IBM Tivoli Composite Application Manager Agent for DB2
Version 7.1

User's Guide

SC27-2491-01

IBM Tivoli Composite Application Manager Agent for DB2
Version 7.1

User's Guide

SC27-2491-01

Note

Before using this information and the product it supports, read the information in Appendix H, “Notices,” on page 479.

This edition applies to version 7.1 of the IBM Tivoli Composite Application Manager Agent for DB2 (5724-V09) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Tables

Chapter 1. Overview of the DB2 agent

The IBM® Tivoli® Composite Application Manager Agent for DB2® provides you with the capability to monitor DB2 databases, and to perform basic actions with DB2 databases. This chapter provides a description of the features, components, and interface options for the DB2 agent.

IBM Tivoli Monitoring overview

IBM Tivoli Monitoring is the base software for the DB2 agent. IBM Tivoli Monitoring provides a way to monitor the availability and performance of all the systems in your enterprise from one or several designated workstations. It also provides useful historical data that you can use to track trends and to troubleshoot system problems.

You can use IBM Tivoli Monitoring to do the following things:

- Monitor for alerts on the systems that you are managing by using predefined situations or custom situations.
- Establish your own performance thresholds.
- Trace the causes leading to an alert.
- Gather comprehensive data about system conditions.
- Use policies to perform actions, schedule work, and automate manual tasks.

The Tivoli Enterprise Portal is the interface for IBM Tivoli Monitoring products. By providing a consolidated view of your environment, the Tivoli Enterprise Portal can be used to monitor and resolve performance issues throughout the enterprise.

For complete information about IBM Tivoli Monitoring and the Tivoli Enterprise Portal, see the IBM Tivoli Monitoring publications listed in "Prerequisite publications" on page 475.

Features of the DB2 agent

The DB2 agent offers a central point of management for your DB2 environment. It provides a comprehensive means for gathering exactly the information you need to detect problems early and to prevent them. Information is standardized across the system. You can monitor a multitude of servers from a single Tivoli Enterprise Portal, each server monitored by a DB2 agent. Using the DB2 agent, you can easily collect and analyze specific information, including information on the following items:

- Applications with the highest percentage of failed SQL statements, sort overflows, lock timeouts and deadlocks, and the lowest buffer pool hit ratio
- Buffer pool hit ratio by buffer pool, buffer pool hit ratio by database, average read and write times, asynchronous and synchronous I/O activity, extended store and non-buffer pool I/O activity
- Databases with the highest percentage of failed SQL statements, the lowest buffer pool hit ratio, and the highest number of connections, lock timeouts, and deadlocks
- Database High Availability and Disaster Recovery (HADR) configuration and runtime information.

- Applications currently waiting for locks and other details about lock resources
- Server key events, the number of server connections, the databases with the lowest buffer pool hit ratio, and applications with the highest percentage of failed SQL statements
- Usage and page size of tablespaces
- Resources of the system where the DB2 Server is running, including CPU usage, memory usage, and network information
- Log information, such as the amount of active log space, the percentage of used secondary log, and the amount of space that is used by archive logs
- Diagnostic messages from the db2diag.log file of the DB2 database, including information on administration notification log, event log, and diagnostic log
- Customized SQL statement definitions, last execution statuses and results

New in this release

For version 7.1 of the DB2 agent, the following enhancements have been made since version 6.2.2, including the fix packs:

- New attribute groups
  - DB2 HADR (KUD_DB2_HADR)
  - DB2 Customized SQL Definition (KUD_Customized_SQL_Definition)
  - DB2 Customized SQL Status (KUD_Customized_SQL_ Status)
  - DB2 Customized SQL Detail (KUD_Customized_SQL_Detail)
  - DB2 Agent Event (KUD_Agent_Event)
  - DB2 Tablespace Auto-resize (KUD_Tablespace_Auto_Resize)
- New predefined situations
  - UDB_HADR_Con_Status_Congest
  - UDB_HADR_Con_Status_Disconnect
  - UDB_HADR_Primary_Status_Warn
  - UDB_HADR_Primary_Status_Warn_2
  - UDB_HADR_Primary_Down
  - UDB_HADR_Standby_Down
  - UDB_HADR_Stopped_Warn
  - UDB_Customized_SQL_Failed
  - UDB_Agent_Insufficient_Auth
  - UDB_Agent_DM_Down
  - UDB_Agent_DB_Standby
  - UDB_TS_Utilization_Crit
  - UDB_TS_Utilization_Warn
- Support for High Availability and Disaster Recovery (HADR) monitoring:
  - The Database HADR workspace was added to the Database node. In this workspace, you can view the HADR configuration information and the HADR status.
- Support for customized SQL monitoring:
  - The Customized SQLs navigator node was added to DB2 agent node.
  - Two workspaces, Customized SQLs and Customized SQL Result are under the new Customized SQLs navigator node. Use these workspaces to view customized SQL definitions, the latest execution status and customized SQL execution results.

- Support for predefined agent events monitoring:
  – The Agent Event workspace that features predefined events was added. This workspace helps to determine problems with the monitored database.
- Support for monitoring the automatic resizing of tablespaces:
  – Two situations were added which monitor the resizing of DMS tablespaces.

DB2 agent components

After you install the DB2 agent (product code "kud" or "ud") as directed in the *IBM Tivoli Monitoring Installation and Setup Guide*, you have an environment that contains the client, server, and monitoring agent implementation for IBM Tivoli Monitoring that contains the following components:

Tivoli Enterprise Portal client
: The portal has a user interface based on Java for viewing and monitoring your enterprise.

Tivoli Enterprise Portal Server
: The portal server is placed between the client and the Tivoli Enterprise Monitoring Server and enables retrieval, manipulation, and analysis of data from the monitoring agents. The Tivoli Enterprise Portal Server is the central repository for all user data.

Tivoli Enterprise Monitoring Server
: The monitoring server acts as a collection and control point for alerts received from the monitoring agents, and collects their performance and availability data. The Tivoli Enterprise Monitoring Server is also a repository for historical data.

Tivoli Enterprise Monitoring Agent, ITCAM Agent for WebSphere DB2 (one or more instances of the monitoring agent).
: The instances communicate with the systems or subsystems that you want to monitor. This monitoring agent collects and distributes data to a Tivoli Enterprise Portal Server.

IBMTivoli Enterprise Console®
: The Tivoli Enterprise Console is an optional component that acts as a central collection point for events from a variety of sources, including events from other Tivoli software applications, Tivoli partner applications, custom applications, network management platforms, and relational database systems. You can view these events through the Tivoli Enterprise Portal (by using the event viewer), and you can forward events from IBMTivoli Monitoring situations to the IBMTivoli Enterprise Console component.

IBMTivoli Netcool/OMNIbus
: Tivoli Netcool/OMNIbus is an optional component and an alternative to the IBMTivoli Enterprise Console. The Netcool/OMNIbus software is a service level management (SLM) system that delivers real-time, centralized monitoring of complex networks and IT domains. The Tivoli Netcool/OMNIbus components work together to collect and manage network event information.

Tivoli(r) Common Reporting
: Tivoli(r) Common Reporting is a separately installable feature available to users of Tivoli software that provides a consistent approach to generating and customizing reports. Some individual products provide reports that are designed for use with Tivoli(r) Common Reporting and have a consistent look and feel.For IBM Tivoli Monitoring for Virtual

Environments, you can use Tivoli(r) Common Reporting as a separate installation or as part of the IBM Tivoli Monitoring for Virtual Environments Performance and Capacity Management Reports capability.

IBMTivoli Monitoring for Virtual Environments Dashboard, reporting, and Capacity Planner capabilities
The dashboard capability provides a summary view of the health of the entire environment so you can quickly assess if a problem exists and take action to address the problem. Predefined performance and capacity management reports provide a complete assessment of the capacity (including forecast) of the virtual environment based on actual historical usage. With capacity planner analytics and reports you can create what-if planning scenarios that can be used to optimize and consolidate the virtual environment.

IBM(r)Tivoli(r) Application Dependency Discovery Manager (TADDM)
TADDM delivers automated discovery and configuration tracking capabilities to build application maps that provide real-time visibility into application complexity.

IBM(r)Tivoli(r) Business Service Manager
The IBM(r)Tivoli(r) Business Service Manager component delivers real-time information to help you respond to alerts effectively based on business requirements. Optionally, you can use this component to meet service-level agreements (SLAs). Use the Tivoli(r) Business Service Manager tools to help build a service model that you can integrate with Tivoli(r) Netcool/OMNIbus alerts or optionally integrate with data from an SQL data source. Optional components provide access to data from other IBMTivoli applications such as Tivoli(r) Monitoring and TADDM.

Agent Management Services

The Tivoli Composite Application Manager Agent for DB2 can be managed by the Agent Management Services of IBM Tivoli Monitoring 7.1. These services are available in the IBM Tivoli Monitoring OS Monitoring Agent for Windows, Linux, and UNIX systems. The services keep the Tivoli Composite Application Manager Agent for DB2 available, and provide information about the status of the agent to the Tivoli Enterprise Portal. For information about the Agent Management Services, see "Agent Management Services" in the *IBM Tivoli Monitoring Administrator's Guide*.

User interface options

Installation of the base software and other integrated applications provides the following interfaces that you can use to work with your resources and data:

Tivoli Enterprise Portal user interface
You can run the Tivoli Enterprise Portal as a desktop application or a browser application. The client interface is a graphical user interface (GUI) based on Java on a Windows or Linux workstation. The browser application is automatically installed with the Tivoli Enterprise Portal server. The desktop application is installed by using the Tivoli Monitoring installation media or with a Java Web Start application. To start the Tivoli Enterprise Portal browser client in your Internet browser, enter the URL for a specific Tivoli Enterprise Portal browser client installed on your Web server.

Command-line Interface
You can use IBM Tivoli Monitoring commands to manage the Tivoli Monitoring components and their configuration. You can also run commands at the Tivoli Enterprise Console event server or the Tivoli Netcool/OMNIbus ObjectServer to configure event synchronization for enterprise situations.

IBM Tivoli Enterprise Console
You can use the Tivoli Enterprise Console to help ensure the optimal availability of an IT service for an organization. The Tivoli Enterprise Console is an event management application that integrates system, network, database, and application management.

Manage Tivoli Enterprise Monitoring Services window
You can use the window for the Manage Tivoli Enterprise Monitoring Services utility to configure the agent and start Tivoli services not designated to start automatically.

Tivoli Netcool/OMNIbus event list
You can use the event list to monitor and manage alerts. An alert is created when the ObjectServer receives an event, alarm, message, or data item. Each alert is made up of columns (or fields) of information that are held in a row in the ObjectServer alerts.status table. The Tivoli Netcool/OMNIbus web GUI is also a web-based application that processes network events from one or more data sources and presents the event data in various graphical formats.

IBM(r)Tivoli Enterprise Console(r)
You can use the Tivoli Enterprise Console to help ensure the optimal availability of an IT service for an organization. The IBM(r)Tivoli Enterprise Console(r) is an event management application that integrates system, network, database, and application management. If you do not already use IBM(r)Tivoli Enterprise Console(r) and need an event management component, you can choose to use Tivoli(r) Netcool/OMNIbus.

Tivoli(r) Common Reporting
Use the Tivoli(r) Common Reporting web user interface for specifying report parameters and other report properties, generating formatted reports, scheduling reports, and viewing reports. This user interface is based on the Tivoli(r) Integrated Portal.

IBMTivoli Monitoring for Virtual Environments Dashboard, reporting, and Capacity Planner capabilities
This user interface is based on the Tivoli(r) Integrated Portal. The Dashboard provides predefined contextual summary views of the health (availability, performance, and capacity) of the complete virtual environment. Performance and Capacity Management Reports provides predefined Cognos-based reports that contain historical data, and a data model with tools for creating ad hoc reports. Capacity Planner provides you with a tool to import data for analysis and observe trends and patterns that you use to generate recommendations and create reports in the dashboard.

IBM(r)Tivoli(r) Application Dependency Discovery Manager
The Discovery Management Console is the TADDM client user interface for managing discoveries.

IBM(r)Tivoli(r) Business Service Manager
The Tivoli(r) Business Service Manager console provides a graphical user

interface that you can use to logically link services and business requirements within the service model. The service model provides an operator with a second-by-second view of how an enterprise is performing at any moment in time or how the enterprise performed over a time period.

IBM SmartCloud™ Application Performance Management UI
This user interface provides easy to understand and customize dashboards for IBM SmartCloud Application Performance Management, Tivoli(r) Monitoring, and IBMTivoli Composite Application Manager products. The UI includes predefined templates for simple J2EE, complex J2EE, and SAP applications, based on industry best practices. You can use these templates to quickly and easily to build your own dashboards to monitor the resources of your IT environments. The UI contains integrated views to monitor the Tivoli(r) Monitoring situation-based events and transactions of your applications.

Chapter 2. Agent-specific installation and configuration for the monitoring agent

Before installing and configuring the agent, make sure that your environment meets the requirements for the DB2 agent. See the Prerequisites topic for the agent in the IBM Tivoli Composite Application Manager for Applications information center at http://publib.boulder.ibm.com/infocenter/tivihelp/v24r1/topic/com.ibm.itcama.doc_7.1/welcome_itcamfapps71.html.

To install and configure the DB2 agent, use the procedures for installing monitoring agents in the *IBM Tivoli Monitoring Installation and Setup Guide* along with the agent-specific information in this chapter.

If you are performing a silent installation using a response file, see the information about performing a silent installation in the *IBM Tivoli Monitoring Installation and Setup Guide*.

This chapter contains information about the following topics and procedures relevant to the installation and configuration of the DB2 agent:

- "Upgrading note"
- "Installing language packs" on page 8
- "Installation and Configuration" on page 11
  - "Basic installation and configuration" on page 11
  - "Reconfiguration" on page 17
  - "Running the agent as a non-Administrator user" on page 18
  - "Setting environment variables" on page 21
  - "Starting and stopping the DB2 agent" on page 22
  - "Setting up the DB2 agent in a Microsoft Cluster Server environment" on page 25

Upgrading note

If you upgraded DB2 agent Version 6.2 to DB2 agent Version 7.1, the node name that is displayed in the Tivoli Enterprise Portal for the DB2 agent has the following format: *Instance*:*Hostname*:UD. Where *Instance* is the name of the agent instance, and *Hostname* is the name of the system where the agent is running.

If you upgraded DB2 agent Version 6.1 to DB2 agent Version 7.1, the node name that is displayed in the Tivoli Enterprise Portal for the DB2 agent has the following format: *Instance*:*Hostname*, instead of *Instance*:*Hostname*:UD.

If you upgrade the DB2 database software you must restart the DB2 agent to continue using the agent to monitor databases.

If you intend to upgrade DB2 from a version 9 installation to a version 10 installation you must first stop the DB2 agent, then do the upgrade and then restart the agent.

Prerequisite Checking

By using new tools in Tivoli Monitoring, you can perform prerequisite checking for DB2 agents before carrying out an installation. The two mechanisms available are a manually executed, stand-alone prerequisite scanner and a remote prerequisite scanner facility that extends the capabilities of the IBM Tivoli Monitoring remote deployment component. The prerequisite checking tools check the OS version, memory, disk, and DB2 version of the target server where the DB2 agent is to be installed, and delivers a pass or fail status according to the checking results. For more information about the prerequisite checker, see "Prerequisite Checking for IBM Tivoli Monitoring Agents" in the *IBM Tivoli Monitoring Installation and Setup Guide*.

Note:

- The product code for the DB2 agent is KUD: specify KUD as the product code when running the commands described in the information about how to use the checking tool.
- On Windows, the DB2 version check script calls the`db2level` command to retrieve the DB2 version. Make sure this command is available, otherwise the tool fails .
- On UNIX/LINUX, the DB2 version check script calls the `db2level` command to detect the DB2 version. If the command is not found then the script checks the default DB2 user home folder to find the DB2 version. If multiple DB2 users for different DB2 installations exist in the default home directory for example */home/*, the first one is used and the DB2 version of the installation is returned.

Installing language packs

To install a language pack for the agent support files on the Tivoli Enterprise Monitoring Server, the Tivoli Enterprise Monitoring Agent, and the Tivoli Enterprise Portal Server, first make sure that you installed the product in the English language. Then, perform the following steps depending on which operating system and mode of installation you are using.

Windows systems

1. On the language pack CD, double-click the `lpinstaller.bat` file to start the installation program.
2. Select the language of the installer and click **OK**.
3. In the Introduction panel, click **Next**
4. Click **Add/Update** and click **Next**.
5. Select the folder in which the National Language Support package (`NLSPackage`) files are located.

   Typically the `NLSPackage` files are located in the `nlspackage` folder where the installer executable file is located.
6. Select the language support for the agent of your choice and click **Next**.

   Note: To make multiple selections, press Ctrl and select the language that you want.
7. Select the languages that you want to install and click **Next**.
8. Examine the installation summary page and click **Next** to begin installation.
9. After installation completes, click **Finish** to exit the installer.

10. Restart the Tivoli Enterprise Portal, Tivoli Enterprise Portal Server, and Eclipse Help Server if any of these components are installed.

UNIX or Linux systems

1. Run the following command to create a temporary directory on the computer. Make sure that the full path of the directory does not contain any spaces:
   ```
   mkdir dir_name
   ```
2. Mount the language pack CD to the temporary directory that you created.
3. Run the following command to start the installation program:
   ```
   cd dir_name
   lpinstaller.sh -c install_dir
   ```
   where *install_dir* is where you installed IBM Tivoli Monitoring. Typically it is `/opt/IBM/ITM` for AIX and Linux systems.
4. Select the language of the installer and click **OK**.
5. In the Introduction panel, click **Next**.
6. Click **Add/Update** and click **Next**.
7. Select the folder in which the National Language Support package (`NLSPackage`) files are located.

   Typically, the `NLSPackage` files are located in the `nlspackage` folder where the installer executable file is located.
8. Select the language support for the agent of your choice and click **Next**.

   Note: To make multiple selections, press Ctrl and select the language that you want.
9. Select the languages that you want to install and click **Next**.
10. Examine the installation summary page and click **Next** to begin installation.
11. After installation completes, click **Finish** to exit the installer.
12. Restart the Tivoli Enterprise Portal, Tivoli Enterprise Portal Server, and Eclipse Help Server if any of these components are installed.

Silent installation of language packs for agents

You can install the language pack using a silent installation method:

1. Copy and paste the following response file template as `ITM_Agent_LP_silent.rsp`.
2. Change the following parameter setting: *`NLS_PACKAGE_FOLDER`*; *`PROD_SELECTION_PKG`*; *`BASE_AGENT_FOUND_PKG_LIST`*; *`LANG_SELECTION_LIST`*.

   Where:

   NLS_PACKAGE_FOLDER
   : Folder in which the National Language Support package (NLSPackage) files are located. Typically the NLSPackage files are located in the nlspackage folder. For example: NLS_PACKAGE_FOLDER=//tmp//LP//nlspackage.

   PROD_SELECTION_PKG
   : Which LangPack to install. There can be several components in one language package, but you might want only some of them.

   BASE_AGENT_FOUND_PKG_LIST
   : Agent for which you are installing language support. This value is usually the same as PROD_SELECTION_PKG.

LANG_SELECTION_LIST
Language you want to install.

3. Run the command to silently install the language pack:
   - For Windows systems:
     `lpinstaller.bat -f` *`path_to_response_file`*
   - For UNIX or Linux systems:
     `lpinstaller.sh -c` *`candle_home`* `-f` *`path_to_response_file`*

     where *candle_home* is the IBM Tivoli Monitoring base directory

Response file template:

```
#     IBM Tivoli Monitoring Agent Language Pack Silent Installation Operation
#
#This is a sample response file for silent installation mode for the IBM Tivoli
#Monitoring Common Language Pack Installer.
#.
#This file uses the IBM Tivoli Monitoring Common Agent Language Pack with the
#install package as an example.
#Note:
#This response file is for the INSTALLATION of language packs only.
#This file does not support UNINSTALLATION of language packs in silent mode.
#-------------------------------------------------------------------------------
#-------------------------------------------------------------------------------
#To successfully complete a silent installation of the the example of Common Agent
#localization pack, complete the following steps:
#
#1.Copy ITM_Agent_LP_silent.rsp to the directory where lpinstaller.bat or
#lpinstaller.sh is located (IBM Tivoli Monitoring Agent Language Pack build
#location).
#
#2.Modify the response file so that it is customized correctly and completely for
#your site.
#  Complete all steps listed below in the response file.
#
#3.After customizing the response file, invoke the silent installation using the
#following command:
#For Windows:
#    lpinstaller.bat -f <path_to_response_file>
#For UNIX and Linux:
#    lpinstaller.sh -c <candle_home> -f <path_to_response_file>
#Note:<candle_home> is the IBM Tivoli Monitoring base directory.
#--------------------------------------------------------------------------------

#--------------------------------------------------------------------------------
#Force silent install mode.
#--------------------------------------------------------------------------------
INSTALLER_UI=silent

#---------------------------------------------------------------------------------
#Run add and update actions.
#---------------------------------------------------------------------------------
CHOSEN_INSTALL_SET=ADDUPD_SET

#---------------------------------------------------------------------------------
#NLS Package Folder, where the NLS Packages exist.
#For Windows:
#   Use the backslash-backslash(\\) as a file separator (for example,
#C:\\zosgmv\\LCD7-3583-01\\nlspackage).
#For UNIX and Linux:
#   Use the slash-slash (//) as a file separator (for example,
#//installtivoli//lpsilenttest//nlspackage).
#---------------------------------------------------------------------------------
```

```
#NLS_PACKAGE_FOLDER=C:\\zosgmv\\LCD7-3583-01\\nlspackage
NLS_PACKAGE_FOLDER=//tmp//LP//nlspackage

#-------------------------------------------------------------------------------
#List the packages to process; both variables are required.
#Each variable requires that full paths are specified.
#Separate multiple entries with a semicolon (;).
#For Windows:
#          Use the backslash-backslash(\\) as a file separator.
#For Unix and Linux:
#          Use the slash-slash (//) as a file separator.
#-------------------------------------------------------------------------------
#PROD_SELECTION_PKG=C:\\zosgmv\\LCD7-3583-01\\nlspackage\\KIP_NLS.nlspkg
#BASE_AGENT_FOUND_PKG_LIST=C:\\zosgmv\\LCD7-3583-01\\nlspackage\\KIP_NLS.nlspkg
PROD_SELECTION_PKG=//tmp//LP//nlspackage//kex_nls.nlspkg;//tmp//LP//nlspackage//
koq_nls.nlspkg
BASE_AGENT_FOUND_PKG_LIST=//tmp//LP//nlspackage//kex_nls.nlspkg;//
tmp//LP//nlspackage//koq_nls.nlspkg

#------------------------------------------------------------------------------
#List the languages to process.
#Separate multiple entries with semicolons.
#------------------------------------------------------------------------------
LANG_SELECTION_LIST=pt_BR;fr;de;it;ja;ko;zh_CN;es;zh_TW
```

Installation and Configuration

When performing the steps to install and configure the DB2 agent as described in the *IBM Tivoli Monitoring Installation and Setup Guide*, "Installing monitoring agents," use the agent-specific configuration information provided for the DB2 agent.

Agent-specific information includes the following procedures:
- "Basic installation and configuration"
  - "Local"
  - "Remote" on page 15
- "Reconfiguration" on page 17
  - "Local" on page 17
  - "Remote" on page 18
- "Running the agent as a non-Administrator user" on page 18
- "Setting environment variables" on page 21
- "Starting and stopping the DB2 agent" on page 22
- "Setting up the DB2 agent in a Microsoft Cluster Server environment" on page 25

Do not attempt to start the DB2 agent until you have completed the configuration steps for installing the DB2 agent.

Basic installation and configuration

You can install and configure the DB2 agent locally or remotely using a GUI or the command line.

Local

If you are installing and configuring locally, use the steps in the *IBM Tivoli Monitoring Installation and Setup Guide*, "Installing monitoring agents." Also, use the agent-specific configuration information in this section.

Using the Managed Tivoli Enterprise Monitoring Services: To configure a DB2 agent instance after installation, do one of the following procedures:

- Windows systems:
  1. Click **Start** > **Programs**> **IBM Tivoli Monitoring**> **Manage Tivoli Monitoring Services**
  2. *(Optional)* If you need to change the communication settings, right-click the **ITCAM Agent for DB2** item with the **Task/SubSystem** column value of **Template**, and click **Advanced**>**Configure Advanced**.
  3. Right-click the **ITCAM Agent for DB2** item with the **Task/SubSystem** column value of **Template**, and click **Configure Using Defaults**.
  4. Enter the real name of the DB2 instance that you want to monitor, and click **OK**. The DB2 instance must be of type (ese) which is the default type for DB2 Enterprise Server Edition and DB2 Advanced Enterprise Server Edition. The DB2 instance name must also conform to the following guidelines:
     – Each name must begin with an alpha character.
     – Do not use blanks or special characters ("$#@").
     – The name must be short enough to fit within the total managed system name, which must be between 2 and 32 characters in length.
     – DB2 agent naming is case sensitive on all operating systems.
  5. Specify the DB2 properties settings. For the descriptions of the fields, see Table 1.

Table 1. DB2 Properties field description

Field Name	Description
The db2diag log file path	The directory of the DB2 diagnostics log file. If the db2diag log file is in the default directory, you can leave this field blank; if the file is not in the default directory, enter the path of the directory. The default directory is `C:\Documents and Settings\All Users\Application Data\IBM\DB2\`*`DB2INSTANCENAME`* on Windows systems, and `/home/`*`DB2owner_home_dir`*`/sqllib/db2dump` on UNIX and Linux systems. This field is only required if the DB2 instance to be monitored is a local instance. If the DB2 instance is a remote instance, the value in this field is ignored.
MSGID filter in regular expression	This is an optional field. Fill MSGIDs in this field to monitor only the messages that you are interested in. The MSGID is a combination of the message type, message number, and severity level. You can also use a regular expression. For example, ADM1\d*1E|ADM222\d2W.
Enable monitoring partitions on a remote host	Choose Yes to enable the DB2 agent instance to monitor partitions in a remote host.
Enable monitoring of all databases	Choose Yes to enable the DB2 agent instance to monitor all databases including the inactive databases and non-connection databases in the DB2 instance named in the DB2 instance name field.

Table 1. DB2 Properties field description (continued)

Field Name	Description
Customized SQL definition file name	This attribute is the full file path name of the customized SQL definition file. The SQL statements defined in this file and related information is displayed in the Customized SQLs workspace. The default file name is `CANDLEHOME/config/kudcussql.properties` on UNIX or Linux systems, `CANDLEHOME\TMAITM6\kudcussql.properties` on Windows systems, and `CANDLEHOME\TMAITM6_x64\kudcussql.properties` on Windows 64-bit systems.

6. Click **OK** to complete the configuration.

   To create multiple instances of the DB2 agent, repeat step 3 to step 6. For each DB2 agent instance, a KUDENV_*Instance* file and a *hostname*_ud_*Instance*.cfg file are created, where *Instance* is the name of the DB2 agent instance.

- UNIX and Linux systems:
  1. To open the Manage Tivoli Enterprise Monitoring Services window, navigate to the *ITMinstall_dir*/bin directory, where *ITMinstall_dir* is the installation directory of IBM Tivoli Monitoring, and run the following command:
     ```
     ./itmcmd manage
     ```
  2. Select **IBM Tivoli Composite Application Manager Agent for DB2**, and right-click it.
  3. Click **Configure**.
  4. In the Manage Application Instances window, click **Add Instances**.
  5. Enter the real name of the DB2 instance that you want to monitor, and click **OK**. The DB2 instance must be of type (ese) which is the default type for DB2 Enterprise Server Edition and DB2 Advanced Enterprise Server Edition. The DB2 instance name must also conform to the following guidelines:
     - Each name must begin with an alpha character.
     - Do not use blanks or special characters ("$#@").
     - The name must be short enough to fit within the total managed system name, which must be between 2 and 32 characters in length.
     - DB2 agent naming is case sensitive on all operating systems.
  6. Specify the DB2 properties settings. For the descriptions of the fields, see Table 1 on page 12.
  7. Click **OK**.
  8. Configure the connection between the agent and the Tivoli Enterprise Monitoring Server.
  9. Click **Save** to complete the configuration.

  To create multiple instances of the DB2 agent, repeat step 2 to step 9. For each DB2 agent instance, a *hostname*_ud_*Instance*.cfg file is created, where *Instance* is the name of the DB2 agent instance.

Using the itmcmd command line: To configure the DB2 agent on UNIX or Linux systems from the itmcmd command line, use the itmcmd config command. See the *IBM Tivoli Monitoring Command Reference* for complete information about this command.

1. Navigate to the *ITMinstall_dir*/bin directory, and run the following command:

```
./itmcmd config -A ud
```

   where *ITMinstall_dir* is the installation directory of IBM Tivoli Monitoring.
2. Type in the name of the DB2 instance that you want to monitor, and press Enter.
3. Press Enter when you are asked whether you want to edit IBM Tivoli Composite Application Manager Agent for DB2 settings. The default value is Yes.
4. Press Enter when you are asked whether you want to edit DB2 properties settings. The default value is Yes.
5. Specify the db2diag log file path:
   - If the db2diag log file is in the default directory, press enter.
   - If the file is in a directory other than the default one, type the path of the directory, and press Enter.

   The default directory is `C:\Documents and Settings\All Users\Application Data\IBM\DB2\`*`DB2INSTANCENAME`* on Windows systems, and `/home/`*`DB2owner_home_dir`*`/sqllib/db2dump` on UNIX and Linux systems.
6. *(Optional)* For the MSGID filter, type the msgid and press Enter to monitor only the messages that you are interested in. The msgid is a combination of the message type, message number, and severity level. You can also use a regular expression. For example, ADM1\d*1E|ADM222\d2W.
7. Specify whether to enable monitoring partitions in a remote host, and press Enter. The default value is YES.
8. Specify whether to enable monitoring of all databases, and press Enter. The default value is YES.
9. Specify the absolute name of the customized sql statement definitions file
   - If the customized sql statement definition file is in the default directory press Enter.
   - If the file is in a directory other than the default one, type the absolute name of the file, and then press Enter.

   The default file name is `CANDLEHOME/config/kudcussql.properties` on UNIX or Linux systems, `CANDLEHOME\TMAITM6\kudcussql.properties` on Windows systems, and `CANDLEHOME\TMAITM6_x64\kudcussql.properties` on Windows 64-bit systems.
10. Press Enter when you are asked whether the agent connects to a Tivoli Enterprise Monitoring Server.
11. Configure the connection between the agent and the Tivoli Enterprise Monitoring Server:
    a. Type the host name of the Tivoli Enterprise Monitoring Server, and press Enter.
    b. Choose the network protocol that the Tivoli Enterprise Monitoring Server uses to communicate with the agent, you have four choices: IP, SNA, IP.PIPE, or IP.SPIPE. The default value is IP.PIPE
    c. Depending on the type of protocol you specified, provide required information when prompted:
    d. Choose whether you want to configure the connection to a secondary Tivoli Enterprise Monitoring Server. The default value is No.

e. Press Enter to accept the default value for the optional primary network name. The default value is none.

You can create multiple DB2 agent instances. For each agent instance, a *hostname*_ud_*Instance*.cfg file is created. For each agent instance on Windows systems, a KUDENV_*Instance* file is also created.

Silent installation: If you are performing a silent installation using a response file, see the *IBM Tivoli Monitoring Installation and Setup Guide*, "Performing a silent installation of IBM Tivoli Monitoring."

Remote

IBM Tivoli Monitoring provides the ability to deploy monitoring agents from a central location, which is the monitoring server. You can also use the remote agent deployment function to configure deployed agents and install maintenance on your agents.

Before you can deploy any agents from a monitoring server, you must first populate the agent depot with bundles. For information about populating your agent depot, see the *IBM Tivoli Monitoring Installation and Setup Guide*.

Important: You can create multiple DB2 agent instances. For each agent instance, a *hostname*_ud_*Instance*.cfg file is created. For each agent instance on Windows systems, a KUDENV_*Instance* file is also created.

Using the Tivoli Enterprise Portal: For detailed information about deploying non-OS agents, see the steps in the *IBM Tivoli Monitoring Installation and Setup Guide*, "Deploying non-OS agents."

Important: After you add the agent bundle to the Tivoli Enterprise Monitoring Server, if the IBM Tivoli Composite Application Manager Agent for DB2 is still not listed in the Select a Monitoring Agent window, deploy the agent through the command line.

In the New Managed System Configuration window, enter configuration information of the monitoring agent. See Table 2 on page 16 for the descriptions of the fields.

Using the tacmd command line: On UNIX or Linux systems, each time that the DB2 agent is started, it starts as the DB2 user ID specified during DB2 agent instance configuration. In order to start the DB2 agent with the specified user ID, the OS monitoring agent requires root authority to change to the specified user before starting the monitoring agent.

To deploy this monitoring agent remotely using the command line, use the procedure, "Deploying through the command line," in the *IBM Tivoli Monitoring Installation and Setup Guide*. Also, use the agent-specific configuration information in Table 2 on page 16 for the tacmd addSystem command. The *IBM Tivoli Monitoring Command Reference* has complete information about the tacmd addSystem command.

Use the -t or--type TYPE parameter to specify that the you are configuring a DB2 agent. The product code for DB2 agent is UD.

Specify the agent properties with the -p or -property option.

The following example is for Windows systems:

```
tacmd addSystem -t ud -n Primary:hostname:NT -p INSTANCE="DB2"
```

The following example is for UNIX and Linux systems:

```
tacmd addSystem -t ud -n hostname:KUX -p INSTANCE="db2inst1"
_UNIX_STARTUP_.Username="db2inst1"
```

Table 2. Names and descriptions of configuration settings for each interface

Tivoli Enterprise Portal		tacmd command line	Description	Example
On the **DB2 Properties** tab	DB2 Instance Name	INSTANCE= *InstanceName*	The name of the DB2 instance that is to be monitored. See the instance naming guidelines on page 12.	If the DB2 instance being monitored is a named instance with the instance name of `db2server` and the hostname is `popcorn`, enter `db2server` in this field.
	The db2diag log file path	DBSETTINGS.KUD _DIAGLOG _PATH=*logpath*	The file path of the DB2 diagnostics log file. If the db2diag log file is in the default directory, you can leave this field blank; if the file is not in the default directory, enter the path of the directory.	The default directory is: Windows Install_Driver:\Documents and Settings\All Users\Application Data\IBM\DB2\ *DB2COPYNAME*\ *DB2INSTANCENAME* on Windows systems. For example, C:\Documents and Settings\All Users\Application Data\IBM\DB2\ DB2COPY1\DB2. *`DB2InstanceOwner _Home_directory`*`/sqllib/db2dump` on UNIX and Linux systems. For example, /home/db2inst2/sqllib/db2dump.
	MSGID filter in regular expression	DBSETTINGS.KUD _DIAGLOG _MSGID_FILTER =*MSGID*	This is an optional field. Fill MSGIDs in this field to monitor only the messages that you are interested in. The MSGID is a combination of the message type, message number, and severity level. You can also use a regular expression.	ADM1\d*1E|ADM222\d2W
	Enable monitoring partitions in a remote host	DBSETTINGS. KUD_MONITOR _REMOTE _PARTITIONS= Yes (No)	Choose Yes to enable the DB2 agent instance to monitor partitions in a remote host.	The valid values are Yes and No.
	Enable monitoring of all databases	DBSETTINGS.KUD _MONITOR_ ALL_DATABASES =Yes(No)	Choose Yes to enable the DB2 agent instance to monitor all databases including the inactive databases and non-connection databases in the DB2 instance named in the DB2 instance name field.	The valid values are Yes and No
	Customized sql statement definition file name	DBSETTINGS. KUD_DB2_ SQL_PATH=sql file path	The absolute file name of the customized sql statement definitions file.	The default file name is: `CANDLEHOME/config/kudcussql.properties` on UNIX or Linux systems, `CANDLEHOME\TMAITM6\kudcussql.properties` on Windows systems, and `CANDLEHOME\TMAITM6_x64\ kudcussql.properties` on Windows 64-bit systems.

Table 2. Names and descriptions of configuration settings for each interface (continued)

Tivoli Enterprise Portal		tacmd command line	Description	Example
On the **Agent** tab	**Use this account** If selecting this option complete the following fields as described: • **Username**: the ID under which the agent instance is to run • **Password**: the password associated with the given ID	For UNIX and Linux systems: _UNIX_STARTUP_ .Username=*user* For Windows systems:_WIN32_ STARTUP _.Username and _WIN32_STARTUP _.Password	See "Running the agent as a non-Administrator user" on page 18. For UNIX and Linux systems, when entering information under **Run As**, be sure to enter the name of the DB2 instance owner in the **Username** field. For Windows systems, accept the default, LocalSystem account, or specify the DB2 instance owner. On Windows systems, the ID and passwords must have Windows Administrator authority for the system on which the DB2 agent is to run.	If the DB2 instance being monitored is a named instance, and you want to run as the database instance owner, enter `db2inst1` in this field.

Reconfiguration

If you need to reconfigure the DB2 agent, ensure that the steps for installing the monitoring agent in the *IBM Tivoli Monitoring Installation and Setup Guide* were completed.

Local

If you are reconfiguring an instance locally, use the Manage Tivoli Enterprise Monitoring Services window. See Table 2 on page 16 for the configuration settings.

- Windows systems:
  1. To open the Manage Tivoli Enterprise Monitoring Services window, click **Start** > **Programs**> **IBM Tivoli Monitoring**> **Manage Tivoli Monitoring Services**.
  2. Right-click the DB2 agent instance that you want to re-configure, and do one of the following steps:
     – To change the communication settings or the DB2 properties setting, click **Reconfigure**, and change the settings as you need.
     – To change the user ID under which the agent instance is running, click **Change Startup**, and change the settings as you need. To run the agent instance under a user ID other than the local system account, you can select **Use this account**, and fill in the user ID and password. The user ID that you use must meet the requirement in "User ID permissions required to run the DB2 agent" on page 18.
  3. Click **OK** to save your changes.
- UNIX and Linux systems:
  1. To open the Manage Tivoli Enterprise Monitoring Services window, navigate to the *ITMinstall_dir*/bin directory, where *ITMinstall_dir* is the installation directory of IBM Tivoli Monitoring, and run the following command:

     ```
     ./itmcmd manage
     ```
  2. Select **IBM Tivoli Composite Application Manager Agent for DB2**, and right-click it.

3. Click **Configure**.
4. In the Manage Application Instances window, select the DB2 agent instance that you want to re-configure, and click **OK**.
5. Change the settings as you need, and click **OK** to save your changes.
6. *(Optional)* To remove an existing DB2 agent instance, do the following steps:
   a. In the Manage Application Instances window, select the instance that you want to remove, and click **Remove Instance**.
   b. In the Remove Instance window, click **YES**.
   c. Click **OK**.

Remote

If you are reconfiguring remotely, use one of the following interfaces:

- tacmd command line

  Use the configureSystem command. See the *IBM Tivoli Monitoring Command Reference* for complete information about this command.

  The following example shows the use of configureSystem command:

  ```
  tacmd configuresystem -m db2inst1:hostname:UD
  -p INSTANCE="db2inst1"
  DBSETTINGS.KUD_DIAGLOG_PATH="/home/db2inst1/sqllib/db2dump"
  DBSETTINGS.KUD_DIAGLOG_MSGID_FILTER="ADM\d*W"
  DBSETTINGS.KUD_DB2_SQL_PATH="/opt/IBM/ITM/config/kudcussql.properties"
  ```

  When re-configuring, enter the information for the property that you are changing. See Table 2 on page 16 for the configuration settings.
- Tivoli Enterprise Portal
  1. Open the Tivoli Enterprise Portal.
  2. Navigate to the system where the agent that you want to configure is installed.
  3. Select the agent.
  4. Right-click the agent; then click **Configuration**.
  5. Modify the parameters as you need to. Refer to Table 2 on page 16 for the configuration settings.
  6. Click **OK** to save the changes.

Running the agent as a non-Administrator user

By default, the user ID that is used to run the DB2 agent is set to **LocalSystem**. You can also use a different user ID to run the DB2 agent; however, this user ID must adhere to the user ID requirements described in "User ID permissions required to run the DB2 agent."

User ID permissions required to run the DB2 agent

The DB2 agent can run under any user ID. Each DB2 agent acquires its permissions from the user ID under which the DB2 agent is running.

The ID used to run the DB2 agent must have DB2 SYSADM authority. SYSADM authority is required in order for the agent to turn on all monitor switches. For information about how to change the user ID under which the DB2 agent is run, see "Configuring the run-as user ID for the DB2 agent."

Configuring the run-as user ID for the DB2 agent

The DB2 agent is configured to run under an incorrect user ID, and you want to configure the DB2 agent to run under another user ID. This situation can occur if you installed the DB2 agent while running as a root or Administrator ID. Perform each of the following steps as it applies to your environment:

- Local

  Use the Manage Tivoli Monitoring Services to change the user ID. For detailed instructions, see the reconfiguration steps in “Local” on page 17.
- Remote
  - Change the user ID and password in the Run as section on the Tivoli Enterprise Portal GUI tab for the agent.
  - From the remote deploy command line, this Run-As user ID can be specified using the tacmd command-line parameter in Table 2 on page 16.

  Remember: On UNIX and Linux systems, when an agent instance is installed with the root user ID and started with the DB2 instance owner, and you need to stop the agent remotely from the Tivoli Enterprise Portal, you must change the Run-As user ID of the agent instance to the same DB2 instance owner before stopping the agent instance. Otherwise, the agent instance runs with the root user ID when you restart the instance.
- Restart

  By default the agent is started with the root user account in the autostart script. Refer to “Agent upgrade and restart using non-root” on page 352 for information on about updating the autostart script to start the agent with the SYSADM user ID.

For the list of user ID permissions required by the DB2 agent, see “User ID permissions required to run the DB2 agent” on page 18. For information about how to ensure that each instance of the DB2 agent is started and running under the correct user ID, see “Permissions for starting and stopping the DB2 agent on UNIX and Linux systems.”

Permissions for starting and stopping the DB2 agent on UNIX and Linux systems

On UNIX systems, the Monitoring Agent for UNIX OS requires root permission to start, stop, or restart the DB2 agent remotely.

- Starting the agent

  To change the user account from root to a non-administrator user account, use the **CHMOD** command to grant the privileges for specific directories in the installation path for IBM Tivoli Monitoring to the non-administrator user account. For example, if you use the root user account to install the DB2 agent and attempt to run the agent as the DB2 instance owner, you receive an error similar to the following example:

  ```
  db2inst3@aix7%> ./bin/itmcmd Agent -o db2inst3 start ud
  CandleAgent : installer level 400 / 100.
  find: 0652-023 Cannot open file /home/ITM/images.
  find: 0652-023 Cannot open file /home/ITM/images.
  find: 0652-023 Cannot open file /home/ITM/images.
  find: 0652-023 Cannot open file /home/ITM/images.
  CandleAgent : running aix523 jre.
  Starting agent...
  Agent Started...
  db2inst3@aix7%>
  ```

  In this example, the error is only for the `images` directory. The DB2 agent runs correctly, in spite of the error message. To avoid getting the `Cannot open file...` message, run the `chmod -R 755 *directory*` command, where *directory* is the directory that is specified in the error.

- Stopping the agent

  When the DB2 agent is running under a non-root user ID on a UNIX or Linux system, you must log in with the same user ID to stop the agent. Use the following command to obtain the non-root user ID:

  install_dir/bin/cinfo -r

Defining customized SQL statements

The customized SQL feature is designed to provide you with the ability to define your own SQL-based monitoring parameters for the DB2 agent.

The customized SQL feature supports only SELECT statements. You can define SQL statements in the `kudcussql.properties` file, which is in the following directory:

- Windows 32-bit systems: *Installation_Dir*/TMAITM6
- Windows 64-bit systems: *Installation_Dir*/TMAITM6_x64
- UNIX and Linux systems: *Installation_Dir*/config

Where *Installation_Dir* is the installation directory of the DB2 agent.

The following guidelines apply when you define SQL statements:

- Only ASCII characters are supported in the `kudcussql.properties` file.
- Each valid entry starts with `[SELECTSQL]`, and the entry ends when reaching another left-bracket character ([) at the far left of a line.
- In each `SELECTSQL` entry, you can define the `SQL_ID` and the `SQL_TEXT`.
  - `SQL_ID`: Each customized SQL entry must have a valid `SQL_ID` variable defined.

    The `SQL_ID` variable is case-sensitive, and can consist of only uppercase letters (A-Z), lowercase letters (a-z), the underscore character (_), and the hyphen character(-). If multiple `SELECTSQL` entries exist with the same `SQL_ID`, only the first entry is executed.
  - `SQL_TEXT`: Each customized SQL entry must have an `SQL_TEXT` variable.

    For the `SQL_TEXT` variable, only SELECT statements are supported. INSERT, UPDATE, or DELETE statements are not supported. The `SQL_TEXT` variable supports multiple lines and DB2 valid comment characters, **--**.
- The following table contains information about the supported DB2 database data type and data type mapping rules:

Table 3. Supported DB2 database data types

Customized SQL IBM Tivoli Monitoring data type	DB2 database data type	Mapping rule
S,256	SQL_CHAR	If length exceeds 256, it is truncated
S,256	SQL_VARCHAR	If length exceeds 256, it is truncated
S,256	SQL_WCHAR	If length exceeds 256, it is truncated
S,256	SQL_WVARCHAR	If length exceeds 256, it is truncated
S,256	SQL_WLONGVARCHAR	If length exceeds 256, it is truncated

Table 3. Supported DB2 database data types (continued)

Customized SQL IBM Tivoli Monitoring data type	DB2 database data type	Mapping rule
I,8 (BIGINT)	SQL_NUMERIC	To be rounded of to the nearest integer. For example 3.4 becomes 3.
I,8 (BIGINT)	SQL_DECIMAL	To be rounded of to the nearest integer. For example 3.4 becomes 3.
I,8 (BIGINT)	SQL_INTEGER	
I,8 (BIGINT)	SQL_SMALLINT	
I,8 (BIGINT)	SQL_FLOAT	To be rounded of to the nearest integer. For example 3.4 becomes 3.
I,8 (BIGINT)	SQL_REAL	To be rounded of to the nearest integer. For example 3.4 becomes 3.
I,8 (BIGINT)	SQL_DOUBLE	To be rounded of to the nearest integer. For example 3.4 becomes 3.
I,8 (BIGINT)	SQL_DECFLOAT	To be rounded of to the nearest integer. For example 3.4 becomes 3.
I,8 (BIGINT)	SQL_BIGINT	
T,16	SQL_DATETIME	All dates and times to be converted to YYYY-MM-DD HH:mm:SS, for example 2011-08-10 12:12:12.
T,16	SQL_TYPE_DATE	All dates and times to be converted to YYYY-MM-DD 00:00:00.
T,16	SQL_TYPE_TIME	The time is to be set to HH:mm:SS, and the date is to be set to the current date.
T,16	SQL_TYPE_TIMESTAMP	All dates and times to be converted to YYYY-MM-DD HH:mm:SS, for example 2011-08-10 12:12:12.

Setting environment variables

Perform one of the following procedures to set the environment variables for the DB2 agent:

- On Window systems, UNIX and Linux systems, open the configuration file, set the environment variables, and restart the agent instance for the changes to take effect. See Table 4 on page 22 for the description of the variables. On Windows systems, the file path of the configuration file is `install_dir\TMAITM6\KUDENV_InstanceName`; On UNIX and Linux systems, the file path of the configuration file is `install_dir/config/ud.ini`. Where *install_dir* is the installation directory of the DB2 agent, and *InstanceName* is the name of the agent instance.

- On Windows systems, you can use the Manage Tivoli Monitoring Services to set environment variables:
  1. To open the Manage Tivoli Enterprise Monitoring Services window, click **Start** > **Programs**> **IBM Tivoli Monitoring**> **Manage Tivoli Monitoring Services**.
  2. Right-click the agent instance that you want to configure, and click **Advanced**> **Edit Variables**.
  3. In the Override Local Variable Settings window, click **Add**.
  4. Fill in the names and values of the variables that you want to set, refer to Table 4 for the description of the variables.
  5. Click **OK** to close the Add Environment Setting Override window.
  6. Click **OK**.
  7. Restart the agent instance for the changes to take effect.

Table 4. Environment variable description

Variable	Description
KUD_MONITOR_ALL_DATABASES	Setting this parameter to YES enables the DB2 agent to monitoring all existing database instances including inactive ones.
KUD_MON_SWITCH_OVERRIDE	Add KUD_MON_SWITCH_OVERRIDE=YES in the configure file, and the DB2 agent does not set the existing OFF switch settings of DB2 instances to ON when the DB2 agent is restarted. This keeps the current switch settings of DB2 database instances unchanged.
KUD_DIAGLOG_CACHE	The maximum number of the message records or log records that are displayed in the Tivoli Enterprise Portal workspaces. The records are cached by agent using rotating mode. The oldest record is replaced by the new coming record. The default value is 20.
KUD_DIAGLOG_TAILCOUNT	The maximum number of message records or event records that are returned by Agent Builder/Factory Log Monitoring parsing when the DB2 agent starts up. This parameter limits the agent to process a large log file. Only the latest messages and events are monitored. The default value is 1000.
KUD_DBHISTORY_MAXROW	The maximum number of rows of archive log related history record from the SYSIBMADM.DB_HISTORY table that are displayed in the Tivoli Enterprise Portal workspace. The default value is 500.

Starting and stopping the DB2 agent

You can start or stop the DB2 agent for a given database instance.

