_High_NumberDeadlocksPerSec
TEC_CLASS :
  Microsoft_SQL_Server_High_NumberDeadlocksPerSec ISA MSSQL_Server
  DEFINES {
    mssqlresourcetype : STRING;
    mssqlnumberdeadlockspersec : REAL;
    upperbound : REAL;
    severity: default = CRITICAL;
  }
END

Replication resource model

Microsoft_SQL_Server_High_DownloadedChangesPerSec
TEC_CLASS :
  Microsoft_SQL_Server_High_DownloadedChangesPerSec ISA MSSQL_Server
  DEFINES {
    mssqldownloadedchangespersec : REAL;
    upperbound : REAL;
    severity: default = WARNING;
  }
END

Microsoft_SQL_Server_High_LogreaderDeliveryLatency
TEC_CLASS :
  Microsoft_SQL_Server_High_LogreaderDeliveryLatency ISA MSSQL_Server
  DEFINES {
    mssqllogreaderdeliverylatency : REAL;
    upperbound : REAL;
    severity: default = WARNING;
  }
END

Microsoft_SQL_Server_Low_LogreaderDeliveredCmdsPerSec
TEC_CLASS :
  Microsoft_SQL_Server_Low_LogreaderDeliveredCmdsPerSec ISA MSSQL_Server
  DEFINES {
    mssqllogreaderdeliveredcmdspersec : REAL;
    lowerbound : REAL;
    severity: default = WARNING;
  }
END

Microsoft_SQL_Server_High_DistDeliveryLatency
TEC_CLASS :
  Microsoft_SQL_Server_High_DistDeliveryLatency ISA MSSQL_Server
  DEFINES {
    mssqldistdeliverylatency : REAL;
  
Appendix B. Tivoli Enterprise Console classes   213
upperbound : REAL;
severity: default = WARNING;
};
END

Microsoft_SQL_Server_Low_ReplTransactionRate
TEC_CLASS :
Microsoft_SQL_Server_Low_ReplTransactionRate ISA MSSQL_Server
DEFINES {
  mssqldatabasename : STRING;
mssqlrepltransactionrate : REAL;
lowerbound : REAL;
  severity: default = WARNING;
};
END

Microsoft_SQL_Server_High_MergeConflictsPerSec
TEC_CLASS :
Microsoft_SQL_Server_High_MergeConflictsPerSec ISA MSSQL_Server
DEFINES {
  mssqlmergeconflictspersec : REAL;
  upperbound : REAL;
  severity: default = WARNING;
};
END

Microsoft_SQL_Server_Low_LogreaderDeliveredTransPerSec
TEC_CLASS :
Microsoft_SQL_Server_Low_LogreaderDeliveredTransPerSec ISA MSSQL_Server
DEFINES {
  mssqllogreaderdeliveredtranspersec : REAL;
lowerbound : REAL;
  severity: default = WARNING;
};
END

Microsoft_SQL_Server_Low_DistDeliveredTransPerSec
TEC_CLASS :
Microsoft_SQL_Server_Low_DistDeliveredTransPerSec ISA MSSQL_Server
DEFINES {
  mssqldistdeliveredtranspersec : REAL;
lowerbound : REAL;
  severity: default = WARNING;
};
END

Microsoft_SQL_Server_Low_DistDeliveredCmdsPerSec
TEC_CLASS :
Microsoft_SQL_Server_Low_DistDeliveredCmdsPerSec ISA MSSQL_Server
DEFINES {
  mssqldistdeliveredcmdspersec : REAL;
lowerbound : REAL;
  severity: default = WARNING;
};
END

Microsoft_SQL_Server_High_UploadedChangesPerSec
TEC_CLASS :
Microsoft_SQL_Server_High_UploadedChangesPerSec ISA MSSQL_Server
DEFINES {
  mssqluploadedchangespersec : REAL;
};
upperbound : REAL;
severity: default = WARNING;
);
END

**Microsoft_SQL_Server_High_ReplPendingTransactions**

TEC_CLASS :
Microsoft_SQL_Server_High_ReplPendingTransactions ISA MSSQL_Server
DEFINES {
  mssqldatabasename : STRING;
mssqlreplpendingtransactions : REAL;
  upperbound : REAL;
  severity: default = WARNING;
};
END

**Replication Jobs resource model**

**Microsoft_SQL_Server_WaitingForThread_Replication_Job**

TEC_CLASS :
Microsoft_SQL_Server_WaitingForThread_Replication_Job ISA MSSQL_Server
DEFINES {
  mssqlreplicationjobname : STRING;
  severity: default = MINOR;
};
END

**Microsoft_SQL_Server_Idle_Replication_Job**

TEC_CLASS :
Microsoft_SQL_Server_Idle_Replication_Job ISA MSSQL_Server
DEFINES {
  mssqlreplicationjobname : STRING;
  severity: default = MINOR;
};
END