The DB2 agent does not require advanced configuration. However, the DB2 agent must run under a user ID that adheres to the requirements described in "Running the agent as a non-Administrator user" on page 18. When starting or stopping this DB2 agent locally on UNIX and Linux systems, you must be logged in as this required user ID.

You can run multiple copies of the DB2 agent by specifying different database instance names. Only one process can be started for each database instance.

You can start and stop this monitoring agent using the Manage Tivoli Enterprise Monitoring Services utility, the Tivoli Enterprise Portal, or the itmcmd or tacmd command lines. Table 5 shows which interfaces you can use with Windows, UNIX and Linux systems locally and remotely.

Table 5. Interfaces for starting and stopping the DB2 agent

Operating system	Local	Remote
Windows	• Manage Tivoli Enterprise Monitoring Services • tacmd startAgent • tacmd stopAgent • tacmd restartAgent	• Tivoli Enterprise Portal • tacmd startAgent • tacmd stopAgent • tacmd restartAgent
UNIX and Linux	• Manage Tivoli Enterprise Monitoring Services • itmcmd agent	• Tivoli Enterprise Portal • tacmd startAgent • tacmd stopAgent • tacmd restartAgent

Manage Tivoli Enterprise Monitoring Services

To use Manage Tivoli Enterprise Monitoring Services to start the DB2 agent, use the ID under which the agent runs.

When starting or stopping the DB2 agent on UNIX and Linux systems using the Manage Tivoli Enterprise Monitoring Services, you are prompted for the name of the database instance to start or stop.

Enter only ASCII characters in the fields for the Manage Tivoli Enterprise Monitoring Services window.

Tivoli Enterprise Portal

See the "Working with monitoring agents," "Starting and stopping a monitoring agent" in the *IBM Tivoli Monitoring Administrator's Guide* for information about using the Tivoli Enterprise Portal to start or stop the DB2 agent.

itmcmd command line

Important: For information about how upgrading affects stopping the DB2 agent, see “Using the itmcmd and tacmd commands to stop the DB2 agent when upgrading to V7.1 from V6.1 (UNIX and Linux)” on page 24.

When using the itmcmd agent commands to start or stop this monitoring agent, include the following command option:

-o Specifies the database instance to start or stop. The database instance name must match the name used for starting the database.

If you start the DB2 agent without specifying the -o option, you receive the following error message: `This agent requires the -o option...`

For example:

```
./itmcmd agent -o db2inst1 start ud
```

For information about using the itmcmd commands, see the *IBM Tivoli Monitoring Command Reference*.

tacmd command line

Important: For information about how upgrading affects stopping the DB2 agent, see "Using the itmcmd and tacmd commands to stop the DB2 agent when upgrading to V7.1 from V6.1 (UNIX and Linux)."

In the following examples, the tacmd command is used to start, stop, or restart the DB2 agent:

- Local on Windows systems
  - `tacmd startAgent -t ud`
  - `tacmd stopAgent -t ud`
  - `tacmd restartAgent -t ud`
- Remote on Windows systems
  - `tacmd stopagent -t ud -n Primary:hostname:NT`
  - `tacmd startagent -t ud -n Primary:hostname:NT`
  - `tacmd restartagent -t ud -n Primary:hostname:NT`
- Remote on UNIX and Linux systems
  - `tacmd stopagent -t ud -n hostname:LZ`
  - `tacmd startagent -t ud -n hostname:LZ`
  - `tacmd restartagent -t ud -n hostname:LZ`

For more information about using the tacmd commands, see the *IBM Tivoli Monitoring Command Reference*.

Using the itmcmd and tacmd commands to stop the DB2 agent when upgrading to V7.1 from V6.1 (UNIX and Linux)

After updating to IBM Tivoli Monitoring V7.1 on UNIX or Linux operating systems, you can no longer stop the DB2 agent using the tacmd or itmcmd commands that call the following command:

```
CandleAgent -o instance stop ud
```

If your system is still running the IBM Tivoli Monitoring V6.1 code, you can avoid being in this position by stopping the DB2 agent prior to upgrading to version 6.2.

If you have already upgraded to IBM Tivoli Monitoring V6.2.2, you can install version 7.1 of the DB2 agent using a local installation or a remote installation, but you must first manually stop the DB2 agent process. To manually stop this DB2 agent, perform the following steps:

1. Use the following command to obtain a list of the Process IDs (PIDs) for the DB2 agent processes that are running:
   ```
   ./cinfo -r
   ```
2. In the output, look for lines containing `ud`.
3. Run the following command to stop the processes that are running:
   ```
   kill -9 PID
   ```
4. Update the status of the DB2 agent by running the following command for each instance of the DB2 agent:
   ```
   itmcmd agent -o instance stop ud
   ```

Setting up the DB2 agent in a Microsoft Cluster Server environment

To use this monitoring agent in a Microsoft Cluster Server environment requires special configuration.

The *IBM Tivoli Monitoring Installation and Setup Guide* contains an overview of clustering. The information provided here is specifically for installing and setting up the DB2 agent in a Microsoft Cluster Server environment.

Requirements

In addition to installing and setting up the DB2 agent, the following three additional steps are required for the cluster environment:

1. Setting CTIRA_HOSTNAME to a common value for all monitoring agents (usually the cluster name)
2. Setting CTIRA_HIST_DIR to a common disk location if history is stored at the DB2 agents (if history for the DB2 agent is configured to be stored at the DB2 agent)
3. Add monitoring agent resources to a cluster group.

On Windows systems, IBM Tivoli Monitoring requires that monitoring agents are installed in the same directory path as the OS agent. Therefore, each node in a cluster must have installed all monitoring agents (on the nodes system disk) that are required to support the cluster applications that can run on that cluster node.

Editing environment variables

Install the DB2 agent on each node in the cluster where the DB2 software is installed. Use the same installation path on all nodes in the cluster. After the installation, edit the following variables:

- The CTIRA_HOSTNAME variable for the DB2 agent: a common value for all monitoring agents (usually the cluster name)
- The CTIRA_HIST_DIR variable for the DB2 agent: a common disk location if history is stored at the DB2 agents
- The KFW_TOPOLOGY_CLUSTER_LIST variable for the Tivoli Enterprise Portal

To edit the variables, do the following steps:

1. To open the Manage Tivoli Enterprise Monitoring Services window, click **Start** > **Programs** > **IBM Tivoli Monitoring** > **Manage Tivoli Monitoring Services**.
2. If the DB2 agent instances are not stopped already, stop each instance.
3. Do the following steps for each agent instance:
   a. Right-click the agent instance, and click **Advanced** > **Edit Variables**.
   b. In the DB2 agent: Override Local Variable Settings window, click **Add**.
   c. Select CTIRA_HOSTNAME for the **Variable** field, edit the value, and click **OK**.

      Important: The directory path that you specify for CTIRA_HIST_DIR must exist if the logging is done on the DB2 agent to ensures that all the directories in the path are created. If you specify a non-existent path for CTIRA_HIST_DIR, historical data collection fails.
   d. In the DB2 agent: Override Local Variable Settings window, click **Add**.
   e. Select CTIRA_HOSTNAME for the **Variable** field, edit the value, and click **OK**.

Remember: Leave .TYPE=REG_EXPAND_SZ in the CTIRA_HOSTNAME variable when changing it so the registry is updated correctly.

f. In the DB2 agent: Override Local Variable Settings window, click **OK**.

g. Restart the DB2 agent instance.

4. Stop the Tivoli Enterprise Portal Server.
5. Right-click the Tivoli Enterprise Portal Server agent, and click **Advanced** > **Edit Variables**
6. Click **Add**.
7. Select KFW_TOPOLOGY_CLUSTER_LIST for the **Variable** field.
8. In the **Value** field, change the value to AFF_UNIVERSAL_DATABASE.
9. Restart the Tivoli Enterprise Portal Server.

Adding DB2 agent resources to a cluster group

To ensure that the correct monitoring agent is monitoring the correct server instance, the cluster group representing the monitored server must have the DB2 agent services added. To add the DB2 agent services is accomplished, identify the services and parameters that are required to run the DB2 agent as follows:

1. Click **Start** > **Administrative Tools** > **Services**.
2. In the list of Service names, click **DB2 Agent–** *InstanceName*.
3. Right-click **Properties**.
4. Make a note of the Service name: kudcma_ *InstanceName*
5. From the Cluster Administrator, add a new resource to the group to represent the DB2 agent services.
6. Click **Start** > **Administrative Tools** > **Cluster Administrator**.
7. Expand **Groups** in the left panel.
8. Click the group for the Instance that is being configured > **DB2 Group 0**.
9. Right-click, and then click **New** > **Resource**.
10. Fill in the following fields:

    Name: kudcma_ *InstanceName*

    Description: DB2 Agent

    Resource Type: Generic Service

    Group: DB2 Group 0
11. Click **Next**.
12. Select all **Available Nodes**.
13. Click **Add**, and then click **Next**.
14. Do not add any dependencies, and then click **Next**.
15. Fill in kudcma_ *InstanceName* for the **Service name** field, and keep the **Start Parameters** field blank.
16. Click **Next**; then click **Finish**.
17. Select the newly created resource, right-click it, and click **Properties**.
18. Select **Advance**, and clear the **Affect the group** check box. This step prevents a failover in case the DB2 agent fails.
19. Click **OK**.
20. To bring the DB2 agent online, select **kudcma_** *InstanceName*, right-click it, and click **Bring Online**.
21. Repeat the steps for a second DB2 agent on the second server instance.

Verifying that DB2 agents work in a cluster

To verify that the DB2 agent is working correctly, open the Tivoli Enterprise Portal and verify that the correct virtual server instances are being monitored for the correct node. From the Cluster Administrator, select the group representing the instance being monitored, and move it by using the following steps:

1. Click **Start** > **Administrative Tools** > **Cluster Manager**.
2. Expand **Groups** and select the correct group: **DB2 Group 0**.
3. Right-click, and then click **Move Group**.

   All the resource states shown for this group in the right pane cycle through Offline, Online Pending, and Online. After Online, the Owner shows the second Node.
4. Return to the Tivoli Enterprise Portal, and verify that the data for the DB2 instance is now being collected from the second node.

Setting up the DB2 agent in a High Availability Cluster Multi-Processing (HACMP) environment

Before installing the DB2 agent, install the Monitoring Agent for UNIX OS to all nodes in the IBM HACMP™ cluster. For the DB2 agent to run on each node in the cluster, some components of the infrastructure must be installed on each node. Installing the Monitoring Agent for UNIX OS is the easiest way to install the required infrastructure components. With the infrastructure available on all nodes, the DB2 agent can be installed to the shared disk and started on any node in the cluster.

On UNIX systems, monitoring agents can be installed in different directory trees. However, to perform the remote operations, install the DB2 agent in the same directory tree as the Monitoring Agent for UNIX OS.

Make sure that DB2 database is installed in the HACMP cluster and has an instance successfully failing over with the shared disk. Identify the node on which the DB2 instance is active, and make sure that the node has access to the DB2 shared disk. Do the following steps:

1. Mount the installation medium that you choose on that system.
2. Run install.sh, and change the default installation location from `/opt/IBM/ITM` to a location on the shared disk (for example, /ha_disk1/ITM).
3. Follow the installatiom steps.
4. After installation, open the *`install_dir`*`/config/ud.ini` file.
5. Set the value of the CTIRA_HOSTNAME variable to the cluster name (CTIRA_HOSTNAME=clusterName).
6. Make sure the value of the CTIRA_NODETYPE variable is set to UD (CTIRA_NODETYPE=UD).

   Important: Updating the CTIRA_NODETYPE variable for the DB2 agent on a UNIX system changes the managed system name. If you change the variable value, any custom situations or polices used by the DB2 agent must be recreated.

All instances of the DB2 agent are displayed in the Navigator view in the Tivoli Enterprise Portal under UNIX Systems with the clusterName specified in CTIRA_HOSTNAME.

To integrate the DB2 agent startup and shutdown into HACMP, add the commands required to start and stop the DB2 agent to the DB2 HACMP Application Server startup and shutdown scripts. In the startup script, start the DB2 instance before starting the DB2 agent. In the shutdown script, stop the DB2 agent before stopping the DB2 instance.

The following example contains a start script where the shared disk name is hadisk1:

```
# Start a DB2 admin server
su - dasusr1 -c '/hadisk1/dasusr1/das/bin/db2admin start'
# Start the DB2 instance, and finally ITM
/usr/bin/rc.db2pe db2inst1 start
# Start the ITM Monitoring Agent for DB2
su - db2inst1 -c '/hadisk1/ITM/bin/itmcmd agent -o db2inst1 start ud'
```

The following example contains a stop script where the shared disk name is hadisk1:

```
# Stop the ITM Monitoring Agent for DB2
su - db2inst1 -c '/hadisk1/ITM/bin/itmcmd agent -o db2inst1 stop ud'
# Stop the DB2 instance
/usr/bin/rc.db2pe db2inst1 stop
# Stop the DB2 admin server
su - dasusr1 -c '/hadisk1/dasusr1/das/bin/db2admin stop'
```

Setting up the Tivoli Enterprise Portal Server for the DB2 agent

By default, the instances of the DB2 agent are displayed in the Tivoli Enterprise Portal Navigator view based on the host system hostname. To display the DB2 agent with a particular cluster name instead of a particular node host name, set up the Tivoli Enterprise Portal Server by doing the following steps:

Remember: Altering the portal variable described in the following procedure can cause existing instances of the DB2 agent to be displayed in the wrong location in the portal. For each DB2 agent for UNIX installation, ensure that the following entry is in the config/ud.ini file: CTIRA_NODETYPE=UD. With CTIRA_NODETYPE unset, the DB2 agent defaults to being displayed by the instance name instead of the host name. Updating the CTIRA_NODETYPE variable for the DB2 agent on a UNIX system changes the managed system name. If you change the variable value, any custom situations or polices used by the DB2 agent on a UNIX system must be recreated.

To set up the Tivoli Enterprise Portal Server, do the following steps:

1. To open the Manage Tivoli Enterprise Monitoring Service window, navigate to the *ITMinstall_dir*/bin directory, where *ITMinstall_dir* is the installation directory of IBM Tivoli Monitoring, and run the following command:
   ```
   ./itmcmd manage
   ```
2. Stop the Tivoli Enterprise Portal Server, if it is running.
3. Right-click the Tivoli Enterprise Portal Server, and Click **Advanced** > **Edit Variables**.
4. Click **Add**.
5. In the Add Environment Setting Override window, type KFW_TOPOLOGY_CLUSTER_LIST in the **Variable** field, and set the value to AFF_UNIVERSAL_DATABASE.
6. Start the Tivoli Enterprise Portal Server for the changes to take effect.

Chapter 3. ITCAM agent for DB2 data model and reports for Tivoli Common Reporting (Cognos)

The ITCAM for Applications Data Model and Reports are historical reports, reporting against raw and summarized data collected in Tivoli Data Warehouse. These reports are built to run with the ITCAM agent for DB2 version 7.1.0 release only. The following databases for Tivoli Data Warehouse are supported:

- DB2
- Oracle
- SQL Server

The data model and reports are primarily focused on the DB2 agent. The reports are a set of sample reports and try to guide users on how to customize their own reports on the data model. The reports can be administered and run on Tivoli Common Reporting V2.1. For more information about Tivoli Common Reporting see www.ibm.com/developerworks/spaces/tcr

Prerequisites

Before installing the Tivoli Common Reporting data model and reports for the DB2 agent ensure that the following conditions are met:

- The DB2 agent is running
- The warehouse proxy agent is running
- The summarization and pruning agent in running
- Historical data collection is turned on and collection of data is ongoing.
- Summarized tables and views are created and populated in the Tivoli Data Warehouse.

To run the sample reports on a yearly, quarterly, monthly, weekly, daily or hourly basis, summarization must be enabled for the following tables:

- KUD_DB2_Database00
- KUD_DB2_Tablespace
- KUD_DB2_Table
- KUD_DB2_Application00

In order to ensure that the required views are present, run the following query against the Tivoli Data Warehouse:

- DB2 :
  ```
  select distinct "TABNAME" from SYSCAT.TABLES where "TABSCHEMA" = 'ITMUSER'
  ```
- Oracle: Log in as ITMUSER and run
  ```
  select distinct "TABLE_NAME" from USER_TABLES
  ```
- MS SQL :
  ```
  select TABLE_NAME "VIEWNAME" from INFORMATION_SCHEMA.TABLES
  ```

The result sent should contain the following tables

```
KUD_DB2_Tablespace
KUD_DB2_Tablespace_DV
KUD_DB2_Tablespace_HV
KUD_DB2_Tablespace__MV
KUD_DB2_Tablespace__QV
```



```
KUD_DB2_Tablespace__WV
KUD_DB2_Tablespace__YV
KUD_DB2_Database00
KUD_DB2_Database00_DV
KUD_DB2_Database00_HV
KUD_DB2_Database00__MV
KUD_DB2_Database00__QV
KUD_DB2_Database00__WV
KUD_DB2_Database00__YV
KUD_DB2_Table
KUD_DB2_Table _DV
KUD_DB2_Table _HV
KUD_DB2_Table __MV
KUD_DB2_Table __QV
KUD_DB2_Table __WV
KUD_DB2_Table __YV
KUD_DB2_Application00
KUD_DB2_Application00 _DV
KUD_DB2_Application00 _HV
KUD_DB2_Application00 __MV
KUD_DB2_Application00 __QV
KUD_DB2_Application00 __WV
KUD_DB2_Application00 __YV,
```

Therefore the result set should contain the _HV, _DV,_WV,_MV,_QV and _YV tables of every base table. The DB2 Tivoli Common Reporting data model and reports use ITMUSER as the schema name by default. Make sure you are using ITMUSER as your schema name in the Tivoli Data Warehouse. If you are using any schema name other than ITMUSER, see the Cognos user guide for information about how to update the schema name in a data model package after installing the report.

Installation

- **Cognos**: Cognos uses ODBC to connect to the database. Hence it is important to first install a database client on the Tivoli Common Reporting server and have it connect to the Tivoli Data Warehouse.
- **DB2**: Make sure you have deployed a DB2 database client on the machine where the Cognos-based Tivoli Common Reporting engine is installed. The client should be of the same version as the database that Tivoli Data Warehouse is using. Create a client database connection entry by running the Configuration Assistant.
- **Oracle**: Make sure you have deployed the Oracle database client on the machine where the Cognos-based Tivoli Common Reporting engine is installed. The client should be of the same version as the database that Tivoli Data Warehouse is using. Connect the Oracle database client to the database server by running the Oracle Net Configuration Assistant and configuring the local net service name configuration.
- **MS SQL Server**: Make sure you have deployed the MS SQL database client on the machine where Cognos-based Tivoli Common Reporting engine is installed. Connect the MS SQL client to the database server by running the MS SQL Management Studio Express and configuring the local net service name configuration.

Note: Name the connection you have created as it is used in Tivoli Common Reporting by the report installer.

Do the following steps to install the required Tivoli Common Reporting component:

1. Unzip the <CD_IMAGE>\REPORTS\KUD\IBM_Tivoli_Composite_Application_Manager_Agent_for_DB2_v71.zip file.
2. Run the setup file for your platform:
   - Windows: `setup_windows.bat`
   - Linux: `setup_linux.bin`
   - Aix: `setup_aix.bin`
   - Solaris: `setup_solaris.bin`
3. Specify the path to the Tivoli Common Reporting component directory.
4. Select `IBM Tivoli Composite Application Manager Agent for DB2 Reports v7.1`.
5. Enter the Tivoli Common Reporting user name and password.
6. Enter the Cognos Data Source TDW user name, password along with the database type and database name. The Tivoli Common Reporting installer will create a data source on the Cognos server using this information. Cognos will connect to the database using this user name and password to query data.

   The user must have enough authority to query data in your Warehouse database. The information you provide in this panel must match the information provided for the previously defined database.

   This step lets you skip defining the required data sources. Select the Checkbox if you want to do so. Do not skip defining the data sources unless you already have them created.
7. Enter the database user name and password for the data script. The database credentials provided are used by the installer to run the db script to create shared dimensions tables and to populate the time dimensions table. The scripts are located at <INSTALLATION_PATH>\export\reports\cognos_reports\itcamfdb2\db_scripts.

   Note: For oracle the user should have system rights, such as:

   ```
   SYSTEM
   ```

   This step lets you skip running the required script. Select the Checkbox if you want to do so. Do not skip this step unless you have already created the tables.
8. Enter the following database credentials
   - The database type which can be either DB2, Oracle or MSSQL
   - The JDBC URL
   - The JDBC driver files
   - The JDBC driver class
9. Verify your selections and the information you provided on the Summary page and then install.

Running a Report

For instructions on how to run a report, see http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp?topic=/com.ibm.tivoli.tcr.doc_21/ttcr_working.html

Customizing or Creating Reports

Reports can be customized or created using the Cognos Report Studio which is a web based report editor. To edit a report click on the Edit icon next to a report and the report studio will launch in a separate window. Once edited, you can save the report and it will be stored in the same location.

Ad-hoc querying

1. Click Launch.
2. Click Query Studio.
3. Select the ITCAM agent for DB2 package.

Data will then be loaded. You can drag and drop the data to create reports. The ITCAM agent for DB2 data model is a star schema with dimensions or identifiers separated from facts or metrics. Metrics are measurable attributes (numeric) which can be aggregated by identifiers. The relationship between the metric tables is defined using two common identifiers which areresources and time. You can either see the raw metrics or summarized (daily and hourly) metrics.

Identifiers are used to link the metrics data across agents. The two primary identifiers used with the ITCAM agent for DB2 are resources and time. Resources have database names and instance names. Time identifiers include various attributes of time by which the metrics can be grouped like Date (04/14/2012), Standard Timestamp (04/14/2012 12:00 AM), Weekday (Friday), Month (August), Quarter (3), Year (2012) etc.

Sample Reports

The package is shipped with 3 sample reports, these reports provide a simple guide on how to customize reports based on the DB2 agent data model.

Table 6. DB2 Top Application By Deadlock

Name	DB2 Top Application By Deadlock
Description	This report shows the top *n* deadlocks on the application, *n* can be customized as an input parameter, and by default the report will show the top 5 applications.
Purpose	Helps to identify database performance problems related to deadlocks.

Table 6. DB2 Top Application By Deadlock (continued)

Parameters	**Date Range:** **Report Period:** The user can choose from a pre-defined date range such as Last Week, Current Month, Last 30 Days. The user can also enter a start and end date and time for the reporting period. **Start Date:** The user can choose a start date from a calendar and a start time from the time widget. Both date and time need to be provided. **End Date:** The user can choose an end date from a calendar and an end time from the time widget. Both date and time need to be provided. **Summarization Selection:** **Summarization Type:** The user can choose to summarize data in hourly, daily, weekly, monthly, quarterly and yearly intervals. **Top N:** *The user can provide an integer to filter the number of top applications that are displayed.***Instance:** *The user can select one or more database instances.*
Output	This report shows the deadlock held by applications in a bar chart and a table.
Usage	The report indicates which applications are over-utilized by a collection of systems. The report can be run hourly, daily, weekly, monthly, quarterly and yearly. The user can select different DB2 instances to generate the report.

Table 7. DB2 Top Databases By Average Used Size Report

Name	DB2 Top Databases By Average Used Size Report
Description	This report shows database space utilization over a period of time.
Purpose	Helps to identify trends of database space utilization.

Table 7. DB2 Top Databases By Average Used Size Report (continued)

Parameters	**Date Range:** **Report Period:** The user can choose from a pre-defined date range such as Last Week, Current Month, Last 30 Days. The user can also enter a start and end date and time for the reporting period. **Start Date:** The user can choose a start date from a calendar and a start time from the time widget. Both date and time need to be provided. **End Date:** The user can choose an end date from a calendar and an end time from the time widget. Both date and time need to be provided. **Summarization Selection:** **Summarization Type:** The user can choose to summarize data in hourly, daily, weekly, monthly, quarterly and yearly intervals. **Top N:** *The user can provide an integer to filter the number of top applications that are displayed.***Instance:** *The user can select one or more database instances.*
Output	A bar chart and a table, in the bar chart, the size of database space usage will be represented by a bar in the specified time range. The table will feature more detailed information.
Usage	The report indicates database space usage over a specified period. The report can be run hourly, daily, weekly, monthly, quarterly and yearly.

Table 8. DB2 Top Table By IO Rates Report

Name	DB2 Top Table By IO Rates Report
Description	This report shows the top *n* tables with the highest IO read and write rates.
Purpose	Help to identify which tables have potential performance issues due to high IO rates.

Table 8. DB2 Top Table By IO Rates Report (continued)

Parameters	**Date Range:** **Report Period:** The user can choose from a pre-defined date range such as Last Week, Current Month, Last 30 Days. The user can also enter a start and end date and time for the reporting period. **Start Date:** The user can choose a start date from a calendar and a start time from the time widget. Both date and time need to be provided. **End Date:** The user can choose an end date from a calendar and an end time from the time widget. Both date and time need to be provided. **Summarization Selection:** **Summarization Type:** The user can choose to summarize data in hourly, daily, weekly, monthly, quarterly and yearly intervals. **Top N:** *The user can provide an integer to filter the number of top applications that are displayed.***Instance:** *The user can select one or more database instances.*
Output	A bar chart and a table, in the bar chart, average read and write rates will be represented by bars in the specified time range. The table will feature more detailed information.
Usage	The report can be run hourly, daily, weekly, monthly, quarterly and yearly.

Debugging and Known Problems

Arithmetic Overflow Errors in Ad Hoc Querying: If you drag certain columns during an ad-hoc query and an arithmetic overflow error is returned, switch to the Limited Data or the No Data preview and add a Standard Timestamp to the query. Certain columns may average or sum up to a number that is bigger than the size that the database supports. Hence an SQL arithmetic overflow error is returned. If the data is set to an hourly timestamp or a daily timestamp or set to query to limit the data, the aggregated value is forced to be within the supported size.

No data is available in Ad Hoc Querying on querying two tables but data is displayed when the two tables are queried individually: This happens if there is no relationship defined between the two tables. Make sure all your ad hoc queries have at least one identifier.

Missing table or attribute: Make sure all the prerequisites are met and the warehouse is collecting historical data. If you turn on appropriate historical performance, you should be able to use the data models for the ITCAM agent for DB2. If you think you have all the required tables and you are still getting this error, it is possible that the ITCAM agent for DB2 is incompatible with the version used in the generic model. To check the query, open the report in Report Studio and then go to Tools -> Show Generated SQL/MDX. The queries in the report will show up. You can view the native SQL.

Missing page numbers: After specifying that a large number of records be returned by Tivoli Common Reporting, when you attempt to access the last page of the results directly the page number of that last page is displayed as "?". This is a known issue with Cognos 8.4.1 and no solution is available at this time.

Chapter 4. Workspaces reference

This chapter contains an overview of workspaces, references for detailed information about workspaces, and descriptions of the predefined workspaces included in this monitoring agent.

About workspaces

A workspace is the working area of the Tivoli Enterprise Portal application window. At the left of the workspace is a Navigator that you use to select the workspace you want to see.

As you select items in the Navigator, the workspace presents views pertinent to your selection. Each workspace has at least one view. Some workspace views have links to other workspaces. Every workspace has a set of properties associated with it.

This monitoring agent provides predefined workspaces. You cannot modify or delete the predefined workspaces, but you can create new workspaces by editing them and saving the changes with a different name.

More information about workspaces

For more information about creating, customizing, and working with workspaces, see the *IBM Tivoli Monitoring User's Guide*.

For a list of the predefined workspaces for this monitoring agent and a description of each workspace, refer to the Predefined workspaces section below and the information in that section for each individual workspace.

For additional information about workspaces for this monitoring agent, see the Appendix B, "DB2 agent Workspaces," on page 395.

Predefined workspaces

The DB2 agent provides predefined workspaces. Workspaces can be accessed either from the Navigator tree item or by selecting links within another workspace.

Each Navigator tree item has a default workspace. This is the workspace that is displayed by default when that Navigator tree item is selected. Other, non-default, workspaces can be associated with (and accessed from) a Navigator tree item. To access one of the non-default workspaces, right-click the Navigator tree item and select Workspace, and the list of all workspaces associated with that Navigator tree item is displayed with a check mark on the left of the currently displayed workspace.

Some predefined workspaces are not available from the Navigator tree item, but are accessed by selecting the link indicator next to a row of data in a view. Left-clicking a link indicator selects the default workspace associated with that link. Right-clicking a link indicator displays all linked workspaces that can be selected.

If there is only one instance of DB2 on a computer, the Navigator tree has one item for the DB2 workspaces: DB2 – instance. When you click this item, a default workspace is displayed. If multiple instances of DB2 exist on a computer, the Navigator tree has an item for DB2 with items for the instances under it. When you click the DB2 item, no workspace is displayed. When you click one of the instances, other workspaces are displayed.

In the list of predefined workspaces for the DB2 agent, the highest level bullet is the Navigator tree item. The first indention represents workspaces that are directly available from that Navigator tree item (by selecting the Navigator tree item, right-clicking, hovering over **Workspace**, and selecting the desired workspace). Subsequent indentations represent workspaces that are accessible using links on the workspace under which they are indented. For example:

- Navigator tree item
  - Workspaces available directly from the main Navigator tree item
    - Workspaces available from links on the preceding workspace

The DB2 agent provides the following predefined workspaces, which are organized by Navigator item:

- "DB2 UDB Overview workspace" on page 53
- "DB2 UDB Overview (Superseded) workspace" on page 54
- Application
  - "Application Identification workspace" on page 40
  - "Application I/O Activity workspace" on page 41
  - "Application Lock Activity workspace" on page 41
  - "Application Overview workspace" on page 41
  - "Application Package and Catalog Cache Activity workspace" on page 42
  - "Apply Program/Subscription workspace" on page 46
  - "Application SQL Activity workspace" on page 43
  - "Application SQL Statement Text workspace" on page 44
  - "Application Sort and Hash Join Activity workspace" on page 43
  - "Application Time Information workspace" on page 45
  - "Application Top 10 Summary workspace" on page 45
  - "Application Identification (Superseded) workspace" on page 40
  - "Application I/O Activity (Superseded) workspace" on page 41
  - "Application Lock Activity (Superseded) workspace" on page 41
  - "Application Overview (Superseded) workspace" on page 42
  - "Application Package and Catalog Cache Activity (Superseded) workspace" on page 42
  - "Application SQL Activity (Superseded) workspace" on page 43
  - "Application SQL Statement Text (Superseded) workspace" on page 44
  - "Application Sort and Hash Join Activity (Superseded) workspace" on page 43
  - "Application Summary (Superseded) workspace" on page 44
  - "Application Top 10 Summary (Superseded) workspace" on page 45
  - "Application Time Information (Superseded) workspace" on page 45
- Buffer Pool Activity
  - "Buffer Pool workspace" on page 46

- “Buffer Pool Detail workspace” on page 47
  - “Buffer Pool (Superseded) workspace” on page 46
  - “Buffer Pool Detail (Superseded) workspace” on page 47
- Customized SQLs
  - “Customized SQLs workspace” on page 48
  - “Customized SQL Result workspace” on page 48
- Database
  - “Database Bottom 10 Summary by BP Hit Ratio workspace” on page 49
  - “Database Identification/Status information workspace” on page 50
  - “Database I/O Activity workspace” on page 50
  - “Database Lock Activity workspace” on page 51
  - “Database Pkg/Cat Cache workspace” on page 51
  - “Database SQL Activity workspace” on page 52
  - “Database Sort/Hash Join Activity workspace” on page 52
  - “Database workspace” on page 48
  - “Database Table workspace” on page 53
  - “DCS Database workspace” on page 54
  - “Database Bottom 10 Summary by BP Hit Ratio (Superseded) workspace” on page 49
  - “Database HADR workspace” on page 49
  - “Agent Event workspace” on page 40
  - “Database Identification/Status/Logging information (Superseded) workspace” on page 50
  - “Database Lock Activity (Superseded) workspace” on page 51
  - “Database Pkg/Cat Cache workspace” on page 51
  - “Database Pkg/Cat Cache (Superseded) workspace” on page 52
  - “Database SQL Activity (Superseded) workspace” on page 53
  - “Database Sort/Hash Join Activity (Superseded) workspace” on page 52
  - “Database Summary by BP Hit Ratio (Superseded) workspace” on page 53
  - “Database I/O Activity (Superseded) workspace” on page 50
  - “Database (Superseded) workspace” on page 49
- Locking Conflict
  - “Locking Conflict (Superseded) workspace” on page 55
  - “Locking Conflict workspace” on page 55
- Log Manager
  -
    - “Database Logging Overview workspace” on page 51
    - “Archive Log History workspace” on page 46
    - “Diagnostic Messages workspace” on page 54
- System Overview
  - “Connection workspace” on page 47
  - “General Information workspace” on page 54
  - “System Overview workspace” on page 56
  - “System Resources workspace” on page 57
  - “Connection (Superseded) workspace” on page 48

- "General Information (Superseded) workspace" on page 55
  - "System Overview (Superseded) workspace" on page 56
- Tablespace
  - "Tablespace (Superseded) workspace" on page 57
  - "Tablespace workspace" on page 57

The remaining sections of this chapter contain descriptions of each of these predefined workspaces. The workspace descriptions are organized by the Navigator item to which the workspaces are relevant.

Agent Event workspace

The Agent Event workspace is a predefined workspace that provides detailed information about predefined triggered events that can help you determine problems with the monitored database. You can view details such as the database name, timestamps, suggestions, event levels (Error, Warning, Info, Misc) and categories for the following events:

- **Attach Instance Failure event**: This event is triggered when the agent fails to attach to a DB2 instance.
- **Non DB2 Admin event**: This event is triggered when the user ID that is used to run the DB2 agent does not have DB2 SYSADM authority.
- **Standby Server event**: This event is triggered when the monitored database is an HADR standby database and you issue SQL commands to that database.

This workspace supports the display of large integers.

Application Identification workspace

The Application Identification workspace is a predefined workspace that provides information about status and space usage for the database. You can view the following topics:

- Application status and identifiers
- Client information
- Buffer pool/catalog cache hit ratio
- SQL activity
- Sort/hash join overflows

This workspace supports the display of large integers.

Application Identification (Superseded) workspace

The Application Identification workspace is a predefined workspace that provides information about status and space usage for the database. You can view the following topics:

- Application status and identifiers
- Client information
- Buffer pool/catalog cache hit ratio
- SQL activity
- Sort/hash join overflows

This workspace is superseded. There is a workspace with the same name that replaces it.

Application I/O Activity workspace

The Application I/O Activity workspace is a predefined workspace that provides information about various I/O activities. You can view the following topics:

- Buffer pool activity
- Direct I/O activity
- Extended store activity

This workspace supports the display of large integers.

Application I/O Activity (Superseded) workspace

The Application I/O Activity workspace is a predefined workspace that provides information about various I/O activities. You can view the following topics:

- Application identification and status
- Buffer pool activity
- Direct I/O activity
- Extended store activity

This workspace is superseded. There is a workspace with the same name that replaces it.

Application Lock Activity workspace

The Application Lock Activity workspace is a predefined workspace that provides information about various lock activities. You can view the following topics:

- Lock activity
- Lock information, such as average lock wait time, the number of the held locks
- Information about lock waiting, such as lock mode, lock wait start time
- IDs of the applications and agents that hold locks

This workspace supports the display of large integers.

Application Lock Activity (Superseded) workspace

The Application Lock Activity workspace is a predefined workspace that provides information about various lock activities. You can view the following topics:

- Lock activity
- Application identification and status data such as agent ID, application status, application name, snapshot time, and database name
- Lock information, such as average lock wait time, the number of the held locks
- Information about lock waiting, such as lock mode, lock wait start time
- IDs of the applications and agents that hold locks

This workspace is superseded. There is a workspace with the same name that replaces it.

Application Overview workspace

The Application Overview workspace is a predefined workspace that provides information about various application-related activities. You can view the applications that have the following characteristics:

- Highest number of deadlocks
- Highest number of lock timeouts

- Highest number of sort overflows
- Highest percentage of failed SQL statements
- Lowest buffer pool hit ratio

This workspace supports the display of large integers.

Application Overview (Superseded) workspace

The Application Overview workspace is a predefined workspace that provides information about various application-related activities. You can view the applications that have the following characteristics:

- Highest number of deadlocks
- Highest number of lock timeouts
- Highest number of sort overflows
- Highest percentage of failed SQL statements
- Lowest buffer pool hit ratio

This workspace is superseded. There is a workspace with the same name that replaces it.

Application Package and Catalog Cache Activity workspace

The Application Package and Catalog Cache Activity workspace is a predefined workspace that provides information about various package and catalog cache activities. You can view the following topics:

- Package/catalog cache hit ratio
- Catalog cache overflows
- Application package/catalog cache activities
  - Package cache lookups
  - Package cache inserts
  - Catalog cache lookups
  - Catalog cache inserts
  - Catalog cache heap full

This workspace supports the display of large integers.

Application Package and Catalog Cache Activity (Superseded) workspace

The Application Package and Catalog Cache Activity workspace is a predefined workspace that provides information about various package and catalog cache activities. You can view the following topics:

- Package/catalog cache hit ratio
- Catalog cache overflows
- Application identification and status data such as agent ID, application status, application name, snapshot time, and database name
- Application package/catalog cache activities
  - Package cache lookups
  - Package cache inserts
  - Catalog cache lookups
  - Catalog cache inserts

- Catalog cache heap full

This workspace is superseded. There is a workspace with the same name that replaces it.

Application Sort and Hash Join Activity workspace

The Application Sort and Hash Join Activity workspace is a predefined workspace that provides information about sort and hash join activities. You can view the following topics:

- Hash join overflows
- Sort overflows
- Total hash joins, total hash loops, hash join overflows
- Total sorts, total sort time, average sort time, sort overflows percent

This workspace supports the display of large integers.

Application Sort and Hash Join Activity (Superseded) workspace

The Application Sort and Hash Join Activity workspace is a predefined workspace that provides information about sort and hash join activities. You can view the following topics:

- Hash join overflows
- Sort overflows
- Total hash joins, total hash loops, hash join overflows
- Total sorts, total sort time, average sort time, sort overflows percent
- Application identification and status data such as agent ID, application status, application name, snapshot time, and database name

This workspace is superseded. There is a workspace with the same name that replaces it.

Application SQL Activity workspace

The Application SQL Activity workspace is a predefined workspace that provides information about various SQL activities. You can view such topics as

- Application SQL activity
- Statement and cursor information
- Internal and row counts

This workspace supports the display of large integers.

Application SQL Activity (Superseded) workspace

The Application SQL Activity workspace is a predefined workspace that provides information about various SQL activities. You can view such topics as

- Application SQL activity
- Application identification and status data
- Statement and cursor information
- SQL statement counts
- Internal and row counts

This workspace is superseded. There is a workspace with the same name that replaces it.

Application SQL Statement Text workspace

The Application SQL Statement Text workspace is a predefined workspace that provides the text of SQL statements. You can view the following topics:

- SQL activity such as total SQL statements, failed SQL statements, internal rollbacks, internal deadlock rollbacks, and rollback SQL statements
- Statement and cursor information
- SQL statement text

This workspace supports the display of large integers.

Important: The statement text can only be fetched during the time when the dynamic SQL query is executing. If the query executing time is short, the statement text might not be retrieved.

Application SQL Statement Text (Superseded) workspace

The Application SQL Statement Text workspace is a predefined workspace that provides the text of SQL statements. You can view the following topics:

- SQL activity such as total SQL statements, failed SQL statements, internal rollbacks, internal deadlock rollbacks, and rollback SQL statements
- Application identification and status data such as agent ID, application status, application name, snapshot time, and database name
- Statement and cursor information
- SQL statement text

Important: The statement text can only be fetched during the time when the dynamic SQL query is executing. If the query executing time is short, the statement text might not be retrieved.

This workspace is superseded. There is a workspace with the same name that replaces it.

Application Summary (Superseded) workspace

The Application Summary workspace is a predefined workspace that provides information about various application activities. You can view the following topics:

- A summary (by agent ID) of the following application data:
  - Application name
  - Application status
  - Deadlocks
  - Lock waits
  - Locks held
  - Pool hit ratio
  - Snapshot time
  - Statement operation
- Ten applications with the highest number of failed SQL statements
- Ten applications with the lowest buffer pool hit ratios

Application Time Information workspace

The Application Time Information workspace is a predefined workspace that provides information about various time-related activities. You can view the following topics:

- Application identification and status
- Application time such as application idle time, amount of CPU time spent in system calls, amount of CPU time spent in executing database manager code, and application connect time
- Buffer pool and direct I/O time
- Lock wait time

This workspace supports the display of large integers.

Application Time Information (Superseded) workspace

The Application Time Information workspace is a predefined workspace that provides information about various time-related activities. You can view the following topics:

- Application identification and status
- Application time such as application idle time, amount of CPU time spent in system calls, amount of CPU time spent in executing database manager code, and application connect time
- Buffer pool and direct I/O time
- Lock wait time

This workspace is superseded. There is a workspace with the same name that replaces it.

Application Top 10 Summary workspace

The Application Top 10 Summary workspace is a predefined workspace that provides information about various application-related activities. You can view the following topics:

- A summary (by ageng ID) of the following application data:
  - Application name
  - Application status
  - Deadlocks
  - Lock waits
  - Locks held
  - Pool hit ratio
  - Snapshot time
  - Statement operation
- Ten applications with the highest percentage of failed SQL statements
- Ten applications with the lowest buffer pool hit ratio

This workspace supports the display of large integers.

Application Top 10 Summary (Superseded) workspace

The Application Top 10 Summary workspace is a predefined workspace that provides information about various application-related activities. You can view the following topics:

- A summary (by ageng ID) of the following application data:
  - Application name
  - Application status
  - Deadlocks
  - Lock waits
  - Locks held
  - Pool hit ratio
  - Snapshot time
  - Statement operation
- Ten applications with the highest percentage of failed SQL statements
- Ten applications with the lowest buffer pool hit ratio

This workspace is superseded. There is a workspace with the same name that replaces it.

Apply Program/Subscription workspace

The Apply Program/Subscription workspace is a predefined workspace that provides information about Apply Program processes and Apply Program subscription sets that are configured to run on the database manager server. You can view the following topics:

- The number of subscriptions that fail to replicate or to complete
- The number of subscriptions the Apply Program failed to replicate because refresh copying was disabled
- Apply application information
- Apply subscription information

Archive Log History workspace

This workspace is only available for DB2 Version 9.1 and later.

The Archive Log History workspace is only a selectable workspace as the target of Database Logging Overview workspace link. It provides historical information about archived logs. This workspace consists of the following view:

- Archive Log Record

Buffer Pool workspace

The Buffer Pool workspace is a predefined workspace that provides information about various buffer pool activities. You can view the following topics:

- Buffer pool hit ratio by individual buffer pool
- Summary data such as buffer pool name, database name, pool total reads, average pool read time, and average pool write time
- Ten databases with the lowest buffer pool hit ratios

This workspace supports the display of large integers.

Buffer Pool (Superseded) workspace

The Buffer Pool workspace is a predefined workspace that provides information about various buffer pool activities. You can view the following topics:

- Buffer pool hit ratio by individual buffer pool

- Summary data such as buffer pool name, database name, pool total reads, average pool read time, and average pool write time
- Ten databases with the lowest buffer pool hit ratios

This workspace is superseded. There is a workspace with the same name that replaces it.

Buffer Pool Detail workspace

The Buffer Pool Detail workspace is a predefined workspace that provides information about various buffer pool activities. You can view the following topics:

- Asynchronous and synchronous I/O activity attributes such as pool asynchronous data reads, pool asynchronous data writes, pool sync data reads, pool sync data writes
- Average read and write times
- Databases with the lowest buffer pool hit ratios
- Extended store and non-buffer-pool I/O activity such as pool data to estore, pool index to estore, direct reads, and direct writes
- Various buffer pool activities such as data logical reads, data physical reads, index logical reads, index physical reads, pool total reads

This workspace supports the display of large integers.

Buffer Pool Detail (Superseded) workspace

The Buffer Pool Detail workspace is a predefined workspace that provides information about various buffer pool activities. You can view the following topics:

- Asynchronous and synchronous I/O activity attributes such as pool asynchronous data reads, pool asynchronous data writes, pool sync data reads, pool sync data writes
- Average read and write times
- Buffer pool identification data
- Databases with the lowest buffer pool hit ratios
- Extended store and non-buffer-pool I/O activity such as pool data to estore, pool index to estore, direct reads, and direct writes
- Various buffer pool activities such as data logical reads, data physical reads, index logical reads, index physical reads, pool total reads

This workspace is superseded. There is a workspace with the same name that replaces it.

Connection workspace

The Connection workspace is a predefined workspace that provides information about various server-related activities. You can view the following topics:

- Server connection data such as remote connections initiated, local connections, number of local databases with current connections
- Agent information such as agents waiting on token, agents stolen, and idle agents

This workspace supports the display of large integers.

Connection (Superseded) workspace

The Connection workspace is a predefined workspace that provides information about various server-related activities. You can view the following topics:

- Server connection data such as remote connections initiated, local connections, number of local databases with current connections
- Agent information such as agents waiting on token, agents stolen, and idle agents

This workspace is superseded. There is a workspace with the same name that replaces it.

Customized SQLs workspace

The Customized SQLs workspace is a predefined workspace that provides the latest execution status information and definition details for customized SQL statements defined by users. This workspace contains the following views:

- Customized SQL Status:
  - Provides the latest execution status information for the defined SQL statements, including details such as the last execution error code.
- Customized SQL Definition:
  - Provides the definition details of the customized SQL statements, including information such as the name of the definition file.

Customized SQL Result workspace

The Customized SQL Result workspace is a predefined link to workspace that provides the results of the selected customized SQL statements. You can access the Customized SQL Result workspace by right-clicking a row in the Customized SQLs workspace and then linking to the Customized SQL Result workspace. The information displayed is based on the provided database alias and the definition of the SQL statement. This workspace contains the following views:

- Customized SQL Definition:
  - Provides the definition details of the customized SQL statements, including information such as the name of the definition file.
- Customized SQL Detail:
  - Provides the result of executing the SQL statement including 5 named int type columns and their respective values, 5 named string type columns and values, and their respective values and datetime type columns.

Database workspace

The Database workspace is a predefined workspace that provides information about status and space usage from a database perspective. You can view databases that have the following characteristics:

- Highest number of connections
- Highest number of deadlock situations
- Highest number of failed SQL statements
- Highest number of lock timeouts
- Lowest buffer pool hit ratio

This workspace supports the display of large integers.

Database (Superseded) workspace

The Database workspace is a predefined workspace that provides information about status and space usage from a database perspective. You can view databases that have the following characteristics:

- Highest number of connections
- Highest number of deadlock situations
- Highest number of failed SQL statements
- Highest number of lock timeouts
- Lowest buffer pool hit ratio

This workspace is superseded. There is a workspace with the same name that replaces it.

Database Bottom 10 Summary by BP Hit Ratio workspace

The Database Bottom 10 Summary by BP Hit Ratio Workspace is a predefined workspace that provides information about various database-related activities. The workspace provides information about a maximum of 10 databases. You can view databases that have the following characteristics:

- Highest number of connections
- Highest number of failed SQL statements
- Lowest buffer pool hit ratio
- Lowest buffer pool hit ratio (summary)

This workspace supports the display of large integers.

Database Bottom 10 Summary by BP Hit Ratio (Superseded) workspace

The Database Bottom 10 Summary by BP Hit Ratio Workspace is a predefined workspace that provides information about various database-related activities. The workspace provides information about a maximum of 10 databases. You can view databases that have the following characteristics:

- Highest number of connections
- Highest number of failed SQL statements
- Lowest buffer pool hit ratio
- Lowest buffer pool hit ratio (summary)

This workspace is superseded. There is a workspace with the same name that replaces it.

Database HADR workspace

The Database HADR workspace is a predefined workspace that provides information about the configuration and status of High Availability Disaster Recovery (HADR). You can view the following topics:

- HADR configuration information, such as the remote and local host of the HADR
- HADR runtime information, such as the connection status of the HADR

Database I/O Activity workspace

The Database I/O Activity workspace is a predefined workspace that provides information about various database-related I/O activities. You can view the following topics:

- Buffer pool hit ratio for the monitored database
- Counts of various buffer pool activities such as pool read time, pool data logical reads, pool index physical reads, files closed
- Direct I/O activities such as number of: direct reads, direct writes, direct read time, direct write time
- Number of buffer pool reads and writes to the monitored database

This workspace supports the display of large integers.

Database I/O Activity (Superseded) workspace

The Database I/O Activity workspace is a predefined workspace that provides information about various database-related I/O activities. You can view the following topics:

- Database identification and status
- Buffer pool hit ratio for the monitored database
- Counts of various buffer pool activities such as pool read time, pool data logical reads, pool index physical reads, files closed
- Direct I/O activities such as number of: direct reads, direct writes, direct read time, direct write time
- Number of buffer pool reads and writes to the monitored database

This workspace is superseded. There is a workspace with the same name that replaces it.

Database Identification/Status information workspace

The Database Identification/Status information workspace is a predefined workspace that provides information about various database identification, status, and connection information. You can view the following topics:

- Connection information
- Database highest agent information
- Database identification and status information

This workspace supports the display of large integers.

Database Identification/Status/Logging information (Superseded) workspace

The Database Identification/Status/Logging workspace is a predefined workspace that provides information about various database identification, status, and logging activities. You can view the following topics:

- Connection information
- Database logging activity
- Database highest agent information
- Database identification and status information

This workspace is superseded, and replaced by the Database Identification/Status information workspace.