**Microsoft_SQL_Server_Failed_Replication_Job**

TEC_CLASS :
Microsoft_SQL_Server_Failed_Replication_Job ISA MSSQL_Server
DEFINES {
  mssqlreplicationjobname : STRING;
mssqlfailedreplicationjoberrorcode : STRING;
  severity: default = CRITICAL;
};
END

**Microsoft_SQL_Server_Cancelled_Replication_Job**

TEC_CLASS :
Microsoft_SQL_Server_Cancelled_Replication_Job ISA MSSQL_Server
DEFINES {
  mssqlreplicationjobname : STRING;
  severity: default = CRITICAL;
};
END

**Microsoft_SQL_Server_Suspended_Replication_Job**

TEC_CLASS :
Microsoft_SQL_Server_Suspended_Replication_Job ISA MSSQL_Server
DEFINES {
  mssqlreplicationjobname : STRING;
  severity: default = CRITICAL;
};
END
Space Usage resource model

Microsoft_SQL_Server_High_LogSpacePercentUsed

TEC_CLASS : Microsoft_SQL_Server_High_LogSpacePercentUsed ISA MSSQL_Server
DEFINES {
  mssqllogfilename : STRING;
  mssqldatabasename : STRING;
  mssqllogmaximumgrowthsize : STRING;
  mssqllogspacepercentusedthreshold : REAL;
  mssql_log_space_percent_used_actual : REAL;
  mssql_log_file_size : REAL;
  severity: default = CRITICAL;
};
END

Microsoft_SQL_Server_High_DatabaseSpacePercentUsed

TEC_CLASS : Microsoft_SQL_Server_High_DatabaseSpacePercentUsed ISA MSSQL_Server
DEFINES {
  mssqldatabasename : STRING;
  mssqldatabasefilenames : STRING;
  mssqldatabasemaxgrowthsize : STRING;
  mssqldatabasespacepercentusedthreshold : REAL;
  mssql_database_size : REAL;
  mssql_database_space_percent_used_actual : REAL;
  severity: default = CRITICAL;
};
END

Microsoft_SQL_Server_High_FilegroupPercentUsed

TEC_CLASS : Microsoft_SQL_Server_High_FilegroupPercentUsed ISA MSSQL_Server
DEFINES {
  mssqldatabasename : STRING;
  mssqlfilegroupname : STRING;
  mssqlfilegroupmaximumgrowthsize : STRING;
  mssqlfilegroupspacepercentusedthreshold : REAL;
  mssql_filegroupid : REAL;
  mssql_filegroup_space_percent_used_actual : REAL;
  mssql_filegroup_size : REAL;
  severity: default = CRITICAL;
};
END

Microsoft_SQL_Server_High_DatabaseGrowth

TEC_CLASS : Microsoft_SQL_Server_High_DatabaseGrowth ISA MSSQL_Server
DEFINES {
  mssqldatabasename : STRING;
  mssqlcurrentdatabasesize : REAL;
  mssqllastdatabasesize : REAL;
  mssql_databasesizegrowthpercentthreshold : REAL;
  mssql_database_growth_percent_actual : REAL;

  severity: default = CRITICAL;
};
END
Users Transactions resource model

Microsoft_SQL_Server_High_LoginsPerSec
TEC_CLASS :
Microsoft_SQL_Server_High_LoginsPerSec ISA MSSQL_Server
DEFINES {
mssqlloginspersec : REAL;
upperbound : REAL;
severity: default = CRITICAL;
};
END

Microsoft_SQL_Server_High_ActiveTransactions
TEC_CLASS :
Microsoft_SQL_Server_High_ActiveTransactions ISA MSSQL_Server
DEFINES {
mssqldatabasename : STRING;
mssqlactivetransactions : REAL;
upperbound : REAL;
severity: default = CRITICAL;
};
END

Microsoft_SQL_Server_High_LogoutsPerSec
TEC_CLASS :
Microsoft_SQL_Server_High_LogoutsPerSec ISA MSSQL_Server
DEFINES {
mssqllogoutspersec : REAL;
upperbound : REAL;
severity: default = CRITICAL;
};
END

Microsoft_SQL_Server_High_BlockedProcesses
TEC_CLASS :
Microsoft_SQL_Server_High_BlockedProcesses ISA MSSQL_Server
DEFINES {
mssqlblockedprocesses : REAL;
upperbound : REAL;
severity: default = WARNING;
};
END

Microsoft_SQL_Server_High_TransactionsPerSec
TEC_CLASS :
Microsoft_SQL_Server_High_TransactionsPerSec ISA MSSQL_Server
DEFINES {
mssqldatabasename : STRING;
mssqltransactionspersec : REAL;
upperbound : REAL;
severity: default = CRITICAL;
};
END