Database Lock Activity workspace

The Database Lock Activity workspace is a predefined workspace that provides information about various locking activities from a database perspective. You can view the following topics:

- Lock timeouts and deadlocks
- The number of held locks, deadlocks, lock escalations, and lock timeouts; average lock time; lock wait time
- SQL activity such as the number of: internal rollbacks, internal deadlock rollbacks, failed SQL statements

This workspace supports the display of large integers.

Database Lock Activity (Superseded) workspace

The Database Lock Activity workspace is a predefined workspace that provides information about various locking activities from a database perspective. You can view the following topics:

- Database identification and status data such as database path, database name, database connect time, last backup
- Lock timeouts and deadlocks
- The number of held locks, deadlocks, lock escalations, and lock timeouts; average lock time; lock wait time
- SQL activity such as the number of: internal rollbacks, internal deadlock rollbacks, failed SQL statements

This workspace is superseded. There is a workspace with the same name that replaces it.

Database Logging Overview workspace

The Database Logging Overview workspace is a predefined workspace that provides information about active logs and archived logs. You can view the following topics:

- Usage information for active logs
- Disk space usage for the logging directory
- Active log related information including the configuration and dynamic monitoring elements
- Archived log related information, including the configuration and dynamic monitoring elements

Database Pkg/Cat Cache workspace

The Database Pkg/Cat Cache workspace is a predefined workspace that provides information about various package and catalog cache activities. You can view the following topics:

- Catalog cache activity such as catalog cache lookups
- Catalog cache overflows and catalog cache heap full
- Package and catalog cache hit ratio
- Package cache activity such as package cache lookups

This workspace supports the display of large integers.

Database Pkg/Cat Cache (Superseded) workspace

The Database Pkg/Cat Cache workspace is a predefined workspace that provides information about various package and catalog cache activities. You can view the following topics:

- Catalog cache activity such as catalog cache lookups
- Catalog cache overflows and catalog cache heap full
- Package and catalog cache hit ratio
- Package cache activity such as package cache lookups

This workspace is superseded. There is a workspace with the same name that replaces it.

Database Sort/Hash Join Activity workspace

The Database Sort/Hash Join Activity workspace is a predefined workspace that provides information about various sort and hash join activities. You can view the following topics:

- Sort heap allocated, number of active sorts, average sort time
- Total number of hash joins and hash join overflows
- Total number of hash loops and hash join small overflows
- Total number of sorts and sort overflows

This workspace supports the display of large integers.

Database Sort/Hash Join Activity (Superseded) workspace

The Database Sort/Hash Join Activity workspace is a predefined workspace that provides information about various sort and hash join activities. You can view the following topics:

- Sort heap allocated, number of active sorts, average sort time
- Total number of hash joins and hash join overflows
- Total number of hash loops and hash join small overflows
- Total number of sorts and sort overflows

This workspace is superseded. There is a workspace with the same name that replaces it.

Database SQL Activity workspace

The Database SQL Activity workspace is a predefined workspace that provides information about various SQL activities from a database perspective. You can view the following topics:

- Number of dynamic SQL statements
- Number of internal rollbacks, internal deadlock rollbacks, failed SQL statements
- Number of SQL COMMIT statements
- Number of static SQL statements
- Row counts
- SQL activity
- SQL statement data
- Total number of SQL statements

This workspace supports the display of large integers.

Database SQL Activity (Superseded) workspace

The Database SQL Activity workspace is a predefined workspace that provides information about various SQL activities from a database perspective. You can view the following topics:

- Database identification and status
- Number of dynamic SQL statements
- Number of internal rollbacks, internal deadlock rollbacks, failed SQL statements
- Number of SQL COMMIT statements
- Number of static SQL statements
- Row counts
- SQL activity
- SQL statement data
- Total number of SQL statements

This workspace is superseded. There is a workspace with the same name that replaces it.

Database Summary by BP Hit Ratio (Superseded) workspace

The Database Summary by BP Hit Ratio workspace is a predefined workspace that provides information about various database-related activities. The workspace provides information about a maximum of 10 databases. You can view the databases with the following characteristics:

- Highest number of connections
- Highest number of failed SQL statements
- Lowest buffer pool hit ratio
- Lowest buffer pool hit ratio (summary)

This workspace is superseded. There is a workspace with the same name that replaces it.

Database Table workspace

The Database Table workspace is a predefined workspace that provides information about tables of the monitored database instance. You can view the following topics:

- Five tables with the highest row read rate
- Five tables with the highest row write rate
- Information of the tables in the database, such as table name and table schema

Note: Only active tables are displayed in this workspace.

DB2 UDB Overview workspace

The DB2 UDB Overview workspace is a predefined workspace that provides information about server instance. You can view the following topics:

- Server status, including instance name, version, and server type.
- Locking conflict information
- Number of server key events, including post threshold sorts, agents waiting on token, maximum agent overflows, agent stolen, and post threshold hash joins.
- Number of server connections
- Ten lowest buffer pool hit ratios

- Ten highest failed SQL statements

This workspace supports the display of large integers.

DB2 UDB Overview (Superseded) workspace

The DB2 UDB Overview workspace is a predefined workspace that provides information about server instance. You can view the following topics:

- Server status, including instance name, version, and server type.
- Locking conflict information
- Number of server key events, including post threshold sorts, agents waiting on token, maximum agent overflows, agent stolen, and post threshold hash joins.
- Number of server connections
- Ten lowest buffer pool hit ratios
- Ten highest failed SQL statements

This workspace is superseded. There is a workspace with the same name that replaces it.

DCS Database workspace

The DCS Database workspace is a predefined workspace that provides information about DCS database. You can view the following topics:

- Communication errors
- Host response time
- DCS database statistics

Diagnostic Messages workspace

The Diagnostics Messages workspace is a predefined workspace that provides information about DB2 diagnostic messages. You can view the following topics:

- Administration notification log information
- Event log information
- Diagnostic log information

Note: Currently the agent supports messages which comply with following format in the db2diag.log file:

```
2012-04-27-04.57.54.466140+480 E1168573A668       LEVEL: Error
PID     : 23003166             TID  : 1           PROC : db2agent (idle) 0
INSTANCE: db2inst1             NODE : 000
APPID   : *LOCAL.db2inst1.120426205304
AUTHID  : DB2INST1
FUNCTION: DB2 UDB, oper system services, sqloEDUSIGDANGERHandler,
 probe:40
MESSAGE : ADM0505E  DB2 received a SIGDANGER signal
from the operating system.
This signal indicates that the system is running low on paging space.
If the paging space gets too low, the operating system will forcibly
terminate user processes.  Contact your system administrator to
increase your paging space.
```

General Information workspace

The General Information workspace is a predefined workspace that provides information about various server activities. You can view the following topics:

- Key events of a server, such as post threshold sorts, agents waiting on a token, agents stolen, and the number of attempts to exceed the maximum number of agents
- Data about sort/hash join, such as sort heap allocated, piped sorts requested, and piped sorts accepted
- Data about the status of a DB2 instance, such as start time, last reset, snapshot time, and instance name
- Server connection data such as remote connections initiated, local connections, and number of local databases with current connections

This workspace supports the display of large integers.

General Information (Superseded) workspace

The General Information workspace is a predefined workspace that provides information about various server activities. You can view the following topics:

- Key events of a server, such as post threshold sorts, agents waiting on a token, agents stolen, and the number of attempts to exceed the maximum number of agents
- Data about sort/hash join, such as sort heap allocated, piped sorts requested, and piped sorts accepted
- Data about the status of a DB2 instance, such as start time, last reset, snapshot time, and instance name
- Server connection data such as remote connections initiated, local connections, and number of local databases with current connections

This workspace is superseded. There is a workspace with the same name that replaces it.

Locking Conflict workspace

The Locking Conflict workspace is a predefined workspace that provides information about various locking activities and resources. You can view the following topics:

- Ten applications with the highest number of lock timeouts
- Lock wait time for each application
- Specific information about lock conflicts in the following areas:
  - Application identification
  - Application name
  - Client database alias
  - Lock wait time
  - Locks held
  - Snapshot time

This workspace supports the display of large integers.

Locking Conflict (Superseded) workspace

The Locking Conflict workspace is a predefined workspace that provides information about various locking activities and resources. You can view the following topics:

- Ten applications with the highest number of lock timeouts
- Lock wait time for each application

- Specific information about lock conflicts in the following areas:
  - Application identification
  - Application name
  - Client database alias
  - Lock wait time
  - Locks held
  - Snapshot time

This workspace is superseded. There is a workspace with the same name that replaces it.

System Overview workspace

Important: The values of the following attributes are only available when the monitored database is multi-partitioned, or when the monitored database is single-partitioned and the intra-partition parallelism is enabled by setting the intra_parallel parameter to YES.

- Buff Max Used Percent
- Buff Used Percent
- CE Max Used Percent
- CE Used Percent
- MA Max Used Percent
- RB Max Used Percent
- RB Used Percent

The System Overview workspace is a predefined workspace that provides information about various server-related activities. You can view the following topics:

- Locking conflict information such as agent ID, application name, and application ID
- Server connection data such as remote connections initiated, local connections, and number of local databases with current connections
- Server key events such as post threshold sorts, agents waiting on a token, agents stolen, and the number of attempts to exceed the maximum number of agents
- Status of the DB2 UDB instance
- Ten applications with the highest percentage of failed SQL statements
- Ten databases with the lowest buffer pool hit ratio

This workspace supports the display of large integers.

System Overview (Superseded) workspace

The System Overview workspace is a predefined workspace that provides information about various server-related activities. You can view the following topics:

- Locking conflict information such as agent ID, application name, and application ID
- Server connection data such as remote connections initiated, local connections, and number of local databases with current connections
- Server key events such as post threshold sorts, agents waiting on a token, agents stolen, and the number of attempts to exceed the maximum number of agents

- Status of the DB2 UDB instance
- Ten applications with the highest percentage of failed SQL statements
- Ten databases with the lowest buffer pool hit ratio

This workspace is superseded. There is a workspace with the same name that replaces it.

System Resources workspace

The System Resources workspace is a predefined workspace that provides information about the resources of the system in which the monitored DB2 instance is running. You can view the following topics:

- Network connection
- System resources, this table view is only available for DB2 Version 9.5 and later
- The usage of the CPU, physical memory, virtual memory, and swap memory of the system, these four views are only available for DB2 Version 9.5 and later

Tablespace workspace

The Tablespace workspace is a predefined workspace that provides tablespace information at the database level. You can use this information to monitor page size and usage characteristics. You can view the following information about paging statistics:

- Free pages
- Pending free pages
- Total pages
- Usable pages
- Used pages

This workspace supports the display of large integers.

Tablespace (Superseded) workspace

The Tablespace workspace is a predefined workspace that provides tablespace information at the database level. You can use this information to monitor page size and usage characteristics. You can view the following information about paging statistics:

- Free pages
- Pending free pages
- Total pages
- Usable pages
- Used pages

This workspace is superseded. There is a workspace with the same name that replaces it.

Chapter 5. Attributes reference

This chapter contains information about the following topics:
- Overview of attributes
- References for detailed information about attributes
- Descriptions of the attributes for each attribute group included in this monitoring agent
- Disk space requirements for historical data

About attributes

Attributes are the application properties being measured and reported by the DB2 agent, such as the amount of memory usage or the message ID. Some monitoring agents have fewer than 100 attributes, while others have over 1000.

Attributes are organized into groups according to their purpose. The attributes in a group can be used in the following two ways:
- Chart or table views

  Attributes are displayed in chart and table views. The chart and table views use queries to specify which attribute values to request from a monitoring agent. You use the Query editor to create a new query, modify an existing query, or apply filters and set styles to define the content and appearance of a view based on an existing query.
- Situations

  You use attributes to create situations that monitor the state of your operating system, database, or application. A situation describes a condition you want to test. When you start a situation, the Tivoli Enterprise Portal compares the values you have assigned to the situation attributes with the values collected by the DB2 agent and registers an *event* if the condition is met. You are alerted to events by indicator icons that are displayed in the Navigator.

Some of the attributes in this chapter are listed twice, with the second attribute having a "(Unicode)" designation after the attribute name. These Unicode attributes were created to provide access to globalized data.

More information about attributes

For more information about using attributes and attribute groups, see the *IBM Tivoli Monitoring User's Guide*.

For a list of the attributes groups, a list of the attributes in each attribute group, and descriptions of the attributes for this monitoring agent, refer to the Attribute groups and attributes section in this chapter.

Attribute groups and attributes for the DB2 agent

This monitoring agent contains the following attribute groups:
- Application00 (KUD_DB2_Application00
- Application00 (KUDDB2APPLGROUP00) (Superseded)
- Application00U (KUDDB2APPLGROUP00_U) (Superseded)

- Application01 (KUD_DB2_Application01)
- Application01 (KUDDB2APPLGROUP01) (Superseded)
- Apply Program (KUD_DB2_Apply_Program)
- Apply Subscription (KUD_DB2_Apply_Subscription)
- Buffer Pool (KUD_DB2_Buffer_Pool)
- Buffer Pool (KUDBUFFERPOOL00) (Superseded)
- Customized SQL Definition (KUD_Customized_SQL_Definition)
- Customized SQL Status (KUD_Customized_SQL_Status)
- Customized SQL Detail (KUD_Customized_SQL_Detail)
- Database00 (KUD_DB2_Database00)
- Database00 (KUDDBASEGROUP00) (Superseded)
- Database01 (KUD_DB2_Database01)
- Database01 (KUDDBASEGROUP01) (Superseded)
- Database02 (KUD_DB2_Database02)
- DB2 Agent Event (KUD_Agent_Event)
- DB2 HADR (KUD_DB2_HADR)
- DCS Database (KUD_DB2_DCS_Database)
- Diagnostic Log (KUD_DB2_Diagnostic_Log)
- Diagnostic Messages (KUD_DB2_Diagnostic_Messages) (Superseded)
- Locking Conflict (KUDLOCKCONFLICT00)
- Log (KUD_DB2_LOG)
- Log Record (KUD_DB2_LOG_RECORD)
- Network Info (KUD_DB2_IPADDR_TABLE)
- System Overview (KUD_DB2_System_Overview)
- System Overview (KUDINFO00) (Superseded)
- System Resources (KUD_DB2_System_Resources)
- Table (KUD_DB2_Table)
- Tablespace (KUD_DB2_Tablespace)
- Tablespace (KUDTABSPACE) (Superseded)
- DB2 Tablespace Auto-resize (KUD_Tablespace_Auto_Resize)

Because attribute groups have limits on the number of attributes that they can hold, multiple application groups exist for some types of data, such as multiple groups for application database information.

IBM Tivoli Monitoring provides other attribute groups that are available to all monitoring agents, for example Universal Time and Local Time. The attributes in these common attribute groups are documented in the Tivoli Enterprise Portal Help.

The following sections contain descriptions of the DB2 agent attribute groups, which are listed alphabetically. Each description contains a list of attributes in the attribute group.

Application00 (KUD_DB2_Application00) attributes

The Application00 attribute group provides information about the database and the application. By using this information, you can determine the efficiency of the database and identify any problem areas for corrective action. All values are integers that are calculated from the first application connection, unless otherwise noted.

Acc Curs Blk The number of times that a request for an I/O block was accepted. The value format is an integer. Use this attribute with the Rejected Block Cursor Requests attribute to calculate the percentage of blocking requests that are accepted or rejected. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Agent ID The application handle, which is a system-wide unique ID for the application. The value format is an integer. On multi-node systems, where a database is partitioned, this ID is the same on every node where the application might make a secondary connection.

Agent ID Holding Lock The application handle of the agent holding a lock for which this application is waiting. The value format is an integer. The lock monitor group must be turned on to obtain this information.

Appl Conn Timestamp The date and time that an application started a connection request.

Appl ID The identifier generated when the application connects to the database at the database manager or when DDCS receives a request to connect to a DRDA database. The value format is a text string, with a maximum of 32 characters.

Appl ID Holding Lock The application ID of the application that is holding a lock on the object that this application is waiting to obtain (Unicode). The value format is a text string with a maximum of 96 bytes.

Appl Idle Time The number of seconds since an application issued a request to the server. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Appl Name The name of the application running at the client as it is known to the database manager or DB2 Connect. The value format is a text string, with a maximum of 60 bytes. For example: `*Local.db2inst1.990212202018`.

Appl Status The status of the application being monitored. This attribute can help you diagnose potential application problems. The value format is a text string with a maximum of 64 characters. The following values are valid:

External value	Internal value
Backing Up Database	Backing_Up_Database
Commit Active	Commit_Active
Compiling SQL Stmt	Compiling_SQL_Stmt
Connect Pending	Connect_Pending
Connected	Connected

External value	Internal value
Creating Database	Creating_Database
Decoupled	Decoupled
Disconnect Pending	Disconnect_Pending
I/O Error Waiting	I/O_Error_Waiting
Loading Database	Loading_Database
Lock Waiting	Lock_Waiting
Prepared Transaction	Prepared_Transaction
Quiescing a Tablespace	Quiescing_a_Tablespace
Recompiling Plan	Recompiling_Plan
Request Interrupted	Request_Interrupted
Restarting Database	Restarting_Database
Restoring Database	Restoring_Database
Rollback Active	Rollback_Active
Trans. heuristically aborted	Trans._heuristically_aborted
Trans. heuristically committed	Trans._heuristically_committed
Transaction ended	Transaction_ended
UOW Executing	UOW Executing
UOW Waiting in the application	UOW Waiting in the application
Unloading Database	Unloading Database
UNKNOWN	UNKNOWN

Auth ID The authorization ID of the user who invoked the application that is being monitored. On a DB2 Connect gateway node, this ID is the user authorization ID on the host. The value format is a text string with a maximum of 60 bytes. Use this attribute to determine who invoked the application.

Avg Lock Wait Time The average elapsed time (in milliseconds) that was spent waiting for a lock. The value format is an integer. If the average lock wait time is high, you must look for applications that hold many locks, or have lock escalations, with a focus on tuning your applications to improve concurrency, if appropriate. If escalations are the reason for a high average lock wait time, the values of one or both of the LOCKLIST and MAXLOCKS configuration parameters might be too low. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Pool Read Time The average elapsed time for a read request. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Pool Write Time The average elapsed time for a write request. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Sort Time The average derived by dividing value of the Total Sort Time attribute by the value of the Total Sorts attribute. The average is expressed as elapsed time and has an integer value. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Binds Precompiles The number of binds and precompiles attempted. The value format is an integer. Use this attribute to gain insight into the current level of activity within the database manager. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Cat Cache Heap Full The number of times that an insert into the catalog cache failed because of a heap full condition in the database heap. The value format is an integer. The catalog cache draws its storage dynamically from the database heap. Even if the cache storage has not reached its limit, inserts into the catalog cache might fail due to a lack of space in the database heap. If the catalog cache heap full count is not zero, you can correct the insert failure condition by increasing the database heap size or by reducing the catalog cache size. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Cat Cache Hit Ratio The percentage of catalog sections that are found in the cache. The value format is an integer.

Cat Cache Inserts The number of times that the system tried to insert table descriptor information into the catalog cache. The value format is an integer. Table descriptor information is inserted into the cache following a failed lookup to the catalog cache while processing a table, view, or alias reference in an SQL statement. The catalog cache inserts value includes attempts to insert table descriptor information that fail due to catalog cache overflow and heap full conditions. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Cat Cache Lookups The number of times that the catalog cache was referenced to obtain table descriptor information. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Cat Cache Overflows The number of times that an insert into the catalog cache failed because the catalog cache was full. The value format is an integer. If the catalog cache overflows value is large, the catalog cache might be too small for the workload. Increasing the size of the catalog cache might improve its performance. If the workload includes transactions that compile a large number of SQL statements referencing many tables, views, and aliases in a single unit of work, compiling fewer SQL statements in a single transaction might improve the performance of the catalog cache. Or if the workload includes the binding of packages containing many SQL statements referencing many tables, views or aliases, you might want to split the packages so that they include fewer SQL statements to improve performance. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Client PID The process ID of the client application that made the connection to the database. The value format is an integer. Use this attribute to correlate monitor information such as CPU and I/O time to your client application. If a DRDA AS connection is used, this element is set to 0.

Client Platform The operating system on which the client application is running. Use this attribute to analyze problems for remote applications. The value format is a text string with a maximum of 20 characters. The following values are valid:

External value	Internal value
OS/2	OS/2
Windows3.x	Windows3.x
AIX	AIX
AS400 DRDA	AS400_DRDA
DOS	DOS
HP	HP
MAC	MAC
MVS DRDA	MVS_DRDA
SCO	SCO
SGI	SGI
SNI	SNI
SUN	SUN
LINUX	LINUX
UNKNOWN DRDA	UNKNOWN_DRDA
Unknown	Unknown
VM DRDA	VM_DRDA
VSE DRDA	VSE_DRDA
Windows95	Windows95
WindowsNT	WindowsNT

Client Prdid The product and version identifier for the software on the client. The value format is a text string with a maximum of 20 characters. For example: `SQL06010`.

Client Protocol The communication protocol that the client application is using to communicate with the server. The value format is a text string with a maximum of 12 characters. Use this attribute for troubleshooting of remote applications. The following values are valid:

External value	Internal value
IPX/SPX	IPX/SPX
Named Pipe	Named_Pipe
APPC	APPC
APPN	APPN
CPIC	CPIC
Local	Local
Netbios	Netbios
TCPIP	TCPIP
UNKNOWN	UNKNOWN

Commit SQL Stmts The total number of SQL COMMIT statements that have been attempted. The value format is an integer. A small rate of change in this counter during the monitor period might indicate that applications are not doing frequent commits. The lack of frequent commits can lead to problems with logging and data concurrency. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Conn Complete Timestamp The date and time that a connection request was granted.

Corr Token The DRDA AS correlation token. The value format is a text string with a maximum of 96 bytes.

Country Code The country code of the client application. The value format is an integer.

Creator The authorization ID of the user that precompiled the application (Unicode). The value format is a text string with a maximum of 60 bytes. Use this attribute to help identify the SQL statement that is processing, with the CREATOR column of the package section information in the catalogs.

Cursor Name The name of the cursor corresponding to this SQL statement. The value format is a text string with a maximum of 60 bytes.

DB Name The real name of the database for which information is collected or to which the application is connected. This name was given to the database when it was created. The value format is a simple text string with a maximum of 60 bytes. Use this attribute to identify the specific database to which the data applies.

DB Partition The DB2 database partition node number, which can range from 0 to 999. The Aggregated and Current Partition values can be used within a query or situation filter. If a db partition filter is not specified, data is returned for the current database partition. If a db partition filter is set to Aggregated, only aggregated partition data is returned. Historical data collection includes both aggregated and individual partition attribute data. In addition to numeric partition numbers in the 0 to 999 range, the following values are also valid:

External value	Internal value
Aggregated	-1
Current Partition	-2
All	-3

DDL SQL Stmts The number of SQL Data Definition Language (DDL) statements that were issued. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Deadlocks The total number of deadlocks that have occurred. The value format is an integer. This attribute can indicate that applications are experiencing contention problems. To resolve the problem, determine in which applications (or application processes) the deadlocks are occurring. You can then modify the application to enable it to run concurrently. Some applications, however, might not be capable of running concurrently. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Degree Parallelism The degree of parallelism requested when the query was bound. The value format is an integer. Use with the Agents Top attribute to determine if the query achieved maximum level of parallelism. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Direct Read Reqs The number of requests to perform a direct read of one or more sectors of data. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Direct Read Time The elapsed time (in milliseconds) required to perform the direct reads. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Direct Reads The number of read operations that do not use the buffer pool. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Direct Write Reqs The number of requests to perform a direct write of one or more sectors of data. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Direct Write Time The elapsed time (in milliseconds) required to perform the direct writes. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Direct Writes The number of write operations that do not use the buffer pool. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Dynamic SQL Stmts The number of dynamic SQL statements that were attempted. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Execution ID The ID that the user specified when logging in to the operating system. This ID is distinct from the Authorization ID, which the user specifies when connecting to the database. The value format is a text string with a maximum of 60 bytes. Use this attribute to determine the operating system user ID of the individual running the monitored application.

Failed SQL Stmts The number of SQL statements that were attempted, but failed. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Failed SQL Stmts Percent The percentage of SQL statements that failed to run successfully. The value format is an integer. This value is derived by dividing the value of the Failed SQL Statements attribute by the value of the Total SQL Statements attribute.

Hash Join Overflows The number of times that hash join data exceeded the available sort heap space. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Hash Join Small Overflows The number of times that hash join data exceeded the available sort heap space by less than 10%. The value format is an integer. If this value and the value of the Hash Join Overflows attribute are high, you must consider increasing the sort heap threshold. If this value is greater than 10% of Hash Join Overflows, you must consider increasing the sort heap size. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Int Auto Rebinds The number of automatic rebinds (or recompiles) that have been attempted. The value format is an integer. Automatic rebinds are the internal binds the system performs when a package has been invalidated. Use this attribute to determine the level of database activity at the application or database level. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Internal Commits The total number of commits initiated internally by the database manager. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Int Deadlock Rollbacks The total number of forced rollbacks initiated by the database manager due to a deadlock. The value format is an integer. The database manager initiates a rollback for the current unit of work in an application that is experiencing a deadlock. This attribute shows the number of deadlocks that have been broken. It can indicate the possibility of concurrency problems. It is also important because internal rollbacks due to deadlocks can cause performance degradation. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Int Rollbacks The total number of rollbacks initiated internally by the database manager. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Int Rows Deleted The number of rows deleted from the database as a result of internal activity. The value format is an integer. This attribute can help to gain insight into internal activity within the database manager. If this activity is high, you must evaluate the table design to determine if the referential constraints or triggers that you defined on the database are necessary. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Int Rows Inserted The number of rows inserted into the database as a result of internal activity caused by triggers. The value format is an integer. This attribute can help to gain insight into the internal activity within the database manager. If this activity is high, you must evaluate the design to determine if you can alter it to reduce this activity. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Int Rows Updated The number of rows updated from the database as a result of internal activity. The value format is an integer. This attribute can help to gain insight into internal activity within the database manager. If this activity is high, you must evaluate the table design to determine if the referential constraints that you defined on the database are necessary. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Lock Escals The number of times that locks have been escalated from several row locks to a table lock. A lock is escalated when the total number of locks held by an application reaches the maximum amount of lock list space available to the application, or the lock list space consumed by all applications is approaching the total lock list space. This data item includes a count of all lock escalations, including exclusive lock escalations. When an application reaches the maximum number of locks allowed and there are no more locks to escalate, the application uses space in the lock list that is allocated for other applications. When the entire lock list is full, an error occurs. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Lock Mode The type of lock being held. Use the lock mode to determine the source of contention for resources. The value format is a text string with a maximum of 32 characters. The following values are valid:

External value	Internal value
Exclusive Lock	Exclusive_Lock
Intent None (For Dirty Read)	Intent_None
Intention Exclusive Lock	Intn_Excl_Lock
Intention Share Lock	Intn_Share_Lock
Next-key Exclusive Lock	Next-key_Exclusive_Lock
Next-key Share Lock	Next-key_Share_Lock

External value	Internal value
Next key Weak Exclusive Lock	Next-key_Weak_Exclusive_Lock
No Lock	No_Lock
Share Lock	Share_Lock
Share with Intn Excl Lock	Share_Int_Ex_Lck
Super Exclusive Lock	Super_Excl_Lck
Unknown	Unknown
U-Lock	U-Lock
Weak Exclusive Lock	Weak_Exclusive_Lock

Lock Object Type The type of object against which the application holds a lock (for object-lock-level information), or the type of object for which the application is waiting to obtain a lock (for application-level and deadlock-level information). The value format is a text string with a maximum of 16 characters. The following values are valid:

External value	Internal value
Block lock type	Block_lock_type
Bufferpool	Bufferpool
End of Table	End_of_Table
Key Value	Key_Value
Int Cat Cache	Int_Cat_Cache
Int DMS Seq	Int_DMS_Seq
Int Long/Lob	Int_Long/Lob
Int Obj Table	Int_Obj_Table
Int Online Bkup	Int_Online_Bkup
Int Sequence	Int_Sequence
Int Table Alter	Int_Table_Alter
Int Variation	Int_Variation
Inplace reorg	Inplace_reorg
INTERNAL	INTERNAL
Internal Plan	Internal_Plan
No Lock	No_Lock
ROW	ROW
TABLE	TABLE
TABLESPACE	TABLESPACE
UNKNOWN	UNKNOWN

Lock Timeouts The number of times that a request to lock an object time out instead of being granted. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Lock Wait Start Time The date and time that the application started waiting to obtain a lock on the object that is currently locked by another application.

Lock Wait Time The total elapsed time (in milliseconds) that was spent waiting for a lock. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Lock Waits The total number of times that connections waited for locks. The value format is an integer. At the database level, the lock waits value is the total number of times that applications waited for locks within this database. At the application-connection level, the lock waits value is the total number of times that this connection requested a lock but waited because another connection was already holding a lock on the data. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Locks Held The number of locks currently held. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Node Name For new installations of version 6, release 2, the format is instanceid:hostname:UD for all operating systems. The format for version 6, release 1 of the DB2 agent on Windows systems is instanceid:hostname:UD; on UNIX and Linux systems, the format is instanceid:hostname.

Open Local Curs The number of local cursors currently open for this application, including those cursors counted by Open Local Cursors with Blocking attribute. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Open Local Curs Blk The number of local blocking cursors currently open for this application. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Open Rem Curs The number of remote cursors currently open for this application, including the cursors counted by the Open Remote Cursors with Blocking attribute. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Open Rem Curs Blk The number of remote blocking cursors currently open for this application. The value format is an integer. Use this attribute with the Open Remote Cursors attribute to calculate the percentage of remote cursors that are blocking cursors. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Package Name The name of the package that contains the SQL statement currently executing. The value format is a text string with a maximum of 60 bytes.

Pkg Cache Hit Ratio The percentage of package sections that were found in the cache. The value format is an integer.

Pkg Cache Inserts The total number of times that a requested section was not available for use and had to be loaded into the package cache. The value format is an integer. This count includes any implicit prepares performed by the system. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pkg Cache Lookups The number of times that an application looked for a section or package in the package cache. The value format is an integer. At a database level, it indicates the overall number of references since the database was started, or monitor data was reset. Note that this counter includes the cases where the section is already loaded in the cache and when the section has to be loaded into the cache. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Data from Estore Number of buffer pool data pages copied from extended storage. The value format is an integer.

Pool Data L Reads The number of logical read requests for data pages that have gone through the buffer pool. The value format is an integer. This count includes accesses to data that is already in the buffer pool when the database manager needs to process the page or read into the buffer pool before the database manager can process the page. The following value also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Data P Reads The number of read requests that required I/O to get data pages into the buffer pool. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Data to Estore Number of buffer pool data pages copied to extended storage. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Data Writes The number of times a buffer pool data page was physically written to disk. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Hit Ratio The buffer pool hit ratio (as a percentage). The value format is an integer. The sum of the Pool Data Logical Reads and Pool Index Logical Reads attributes is divided by the value of the Pool Total Reads attribute to derive the percentage.

Pool Index from Estore Number of buffer pool index pages copied from extended storage. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Index L Reads The number of logical read requests for index pages that have gone through the buffer pool. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Index P Reads The number of physical read requests to get index pages into the buffer pool. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Index to Estore Number of buffer pool index pages copied to extended storage. The value format is an integer. Pages are copied from the buffer pool to extended storage when they are selected as victim pages. As a result of the copying process, there is sufficient space for new pages in the buffer pool. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Index Writes The number of times a buffer pool index page was physically written to disk. The value format is an integer. If a buffer pool index page is written to disk for a high percentage of the Pool Index Physical Reads, performance might improve by increasing the number of buffer pool pages available for the database. If all applications are updating the database, increasing the size of the buffer pool might have minimal impact on performance; most pages contain updated data that must be written to disk. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Read Time The total amount of elapsed time spent processing read requests that caused data or index pages to be physically read from disk to buffer pool. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Total Reads The total number of read requests that required I/O to get data pages and index pages into the buffer pool. The value format is an integer. This attribute is the total of the Pool Data Physical Reads and Pool Index Physical Reads attributes. Use this attribute to determine how busy the DB2 server is in terms of I/O activity. Values that are greater than or equal to 2147483647 are indicated in the portal with the `Value Exceeds Maximum` text, and values that are smaller than -2147483648 are indicated with the `Value Exceeds Minimum` text. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Total Writes The total number of write requests. The value format is an integer. This attribute is the total of the Pool Data Writes and Pool Index Writes attributes. Use this attribute to determine how busy the DB2 server is in terms of write I/O activity. Values that are greater than or equal to 2147483647 are indicated in the portal with the `Value Exceeds Maximum` text, and values that are smaller than -2147483648 are indicated with the `Value Exceeds Minimum` text. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Write Time The total amount of time spent physically writing data or index pages from the buffer pool to disk. The value format is an integer. Use this attribute with the Buffer Pool Data Writes and Buffer Pool Index Writes attributes to calculate the average page-write time. This average is important because it might indicate the presence of an I/O wait, which in turn might indicate that you must move data to a different device. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Query Card Estimate An estimate of the number of rows that are returned by a query. The value format is an integer. You can compare this estimate by the SQL compiler with the actual runtime values. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Query Cost Estimate Estimated cost, in timerons, for a query, as determined by the SQL compiler. The value format is an integer. This attribute allows correlation of actual runtime values with the compile-time estimates. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Rej Curs Blk The number of times that a request for an I/O block at the server was rejected and the request was converted to non-blocked I/O. If there are many cursors blocking data, the communication heap might become full. The value format is an integer. When this heap is full, I/O blocks are not allocated for blocking cursors; however, an error condition does not alert you to this condition. If cursors are unable to block data, performance can be affected adversely. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Rollback SQL Stmts The total number of SQL ROLLBACK statements that have been attempted. The value format is an integer. A rollback can result from an application request, a deadlock, or an error situation. This attribute counts only the number of rollback statements issued from applications. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Rows Deleted The number of row deletions attempted. The value format is an integer. Use this attribute to gain insight into the current level of activity within the database manager. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Rows Inserted The number of row insertions attempted. The value format is an integer. Use this attribute to gain insight into the current level of activity within the database manager. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Rows Read The number of rows read from the table. The value format is an integer. This attribute helps to identify tables with heavy usage for which you might want to create additional indexes. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Rows Selected The number of rows that have been selected and returned to the application. The value format is an integer. Use this attribute to gain insight into the current level of activity within the database manager. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Rows Updated The number of row updates attempted. The value format is an integer. Use this attribute to gain insight into the current level of activity within the database manager. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Rows Written The number of rows changed (inserted, deleted, or updated) in the table. The value format is an integer. A high value for table-level information indicates heavy usage of the table. If so, you might want to use the Run Statistics (RUNSTATS) utility to maintain efficiency of the packages used for this table. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Select SQL Stmts The number of SQL SELECT statements that were issued. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Snapshot Timestamp The date and time when the database system monitor information was collected. Use this attribute to help relate data chronologically if you are saving the results in a file or database for ongoing analysis.

Sort Overflows The total number of sorts that ran out of sort heap space and might have required disk space for temporary storage. The value format is an integer. at the database or application level, use this element with the Total Sorts attribute. This attribute can help to determine the source of contention for resources. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Sort Overflows Percent The percentage of sorts that ran out of sort heap space and might have required disk space for temporary storage. The value format is an integer. This percentage is calculated by dividing the value of the Sort Overflows attribute by the value of the Total Sorts attribute. at the database or application level, use this attribute to evaluate the percentage of sorts that required overflow to disk. If this percentage is high, you might want to adjust the database configuration by increasing the value of the SORTHEAP configuration parameter.

Static SQL Stmts The number of static SQL statements that were attempted. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Stmt Operation The statement operation currently being processed or most recently processed (if none is currently running). The value format is a text string with a maximum of 20 characters. The following values are valid:

External value	Internal value
EXECUTE IMMEDIATE	EXECUTE_IMMEDIATE
STATIC COMMIT	STATIC_COMMIT
STATIC ROLLBACK	STATIC_ROLLBACK
0	0
CLOSE	CLOSE
DESCRIBE	DESCRIBE
EXECUTE	EXECUTE
FETCH	FETCH
OPEN	OPEN
PREPARE	PREPARE
UNKNOWN	UNKNOWN

Stmt Text The text of the dynamic SQL statement. For application snapshots, the statement text helps you identify what the application was executing when the snapshot was taken, or most recently processed if no statement was being processed at the time the snapshot was taken. For dynamic SQL statements, this attribute identifies the SQL text associated with a package. The value format is a text string with a maximum of 2000 bytes.

The statement text can only be fetched during the time when the dynamic SQL query is executing. If the query executing time is short, the statement text might not be retrieved.

Stmt Type The type of SQL statement processed. The value format is a text string with a maximum of 32 characters. The following values are valid:

External value	Internal value
DYNAMIC	DYNAMIC
NON-STATEMENT OPERATION	NON-STATEMENT_OPERATION
STATIC	STATIC
UNKNOWN STMT TYPE	UNKNOWN_STMT_TYPE

Table Name The name of the table the application is waiting to lock. The value format is a text string with a maximum of 60 bytes. Use this attribute with the Table Schema attribute to determine the source of contention for resources.

Table Schema The schema of the table the application is waiting to lock. The value format is a text string with a maximum of 60 bytes. Along with the Table Name attribute, this attribute can help to determine the source of contention for resources.

Tablespace Name The name of the tablespace that the application is waiting to lock. The value format is a text string with a maximum of 60 bytes. This attribute can help you to determine the source of contention for resources.

Total Hash Joins The total number of hash joins that ran. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Total Hash Loops The total number of times that a single partition of a hash join was larger than the available sort heap space. The value format is an integer. Values for this attribute indicate inefficient execution of hash joins. This might indicate that the sort heap size is too small or the sort heap threshold is too small. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Total Sorts The total number of sorts that have been issued. The value format is an integer. at the database or application level, use this value with the Sort Overflows attribute to calculate the percentage of sorts that need more heap space. You can also use it with the Total Sort Time attribute to calculate the average sort time. If the number of sort overflows is small with respect to the total sorts, increasing the sort heap size might have little impact on performance, unless this buffer size is increased substantially. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Total Sort Time The total elapsed time (in milliseconds) for all sorts that ran. The value format is an integer. at the database or application level, use this element with the Total Sorts attribute to calculate the average sort time. This average can indicate whether sorting is a performance concern. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Total SQL Stmt The total number of dynamic and static SQL statements. This value is derived by adding the values of the Dynamic SQL Statements and the Static SQL Statements attributes. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

UID SQL Stmts The number of SQL UPDATE, INSERT, and DELETE statements that were issued. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

UOW Lock Wait Time The time the UOW (unit of work) waited on locks (in seconds). The valid format is integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

X Lock Escals The number of times that locks have been escalated from several row locks to one exclusive table lock, or the number of times an exclusive lock on a row caused the table lock to become an exclusive lock. The value format is an integer. A lock is escalated when the total number of locks held by an application reaches the maximum amount of lock list space available to the application. The amount of lock list space available is determined by the LOCKLIST and MAXLOCKS configuration parameters. Other applications cannot access data held by an exclusive lock. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Application00 (KUDDB2APPLGROUP00) attributes (Superseded)

The Application00 group provides information about application activities. You can use this information to monitor the performance of your applications and identify problem areas for corrective action. All values are integers that are calculated from the first application connection, unless otherwise noted.

Acc Curs Blk The number of times that a request for an I/O block was accepted. Use this attribute with the Rejected Block Cursor Requests attribute to calculate the percentage of blocking requests that are accepted or rejected.

Agent ID The application handle, which is a system-wide unique ID for the application. On multi-node systems, where a database is partitioned, this ID is the same on every node where the application might make a secondary connection. Use the application handle to identify an active application (application handle is synonymous with agent ID) uniquely.

Agent ID Holding Lock The application handle of the agent holding a lock for which this application is waiting. The lock monitor group must be turned on to obtain this information. This attribute can help you to determine which applications are in contention for resources.

Agent Sys CPU Time The total system CPU time (in seconds) that the database manager agent process spent executing database manager code. This element includes CPU time for both SQL and non-SQL statements, and CPU time for any unfenced user-defined functions (UDFs).

Agent User CPU Time The total CPU time (in microseconds) that the database manager agent process used. This counter includes time spent on both SQL and non-SQL statements, and any fenced user-defined functions (UDFs) or stored procedures issued by the application. System CPU represents the time spent in system calls. User CPU represents time spent executing database manager code. The value format is a text string with a maximum of 10 characters. Use this attribute with the other CPU-time related attributes to help you identify applications or queries that consume large amounts of CPU time.

Appl Conn Time The string date and time that an application started a connection request. The value format is CYYMMDDHHMMSSmmm, where the following measures apply:

Value	Description
C	Century (0 for 20th, 1 for 21st)
YY	Year
MM	Month
DD	Day
HH	Hour
MM	Minute
SS	Second
mmm	Millisecond

Use this attribute to determine when the application started its connection request to the database.

Appl ID The identifier generated when the application connects to the database at the database manager or when DDCS receives a request to connect to a DRDA database. The identifier is unique across the network. The application ID is displayed in various formats that depend on the communication protocol between the client and the server system on which the database manager or DDCS is running. Each of the formats consists of three parts that are separated by periods. Use this ID (known on both the client and server) to correlate the client and server parts of the application.

Appl ID Holding Lock The application ID of the application that is holding a lock on the object that this application is waiting to obtain. The value format is a text string with a maximum of 32 characters. This attribute can help you determine which applications are in contention for resources. You can use it to identify the application handle (agent ID) and table ID that are holding the lock.

Appl Idle Time The number of seconds since an application issued a request to the server. This amount includes applications that have not ended a transaction (for example, have not issued a commit or rollback). Use this information to force users who have been idle for a specified number of seconds.

Appl Name The name of the application running at the client as it is known to the database manager or DB2 Connect. The value format is a text string, with a maximum of 20 characters. For example: `*Local.db2inst1.990212202018`. Use this attribute with the Application ID attribute to relate data items with your application. In a client/server environment, this name is passed from the client to the server to establish the database connection. For DRDA-AS connections, this name is the DRDA external name.

Appl Status The status of the monitored application. This attribute can help you diagnose potential application problems. The following values are valid:

External value	Internal value
Backing Up Database	Backing Up Database
Commit Active	Commit Active
Compiling SQL Stmt	Compiling SQL Stmt
Connect Pending	Connect Pending
Connected	Connected
Creating Database	Creating Database
Decoupled	Decoupled
Disconnect Pending	Disconnect Pending
I/O Error Waiting	I/O Error Waiting
Loading Database	Loading Database
Lock Waiting	Lock Waiting
Prepared Transaction	Prepared Transaction
Quiescing a Tablespace	Quiescing a Tablespace
Recompiling Plan	Recompiling Plan
Request Interrupted	Request Interrupted
Restarting Database	Restarting Database
Restoring Database	Restoring Database
Rollback Active	Rollback Active
Trans. heuristically aborted	Trans. heuristically aborted
Trans. heuristically committed	Trans. heuristically committed
Transaction ended	Transaction ended
UOW Executing	UOW Executing
UOW Waiting in the application	UOW Waiting in the application
Unloading Database	Unloading Database
UNKNOWN	UNKNOWN

Auth ID The authorization ID of the user who invoked the monitored application. On a DB2 Connect gateway node, this ID is the user authorization ID on the host. The value format is a text string with a maximum of 20 characters. Use this attribute to determine who invoked the application.

Avg Lock Wait Time The average elapsed time (in milliseconds) that was spent waiting for a lock. If the average lock wait time is high, you must look for applications that hold many locks, or have lock escalations, to focus on tuning your applications to improve concurrency. If escalations cause a high average lock wait time, the values of one or both of the LOCKLIST and MAXLOCKS configuration parameters might be too low.

Avg Pool Read Time The average elapsed time for a read request. This value is derived by dividing the value of the Pool Read Time attribute by the value of the Pool Total Reads attribute. This average can indicate the presence of an I/O wait, which might indicate that you must move data to a different device.

Avg Pool Write Time The average elapsed time for a write request. This value is derived by dividing the value of the Pool Write Time attribute by the value of the Pool Total Writes attribute.

Avg Sort Time The average derived by dividing value of the Total Sort Time attribute by the value of the Total Sorts attribute. The average is expressed as elapsed time. at the database or application level, this attribute can indicate whether sorting is a performance issue. System load affects elapsed time. As you increase the running processes, this elapsed time value becomes higher.

Binds Precompiles The number of attempted binds and precompiles. Use this attribute to gain insight into the current level of activity within the database manager.

Cat Cache Heap Full The number of times that a heap full condition in the database heap caused an insert into the catalog cache to fail. The catalog cache draws its storage dynamically from the database heap. Even if the cache storage has not reached its limit, inserts into the catalog cache can fail if space is lacking in the database heap. If the catalog cache heap full count is not zero, you can correct the insert failure condition by increasing the database heap size or by reducing the catalog cache size.

Cat Cache Hit Ratio The percentage of catalog sections found in the cache. This attribute indicates how well the catalog cache is avoiding catalog accesses. If the value is high (more than 0.8), the cache is performing well. A smaller value might indicate that you must increase the size of the catalog cache. You must expect a large value immediately following the first connection to the database.

Cat Cache Inserts The number of times that the system tried to insert table descriptor information into the catalog cache. Table descriptor information is inserted into the cache following a failed lookup to the catalog cache while processing a table, view, or alias reference in an SQL statement. The catalog cache inserts value includes attempts to insert table descriptor information that fail if the catalog cache overflows and the heap is full.

Cat Cache Lookups The number of times that the catalog cache was referenced to obtain table descriptor information. This attribute includes both successful and unsuccessful accesses to the catalog cache. To calculate the catalog cache hit ratio use the following formula:

```
(1 - (cat_cache_inserts / cat_cache_lookups))
```

This ratio indicates how well the catalog cache is avoiding catalog accesses. If the ratio is high (more than 0.8), the cache is performing well. A smaller ratio might indicate that you must increase the size of the catalog cache. You must expect a large ratio immediately following the first connection to the database.

Cat Cache Overflows The number of times that an insert into the catalog cache failed because the catalog cache was full. If the catalog cache overflows value is large, the catalog cache might be too small for the workload. Increasing the size of the catalog cache might improve its performance. If the workload includes transactions that compile a large number of SQL statements referencing many tables, views, and aliases in a single unit of work, compiling fewer SQL statements in a single transaction might improve the performance of the catalog cache. Or if the workload includes the binding of packages containing many SQL statements referencing many tables, views or aliases, you might want to split the packages so that they include fewer SQL statements to improve performance.

Client PID The process ID of the client application that made the connection to the database. Use this attribute to correlate monitor information such as CPU and I/O time to your client application. If a DRDA AS connection is used, this element is set to 0.

Client Platform The operating system on which the client application is running. Use this attribute to analyze problems for remote applications. The following values are valid:

External value	Internal value
OS/2	OS/2
Windows3.x	Windows3.x
AIX	AIX
AS400 DRDA	AS400 DRDA
DOS	DOS
HP	HP
MAC	MAC
MVS DRDA	MVS DRDA
SCO	SCO
SGI	SGI
SNI	SNI
SUN	SUN
LINUX	LINUX
UNKNOWN DRDA	UNKNOWN DRDA
Unknown	Unknown
VM DRDA	VM DRDA
VSE DRDA	VSE DRDA
Windows95	Windows95
WindowsNT	WindowsNT

Client Prdid The product and version identifier for the software on the client. The value format is a text string with a maximum of 20 characters, for example:

```
SQL06010
```

Client Protocol The communication protocol that the client application is using to communicate with the server. Use this attribute for troubleshooting of remote applications. The following values are valid:

External value	Internal value
IPX/SPX	IPX/SPX
Named Pipe	Named Pipe
APPC	APPC

External value	Internal value
APPN	APPN
CPIC	CPIC
Local	Local
Netbios	Netbios
TCPIP	TCPIP
UNKNOWN	UNKNOWN

Commit SQL Stmts The total number of SQL COMMIT statements that have been attempted. A small rate of change in this counter during the monitor period might indicate that applications are not doing frequent commits. The lack of frequent commits can lead to problems with logging and data concurrency. You can also use this attribute to calculate the total number of units of work by calculating the sum of the following values:
- Commit statements attempted
- Internal commits
- Rollback statements attempted
- Internal rollbacks

Conn Complete Time The string date and time that a connection request was granted. The value format is CYYMMDDHHMMSSmmm, where the following measures apply:

Value	Description
C	Century (0 for 20th, 1 for 21st)
YY	Year
MM	Month
DD	Day
HH	Hour
MM	Minute
SS	Second
mmm	Millisecond

Use this attribute to determine when a connection request to the database was granted.

Corr Token The DRDA AS correlation token. The value format is a text string with a maximum of 32 characters. Use the DRDA correlation token to correlate the processing between the application server and the application requester. It is an identifier dumped into logs when errors arise. As a result, you can use it to identify the conversation that is in error. In some cases, it is the LUWID of the conversation. If communications are not using DRDA, this element returns the appl_id attribute.

Country Code The country code of the client application.

Creator The authorization ID of the user that pre-compiled the application. The value format is a text string with a maximum of 20 characters. Use this attribute to help identify the SQL statement that is processing, with the CREATOR column of the package section information in the catalogs.

Cursor Name The name of the cursor corresponding to this SQL statement. The value format is a text string with a maximum of 20 characters. Use this attribute to identify the SQL statement that is processing. This name is used on an OPEN, FETCH, CLOSE, and PREPARE of an SQL SELECT statement. If a cursor is not used, this field is blank.