Microsoft_SQL_Server_High_PctUserConnectionsUsed
TEC_CLASS :
Microsoft_SQL_Server_High_PctUserConnectionsUsed ISA MSSQL_Server
DEFINES {
mssqlpctuserconnectionsused : REAL;
upperbound : REAL;
mssqluserconnections : REAL;
severity: default = CRITICAL;
};
END
Microsoft_SQL_Server_High_LongRunningProcesses

TEC_CLASS:
Microsoft_SQL_Server_High_LongRunningProcesses ISA MSSQL_Server
DEFINES {
  mssqllongrunningprocesscount : REAL;
  upperbound : REAL;
  severity: default = WARNING;
};
END
Appendix C. Resource model CIM class quick reference

The following table lists the IBM Tivoli Monitoring for Databases: Microsoft SQL Server resource models and the CIM classes they reference. For complete details on the IBM Tivoli Monitoring for Databases: Microsoft SQL Server resource models, see Chapter 2, “Resource models”, on page 5.

Table 11. Resource models and their referenced CIM classes

<table>
<thead>
<tr>
<th>Resource Model</th>
<th>CIM Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>Win32_Service, ITMMSSQLprovClass</td>
</tr>
<tr>
<td>Cache/CPU</td>
<td>Win32_Service, ITMMSSQLprovClass</td>
</tr>
<tr>
<td>Errors</td>
<td>ITMMSSQLprovClass, root\default:StdRegProv</td>
</tr>
<tr>
<td>Jobs</td>
<td>Win32_Service, ITMMSSQLprovClass</td>
</tr>
<tr>
<td>Locks</td>
<td>Win32_Service, ITMMSSQLprovClass</td>
</tr>
<tr>
<td>Replication</td>
<td>Win32_Service, ITMMSSQLprovClass</td>
</tr>
<tr>
<td>Replication Jobs</td>
<td>Win32_Service, ITMMSSQLprovClass</td>
</tr>
<tr>
<td>Space Usage</td>
<td>Win32_Service, Win32_LogicalDisk, ITMMSSQLprovClass</td>
</tr>
<tr>
<td>Users/Transactions</td>
<td>Win32_Service, ITMMSSQLprovClass</td>
</tr>
</tbody>
</table>
Appendix D. Creating custom resource models

The IBM Tivoli Monitoring Workbench is a programming tool for creating, modifying, debugging, and packaging resource models for use with IBM Tivoli Monitoring products. Samples of Best Practice Resource Models have also been provided for you to use within the IBM Tivoli Monitoring Workbench. These resource models are intended to be used as working examples for creating new resource models. Customers need to have a current Tivoli Maintenance & Support Contract to receive assistance with problems and issues related to the operation of the IBM Tivoli Monitoring Workbench.

Note: The Tivoli Maintenance and Support Contract covers assistance with problems relating to the operation of the IBM Tivoli Monitoring Workbench, but does not cover assistance for new or modified resource models other than the ones that are included in the IBM Tivoli Monitoring Workbench.

Overview

Resource models for IBM Tivoli Monitoring for Databases: Microsoft SQL Server depend upon a data provider, which serves as a SQL engine. This engine, implemented in the code, accepts well-formed SQL commands and returns the appropriate result set or status message.

Note

Result sets returned from the data provider must be programatically manipulated by the resource model author in the JavaScript code in order for the resource model to work properly.

ITMMSQLprovClass

Typically, when viewing a Windows Management Instrumentation class from the IBM Tivoli Monitoring Workbench’s Dynamic Model window, you see a list of the available, resource-specific properties for the Windows Management Instrumentation class. These properties can be selected for inclusion in a resource model. For example, the Windows Management Instrumentation class named Win32_Services displays properties specific to a Windows Service, such as Name, State, StartMode, ProcessId, and ServiceType.

There are no properties specific to a Microsoft SQL Server or database that are explicitly defined and exposed with the data provider. However, two general properties are exposed. One of these properties is the focus of any resource model built upon the data provider, ITMMSQLprovClass.

The two properties exposed are sOutArg and DataLen. The sOutArg property is a "string" type and contains the SQL result set or status message returned by the data provider. The DataLen property is a "numeric" type and contains the string length value of the sOutArg property.

To build resource models based on the ITMMSQLprovClass data provider, you should be familiar with the following items:

- The SQL query to execute on a monitored Microsoft SQL Server
The structure of the result set returned from the query
The logic you need to apply to the result set

Note: Regarding sOutArg, the SQL result set returned by the data provider uses the following delimiters:
- A newline character terminates a row in a result set
- A semicolon character (;) separates each column in a row where the last column value is not terminated by a semicolon
- Empty result sets return the string "Command complete."
- An internal error for the data provider returns the string, "DPERROR: "<...>
  where <...> is case-specific text

The following procedure explains the steps you can follow to create your own customized resource model.