DB Name The real name of the database for which information is collected or to which the application is connected. This name was given to the database when it was created. The value format is a simple text string with a maximum of 20 characters. Use this attribute to identify the specific database to which the data applies.

DDL SQL Stmts The number of SQL Data Definition Language (DDL) statements that ran. Use this attribute to determine the level of database activity at the application or database level. DDL statements are expensive to run because of their impact on the system catalog tables. As a result, if the value of this attribute is high, you must determine the cause and possibly restrict the identified activity from being performed.

Deadlocks The total number of deadlocks that have occurred. This attribute can indicate that applications are experiencing contention problems. To resolve the problem, determine in which applications (or application processes) the deadlock are occurring. You can then modify the application to enable it to run concurrently. Some applications, however, might not be capable of running concurrently.

Degree Parallelism The degree of parallelism requested when the query was bound. Use with the Agents Top attribute to determine if the query achieved maximum level of parallelism.

Direct Read Reqs The number of requests to perform a direct read of one or more sectors of data. Use the following formula to calculate the average number of sectors that are read by a direct read:

```
direct reads from database / direct read requests
```

Direct Read Time The elapsed time (in milliseconds) required to perform the direct reads. Use the following formula to calculate the average direct read time per sector:

```
direct read time / direct reads from database
```

A high average time might indicate an I/O conflict.

Direct Reads The number of read operations that do not use the buffer pool. Use the following formula to calculate the average number of sectors that are read by a direct read:

```
direct reads from database / direct read requests
```

When using system monitors to track I/O, this data attribute helps to distinguish database I/O from non-database I/O on the device.

Direct Write Reqs The number of requests to perform a direct write of one or more sectors of data. Use the following formula to calculate the average number of sectors that are written by a direct write:

```
direct writes to database / direct write requests
```

Direct Write Time The elapsed time (in milliseconds) required to perform the direct writes. Use the following formula to calculate the average direct write time per sector:

```
direct write time / direct writes to database
```

A high average time might indicate an I/O conflict.

Direct Writes The number of write operations that do not use the buffer pool. Use the following formula to calculate the average number of sectors that are written by a direct write:

```
direct writes to database / direct write requests
```

When using system monitors to track I/O, this data attribute helps to distinguish database I/O from non-database I/O on the device.

Dynamic SQL Stmts The number of dynamic SQL statements that were attempted. Use this attribute to calculate the total number of successful SQL statements at the database or application level:

1. Sum the number of Dynamic SQL Statements Attempted and the Static SQL Statements Attempted.
2. Subtract the number of Failed Statement Operations.

The remainder equals the throughput (the number of successful SQL statements) during the current monitoring period.

Execution ID The ID that the user specified when logging in to the operating system. This ID is distinct from the Authorization ID, which the user specifies when connecting to the database. The value format is a text string with a maximum of 20 characters. Use this attribute to determine the operating system user ID of the individual running the monitored application.

Failed SQL Stmts The number of SQL statements that were attempted, but failed. This count includes all SQL statements that received a negative SQLCODE. Use this attribute to calculate the total number of successful SQL statements at the database or application level:

1. Sum the number of Dynamic SQL Statements Attempted and the Static SQL Statements Attempted.
2. Subtract the number of Failed Statement Operations.

The remainder equals the throughput (the number of successful SQL statements) during the current monitoring period. This attribute can also help you to determine the reasons for poor performance; failed statements indicate time wasted by the database manager, which results in lower throughput for the database.

Failed SQL Stmts Pct The percentage of SQL statements that failed to run. This value is derived by dividing the value of the Failed SQL Statements attribute by the value of the Total SQL Statements attribute.

Hash Join Overflows The number of times that hash join data exceeded the available sort heap space. At the database level, if the percentage of Hash Join Small Overflows is greater than 10% of this value, you must consider increasing the sort heap size. You can use values at the application level to evaluate hash join performance for individual applications.

Hash Join Small Overflows The number of times that hash join data exceeded the available sort heap space by less than 10%. If this value and the value of the Hash

Join Overflows attribute are high, you must consider increasing the sort heap threshold. If this value is greater than 10% of Hash Join Overflows, you must consider increasing the sort heap size.

Int Auto Rebinds The number of automatic rebinds (or recompiles) that have been attempted. Automatic rebinds are the internal binds the system performs when the validity of a package is removed. Use this attribute to determine the level of database activity at the application or database level. Because internal automatic rebinds can have a significant impact on performance, they must be minimized where possible.

Int Commits The total number of commits initiated internally by the database manager. An internal commit might occur during one of the following operations:
- A reorganization
- An import
- A bind or pre-compile
- An application that ends without executing an explicit SQL COMMIT statement (on UNIX and Linux systems).

Use this attribute to calculate the total number of units of work by calculating the sum of the following values: commit statements attempted, internal commits, rollback statements attempted, and internal rollbacks.

Int Deadlock Rollbacks The total number of forced rollbacks initiated by the database manager due to a deadlock. The database manager initiates a rollback for the current unit of work in an application that is experiencing a deadlock. This attribute shows the number of deadlocks that were broken. It can indicate the possibility of concurrency problems. It is also important because internal rollbacks due to deadlocks can cause performance degradation.

Int Rollbacks The total number of rollbacks initiated internally by the database manager. An internal rollback occurs when any of the following operations cannot be completed successfully:
- A reorganization
- An import
- A bind or pre-compile
- An application that ends as a result of a deadlock situation or lock timeout situation
- An application that ends without executing an explicit COMMIT or ROLLBACK statement (on Windows systems).

Use this attribute to calculate the total number of units of work by calculating the sum of the following values: commit statements attempted, internal commits, rollback statements attempted, and internal rollbacks.

Int Rows Deleted The number of rows deleted from the database as a result of internal activity. This attribute can help to gain insight into internal activity within the database manager. If this activity is high, you must evaluate the table design to determine if the referential constraints or triggers that you defined on the database are necessary.

Int Rows Inserted The number of rows inserted into the database as a result of internal activity caused by triggers. This attribute can help to gain insight into the internal activity within the database manager. If this activity is high, you must evaluate the design to determine if you can alter it to reduce this activity.

Int Rows Updated The number of rows updated from the database as a result of internal activity. This attribute can help to gain insight into internal activity within the database manager. If this activity is high, you must evaluate the table design to determine if the referential constraints that you defined on the database are necessary.

Lock Escals The number of times that locks have been escalated from several row locks to a table lock. A lock is escalated when the total number of locks held by an application reaches the maximum amount of lock list space available to the application, or the lock list space consumed by all applications is approaching the total lock list space. This data item includes a count of all lock escalations, including exclusive lock escalations. When an application reaches the maximum number of locks allowed and there are no more locks to escalate, the application uses space in the lock list that is allocated for other applications. When the entire lock list is full, an error occurs. The value format is an integer.

Lock Mode The type of lock being held. Use the lock mode to determine the source of contention for resources. You can specify one of the following valid values:

External value	Internal value
Exclusive Lock	Exclusive Lock
Intent None (For Dirty Read)	Intent None
Intention Exclusive Lock	Intn Excl Lock
Intention Share Lock	Intn Share Lock
Next-key Exclusive Lock	Next-key Exclusive Lock
Next-key Share Lock	Next-key Share Lock
Next key Weak Exclusive Lock	Next-key Weak Exclusive Lock
No Lock	No Lock
Share Lock	Share Lock
Share with Intn Excl Lock	Share Intn Ex Lck
Super Exclusive Lock	Super Excl Lck
Unknown	Unknown
U-Lock	U-Lock
Weak Exclusive Lock	Weak Exclusive Lock

This attribute indicates one of the following lock types, depending on the type of monitoring information that is being examined:

- The type of lock another application holds on the object that this application is waiting to lock (for application-monitoring and deadlock-monitoring levels)
- The type of lock held on the object by this application (for object-lock levels)

Lock Object Type The type of object against which the application holds a lock (for object-lock-level information), or the type of object for which the application is waiting to obtain a lock (for application-level and deadlock-level information). The following values are valid:

External value	Internal value
Block lock type	Block lock type
Bufferpool	Bufferpool

External value	Internal value
End of Table	End of Table
Key Value	Key Value
Int Cat Cache	Int Cat Cache
Int DMS Seq	Int DMS Seq
Int Long/Lob	Int Long/Lob
Int Obj Table	Int Obj Table
Int Online Bkup	Int Online Bkup
Int Sequence	Int Sequence
Int Table Alter	Int Table Alter
Int Variation	Int Variation
Inplace reorg	Inplace reorg
INTERNAL	INTERNAL
Internal Plan	Internal Plan
No Lock	No Lock
ROW	ROW
TABLE	TABLE
TABLESPACE	TABLESPACE
UNKNOWN	UNKNOWN

Use this attribute to help you determine the source of contention for resources.

Lock Timeouts The number of times that a request to lock an object time out instead of being granted. This attribute can help you adjust the setting for the LOCKTIMEOUT database configuration parameter. If the number of lock timeouts becomes excessive when compared to normal operating levels, an application might be holding locks for long durations. In this case, this attribute might indicate that you must analyze some of the other attributes related to locks and deadlocks to determine if an application problem exists. You can have too few lock timeouts if the LOCKTIMEOUT database configuration parameter is set too high. In this case, applications might wait excessively to obtain a lock.

Lock Wait Start Time The string date and time that the application started waiting to obtain a lock on the object that is currently locked by another application. The value format is CYYMMDDHHMMSSmmm, where the following measures apply:

Value	Description
C	Century (0 for 20th, 1 for 21st)
YY	Year
MM	Month
DD	Day
HH	Hour
MM	Minute
SS	Second
mmm	Millisecond

This attribute can help you to determine the severity of resource contention.

Lock Wait Time The total elapsed time (in milliseconds) that was spent waiting for a lock. At the database level, this is the total amount of elapsed time that all applications were waiting for a lock within this database. At the application-connection and transaction levels, this is the total amount of elapsed time that this connection or transaction has waited for a lock to be granted. This attribute might be used with the Lock Waits attribute to calculate the average wait time for a lock. This calculation can be performed at either the database or the application-connection level.

Lock Waits The total number of times that applications or connections waited for locks. At the database level, the lock waits value is the total number of times that applications waited for locks within this database. At the application-connection level, the lock waits value is the total number of times that this connection requested a lock but waited because another connection was already holding a lock on the data. Use this attribute with the Lock Wait Time attribute to calculate, at the database level, the average wait time for a lock. This calculation can be performed at either the database or the application-connection level. If the average lock wait time is high, look for applications that hold many locks, or have lock escalations, with a focus on tuning your applications to improve concurrency, if appropriate. If escalations are the reason for a high average lock wait time, the values of one or both of the LOCKLIST and MAXLOCKS configuration parameters might be too low.

Locks Held The number of locks currently held. If the monitor information is at the database level, this number represents the total number of locks currently held by all applications in the database. If the information is at the application level, this number represents the total number of locks currently held by all agents for the application.

Node Name For new installations of version 6, release 2, the format is instanceid:hostname:UD for all operating systems. The format for version 6, release 1 of the DB2 agent on Windows systems is instanceid:hostname:UD; on UNIX and Linux system, the format is instanceid:hostname.

Open Local Curs The number of local cursors currently open for this application, including those cursors counted by Open Local Cursors with Blocking attribute. Use this attribute with the Open Local Cursors with Blocking attribute to calculate the percentage of local cursors that are blocking cursors. If the percentage is low, you might be able to improve performance by improving the row blocking in the application. For cursors used by remote applications, see the Open Remote Cursors attribute.

Open Local Curs Blk The number of local blocking cursors currently open for this application. Use this attribute with the Open Local Cursors attribute to calculate the percentage of local cursors that are blocking cursors. Rejected Block Cursor Requests and Accepted Block Cursor Requests attributes provide additional information that might help you tune your configuration parameters to improve row blocking in your application. For blocking cursors used by remote applications, see the Open Remote Cursors with Blocking attribute.

Open Rem Curs The number of remote cursors currently open for this application, including the cursors counted by the Open Remote Cursors with Blocking attribute. Use this attribute with the Open Remote Cursors with Blocking attribute to calculate the percentage of remote cursors that are blocking cursors. If the percentage is low, you might be able to improve performance by improving the row blocking in the application. For the number of open cursors used by applications connected to a local database, see the Open Local Cursors attribute.

Open Rem Curs Blk The number of remote blocking cursors currently open for this application. Use this attribute with the Open Remote Cursors attribute to calculate the percentage of remote cursors that are blocking cursors. If the percentage is low, you might be able to use the following steps to improve performance by improving the row blocking in the application:

1. Check the pre-compile options for record blocking for treatment of ambiguous cursors.
2. Redefine cursors to allow for blocking.

Rejected Block Cursor Requests and Accepted Block Cursor Requests attributes provide additional information that might help you tune your configuration parameters to improve row blocking in your application. For the number of open blocking cursors that are used by applications connected to a local database, see the Open Local Cursors with Blocking attribute.

Package Name The name of the package that contains the SQL statement currently executing. The value format is a text string with a maximum of 20 characters. Use this attribute to help you identify the application program and the SQL statement that is executing.

Pkg Cache Hit Ratio The percentage of package sections that were found in the cache. This attribute tells you whether the package cache is being used effectively. If the hit ratio is high (more than 0.8), the cache is performing well. A smaller ratio might indicate that the package cache must be increased.

Pkg Cache Inserts The total number of times that a requested section was not available for use and had to be loaded into the package cache. This count includes any implicit prepares performed by the system. By using the Package Cache Lookups attribute, you can calculate the package cache hit ratio using the following formula:

```
1 - (Package Cache Inserts / Package Cache Lookups)
```

Pkg Cache Lookups The number of times that an application looked for a section or package in the package cache. At a database level, it indicates the overall number of references since the database was started, or monitor data was reset. Note that this counter includes the cases where the section is already loaded in the cache and when the section has to be loaded into the cache. To calculate the package cache hit ratio use the following formula:

```
1 - (Package Cache Inserts / Package Cache Lookups)
```

The package cache hit ratio tells you whether the package cache is being used effectively. If the hit ratio is high (more than 0.8), the cache is performing well. A smaller ratio might indicate that the package cache must be increased.

Pool Data from Estore Number of buffer pool data pages copied from extended storage. Required pages are copied from extended storage to the buffer pool. The copy process might incur the cost of connecting to the shared memory segment, but it saves the cost of a disk read.

Pool Data L Reads The number of logical read requests for data pages that have gone through the buffer pool. This count includes accesses to the following types of data:

- Data that is already in the buffer pool when the database manager needs to process the page
- Data that is read into the buffer pool before the database manager can process the page

By using the Pool Data Physical Reads attribute, you can calculate the data page hit ratio for the buffer pool as in the following formula:

```
1 - (buffer pool data physical reads / buffer pool data logical reads)
```

By using the Pool Data Physical Reads, Pool Index Physical Reads, and Pool Index Logical Reads attributes, you can calculate the overall buffer pool hit ratio as in the following example:

```
1 - ((buffer pool data physical reads + buffer pool index physical reads)
/ (buffer pool data logical reads + buffer pool index logical reads))
```

Increasing buffer pool size generally improves the hit ratio until you reach a point of diminishing returns.

Pool Data P Reads The number of read requests that required I/O to get data pages into the buffer pool.

Pool Data to Estore Number of buffer pool data pages copied to extended storage. Pages are copied from the buffer pool to extended storage when they are selected as victim pages. As a result of the copying process, there is sufficient space for new pages in the buffer pool.

Pool Data Writes The number of times a buffer pool data page was physically written to disk. A buffer pool data page is written to disk for the following reasons:

- To free a page in the buffer pool so that another page can be read
- To flush the buffer pool

If a buffer pool data page is written to disk for a high percentage of Buffer Pool Data Physical Reads, performance might improve by increasing the number of buffer pool pages available for the database.

Pool Hit Ratio The buffer pool hit ratio (as a percentage). The sum of the Pool Data Logical Reads and Pool Index Logical Reads attributes is divided by the value of the Pool Total Reads attribute to derive the ratio. Use this attribute to determine whether buffer pool assignment is efficient. If the pool hit ratio is low, increasing the number of buffer pool pages might improve performance.

Pool Index from Estore Number of buffer pool index pages copied from extended storage. Required index pages are copied from extended storage to the buffer pool. The copy process might incur the cost of connecting to the shared memory segment, but it saves the cost of a disk read.

Pool Index L Reads The number of logical read requests for index pages that have gone through the buffer pool. This count includes accesses to index pages that are already in the buffer pool when the database manager needs to process the page or read into the buffer pool before the database manager can process the page. By using the Pool Index Physical Reads attribute, you can calculate the index page hit ratio for the buffer pool using the following formula:

```
1 - (buffer pool index physical reads / buffer pool index logical reads)
```

To calculate the overall buffer pool hit ratio, see the Pool Data Logical Reads attribute. If the hit ratio is low, increasing the number of buffer pool pages might improve performance.

Pool Index P Reads The number of physical read requests to get index pages into the buffer pool.

Pool Index to Estore Number of buffer pool index pages copied to extended storage. Pages are copied from the buffer pool to extended storage when they are selected as victim pages. As a result of the copying process, there is sufficient space for new pages in the buffer pool.

Pool Index Writes The number of times a buffer pool index page was physically written to disk. If a buffer pool index page is written to disk for a high percentage of the Pool Index Physical Reads, performance might improve by increasing the number of buffer pool pages available for the database. If all applications are updating the database, increasing the size of the buffer pool might have minimal impact on performance; most pages contain updated data that must be written to disk.

Pool Read Time The total amount of elapsed time spent processing read requests that caused data or index pages to be physically read from disk to buffer pool. Use this attribute with the Buffer Pool Data Physical Reads and Buffer Pool Index Physical Reads attributes to calculate the average page-read time. This average is important because it might indicate the presence of an I/O wait, which can indicate that you must move data to a different device.

Pool Total Reads The total number of read requests that required I/O to get data pages and index pages into the buffer pool. This attribute is the total of the Pool Data Physical Reads and Pool Index Physical Reads attributes. Use this attribute to determine how busy the DB2 server is in terms of I/O activity. Values that are greater than or equal to 2147483647 are indicated in the portal with the `Value Exceeds Maximum` text, and values that are smaller than -2147483648 are indicated with the `Value Exceeds Minimum` text. The following values are valid:

External value	Internal value
Value Exceeds Maximum	2147483647
Value Exceeds Minimum	-2147483648

Pool Total Writes The total number of write requests. This attribute is the total of the Pool Data Writes and Pool Index Writes attributes. Use this attribute to determine how busy the DB2 server is in terms of write I/O activity. Values that are greater than or equal to 2147483647 are indicated in the portal with the `Value Exceeds Maximum` text, and values that are smaller than -2147483648 are indicated with the `Value Exceeds Minimum` text. The following values are valid:

External value	Internal value
Value Exceeds Maximum	2147483647
Value Exceeds Minimum	-2147483648

Pool Write Time The total amount of time spent physically writing data or index pages from the buffer pool to disk. Use this attribute with the Buffer Pool Data Writes and Buffer Pool Index Writes attributes to calculate the average page-write time. This average is important because it might indicate the presence of an I/O wait, which in turn might indicate that you must move data to a different device.

Prefetch Wait Time The time an application spent waiting for an I/O server or prefetcher to finish loading pages into the buffer pool. This attribute can be used to experiment with changing the number of I/O servers and the I/O server sizes.

Prev UOW Stop Time The string date and time that the unit of work completed. The value format is CYYMMDDHHMMSSmmm, where the following measures apply:

Value	Description
C	Century (0 for 20th, 1 for 21st)
YY	Year
MM	Month
DD	Day
HH	Hour
MM	Minute
SS	Second
mmm	Millisecond

Use this attribute with the UOW Stop Time attribute to calculate the total elapsed time between COMMIT/ROLLBACK points, and with the UOW Start Time attribute to calculate the time spent in the application between units of work.

Query Card Estimate An estimate of the number of rows that are returned by a query. You can compare this estimate by the SQL compiler with the actual runtime values.

Query Cost Estimate Estimated cost, in timerons, for a query, as determined by the SQL compiler. This attribute allows correlation of actual runtime values with the compile-time estimates.

Rej Curs Blk The number of times that a request for an I/O block at the server was rejected and the request was converted to non-blocked I/O. If there are many cursors blocking data, the communication heap might become full. When this heap is full, I/O blocks are not allocated for blocking cursors; however, an error condition does not alert you to this condition. If cursors are unable to block data, performance can be affected adversely.

Rollback SQL Stmts The total number of SQL ROLLBACK statements that have been attempted. A rollback can result from an application request, a deadlock, or an error situation. This attribute counts only the number of rollback statements issued from applications. At the application level, this attribute can help you determine the level of database activity for the application and the amount of conflict with other applications. At the database level, it can help you determine the amount of activity in the database and the amount of conflict between applications on the database.

Rows Deleted The number of row deletions attempted. Use this attribute to gain insight into the current level of activity within the database manager.

Rows Inserted The number of row insertions attempted. Use this attribute to gain insight into the current level of activity within the database manager.

Rows Read The number of rows read from the table. This attribute helps to identify tables with heavy usage for which you might want to create additional indexes.

Rows Selected The number of rows that have been selected and returned to the application. Use this attribute to gain insight into the current level of activity within the database manager.

Rows Updated The number of row updates attempted. Use this attribute to gain insight into the current level of activity within the database manager.

Rows Written The number of rows changed (inserted, deleted, or updated) in the table. A high value for table-level information indicates heavy usage of the table. If so, you might want to use the Run Statistics (RUNSTATS) utility to maintain efficiency of the packages used for this table.

Section Number The internal section number in the package for the SQL statement currently processing or most recently processed.

Select SQL Stmts The number of SQL SELECT statements that ran. Use this attribute to determine the level of database activity at the application or database level. You can also use the following steps to determine the ratio of SELECT statements to the total statements:
1. Sum the number of attempted static SQL statements and the number of attempted dynamic SQL statements.
2. Divide the sum by the number of select SQL statements that ran.

Snapshot Time The string date and time when the database system monitor information was collected. Use this attribute to help relate data chronologically if you are saving the results in a file or database for ongoing analysis. The Tivoli Enterprise Portal formats the timestamp value into a date and time string. The internal timestamp value stored in the database is in the format cYYMMDDhhmmss000, which is described in the following table:

Value	Description
C	Century (0 for 20th, 1 for 21st)
YY	Year (last two digits of the year, for example (00 - 99)
MM	Month (01 for January, 02 for February, and so on)
DD	Day (the day of the month, for example 01 - 31)
HH	Hour (the hour of the day in 24-hour format from 00 - 23)
MM	Minute (the minutes of the hour from 00 - 59)
SS	Second (the seconds of the hour (00 - 59)

Sort Overflows The total number of sorts that ran out of sort heap space and might have required disk space for temporary storage. at the database or application level, use this element with the Total Sorts attribute, this attribute can help to determine the source of contention for resources.

Sort Overflows Pct The percentage of sorts that ran out of sort heap space and might have required disk space for temporary storage. This percentage is calculated by dividing the value of the Sort Overflows attribute by the value of the Total Sorts attribute. at the database or application level, use this attribute to evaluate the percentage of sorts that required overflow to disk. If this percentage is high, you might want to adjust the database configuration by increasing the value of the SORTHEAP configuration parameter.

Static SQL Stmts The number of static SQL statements that were attempted. Use this attribute to calculate the total number of successful SQL statements at the database or application level using the following steps:

1. Sum the number of Dynamic SQL Statements Attempted and the Static SQL Statements Attempted.
2. Subtract the number of Failed Statement Operations.

The remainder equals the throughput (the number of successful SQL statements) during the current monitoring period.

Stmt Operation The statement operation currently being processed or most recently processed (if none is currently running). The following values are valid:

External value	Internal value
EXECUTE IMMEDIATE	EXECUTE IMMEDIATE
STATIC COMMIT	STATIC COMMIT
STATIC ROLLBACK	STATIC ROLLBACK
0	0
CLOSE	CLOSE
DESCRIBE	DESCRIBE
EXECUTE	EXECUTE
FETCH	FETCH
OPEN	OPEN
PREPARE	PREPARE
UNKNOWN	UNKNOWN

Use this attribute to determine the operation that is executing or recently finished.

Stmt Start The string date and time that the most recent SQL statement operation started. The value format is CYYMMDDHHMMSSmmm, where the following measures apply:

Value	Description
C	Century (0 for 20th, 1 for 21st)
YY	Year
MM	Month
DD	Day
HH	Hour
MM	Minute
SS	Second
mmm	Millisecond

Use this attribute with the Statement Stop attribute to calculate the elapsed execution time for the statement operation.

Stmt Stop The string date and time that the most recent SQL statement operation stopped. If the statement is still running, this field is 0 (zero). Use this attribute with the Statement Start attribute to calculate the elapsed execution time for the statement operation.

Stmt Text The text of the dynamic SQL statement. For application snapshots, the statement text helps you identify what the application was executing when the snapshot was taken, or most recently processed if no statement was being processed at the time the snapshot was taken. For dynamic SQL statements, this attribute identifies the SQL text associated with a package.

The statement text can only be fetched during the time when the dynamic SQL query is executing. If the query executing time is short, the statement text might not be retrieved.

Stmt Type The type of SQL statement processed. The following values are valid:

External value	Internal value
DYNAMIC	DYNAMIC
NON-STATEMENT OPERATION	NON-STATEMENT OPERATION
STATIC	STATIC
UNKNOWN STMT TYPE	UNKNOWN STMT TYPE

Use this attribute to determine the type of statement that is executing.

Table Name The name of the table the application is waiting to lock. The value format is a text string with a maximum of 20 characters. Use this attribute with the Table Schema attribute to determine the source of contention for resources.

Table Schema The schema of the table the application is waiting to lock. The value format is a text string with a maximum of 20 characters. Along with the Table Name attribute, this attribute can help to determine the source of contention for resources.

Tablespace Name the name of the tablespace that the application is waiting to lock. The value format is a text string with a maximum of 20 characters. This attribute can help you to determine the source of contention for resources.

Total Hash Joins The total number of hash joins that ran. At the database or application level, use this value with the Hash Join Overflows attribute and the Hash Join Small Overflows attribute to determine if a significant percentage of hash joins would benefit from modest increases in the sort heap size.

Total Hash Loops The total number of times that a single partition of a hash join was larger than the available sort heap space. Values for this attribute indicate inefficient execution of hash joins. This might indicate that the sort heap size is too small or the sort heap threshold is too small. Use this value with the other hash join variables to tune the sort heap size (sortheap) and sort heap threshold (sheapthres) configuration parameters.

Total Sort Time The total elapsed time (in milliseconds) for all sorts that ran. at the database or application level, use this element with the Total Sorts attribute to calculate the average sort time. This average can indicate whether sorting is a performance concern.

Total Sorts The total number of sorts that ran. at the database or application level, use this value with the Sort Overflows attribute to calculate the percentage of sorts that need more heap space. You can also use it with the Total Sort Time attribute to calculate the average sort time. If the number of sort overflows is small with respect to the total sorts, increasing the sort heap size might have little impact on performance, unless this buffer size is increased substantially.

Total SQL Stmt The total number of dynamic and static SQL statements. This value is derived by adding the values of the Dynamic SQL Statements and the Static SQL Statements attributes.

UID SQL Stmts The number of SQL UPDATE, INSERT, and DELETE statements that ran. Use this attribute to determine the level of database activity at the application or database level. You can also use the following steps to determine the ratio of UPDATE, INSERT, and DELETE statements to the total number of statements:

1. Sum the number of attempted static SQL statements and the number of attempted dynamic SQL statements.
2. Divide the number of UPDATE/INSERT/DELETE SQL statements that ran by the sum derived in step 1.

UOW Comp Status The completion status of the previous UOW (unit of work). Use this attribute to determine if the unit of work ended due to a deadlock or an abnormal termination. The following values are valid:

External value	Internal value
Appl Normal Termination	Appl Normal Termination
UOW Commit	UOW Commit
UOW RB - Lock Timeout	UOW RB - Lock Timeout
UOW RB due to Abend	UOW RB due to Abend
UOW RB due to Deadlock	UOW RB due to Deadlock
UOW Rolled Back	UOW Rolled Back
0	0
Unknown	Unknown

UOW Lock Wait Time The total amount of elapsed time this unit of work has spent waiting for locks. This attribute can help you to determine the severity of the resource contention problem.

UOW Log Space Used The amount of log space (in bytes) used in the current unit of work of the monitored application. Use this attribute to understand the logging requirements at the unit-of-work level. Values that are greater than or equal to 2147483647 are indicated in the portal with the `Value Exceeds Maximum` text, and values that are smaller than -2147483648 are indicated with the `Value Exceeds Minimum` text. The following values are valid:

External value	Internal value
Value Exceeds Maximum	2147483647
Value Exceeds Minimum	-2147483648

UOW Start Time The string date and time that the unit of work first required database resources. This resource requirement occurs at the first SQL statement execution for the unit of work. The value format is CYYMMDDHHMMsss where the following measures apply:

Value	Description
C	Century (0 for 20th, 1 for 21st)
YY	Year
MM	Month
DD	Day
HH	Hour
MM	Minute
SS	Second
mmm	Millisecond

Use this attribute with the UOW Stop Time attribute to calculate the total elapsed time of the unit of work and with the Previous Unit of Work Completion Timestamp attribute to calculate the time spent in the application between units of work.

UOW Stop Time The string date and time that the most recent unit of work completed, which occurs when database changes are committed or rolled back. The value format is CYYMMDDHHMMsss where the following measures apply:

Value	Description
C	Century (0 for 20th, 1 for 21st)
YY	Year
MM	Month
DD	Day
HH	Hour
MM	Minute
SS	Second
mmm	Millisecond

X Lock Escals The number of times that locks have been escalated from several row locks to one exclusive table lock, or the number of times an exclusive lock on a row caused the table lock to become an exclusive lock. A lock is escalated when the total number of locks held by an application reaches the maximum amount of lock list space available to the application. The amount of lock list space available is determined by the LOCKLIST and MAXLOCKS configuration parameters. Other applications cannot access data held by an exclusive lock. Because exclusive locks can affect the concurrency of your data, it is important to track them. When an application reaches the maximum number of locks allowed and there are no more locks to escalate, it uses space in the lock list allocated for other applications. When the entire lock list is full, an error occurs. Use this attribute with the Previous UOW Stop Time attribute to calculate the total elapsed time between COMMIT/ROLLBACK points, and with the UOW Start Time attribute to calculate the elapsed time of the latest unit of work.

Application00U (KUDDB2APPLGROUP00_U) attributes (Superseded)

The Application00U attribute group provides information about the database and the application. By using this information, you can determine the efficiency of the database and identify any problem areas for corrective action. All values are integers that are calculated from the first application connection, unless otherwise noted. This attribute group is superseded. Use the attributes in the Application00 (KUDAPPL00) attribute group.

Acc Curs Blk The number of times that a request for an I/O block was accepted. The value format is an integer. Use this attribute with the Rejected Block Cursor Requests attribute to calculate the percentage of blocking requests that are accepted or rejected.

Agent ID The application handle, which is a system-wide unique ID for the application. The value format is an integer. On multi-node systems, where a database is partitioned, this ID is the same on every node where the application might make a secondary connection.

Agent ID Holding Lock The application handle of the agent holding a lock for which this application is waiting. The value format is an integer. The lock monitor group must be turned on to obtain this information.

Agent Sys CPU Time The total system CPU time (in seconds) that the database manager agent process spent executing database manager code. This element includes CPU time for both SQL and non-SQL statements, and CPU time for any unfenced user-defined functions (UDFs).

Agent User CPU Time The total CPU time (in microseconds) used by the database manager agent process.

Appl Conn Time The string date and time that an application started a connection request. The value format is CYYMMDDHHMMSSmmm, where the following measures apply:

Value	Description
C	Century (0 for 20th, 1 for 21st)
YY	Year
MM	Month
DD	Day
HH	Hour
MM	Minute
SS	Second
mmm	Millisecond

Appl ID The identifier generated when the application connects to the database at the database manager or when DDCS receives a request to connect to a DRDA database. The value format is a text string, with a maximum of 32 characters.

Appl ID Holding Lock (Unicode) The application ID of the application that is holding a lock on the object that this application is waiting to obtain (Unicode). The value format is a text string with a maximum of 96 bytes.

Appl Idle Time The number of seconds since an application issued a request to the server. The value format is an integer.

Appl Name (Unicode) The name of the application running at the client as it is known to the database manager or DB2 Connect (Unicode). The value format is a text string, with a maximum of 60 bytes. For example: `*Local.db2inst1.990212202018`.

Appl Status The status of the application being monitored. This attribute can help you diagnose potential application problems. The value format is a text string with a maximum of 64 characters. The following values are valid:

External value	Internal value
Backing Up Database	Backing_Up_Database
Commit Active	Commit_Active
Compiling SQL Stmt	Compiling_SQL_Stmt
Connect Pending	Connect_Pending
Connected	Connected
Creating Database	Creating_Database
Decoupled	Decoupled
Disconnect Pending	Disconnect_Pending
I/O Error Waiting	I/O_Error_Waiting
Loading Database	Loading_Database
Lock Waiting	Lock_Waiting
Prepared Transaction	Prepared_Transaction
Quiescing a Tablespace	Quiescing_a_Tablespace
Recompiling Plan	Recompiling_Plan
Request Interrupted	Request_Interrupted
Restarting Database	Restarting_Database
Restoring Database	Restoring_Database
Rollback Active	Rollback_Active
Trans. heuristically aborted	Trans._heuristically_aborted
Trans. heuristically committed	Trans._heuristically_committed
Transaction ended	Transaction_ended
UOW Executing	UOW Executing
UOW Waiting in the application	UOW Waiting in the application
Unloading Database	Unloading Database
UNKNOWN	UNKNOWN

Auth ID (Unicode) The authorization ID of the user who invoked the application that is being monitored (Unicode). On a DB2 Connect gateway node, this ID is the user authorization ID on the host. The value format is a text string with a maximum of 60 bytes. Use this attribute to determine who invoked the application.

Avg Lock Wait Time The average elapsed time (in milliseconds) that was spent waiting for a lock. The value format is an integer. If the average lock wait time is high, you must look for applications that hold many locks, or have lock escalations, with a focus on tuning your applications to improve concurrency, if appropriate. If escalations are the reason for a high average lock wait time, the values of one or both of the LOCKLIST and MAXLOCKS configuration parameters might be too low.

Avg Pool Read Time The average elapsed time for a read request. The value format is an integer.

Avg Pool Write Time The average elapsed time for a write request. The value format is an integer.

Avg Sort Time The average derived by dividing value of the Total Sort Time attribute by the value of the Total Sorts attribute. The average is expressed as elapsed time and has an integer value.

Binds Precompiles The number of binds and precompiles attempted. The value format is an integer. Use this attribute to gain insight into the current level of activity within the database manager.

Cat Cache Heap Full The number of times that an insert into the catalog cache failed because of a heap full condition in the database heap. The value format is an integer. The catalog cache draws its storage dynamically from the database heap. Even if the cache storage has not reached its limit, inserts into the catalog cache might fail due to a lack of space in the database heap. If the catalog cache heap full count is not zero, you can correct the insert failure condition by increasing the database heap size or by reducing the catalog cache size.

Cat Cache Hit Ratio The percentage of catalog sections found in the cache. The value format is an integer.

Cat Cache Inserts The number of times that the system tried to insert table descriptor information into the catalog cache. The value format is an integer. Table descriptor information is inserted into the cache following a failed lookup to the catalog cache while processing a table, view, or alias reference in an SQL statement. The catalog cache inserts value includes attempts to insert table descriptor information that fail due to catalog cache overflow and heap full conditions.

Cat Cache Lookups The number of times that the catalog cache was referenced to obtain table descriptor information. The value format is an integer.

Cat Cache Overflows The number of times that an insert into the catalog cache failed because the catalog cache was full. The value format is an integer. If the catalog cache overflows value is large, the catalog cache might be too small for the workload. Increasing the size of the catalog cache might improve its performance. If the workload includes transactions that compile a large number of SQL statements referencing many tables, views, and aliases in a single unit of work, compiling fewer SQL statements in a single transaction might improve the performance of the catalog cache. Or if the workload includes the binding of packages containing many SQL statements referencing many tables, views or aliases, you might want to split the packages so that they include fewer SQL statements to improve performance.

Client PID The process ID of the client application that made the connection to the database. The value format is an integer. Use this attribute to correlate monitor information such as CPU and I/O time to your client application. If a DRDA AS connection is used, this element is set to 0.

Client Platform The operating system on which the client application is running. Use this attribute to analyze problems for remote applications. The value format is a text string with a maximum of 20 characters. The following values are valid:

External value	Internal value
OS/2	OS/2
Windows3.x	Windows3.x
AIX	AIX
AS400 DRDA	AS400_DRDA
DOS	DOS
HP	HP
MAC	MAC
MVS DRDA	MVS_DRDA
SCO	SCO
SGI	SGI
SNI	SNI
SUN	SUN
LINUX	LINUX
UNKNOWN DRDA	UNKNOWN_DRDA
Unknown	Unknown
VM DRDA	VM_DRDA
VSE DRDA	VSE_DRDA
Windows95	Windows95
WindowsNT	WindowsNT

Client Prdid The product and version identifier for the software on the client. The value format is a text string with a maximum of 20 characters. For example: `SQL06010`.

Client Protocol The communication protocol that the client application is using to communicate with the server. The value format is a text string with a maximum of 12 characters. Use this attribute for troubleshooting of remote applications. The following values are valid:

External value	Internal value
IPX/SPX	IPX/SPX
Named Pipe	Named_Pipe
APPC	APPC
APPN	APPN
CPIC	CPIC
Local	Local
Netbios	Netbios
TCPIP	TCPIP
UNKNOWN	UNKNOWN

Commit SQL Stmts The total number of SQL COMMIT statements that have been attempted. The value format is an integer. A small rate of change in this counter during the monitor period might indicate that applications are not doing frequent commits. The lack of frequent commits can lead to problems with logging and data concurrency.

Conn Complete Time The string date and time that a connection request was granted. The value format is CYYMMDDHHMMSSmmm, where the following measures apply:

Value	Description
C	Century (0 for 20th, 1 for 21st)
YY	Year
MM	Month
DD	Day
HH	Hour
MM	Minute
SS	Second
mmm	Millisecond

Corr Token (Unicode) The DRDA AS correlation token (Unicode). The value format is a text string with a maximum of 96 bytes.

Country Code The country code of the client application. The value format is an integer.

Creator (Unicode) The authorization ID of the user that precompiled the application (Unicode). The value format is a text string with a maximum of 60 bytes. Use this attribute to help identify the SQL statement that is processing, with the CREATOR column of the package section information in the catalogs.

Cursor Name (Unicode) The name of the cursor corresponding to this SQL statement (Unicode). The value format is a text string with a maximum of 60 bytes.

DB Name (Unicode) The real name of the database for which information is collected or to which the application is connected (Unicode). This name was given to the database when it was created. The value format is a simple text string with a maximum of 60 bytes. Use this attribute to identify the specific database to which the data applies.

DB Partition The DB2 database partition node number, which can range from 0 to 999. The Aggregated and Current Partition values can be used within a query or situation filter. If a db partition filter is not specified, data is returned for the current database partition. If a db partition filter is set to Aggregated, only aggregated partition data is returned. Historical data collection includes both aggregated and individual partition attribute data. In addition to numeric partition numbers in the 0 to 999 range, the following values are also valid:

External value	Internal value
Aggregated	-1
Current Partition	-2

External value	Internal value
All	-3

DDL SQL Stmts The number of SQL Data Definition Language (DDL) statements that ran. The value format is an integer.

Deadlocks The total number of deadlocks that have occurred. The value format is an integer. This attribute can indicate that applications are experiencing contention problems. To resolve the problem, determine in which applications (or application processes) the deadlocks are occurring. You can then modify the application to enable it to run concurrently. Some applications, however, might not be capable of running concurrently.

Degree Parallelism The degree of parallelism requested when the query was bound. The value format is an integer. Use with the Agents Top attribute to determine if the query achieved maximum level of parallelism.

Direct Read Reqs The number of requests to perform a direct read of one or more sectors of data. The value format is an integer.

Direct Read Time The elapsed time (in milliseconds) required to perform the direct reads. The value format is an integer.

Direct Reads The number of read operations that do not use the buffer pool. The value format is an integer.

Direct Write Reqs The number of requests to perform a direct write of one or more sectors of data. The value format is an integer.

Direct Write Time The elapsed time (in milliseconds) required to perform the direct writes. The value format is an integer.

Direct Writes The number of write operations that do not use the buffer pool. The value format is an integer.

Dynamic SQL Stmts The number of dynamic SQL statements that were attempted. The value format is an integer.

Execution ID (Unicode) The ID that the user specified when logging in to the operating system (Unicode). This ID is distinct from the Authorization ID, which the user specifies when connecting to the database. The value format is a text string with a maximum of 60 bytes. Use this attribute to determine the operating system user ID of the individual running the monitored application.

Failed SQL Stmts The number of SQL statements that were attempted, but failed. The value format is an integer.

Failed SQL Stmts Pct The percentage of SQL statements that failed to run succesfully. The value format is an integer. This value is derived by dividing the value of the Failed SQL Statements attribute by the value of the Total SQL Statements attribute.

Hash Join Overflows The number of times that hash join data exceeded the available sort heap space. The value format is an integer.

Hash Join Small Overflows The number of times that hash join data exceeded the available sort heap space by less than 10%. The value format is an integer. If this value and the value of the Hash Join Overflows attribute are high, you must consider increasing the sort heap threshold. If this value is greater than 10% of Hash Join Overflows, you must consider increasing the sort heap size.

Int Auto Rebinds The number of automatic rebinds (or recompiles) that have been attempted. The value format is an integer. Automatic rebinds are the internal binds the system performs when a package has been invalidated. Use this attribute to determine the level of database activity at the application or database level.

Int Commits The total number of commits initiated internally by the database manager. The value format is an integer.

Int Deadlock Rollbacks The total number of forced rollbacks initiated by the database manager due to a deadlock. The value format is an integer. The database manager initiates a rollback for the current unit of work in an application that is experiencing a deadlock. This attribute shows the number of deadlocks that have been broken. It can indicate the possibility of concurrency problems. It is also important because internal rollbacks due to deadlocks can cause performance degradation.

Int Rollbacks The total number of rollbacks initiated internally by the database manager. The value format is an integer.

Int Rows Deleted The number of rows deleted from the database as a result of internal activity. The value format is an integer. This attribute can help to gain insight into internal activity within the database manager. If this activity is high, you must evaluate the table design to determine if the referential constraints or triggers that you defined on the database are necessary.

Int Rows Inserted The number of rows inserted into the database as a result of internal activity caused by triggers. The value format is an integer. This attribute can help to gain insight into the internal activity within the database manager. If this activity is high, you must evaluate the design to determine if you can alter it to reduce this activity.

Int Rows Updated The number of rows updated from the database as a result of internal activity. The value format is an integer. This attribute can help to gain insight into internal activity within the database manager. If this activity is high, you must evaluate the table design to determine if the referential constraints that you defined on the database are necessary.

Lock Escals The number of times that locks have been escalated from several row locks to a table lock. A lock is escalated when the total number of locks held by an application reaches the maximum amount of lock list space available to the application, or the lock list space consumed by all applications is approaching the total lock list space. This data item includes a count of all lock escalations, including exclusive lock escalations. When an application reaches the maximum number of locks allowed and there are no more locks to escalate, the application uses space in the lock list that is allocated for other applications. When the entire lock list is full, an error occurs. The value format is an integer.

Lock Mode The type of lock being held. Use the lock mode to determine the source of contention for resources. The value format is a text string with a maximum of 32 characters. The following values are valid:

External value	Internal value
Exclusive Lock	Exclusive_Lock
Intent None (For Dirty Read)	Intent_None
Intention Exclusive Lock	Intn_Excl_Lock
Intention Share Lock	Intn_Share_Lock
Next-key Exclusive Lock	Next-key_Exclusive_Lock
Next-key Share Lock	Next-key_Share_Lock
Next key Weak Exclusive Lock	Next-key_Weak_Exclusive_Lock
No Lock	No_Lock
Share Lock	Share_Lock
Share with Intn Excl Lock	Share_Int_Ex_Lck
Super Exclusive Lock	Super_Excl_Lck
Unknown	Unknown
U-Lock	U-Lock
Weak Exclusive Lock	Weak_Exclusive_Lock

Lock Object Type The type of object against which the application holds a lock (for object-lock-level information), or the type of object for which the application is waiting to obtain a lock (for application-level and deadlock-level information). The value format is a text string with a maximum of 16 characters. The following values are valid:

External value	Internal value
Block lock type	Block_lock_type
Bufferpool	Bufferpool
End of Table	End_of_Table
Key Value	Key_Value
Int Cat Cache	Int_Cat_Cache
Int DMS Seq	Int_DMS_Seq
Int Long/Lob	Int_Long/Lob
Int Obj Table	Int_Obj_Table
Int Online Bkup	Int_Online_Bkup
Int Sequence	Int_Sequence
Int Table Alter	Int_Table_Alter
Int Variation	Int_Variation
Inplace reorg	Inplace_reorg
INTERNAL	INTERNAL
Internal Plan	Internal_Plan
No Lock	No_Lock
ROW	ROW
TABLE	TABLE
TABLESPACE	TABLESPACE
UNKNOWN	UNKNOWN

Lock Timeouts The number of times that a request to lock an object time out instead of being granted. The value format is an integer.

Lock Wait Start Time The string date and time that the application started waiting to obtain a lock on the object that is currently locked by another application. The value format is CYYMMDDHHMMSSmmm, where the following measures apply:

Value	Description
C	Century (0 for 20th, 1 for 21st)
YY	Year
MM	Month
DD	Day
HH	Hour
MM	Minute
SS	Second
mmm	Millisecond

Lock Wait Time The total elapsed time (in milliseconds) that was spent waiting for a lock.

Lock Waits The total number of times that connections waited for locks. The value format is an integer. At the database level, the lock waits value is the total number of times that applications waited for locks within this database. At the application-connection level, the lock waits value is the total number of times that this connection requested a lock but waited because another connection was already holding a lock on the data.

Locks Held The number of locks currently held. The value format is an integer.

Node Name For new installations of version 6, release 2, the format is instanceid:hostname:UD for all operating systems. The format for version 6, release 1 of the DB2 agent on Windows systems is instanceid:hostname:UD; on UNIX and Linux systems, the format is instanceid:hostname.

Open Local Curs The number of local cursors currently open for this application, including those cursors counted by Open Local Cursors with Blocking attribute. The value format is an integer.

Open Local Curs Blk The number of local blocking cursors currently open for this application. The value format is an integer.

Open Rem Curs The number of remote cursors currently open for this application, including the cursors counted by the Open Remote Cursors with Blocking attribute. The value format is an integer.

Open Rem Curs Blk The number of remote blocking cursors currently open for this application. The value format is an integer. Use this attribute with the Open Remote Cursors attribute to calculate the percentage of remote cursors that are blocking cursors.

Package Name (Unicode) The name of the package that contains the SQL statement currently executing (Unicode). The value format is a text string with a maximum of 60 bytes.

Pkg Cache Hit Ratio The percentage of package sections that were found in the cache. The value format is an integer.

Pkg Cache Inserts The total number of times that a requested section was not available for use and had to be loaded into the package cache. The value format is an integer. This count includes any implicit prepares performed by the system.

Pkg Cache Lookups The number of times that an application looked for a section or package in the package cache. The value format is an integer. At a database level, it indicates the overall number of references since the database was started, or monitor data was reset. Note that this counter includes the cases where the section is already loaded in the cache and when the section has to be loaded into the cache.

Pool Data from Estore The number of buffer pool data pages that are copied from extended storage. The value format is an integer.

Pool Data L Reads The number of logical read requests for data pages that have gone through the buffer pool. The value format is an integer. This count includes accesses to data that is already in the buffer pool when the database manager needs to process the page or read into the buffer pool before the database manager can process the page.

Pool Data P Reads The number of read requests that required I/O to get data pages into the buffer pool. The value format is an integer.

Pool Data to Estore Number of buffer pool data pages copied to extended storage. The value format is an integer.

Pool Data Writes The number of times a buffer pool data page was physically written to disk. The value format is an integer.

Pool Hit Ratio The buffer pool hit ratio (as a percentage). The value format is an integer. The sum of the Pool Data Logical Reads and Pool Index Logical Reads attributes is divided by the value of the Pool Total Reads attribute to derive the ratio.

Pool Index from Estore Number of buffer pool index pages copied from extended storage. The value format is an integer.

Pool Index L Reads The number of logical read requests for index pages that have gone through the buffer pool. The value format is an integer.

Pool Index P Reads The number of physical read requests to get index pages into the buffer pool. The value format is an integer.

Pool Index to Estore Number of buffer pool index pages copied to extended storage. The value format is an integer. Pages are copied from the buffer pool to extended storage when they are selected as victim pages. As a result of the copying process, there is sufficient space for new pages in the buffer pool.

Pool Index Writes The number of times a buffer pool index page was physically written to disk. The value format is an integer. If a buffer pool index page is written to disk for a high percentage of the Pool Index Physical Reads, performance might improve by increasing the number of buffer pool pages available for the database. If all applications are updating the database, increasing the size of the buffer pool might have minimal impact on performance; most pages contain updated data that must be written to disk.

Pool Read Time The total amount of elapsed time spent processing read requests that caused data or index pages to be physically read from disk to buffer pool. The value format is an integer.