Creating custom resource models using the IBM Tivoli Monitoring Workbench

Objective
To create customized resource models in the Workbench using the ITMMSSQLprovClass CIM class.

Background information
The IBM Tivoli Monitoring Workbench is a programming tool for creating, modifying, debugging, and packaging resource models for use with IBM Tivoli Monitoring products. Samples of the Best Practice Resource Models have been provided for you to use within the IBM Tivoli Monitoring Workbench. The sample resource models are intended to be used as working examples for creating new resource models.

The Resource Model Wizard guides you through the process of creating resource models using the IBM Tivoli Monitoring for Databases: Microsoft SQL Server CIM class. Refer to the IBM Tivoli Monitoring Workbench documentation for more information on how to create resource models.

Required authorization role
None, however, you must have access to the IBM Tivoli Monitoring Workbench

Before you begin
Before you begin, you must perform the following steps:
1. Install and configure Windows Management Instrumentation.
2. Install and configure IBM Tivoli Monitoring Workbench.
3. Load the IBM Tivoli Monitoring for Databases: Microsoft SQL Server CIM class, ITMMSSQLprovClass.
   Additional Information: Load the class by running the mofcomp command from the Workbench/w32–ix86 directory in the IBM Tivoli Monitoring for Databases: Microsoft SQL Server installation CD. For more information about the mofcomp command, refer to the Windows Management Instrumentation documentation.
After you finish
Refer to IBM Tivoli Monitoring Workbench documentation for instructions on how to build your new resource model. Refer to the IBM Tivoli Monitoring documentation for instructions on how to deploy the resource model. Resource models created to use the ITMMSQLprovClass CIM class need to be distributed to the endpoint.

Procedure
1. Open the IBM Tivoli Monitoring Workbench.
2. Select File to display the File drop-down menu.
3. Select New from the File drop-down menu to display the New window.
5. Click OK to display the New Resource Model Workspace window.
7. Click OK.
8. Select the CIM/WMI data source type.
9. Select all of the available operating systems on which you want the resource model to run.
10. Click Next.
11. Type ROOT\CIMV2 in the Connect to namespace field.
    Additional Information: ROOT\CIMV2 is the destination that stores the ITMMSQLprovClass CIM class.
12. Optional: Type your password.
13. Click OK to display the Dynamic Model window.
14. In the left CIM class tree, click on ITMMSQLprovClass to highlight the CIM class.
    Additional Information: The Selected Class field displays the CIM class for use in your custom resource models. The IBM Tivoli Monitoring for Databases: Microsoft SQL Server CIM class is ITMMSQLprovClass.
15. Type an alias name for the CIM class in the Use Alias text box.
    Additional Information: You can type any string value that you prefer for use in the resource model JavaScript code. In the example below, "basic" is the designated alias for the ITMMSQLprovClass CIM class.
16. Select the sOutArg and DataLen CIM class properties from the Class Properties group box.
17. Define and associate the execMethod and execCommand parameters to the provider using JavaScript such as the following:

```javascript
var method = "ExecSQL";
Svc.DefineStrParameter("execMethod", method);
Svc.AssociateParameterToClass("execMethod", "basic");

Svc.DefineStrParameter("execCommand", SQL_LONGRUNNING_PROCESS);
Svc.AssociateParameterToClass("execCommand", "basic"); . . .
```

where "basic" represents the ITMMSQLprovClass CIM class.

Additional Information: To execute an SQL command using the data provider, you must specify two parameters to pass to the provider. The execMethod parameter instructs the provider to use a certain method. The value should be "ExecSQL." The execCommand parameter is the SQL command the provider should execute.

18. Call the provider to collect data using JavaScript such as the following:
Svc.CollectClassData("basic");

where "basic" represents the ITMMSSQLprovClass CIM class.

Additional Information: This method call is present because the provider was defined as "Collect On Demand." For more information regarding defining CIM classes, refer to the IBM Tivoli Monitoring Workbench Reference Manual.

19. Retrieve the result set from the provider using JavaScript such as the following:

```
sOutArg = Svc.GetStrProperty("basic", 0, "sOutArg");
```

where "basic" represents the ITMMSSQLprovClass CIM class.

Additional Information: The data provider returns one complete result set. Access the result set by retrieving the sOutArg property defined in the dynamic model.

20. Programatically manipulate the result set as explained in the JavaScript coding example below.

### JavaScript coding example

The following JavaScript code example details the behavior, technical issues, and logic specific to the use of the resource model data provider, the Windows Management Instrumentation class named ITMMSSQLprovClass.