Pool Total Reads The total number of read requests that required I/O to get data pages and index pages into the buffer pool. The value format is an integer. This attribute is the total of the Pool Data Physical Reads and Pool Index Physical Reads attributes. Use this attribute to determine how busy the DB2 server is in terms of I/O activity. Values that are greater than or equal to 2147483647 are indicated in the portal with the `Value Exceeds Maximum` text, and values that are smaller than -2147483648 are indicated with the `Value Exceeds Minimum` text. The following values are valid:

External value	Internal value
Value Exceeds Maximum	2147483647
Value Exceeds Minimum	-2147483648

Pool Total Writes The total number of write requests. The value format is an integer. This attribute is the total of the Pool Data Writes and Pool Index Writes attributes. Use this attribute to determine how busy the DB2 server is in terms of write I/O activity. Values that are greater than or equal to 2147483647 are indicated in the portal with the `Value Exceeds Maximum` text, and values that are smaller than -2147483648 are indicated with the `Value Exceeds Minimum` text. The following values are valid:

External value	Internal value
Value Exceeds Maximum	2147483647
Value Exceeds Minimum	-2147483648

Pool Write Time The total amount of time spent physically writing data or index pages from the buffer pool to disk. The value format is an integer. Use this attribute with the Buffer Pool Data Writes and Buffer Pool Index Writes attributes to calculate the average page-write time. This average is important because it might indicate the presence of an I/O wait, which in turn might indicate that you must move data to a different device.

Prefetch Wait Time The time an application spent waiting for an I/O server (prefetcher) to finish loading pages into the buffer pool. The value format is an integer. This attribute can be used to experiment with changing the number of I/O servers and the I/O server sizes.

Prev UOW Stop Time The string date and time that the unit of work completed. The value format is CYYMMDDHHMMSSmmm, where the following measures apply:

Value	Description
C	Century (0 for 20th, 1 for 21st)
YY	Year
MM	Month

Value	Description
DD	Day
HH	Hour
MM	Minute
SS	Second
mmm	Millisecond

Query Card Estimate An estimate of the number of rows that are returned by a query. The value format is an integer. You can compare this estimate by the SQL compiler with the actual runtime values.

Query Cost Estimate Estimated cost, in timerons, for a query, as determined by the SQL compiler. The value format is an integer. This attribute allows correlation of actual runtime values with the compile-time estimates.

Rej Curs Blk The number of times that a request for an I/O block at the server was rejected and the request was converted to non-blocked I/O. If there are many cursors blocking data, the communication heap might become full. The value format is an integer. When this heap is full, I/O blocks are not allocated for blocking cursors; however, an error condition does not alert you to this condition. If cursors are unable to block data, performance can be affected adversely.

Rollback SQL Stmts The total number of SQL ROLLBACK statements that have been attempted. The value format is an integer. A rollback can result from an application request, a deadlock, or an error situation. This attribute counts only the number of rollback statements issued from applications.

Rows Deleted The number of row deletions attempted. The value format is an integer. Use this attribute to gain insight into the current level of activity within the database manager.

Rows Inserted The number of row insertions attempted. The value format is an integer. Use this attribute to gain insight into the current level of activity within the database manager.

Rows Read The number of rows read from the table. The value format is an integer. This attribute helps to identify tables with heavy usage for which you might want to create additional indexes.

Rows Selected The number of rows that have been selected and returned to the application. The value format is an integer. Use this attribute to gain insight into the current level of activity within the database manager.

Rows Updated The number of row updates attempted. The value format is an integer. Use this attribute to gain insight into the current level of activity within the database manager.

Rows Written The number of rows changed (inserted, deleted, or updated) in the table. The value format is an integer. A high value for table-level information indicates heavy usage of the table. If so, you might want to use the Run Statistics (RUNSTATS) utility to maintain efficiency of the packages used for this table.

Section Number The internal section number in the package for the SQL statement currently processing or most recently processed. The value format is an integer.

Select SQL Stmts The number of SQL SELECT statements that ran. The value format is an integer.

Snapshot Time The string date and time when the database system monitor information was collected. Use this attribute to help relate data chronologically if you are saving the results in a file or database for ongoing analysis. The Tivoli Enterprise Portal formats the timestamp value into a date and time string. The internal timestamp value stored in the database is in the format cYYMMDDhhmmss000, which is described in the following table:

Value	Description
C	Century (0 for 20th, 1 for 21st)
YY	Year (last two digits of the year, for example (00 - 99)
MM	Month (01 for January, 02 for February, and so on)
DD	Day (the day of the month, for example 01 - 31)
HH	Hour (the hour of the day in 24-hour format from 00 - 23)
MM	Minute (the minutes of the hour from 00 - 59)
SS	Second (the seconds of the hour (00 - 59)

Sort Overflows The total number of sorts that ran out of sort heap space and might have required disk space for temporary storage. The value format is an integer. at the database or application level, use this element with the Total Sorts attribute. This attribute can help to determine the source of contention for resources.

Sort Overflows Pct The percentage of sorts that ran out of sort heap space and might have required disk space for temporary storage. The value format is an integer. This percentage is calculated by dividing the value of the Sort Overflows attribute by the value of the Total Sorts attribute. at the database or application level, use this attribute to evaluate the percentage of sorts that required overflow to disk. If this percentage is high, you might want to adjust the database configuration by increasing the value of the SORTHEAP configuration parameter.

Static SQL Stmts The number of static SQL statements that were attempted. The value format is an integer.

Stmt Operation The statement operation currently being processed or most recently processed (if none is currently running). The value format is a text string with a maximum of 20 characters. The following values are valid:

External value	Internal value
EXECUTE IMMEDIATE	EXECUTE_IMMEDIATE
STATIC COMMIT	STATIC_COMMIT
STATIC ROLLBACK	STATIC_ROLLBACK
0	0
CLOSE	CLOSE
DESCRIBE	DESCRIBE

External value	Internal value
EXECUTE	EXECUTE
FETCH	FETCH
OPEN	OPEN
PREPARE	PREPARE
UNKNOWN	UNKNOWN

Stmt Start The string date and time that the most recent SQL statement operation started. The value format is CYYMMDDHHMMSSmmm, where the following measures apply:

Value	Description
C	Century (0 for 20th, 1 for 21st)
YY	Year
MM	Month
DD	Day
HH	Hour
MM	Minute
SS	Second
mmm	Millisecond

Stmt Stop The string date and time that the most recent SQL statement operation stopped. The value format is a text string with a maximum of 16 characters. If the statement is still running, this field is 0 (zero). Use this attribute with the Statement Start attribute to calculate the elapsed execution time for the statement operation.

Stmt Text (Unicode) The text of the dynamic SQL statement (Unicode). For application snapshots, the statement text helps you identify what the application was executing when the snapshot was taken, or most recently processed if no statement was being processed at the time the snapshot was taken. For dynamic SQL statements, this attribute identifies the SQL text associated with a package. The value format is a text string with a maximum of 2000 bytes.

The statement text can only be fetched during the time when the dynamic SQL query is executing. If the query executing time is short, the statement text might not be retrieved.

Stmt Type The type of SQL statement processed. The value format is a text string with a maximum of 32 characters. The following values are valid:

External value	Internal value
DYNAMIC	DYNAMIC
NON-STATEMENT OPERATION	NON-STATEMENT_OPERATION
STATIC	STATIC
UNKNOWN STMT TYPE	UNKNOWN_STMT_TYPE

Table Name (Unicode) The name of the table the application is waiting to lock (Unicode). The value format is a text string with a maximum of 60 bytes. Use this attribute with the Table Schema attribute to determine the source of contention for resources.

Table Schema (Unicode) The schema of the table the application is waiting to lock (Unicode). The value format is a text string with a maximum of 60 bytes. Along with the Table Name attribute, this attribute can help to determine the source of contention for resources.

Tablespace Name (Unicode) the name of the tablespace that the application is waiting to lock (Unicode). The value format is a text string with a maximum of 60 bytes. This attribute can help you to determine the source of contention for resources.

Total Hash Joins The total number of hash joins that ran. The value format is an integer.

Total Hash Loops The total number of times that a single partition of a hash join was larger than the available sort heap space. The value format is an integer. Values for this attribute indicate inefficient execution of hash joins. This might indicate that the sort heap size is too small or the sort heap threshold is too small.

Total Sort Time The total elapsed time (in milliseconds) for all sorts that ran. The value format is an integer. at the database or application level, use this element with the Total Sorts attribute to calculate the average sort time. This average can indicate whether sorting is a performance concern.

Total Sorts The total number of sorts that ran. The value format is an integer. at the database or application level, use this value with the Sort Overflows attribute to calculate the percentage of sorts that need more heap space. You can also use it with the Total Sort Time attribute to calculate the average sort time. If the number of sort overflows is small with respect to the total sorts, increasing the sort heap size might have little impact on performance, unless this buffer size is increased substantially.

Total SQL Stmt The total number of dynamic and static SQL statements. This value is derived by adding the values of the Dynamic SQL Statements and the Static SQL Statements attributes.

UID SQL Stmts The number of SQL UPDATE, INSERT, and DELETE statements that ran. The value format is an integer.

UOW Comp Status The completion status of the previous UOW (unit of work). The value format is a text string with a maximum of 32 characters. The following values are valid:

External value	Internal value
Appl Normal Termination	Appl_Normal_Termination
UOW Commit	UOW_Commit
UOW RB - Lock Timeout	UOW_ RB_-_Lock_Timeout
UOW RB due to Abend	UOW_RB_due_To_Abend
UOW RB due to Deadlock	UOW_RB_due_to_Deadlock
UOW Rolled Back	UOW_Rolled_Back

External value	Internal value
0	0
Unknown	Unknown

UOW Lock Wait Time The time the UOW (unit of work) waited on locks (in seconds).

UOW Log Space Used The log space used in the most recent UOW (unit of work). Values that are greater than or equal to 2147483647 are indicated in the portal with the `Value Exceeds Maximum` text, and values that are smaller than -2147483648 are indicated with the `Value Exceeds Minimum` text. The following values are valid:

External value	Internal value
Value Exceeds Maximum	2147483647
Value Exceeds Minimum	-2147483648

UOW Start Time The string date and time that the unit of work first required database resources. This resource requirement occurs at the first SQL statement execution for the unit of work. The value format is CYYMMDDHHMMsss where the following measures apply:

Value	Description
C	Century (0 for 20th, 1 for 21st)
YY	Year
MM	Month
DD	Day
HH	Hour
MM	Minute
SS	Second
mmm	Millisecond

UOW Stop Time The string date and time that the most recent unit of work completed, which occurs when database changes are committed or rolled back. The value format is CYYMMDDHHMMsss where the following measures apply:

Value	Description
C	Century (0 for 20th, 1 for 21st)
YY	Year
MM	Month
DD	Day
HH	Hour
MM	Minute
SS	Second
mmm	Millisecond

X Lock Escals The number of times that locks have been escalated from several row locks to one exclusive table lock, or the number of times an exclusive lock on a row caused the table lock to become an exclusive lock. The value format is an integer. A lock is escalated when the total number of locks held by an application reaches the maximum amount of lock list space available to the application. The amount of lock list space available is determined by the LOCKLIST and MAXLOCKS configuration parameters. Other applications cannot access data held by an exclusive lock.

Application01 (KUD_DB2_Application01) attributes

The Application01 attribute group provides information about the database and the application. By using this information, you can determine the efficiency of the database and identify any problem areas for corrective action. All values are integers that are calculated from the first application connection, unless otherwise noted.

Agents Stolen The number of times that agents are stolen from an application. When another application requires a new subagent and has no subagents in its associated agent pool, it steals subagents from the agent pools of other applications. If the number of agents stolen from this application is high compared to normal operating levels, the number of pool agents might be too low. When the agent pool size is too small, one application might fill the pool with associated subagents. When another application requires a new subagent and has no subagents in its associated agent pool, it steals subagents from the agent pools of other applications. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Agent Sys CPU Time The total system CPU time (in seconds) that the database manager agent process spent executing database manager code. This element includes CPU time for both SQL and non-SQL statements, and CPU time for any unfenced user-defined functions (UDFs). The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Agent User CPU Time The total CPU time (in seconds) that the database manager agent process spent in system calls. This counter includes time spent on both SQL and non-SQL statements, and any unfenced user-defined functions (UDFs) or stored procedures issued by the application. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Appl ID The application ID.

Appl Name The name of the application running at the client as it is known to the database manager or DB2 Connect. The value format is a text string, with a maximum of 60 characters. For example: `*Local.db2inst1.990212202018`.

Appl Section Inserts The number of inserts of SQL sections by an application from its SQL work area. The working copy of any executable section is stored in a unique SQL work area. The returned value is a count of how many times a copy was not available and had to be inserted. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Appl Section Lookups The number of lookups of SQL sections by an application from its SQL work area. This counter indicates how many times the SQL work area for an application was accessed. The total is a cumulative figure of all lookups on all SQL work heaps for agents working on this application. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Appl Work Load The ratio of the maximum number of subagents associated with this application to the number of agents that are stolen from the application by DB2 to work on a different application. Use the returned value to evaluate the load that this application places on the system. An agent working for an application is associated with that application. After the agent completes the work for the application, it is placed in the agent pool as an idle agent, but it remains associated with the application. When the application requires an agent again, DB2 searches the agent pool for an agent already associated with the application and assigns work to the associated agent. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Associated Agents Top The maximum number of associated agents. The valid format is integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Sect Read per Direct Read The average number of sectors that are read by a direct read for the database. The value is derived through this formula:

```
direct reads / direct read reqs
```

Direct reads do not use the buffer pool, and so result in poor performance because the data is physically read from disk each time. If you are using system monitors to track input and output for the device, this value helps you distinguish database input and output from non-database input and output. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Sect Written per Direct Write The average number of sectors that are written in a direct write by this application. The valid format is integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

DB Name The real name of the database for which information is collected or to which the application is connected. This name was given to the database when it was created. The value format is a simple text string with a maximum of 60 characters. Use this attribute to identify the specific database to which the data applies.

DB Partition The DB2 database partition node number, which can range from 0 to 999. The Aggregated and Current Partition values can be used within a query or situation filter. If a db partition filter is not specified, data is returned for the current database partition. If a db partition filter is set to Aggregated, only aggregated partition data is returned. Historical data collection includes both aggregated and individual partition attribute data. In addition to numeric partition numbers in the 0 to 999 range, the following values are also valid:

External value	Internal value
Aggregated	-1
Current Partition	-2
All	-3

DDL SQL Percent for Interval The percentage of total SQL statements that are SQL DDL statements issued by the application during the monitoring interval. Due to the high activity in the system catalog tables, try to keep DDL statement activity to a minimum. If the returned value is high compared to normal operating levels, determine the activity causing it to be high and restrict it from being performed. Examples of DDL statements are CREATE TABLE, CREATE VIEW, ALTER TABLE, and DROP INDEX. You can also use the returned value to refine the package cache hit ratio for this application. DDL statements can also affect the package cache by invalidating sections that are stored there and causing additional system overhead due to section recompilation.

Deadlocks for Interval The total number of deadlocks that occurred for the application during the monitoring interval. Use the returned value to determine if the application is experiencing contention problems. Modify the application to better enable it to run concurrently. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Instance Name The name of the monitored DB2 instance. The valid format is a text string with a maximum of 60 bytes.

Lock Escalation for Interval The total number of lock escalations for the application during the monitoring interval. Exclusive lock escalations are included in this number. Use the returned value to help you evaluate the settings of the

LOCKLIST and MAXLOCKS configuration parameters. Lock escalations can result in a decrease in concurrency among the applications connected to a database. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Lock List in Use Percent The percentage of space used in the lock list by a connected application. The value format is a percentage. When an application reaches the maximum number of allowed locks and no additional locks are escalated, the application uses space in the lock list that is allocated for other applications. When an application holds too much of the lock list, other applications can experience lock escalations.

Lock Wait Time for Interval The total elapsed time, in seconds, that the application waited for a lock to be granted during the monitoring interval. The value format is an integer.

Node Name For new installations of version 6, release 2, the format is instanceid:hostname:UD for all operating systems. The format for version 6, release 1 of the DB2 agent on Windows systems is instanceid:hostname:UD; on UNIX and Linux systems, the format is instanceid:hostname.

Open Curs The number of local and remote cursors that are currently open for this application, including the number of local and remote blocking cursors currently open for this application. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Open Curs Blk The number of local and remote blocking cursors that are currently open for this application.

Pkg Cache Hit Percent The application package cache hit ratio (as a percentage) for the last monitoring interval. The package cache hit ratio is the ratio of the difference between the package cache lookups and the package cache inserts to all package cache lookups. This percentage tells you whether the package cache is being used efficiently by this application. If the hit ratio is high (greater than 80%), the package cache is performing well. A smaller percentage can indicate that the package cache must be increased. However, it is not always worthwhile to increase the size of the package cache for an application that runs only once a day. The size of the package cache is set by the pckcachesz configuration parameter.

Pool Hit Ratio Pct for Interval The overall buffer pool hit ratio (as a percentage) for the database during the monitoring interval. This hit ratio includes both index and data page activity. The overall buffer pool hit ratio indicates the percentage of page requests for which the database manager did not need to load a page from disk to service. (That is, the page was already in the buffer pool.) The greater the buffer pool hit ratio, the lower the frequency of disk input and output. If the hit ratio is low compared to normal operating levels, increasing the number of buffer pool pages can improve performance. A ratio of zero indicates that pages needed to be read for every request. For a large database, increasing the buffer pool size can have a minimal effect on the buffer pool hit ratio. Such a database can have so large a number of data pages that the statistical chance of a hit is not increased by an increase of the buffer pools. However, even though the data might be too large to fit in the buffer pool, the entire index can fit. In this case, you can refine buffer pool sizes until the overall buffer pool hit ratio stops increasing, and then refine the buffer pool until the buffer pool index hit ratio no longer increases.

Pool Index Hit Ratio Percent for Interval The application buffer pool index page hit ratio (as a percentage) during the monitoring interval. The index page hit ratio for the buffer pool indicates the percentage of index page requests for which the database manager did not need to load an index page from disk to service. That is, the index page was already in the buffer pool. The higher the returned value, the lower the frequency of disk input and output, and the faster the performance. If the hit ratio is low compared to normal operating levels, increasing the number of buffer pool pages can improve performance.

Prefetch Wait Time The time an application spent waiting for an I/O server (prefetcher) to finish loading pages into the buffer pool. The value format is an integer. This attribute can be used to experiment with changing the number of I/O servers and the I/O server sizes. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Prev UOW Stop Timestamp The date and time that the unit of work completed.

Section Number The internal section number in the package for the SQL statement currently processing or most recently processed. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Snapshot Timestamp The date and time when the database system monitor information was collected. Use this attribute to help relate data chronologically if you are saving the results in a file or database for ongoing analysis.

SQL Reqs Since Commit The number of SQL requests that were submitted by the application since the last commit. Use the returned value to monitor the progress of a transaction. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Stmts Sorts The total number of times that a set of data was sorted to process the OPEN operation of the current SQL statement. Use the returned value to help identify the need for an index, because indexes can reduce the need for sorting a set of data. Identify the SQL statement for which this returned value is providing sort information. Then, analyze this SQL statement to determine index candidates by looking at columns that are being sorted. For example, a column used in an ORDER BY clause can be an index candidate. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Stmt Start Timestamp The date and time that the most recent SQL statement operation started.

Stmt Stop Timestamp The date and time that the most recent SQL statement operation stopped. If the statement is still running, this field is 0 (zero). Use this attribute with the Statement Start attribute to calculate the elapsed execution time for the statement operation.

Total Pool IO Time The total time (in seconds) that an application spent performing buffer pool input and output operations (reading or writing pages). The returned value is an indication of how much time the application performs input and output operations using the buffer pool. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Total Sorts for Interval The total number of sorts that are issued by the application during the monitoring interval. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

UID SQL Percent for Interval The percentage of total SQL statements that are SQL UPDATE, INSERT, and DELETE statements issued by the application during the monitoring interval. Use the returned value to determine if the application performs frequent updates. If the returned value is low compared to normal operating levels, the application is query-based; otherwise, it is update-based. Knowing what type of applications you have (query-based or update-based) can aid you in refining the database configuration parameters.

UOW Comp Status The completion status of the previous UOW (unit of work). The value format is a text string with a maximum of 32 characters. The following values are valid:

External value	Internal value
Appl Normal Termination	Appl_Normal_Termination
UOW Commit	UOW_Commit
UOW RB - Lock Timeout	UOW_ RB_-_Lock_Timeout
UOW RB due to Abend	UOW_RB_due_To_Abend
UOW RB due to Deadlock	UOW_RB_due_to_Deadlock
UOW Rolled Back	UOW_Rolled_Back
0	0
Unknown	Unknown

UOW Log Space Used The log space used in the most recent UOW (unit of work). Values that are greater than or equal to 9223372036854775807 are indicated with the text `Value Exceeds Maximum` in the portal. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

UOW Start Timestamp The date and time that the unit of work first required database resources. This resource requirement occurs at the first SQL statement execution for the unit of work.

UOW Stop Timestamp The date and time that the most recent unit of work completed, which occurs when database changes are committed or rolled back.

Application01 (KUDDB2APPLGROUP01) attributes (Superseded)

The Application Information attribute group provides information about the database and the application. By using this information, you can determine the efficiency of the database and identify any problem areas for corrective action. All values are integers that are calculated from the first application connection, unless otherwise noted. This attribute group is superseded. There is a new attribute group with the same name that replaces it.

Agents Stolen The number of times that agents are stolen from an application. When another application requires a new subagent and has no subagents in its associated agent pool, it steals subagents from the agent pools of other applications. If the number of agents stolen from this application is high compared to normal operating levels, the number of pool agents might be too low. When the agent pool size is too small, one application might fill the pool with associated subagents. When another application requires a new subagent and has no subagents in its associated agent pool, it steals subagents from the agent pools of other applications.

Agent Sys CPU Time The total system CPU time (in seconds) that the database manager agent process spent executing database manager code. This element includes CPU time for both SQL and non-SQL statements, and CPU time for any unfenced user-defined functions (UDFs).

Agent User CPU Time The total CPU time (in seconds) that the database manager agent process spent in system calls. This counter includes time spent on both SQL and non-SQL statements, and any unfenced user-defined functions (UDFs) or stored procedures that are issued by the application.

Appl Connect Timestamp The date and time that an application started a connection request. The following special values cause the Tivoli Enterprise Portal to display fixed string messages:

Internal value	Tivoli Enterprise Portal display
0000000000000001	N/A (Not Applicable or Not Available)
0000000000000002	N/C (Not Calculated)
0000000000000003	N/P (Not Present)

Appl ID (Unicode) The application ID.

Appl Name The name of the application running at the client as it is known to the database manager or DB2 Connect. The value format is a text string, with a maximum of 20 characters. For example: `*Local.db2inst1.990212202018`.

Appl Name (Unicode) The name of the application running at the client as it is known to the database manager or DB2 Connect.

Appl Section Inserts The number of inserts of SQL sections by an application from its SQL work area. The working copy of any executable section is stored in a unique SQL work area. The returned value is a count of how many times a copy was not available and had to be inserted.

Appl Section Lookups The number of lookups of SQL sections by an application from its SQL work area. This counter indicates how many times the SQL work area for an application was accessed. The total is a cumulative figure of all lookups on all SQL work heaps for agents working on this application.

Appl Work Load The ratio of the maximum number of subagents associated with this application to the number of agents that are stolen from the application by DB2 to work on a different application. Use the returned value to evaluate the load that this application places on the system. An agent working for an application is associated with that application. After the agent completes the work for the application, it is placed in the agent pool as an idle agent, but it remains associated with the application. When the application requires an agent again, DB2 searches the agent pool for an agent already associated with the application and assigns work to the associated agent.

Associated Agents Top The maximum number of associated agents.

Avg Sect Read per Direct Read The average number of sectors that are read by a direct read for the database. The value is derived through this formula:

```
direct reads / direct read reqs
```

Direct reads do not use the buffer pool, and so result in poor performance because the data is physically read from disk each time. If you are using system monitors to track input and output for the device, this value helps you distinguish database input and output from non-database input and output.

Avg Sect Written per Direct Write The average number of sectors that are written in a direct write by this application.

Binds Precompiles The number of binds and precompiles attempted by an application. Use the returned value to determine the current level of activity in the database manager.

Connection Complete Timestamp The date and time that a connection request was granted. The following special values cause the Tivoli Enterprise Portal to display fixed string messages:

Internal value	Tivoli Enterprise Portal display
0000000000000001	N/A (Not Applicable or Not Available)
0000000000000002	N/C (Not Calculated)
0000000000000003	N/P (Not Present)

DB Name The real name of the database for which information is collected or to which the application is connected. This name was given to the database when it was created. The value format is a simple text string with a maximum of 20 characters. Use this attribute to identify the specific database to which the data applies.

DB Name (Unicode) The real name of the database for which information is collected or to which the application is connected (Unicode). This name was given to the database when it was created. The value format is a simple text string with a maximum of 60 bytes. Use this attribute to identify the specific database to which the data applies.

DB Partition The DB2 database partition node number, which can range from 0 to 999. The Aggregated and Current Partition values can be used within a query or situation filter. If a db partition filter is not specified, data is returned for the current database partition. If a db partition filter is set to Aggregated, only aggregated partition data is returned. Historical data collection includes both aggregated and individual partition attribute data. In addition to numeric partition numbers in the 0 to 999 range, the following values are also valid:

External value	Internal value
Aggregated	-1
Current Partition	-2
All	-3

DDL SQL Pct for Interval The percentage of total SQL statements that are SQL DDL statements issued by the application during the monitoring interval. Due to the high activity in the system catalog tables, try to keep DDL statement activity to a minimum. If the returned value is high compared to normal operating levels, determine the activity causing it to be high and restrict it from being performed. Examples of DDL statements are CREATE TABLE, CREATE VIEW, ALTER TABLE, and DROP INDEX. You can also use the returned value to refine the package cache hit ratio for this application. DDL statements can also affect the package cache by invalidating sections that are stored there and causing additional system overhead due to section recompilation.

Deadlocks for Interval The total number of deadlocks that occurred for the application during the monitoring interval. Use the returned value to determine if the application is experiencing contention problems. Modify the application to better enable it to run concurrently.

Instance Name (Unicode) The name of the monitored DB2 instance (Unicode). The valid format is a text string with a maximum of 60 bytes.

Lock Escalation for Interval The total number of lock escalations for the application during the monitoring interval. Exclusive lock escalations are included in this number. Use the returned value to help you evaluate the settings of the LOCKLIST and MAXLOCKS configuration parameters. Lock escalations can result in a decrease in concurrency among the applications connected to a database.

Lock List in Use Pct The percentage of space used in the lock list by a connected application. The value format is a percentage. When an application reaches the maximum number of allowed locks and no additional locks are escalated, the application uses space in the lock list that is allocated for other applications. When an application holds too much of the lock list, other applications can experience lock escalations.

Lock Wait Start Timestamp The date and time that the application started waiting to obtain a lock on the object that is currently locked by another application. The following special values cause the Tivoli Enterprise Portal to display fixed string messages:

Internal value	Tivoli Enterprise Portal display
0000000000000001	N/A (Not Applicable or Not Available)
0000000000000002	N/C (Not Calculated)
0000000000000003	N/P (Not Present)

Lock Wait Time for Interval The total elapsed time, in seconds, that the application waited for a lock to be granted during the monitoring interval. The value format is an integer.

Node Name For new installations of version 6, release 2, the format is instanceid:hostname:UD for all operating systems. The format for version 6, release 1 of the DB2 agent on Windows systems is instanceid:hostname:UD; on UNIX and Linux systems, the format is instanceid:hostname.

Open Curs The number of local and remote cursors that are currently open for this application, including the number of local and remote blocking cursors currently open for this application.

Open Curs Blk The number of local and remote blocking cursors that are currently open for this application.

Pkg Cache Hit Pct The application package cache hit ratio (as a percentage) for the last monitoring interval. The package cache hit ratio is the ratio of the difference between the package cache lookups and the package cache inserts to all package cache lookups. This percentage tells you whether the package cache is being used efficiently by this application. If the hit ratio is high (greater than 80%), the package cache is performing well. A smaller percentage can indicate that the package cache must be increased. However, it is not always worthwhile to increase the size of the package cache for an application that runs only once a day. The size of the package cache is set by the pckcachesz configuration parameter.

Pool Hit Ratio Pct for Interval The overall buffer pool hit ratio (as a percentage) for the database during the monitoring interval. This hit ratio includes both index and data page activity. The overall buffer pool hit ratio indicates the percentage of page requests for which the database manager did not need to load a page from disk to service. (That is, the page was already in the buffer pool.) The greater the buffer pool hit ratio, the lower the frequency of disk input and output. If the hit ratio is low compared to normal operating levels, increasing the number of buffer pool pages can improve performance. A ratio of zero indicates that pages needed to be read for every request. For a large database, increasing the buffer pool size can have a minimal effect on the buffer pool hit ratio. Such a database can have so large a number of data pages that the statistical chance of a hit is not increased by an increase of the buffer pools. However, even though the data might be too large to fit in the buffer pool, the entire index can fit. In this case, you can refine buffer pool sizes until the overall buffer pool hit ratio stops increasing, and then refine the buffer pool until the buffer pool index hit ratio no longer increases.

Pool Index Hit Ratio Pct for Interval The application buffer pool index page hit ratio (as a percentage) during the monitoring interval. The index page hit ratio for the buffer pool indicates the percentage of index page requests for which the database manager did not need to load an index page from disk to service. That is, the index page was already in the buffer pool. The higher the returned value, the lower the frequency of disk input and output, and the faster the performance. If the hit ratio is low compared to normal operating levels, increasing the number of buffer pool pages can improve performance.

Pool Total Reads (K) The total number of read requests in thousands (K) that required I/O operations to get data pages and index pages into the buffer pool. The value format is an integer. This attribute is the total of the Pool Data Physical Reads and Pool Index Physical Reads attributes. Values that are greater than or equal to 2147483647 are indicated in the portal with the `Value Exceeds Maximum` text, and values that are smaller than -2147483648 are indicated with the `Value Exceeds Minimum` text. The following values are valid:

External value	Internal value
Value Exceeds Maximum	2147483647
Value Exceeds Minimum	-2147483648

Pool Total Writes (K) The total number of write requests in thousands (K). The value format is an integer. This attribute is the total of the Pool Data Writes and Pool Index Writes attributes. Values that are greater than or equal to 2147483647 are indicated in the portal with the `Value Exceeds Maximum` text, and values that are smaller than -2147483648 are indicated with the `Value Exceeds Minimum` text. The following values are valid:

External value	Internal value
Value Exceeds Maximum	2147483647
Value Exceeds Minimum	-2147483648

Prev UOW Stop Timestamp The date and time that the unit of work completed. The following special values cause the Tivoli Enterprise Portal to display fixed string messages:

Internal value	Tivoli Enterprise Portal display
0000000000000001	N/A (Not Applicable or Not Available)
0000000000000002	N/C (Not Calculated)
0000000000000003	N/P (Not Present)

Snapshot Time The string date and time when the database system monitor information was collected. Use this attribute to help relate data chronologically if you are saving the results in a file or database for ongoing analysis. The Tivoli Enterprise Portal formats the timestamp value into a date and time string. The internal timestamp value stored in the database is in the format cYYMMDDhhmmss000, which is described in the following table:

Value	Description
C	Century (0 for 20th, 1 for 21st)
YY	Year (last two digits of the year, for example (00 - 99)
MM	Month (01 for January, 02 for February, and so on)
DD	Day (the day of the month, for example 01 - 31)
HH	Hour (the hour of the day in 24-hour format from 00 - 23)
MM	Minute (the minutes of the hour from 00 - 59)
SS	Second (the seconds of the hour (00 - 59)

Snapshot Timestamp The string date and time when the database system monitor information was collected. Use this attribute to help relate data chronologically if you are saving the results in a file or database for ongoing analysis. The Tivoli Enterprise Portal formats the timestamp value into a date and time string. The internal timestamp value stored in the database is in the format cYYMMDDhhmmss000, which is described in the following table:

Value	Description
C	Century (0 for 20th, 1 for 21st)
YY	Year (last two digits of the year, for example (00 - 99)
MM	Month (01 for January, 02 for February, and so on)
DD	Day (the day of the month, for example 01 - 31)
HH	Hour (the hour of the day in 24-hour format from 00 - 23)
MM	Minute (the minutes of the hour from 00 - 59)
SS	Second (the seconds of the hour (00 - 59)

The following special values cause the Tivoli Enterprise Portal to display fixed string messages:

Internal value	Tivoli Enterprise Portal display
0000000000000001	N/A (Not Applicable or Not Available)
0000000000000002	N/C (Not Calculated)
0000000000000003	N/P (Not Present)

SQL Reqs Since Commit The number of SQL requests that were submitted by the application since the last commit. Use the returned value to monitor the progress of a transaction.

Stmt Start Timestamp The date and time that the most recent SQL statement operation started. The following special values cause the Tivoli Enterprise Portal to display fixed string messages:

Internal value	Tivoli Enterprise Portal display
0000000000000001	N/A (Not Applicable or Not Available)
0000000000000002	N/C (Not Calculated)
0000000000000003	N/P (Not Present)

Stmt Stop Timestamp The date and time that the most recent SQL statement operation stopped. The following special values cause the Tivoli Enterprise Portal to display fixed string messages:

Internal value	Tivoli Enterprise Portal display
0000000000000001	N/A (Not Applicable or Not Available)
0000000000000002	N/C (Not Calculated)
0000000000000003	N/P (Not Present)

Stmts Sorts The total number of times that a set of data was sorted to process the OPEN operation of the current SQL statement. Use the returned value to help identify the need for an index, because indexes can reduce the need for sorting a set of data. Identify the SQL statement for which this returned value is providing sort information. Then, analyze this SQL statement to determine index candidates by looking at columns that are being sorted. For example, a column used in an ORDER BY clause can be an index candidate.

Total Pool IO Time The total time (in seconds) that an application spent performing buffer pool input and output operations (reading or writing pages). The returned value is an indication of how much time the application performs input and output operations using the buffer pool.

Total Sorts for Interval The total number of sorts that are issued by the application during the monitoring interval. The value format is an integer.

UID SQL Pct for Interval The percentage of total SQL statements that are SQL UPDATE, INSERT, and DELETE statements issued by the application during the monitoring interval. Use the returned value to determine if the application performs frequent updates. If the returned value is low compared to normal operating levels, the application is query-based; otherwise, it is update-based. Knowing what type of applications you have (query-based or update-based) can aid you in refining the database configuration parameters.

UOW Log Space Used (MB) The amount of log space (in MB) used in the current unit of work of the monitored application. The value format is an integer. Values that are greater than or equal to 2147483647 are indicated in the portal with the `Value Exceeds Maximum` text, and values that are smaller than -2147483648 are indicated with the `Value Exceeds Minimum` text. Use this attribute to understand the logging requirements at the unit-of-work level. The following values are valid:

External value	Internal value
Value Exceeds Maximum	2147483647
Value Exceeds Minimum	-2147483648

UOW Start Timestamp The date and time that the unit of work first required database resources. The following special values cause the Tivoli Enterprise Portal to display fixed string messages:

Internal value	Tivoli Enterprise Portal display
0000000000000001	N/A (Not Applicable or Not Available)
0000000000000002	N/C (Not Calculated)
0000000000000003	N/P (Not Present)

UOW Stop Timestamp The date and time that the most recent unit of work completed, which occurs when database changes are committed or rolled back. The following special values cause the Tivoli Enterprise Portal to display fixed string messages:

Internal value	Tivoli Enterprise Portal display
0000000000000001	N/A (Not Applicable or Not Available)
0000000000000002	N/C (Not Calculated)
0000000000000003	N/P (Not Present)

Apply Program (KUD_DB2_Apply_Program) attributes

The Apply Program attributes provide status information about the Apply Program processes that are configured to run on a database manager server. To collect Apply Program attributes, the Apply Program must be configured successfully. The DB2 agent must be located on the control server to collect Apply Program attributes. The control server is often the same as the target database server in an Apply subscription set.

Apply ID The subscriber user ID that started the Apply Program.

Apply Qualifier Uniquely identifies which Apply Program processes this subscription set.

Apply Status Indicates the state of the Apply subscription process for every distinct apply ID in the Apply Program subscription sets. The following values are valid:

External value	Internal value
Up	1
Down	0

DB Name The database name on the Apply control server where the subscription set table is stored.

Instance Name The name of the monitored DB2 instance. The valid format is a text string with a maximum of 60 bytes.

Node Name For new installations of version 6, release 2, the format is instanceid:hostname:UD for all operating systems. The format for version 6, release 1 of the DB2 agent on Windows systems is instanceid:hostname:UD; on UNIX and Linux systems, the format is instanceid:hostname.

Snapshot Timestamp The date and time when the database system monitor information was collected. Use this attribute to help relate data chronologically if you are saving the results in a file or database for ongoing analysis.

Total Apply Sub Fail The number of subscriptions with the same apply ID that the Apply Program failed to replicate. This number includes only active subscriptions that failed with a status that is equal to -1. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Total Apply Sub Fail (Superseded) The number of subscriptions with the same apply ID that the Apply Program failed to replicate. This number includes only active subscriptions that failed with a status that is equal to -1.

Total Apply Sub Lag The total number of Apply Program subscriptions that have not completed within their scheduled replication interval. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Total Apply Sub Lag (Superseded) The total number of Apply Program subscriptions that have not completed within their scheduled replication interval.

Apply Subscription (KUD_DB2_Apply_Subscription) attributes

The Apply Subscription attributes provide information about Apply Program subscription sets that are configured to run on a database manager server. To collect Apply Program attributes successfully, the Apply Program must be configured. The DB2 agent must be located on the control server to collect Apply Program attributes. The control server is often the same as the target database server in an Apply subscription set.

Apply ID Subscriber user ID that started the Apply Program.

Apply Num Reqs Refresh Indicates the number of subscriptions the Apply Program failed to replicate because refresh copying was disabled. While attempting to perform a full refresh, the Apply Program encountered a DISABLE_REFRESH column in the register table, which was set to `On`. You can either turn off the DISABLE_REFRESH column or bypass the Apply Program and perform a manual refresh. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Apply Num Reqs Refresh (Superseded) Indicates the number of subscriptions the Apply Program failed to replicate because refresh copying was disabled. While attempting to perform a full refresh, the Apply Program encountered a DISABLE_REFRESH column in the register table, which was set to `On`. You can either turn off the DISABLE_REFRESH column or bypass the Apply Program and perform a manual refresh.

Apply Sub Lag Time The difference (in number of minutes) between how much time has elapsed since the last run of the Apply Program and the expected sleep interval between executions of the Apply Program for the target table. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Apply Sub Lag Time (Superseded) The difference (in number of minutes) between how much time has elapsed since the last run of the Apply Program and the expected sleep interval between executions of the Apply Program for the target table.

Apply Sub Status The Apply Program subscription status. The following values are valid:

External value	Internal value	Description
No Errors	0	No errors
Replication failed	-1	Replication failed
Successful single set processing multiple cycles	2	Successful single set processing multiple cycles
Some errors	16	Some errors
Some errors processing multiple cycles	18	Some errors processing multiple cycles

DB Name Target Database name.

Instance Name The name of the monitored DB2 instance. The valid format is a text string with a maximum of 60 bytes.

Node Name For new installations of version 6, release 2, the format is instanceid:hostname:UD for all operating systems. The format for version 6, release 1 of the DB2 agent on Windows systems is instanceid:hostname:UD; on UNIX and Linux systems, the format is instanceid:hostname.

Snapshot Timestamp The date and time when the database system monitor information was collected. Use this attribute to help relate data chronologically if you are saving the results in a file or database for ongoing analysis.

Target Owner The name of the target owner for this member.

Target Table The name of the target table or view for this member.

Buffer Pool (KUD_DB2_Buffer_Pool) attributes

The Buffer Pool attributes provide information about buffer pool activities. You can use this information to monitor the performance of your buffer pools and to identify problem areas for corrective action. All values are integers that are calculated from the first application connection, unless otherwise noted.

Avg Data Page Read per Async Req The average number of pages read for each asynchronous request. This value is derived by dividing the value of the Pool Async Data Reads attribute by the value of the Pool Async Data Read Reqs attribute. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Direct Read Time The average elapsed time for a direct read request. This value is calculated by dividing the value of the Direct Read Time attribute by the value of the the Direct Reads attribute. This average is important because it might indicate the presence of an I/O wait, which in turn might indicate that you must move data to a different device. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Direct Write Time The average elapsed time for a direct write request. This value is calculated by dividing the value of the Direct Write Time attribute by the value of the Direct Writes attribute. This average is important because it might indicate the presence of an I/O wait, which in turn might indicate that you must move data to a different device. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Pool Read Time The average elapsed time for a read request. This value is derived by dividing the value of the Pool Read Time attribute by the value of the Pool Total Reads attribute. This average is important because it might indicate the presence of an I/O wait, which in turn might indicate that you must move data to a different device. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Pool Write Time The average elapsed time for a write request. This value is derived by dividing the value of the Pool Write Time attribute by the value of the Pool Total Writes attribute. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Sync Read Time The average elapsed time used to perform a synchronous read. This value is derived by dividing the value of the Pool Sync Read Time attribute by the value of the Pool Sync Read attribute. This average is important because it might indicate the presence of an I/O wait, which in turn might indicate that you must move data to a different device. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Sync Write Time The average elapsed time used to perform a synchronous write. This value is derived by dividing the value of the Pool Sync Write Time attribute by the value of the Pool Sync Write attribute. This average is important because it might indicate the presence of an I/O wait, which in turn might indicate that you must move data to a different device. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

BP ID The internal identifier for the buffer pool. This attribute is for internal use only.

BP Name The name of the buffer pool. A new database has a default buffer pool (named IBMDEFAULTBP). The size of the default buffer pool is determined by the platform. Depending on your needs you might choose to create several buffer pools, each of a different size, for a single database. The CREATE, ALTER, and DROP BUFFERPOOL statements allow you to create, change, or remove a buffer pool.

DB Name The real name of the database for which information is collected or to which the application is connected. This name was given to the database when it was created. The value format is a simple text string with a maximum of 60 characters. Use this attribute to identify the specific database to which the data applies.

DB Partition The DB2 database partition node number, which can range from 0 to 999. The Aggregated and Current Partition values can be used within a query or situation filter. If a db partition filter is not specified, data is returned for the current database partition. If a db partition filter is set to Aggregated, only aggregated partition data is returned. Historical data collection includes both aggregated and individual partition attribute data. In addition to numeric partition numbers in the 0 to 999 range, the following values are also valid:

External value	Internal value
Aggregated	-1
Current Partition	-2
All	-3

DB Path The full path of the location where the database is stored on the monitored system. Use this attribute with the Database Name attribute to identify the specific database to which the data applies.

Direct Read Reqs The number of requests to perform a direct read of one or more sectors of data. Use the following formula to calculate the average number of sectors that are read by a direct read:

```
direct reads from database / direct read requests
```

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Direct Read Time The elapsed time (in milliseconds) required to perform the direct reads. Use the following formula to calculate the average direct read time per sector:

```
direct read time / direct reads from database
```

A high average time might indicate an I/O conflict.

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Direct Reads The number of read operations that do not use the buffer pool. Use the following formula to calculate the average number of sectors that are read by a direct read:

```
direct reads from database / direct read requests
```

When using system monitors to track I/O, this data attribute helps to distinguish database I/O from non-database I/O on the device.

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Direct Write Reqs The number of requests to perform a direct write of one or more sectors of data. Use the following formula to calculate the average number of sectors that are written by a direct write:

```
direct writes to database / direct write requests
```

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Direct Write Time The elapsed time (in milliseconds) required to perform the direct writes. Use the following formula to calculate the average direct write time per sector:

```
direct write time / direct writes to database
```

A high average time might indicate an I/O conflict.

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Direct Writes The number of write operations that do not use the buffer pool. Use the following formula to calculate the average number of sectors that are written by a direct write:

```
direct writes to database / direct write requests
```

When using system monitors to track I/O, this data attribute helps to distinguish database I/O from non-database I/O on the device.

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Files Closed The total number of database files closed. The database manager opens files for reading and writing into and out of the buffer pool. The maximum number of database files open by an application at any time is controlled by the MAXFILOP configuration parameter. If the maximum is reached, one file is closed before the new file is opened. Note that the actual number of files opened might not equal the number of files closed. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Input DB Alias The alias of the database provided when calling the snapshot function. The value format is a simple text string with a maximum of 60 characters. Use this attribute to help you identify the specific database to which the monitor data applies. It contains blanks unless you requested monitor information related to a specific database.

Node Name For new installations of version 6, release 2, the format is instanceid:hostname:UD for all operating systems. The format for version 6, release 1 of theDB2 agent on Windows systems is instanceid:hostname:UD; on UNIX and Linux systems, the format is instanceid:hostname.

Pool Async Data Read Reqs The number of asynchronous read requests. To calculate the average number of data pages read per asynchronous request, use the following formula:

```
buffer pool asynchronous data reads / buffer pool asynchronous read requests
```

This average can help to determine the amount of asynchronous I/O in each interaction with the prefetcher.

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Async Data Reads The number of pages read asynchronously into the buffer pool. Use this attribute with the Buffer Pool Data Physical Reads attribute to calculate the number of physical reads that were performed synchronously (that is, physical data page reads that were performed by database manager agents). Use the following formula:

```
buffer pool data physical reads - buffer pool synchronous data reads
```

By comparing the ratio of asynchronous to synchronous reads, you can gain insight into how well the prefetchers are working.

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Async Data Writes The number of times a buffer pool data page was physically written to disk by an asynchronous page cleaner or by a prefetcher. A prefetcher might have written dirty pages to disk to make space for the pages being prefetched. Use this attribute with the Buffer Pool Data Writes attribute to calculate the number of physical write requests that were performed synchronously (that is, physical data page writes that were performed by database manager agents). Use the following formula:

```
buffer pool data writes - buffer pool asynchronous data writes
```

By comparing the ratio of asynchronous to synchronous writes, you can gain insight into how well the buffer pool page cleaners are performing.

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Async Index Reads The number of index pages read asynchronously into the buffer pool by a prefetcher. Asynchronous reads are performed by database manager prefetchers. Use this attribute with the Buffer Pool Index Physical Reads attribute to calculate the number of physical reads that were performed synchronously (that is, physical index page reads that were performed by database manager agents). Use the following formula:

```
buffer pool index physical reads - buffer pool asynchronous index reads
```

By comparing the ratio of asynchronous to synchronous reads, you can gain insight into how well the prefetchers are working.

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Async Index Writes The number of times a buffer pool index page was physically written to disk by an asynchronous page cleaner or a prefetcher. A prefetcher might have written dirty pages to disk to make space for the pages being prefetched. Use this attribute with the Buffer Pool Index Writes attribute to calculate the number of physical index write requests that were performed synchronously. That is, physical index page writes that were performed by database manager agents. Use the following formula:

```
buffer pool index writes - buffer pool asynchronous index writes
```

By comparing the ratio of asynchronous to synchronous writes, you can gain insight into how well the buffer pool page cleaners are performing.

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Async Read Time The total elapsed time spent reading by database manager prefetchers. Use this attribute to calculate the elapsed time for synchronous reading, using the following formula:

```
total buffer pool physical read time - buffer pool synchronous read time
```

You can also use this attribute to calculate the average asynchronous read time using the following formula:

```
buffer pool asynchronous read time / buffer pool asynchronous data reads
```

These calculations can be used to understand the I/O work being performed.

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Async Write Time The total elapsed time spent writing data or index pages from the buffer pool to disk by database manager page cleaners. Calculate the elapsed time spent writing pages synchronously by subtracting the value of the Pool Async Write Time attribute from the value of the Pool Physical Write Time attribute. You can also use this attribute to calculate the average asynchronous read time:

1. Sum the value of the Pool Async Data Writes attribute and the value of the Pool Async Index Writes attribute.
2. Divide the value of the Pool Async Write Time attribute by the sum from step 1.

These calculations can be used to understand the I/O work being performed.

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Data from Estore Number of buffer pool data pages copied from extended storage. Required pages are copied from extended storage to the buffer pool. The copy process might incur the cost of connecting to the shared memory segment, but it saves the cost of a disk read. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Data L Reads The number of logical read requests for data pages that have gone through the buffer pool. This count includes accesses to the following data:

- Data that is already in the buffer pool when the database manager needs to process the page.
- Data that is read into the buffer pool before the database manager can process the page.

By using the Pool Data Physical Reads attribute, you can calculate the data page hit ratio for the buffer pool as follows:

```
1 - (buffer pool data physical reads / buffer pool data logical reads)
```

By using the Pool Data Physical Reads, Pool Index Physical Reads, and Pool Index Logical Reads attributes, you can calculate the overall buffer pool hit ratio as follows:

```
1 - ((buffer pool data physical reads + buffer pool index physical reads)
/ (buffer pool data logical reads + buffer pool index logical reads))
```

Increasing buffer pool size generally improves the hit ratio until you reach a point of diminishing returns.

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Data P Reads The number of read requests that required I/O to get data pages into the buffer pool.

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Data to Estore Number of buffer pool data pages copied to extended storage. Pages are copied from the buffer pool to extended storage when they are selected as victim pages. As a result of the copying process, there is sufficient space for new pages in the buffer pool. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Data Writes The number of times a buffer pool data page was physically written to disk. A buffer pool data page is written to disk for the following reasons:

- To free a page in the buffer pool so another page can be read
- To flush the buffer pool.

If a buffer pool data page is written to disk for a high percentage of Buffer Pool Data Physical Reads, performance might improve by increasing the number of buffer pool pages available for the database.

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Hit Ratio The buffer pool hit ratio (as a percentage). The sum of the Pool Data Logical Reads and Pool Index Logical Reads attributes is divided by the value of the Pool Total Reads attribute to derive the pool hit ratio. This attribute can determine whether buffer pool assignment is efficient. If the pool hit ratio is low, increasing the number of buffer pool pages might improve performance.

Pool Index from Estore Number of buffer pool index pages copied from extended storage. Required index pages are copied from extended storage to the buffer pool. The copy process might incur the cost of connecting to the shared memory segment, but it saves the cost of a disk read. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Index L Reads The number of logical read requests for index pages that have gone through the buffer pool. This count includes accesses to the following index pages:

- Pages that are already in the buffer pool when the database manager needs to process the page.
- Pages that are read into the buffer pool before the database manager can process the page.

By using the Buffer Pool Index Physical Reads attribute, you can calculate the index page hit ratio for the buffer pool as follows:

```
1 - (buffer pool index physical reads / buffer pool index logical reads)
```

If the hit ratio is low, increasing the number of buffer pool pages might improve performance.

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Index P Reads The number of physical read requests to get index pages into the buffer pool. see the Pool Index Logical Reads attribute for information about how to use this element. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Index to Estore Number of buffer pool index pages copied to extended storage. Pages are copied from the buffer pool to extended storage when they are selected as victim pages. As a result of the copying process, there is sufficient space for new pages in the buffer pool. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Index Writes The number of times a buffer pool index page was physically written to disk. If a buffer pool index page is written to disk for a high percentage of Buffer Pool Index Physical Reads, performance might improve by increasing the number of buffer pool pages available for the database. If all applications are updating the database, increasing the size of the buffer pool might have minimal impact on performance; most pages contain updated data that must be written to disk. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Read Time The total amount of elapsed time spent processing read requests that caused data or index pages to be physically read from buffer pool to disk. Use this attribute with the Buffer Pool Data Physical Reads and Buffer Pool Index Physical Reads attributes to calculate the average page-read time. This average is important because it might indicate the presence of an I/O wait, which in turn might indicate that you must move data to a different device. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Sync Data Reads The number of physical data page reads that were performed by database manager agents. This value is derived by subtracting the value of the Pool Async Data Reads attribute from the Pool Data Physical Reads attribute. By comparing the ratio of asynchronous to synchronous reads, you can gain insight into how well the prefetchers are working. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Sync Data Writes The total number of physical write requests that were performed synchronously (that is, physical data page writes that were performed by database manager agents). This value is derived by subtracting the value of the Pool Async Data Writes attribute from the value of the Pool Data Writes attribute. By comparing the ratio of asynchronous to synchronous writes, you can gain insight into how well the buffer pool page cleaners are performing. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Sync Index Reads The number of index pages read synchronously (that is, physical index page reads that were performed by database manager agents) into the buffer pool. This value is derived by subtracting the value of the Pool Async Index Reads attribute from Pool Index Physical Reads attribute. By comparing the ratio of asynchronous to synchronous reads, you can gain insight into how well the prefetchers are working. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Sync Index Writes The number of physical index write requests that were performed synchronously (that is, physical index page writes that were performed by database manager agents). This value is derived by subtracting the value of the Pool Async Index Writes attribute from the value of the Pool Index Writes attribute. By comparing the ratio of asynchronous to synchronous writes, you can gain insight into how well the buffer pool page cleaners are performing. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Sync Read The total number of synchronous reads. This value is derived by adding the values of the Pool Sync Data Reads and Pool Sync Index Reads attributes. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Sync Read Time The elapsed time used to perform all synchronous reads. This value is derived by subtracting the value of the Pool Async Read Time attribute from the value of the Pool Read Time attribute. Use this attribute to understand the I/O work being performed. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Sync Write The total number of synchronous index writes. The value is derived by adding the values of the Pool Sync Data Writes attribute and Pool Sync Index Writes attribute. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Sync Write Time The total elapsed time used to perform all synchronous writes. This value is derived by subtracting the value of the Pool Async Write Time attribute from the value of the Pool Write Time attribute. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Total Reads The total number of read requests that required I/O to get data pages and index pages into the buffer pool. This attribute is the total of the Pool Data Physical Reads and Pool Index Physical Reads attributes. Values that are greater than or equal to 9223372036854775807 are indicated with the `Value Exceeds Maximum` text in the portal. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Total Writes The total number of write requests. This attribute is the total of the Pool Data Writes and Pool Index Writes attributes. Values that are greater than or equal to 9223372036854775807 are indicated with the `Value Exceeds Maximum` text in the portal. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Write Time The total amount of time spent physically writing data or index pages from the buffer pool to disk. Use this attribute with the Buffer Pool Data Writes and Buffer Pool Index Writes attributes to calculate the average page-write time. This average is important because it might indicate the presence of an I/O wait, which in turn might indicate that you must move data to a different device. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Snapshot Timestamp The date and time when the database system monitor information was collected. Use this attribute to help relate data chronologically if you are saving the results in a file or database for ongoing analysis.

Buffer Pool (KUDBUFFERPOOL00) attributes (Superseded)

The Buffer Pool attributes provide information about buffer pool activities. You can use this information to monitor the performance of your buffer pools and to identify problem areas for corrective action. All values are integers that are calculated from the first application connection, unless otherwise noted. This attribute group is superseded. There is a new attribute group with the same name that replaces it.

Avg Data Page Read per Async Req The average number of pages read for each asynchronous request. This value is derived by dividing the value of the Pool Async Data Reads attribute by the value of the Pool Async Data Read Reqs attribute.

Avg Direct Read Time The average elapsed time for a direct read request. This value is calculated by dividing the value of the Direct Read Time attribute by the value of the the Direct Reads attribute. This average is important because it might indicate the presence of an I/O wait, which in turn might indicate that you must move data to a different device.

Avg Direct Write Time The average elapsed time for a direct write request. This value is calculated by dividing the value of the Direct Write Time attribute by the value of the Direct Writes attribute. This average is important because it might indicate the presence of an I/O wait, which in turn might indicate that you must move data to a different device.

Avg Pool Read Time The average elapsed time for a read request. This value is derived by dividing the value of the Pool Read Time attribute by the value of the Pool Total Reads attribute. This average is important because it might indicate the presence of an I/O wait, which in turn might indicate that you must move data to a different device.

Avg Pool Write Time The average elapsed time for a write request. This value is derived by dividing the value of the Pool Write Time attribute by the value of the Pool Total Writes attribute.

Avg Sync Read Time The average elapsed time used to perform a synchronous read. This value is derived by dividing the value of the Pool Sync Read Time attribute by the value of the Pool Sync Read attribute. This average is important because it might indicate the presence of an I/O wait, which in turn might indicate that you must move data to a different device.

Avg Sync Write Time The average elapsed time used to perform a synchronous write. This value is derived by dividing the value of the Pool Sync Write Time attribute by the value of the Pool Sync Write attribute. This average is important because it might indicate the presence of an I/O wait, which in turn might indicate that you must move data to a different device.

BP ID The internal identifier for the buffer pool. This attribute is for internal use only.

BP Name The name of the buffer pool. A new database has a default buffer pool (named IBMDEFAULTBP). The size of the default buffer pool is determined by the platform. Depending on your needs you might choose to create several buffer pools, each of a different size, for a single database. The CREATE, ALTER, and DROP BUFFERPOOL statements allow you to create, change, or remove a buffer pool.\

BP Name (Unicode) The name of the buffer pool (Unicode). A new database has a default buffer pool (named IBMDEFAULTBP). The size of the default buffer pool is determined by the platform. Depending on your needs you might choose to create several buffer pools, each of a different size, for a single database. The CREATE, ALTER, and DROP BUFFERPOOL statements allow you to create, change, or remove a buffer pool.\

DB Name The real name of the database for which information is collected or to which the application is connected. This name was given to the database when it was created. The value format is a simple text string with a maximum of 20 characters. Use this attribute to identify the specific database to which the data applies.

DB Name (Unicode) The real name of the database for which information is collected or to which the application is connected (Unicode). This name was given to the database when it was created. The value format is a simple text string with a maximum of 60 bytes.

DB Partition The DB2 database partition node number, which can range from 0 to 999. The Aggregated and Current Partition values can be used within a query or situation filter. If a db partition filter is not specified, data is returned for the current database partition. If a db partition filter is set to Aggregated, only aggregated partition data is returned. Historical data collection includes both aggregated and individual partition attribute data. In addition to numeric partition numbers in the 0 to 999 range, the following values are also valid:

External value	Internal value
Aggregated	-1
Current Partition	-2
All	-3

DB Path The full path of the location where the database is stored on the monitored system. Use this attribute with the Database Name attribute to identify the specific database to which the data applies.

DB Path (Unicode) The full path of the location where the database is stored on the monitored system (Unicode).

Direct Read Reqs The number of requests to perform a direct read of one or more sectors of data. Use the following formula to calculate the average number of sectors that are read by a direct read:

```
direct reads from database / direct read requests
```

Direct Read Time The elapsed time (in milliseconds) required to perform the direct reads. Use the following formula to calculate the average direct read time per sector:

```
direct read time / direct reads from database
```

A high average time might indicate an I/O conflict.

Direct Reads The number of read operations that do not use the buffer pool. Use the following formula to calculate the average number of sectors that are read by a direct read:

```
direct reads from database / direct read requests
```

When using system monitors to track I/O, this data attribute helps to distinguish database I/O from non-database I/O on the device.

Direct Write Reqs The number of requests to perform a direct write of one or more sectors of data. Use the following formula to calculate the average number of sectors that are written by a direct write:

```
direct writes to database / direct write requests
```

Direct Write Time The elapsed time (in milliseconds) required to perform the direct writes. Use the following formula to calculate the average direct write time per sector:

```
direct write time / direct writes to database
```

A high average time might indicate an I/O conflict.

Direct Writes The number of write operations that do not use the buffer pool. Use the following formula to calculate the average number of sectors that are written by a direct write:

```
direct writes to database / direct write requests
```

When using system monitors to track I/O, this data attribute helps to distinguish database I/O from non-database I/O on the device.

Files Closed The total number of database files closed. The database manager opens files for reading and writing into and out of the buffer pool. The maximum number of database files open by an application at any time is controlled by the MAXFILOP configuration parameter. If the maximum is reached, one file is closed before the new file is opened. Note that the actual number of files opened might not equal the number of files closed.

Input DB Alias The alias of the database provided when calling the snapshot function. The value format is a simple text string with a maximum of 20 characters. Use this attribute to help you identify the specific database to which the monitor data applies. It contains blanks unless you requested monitor information related to a specific database.

Input DB Alias (Unicode) The alias of the database provided when calling the snapshot function (Unicode). The value format is a simple text string with a maximum of 60 bytes.

Node Name For new installations of version 6, release 2, the format is instanceid:hostname:UD for all operating systems. The format for version 6, release 1 of the DB2 agent on Windows systems is instanceid:hostname:UD; on UNIX and Linux systems, the format is instanceid:hostname.

Pool Async Data Read Reqs The number of asynchronous read requests. To calculate the average number of data pages read per asynchronous request, use the following formula:

```
buffer pool asynchronous data reads / buffer pool asynchronous read requests
```

This average can help to determine the amount of asynchronous I/O in each interaction with the prefetcher.

Pool Async Data Reads The number of pages read asynchronously into the buffer pool. Use this attribute with the Buffer Pool Data Physical Reads attribute to calculate the number of physical reads that were performed synchronously (that is, physical data page reads that were performed by database manager agents). Use the following formula:

```
buffer pool data physical reads - buffer pool synchronous data reads
```

By comparing the ratio of asynchronous to synchronous reads, you can gain insight into how well the prefetchers are working.

Pool Async Data Writes The number of times a buffer pool data page was physically written to disk by an asynchronous page cleaner or by a prefetcher. A prefetcher might have written dirty pages to disk to make space for the pages being prefetched. Use this attribute with the Buffer Pool Data Writes attribute to calculate the number of physical write requests that were performed synchronously (that is, physical data page writes that were performed by database manager agents). Use the following formula:

```
buffer pool data writes - buffer pool asynchronous data writes
```

By comparing the ratio of asynchronous to synchronous writes, you can gain insight into how well the buffer pool page cleaners are performing.

Pool Async Index Reads The number of index pages read asynchronously into the buffer pool by a prefetcher. Asynchronous reads are performed by database manager prefetchers. Use this attribute with the Buffer Pool Index Physical Reads attribute to calculate the number of physical reads that were performed synchronously (that is, physical index page reads that were performed by database manager agents). Use the following formula:

```
buffer pool index physical reads - buffer pool asynchronous index reads
```

By comparing the ratio of asynchronous to synchronous reads, you can gain insight into how well the prefetchers are working.

Pool Async Index Writes The number of times a buffer pool index page was physically written to disk by an asynchronous page cleaner or a prefetcher. A prefetcher might have written dirty pages to disk to make space for the pages being prefetched. Use this attribute with the Buffer Pool Index Writes attribute to calculate the number of physical index write requests that were performed synchronously. That is, physical index page writes that were performed by database manager agents. Use the following formula:

```
buffer pool index writes - buffer pool asynchronous index writes
```

By comparing the ratio of asynchronous to synchronous writes, you can gain insight into how well the buffer pool page cleaners are performing.

Pool Async Read Time The total elapsed time spent reading by database manager prefetchers. Use this attribute to calculate the elapsed time for synchronous reading, using the following formula:

```
total buffer pool physical read time - buffer pool synchronous read time
```

You can also use this attribute to calculate the average asynchronous read time using the following formula:

```
buffer pool asynchronous read time / buffer pool asynchronous data reads
```

These calculations can be used to understand the I/O work being performed.

Pool Async Write Time The total elapsed time spent writing data or index pages from the buffer pool to disk by database manager page cleaners. Calculate the elapsed time spent writing pages synchronously by subtracting the value of the Pool Async Write Time attribute from the value of the Pool Physical Write Time attribute. You can also use this attribute to calculate the average asynchronous read time:

1. Sum the value of the Pool Async Data Writes attribute and the value of the Pool Async Index Writes attribute
2. Divide the value of the Pool Async Write Time attribute by the sum from step 1.

These calculations can be used to understand the I/O work being performed.

Pool Data from Estore Number of buffer pool data pages copied from extended storage. Required pages are copied from extended storage to the buffer pool. The copy process might incur the cost of connecting to the shared memory segment, but it saves the cost of a disk read.

Pool Data L Reads The number of logical read requests for data pages that have gone through the buffer pool. This count includes accesses to the following data:

- Data that is already in the buffer pool when the database manager needs to process the page.
- Data that is read into the buffer pool before the database manager can process the page.

By using the Pool Data Physical Reads attribute, you can calculate the data page hit ratio for the buffer pool as follows:

```
1 - (buffer pool data physical reads / buffer pool data logical reads)
```

By using the Pool Data Physical Reads, Pool Index Physical Reads, and Pool Index Logical Reads attributes, you can calculate the overall buffer pool hit ratio as follows:

```
1 - ((buffer pool data physical reads + buffer pool index physical reads)
/ (buffer pool data logical reads + buffer pool index logical reads))
```

Increasing buffer pool size generally improves the hit ratio until you reach a point of diminishing returns.

Pool Data P Reads The number of read requests that required I/O to get data pages into the buffer pool.

Pool Data to Estore Number of buffer pool data pages copied to extended storage. Pages are copied from the buffer pool to extended storage when they are selected as victim pages. As a result of the copying process, there is sufficient space for new pages in the buffer pool.

Pool Data Writes The number of times a buffer pool data page was physically written to disk. A buffer pool data page is written to disk for the following reasons:

- To free a page in the buffer pool so another page can be read
- To flush the buffer pool.

If a buffer pool data page is written to disk for a high percentage of Buffer Pool Data Physical Reads, performance might improve by increasing the number of buffer pool pages available for the database.

Pool Hit Ratio The buffer pool hit ratio (as a percentage). The sum of the Pool Data Logical Reads and Pool Index Logical Reads attributes is divided by the value of the Pool Total Reads attribute to derive the pool hit ratio. This attribute can determine whether buffer pool assignment is efficient. If the pool hit ratio is low, increasing the number of buffer pool pages might improve performance.

Pool Index from Estore Number of buffer pool index pages copied from extended storage. Required index pages are copied from extended storage to the buffer pool. The copy process might incur the cost of connecting to the shared memory segment, but it saves the cost of a disk read.

Pool Index L Reads The number of logical read requests for index pages that have gone through the buffer pool. This count includes accesses to the following index pages:

- Pages that are already in the buffer pool when the database manager needs to process the page.
- Pages that are read into the buffer pool before the database manager can process the page.

By using the Buffer Pool Index Physical Reads attribute, you can calculate the index page hit ratio for the buffer pool as follows:

```
1 - (buffer pool index physical reads / buffer pool index logical reads)
```

If the hit ratio is low, increasing the number of buffer pool pages might improve performance.

Pool Index P Reads The number of physical read requests to get index pages into the buffer pool. see the Pool Index Logical Reads attribute for information about how to use this element.

Pool Index to Estore Number of buffer pool index pages copied to extended storage. Pages are copied from the buffer pool to extended storage when they are selected as victim pages. As a result of the copying process, there is sufficient space for new pages in the buffer pool.

Pool Index Writes The number of times a buffer pool index page was physically written to disk. If a buffer pool index page is written to disk for a high percentage of Buffer Pool Index Physical Reads, performance might improve by increasing the number of buffer pool pages available for the database. If all applications are updating the database, increasing the size of the buffer pool might have minimal impact on performance; most pages contain updated data that must be written to disk.

Pool Read Time The total amount of elapsed time spent processing read requests that caused data or index pages to be physically read from buffer pool to disk. Use this attribute with the Buffer Pool Data Physical Reads and Buffer Pool Index Physical Reads attributes to calculate the average page-read time. This average is important because it might indicate the presence of an I/O wait, which in turn might indicate that you must move data to a different device.

Pool Sync Data Reads The number of physical data page reads that were performed by database manager agents. This value is derived by subtracting the value of the Pool Async Data Reads attribute from the Pool Data Physical Reads attribute. By comparing the ratio of asynchronous to synchronous reads, you can gain insight into how well the prefetchers are working.

Pool Sync Data Writes The total number of physical write requests that were performed synchronously (that is, physical data page writes that were performed by database manager agents). This value is derived by subtracting the value of the Pool Async Data Writes attribute from the value of the Pool Data Writes attribute. By comparing the ratio of asynchronous to synchronous writes, you can gain insight into how well the buffer pool page cleaners are performing.

Pool Sync Index Reads The number of index pages read synchronously (that is, physical index page reads that were performed by database manager agents) into the buffer pool. This value is derived by subtracting the value of the Pool Async Index Reads attribute from Pool Index Physical Reads attribute. By comparing the ratio of asynchronous to synchronous reads, you can gain insight into how well the prefetchers are working.

Pool Sync Index Writes The number of physical index write requests that were performed synchronously (that is, physical index page writes that were performed by database manager agents). This value is derived by subtracting the value of the Pool Async Index Writes attribute from the value of the Pool Index Writes attribute. By comparing the ratio of asynchronous to synchronous writes, you can gain insight into how well the buffer pool page cleaners are performing.

Pool Sync Read The total number of synchronous reads. This value is derived by adding the values of the Pool Sync Data Reads and Pool Sync Index Reads attributes.

Pool Sync Read Time The elapsed time used to perform all synchronous reads. This value is derived by subtracting the value of the Pool Async Read Time attribute from the value of the Pool Read Time attribute. Use this attribute to understand the I/O work being performed.

Pool Sync Write The total number of synchronous index writes. The value is derived by adding the values of the Pool Sync Data Writes attribute and Pool Sync Index Writes attribute.

Pool Sync Write Time The total elapsed time used to perform all synchronous writes. This value is derived by subtracting the value of the Pool Async Write Time attribute from the value of the Pool Write Time attribute.

Pool Total Reads The total number of read requests that required I/O to get data pages and index pages into the buffer pool. This attribute is the total of the Pool Data Physical Reads and Pool Index Physical Reads attributes. Values that are greater than or equal to 2147483647 are indicated in the portal with the `Value Exceeds Maximum` text, and values that are smaller than -2147483648 are indicated with the `Value Exceeds Minimum` text. The following values are valid:

External value	Internal value
Value Exceeds Maximum	2147483647
Value Exceeds Minimum	-2147483648

Pool Total Reads (K) The total number of read requests in thousands (K) that required I/O operations to get data pages and index pages into the buffer pool. The value format is an integer. This attribute is the total of the Pool Data Physical Reads and Pool Index Physical Reads attributes. Values that are greater than or equal to 2147483647 are indicated in the portal with the `Value Exceeds Maximum` text, and values that are smaller than -2147483648 are indicated with the `Value Exceeds Minimum` text. The following values are valid:

External value	Internal value
Value Exceeds Maximum	2147483647
Value Exceeds Minimum	-2147483648

Pool Total Writes The total number of write requests. This attribute is the total of the Pool Data Writes and Pool Index Writes attributes. Values that are greater than or equal to 2147483647 are indicated in the portal with the `Value Exceeds Maximum` text, and values that are smaller than -2147483648 are indicated with the `Value Exceeds Minimum` text. The following values are valid:

External value	Internal value
Value Exceeds Maximum	2147483647
Value Exceeds Minimum	-2147483648

Pool Total Writes (K) The total number of write requests in thousands (K). The value format is an integer. This attribute is the total of the Pool Data Writes and Pool Index Writes attributes. Values that are greater than or equal to 2147483647 are indicated in the portal with the `Value Exceeds Maximum` text, and values that are smaller than -2147483648 are indicated with the `Value Exceeds Minimum` text. The following values are valid:

External value	Internal value
Value Exceeds Maximum	2147483647
Value Exceeds Minimum	-2147483648

Pool Write Time The total amount of time spent physically writing data or index pages from the buffer pool to disk. Use this attribute with the Buffer Pool Data Writes and Buffer Pool Index Writes attributes to calculate the average page-write time. This average is important because it might indicate the presence of an I/O wait, which in turn might indicate that you must move data to a different device.

Customized SQL Definition (KUD_Customized_SQL_Definition) attributes

Use the Customized SQL Definition attribute group to obtain definition information for customized SQL statements, such as the name of the definition file, the last modified time of the definition file, SQL ID and SQL content.

Node Name The managed system name of the agent. For new installations of version 7.1, the format is instanceid:hostname:UD for all operating systems.

Timestamp The local time at the agent when the data was collected.

SQL ID The SQL ID that is defined in the definition file.

SQL Content The SQL content that is defined in the definition file. The carriage return is replaced by a blank. The shown text is limited to 512 bytes.

Customized Definition File The location of the definition file for customized SQL, which is a key attribute.

Last Modified Time The last time that the definition file was modified.

Customized SQL Status (KUD_Customized_SQL_Status) attributes

Use the Customized SQL status attribute group to obtain information about customized SQL statements, such as the SQL ID, the DB Alias, error code, SQL state, error message and last execution time.

Node Name The managed system name of the agent. For new installations of version 7.1, the format is instanceid:hostname:UD for all operating systems.

Timestamp The local time at the agent when the data was collected.

SQL ID The SQL ID that is defined in the definition file.

DB Alias The alias name of the database on which the SQL Statement associated with the SQL ID is executed, which is a key attribute.

SQL State The SQL STATE returned by DB2 for the last SQL execution, which has a length of 10 characters.

Last Execution Error Code The native error code returned by DB2 for the last SQL execution.

Last Execution Error Message The error message returned by DB2 for the last SQL execution, which has a maximum length of 256 characters.

Last Execution Time The last date and time when the SQL is executed.

Customized SQL Detail (KUD_Customized_SQL_Detail) attributes

Use the Customized SQL Detail attribute group to obtain information about the results of customized SQL statement executions, including five string columns, five number columns, and two date and time columns. This attribute group is eligible for use with Tivoli Data Warehouse.

Note: A database alias as the DB Alias Filter Name and an SQL ID are mandatory when defining a situation or when configuring historical data collection for this attribute group.

Node Name The managed system name of the agent.

Timestamp The local time at the agent when the data was collected.

SQL ID The SQL ID that is defined in the definition file.

DB Alias The alias name of the database on which the SQL Statement associated with the SQL ID is executed, which is a key attribute.

DB Alias Filter Name The Database alias filter name that can be defined as:

- The character (*), is required if you want to execute the SQL statement associated with the SQL ID on all the databases of the DB2 server excluding all HADR standby databases.
- A database alias, this is required if you want to execute the SQL statement associated with the SQL ID on a specific database.

Note: If the database filter alias name contains blank spaces at the beginning and at the end, the blank spaces at the end are trimmed.

First String Column Name The name of the first string type column in the result of the customized SQL statement execution.

First String Value The first string value in the result of the customized SQL statement execution.

Second String Column Name The name of the second string type column in the result of the customized SQL statement execution.

Second String Value The second string value in the result of the customized SQL statement execution.

Third String Column Name The name of the third string type column in the result of the customized SQL statement execution.

Third String Value The third string value in the result of the customized SQL statement execution.

Fourth String Column Name The name of the fourth string type column in the result of the customized SQL statement execution.

Fourth String Value The fourth string value in the result of the customized SQL statement execution.

Fifth String Column Name The name of the fifth string type column in the result of the customized SQL statement execution.

Fifth String Value The fifth string value in the result of the customized SQL statement execution.

First Number Column Name The name of the first number type column in the result of the customized SQL statement execution.

First Number Value The first number value in the result of the customized SQL statement execution.

DEFAULT with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values shown in parentheses. The following values are defined:
- Not Available (-999999999)
- Any other values will display the actual value returned by the agent in the Tivoli Enterprise Portal.

Second Number Column Name The name of the second number type column in the result of the customized SQL statement execution.

Second Number Value The second number value in the result of the customized SQL statement execution.

DEFAULT with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values shown in parentheses. The following values are defined:
- Not Available (-999999999)
- Any other values will display the actual value returned by the agent in the Tivoli Enterprise Portal.

Third Number Column Name The name of the third number type column in the result of the customized SQL statement execution.

Third Number Value The third number value in the result of the customized SQL statement execution.

DEFAULT with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values shown in parentheses. The following values are defined:
- Not Available (-999999999)
- Any other values will display the actual value returned by the agent in the Tivoli Enterprise Portal.

Fourth Number Column Name The name of the fourth number type column in the result of the customized SQL statement execution.

Fourth Number Value The fourth number value in the result of the customized SQL statement execution.

DEFAULT with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values shown in parentheses. The following values are defined:

- Not Available (-999999999)
- Any other values will display the actual value returned by the agent in the Tivoli Enterprise Portal.

Fifth Number Column Name The name of the fifth number type column in the result of the customized SQL statement execution.

Fifth Number Value The fifth number value in the result of the customized SQL statement execution.

DEFAULT with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values shown in parentheses. The following values are defined:
- Not Available (-999999999)
- Any other values will display the actual value returned by the agent in the Tivoli Enterprise Portal.

First Date Column Name The name of the first date time type column in the result of the customized SQL statement execution.

First Date Value The first date time value in the result of the customized SQL statement execution.

Second Date Column Name The name of the second date time type column in the result of the customized SQL statement execution.

Second Date Value The second date time value in the result of the customized SQL statement execution.

Database00 (KUD_DB2_Database00) attributes

Use these attributes to obtain information about database activities. By using this information, you can determine the efficiency of the databases and identify any problem areas for corrective action. All values are integers that are calculated from the first database connection, unless otherwise noted.

Active Sorts The number of sorts in the database that currently have a sort heap allocated. Use this value with the Sort Heap Allocated attribute to determine the average sort heap space used by each sort. If the SORTHEAP configuration parameter is substantially larger than the average sort heap used, you might be able to lower the value of this parameter. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Agents Top The maximum number of agents (at one time) associated with applications that are connected to the monitored database. Use this attribute to indicate how well the intra-query parallelism was realized. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Appls Cur Cons The number of applications currently connected to the monitored database. Use this attribute to help you understand the level of activity within a database and the amount of system resource being used. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Lock Wait Time The average elapsed time (in milliseconds) that was spent waiting for a lock. If the average lock wait time is high, you must look for applications that hold many locks, or have lock escalations, with a focus on tuning your applications to improve concurrency, if appropriate. If escalations are the reason for a high average lock wait time, the values of one or both of the LOCKLIST and MAXLOCKS configuration parameters might be too low. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Pool Read Time The average elapsed time for a read request. This value is derived by dividing the value of the Pool Read Time attribute by the value of the Pool Total Reads attribute. This average is important because it might indicate the presence of an I/O wait, which in turn might indicate that you must move data to a different device. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Pool Write Time The average elapsed time for a write request. This value is derived by dividing the value of the Pool Write Time attribute by the value of the Pool Total Writes attribute. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Sort Time The average derived by dividing value of the Total Sort Time attribute by the value of the Total Sorts attribute. The average is expressed as elapsed time. at the database or application level, this attribute can indicate whether sorting is a performance issue. Elapsed times are affected by system load. The more processes you have running, the higher this elapsed time value is. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Cat Cache Heap Full The number of times that an insert into the catalog cache failed because of a heap full condition in the database heap. The catalog cache draws its storage dynamically from the database heap. Even if the cache storage has not reached its limit, inserts into the catalog cache might fail due to a lack of space in the database heap. If the catalog cache heap full count is not zero, you can correct the insert failure condition by increasing the database heap size or by reducing the catalog cache size. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Cat Cache Hit Ratio The percentage of catalog sections found in the cache. This ratio indicates how well the catalog cache is avoiding catalog accesses. If the ratio is high (more than 0.8), the cache is performing well. A smaller ratio might indicate that you must increase the size of the catalog cache. You must expect a large ratio immediately following the first connection to the database.

Cat Cache Inserts The number of times that the system tried to insert table descriptor information into the catalog cache. Table descriptor information is inserted into the cache following a failed lookup to the catalog cache while processing a table, view, or alias reference in an SQL statement. The catalog cache inserts value includes attempts to insert table descriptor information that fail due to catalog cache overflow and heap full conditions. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Cat Cache Lookups The number of times that the catalog cache was referenced to obtain table descriptor information. This attribute includes both successful and unsuccessful accesses to the catalog cache. This attribute is used in calculating the catalog cache hit ratio. This ratio indicates how well the catalog cache is avoiding catalog accesses. If the ratio is high (more than 0.8), the cache is performing well. A smaller ratio might indicate that you must increase the size of the catalog cache. You must expect a large ratio immediately following the first connection to the database. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Cat Cache Overflows The number of times that an insert into the catalog cache failed because the catalog cache was full. If the catalog cache overflows value is large, the catalog cache might be too small for the workload. Increasing the size of the catalog cache might improve its performance. If the workload includes transactions that compile a large number of SQL statements referencing many tables, views, and aliases in a single unit of work, compiling fewer SQL statements in a single transaction might improve the performance of the catalog cache. Or if the workload includes the binding of packages containing many SQL statements referencing many tables, views or aliases, you might want to split the packages so that they include fewer SQL statements to improve performance. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Catalog Node Name The network name of the catalog node.

Commit SQL Stmts The total number of SQL COMMIT statements that have been attempted. A small rate of change in this counter during the monitor period might indicate that applications are not doing frequent commits. The lack of frequent commits can lead to problems with logging and data concurrency. You can also use this attribute to calculate the total number of units of work by calculating the sum of the following values:

- Commit statements attempted
- Internal commits
- Rollback statements attempted
- Internal rollbacks

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Connections Top The highest number of simultaneous connections to the database since the database was activated. You can calculate the current number of connections at the time the snapshot was taken by adding the Remote Connections to Database Manager and Local Connections attributes. Use this attribute to evaluate the setting of the MAXAPPLS configuration parameter. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Coord Agents Top The maximum number of coordinating agents working at one time. The MAXCAGENTS configuration parameter determines the number of coordinating agents that can be executing concurrently. If the peak number of coordinating agents results in a workload that is too high for this node, you can reduce the MAXCAGENTS configuration parameter. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Database Status The status of the database. The following values are valid:

External value	Internal value	Description
Active	Active	The database is active.
InActive	InActive	The database is not active.
Quiesce Pending	Quiesce_Pending	The database is in quiesce-pending state. New connections to the database are not permitted and new units of work cannot be started. Depending on the quiesce request, active units of work are allowed to complete or are rolled back immediately.
Quiesced	Quiesced	The database has been quiesced. New connections to the database are not permitted and new units of work cannot be started.
Roll Forward	Roll_Forward	A rollforward is in progress on the database.

External value	Internal value	Description
Unknown	Unknown	The status is unknown.

DB Conn Timestamp The date and time when the first database connection was made.

DB Location The location of the database in relation to the application. The following values are valid:

External value	Internal value
LOCAL	LOCAL
REMOTE	REMOTE
UNKNOWN	UNKNOWN

Determine the relative location of the database server with respect to the application taking the snapshot.

DB Name The real name of the database for which information is collected or to which the application is connected. This name was given to the database when it was created. The value format is a simple text string with a maximum of 60 characters. Use this attribute to identify the specific database to which the data applies.

DB Partition The DB2 database partition node number, which can range from 0 to 999. The Aggregated and Current Partition values can be used within a query or situation filter. If a db partition filter is not specified, data is returned for the current database partition. If a db partition filter is set to Aggregated, only aggregated partition data is returned. Historical data collection includes both aggregated and individual partition attribute data. In addition to numeric partition numbers in the 0 to 999 range, the following values are also valid:

External value	Internal value
Aggregated	-1
Current Partition	-2
All	-3

DB Path The full path of the location where the database is stored on the monitored system. The value format is a simple text string with a maximum of 768 characters. Use this attribute with the Database Name attribute to identify the specific database to which the data applies.

DDL SQL Stmts The number of SQL Data Definition Language (DDL) statements that ran. Use this attribute to determine the level of database activity at the application or database level. DDL statements are expensive to run because of their impact on the system catalog tables. As a result, if the value of this attribute is high, you must determine the cause and possibly restrict the identified activity from being performed. You can also use this attribute to determine the percentage of DDL activity using the following formula: divide the number of DDL SQL statements by the total number of statements. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Deadlocks The total number of deadlocks that have occurred. This attribute can indicate that applications are experiencing contention problems. To resolve the problem, determine in which applications (or application processes) the deadlock are occurring. You can then modify the application to enable it to run concurrently. Some applications, however, might not be capable of running concurrently. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Direct Read Reqs The number of requests to perform a direct read of one or more sectors of data. Use the following formula to calculate the average number of sectors that are read by a direct read:

```
direct reads from database / direct read requests
```

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Direct Read Time The elapsed time (in milliseconds) required to perform the direct reads. Use the following formula to calculate the average direct read time per sector:

```
direct read time / direct reads from database
```

A high average time might indicate an I/O conflict. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Direct Reads The number of read operations that do not use the buffer pool. Use the following formula to calculate the average number of sectors that are read by a direct read:

```
direct reads from database / direct read requests
```

When using system monitors to track I/O, this data attribute helps to distinguish database I/O from non-database I/O on the device. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Direct Write Reqs The number of requests to perform a direct write of one or more sectors of data. Use the following formula to calculate the average number of sectors that are written by a direct write:

```
direct writes to database / direct write requests
```

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Direct Write Time The elapsed time (in milliseconds) required to perform the direct writes. Use the following formula to calculate the average direct write time per sector:

```
direct write time / direct writes to database
```

A high average time might indicate an I/O conflict. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Direct Writes The number of write operations that do not use the buffer pool. Use the following formula to calculate the average number of sectors that are written by a direct write:

```
direct writes to database / direct write requests
```

When using system monitors to track I/O, this data attribute helps to distinguish database I/O from non-database I/O on the device. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Dynamic SQL Stmts The number of dynamic SQL statements that were attempted. Use this attribute to calculate the total number of successful SQL statements at the database or application level by performing the following operations:

1. Add the number of Dynamic SQL Statements Attempted and the Static SQL Statements Attempted.
2. Subtract the number of Failed Statement Operations.

The remainder equals the throughput (the number of successful SQL statements) during the current monitoring period. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Failed SQL Stmts The number of SQL statements that were attempted, but failed. This count includes all SQL statements that received a negative SQLCODE. Use this attribute to calculate the total number of successful SQL statements at the database or application level by performing the following operations:

1. Add the number of Dynamic SQL Statements Attempted and the Static SQL Statements Attempted.
2. Subtract the number of Failed Statement Operations.

The remainder equals the throughput (the number of successful SQL statements) during the current monitoring period. This attribute can also help you to determine the reasons for poor performance; failed statements indicate time wasted by the database manager, which results in lower throughput for the database.

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Files Closed The total number of database files closed. The database manager opens files for reading and writing into and out of the buffer pool. The maximum number of database files open by an application at any time is controlled by the MAXFILOP configuration parameter. If the maximum is reached, one file is closed before the new file is opened. Note that the actual number of files opened might not equal the number of files closed. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Hash Join Overflows The number of times that hash join data exceeded the available sort heap space. At the database level, if the percentage of Hash Join Small Overflows is greater than 10% of this value, you must consider increasing the sort heap size. You can use values at the application level to evaluate hash join performance for individual applications. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Hash Join Small Overflows The number of times that hash join data exceeded the available sort heap space by less than 10%. If this value and the value of the Hash Join Overflows attribute are high, consider increasing the sort heap threshold. If this value is greater than 10% of Hash Join Overflows, consider increasing the sort heap size. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Input DB Alias The alias of the database provided when calling the snapshot function. The value format is a simple text string with a maximum of 60 characters. Use this attribute to help you identify the specific database to which the monitor data applies. It contains blanks unless you requested monitor information related to a specific database.

Int Deadlock Rollbacks The total number of forced rollbacks initiated by the database manager due to a deadlock. The database manager initiates a rollback for the current unit of work in an application that is experiencing a deadlock. This attribute shows the number of deadlocks that have been broken. It can indicate the possibility of concurrency problems. It is also important because internal rollbacks due to deadlocks can cause performance degradation. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Int Rollbacks The total number of rollbacks initiated internally by the database manager. An internal rollback occurs when any of the following operations cannot complete successfully:

- A reorganization
- An import
- A bind or pre-compile
- An application that ends as a result of a deadlock situation or lock timeout situation
- An application that ends without executing an explicit COMMIT or ROLLBACK statement (on Windows systems).

Use this attribute to calculate the total number of units of work by calculating the sum of the following values: commit statements attempted, internal commits, rollback statements attempted, and internal rollbacks.

Instance Name The name of the monitored DB2 instance. The valid format is a text string with a maximum of 60 bytes.

Last Backup The date and time that the latest database backup was completed. Use this attribute to help you identify a database that has not been backed up recently, or to identify which database backup file is the most recent. If the database has never been backed up, this timestamp is initialized to zero.

Lock Escals The number of times that locks have been escalated from several row locks to a table lock. A lock is escalated when the total number of locks held by an application reaches the maximum amount of lock list space available to the application, or the lock list space consumed by all applications is approaching the total lock list space. This data item includes a count of all lock escalations, including exclusive lock escalations. When an application reaches the maximum number of locks allowed and there are no more locks to escalate, the application uses space in the lock list that is allocated for other applications. When the entire lock list is full, an error occurs. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Lock List in Use The total amount of lock list memory (in bytes) that is currently in use. This attribute can be used with the locklist configuration parameter to calculate the lock list utilization. If the lock list utilization is high, you might want to consider increasing the size of that parameter. Values that are greater than or equal to 2147483647 are indicated in the portal with the `Value Exceeds Maximum` text, and values that are smaller than -2147483648 are indicated with the `Value Exceeds Minimum` text. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Lock Timeouts The number of times that a request to lock an object time out instead of being granted. This attribute can help you adjust the setting for the LOCKTIMEOUT database configuration parameter. If the number of lock timeouts becomes excessive when compared to normal operating levels, an application might be holding locks for long durations. In this case, this attribute might indicate that you must analyze some of the other attributes related to locks and deadlocks to determine if an application problem exists. It is also possible to have too few lock timeouts if the LOCKTIMEOUT database configuration parameter is set too high. In this case, applications might wait excessively to obtain a lock. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Lock Wait Time The total elapsed time (in milliseconds) that was spent waiting for a lock. At the database level, this is the total amount of elapsed time that all applications were waiting for a lock within this database. At the application-connection and transaction levels, this is the total amount of elapsed time that this connection or transaction has waited for a lock to be granted. Use this attribute with the Lock Waits attribute to calculate the average wait time for a lock. This calculation can be performed at either the database or the application-connection level. If the average lock wait time is high, you must look for applications that hold many locks, or have lock escalations, with a focus on tuning your applications to improve concurrency, if appropriate. If escalations are the reason for a high average lock wait time, the values of one or both of the LOCKLIST and MAXLOCKS configuration parameters might be too low. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Lock Waits The total number of times that applications or connections waited for locks. At the database level, the lock waits value is the total number of times that applications waited for locks within this database. At the application-connection level, the lock waits value is the total number of times that this connection requested a lock but waited because another connection was already holding a lock on the data. Use this attribute with the Lock Wait Time attribute to calculate, at the database level, the average wait time for a lock. This calculation can be performed at either the database or the application-connection level. If the average lock wait time is high, you must look for applications that hold many locks, or have lock escalations, with a focus on tuning your applications to improve concurrency, if appropriate. If escalations are the reason for a high average lock wait time, the values of one or both of the LOCKLIST and MAXLOCKS configuration parameters might be too low. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Locks Held The number of locks currently held. If the monitor information is at the database level, this number represents the total number of locks currently held by all applications in the database. If the information is at the application level, this number represents the total number of locks currently held by all agents for the application. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Locks Waiting The number of agents that are currently waiting on a lock. When used with Applications Currently Connected attribute, this attribute indicates the percentage of applications waiting on locks. If this number is high, the applications might have concurrency problems. In this case, you must identify applications that are holding locks or exclusive locks for long periods of time. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Log Reads The number of log pages read from disk by the logger. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Log Writes The number of log pages written to disk by the logger. Use this attribute with an operating system monitor to quantify the amount of I/O on a device that is attributable to database activity. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Node Name For new installations of version 6, release 2, the format is instanceid:hostname:UD for all operating systems. The format for version 6, release 1 of the DB2 agent on Windows systems is instanceid:hostname:UD; on UNIX and Linux systems, the format is instanceid:hostname.

Num Assoc Agents The current number of subagents associated with all applications that are connected to the monitored database. Use this attribute to evaluate the settings for the agent configuration parameters. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pkg Cache Hit Ratio The percentage of package sections that were found in the cache. This ratio tells you whether the package cache is being used effectively. If the hit ratio is high (more than 0.8), the cache is performing well. A smaller ratio might indicate that the package cache must be increased. The following values are also valid:

External value	Internal value
Value Exceeds Maximum	2147483647
Not Available	-1
Not Collected	-2

Pkg Cache Inserts The total number of times that a requested section was not available for use and had to be loaded into the package cache. This count includes any implicit prepares performed by the system. By using the Package Cache Lookups attribute, you can calculate the package cache hit ratio using the following formula:

```
1 - (Package Cache Inserts / Package Cache Lookups)
```

See the Package Cache Lookups attribute for information about using this attribute.

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pkg Cache Lookups The number of times that an application looked for a section or package in the package cache. At a database level, it indicates the overall number of references since the database was started, or monitor data was reset. Note that this counter includes the cases where the section is already loaded in the cache and when the section has to be loaded into the cache. To calculate the package cache hit ratio use the following formula:

```
1 - (Package Cache Inserts / Package Cache Lookups)
```

The package cache hit ratio tells you whether the package cache is being used effectively. If the hit ratio is high (more than 0.8), the cache is performing well. A smaller ratio might indicate that the package cache must be increased.

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Async Data Read Reqs The number of asynchronous read requests. To calculate the average number of data pages read per asynchronous request, use the following formula:

```
buffer pool asynchronous data reads / buffer pool asynchronous read requests
```

This average can help to determine the amount of asynchronous I/O in each interaction with the prefetcher.

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Async Data Reads The number of pages read asynchronously into the buffer pool. Use this attribute with the Buffer Pool Data Physical Reads attribute to calculate the number of physical reads that were performed synchronously (that is, physical data page reads that were performed by database manager agents). Use the following formula:

```
buffer pool data physical reads - buffer pool synchronous data reads
```

By comparing the ratio of asynchronous to synchronous reads, you can gain insight into how well the prefetchers are working.

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Async Data Writes The number of times a buffer pool data page was physically written to disk by an asynchronous page cleaner or by a prefetcher. A prefetcher might have written dirty pages to disk to make space for the pages being prefetched. Use this attribute with the Buffer Pool Data Writes attribute to calculate the number of physical write requests that were performed synchronously (that is, physical data page writes that were performed by database manager agents). Use the following formula:

```
buffer pool data writes - buffer pool asynchronous data writes
```

By comparing the ratio of asynchronous to synchronous writes, you can gain insight into how well the buffer pool page cleaners are performing.

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Async Index Reads The number of index pages read asynchronously into the buffer pool by a prefetcher. Asynchronous reads are performed by database manager prefetchers. Use this attribute with the Buffer Pool Index Physical Reads attribute to calculate the number of physical reads that were performed synchronously (that is, physical index page reads that were performed by database manager agents). Use the following formula:

```
buffer pool index physical reads - buffer pool asynchronous index reads
```

By comparing the ratio of asynchronous to synchronous reads, you can gain insight into how well the prefetchers are working.

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Async Index Writes The number of times a buffer pool index page was physically written to disk by an asynchronous page cleaner or a prefetcher. A prefetcher might have written dirty pages to disk to make space for the pages being prefetched. Use this attribute with the Buffer Pool Index Writes attribute to calculate the number of physical index write requests that were performed synchronously. That is, physical index page writes that were performed by database manager agents. Use the following formula:

```
buffer pool index writes - buffer pool asynchronous index writes
```

By comparing the ratio of asynchronous to synchronous writes, you can gain insight into how well the buffer pool page cleaners are performing.

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Async Read Time The total elapsed time spent reading by database manager prefetchers. Use this attribute to calculate the elapsed time for synchronous reading, using the following formula:

```
total buffer pool physical read time - buffer pool synchronous read time
```

You can also use this attribute to calculate the average asynchronous read time using the following formula:

```
buffer pool asynchronous read time / buffer pool asynchronous data reads
```

These calculations can be used to understand the I/O work being performed.

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Async Write Time The total elapsed time spent writing data or index pages from the buffer pool to disk by database manager page cleaners. Calculate the elapsed time spent writing pages synchronously by subtracting the value of the Pool Async Write Time attribute from the value of the Pool Physical Write Time attribute. You can also use this attribute to calculate the average asynchronous read time by performing the following operations:

1. Add the Pool Async Data Writes and the Pool Async Index Writes.
2. Divide the Pool Async Write Time by the sum from step 1.

These calculations can be used to understand the I/O work being performed.

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Data from Estore Number of buffer pool data pages copied from extended storage. Required pages are copied from extended storage to the buffer pool. The copy process might incur the cost of connecting to the shared memory segment, but it saves the cost of a disk read.

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Data L Reads The number of logical read requests for data pages that have gone through the buffer pool. This count includes accesses to the following data:

- Data that is already in the buffer pool when the database manager needs to process the page
- Data that is read into the buffer pool before the database manager can process the page.

By using the Pool Data Physical Reads attribute, you can calculate the data page hit ratio for the buffer pool as follows:

```
1 - (buffer pool data physical reads / buffer pool data logical reads)
```

By using the Pool Data Physical Reads, Pool Index Physical Reads, and Pool Index Logical Reads attributes, you can calculate the overall buffer pool hit ratio as follows:

```
1 - ((buffer pool data physical reads + buffer pool index physical reads)
 / (buffer pool data logical reads + buffer pool index logical reads))
```

Increasing buffer pool size generally improves the hit ratio until you reach a point of diminishing returns.

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Data P Reads The number of read requests that required I/O to get data pages into the buffer pool. See Pool Data Logical Reads and Pool Async Data Reads attributes for information about how to use this attribute. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Data to Estore Number of buffer pool data pages copied to extended storage. Pages are copied from the buffer pool to extended storage when they are selected as victim pages. As a result of the copying process, there is sufficient space for new pages in the buffer pool. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Data Writes The number of times a buffer pool data page was physically written to disk. A buffer pool data page is written to disk for the following reasons:

- To free a page in the buffer pool so another page can be read
- To flush the buffer pool.

If a buffer pool data page is written to disk for a high percentage of Buffer Pool Data Physical Reads, performance might improve by increasing the number of buffer pool pages available for the database.

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Drty Pg Steal Clns Buffer Pool Victim Page Cleaners Triggered is the number of times a page cleaner was invoked because a synchronous write was needed during the victim buffer replacement for the database. Use this attribute, in combination with others, to evaluate the number of page cleaners that are defined. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Drty Pg Thrsh Clns The number of times a page cleaner was invoked because a buffer pool had reached the dirty page threshold criterion for the database. When the number of dirty pages in the pool exceeds this value, the cleaners are triggered. If this value is set too low, pages might be written out too early, requiring them to be read back in. If set too high, too many pages might accumulate, requiring users to write out pages synchronously. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Hit Ratio The buffer pool hit ratio (as a percentage). The sum of the Pool Data Logical Reads and Pool Index Logical Reads attributes is divided by the value of the Pool Total Reads attribute to derive the pool hit ratio. Use this attribute to determine whether buffer pool assignment is efficient. If the pool hit ratio is low, increasing the number of buffer pool pages might improve performance.

Pool Index from Estore Number of buffer pool index pages copied from extended storage. Required index pages are copied from extended storage to the buffer pool. The copy process might incur the cost of connecting to the shared memory segment, but it saves the cost of a disk read. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Index L Reads The number of logical read requests for index pages that have gone through the buffer pool. This count includes accesses to the following index pages:

- Pages that are already in the buffer pool when the database manager needs to process the page.
- Pages that are read into the buffer pool before the database manager can process the page.