The SQL statements included are not the focus of this example. Instead, review the structure and nature of the result set to understand the modifications that must be made to the result set.

**Note:** Items below in **bold** must be programatically manipulated to match the example. Comments in *italics* provide additional information.

```
// GLOBAL constant declarations
var SQL_MAXCONN = "select @@max_connections";
var SQL_BLOCKEDPROCESS = "select count(*) from master.dbo.sysprocesses where blocked = 0";
var SQL_LONGRUNNING_PROCESS = "select spid, datediff(mi,last_batch,getdate()) duration," +
" substring(hostname,1,17) as hostname, blocked," +
" substring(status,1,18) as status, cmd," +
" substring(program_name,1,18) as program_name," +
" substring(loginame,1,18) as login_name," +
" substring(nt_username,1,18) as nt_user" +
" from master..sysprocesses" +
" where (status !='sleeping' or cmd !='AWAITING COMMAND' or
" or blocked !=0) and spid > 8 and spid != @@spid";
var EMPTY_RESULT_SET = "Command complete.";
var TRACE_ERROR= 0;
var TRACE_INFO= 2;
var TRACE_DEBUG= 3;
var TRACE_SOURCE = "RM TRACE: ";

// Return Codes for various error conditions
var RC_INVALID_RESOURCE_TYPE = 801;
var RC_RESOURCE_UNAVAILABLE = 405; //Values 401 - 600 cause IBM Tivoli Monitoring to retry the resource model every three minutes.
var RC_INVALID_METRIC = 406;
var RC_INVALID_VERSION = 802;
```
var RC_COLLECTION_ERROR = 407;

// end GLOBAL constants

function Init(Svc)
{
    Svc.Trace(TRACE_INFO, TRACE_SOURCE + "Start Init");
    ...

    // Initialize the resource model to use the data provider, ITMMSQLprovClass, using "basic" as the class alias name:
    var method = "ExecSQL";
    Svc.DefineStrParameter("execMethod", method);
    Svc.AssociateParameterToClass("execMethod", "basic");
    Svc.DefineStrParameter("execCommand", SQL_LONGRUNNING_PROCESS);
    Svc.AssociateParameterToClass("execCommand", "basic");
    ...
    Svc.Trace(TRACE_INFO, TRACE_SOURCE + "End Init");
    return (0);
}

function VisitTree(Svc)
{
    var sOutArg;
    var DataLen;
    var pattern = new RegExp("^DPERROR");
    var rows = new Array();
    // Used to store result set rows from executing a SQL query.
    var columns = new Array();
    // Used to store a column of a single row from a result set.
    var row_wo_nulls;
    // Used to store a modified result set row from executing a SQL query, where NULL columns have been filled in with a dash.
    Svc.Trace(TRACE_INFO, TRACE_SOURCE + "Start VisitTree");
    ...
    Svc.CollectClassData("basic");
    // This method call is present because Collect On Demand was selected.
    numOfInstances = Svc.GetNumOfInst("basic");
    Svc.Trace(TRACE_DEBUG, TRACE_SOURCE + "Long Running Processes numOfInstances = " + numOfInstances);
    sOutArg = Svc.GetStrProperty("basic", 0, "sOutArg");
    Svc.Trace(TRACE_DEBUG, TRACE_SOURCE + " SQL result set from basic Provider - " + sOutArg);
    DataLen = Svc.GetNumProperty("basic", 0, "DataLen");
    Svc.Trace(TRACE_DEBUG, TRACE_SOURCE + "DataLen = " + DataLen);
    if (numOfInstances == 0) {
        Svc.Trace(TRACE_ERROR, TRACE_SOURCE + "\n\tERROR: No results were received for Long Running Process metrics from Microsoft SQL Server named " + glb_mss_serverName);
        return (RC_COLLECTION_ERROR);
    }
    if (pattern.test(sOutArg)) {
        
    }
Svc.Trace(TRACE_ERROR, TRACE_SOURCE + "\nERROR: basic Provider returned error: " + sOutArg);
return (RC_COLLECTION_ERROR);
}

// "EMPTY_RESULT_SET" displays if there are no long-running processes.

// Get all running processes from sOutArg string, where sOutArg contains a
// result set of N rows with 9 columns.