By using the Buffer Pool Index Physical Reads attribute, you can calculate the index page hit ratio for the buffer pool as follows:

```
1 - (buffer pool index physical reads / buffer pool index logical reads)
```

To calculate the overall buffer pool hit ratio, see the Buffer Pool Data Logical Reads attribute. If the hit ratio is low, increasing the number of buffer pool pages might improve performance.

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Index P Reads The number of physical read requests to get index pages into the buffer pool. See the Pool Index Logical Reads attribute for information about how to use this element.

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Index to Estore Number of buffer pool index pages copied to extended storage. Pages are copied from the buffer pool to extended storage when they are selected as victim pages. As a result of the copying process, there is sufficient space for new pages in the buffer pool. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Index Writes The number of times a buffer pool index page was physically written to disk. If a buffer pool index page is written to disk for a high percentage of Buffer Pool Index Physical Reads, performance might improve by increasing the number of buffer pool pages available for the database. If all applications are updating the database, increasing the size of the buffer pool might have minimal impact on performance; most pages contain updated data that must be written to disk. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool LSN Gap Clns The number of times a page cleaner was invoked because the logging space used had reached a predefined criterion for the database. Use this attribute to evaluate whether you have enough space for logging, and whether you need more log files or larger log files. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Read Time The total amount of elapsed time spent processing read requests that caused data or index pages to be physically read from buffer pool to disk. Use this attribute with the Buffer Pool Data Physical Reads and Buffer Pool Index Physical Reads attributes to calculate the average page-read time. This average is important because it might indicate the presence of an I/O wait, which in turn might indicate that you must move data to a different device. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Sync Data Reads The number of physical data page reads that were performed by database manager agents. This value is derived by subtracting the value of the Pool Async Data Reads attribute from the Pool Data Physical Reads attribute. By comparing the ratio of asynchronous to synchronous reads, you can gain insight into how well the prefetchers are working. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Sync Index Reads The number of index pages read synchronously (that is, physical index page reads that were performed by database manager agents) into the buffer pool. This value is derived by subtracting the value of the Pool Async Index Reads attribute from Pool Index Physical Reads attribute. By comparing the ratio of asynchronous to synchronous reads, you can gain insight into how well the prefetchers are working. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Total Reads The total number of read requests that required I/O to get data pages and index pages into the buffer pool. This attribute is the total of the Pool Data Physical Reads and Pool Index Physical Reads attributes. Values that are greater than or equal to 9223372036854775807 are indicated with the `Value Exceeds Maximum` text in the portal. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Total Writes The total number of write requests. This attribute is the total of the Pool Data Writes and Pool Index Writes attributes. Values that are greater than or equal to 9223372036854775807 are indicated with the `Value Exceeds Maximum` text in the portal. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Write Time The total amount of time spent physically writing data or index pages from the buffer pool to disk. Use this attribute with the Buffer Pool Data Writes and Buffer Pool Index Writes attributes to calculate the average page-write time. This average is important because it might indicate the presence of an I/O wait, which in turn might indicate that you must move data to a different device. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Rollback SQL Stmts The total number of SQL ROLLBACK statements that have been attempted. A rollback can result from an application request, a deadlock, or an error situation. This attribute counts only the number of rollback statements issued from applications. At the application level, this attribute can help you determine the level of database activity for the application and the amount of conflict with other applications. At the database level, it can help you determine the amount of activity in the database and the amount of conflict between applications on the database. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Rows Deleted The number of row deletions attempted. Use this attribute to gain insight into the current level of activity within the database manager. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Rows Inserted The number of row insertions attempted. Use this attribute to gain insight into the current level of activity within the database manager. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Rows Selected The number of rows that have been selected and returned to the application. Use this attribute to gain insight into the current level of activity within the database manager. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Rows Updated The number of row updates attempted. Use this attribute to gain insight into the current level of activity within the database manager. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Sec Logs Allocated The total number of secondary log files that are currently being used for the database. Use this attribute with the Secondary Log Used Top and Total Log Used Top attributes to show the current dependency on secondary logs. If this value is consistently high, you might need larger log files, more primary log files, or more frequent COMMIT statements within your application. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Sec Log Used Top The maximum amount of secondary log space (in bytes) that has been used. Use this attribute with the Secondary Logs Allocated and Total Log Used Top attributes to show the current dependency on secondary logs. If this value is high, you might need larger log files, more primary log files, or more frequent COMMIT statements within your application. Values that are greater than or equal to 9223372036854775807 are indicated with the `Value Exceeds Maximum` text in the portal. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Select SQL Stmts The number of SQL SELECT statements that ran. Use this attribute to determine the level of database activity at the application or database level. You can also use the following formula to determine the ratio of SELECT statements to the total statements by performing the following operations:

- Add the number of static SQL statements attempted and dynamic SQL statements attempted
- Divide the resulting total by the number of select SQL statements that ran

The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Server Platform The operating system upon which the database management system is running. Use this attribute during troubleshooting for remote applications.

Snapshot Timestamp The date and time when the database system monitor information was collected. Use this attribute to help relate data chronologically if you are saving the results in a file or database for ongoing analysis. Use this attribute to help relate data chronologically if you are saving the results in a file or database for ongoing analysis.

Sort Heap Allocated The total number of allocated pages of sort heap space for all sorts at the level chosen (database manager or database) and at the time the snapshot was taken. Memory estimates do not include sort heap space. If excessive sorting occurs, add the extra memory (used for the sort heap) to the base memory requirements for running the database manager. Generally, the larger the sort heap, the more efficient the sort. Appropriate use of indexes can reduce the amount of sorting required. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Sort Overflows The total number of sorts that ran out of sort heap space and might have required disk space for temporary storage. at the database or application level, use this attribute with the Total Sorts attribute to calculate the percentage of sorts that required overflow to disk. If this percentage is high, you might want to adjust the database configuration by increasing the value of the SORTHEAP configuration parameter. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Sort Overflows Percent The percentage of sorts that ran out of sort heap space and might have required disk space for temporary storage. This percentage is calculated by dividing the value of the Sort Overflows attribute by the value of the Total Sorts attribute. at the database or application level, use this attribute to evaluate the percentage of sorts that required overflow to disk. If this percentage is high, you might want to adjust the database configuration by increasing the value of the SORTHEAP configuration parameter.

SQL Stmts Failed Percent The percentage of SQL statements that failed to run successfully. This value is derived by dividing the value of the Failed SQL Statements attribute by the value of the Total SQL Statements attribute. Use this attribute to determine whether an application has some design issues.

SQL Stmts Rollback Percent The percentage of SQL statements that resulted in a rollback. This value is derived by dividing the value of the Rollback SQL Statements attribute by the value of the Total SQL Statements attribute. Use this attribute to determine whether an application has some design issues.

Static SQL Stmts The number of static SQL statements that were attempted. Use this attribute to calculate the total number of successful SQL statements at the database or application level by performing the following operations:

1. Add the number of Dynamic SQL Statements Attempted and the Static SQL Statements Attempted.
2. Subtract the number of Failed Statement Operations.

The remainder equals the throughput (the number of successful SQL statements) during the current monitoring period.

The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Total Cons The number of connections to the database since the first connect, activate, or last reset (coordinator agents). Use this attribute with the Database Activation Timestamp and the Start Database Manager Timestamp attributes to calculate the frequency at which applications have connected to the database. The first connect to a database (such as an initial buffer pool allocation) causes extra overhead. If the frequency of connects is low, it might be beneficial to activate the database explicitly using the ACTIVATE DATABASE command before connecting any other application. As a result, subsequent connects are processed at a higher rate. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Remember: When you reset this attribute, the value of the attribute is set to the number of applications that are currently connected, instead of zero.

Total Hash Joins The total number of hash joins that ran. At the database or application level, use this value with the Hash Join Overflows attribute and the Hash Join Small Overflows attribute to determine if a significant percentage of hash joins would benefit from modest increases in the sort heap size. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Total Hash Loops The total number of times that a single partition of a hash join was larger than the available sort heap space. This attribute might indicate inefficient execution of hash joins (the sort heap size is too small or the sort heap threshold is too small). Use this value with the other hash join variables to tune the sort heap size (SORTHEAP) and sort heap threshold (SHEAPTHRES) configuration parameters. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Total Log Used Top The maximum amount of total log space (in bytes) that has been used. Use this attribute to evaluate the amount of primary log space that is allocated. Comparing the value of this attribute with the amount of primary log space that is allocated can help you to evaluate the configuration parameter settings. Values that are greater than or equal to 9223372036854775807 are indicated with the `Value Exceeds Maximum` text in the portal. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Total Sec Cons The number of connections made by a subagent to the database at the node. Use this attribute with the Connects Since Database Activation, Database Activation Timestamp, and the Start Database Manager Timestamp attributes to calculate the frequency at which applications have connected to the database. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Total Sort Time The total elapsed time (in milliseconds) for all sorts that ran. at the database or application level, use this attribute with the Total Sorts attribute to calculate the average sort time, which can indicate whether sorting is a performance issue. Elapsed times are affected by system load. The more processes you have running, the higher this elapsed time value is. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Total Sorts The total number of sorts that ran. at the database or application level, use this value with the Sort Overflows attribute to calculate the percentage of sorts that need more heap space. You can also use it with the Total Sort Time attribute to calculate the average sort time. If the number of sort overflows is small with respect to the total sorts, increasing the sort heap size might have little impact on performance, unless this buffer size is increased substantially. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Total SQL Stmts The total number of dynamic and static SQL statements. This value is derived by adding the values of the Dynamic SQL Statements and the Static SQL Statements attributes. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

UID SQL Stmts The number of SQL UPDATE, INSERT, and DELETE statements that ran. Use this attribute to determine the level of database activity at the application or database level. You can also use the following formula to determine the ratio of UPDATE, INSERT, and DELETE statements to the total number of statements:

1. Add the number of static SQL statements attempted and the dynamic SQL statements attempted.
2. Divide the number of UPDATE/INSERT/DELETE SQL statements that ran by the sum derived in step 1.

The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

X Lock Escals The number of times that locks have been escalated from several row locks to one exclusive table lock, or the number of times an exclusive lock on a row caused the table lock to become an exclusive lock. A lock is escalated when the total number of locks held by an application reaches the maximum amount of lock list space available to the application. The amount of lock list space available is determined by the LOCKLIST and MAXLOCKS configuration parameters. Other applications cannot access data held by an exclusive lock. Because exclusive locks can affect the concurrency of your data, it is important to track them. When an application reaches the maximum number of locks allowed and there are no more locks to escalate, it uses space in the lock list allocated for other applications. When the entire lock list is full, an error occurs. See the Lock Escals attribute for possible causes and resolutions to excessive exclusive lock escalations. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Database00 (KUDDBASEGROUP00) attributes (Superseded)

Use these attributes to obtain information about database activities. By using this information, you can determine the efficiency of the databases and identify any problem areas for corrective action. All values are integers that are calculated from the first database connection, unless otherwise noted. This attribute group is superseded. There is a new attribute group with the same name that replaces it.

Active Sorts The number of sorts in the database that currently have a sort heap allocated. Use this value with the Sort Heap Allocated attribute to determine the average sort heap space used by each sort. If the SORTHEAP configuration parameter is substantially larger than the average sort heap used, you might be able to lower the value of this parameter.

Agents Top The maximum number of agents (at one time) associated with applications that are connected to the monitored database. Use this attribute to indicate how well the intra-query parallelism was realized.

Appl Section Inserts The number of inserts of SQL sections by an application from its SQL work area. The working copy of any executable section is stored in a unique SQL work area. This value represents the number of times when a copy was not available and therefore was inserted.

Appl Section Lookups The number of lookups of SQL sections by an application from its SQL work area. This counter indicates how many times the SQL work area was accessed by agents for an application. It is a cumulative total of all lookups on all SQL work heaps for agents working on this application.

Appls Cur Cons The number of applications currently connected to the monitored database. Use this attribute to help you understand the level of activity within a database and the amount of system resource being used.

Avg Data Page Read per Async Req The average number of pages read for each asynchronous request. This value is derived by dividing the value of the Pool Async Data Reads attribute by the value of the Pool Async Data Read Reqs attribute. Use this attribute to determine whether good enough data pages were read per asynchronous request.

Avg Lock Wait Time The average elapsed time (in milliseconds) that was spent waiting for a lock. If the average lock wait time is high, you must look for applications that hold many locks, or have lock escalations, with a focus on tuning your applications to improve concurrency, if appropriate. If escalations are the reason for a high average lock wait time, the values of one or both of the LOCKLIST and MAXLOCKS configuration parameters might be too low.

Avg Pool Read Time The average elapsed time for a read request. This value is derived by dividing the value of the Pool Read Time attribute by the value of the Pool Total Reads attribute. This average is important because it might indicate the presence of an I/O wait, which in turn might indicate that you must move data to a different device.

Avg Pool Write Time The average elapsed time for a write request. This value is derived by dividing the value of the Pool Write Time attribute by the value of the Pool Total Writes attribute.

Avg Sort Time The average derived by dividing value of the Total Sort Time attribute by the value of the Total Sorts attribute. The average is expressed as elapsed time. at the database or application level, this attribute can indicate whether sorting is a performance issue. Elapsed times are affected by system load. The more processes you have running, the higher this elapsed time value is.

Avg Sync Read Time The average elapsed time used to perform a synchronous read. This value is derived by dividing the value of the Pool Sync Read Time attribute by the value of the Pool Sync Read attribute. This average is important because it might indicate the presence of an I/O wait, which in turn might indicate that you must move data to a different device.

Avg Sync Write Time The average elapsed time used to perform a synchronous write. This value is derived by dividing the value of the Pool Sync Write Time attribute by the value of the Pool Sync Write attribute. This average is important because it might indicate the presence of an I/O wait, which in turn might indicate that you must move data to a different device.

Binds Precompiles The number of binds and precompiles attempted. Use this attribute to gain insight into the current level of activity within the database manager.

Cat Cache Heap Full The number of times that an insert into the catalog cache failed because of a heap full condition in the database heap. The catalog cache draws its storage dynamically from the database heap. Even if the cache storage has not reached its limit, inserts into the catalog cache might fail due to a lack of space in the database heap. If the catalog cache heap full count is not zero, you can correct the insert failure condition by increasing the database heap size or by reducing the catalog cache size.

Cat Cache Hit Ratio The percentage of catalog sections that are found in the cache. This ratio indicates how well the catalog cache is avoiding catalog accesses. If the ratio is high (more than 0.8), the cache is performing well. A smaller ratio might indicate that you must increase the size of the catalog cache. You must expect a large ratio immediately following the first connection to the database.

Cat Cache Inserts The number of times that the system tried to insert table descriptor information into the catalog cache. Table descriptor information is inserted into the cache following a failed lookup to the catalog cache while processing a table, view, or alias reference in an SQL statement. The catalog cache inserts value includes attempts to insert table descriptor information that fail due to catalog cache overflow and heap full conditions.

Cat Cache Lookups The number of times that the catalog cache was referenced to obtain table descriptor information. This attribute includes both successful and unsuccessful accesses to the catalog cache. This attribute is used in calculating the catalog cache hit ratio. This ratio indicates how well the catalog cache is avoiding catalog accesses. If the ratio is high (more than 0.8), the cache is performing well. A smaller ratio might indicate that you must increase the size of the catalog cache. You must expect a large ratio immediately following the first connection to the database.

Cat Cache Overflows The number of times that an insert into the catalog cache failed because the catalog cache was full. If the catalog cache overflows value is large, the catalog cache might be too small for the workload. Increasing the size of the catalog cache might improve its performance. If the workload includes transactions that compile a large number of SQL statements referencing many tables, views, and aliases in a single unit of work, compiling fewer SQL statements in a single transaction might improve the performance of the catalog cache. Or if the workload includes the binding of packages containing many SQL statements referencing many tables, views or aliases, you might want to split the packages so that they include fewer SQL statements to improve performance.

Catalog Node Name The network name of the catalog node.

Commit SQL Stmts The total number of SQL COMMIT statements that have been attempted. A small rate of change in this counter during the monitor period might indicate that applications are not doing frequent commits. The lack of frequent commits can lead to problems with logging and data concurrency. You can also use this attribute to calculate the total number of units of work by calculating the sum of the following values:

- Commit statements attempted
- Internal commits
- Rollback statements attempted
- Internal rollbacks

Connections Top The highest number of simultaneous connections to the database since the database was activated. You can calculate the current number of connections at the time the snapshot was taken by adding the Remote Connections to Database Manager and Local Connections attributes. Use this attribute to evaluate the setting of the MAXAPPLS configuration parameter.

Coord Agents Top The maximum number of coordinating agents working at one time. The MAXCAGENTS configuration parameter determines the number of coordinating agents that can be executing concurrently. If the peak number of coordinating agents results in a workload that is too high for this node, you can reduce the MAXCAGENTS configuration parameter.

DB Conn Time The string date and time when the first database connection was made. The value format is CYYMMDDHHMMSSmmm, where the following measures apply:

Value	Description
C	Century (0 for 20th, 1 for 21st)
YY	Year
MM	Month
DD	Day
HH	Hour
MM	Minute
SS	Second
mmm	Millisecond

DB Heap Top This data attribute (now maintained for DB2 version compatibility) measures memory usage, but not exclusively usage by the database heap.

DB Location The location of the database in relation to the application. The following values are valid:

External value	Internal value
LOCAL	LOCAL
REMOTE	REMOTE
UNKNOWN	UNKNOWN

Determine the relative location of the database server with respect to the application taking the snapshot.

DB Name The real name of the database for which information is collected or to which the application is connected. This name was given to the database when it was created. The value format is a simple text string with a maximum of 20 characters. Use this attribute to identify the specific database to which the data applies.

DB Name (Unicode) The real name of the database for which information is collected or to which the application is connected (Unicode). This name was given to the database when it was created. The value format is a simple text string with a maximum of 60 bytes.

DB Partition The DB2 database partition node number, which can range from 0 to 999. The Aggregated and Current Partition values can be used within a query or situation filter. If a db partition filter is not specified, data is returned for the current database partition. If a db partition filter is set to Aggregated, only aggregated partition data is returned. Historical data collection includes both aggregated and individual partition attribute data. In addition to numeric partition numbers in the 0 to 999 range, the following values are also valid:

External value	Internal value
Aggregated	-1
Current Partition	-2
All	-3

DB Path The full path of the location where the database is stored on the monitored system. The value format is a simple text string with a maximum of 256 characters. Use this attribute with the Database Name attribute to identify the specific database to which the data applies.

DB Path (Unicode) The full path of the location where the database is stored on the monitored system (Unicode). The value format is a simple text string with a maximum of 768 bytes.

Database Status The status of the database. The following values are valid:

External value	Internal value	Description
Active	Active	The database is active.
InActive	InActive	The database is not active.
Quiesce Pending	Quiesce_Pending	The database is in quiesce-pending state. New connections to the database are not permitted and new units of work cannot be started. Depending on the quiesce request, active units of work are allowed to complete or are rolled back immediately.
Quiesced	Quiesced	The database has been quiesced. New connections to the database are not permitted and new units of work cannot be started.
Roll Forward	Roll_Forward	A rollforward is in progress on the database.
Unknown	Unknown	The status is unknown.

DDL SQL Stmts The number of SQL Data Definition Language (DDL) statements that ran. Use this attribute to determine the level of database activity at the application or database level. DDL statements are expensive to run because of their impact on the system catalog tables. As a result, if the value of this attribute is high, you must determine the cause and possibly restrict the identified activity from being performed. You can also use this attribute to determine the percentage of DDL activity using the following formula: divide the number of DDL SQL statements by the total number of statements.

Deadlocks The total number of deadlocks that have occurred. This attribute can indicate that applications are experiencing contention problems. To resolve the problem, determine in which applications (or application processes) the deadlock are occurring. You can then modify the application to enable it to run concurrently. Some applications, however, might not be capable of running concurrently.

Direct Read Reqs The number of requests to perform a direct read of one or more sectors of data. Use the following formula to calculate the average number of sectors that are read by a direct read:

```
direct reads from database / direct read requests
```

Direct Read Time The elapsed time (in milliseconds) required to perform the direct reads. Use the following formula to calculate the average direct read time per sector:

```
direct read time / direct reads from database
```

A high average time might indicate an I/O conflict.

Direct Reads The number of read operations that do not use the buffer pool. Use the following formula to calculate the average number of sectors that are read by a direct read:

```
direct reads from database / direct read requests
```

When using system monitors to track I/O, this data attribute helps to distinguish database I/O from non-database I/O on the device.

Direct Write Reqs The number of requests to perform a direct write of one or more sectors of data. Use the following formula to calculate the average number of sectors that are written by a direct write:

```
direct writes to database / direct write requests
```

Direct Write Time The elapsed time (in milliseconds) required to perform the direct writes. Use the following formula to calculate the average direct write time per sector:

```
direct write time / direct writes to database
```

A high average time might indicate an I/O conflict.

Direct Writes The number of write operations that do not use the buffer pool. Use the following formula to calculate the average number of sectors that are written by a direct write:

```
direct writes to database / direct write requests
```

When using system monitors to track I/O, this data attribute helps to distinguish database I/O from non-database I/O on the device.

Dynamic SQL Stmts The number of dynamic SQL statements that were attempted. Use this attribute to calculate the total number of successful SQL statements at the database or application level by performing the following operations:

1. Add the number of Dynamic SQL Statements Attempted and the Static SQL Statements Attempted.
2. Subtract the number of Failed Statement Operations.

The remainder equals the throughput (the number of successful SQL statements) during the current monitoring period.

Failed SQL Stmts The number of SQL statements that were attempted, but failed. This count includes all SQL statements that received a negative SQLCODE. Use this attribute to calculate the total number of successful SQL statements at the database or application level by performing the following operations:

1. Add the number of Dynamic SQL Statements Attempted and the Static SQL Statements Attempted.
2. Subtract the number of Failed Statement Operations.

The remainder equals the throughput (the number of successful SQL statements) during the current monitoring period. This attribute can also help you to determine the reasons for poor performance; failed statements indicate time wasted by the database manager, which results in lower throughput for the database.

Files Closed The total number of database files closed. The database manager opens files for reading and writing into and out of the buffer pool. The maximum number of database files open by an application at any time is controlled by the MAXFILOP configuration parameter. If the maximum is reached, one file is closed before the new file is opened. Note that the actual number of files opened might not equal the number of files closed.

Hash Join Overflows The number of times that hash join data exceeded the available sort heap space. At the database level, if the percentage of Hash Join Small Overflows is greater than 10% of this value, you must consider increasing the sort heap size. You can use values at the application level to evaluate hash join performance for individual applications.

Hash Join Small Overflows The number of times that hash join data exceeded the available sort heap space by less than 10%. If this value and the value of the Hash Join Overflows attribute are high, consider increasing the sort heap threshold. If this value is greater than 10% of Hash Join Overflows, consider increasing the sort heap size.

Int Deadlock Rollbacks The total number of forced rollbacks initiated by the database manager due to a deadlock. The database manager initiates a rollback for the current unit of work in an application that is experiencing a deadlock. This attribute shows the number of deadlocks that have been broken. It can indicate the possibility of concurrency problems. It is also important because internal rollbacks due to deadlocks can cause performance degradation.

Int Rollbacks The total number of rollbacks initiated internally by the database manager. An internal rollback occurs when any of the following operations cannot complete successfully:

- A reorganization
- An import
- A bind or pre-compile
- An application that ends as a result of a deadlock situation or lock timeout situation

- An application that ends without executing an explicit COMMIT or ROLLBACK statement (on Windows systems).

Use this attribute to calculate the total number of units of work by calculating the sum of the following values: commit statements attempted, internal commits, rollback statements attempted, and internal rollbacks.

Input DB Alias The alias of the database provided when calling the snapshot function. The value format is a simple text string with a maximum of 20 characters. Use this attribute to help you identify the specific database to which the monitor data applies. It contains blanks unless you requested monitor information related to a specific database.

Input DB Alias (Unicode) The alias of the database provided when calling the snapshot function (Unicode). The value format is a simple text string with a maximum of 60 bytes.

Instance Name (Unicode) The name of the monitored DB2 instance (Unicode). The valid format is a text string with a maximum of 60 bytes.

Last Backup The string date and time that the latest database backup was completed. The value format is CYYMMDDHHMMSSmmm, where the following measures apply:

Value	Description
C	Century (0 for 20th, 1 for 21st)
YY	Year
MM	Month
DD	Day
HH	Hour
MM	Minute
SS	Second
mmm	Millisecond

Use this attribute to help you identify a database that has not been backed up recently, or to identify which database backup file is the most recent. If the database has never been backed up, this timestamp is initialized to zero.

Lock Escals The number of times that locks have been escalated from several row locks to a table lock. A lock is escalated when the total number of locks held by an application reaches the maximum amount of lock list space available to the application, or the lock list space consumed by all applications is approaching the total lock list space. This data item includes a count of all lock escalations, including exclusive lock escalations. When an application reaches the maximum number of locks allowed and there are no more locks to escalate, the application uses space in the lock list that is allocated for other applications. When the entire lock list is full, an error occurs. The value format is an integer.

Lock List in Use The total amount of lock list memory (in bytes) that is currently in use. This attribute can be used with the locklist configuration parameter to calculate the lock list utilization. If the lock list utilization is high, you might want to consider increasing the size of that parameter. Values that are greater than or equal to 2147483647 are indicated in the portal with the `Value Exceeds Maximum` text, and values that are smaller than -2147483648 are indicated with the `Value Exceeds Minimum` text. The following values are valid:

External value	Internal value
Value Exceeds Maximum	2147483647
Value Exceeds Minimum	-2147483648

Lock Timeouts The number of times that a request to lock an object time out instead of being granted. This attribute can help you adjust the setting for the LOCKTIMEOUT database configuration parameter. If the number of lock timeouts becomes excessive when compared to normal operating levels, an application might be holding locks for long durations. In this case, this attribute might indicate that you must analyze some of the other attributes related to locks and deadlocks to determine if an application problem exists. It is also possible to have too few lock timeouts if the LOCKTIMEOUT database configuration parameter is set too high. In this case, applications might wait excessively to obtain a lock.

Lock Wait Time The total elapsed time (in milliseconds) that was spent waiting for a lock. At the database level, this is the total amount of elapsed time that all applications were waiting for a lock within this database. At the application-connection and transaction levels, this is the total amount of elapsed time that this connection or transaction has waited for a lock to be granted. Use this attribute with the Lock Waits attribute to calculate the average wait time for a lock. This calculation can be performed at either the database or the application-connection level. If the average lock wait time is high, you must look for applications that hold many locks, or have lock escalations, with a focus on tuning your applications to improve concurrency, if appropriate. If escalations are the reason for a high average lock wait time, the values of one or both of the LOCKLIST and MAXLOCKS configuration parameters might be too low.

Lock Waits The total number of times that applications or connections waited for locks. At the database level, the lock waits value is the total number of times that applications waited for locks within this database. At the application-connection level, the lock waits value is the total number of times that this connection requested a lock but waited because another connection was already holding a lock on the data. Use this attribute with the Lock Wait Time attribute to calculate, at the database level, the average wait time for a lock. This calculation can be performed at either the database or the application-connection level. If the average lock wait time is high, you must look for applications that hold many locks, or have lock escalations, with a focus on tuning your applications to improve concurrency, if appropriate. If escalations are the reason for a high average lock wait time, the values of one or both of the LOCKLIST and MAXLOCKS configuration parameters might be too low.

Locks Held The number of locks currently held. If the monitor information is at the database level, this number represents the total number of locks currently held by all applications in the database. If the information is at the application level, this number represents the total number of locks currently held by all agents for the application.

Locks Waiting The number of agents that are currently waiting on a lock. When used with Applications Currently Connected attribute, this attribute indicates the percentage of applications waiting on locks. If this number is high, the applications might have concurrency problems. In this case, you must identify applications that are holding locks or exclusive locks for long periods of time.

Log Reads The number of log pages read from disk by the logger.

Log Writes The number of log pages written to disk by the logger. Use this attribute with an operating system monitor to quantify the amount of I/O on a device that is attributable to database activity.

Node Name For new installations of version 6, release 2, the format is instanceid:hostname:UD for all operating systems. The format for version 6, release 1 of the DB2 agent on Windows systems is instanceid:hostname:UD; on UNIX and Linux systems, the format is instanceid:hostname.

Num Assoc Agents The current number of subagents associated with all applications that are connected to the monitored database. Use this attribute to evaluate the settings for the agent configuration parameters.

Pkg Cache Hit Ratio The percentage of package sections that were found in the cache. This ratio tells you whether the package cache is being used effectively. If the hit ratio is high (more than 0.8), the cache is performing well. A smaller ratio might indicate that the package cache must be increased.

Pkg Cache Inserts The total number of times that a requested section was not available for use and had to be loaded into the package cache. This count includes any implicit prepares performed by the system. By using the Package Cache Lookups attribute, you can calculate the package cache hit ratio using the following formula:

```
1 - (Package Cache Inserts / Package Cache Lookups)
```

See the Package Cache Lookups attribute for information about using this attribute.

Pkg Cache Lookups The number of times that an application looked for a section or package in the package cache. At a database level, it indicates the overall number of references since the database was started, or monitor data was reset. Note that this counter includes the cases where the section is already loaded in the cache and when the section has to be loaded into the cache. To calculate the package cache hit ratio use the following formula:

```
1 - (Package Cache Inserts / Package Cache Lookups)
```

The package cache hit ratio tells you whether the package cache is being used effectively. If the hit ratio is high (more than 0.8), the cache is performing well. A smaller ratio might indicate that the package cache must be increased.

Pool Async Data Read Reqs The number of asynchronous read requests. To calculate the average number of data pages read per asynchronous request, use the following formula:

```
buffer pool asynchronous data reads / buffer pool asynchronous read requests
```

This average can help to determine the amount of asynchronous I/O in each interaction with the prefetcher.

Pool Async Data Reads The number of pages read asynchronously into the buffer pool. Use this attribute with the Buffer Pool Data Physical Reads attribute to calculate the number of physical reads that were performed synchronously (that is, physical data page reads that were performed by database manager agents). Use the following formula:

```
buffer pool data physical reads - buffer pool synchronous data reads
```

By comparing the ratio of asynchronous to synchronous reads, you can gain insight into how well the prefetchers are working.

Pool Async Data Writes The number of times a buffer pool data page was physically written to disk by an asynchronous page cleaner or by a prefetcher. A prefetcher might have written dirty pages to disk to make space for the pages being prefetched. Use this attribute with the Buffer Pool Data Writes attribute to calculate the number of physical write requests that were performed synchronously (that is, physical data page writes that were performed by database manager agents). Use the following formula:

```
buffer pool data writes - buffer pool asynchronous data writes
```

By comparing the ratio of asynchronous to synchronous writes, you can gain insight into how well the buffer pool page cleaners are performing.

Pool Async Index Reads The number of index pages read asynchronously into the buffer pool by a prefetcher. Asynchronous reads are performed by database manager prefetchers. Use this attribute with the Buffer Pool Index Physical Reads attribute to calculate the number of physical reads that were performed synchronously (that is, physical index page reads that were performed by database manager agents). Use the following formula:

```
buffer pool index physical reads - buffer pool asynchronous index reads
```

By comparing the ratio of asynchronous to synchronous reads, you can gain insight into how well the prefetchers are working.

Pool Async Index Writes The number of times a buffer pool index page was physically written to disk by an asynchronous page cleaner or a prefetcher. A prefetcher might have written dirty pages to disk to make space for the pages being prefetched. Use this attribute with the Buffer Pool Index Writes attribute to calculate the number of physical index write requests that were performed synchronously. That is, physical index page writes that were performed by database manager agents. Use the following formula:

```
buffer pool index writes - buffer pool asynchronous index writes
```

By comparing the ratio of asynchronous to synchronous writes, you can gain insight into how well the buffer pool page cleaners are performing.

Pool Async Read Time The total elapsed time spent reading by database manager prefetchers. Use this attribute to calculate the elapsed time for synchronous reading, using the following formula:

```
total buffer pool physical read time - buffer pool synchronous read time
```

You can also use this attribute to calculate the average asynchronous read time using the following formula:

```
buffer pool asynchronous read time / buffer pool asynchronous data reads
```

These calculations can be used to understand the I/O work being performed.

Pool Async Write Time The total elapsed time spent writing data or index pages from the buffer pool to disk by database manager page cleaners. Calculate the elapsed time spent writing pages synchronously by subtracting the value of the Pool Async Write Time attribute from the value of the Pool Physical Write Time attribute. You can also use this attribute to calculate the average asynchronous read time by performing the following operations:

1. Add the Pool Async Data Writes and the Pool Async Index Writes.
2. Divide the Pool Async Write Time by the sum from step 1.

These calculations can be used to understand the I/O work being performed.

Pool Data from Estore Number of buffer pool data pages copied from extended storage. Required pages are copied from extended storage to the buffer pool. The copy process might incur the cost of connecting to the shared memory segment, but it saves the cost of a disk read.

Pool Data L Reads The number of logical read requests for data pages that have gone through the buffer pool. This count includes accesses to the following data:

- Data that is already in the buffer pool when the database manager needs to process the page
- Data that is read into the buffer pool before the database manager can process the page.

By using the Pool Data Physical Reads attribute, you can calculate the data page hit ratio for the buffer pool as follows:

```
1 - (buffer pool data physical reads / buffer pool data logical reads)
```

By using the Pool Data Physical Reads, Pool Index Physical Reads, and Pool Index Logical Reads attributes, you can calculate the overall buffer pool hit ratio as follows:

```
1 - ((buffer pool data physical reads + buffer pool index physical reads)
 / (buffer pool data logical reads + buffer pool index logical reads))
```

Increasing buffer pool size generally improves the hit ratio until you reach a point of diminishing returns.

Pool Data P Reads The number of read requests that required I/O to get data pages into the buffer pool. See Pool Data Logical Reads and Pool Async Data Reads attributes for information about how to use this attribute.

Pool Data to Estore Number of buffer pool data pages copied to extended storage. Pages are copied from the buffer pool to extended storage when they are selected as victim pages. As a result of the copying process, there is sufficient space for new pages in the buffer pool.

Pool Data Writes The number of times a buffer pool data page was physically written to disk. A buffer pool data page is written to disk for the following reasons:

- To free a page in the buffer pool so another page can be read
- To flush the buffer pool.

If a buffer pool data page is written to disk for a high percentage of Buffer Pool Data Physical Reads, performance might improve by increasing the number of buffer pool pages available for the database.

Pool Drty Pg Steal Clns Buffer Pool Victim Page Cleaners Triggered is the number of times a page cleaner was invoked because a synchronous write was needed during the victim buffer replacement for the database. Use this attribute, in combination with others, to evaluate the number of page cleaners that are defined.

Pool Drty Pg Thrsh Clns The number of times a page cleaner was invoked because a buffer pool had reached the dirty page threshold criterion for the database. When the number of dirty pages in the pool exceeds this value, the cleaners are triggered. If this value is set too low, pages might be written out too early, requiring them to be read back in. If set too high, too many pages might accumulate, requiring users to write out pages synchronously.

Pool Hit Ratio The buffer pool hit ratio (as a percentage). The sum of the Pool Data Logical Reads and Pool Index Logical Reads attributes is divided by the value of the Pool Total Reads attribute to derive the pool hit ratio. Use this attribute to determine whether buffer pool assignment is efficient. If the pool hit ratio is low, increasing the number of buffer pool pages might improve performance.

Pool Index from Estore Number of buffer pool index pages copied from extended storage. Required index pages are copied from extended storage to the buffer pool. The copy process might incur the cost of connecting to the shared memory segment, but it saves the cost of a disk read.

Pool Index L Reads The number of logical read requests for index pages that have gone through the buffer pool. This count includes accesses to the following index pages:

- Pages that are already in the buffer pool when the database manager needs to process the page.
- Pages that are read into the buffer pool before the database manager can process the page.

By using the Buffer Pool Index Physical Reads attribute, you can calculate the index page hit ratio for the buffer pool as follows:

```
1 - (buffer pool index physical reads / buffer pool index logical reads)
```

To calculate the overall buffer pool hit ratio, see the Buffer Pool Data Logical Reads attribute. If the hit ratio is low, increasing the number of buffer pool pages might improve performance.

Pool Index P Reads The number of physical read requests to get index pages into the buffer pool. See the Pool Index Logical Reads attribute for information about how to use this element.

Pool Index to Estore Number of buffer pool index pages copied to extended storage. Pages are copied from the buffer pool to extended storage when they are selected as victim pages. As a result of the copying process, there is sufficient space for new pages in the buffer pool.

Pool Index Writes The number of times a buffer pool index page was physically written to disk. If a buffer pool index page is written to disk for a high percentage of Buffer Pool Index Physical Reads, performance might improve by increasing the number of buffer pool pages available for the database. If all applications are updating the database, increasing the size of the buffer pool might have minimal impact on performance; most pages contain updated data that must be written to disk.

Pool LSN Gap Clns The number of times a page cleaner was invoked because the logging space used had reached a predefined criterion for the database. Use this attribute to evaluate whether you have enough space for logging, and whether you need more log files or larger log files.

Pool Read Time The total amount of elapsed time spent processing read requests that caused data or index pages to be physically read from buffer pool to disk. Use this attribute with the Buffer Pool Data Physical Reads and Buffer Pool Index Physical Reads attributes to calculate the average page-read time. This average is important because it might indicate the presence of an I/O wait, which in turn might indicate that you must move data to a different device.

Pool Sync Data Reads The number of physical data page reads that were performed by database manager agents. This value is derived by subtracting the value of the Pool Async Data Reads attribute from the Pool Data Physical Reads attribute. By comparing the ratio of asynchronous to synchronous reads, you can gain insight into how well the prefetchers are working.

Pool Sync Data Writes The total number of physical write requests that were performed synchronously (that is, physical data page writes that were performed by database manager agents). This value is derived by subtracting the value of the Pool Async Data Writes attribute from the value of the Pool Data Writes attribute. By comparing the ratio of asynchronous to synchronous writes, you can gain insight into how well the buffer pool page cleaners are performing.

Pool Sync Index Reads The number of index pages read synchronously (that is, physical index page reads that were performed by database manager agents) into the buffer pool. This value is derived by subtracting the value of the Pool Async Index Reads attribute from Pool Index Physical Reads attribute. By comparing the ratio of asynchronous to synchronous reads, you can gain insight into how well the prefetchers are working.

Pool Sync Index Writes The number of physical index write requests that were performed synchronously (that is, physical index page writes that were performed by database manager agents). This value is derived by subtracting the value of the Pool Async Index Writes attribute from the value of the Pool Index Writes attribute. By comparing the ratio of asynchronous to synchronous writes, you can gain insight into how well the buffer pool page cleaners are performing.

Pool Sync Read The total number of synchronous reads. This value is derived by adding the values of the Pool Sync Data Reads and Pool Sync Index Reads attributes.

Pool Sync Read Time The elapsed time used to perform all synchronous reads. This value is derived by subtracting the value of the Pool Async Read Time attribute from the value of the Pool Read Time attribute. Use this attribute to understand the I/O work being performed.

Pool Sync Write The total number of synchronous index writes. The value is derived by adding the values of the Pool Sync Data Writes attribute and Pool Sync Index Writes attribute.

Pool Sync Write Time The total elapsed time used to perform all synchronous writes. This value is derived by subtracting the value of the Pool Async Write Time attribute from the value of the Pool Write Time attribute.

Pool Total Reads The total number of read requests that required I/O to get data pages and index pages into the buffer pool. This attribute is the total of the Pool Data Physical Reads and Pool Index Physical Reads attributes. Values that are greater than or equal to 2147483647 are indicated in the portal with the `Value`

`Exceeds Maximum` text, and values that are smaller than -2147483648 are indicated with the `Value Exceeds Minimum` text. The following values are valid:

External value	Internal value
Value Exceeds Maximum	2147483647
Value Exceeds Minimum	-2147483648

Pool Total Writes The total number of write requests. This attribute is the total of the Pool Data Writes and Pool Index Writes attributes. Values that are greater than or equal to 2147483647 are indicated in the portal with the `Value Exceeds Maximum` text, and values that are smaller than -2147483648 are indicated with the `Value Exceeds Minimum` text. The following values are valid:

External value	Internal value
Value Exceeds Maximum	2147483647
Value Exceeds Minimum	-2147483648

Pool Write Time The total amount of time spent physically writing data or index pages from the buffer pool to disk. Use this attribute with the Buffer Pool Data Writes and Buffer Pool Index Writes attributes to calculate the average page-write time. This average is important because it might indicate the presence of an I/O wait, which in turn might indicate that you must move data to a different device.

Prefetch Wait Time The time an application spent waiting for an I/O server (prefetcher) to finish loading pages into the buffer pool. This attribute can be used to experiment with changing the number of I/O servers and the I/O server sizes.

Rollback SQL Stmts The total number of SQL ROLLBACK statements that have been attempted. A rollback can result from an application request, a deadlock, or an error situation. This attribute counts only the number of rollback statements issued from applications. At the application level, this attribute can help you determine the level of database activity for the application and the amount of conflict with other applications. At the database level, it can help you determine the amount of activity in the database and the amount of conflict between applications on the database.

Rows Deleted The number of row deletions attempted. Use this attribute to gain insight into the current level of activity within the database manager.

Rows Inserted The number of row insertions attempted. Use this attribute to gain insight into the current level of activity within the database manager.

Rows Selected The number of rows that have been selected and returned to the application. Use this attribute to gain insight into the current level of activity within the database manager.

Rows Updated The number of row updates attempted. Use this attribute to gain insight into the current level of activity within the database manager.

Sec Logs Allocated The total number of secondary log files that are currently being used for the database. Use this attribute with the Secondary Log Used Top and Total Log Used Top attributes to show the current dependency on secondary logs. If this value is consistently high, you might need larger log files, more primary log files, or more frequent COMMIT statements within your application.

Sec Log Used Top The maximum amount of secondary log space (in bytes) that has been used. Use this attribute with the Secondary Logs Allocated and Total Log Used Top attributes to show the current dependency on secondary logs. If this value is high, you might need larger log files, more primary log files, or more frequent COMMIT statements within your application. Values that are greater than or equal to 2147483647 are indicated in the portal with the `Value Exceeds Maximum` text, and values that are smaller than -2147483648 are indicated with the `Value Exceeds Minimum` text. The following values are valid:

External value	Internal value
Value Exceeds Maximum	2147483647
Value Exceeds Minimum	-2147483648

Select SQL Stmts The number of SQL SELECT statements that ran. Use this attribute to determine the level of database activity at the application or database level. You can also use the following formula to determine the ratio of SELECT statements to the total statements by performing the following operations:
- Add the number of static SQL statements attempted and dynamic SQL statements attempted
- Divide the resulting total by the number of select SQL statements that ran

Server Platform The operating system upon which the database management system is running. Use this attribute during troubleshooting for remote applications.

Snapshot Time The string date and time when the database system monitor information was collected. Use this attribute to help relate data chronologically if you are saving the results in a file or database for ongoing analysis. The Tivoli Enterprise Portal formats the timestamp value into a date and time string. The internal timestamp value stored in the database is in the format cYYMMDDhhmmss000, which is described in the following table:

Value	Description
C	Century (0 for 20th, 1 for 21st)
YY	Year (last two digits of the year, for example (00 - 99)
MM	Month (01 for January, 02 for February, and so on)
DD	Day (the day of the month, for example 01 - 31)
HH	Hour (the hour of the day in 24-hour format from 00 - 23)
MM	Minute (the minutes of the hour from 00 - 59)
SS	Second (the seconds of the hour (00 - 59)

Sort Heap Allocated The total number of allocated pages of sort heap space for all sorts at the level chosen (database manager or database) and at the time the snapshot was taken. Memory estimates do not include sort heap space. If excessive sorting occurs, add the extra memory (used for the sort heap) to the base memory requirements for running the database manager. Generally, the larger the sort heap, the more efficient the sort. Appropriate use of indexes can reduce the amount of sorting required.

Sort Overflows The total number of sorts that ran out of sort heap space and might have required disk space for temporary storage. at the database or application level, use this attribute with the Total Sorts attribute to calculate the percentage of sorts that required overflow to disk. If this percentage is high, you might want to adjust the database configuration by increasing the value of the SORTHEAP configuration parameter.

Sort Overflows Pct The percentage of sorts that ran out of sort heap space and might have required disk space for temporary storage. This percentage is calculated by dividing the value of the Sort Overflows attribute by the value of the Total Sorts attribute. at the database or application level, use this attribute to evaluate the percentage of sorts that required overflow to disk. If this percentage is high, you might want to adjust the database configuration by increasing the value of the SORTHEAP configuration parameter.

SQL Stmts Failed Pct The percentage of SQL statements that failed to run successfully. This value is derived by dividing the value of the Failed SQL Statements attribute by the value of the Total SQL Statements attribute. Use this attribute to determine whether an application has some design issues.

SQL Stmts Rollback Pct The percentage of SQL statements that resulted in a rollback. This value is derived by dividing the value of the Rollback SQL Statements attribute by the value of the the Total SQL Statements attribute. Use this attribute to determine whether an application has some design issues.

Static SQL Stmts The number of static SQL statements that were attempted. Use this attribute to calculate the total number of successful SQL statements at the database or application level by performing the following operations:

1. Add the number of Dynamic SQL Statements Attempted and the Static SQL Statements Attempted.
2. Subtract the number of Failed Statement Operations.

The remainder equals the throughput (the number of successful SQL statements) during the current monitoring period.

Total Log Used Top The maximum amount of total log space (in bytes) that has been used. Use this attribute to evaluate the amount of primary log space that is allocated. Comparing the value of this attribute with the amount of primary log space that is allocated can help you to evaluate the configuration parameter settings. Values that are greater than or equal to 2147483647 are indicated in the portal with the `Value Exceeds Maximum` text, and values that are smaller than -2147483648 are indicated with the `Value Exceeds Minimum` text. The following values are valid:

External value	Internal value
Value Exceeds Maximum	2147483647
Value Exceeds Minimum	-2147483648

Total Cons The number of connections to the database since the first connect, activate, or last reset (coordinator agents). Use this attribute with the Database Activation Timestamp and the Start Database Manager Timestamp attributes to calculate the frequency at which applications have connected to the database. The first connect to a database (such as an initial buffer pool allocation) causes extra overhead. If the frequency of connects is low, it might be beneficial to activate the database explicitly using the ACTIVATE DATABASE command before connecting any other application. As a result, subsequent connects are processed at a higher rate.

Important: When you reset this attribute, the value of the attribute is set to the number of applications that are currently connected, instead of zero.

Total Hash Joins The total number of hash joins that ran. At the database or application level, use this value with the Hash Join Overflows attribute and the Hash Join Small Overflows attribute to determine if a significant percentage of hash joins would benefit from modest increases in the sort heap size.

Total Hash Loops The total number of times that a single partition of a hash join was larger than the available sort heap space. This attribute might indicate inefficient execution of hash joins (the sort heap size is too small or the sort heap threshold is too small). Use this value with the other hash join variables to tune the sort heap size (SORTHEAP) and sort heap threshold (SHEAPTHRES) configuration parameters.

Total Sec Cons The number of connections made by a subagent to the database at the node. Use this attribute with the Connects Since Database Activation, Database Activation Timestamp, and the Start Database Manager Timestamp attributes to calculate the frequency at which applications have connected to the database.

Total Sort Time The total elapsed time (in milliseconds) for all sorts that ran. at the database or application level, use this attribute with the Total Sorts attribute to calculate the average sort time, which can indicate whether sorting is a performance issue. Elapsed times are affected by system load. The more processes you have running, the higher this elapsed time value is.

Total Sorts The total number of sorts that ran. at the database or application level, use this value with the Sort Overflows attribute to calculate the percentage of sorts that need more heap space. You can also use it with the Total Sort Time attribute to calculate the average sort time. If the number of sort overflows is small with respect to the total sorts, increasing the sort heap size might have little impact on performance, unless this buffer size is increased substantially.

Total SQL Stmts The total number of dynamic and static statements. This value is derived by adding the values of the Dynamic SQL Statements and the Static SQL Statements attributes.

UID SQL Stmts The number of SQL UPDATE, INSERT, and DELETE statements that ran. Use this attribute to determine the level of database activity at the application or database level. You can also use the following formula to determine the ratio of UPDATE, INSERT, and DELETE statements to the total number of statements:

1. Add the number of static SQL statements attempted and the dynamic SQL statements attempted.
2. Divide the number of UPDATE/INSERT/DELETE SQL statements that ran by the sum derived in step 1.

X Lock Escals The number of times that locks have been escalated from several row locks to one exclusive table lock, or the number of times an exclusive lock on a row caused the table lock to become an exclusive lock. A lock is escalated when the total number of locks held by an application reaches the maximum amount of lock list space available to the application. The amount of lock list space available is determined by the LOCKLIST and MAXLOCKS configuration parameters. Other applications cannot access data held by an exclusive lock. Because exclusive locks can affect the concurrency of your data, it is important to track them. When an application reaches the maximum number of locks allowed and there are no more locks to escalate, it uses space in the lock list allocated for other applications. When the entire lock list is full, an error occurs. See the Lock Escals attribute for possible causes and resolutions to excessive exclusive lock escalations.

Database01 (KUD_DB2_Database01) attributes

Use these attributes to obtain information about the efficiency of the database and identify any problem areas for corrective action. All values are integers that are calculated from the first application connection, unless otherwise noted.