// Columns - rows delimited by 'newline,' columns delimited by ';

// The row layout is: spid; duration; hostname; blocked; status; cmd;
// program_name; login_name; nt_user

// Sample Row values: 9;155;;0;background;TASKMANAGER;;sa;
// (two consecutive semicolons is a NULL valued column)

rows = sOutArg.split(/\n/);

if ( ( rows[0] + "" ) != EMPTY_RESULT_SET )
{
    for ( var k=0; k< rows.length ; k++ )
    {
        var j = 0;

        // Replace NULL columns that appear as two consecutive semicolon characters
        // by globally replacing any two consecutive semicolons with a "semicolon
        // dash semicolon" sequence.
        var re = /;;/g;
        row_wo_nulls = rows[k].replace( re, ";-;" );

        // You must make the following modification, or the *.split(/;/) results in the skewing of column values:
        columns = row_wo_nulls.split(/;/);

        // Now the "row" in column array looks like the following:
        // 9 155 – 0 background TASKMANAGER – sa

        // Check the last column in row for a NULL column because two
        // consective semicolons do not flag this.

        // Check if columns[8]( the 9th column ) was set according to
        // the prior split method call:
        if ( columns.length != 9 )
            columns[8] = "-";

        // Now the "row" in column array looks like the following:
        // 9 155 – 0 background TASKMANAGER – sa –

        // The three, formerly-NULL, columns are now set to dash (– )

        for (j=0; j< columns.length; j++)
        {
            Svc.Trace(TRACE_DEBUG, TRACE_SOURCE + "columns[" + j + "] = " + columns[j]);
        }

        curSpid = parseInt( columns[0] );
        curDuration = parseInt( columns[1] );

        ... Resource model-specific processing displays here
    } // end of 'for rows' loop
Remove the prior SQL_LONG_RUNNING_PROCESS parameter value. The resource model must call the Svc.RemoveStrParameter() method.

Svc.RemoveStrParameter("execCommand", 0);

Add in the new SQL statement to execute as a new parameter value:

Svc.AddStrParameter("execCommand", SQL_BLOCKEDPROCESS);

Svc.CollectClassData("basic");

This method call is present because Collect On Demand was selected.

numOfInstances = Svc.GetNumOfInst("basic");

Svc.Trace(TRACE_DEBUG, TRACE_SOURCE + "Blocked Processes numOfInstances = " + numOfInstances);

sOutArg = Svc.GetStrProperty("basic", 0, "sOutArg");
Svc.Trace(TRACE_DEBUG, TRACE_SOURCE + " SQL result set from basic Provider = " + sOutArg);

DataLen = Svc.GetNumProperty("basic", 0, "DataLen");
Svc.Trace(TRACE_DEBUG, TRACE_SOURCE + "DataLen = " + DataLen);

if ( numOfInstances == 0 ) {
    Svc.Trace(TRACE_ERROR, TRACE_SOURCE + "\n\nERROR: No results were received for Blocked Processes from Microsoft SQL Server named " + glb_mss_serverName);
    return (RC_COLLECTION_ERROR);
}

if ( pattern.test(sOutArg) ) {
    Svc.Trace(TRACE_ERROR, TRACE_SOURCE + "\n\nERROR: basic Provider returned error: " + sOutArg);
    return (RC_COLLECTION_ERROR);
}

if ( ( sOutArg + "" ) == EMPTY_RESULT_SET ) {
    Svc.Trace(TRACE_ERROR, TRACE_SOURCE + "\n\nERROR: basic Provider returned empty SQL result set: " + sOutArg);
    return (RC_COLLECTION_ERROR);
}

Get BlockedProcesses from sOutArg string, where sOutArg contains a result set of one row with one column

Rows delimited by 'newline,' columns delimited by ';' 

The row layout is ... numberBlockedProcesses

Sample Row values: 15

columns = sOutArg.split(/;/);
curBlockedProcesses = columns[0];

... resource model-specific processing displays here
Appendix E. 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.

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 appear.

### Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM, the IBM logo, Tivoli, the Tivoli logo, and Tivoli Enterprise Console are trademarks or registered trademarks of International Business Machines Corporation or Tivoli Systems Inc. in the United States, other countries, or both.

Lotus and Lotus Notes are trademarks of International Business Machines Corporation and Lotus Development Corporation in the United States, other countries, or both.

Microsoft, Windows NT and Windows 2000 are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

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.

Other company, product, and service names may be trademarks or service marks of others.
Index

Special characters
.baroc file 207

A
accessibility vii
activity log file 192, 193, 197
Adapter configuration file for Tivoli Enterprise Console 143
allocate space error 41
ANSI Null Default 174, 178
ANSI Nulls 174, 178
ANSI Warnings 174, 178
application_class 4
application_label
resource model attribute 4
application_oid
resource model attribute 4
application_version
resource model attribute 4
attribute
application_class 4
application_label 4
application_oid 4
application_version 4
ENDPOINT_OID 4
HOST 4
INTERP 4
MSSQL_HOME 4
MSSQL_SID 4
authentication
database access 161
server access 161
Authorization role
about 135
Autoclose 175, 179
Autoshrink 175, 179
Availability
about 9

B
backup
database 150, 191
options 154
transaction log 191, 196
bash shell 1, 2
BCP (Bulk Copy) command 176, 181
books
accessing online vii
feedback on v
online v
ordering v
Bourne shell 2
buffer
free 26
buffer cache 18
buffer extended memory cache hit ratio 22
buffers
number of 192, 193, 197
business rules
creating 207
C
C shell 2
cache buffer 26
Cache/CPU
about 18
cancelled job 58
category
for resource model 5
central processing unit 25
checkpoint
logging 24
occurring 23
truncate log 177, 181
checkpoint write 23
checkpoint write per second 23
CIM class
resource model 219
Class Definition Statement (CDS) file 143
clearing event
about 6
CLI command
running 1
Close on commit 175
command
authorization roles 2
BCP (Bulk Copy) 176, 181
DBCC CHECKDB 147
KILL 125
running 1
running on UNIX 2
running on Windows 2000 2
running on Windows NT 2
SELECT INTO 176, 181
setting up environment 3
setting up environment to run 1
sp_dboption 182
sp_spaceused 107
syntax 1
tdpsqlc 194, 199
wdirmlseng 7
wruntask 135
Concat null yields null 175, 179
canonical log file 192, 193, 197
concatenation 175, 179
closing 175, 179
cycle time
about 5

data
logging 7
database
availability 9
backup 147, 150, 191, 194, 199
closing 175, 179
configuration 173
dump 150, 194, 199
filegroup 107
full database error 50
lock 67
login properties 161
merge replication publications 176, 180
offline 176, 180
read-only 176, 180
replication publications 176, 180
setting options 173
shrink automatically 175, 179
single-user mode 177, 181
space percent used 107
space usage 107
status 9
subscribed for publication 177, 181
suspect 13
validate integrity 147
database growth 107
DBCC CHECKDB command 147
dbo use only option 175, 180
DDL command
using 159
deadlock 69
deadlock alert error 45
default cycle time 5
Default Duration Threshold 63
directory names, notation ix
DiscoverMSSQLServers task 144
disk i/o error 46
DISKBDUMP 151
distribution delivery latency 80
DML command
using 159
dump
database 150
DumpDatabase task 147
DumpTransactionLog task 152

233
E
endpoint
setting up the Tivoli environment on 3
starting services on 183
stop services on 187
ENDPOINT_OID
resource model attribute 4
environment variables, notation ix
error 1101 38, 41
1105 38, 41
1205 38, 45, 69
1211 38, 45
16804 38, 43
1703 38, 41
17809 38, 44
1803 38, 41
18450 38, 43
18452 38, 43
3159 38, 49
3163 38, 49
3928 38, 45
4329 38, 49
5032 38, 49
5069 38, 50
5108 38, 49
605 38, 46
823 38, 46
9001 38, 49
9002 38, 52
9006 38, 49
9007 38, 49
9008 38, 49
9009 38, 49
about 38
allocate space 41
allocate space failed 38
connection failed 38, 43
connections high 38, 44
deadlock alert 38, 45
disk i/o 46
ERRORLOG event 48
events in log 38, 49
full database 38, 50
logging 166
monitoring specific 48
transaction log full 38, 52
error numbers 38
ERRORLOG event error 48
ERRORLOG file 38
location 39
errors 60
fatal 166, 168
non-fatal 166
Errors in Log error 49
event 5
logging 166
events 5
filtering 141, 143
ExecuteSQL
task 158
extended memory 22
F
failed job 59
filegroup 107
fragmentation 171
fragmented index
rebuilding 169
free buffer 26
full database error 50
Functionality mapping
resource model to monitor 201
G
GUI data entry fields 135
H
holes about 6
HOST
resource model attribute 4
I
i/o
disk i/o error 46
I/O
incomplete operations 177, 181
I/O rate 24
IBM Tivoli Monitoring Web Health
Console
data 7
IBM Tivoli Monitoring Workbench 221
creating resource model with 222
identifier
quotes 176, 180
Ignored Jobs parameter 63
index 169
clusted 169
fragmentation 171
non-clustered 169
indication
Microsoft SQL Server Active
Transactions too high 119
Microsoft SQL Server Allocate Space
Failed 41
Microsoft SQL Server Blocked
Processes too high 121
Microsoft SQL Server Buffer Cache
Hit Ratio too low 20
Microsoft SQL Server Buffer Extended
Memory Cache Hit Ratio too low 22
Microsoft SQL Server Cancelled
Job 58
Microsoft SQL Server Cancelled
Replication Job 98
Microsoft SQL Server Checkpoint
Writes Per Second too high 23
Microsoft SQL Server Connection
Failed 43
Microsoft SQL Server Connections
High 44
Microsoft SQL Server CPU Percent
Busy too high 25
indication (continued)
Microsoft SQL Server Deadlock
Alert 45
Microsoft SQL Server Disk I/O
Error 46
Microsoft SQL Server Dist Delivered
Cmds Per Second too low 77
Microsoft SQL Server Dist Delivered
Trans Per Second too low
indication 79
Microsoft SQL Server Distribution
Delivery Latency too high 80
Microsoft SQL Server Downloaded
Changes Per Second too high 82
Microsoft SQL Server ERRORLOG
Event 48
Microsoft SQL Server Errors in
Log 49
Microsoft SQL Server Failed Job 59
Microsoft SQL Server Failed
Replication Job 99
Microsoft SQL Server Free Buffers too
low 26
Microsoft SQL Server Full
Database 50
Microsoft SQL Server High
DatabaseGrowth 108
Microsoft SQL Server High
DatabaseSpacePercentUsed 110
Microsoft SQL Server High
FilegroupPercentUsed 111
Microsoft SQL Server High
LogSpacePercentUsed 112
Microsoft SQL Server Idle Replication
Job 100
Microsoft SQL Server Job Duration
Threshold 60
Microsoft SQL Server Job Not
Run 61
Microsoft SQL Server Lazy Writes Per
Second too high 28
Microsoft SQL Server Lock Wait Time
Per Second too high 68
Microsoft SQL Server Logins Per
Second too high 122
Microsoft SQL Server Logouts Per
Second too high 123
Microsoft SQL Server Logreader
Delivered Cmds Per Second too
low 83
Microsoft SQL Server Logreader
Delivered Trans Per Second too
low 84
Microsoft SQL Server Logreader
Delivery Latency too high 85
Microsoft SQL Server Long Running
Processes too high 124
Microsoft SQL Server Merge Conflicts
Per Second too high 87
Microsoft SQL Server Number
Deadlocks Per Second too high 69
Microsoft SQL Server Page Reads Per
Second too high 29
Microsoft SQL Server Page Writes Per
Second too high 30
Microsoft SQL Server Paused
Service 10
server (continued)
status 9
stop services 187
ServerAgent 183, 185, 187, 189
Servers
discovering 144
service
paused 10
stopped 12
services
about 185, 189
starting 183
stopping 187
SetRetrieveDatabaseOptions
task 173
setup scripts 3
severity
resource model 5
shell
about 2
bash 1, 2
Bourne 2
C 2
environment 1
establishing the Tivoli environment
in 3
Korn shell 2
UNIX 2
single-user mode 177, 181
slots 207
software support viii
sp_altermessage 167
sp_configure 24
sp_dboption 182
sp_lock 67
sp_spacesused 107
sp_who 68
Space Usage
about 107
StartMSSQLServerServices
task 183
statistics
auto create 174, 179
auto update 174, 179
status
not displaying 192, 193, 196
Status
resource model attribute 4
stolen page 33
stolen page count growth 33
StopMSSQLServerServices
task 187
stopped service 12
Storage Agent
performing backup 194, 199
stored procedure
sp_lock 67
sp_who 68
subscribed 177, 181
suspect database 13
sysmessages table 168
T
Table fragmentation 171
lock 67
Tivoli Business Systems Manager 164
sending indications to 6
Tivoli Data Protection
full backup task 191
Tivoli Data Protector backup 194
Tivoli Enterprise Console
configuring 138
sending indications to 6
Tivoli Enterprise Console classes 207
Tivoli Enterprise Console NT Event Adapter
customize 141
Tivoli environment
establishing in a shell 3
establishing on an endpoint 3
establishing on UNIX 3
establishing on Windows 3
Tivoli management region
about 1
Tivoli management region server
about 1
Tivoli Software Information Center vii
Tivoli Storage Manager
Node 192, 193, 196
options file 192, 193, 197
password 192, 193, 197
torn page detection 177, 181
trace flag 2902 24
Transact-SQL 158
transaction 128
  delivered 79
transaction log
  backup 191, 196
  truncate 177, 181
TRANSACTION LOG
  backup 152
  dump 152
  truncate 152
transaction log full error 52
transaction per second 128
transactions
  active 119
triggers
  recursive 176, 181
truncate log after backup 191, 193, 196
truncate log on checkpoint 177, 181
TSM Node 192, 193, 196
TSM options file 192, 193, 197
TSM password 192, 193, 197
TSQL 158

U
UNIX
  running commands on 2
    setting up the Tivoli environment 3
    shell 2
user
  connections 126
Users/Transactions
  about 118

V
variables, notation for ix

W
wdmlseng 7
Windows
  setting up the Tivoli environment 3
Windows 2000
  running commands on 2
Windows NT
  running commands on 2
write
  lazy 28
write pages 23
wruntask 135