Appl Control Heap Size The maximum size (in 4-KB pages) for the application control heap in the database during the monitoring interval. The heap is required to share information among agents working on behalf of the same application at a node in a massively parallel processing (MPP) or a symmetric multiprocessor (SMP) system. If complex applications are being run or the MPP configuration has a large number of nodes, you must increase the size of this heap. In a partitioned database environment, this heap is used to store copies of the executing section of SQL statements for agents and subagents. However, symmetric multiprocessor agents (SMP), subagents, and agents in all other environments use appl heap size.

Appl Heap Size The size (in 4-KB pages) of the application heap that is available for each individual agent in the database during the monitoring interval. Increase the value of the parameter if your application receives an error indicating that there is not enough storage in the application heap. The heap is allocated when an agent or subagent is initialized for an application. The amount allocated is the minimum amount needed to process the request given to the agent or subagent. When the agent or subagent requires more heap space to process larger SQL statements, the database manager allocates memory as needed, up to the maximum specified by the parameter.

Appl Section Inserts The number of inserts of SQL sections by an application from its SQL work area. The working copy of any executable section is stored in a unique SQL work area. This value represents the number of times when a copy was not available and therefore was inserted. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Appl Section Lookups The number of lookups of SQL sections by an application from its SQL work area. This counter indicates how many times the SQL work area was accessed by agents for an application. It is a cumulative total of all lookups on all SQL work heaps for agents working on this application. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Appls in DB2 The number of applications currently executing in the database. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Appls The value of the average number of active applications.

Avg Data Page Read per Async Req The average number of pages read for each asynchronous request. This value is derived by dividing the value of the Pool Async Data Reads attribute by the value of the Pool Async Data Read Reqs attribute. Use this attribute to determine whether good enough data pages were read per asynchronous request. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Direct Read Time The average time in milliseconds that is used to perform direct reads to the database. The value is derived through this formula:

```
direct read time / direct reads
```

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Direct Write Time The average time in milliseconds for performing direct writes to the database. A high average time can indicate the existence of an input and output conflict. The value is derived through this formula:

```
direct write time / direct write
```

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Lock Escal per Conn for Interval The average lock escalations per connection for this database during the monitoring interval. The value format is an integer. A lock is escalated when the total number of locks that an application holds reaches the maximum amount of lock list space available to the application, or the lock list space consumed by all applications is approaching the total lock list space. When an application reaches the maximum permitted number of locks and no additional locks can be escalated, the application uses space in the lock list that is allocated for other applications. When the entire lock list is full, an error occurs. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Locks Held The average number of locks held by each currently connected application in the database. The value is derived through this formula:

```
locks held / appls cur cons
```

If the returned value is high compared to normal operating levels, it can indicate that one or more applications is using an excessive number of locks. Refine such applications to improve performance.

The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Pages per Cleaner for Interval The average number of pages written per page cleaner that are invoked for the database during the monitoring interval. Use the returned value to determine how many pages are handled by the page cleaners of this database. If this value increases over time, you can define more page cleaners. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Pool Async Data Reads The average number of buffer pool asynchronous data reads when compared to the total number of pool reads for the database. The value is derived through this formula: `pool async data reads / (pool data p reads + pool index p reads)`. Use the returned value to gain insight into how well the prefetchers are working and to refine the num_ioservers configuration parameter. If the returned value is low compared to normal operating levels, there might not be enough input and output servers to prefetch data into the buffer, causing the database manager agents to spend extra time on physical reads. Increase the number of input and output servers by increasing the value of the num_ioservers configuration parameter. If too many servers are allocated, system performance is not reduced because the extra input and output servers are not used. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Pool Async Data Writes The average number of buffer pool asynchronous data writes (data and index) when compared to the total number of pool writes for the database. The value is derived through this formula: `pool async data writes / (pool data writes + pool index writes)`. Use the returned value to gain insight into how well the page cleaners are working and to refine the num_iocleaners configuration parameter. If the returned value is low compared to normal operating levels, increase the number of input and output cleaners by increasing the value of the num_iocleaners parameter. If the returned value is high compared to normal operating levels, you can save system resources by decreasing the number of input and output cleaners (by decreasing the value of the num_iocleaners parameter). The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Pool I/O Time The average time (in milliseconds) for performing buffer pool input and output operations (reading or writing) to the database. A high average time can indicate the existence of an input and output conflict. In this case, you might need to move data to a different device. The returned value includes the time applied to asynchronous input and output operations (which are performed by prefetchers and page cleaners). The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Pool Writes per Read The ratio of total pool writes to pool reads for the database. The value is derived through this formula: `(pool data writes + pool index writes) / (pool data p reads + pool index p reads)`. If the returned value is greater than 1, you can improve performance by increasing the available buffer pool space. A returned value greater than 1 indicates that at least one write to disk had to occur (either to free a page in the buffer pool, or to flush the buffer pool) before a page can be read into the buffer pool. You can increase the available buffer pool space by freeing the space more often or by increasing the total space for the buffer pool. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Sect Read per Direct Read The average number of sectors that are read by a direct read for the database. The value is derived through this formula: `direct reads / direct read reqs`. Direct reads do not use the buffer pool, and so result in poor performance because the data is physically read from disk each time. If you are using system monitors to track input and output for the device, this value helps you distinguish database input and output from non-database input and output. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Sect Written per Direct Write The average number of sectors that are written by a direct write to the database. The value is derived through this formula: `direct writes / direct write reqs`. Direct writes do not use the buffer pool, which results in poor performance because the data is physically written from disk each time. If you are using system monitors to track input and output for the device, this value helps you distinguish database input and output from non-database input and output. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Sync I/O Time The average time (in milliseconds) to perform synchronous input and output operations for the database. Use the returned value to analyze the input and output work being performed for the database. Synchronous input and output operations for a database are performed by database manager agents. Asynchronous input and output operations are performed by prefetchers (reads) and page cleaners (writes). In general, asynchronous input and output helps your applications run faster. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Sync Read Time The average elapsed time used to perform a synchronous read. This value is derived by dividing the value of the Pool Sync Read Time attribute by the value of the Pool Sync Read attribute. This average is important because it might indicate the presence of an I/O wait, which in turn might indicate that you must move data to a different device. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Sync Write Time The average elapsed time used to perform a synchronous write. This value is derived by dividing the value of the Pool Sync Write Time attribute by the value of the Pool Sync Write attribute. This average is important because it might indicate the presence of an I/O wait, which in turn might indicate that you must move data to a different device. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Binds Precompiles The number of binds and precompiles attempted. Use this attribute to gain insight into the current level of activity within the database manager. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Buff Page Use the returned value (in units of 4-KB pages) to analyze the input and output work being performed for the database. Synchronous input and output operations for a database are performed by database manager agents. Asynchronous input and output operations are performed by prefetchers (reads) and page cleaners (writes). In general, asynchronous input and output helps your applications run faster. In the currently supported releases of DB2, multiple buffer pools might be defined in a single database. For instance, buffer pools can be defined and associated with a particular tablespace. Each buffer pool created can be given its own individual size. The buffpage attribute serves only as a default value for buffer pools created within a particular database. Therefore, the value of the buffpage attribute is much less critical to performance in current releases of DB2, because most buffer pools are given an individual size when created. The buffpage attribute must not be used to evaluate or tune the performance of DB2 unless it is used as the default value when creating buffer pools in a database. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Catalog Cache Size The value in units of 4-KB pages of the catalog cache size. This value is the maximum amount of space that the catalog cache can use from the database heap (dbheap). The catalog cache is referenced whenever a table, view, or alias name is processed during the compilation of an SQL statement. It is dynamically allocated from dbheap, as required, until the catalog cache size is reached.

Change Pages Threshold The value in percentage units of the changed pages threshold. This value sets a limit on how much buffer pool space can be occupied by changed pages before the asynchronous page cleaners are started, if they are not currently active. Asynchronous page cleaners write changed pages from the buffer pool to disk before the space in the buffer pool is required by a database agent. This means that the agents do not need to wait for a changed page to be written out before being able to read a page, and application transactions run faster.

Commit Stmts per Sec The total number of commits initiated internally by the database per second. Use the returned value to determine rates of database activity. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Cur Cons Percent The percentage of applications currently connected.

Current Primary Log Used Percent The percentage of primary log space that is currently in use.

Current Secondary Log Used Percent The percentage of secondary log space that is currently in use.

Database Heap The value in units of 4-KB pages of the database heap. This value is the maximum amount of memory allowed for a database heap. There is one database heap for each database. It is used on behalf of all applications connected to the database. Refining dbheap has minimal impact on performance. The main function of this parameter is to prevent the database manager from allocating an excessive amount of space for a particular database. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Days Since Last Backup The numbers of days since the last database backup was completed. The value format is an integer. The value for no backup completed is 2147483647. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

DB Capture Error The number of errors encountered by the Capture program within the last five minutes. Use the returned value to determine whether the Capture program encountered an error that prevented it from running. If any errors are detected, the Capture program came down at the time the error occurred. The Capture program might or might not still be down. The Capture program is the most critical replication component in the replication system. If the Capture program is not active, there are no new change records to apply to the target systems. If your data concurrency requirements are high and you want to ensure that the Capture program runs continuously, use this monitor to determine when the Capture program encounters an error that prevents it from running. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Values for this attribute are only available in a SQL replication environment.

DB Capture Lag The time difference in minutes between the current timestamp and the last timestamp recorded by the Capture program. This time difference is the Capture lag. Use the returned value to determine whether the Capture program is keeping up with the DB2 database log. The Capture program uses an interface to the DB2 database log or journal to detect and save changes to the data in the tables registered for replication. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Values for this attribute are only available in a SQL replication environment.

DB Capture Prun The number of rows in the unit-of-work table. Use the returned value to help you determine whether you need to prune the unit -of-work (UOW) table or the change data (CD) table. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Values for this attribute are only available in a SQL replication environment.

DB Heap Top This data attribute (now maintained for DB2 version compatibility) measures memory usage, but not exclusively usage by the database heap. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

DB Name The real name of the database for which information is collected or to which the application is connected. This name was given to the database when it was created. The value format is a simple text string with a maximum of 60 characters.

DB Partition The DB2 database partition node number, which can range from 0 to 999. The Aggregated and Current Partition values can be used within a query or situation filter. If a db partition filter is not specified, data is returned for the current database partition. If a db partition filter is set to Aggregated, only aggregated partition data is returned. Historical data collection includes both aggregated and individual partition attribute data. In addition to numeric partition numbers in the 0 to 999 range, the following values are also valid:

External value	Internal value
Aggregated	-1
Current Partition	-2
All	-3

DB Tablespaces The number of Database Managed Space tablespaces in the database. Use this attribute to track database growth over a period of time. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

DDL SQL Percent for Interval The percentage of total SQL statements that were SQL DDL statements during the monitoring interval. Due to the high activity in the system catalog tables, try to keep DDL statement activity to a minimum. If the returned value is high compared to normal operating levels, determine the activity causing it to be high and restrict it from being performed. Examples of DDL statements are CREATE TABLE, CREATE VIEW, ALTER TABLE, and DROP INDEX. You can also use the returned value to refine the package cache hit ratio for this application. DDL statements can also affect the package cache by invalidating sections that are stored there and causing additional system overhead due to section recompilation.

Deadlock Rollbacks Percent The percentage of the total number of rollbacks that deadlock caused. The value format is an integer.

Deadlocks for Interval The number of deadlocks detected in the database during the monitoring interval. Use the returned value to determine whether applications are experiencing conflict problems in the database. You can resolve the problem by determining in which applications the deadlocks are occurring. You can then try to modify the applications to better enable them to run concurrently. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Estore Read/Write Ratio for Interval The ratio as a percentage of data and index pages copied from extended storage to pages copied to extended storage during the monitoring interval. When a page is transferred from extended storage to the buffer pool, you save a system input and output call. However, you still incur the cost of attaching to the extended memory segment, copying the page, and detaching from the segment. Use the returned value to determine if you would benefit from using extended storage. The higher the ratio, the more likely you are to benefit. In general, extended storage is particularly useful if input and output activity is very high on your system. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Event Monitors The number of event monitors defined in the database. Use the returned value to determine how many event monitors are defined for the database. When you define an event monitor, its definition is stored in the database system catalog table. You can create any number of event monitors. However, the maximum number of event monitors that can be active for a database at any given time is 32. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Failed SQL Stmts Percent for Interval The percentage of total Structured Query Language statements that failed during the monitoring interval. The value format is an integer.

Instance Name The name of the monitored DB2 instance. The valid format is a text string with a maximum of 60 bytes.

Internal Auto Rebinds The number of automatic rebinds or recompiles that were attempted in the database. Use the returned value to determine the level of database activity. Automatic rebinds are the internal binds that the system performs when a package is invalidated. They can have a significant impact on performance and must be minimized where possible. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Internal Commits The total number of commits initiated internally by the database. Use the returned value to gain insight into internal activity within the database. The returned value is also used in calculating "Commit statements per second." The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Internal Deadlock Rollbacks Percent The percentage of the total number of internal rollbacks due to deadlocks. Use the returned value to distinguish those rollbacks caused by internal deadlocks from rollbacks caused by other situations (for example, incomplete imports). The returned value is the percentage of internal rollbacks due to internal deadlocks since the first database connection or the last reset of the database monitor counters.

Internal Deadlock Rollbacks Percent for Interval The percentage of rollbacks that were due to deadlock during the monitoring interval. Use the returned value to distinguish those rollbacks caused by internal deadlocks from rollbacks caused by other situations (for example, incomplete imports). The returned value is the percentage of internal rollbacks due to internal deadlocks since the first database connection or the last reset of the database monitor counters.

Internal Rows Deleted The number of rows deleted from the database as a result of internal activity. Use the returned value to gain insight into internal activity within the database. If this activity is high compared to normal operating levels, you can evaluate your table design to determine if the referential constraints or triggers that you defined on your database are necessary. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Internal Rows Inserted The number of rows inserted into a database as a result of internal activity caused by triggers. Use the returned value to gain insight into internal activity within the database. If this activity is high compared to normal operating levels, you can evaluate your design to determine if you can alter it to reduce this activity. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Internal Rows Updated The number of rows updated in the database as a result of internal activity. Use the returned value to gain insight into internal activity within the database. If this activity is high compared to normal operating levels, you can evaluate your table design to determine if the referential constraints that you defined are necessary. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Invalid Packages The number of all packages that are currently marked not valid in the database. Use the returned value as an indication of the current number of packages that are not valid. A package is marked not valid if it depends on an object (for example, a table) and that object is dropped. The number of packages that are not valid can indicate how many automatic rebinds are necessary in the database. Such packages automatically rebound the next time they are accessed, unless a trigger was dropped or the dropped object was not recreated. Use of automatic rebinds can significantly lower performance, and must be minimized if possible. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Invalid System Packages The number of system packages that are currently marked not valid in the database. Use the returned value as an indication of the current number of nonvalid packages owned by the system. A package is marked not valid if it depends on an object (for example, a table) and that object is dropped. The number of packages that are not valid can indicate how many automatic rebinds are necessary in the database. The package is automatically rebound the next time it is accessed, unless it was marked not valid because a trigger was dropped or because the dropped object was not recreated. Use of automatic rebinds can significantly lower performance, and must be minimized where possible. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Invalid Triggers The number of triggers that are marked not valid in the database. Use the returned value to determine the number of triggers that must be revalidated. A trigger is marked not valid if an object on which the trigger depends is dropped. To revalidate such a trigger, retrieve its definition from the database system catalog and submit a new CREATE TRIGGER statement. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

LESCI

Lock Escalation for Interval The total number of lock escalations for applications connected to this database during the monitoring interval. Exclusive lock escalations are included in this number. Use the returned value to help you evaluate the settings of the LOCKLIST and MAXLOCKS configuration parameters. Lock escalations can result in a decrease in concurrency among the applications connected to a database. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Lock List The value in units of 4-KB pages of the maximum storage for lock lists. This value is the amount of storage that is allocated to the lock list. There is one lock list for each database, and it contains the locks held by all applications concurrently connected to the database. Too small a value can lead to excessive lock waits. Too high a value compared to normal operating levels can deprive the system of resources or memory. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Lock List in Use Percent The percentage of space used in the locklist of this database. Use the returned value to determine how much of the locklist space is free for new locks to be requested.

Lock Timeouts for Interval The number of times that a request to lock an object were timed out instead of being granted during the monitoring interval. The value format is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Lock Waits for Interval The number of times that applications had to wait for locks in the database during the monitoring interval. Use the returned value as an indication of how much time is applied to waiting for locks during a particular monitoring interval. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Lock Waits Percent The percentage of currently connected applications that are waiting for a lock in the database. The value is derived through this formula:

```
100 * locks waiting / appls cur cons
```

If the returned value is high compared to normal operating levels, the applications can have concurrency problems. You must identify applications that are holding locks or exclusive locks for long periods of time and determine whether they can release their locks more often.

Log Buffer Size This value specifies the amount of the database heap to use as a buffer for log records before writing these records to disk. It is important that the log buffer can hold the amount of log space used by an average transaction. Otherwise, logging performance decreases and slows the overall system.

Log I/O for Interval The total amount of log input and output. This amount is the sum of the number of log pages read and the number of log pages written within the monitoring interval. Use the returned value to determine whether you must move the log to a different device. If this input and output is beyond the capabilities of the current device, you can determine if moving the log (by changing the newlogpath configuration parameter) improves performance. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Log Primary The number of primary log files.

Max Active Applications The value of the maximum number of active applications. This value is the maximum number of concurrent applications that can be connected (both local and remote) to a database. Because each application that attaches to a database causes some private memory to be allocated, allowing a large number of concurrent applications potentially uses more memory. Increasing the value of this parameter without lowering the maxlocks parameter or increasing the locklist parameter can cause you to reach the database limit on locks (locklist) rather than the application limit. The result can be pervasive lock escalation problems.

Max Locks The value of the maximum percentage of lock list before escalation. This value specifies the percentage of the lock list that an application can hold before the database manager performs lock escalation. Lock escalation can increase contention, which reduces system throughput and increases user response time. The values for the maxlocks and maxappls parameters must satisfy (MAXLOKS x MAXAPPLS) >100, and each lock uses 32 bytes. Rebind application packages after changing this parameter.

Min Commit The value of the number of commits to group. By using this parameter you can delay the writing of log records to disk until a minimum number of commits have been performed. This delay can help reduce the overhead associated with writing log records and can improve performance. The default value for mincommit is 1, which can be too low for your environment. By sampling the number of transactions per-second throughout the day, you can determine the peak per second rate and adjust the value of the mincommit parameter to accommodate all or most transactions. This adjustment minimizes the number of log writes under the heaviest conditions. As you increase the value of the mincommit parameter, you might also need to increase the log buffer size (LOGBUFSZ parameter) to avoid filling the log buffer. Filling the log buffer also forces the writing of log records to disk. If you change the value of the mincommit parameter, you must change the value for the logbufsz configuration parameter.

New Log Path The current value of the newlogpath configuration parameter. A valid value is a text string up to 768 characters in length. You use the newlogpath configuration parameter to specify a new location for the log files. The specified path does not become the current log path until both of the following conditions are met:

- The database is in a consistent state.
- All users are disconnected from the database.

When the first new connection is made to the database, the database manager moves the logs to this location.

Node Name For new installations of version 6, release 2, the format is instanceid:hostname:UD for all operating systems. The format for version 6, release 1 of the DB2 agent on Windows systems is instanceid:hostname:UD; on UNIX and Linux systems, the format is instanceid:hostname.

Number of I/O Cleaners The current value of the number of asynchronous page cleaners. This parameter specifies the number of asynchronous page cleaners for a database. Page cleaners monitor the buffer pool and asynchronously write out changed pages to disk to free space in the buffer pool.

Num IO Servers The current value of the number of input and output servers. This value specifies the number of input and output servers for a database. Input and output servers are used on behalf of the database agents to perform asynchronous input and output operations for utilities such as backup and restore, and to perform prefetch input and output (in which case, they are called prefetchers) operations. Prefetchers read pages from disk into the buffer pool in anticipation of their use. In most situations, these pages are read just before they are needed. However, prefetchers can cause unnecessary input and output operations by reading pages into the buffer pool that might not be used. For example, an application starts reading through a table, and prefetchers read consecutive pages into the buffer pool before the pages are required by the application. Then the application fills the application buffer and stops reading. Meanwhile, the prefetchers already have performed the input and output operations for additional pages and the buffer pool is partially taken up with those pages. To exploit all the input and output devices in the system, a good value for num_ioservers to use is one or two more than the number of physical devices on which the database is established.

Page Cleans for Interval The number of times a page cleaner was invoked for the database (for any reason) during the monitoring interval. Use the returned value to determine how often pages are written to disk by the page cleaners of this database. If this value increases over time, you can define more page cleaners. The number of page cleaners is determined by the number of I/O cleaners configured. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pages per Prefetch for Interval The number of data pages read per prefetch request for the database during the monitoring interval. Use the returned value to determine the amount of asynchronous input and output in each interaction with the prefetcher. An excessively low returned value when compared to normal operating levels indicates that you need more input and output servers. The more input and output servers that you have, the better your query performance. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Package Cache Size The current value in units of 4-KB pages of the package cache size. This value controls the amount of application heap memory to be used for caching static and dynamic SQL statements of a package. You must experiment with the size of the package cache to find the optimal number for this attribute. For example, you can use a smaller package cache size if there is no increase in the number of package cache inserts when you decrease the size of the cache. Decreasing the package cache size frees up system resources for other work. However, increasing the package cache size can improve overall system performance if it results in a decrease of package cache inserts. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Hit Ratio Index Percent for Interval The database index page hit ratio (as a percentage) for the buffer pool during the monitoring interval. The index page hit ratio for the buffer pool indicates the percentage of index page requests for which the database manager did not need to load an index page from disk to service. That is, the index page was already in the buffer pool. The higher the returned value, the lower the frequency of disk input and output, and the faster the performance. If the hit ratio is low compared to normal operating levels, increasing the number of buffer pool pages can improve performance.

Pool Hit Ratio Percent for Interval The overall buffer pool hit ratio as a percentage for the database during the monitoring interval. This hit ratio includes both index and data page activity. The overall buffer pool hit ratio indicates the percentage of page requests for which the database manager did not need to load a page from disk to service. (That is, the page was already in the buffer pool.) The greater the buffer pool hit ratio, the lower the frequency of disk input and output. If the hit ratio is low compared to normal operating levels, increasing the number of buffer pool pages can improve performance. A ratio of zero indicates that pages needed to be read for every request. For a large database, increasing the buffer pool size can have a minimal effect on the buffer pool hit ratio. Such a database can have so large a number of data pages that the statistical chance of a hit is not increased by an increase of the buffer pools. However, even though the data might be too large to fit in the buffer pool, the entire index can fit. In this case, you can refine buffer pool sizes until the overall buffer pool hit ratio stops increasing, and then refine the buffer pool until the buffer pool index hit ratio no longer increases.

Pool I/O per Second The rate (per second) of buffer pool input and output operations for the database. Buffer pool input and output includes all physical data and index pages that go through the buffer pool when read or written. Use the returned value to determine how efficient your data storage device is. A low value indicates the presence of an input and output wait, in which case you must move data to a different device. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Sync Data Writes The total number of physical write requests that were performed synchronously (that is, physical data page writes that were performed by database manager agents). This value is derived by subtracting the value of the Pool Async Data Writes attribute from the value of the Pool Data Writes attribute. By comparing the ratio of asynchronous to synchronous writes, you can gain insight into how well the buffer pool page cleaners are performing. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Sync Index Reads The number of pool index physical reads minus the pool async index reads. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Sync Index Writes The number of physical index write requests that were performed synchronously (that is, physical index page writes that were performed by database manager agents). This value is derived by subtracting the value of the Pool Async Index Writes attribute from the value of the Pool Index Writes attribute. By comparing the ratio of asynchronous to synchronous writes, you can gain insight into how well the buffer pool page cleaners are performing. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Sync Read The total number of synchronous reads. This value is derived by adding the values of the Pool Sync Data Reads and Pool Sync Index Reads attributes. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Sync Read Time The elapsed time used to perform all synchronous reads. This value is derived by subtracting the value of the Pool Async Read Time attribute from the value of the Pool Read Time attribute. Use this attribute to understand the I/O work being performed. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Sync Write The total number of synchronous index writes. The value is derived by adding the values of the Pool Sync Data Writes attribute and Pool Sync Index Writes attribute. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Sync Write Time The total elapsed time used to perform all synchronous writes. This value is derived by subtracting the value of the Pool Async Write Time attribute from the value of the Pool Write Time attribute. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Prefetch Wait Time The time an application spent waiting for an I/O server (prefetcher) to finish loading pages into the buffer pool. This attribute can be used to experiment with changing the number of I/O servers and the I/O server sizes. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Primary Log Used Percent The percentage of total log space used by the primary log. Use the returned value to help you evaluate the allocated amount of primary log space and refine the log buffer size, log file size, and primary log configuration parameters. The returned value is valid only if circular logging is used.

Primary Log Used Top The maximum bytes of primary logs used. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Restore Pending The RESTORE PENDING status in the database during the last monitoring interval.

Rollback Rate for Interval The rate, in rollbacks per second, at which unit-of-work rollbacks were attempted during the monitoring interval. Unit-of-work rollbacks include SQL ROLLBACK statements that are issued from applications and INTERNAL ROLLBACKS that the database manager initiates. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Secondary Log Used Percent The percentage of maximum log space used by the secondary log. Use the returned value to show the current dependency on secondary logs. Secondary logs are used when you have circular logging (log retention off) and the primary log files are full.

Select SQL Percent for Interval The percentage of total SQL statements that were SQL SELECT statements during the monitoring interval. Use the returned value to determine the level of application activity and throughput for the database.

Sequential Detect The current value of the sequential detection flag, which determines if the database manager must perform sequential detection. The database manager can monitor input and output operations. If sequential page reading is occurring, the database manager can activate input and output prefetching. This type of sequential prefetch is known as sequential detection. If this configuration parameter is set to `no`, prefetching takes place only if the database manager determines that it is useful (for example, in table sorts).

Snapshot Timestamp The date and time when the database system monitor information was collected. Use this attribute to help relate data chronologically if you are saving the results in a file or database for ongoing analysis.

Sort Heap The current value in units of 4-KB pages of the sort heap size. This value is the maximum amount of memory that can be allocated as sort heap for each sort within a database. The sort heap is the memory block where data is sorted. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Sort Overflows Percent for Interval The percentage of application sorts that overflowed during the monitoring interval. An overflow occurs when a sort has run out of space in the sort heap and requires disk space for temporary storage. If this percentage is high, you might want to adjust the database configuration by increasing the value of the SORTHEAP configuration parameter. The value format is an integer.

SQL Stmts Rate for Interval The rate, in issued SQL statements per second, at which SQL statements that run during the monitoring interval. The value format is an integer. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

System Tablespaces The number of SMS tablespaces in the database. Use the returned value to evaluate the use of SMS tablespaces and their effects on performance. Table data that is read from disk is available in the database buffer pool. Sometimes a data page is freed from the buffer pool before it is used. For SMS tablespaces, when the database manager requests that data page from the file system, the data page might still be in the cache of the file system. Having the page in the cache saves an input and output operation that would otherwise have been required. (For more information, see the DB2 administration documentation for the version of DB2 that you are using.) If you have many SMS tablespaces, you can increase the size of the file system cache to take advantage of this extra buffering. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Tables The number of tables in the database. Use this attribute to track database growth due to an increased number of tables over a period of time. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Tablespaces The number of tablespaces in the database. Use this attribute to track database growth over a period of time. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Tablespaces Long Data The number of tablespaces that store LONG data in the database. Use this attribute to track database growth over a period of time. LONG data can take up a large amount of space in a database. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Total Direct I/O Time The total time in milliseconds applied to direct reads and writes for the database. The returned value indicates the amount of time that the database performs direct reads and writes. A high returned value compared to normal operating levels can indicate the presence of an input and output conflict. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Total Log Used The total log space used in bytes. Values that are greater than or equal to 9223372036854775807 are indicated with the `Value Exceeds Maximum` text in the portal. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Total Pool Phys I/O The total time in milliseconds applied to physical I/O for the database. A high returned value (as compared to the total number of physical buffer pool input and output operations) can indicate the presence of an input and output wait, which in turn can indicate that you must move data to a different device. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Total Pool Phys Read The total time in milliseconds applied to processing read requests that caused data or index pages to be physically read from disk to the buffer pool for the database. The value is derived through this formula: `pool data p reads + pool index p reads`. The returned value is used to calculate the average pool read time. This average can indicate the presence of an input and output wait, which in turn can indicate that you must move data to a different device. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Total Pool Phys Write The total time in milliseconds for buffer pool physical writes (including asynchronous writes). The value is derived through this formula: `pool data writes + pool index writes`. The returned value is used to calculate the average pool write time. This average can indicate the presence of an input and output wait, which in turn can indicate that you must move data to a different device. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Total Sync I/O The total synchronous input and output. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Total Sync I/O Time The total time in milliseconds applied to processing requests for synchronous reads or writes for the database. The returned value is the sum of the returned values from the average pool write time (ms) and average pool read time (ms). This time is the amount of time that database agents spend doing synchronous reads and writes. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Triggers The number of triggers defined in the database. Use this attribute to track the use of triggers in the database. There are benefits to using triggers, including faster application development, easier maintenance, and global enforcement of business rules. For more information, see the DB2 administration documentation for the version of DB2 that you are using. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

UID SQL Percent for Interval The percentage of total SQL statements that were SQL UPDATE, INSERT, and DELETE statements during the monitoring interval. Use the returned value to determine the level of database data change activity.

User Indexes The number of indexes created by users in the database. Indexes created by SYSIBM are not counted. Use this to track the use of indexes in the database. The use of indexes can improve performance; for example, faster sorting of data. However, indexes can also have adverse effects on performance; for example, each INSERT or DELETE operation performed on a table requires additional updating of each index on that table. For a discussion of this topic, see the DB2 administration documentation for the version of DB2 that you are using. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Views The number of views in the database. Use this attribute to track the use of views in the database. Views can be created to limit access to sensitive data, while allowing more general access to other data. This provides flexibility in the way your programs and end-user queries can look at the table data. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Database01 (KUDDBASEGROUP01) attributes (Superseded)

Use these attributes to obtain information about the efficiency of the database and identify any problem areas for corrective action. All values are integers that are calculated from the first application connection, unless otherwise noted. This attribute group is superseded. There is a new attribute group with the same name that replaces it.

App Ctl Heap Size The maximum size (in 4-KB pages) for the application control heap in the database during the monitoring interval. The heap is required to share information among agents working on behalf of the same application at a node in a massively parallel processing (MPP) or a symmetric multiprocessor (SMP) system. If complex applications are being run or the MPP configuration has a large number of nodes, you must increase the size of this heap. In a partitioned database environment, this heap is used to store copies of the executing section of SQL statements for agents and subagents. However, symmetric multiprocessor agents (SMP), subagents, and agents in all other environments use the applHeapSz attribute.

Appl Heap Size The size (in 4-KB pages) of the application heap that is available for each individual agent in the database during the monitoring interval. Increase the value of the parameter if your application receives an error indicating that there is not enough storage in the application heap. The heap is allocated when an agent or subagent is initialized for an application. The amount allocated is the minimum amount needed to process the request given to the agent or subagent. When the agent or subagent requires more heap space to process larger SQL statements, the database manager allocates memory as needed, up to the maximum specified by the parameter.

Appls in DB2 The number of applications currently executing in the database.

Avg Appls The value of the average number of active applications.

Avg Direct Read Time The average time in milliseconds that is used to perform direct reads to the database. The value is derived through this formula: `direct read time / direct reads`

Avg Direct Write Time The average time in milliseconds for performing direct writes to the database. A high average time can indicate the existence of an input and output conflict. The value is derived through this formula: `direct write time / direct write`

Avg Lock Escal per Conn for Interval The average lock escalations per connection for this database during the monitoring interval. The value format is an integer. A lock is escalated when the total number of locks that an application holds reaches the maximum amount of lock list space available to the application, or the lock list space consumed by all applications is approaching the total lock list space. When an application reaches the maximum permitted number of locks and no additional locks can be escalated, the application uses space in the lock list that is allocated for other applications. When the entire lock list is full, an error occurs.

Avg Locks Held The average number of locks held by each currently connected application in the database. The value is derived through this formula:

```
locks held / appls cur cons
```

If the returned value is high compared to normal operating levels, it can indicate that one or more applications is using an excessive number of locks. Refine such applications to improve performance.

Avg Pages per Cleaner for Interval The average number of pages written per page cleaner that are invoked for the database during the monitoring interval. Use the returned value to determine how many pages are handled by the page cleaners of this database. If this value increases over time, you can define more page cleaners.

Avg Pool Async Data Reads The average number of buffer pool asynchronous data reads when compared to the total number of pool reads for the database. The value is derived through this formula: `pool async data reads / (pool data p reads + pool index p reads)`. Use the returned value to gain insight into how well the prefetchers are working and to refine the num_ioservers configuration parameter. If the returned value is low compared to normal operating levels, there might not be enough input and output servers to prefetch data into the buffer, causing the database manager agents to spend extra time on physical reads. Increase the number of input and output servers by increasing the value of the num_ioservers configuration parameter. If too many servers are allocated, system performance is not reduced because the extra input and output servers are not used.

Avg Pool Async Data Writes The average number of buffer pool asynchronous data writes (data and index) when compared to the total number of pool writes for the database. The value is derived through this formula: `pool async data writes / (pool data writes + pool index writes)`. Use the returned value to gain insight into how well the page cleaners are working and to refine the num_iocleaners configuration parameter. If the returned value is low compared to normal operating levels, increase the number of input and output cleaners by increasing the value of the num_iocleaners parameter. If the returned value is high compared to normal operating levels, you can save system resources by decreasing the number of input and output cleaners (by decreasing the value of the num_iocleaners parameter).

Avg Pool IO Time The average time (in milliseconds) for performing buffer pool input and output operations (reading or writing) to the database. A high average time can indicate the existence of an input and output conflict. In this case, you might need to move data to a different device. The returned value includes the time applied to asynchronous input and output operations (which are performed by prefetchers and page cleaners).

Avg Pool Writes per Read The ratio of total pool writes to pool reads for the database. The value is derived through this formula: `(pool data writes + pool` `index writes) / (pool data p reads + pool index p reads)`. If the returned value is greater than 1, you can improve performance by increasing the available buffer pool space. A returned value greater than 1 indicates that at least one write to disk had to occur (either to free a page in the buffer pool, or to flush the buffer pool) before a page can be read into the buffer pool. You can increase the available buffer pool space by freeing the space more often or by increasing the total space for the buffer pool.

Avg Sect Read per Direct Read The average number of sectors that are read by a direct read for the database. The value is derived through this formula: `direct reads / direct read reqs`. Direct reads do not use the buffer pool, and so result in poor performance because the data is physically read from disk each time. If you are using system monitors to track input and output for the device, this value helps you distinguish database input and output from non-database input and output.

Avg Sect Written per Direct Write The average number of sectors that are written by a direct write to the database. The value is derived through this formula: `direct writes / direct write reqs`. Direct writes do not use the buffer pool, which results in poor performance because the data is physically written from disk each time. If you are using system monitors to track input and output for the device, this value helps you distinguish database input and output from non-database input and output.

Avg Sync IO Time The average time (in milliseconds) to perform synchronous input and output operations for the database. Use the returned value to analyze the input and output work being performed for the database. Synchronous input and output operations for a database are performed by database manager agents. Asynchronous input and output operations are performed by prefetchers (reads) and page cleaners (writes). In general, asynchronous input and output helps your applications run faster.

Buff Page Use the returned value (in units of 4-KB pages) to analyze the input and output work being performed for the database. Synchronous input and output operations for a database are performed by database manager agents. Asynchronous input and output operations are performed by prefetchers (reads) and page cleaners (writes). In general, asynchronous input and output helps your applications run faster. In the currently supported releases of DB2, multiple buffer pools might be defined in a single database. For instance, buffer pools can be defined and associated with a particular tablespace. Each buffer pool created can be given its own individual size. The buffpage attribute serves only as a default value for buffer pools created within a particular database. Therefore, the value of the buffpage attribute is much less critical to performance in current releases of DB2, because most buffer pools are given an individual size when created. The buffpage attribute must not be used to evaluate or tune the performance of DB2 unless it is used as the default value when creating buffer pools in a database.

Catalog Cache Size The value in units of 4-KB pages of the catalog cache size. This value is the maximum amount of space that the catalog cache can use from the database heap (dbheap). The catalog cache is referenced whenever a table, view, or alias name is processed during the compilation of an SQL statement. It is dynamically allocated from dbheap, as required, until the catalog cache size is reached.

Changed Pages Thresh The value in percentage units of the changed pages threshold. This value sets a limit on how much buffer pool space can be occupied by changed pages before the asynchronous page cleaners are started, if they are not currently active. Asynchronous page cleaners write changed pages from the buffer pool to disk before the space in the buffer pool is required by a database agent. This means that the agents do not need to wait for a changed page to be written out before being able to read a page, and application transactions run faster.

Commit Stmts per Sec The total number of commits initiated internally by the database per second. Use the returned value to determine rates of database activity.

Cur Cons Pct The percentage of applications currently connected.

Days Since Last Backup The numbers of days since the last database backup was completed. The value format is an integer. The value for no backup completed is 2147483647.

DB Cap Err The number of errors encountered by the Capture program within the last five minutes. Use the returned value to determine whether the Capture program encountered an error that prevented it from running. If any errors are detected, the Capture program came down at the time the error occurred. The Capture program might or might not still be down. The Capture program is the most critical replication component in the replication system. If the Capture program is not active, there are no new change records to apply to the target systems. If your data concurrency requirements are high and you want to ensure that the Capture program runs continuously, use this monitor to determine when the Capture program encounters an error that prevents it from running.

Values for this attribute are only available in a SQL replication environment.

DB Cap Lag The time difference in minutes between the current timestamp and the last timestamp recorded by the Capture program. This time difference is the Capture lag. Use the returned value to determine whether the Capture program is keeping up with the DB2 database log. The Capture program uses an interface to the DB2 database log or journal to detect and save changes to the data in the tables registered for replication.

Values for this attribute are only available in a SQL replication environment.

DB Cap Prun The number of rows in the unit-of-work table. Use the returned value to help you determine whether you need to prune the unit -of-work (UOW) table or the change data (CD) table.

Values for this attribute are only available in a SQL replication environment.

DB Connection Timestamp The date and time when the first database connection was made. The following special values cause the Tivoli Enterprise Portal to display fixed string messages:

Internal value	Tivoli Enterprise Portal display
0000000000000001	N/A (Not Applicable or Not Available)
0000000000000002	N/C (Not Calculated)
0000000000000003	N/P (Not Present)

DB Heap The value in units of 4-KB pages of the database heap. This value is the maximum amount of memory allowed for a database heap. There is one database heap for each database. It is used on behalf of all applications connected to the database. Refining dbheap has minimal impact on performance. The main function of this parameter is to prevent the database manager from allocating an excessive amount of space for a particular database.

DB Name The real name of the database for which information is collected or to which the application is connected. This name was given to the database when it was created. The value format is a simple text string with a maximum of 32 characters.

DB Name (Unicode) The real name of the database for which information is collected or to which the application is connected in Unicode. This name was given to the database when it was created. The value format is a simple text string with a maximum of 60 bytes.

DB Partition The DB2 database partition node number, which can range from 0 to 999. The Aggregated and Current Partition values can be used within a query or situation filter. If a db partition filter is not specified, data is returned for the current database partition. If a db partition filter is set to Aggregated, only aggregated partition data is returned. Historical data collection includes both aggregated and individual partition attribute data. In addition to numeric partition numbers in the 0 to 999 range, the following values are also valid:

External value	Internal value
Aggregated	-1
Current Partition	-2
All	-3

DB Tablespaces The number of Database Managed Space tablespaces in the database. Use this attribute to track database growth over a period of time.

DDL SQL Pct for Interval The percentage of total SQL statements that were SQL DDL statements during the monitoring interval. Due to the high activity in the system catalog tables, try to keep DDL statement activity to a minimum. If the returned value is high compared to normal operating levels, determine the activity causing it to be high and restrict it from being performed. Examples of DDL statements are CREATE TABLE, CREATE VIEW, ALTER TABLE, and DROP INDEX. You can also use the returned value to refine the package cache hit ratio for this application. DDL statements can also affect the package cache by invalidating sections that are stored there and causing additional system overhead due to section recompilation.

Deadlock Rollbacks Pct The percentage of the total number of rollbacks that deadlock caused. The value format is an integer.

Deadlocks for Interval The number of deadlocks detected in the database during the monitoring interval. Use the returned value to determine whether applications are experiencing conflict problems in the database. You can resolve the problem by determining in which applications the deadlocks are occurring. You can then try to modify the applications to better enable them to run concurrently.

Estore RW Ratio for Interval The ratio as a percentage of data and index pages copied from extended storage to pages copied to extended storage during the monitoring interval. When a page is transferred from extended storage to the buffer pool, you save a system input and output call. However, you still incur the cost of attaching to the extended memory segment, copying the page, and detaching from the segment. Use the returned value to determine if you would benefit from using extended storage. The higher the ratio, the more likely you are to benefit. In general, extended storage is particularly useful if input and output activity is very high on your system.

Event Monitors The number of event monitors defined in the database. Use the returned value to determine how many event monitors are defined for the database. When you define an event monitor, its definition is stored in the database system catalog table. You can create any number of event monitors. However, the maximum number of event monitors that can be active for a database at any given time is 32.

Failed SQL Stmts Pct for Interval The percentage of total Structured Query Language statements that failed during the monitoring interval. The value format is an integer.

Instance Name (Unicode) The name of the monitored DB2 instance in Unicode. The valid format is a text string with a maximum of 60 bytes.

Int Auto Rebinds The number of automatic rebinds or recompiles that were attempted in the database. Use the returned value to determine the level of database activity. Automatic rebinds are the internal binds that the system performs when a package is invalidated. They can have a significant impact on performance and must be minimized where possible.

Int Commits The total number of commits initiated internally by the database. Use the returned value to gain insight into internal activity within the database. The returned value is also used in calculating "Commit statements per second."

Int Deadlock Rollbacks Pct The percentage of the total number of internal rollbacks due to deadlocks. Use the returned value to distinguish those rollbacks caused by internal deadlocks from rollbacks caused by other situations (for example, incomplete imports). The returned value is the percentage of internal rollbacks due to internal deadlocks since the first database connection or the last reset of the database monitor counters.

Int Deadlock Rollbacks Pct for Interval The percentage of rollbacks that were due to deadlock during the monitoring interval. Use the returned value to distinguish those rollbacks caused by internal deadlocks from rollbacks caused by other situations (for example, incomplete imports). The returned value is the percentage of internal rollbacks due to internal deadlocks since the first database connection or the last reset of the database monitor counters.

Int Rows Deleted The number of rows deleted from the database as a result of internal activity. Use the returned value to gain insight into internal activity within the database. If this activity is high compared to normal operating levels, you can evaluate your table design to determine if the referential constraints or triggers that you defined on your database are necessary.

Int Rows Inserted The number of rows inserted into a database as a result of internal activity caused by triggers. Use the returned value to gain insight into internal activity within the database. If this activity is high compared to normal operating levels, you can evaluate your design to determine if you can alter it to reduce this activity.

Int Rows Updated The number of rows updated in the database as a result of internal activity. Use the returned value to gain insight into internal activity within the database. If this activity is high compared to normal operating levels, you can evaluate your table design to determine if the referential constraints that you defined are necessary.

Invalid Pkgs The number of all packages that are currently marked not valid in the database. Use the returned value as an indication of the current number of packages that are not valid. A package is marked not valid if it depends on an object (for example, a table) and that object is dropped. The number of packages that are not valid can indicate how many automatic rebinds are necessary in the database. Such packages automatically rebound the next time they are accessed, unless a trigger was dropped or the dropped object was not recreated. Use of automatic rebinds can significantly lower performance, and must be minimized if possible.

Invalid Sys Pkgs The number of system packages that are currently marked not valid in the database. Use the returned value as an indication of the current number of nonvalid packages owned by the system. A package is marked not valid if it depends on an object (for example, a table) and that object is dropped. The number of packages that are not valid can indicate how many automatic rebinds are necessary in the database. The package is automatically rebound the next time it is accessed, unless it was marked not valid because a trigger was dropped or because the dropped object was not recreated. Use of automatic rebinds can significantly lower performance, and must be minimized where possible.

Invalid Triggers The number of triggers that are marked not valid in the database. Use the returned value to determine the number of triggers that must be re-validated. A trigger is marked not valid if an object on which the trigger depends is dropped. To re-validate such a trigger, retrieve its definition from the database system catalog and submit a new CREATE TRIGGER statement.

Last Backup Timestamp The date and time that the latest database backup was completed. The following special values cause the Tivoli Enterprise Portal to display fixed string messages:

Internal value	Tivoli Enterprise Portal display
0000000000000001	N/A (Not Applicable or Not Available)
0000000000000002	N/C (Not Calculated)
0000000000000003	N/P (Not Present)

UA103

Lock Escalation for Interval The total number of lock escalations for applications connected to this database during the monitoring interval. Exclusive lock escalations are included in this number. Use the returned value to help you evaluate the settings of the LOCKLIST and MAXLOCKS configuration parameters. Lock escalations can result in a decrease in concurrency among the applications connected to a database.

Lock List The value in units of 4-KB pages of the maximum storage for lock lists. This value is the amount of storage that is allocated to the lock list. There is one lock list for each database, and it contains the locks held by all applications concurrently connected to the database. Too small a value can lead to excessive lock waits. Too high a value compared to normal operating levels can deprive the system of resources or memory.

Lock List in Use (KB) The total amount of lock list memory in KB that is currently in use. The value format is an integer. This attribute can be used with the locklist configuration parameter to calculate the lock list utilization. If the lock list utilization is high, consider increasing the size of that parameter. Values that are greater than or equal to 2147483647 are indicated in the portal with the `Value Exceeds Maximum` text, and values that are smaller than -2147483648 are indicated with the `Value Exceeds Minimum` text. The following values are valid:

External value	Internal value
Value Exceeds Maximum	2147483647
Value Exceeds Minimum	-2147483648

Lock List in Use Pct The percentage of space used in the locklist of this database. Use the returned value to determine how much of the locklist space is free for new locks to be requested.

Lock Timeouts for Interval The number of times that a request to lock an object were timed out instead of being granted during the monitoring interval. The value format is an integer

Lock Waits for Interval The number of times that applications had to wait for locks in the database during the monitoring interval. Use the returned value as an indication of how much time is applied to waiting for locks during a particular monitoring interval.

Lock Waits Pct The percentage of currently connected applications that are waiting for a lock in the database. The value is derived through this formula:

```
100 * locks waiting / appls cur cons
```

If the returned value is high compared to normal operating levels, the applications can have concurrency problems. You must identify applications that are holding locks or exclusive locks for long periods of time and determine whether they can release their locks more often.

Log Buff Size This value specifies the amount of the database heap to use as a buffer for log records before writing these records to disk. It is important that the log buffer can hold the amount of log space used by an average transaction. Otherwise, logging performance decreases and slows the overall system.

Log IO for Interval The total amount of log input and output. This amount is the sum of the number of log pages read and the number of log pages written within the monitoring interval. Use the returned value to determine whether you must move the log to a different device. If this input and output is beyond the capabilities of the current device, you can determine if moving the log (by changing the newlogpath configuration parameter) improves performance.

Log Primary The number of primary log files.

Max Appls The value of the maximum number of active applications. This value is the maximum number of concurrent applications that can be connected (both local and remote) to a database. Because each application that attaches to a database causes some private memory to be allocated, allowing a large number of concurrent applications potentially uses more memory. Increasing the value of this parameter without lowering the maxlocks parameter or increasing the locklist parameter can cause you to reach the database limit on locks (locklist parameter) rather than the application limit. The result can be pervasive lock escalation problems.

Max Locks The value of the maximum percentage of lock list before escalation. This value specifies the percentage of the lock list that an application can hold before the database manager performs lock escalation. Lock escalation can increase contention, which reduces system throughput and increases user response time. The values for the maxlocks and maxappls parameters must satisfy (MAXLOCKS x MAXAPPLS) >100, and each lock uses 32 bytes. Rebind application packages after changing this parameter.

Min Commit The value of the number of commits to group. By using this parameter you can delay the writing of log records to disk until a minimum number of commits have been performed. This delay can help reduce the overhead associated with writing log records and can improve performance. The default value for mincommit is 1, which can be too low for your environment. By sampling the number of transactions per-second throughout the day, you can determine the peak per second rate and adjust mincommit to accommodate all or most transactions. This adjustment minimizes the number of log writes under the heaviest conditions. As you increase mincommit, you might also need to increase the log buffer size (LOGBUFSZ parameter) to avoid filling the log buffer. Filling the log buffer also forces the writing of log records to disk. If you change mincommit, you must change the value for the logbufsz configuration parameter.

New Log Path The current value of the newlogpath configuration parameter. A valid value is a text string up to 256 characters in length. You use the newlogpath configuration parameter to specify a new location for the log files. The specified path does not become the current log path until both of the following conditions are met:

- The database is in a consistent state.
- All users are disconnected from the database.

When the first new connection is made to the database, the database manager moves the logs to this location.

New Log Path (Unicode) The current value of the newlogpath configuration parameter in Unicode. A valid value is a text string up to 768 characters in length.

Node Name For new installations of version 6, release 2, the format is instanceid:hostname:UD for all operating systems. The format for version 6, release 1 of the DB2 agent on Windows systems is instanceid:hostname:UD; on UNIX and Linux systems, the format is instanceid:hostname.

Num IO Cleaners The current value of the number of asynchronous page cleaners. This parameter specifies the number of asynchronous page cleaners for a database. Page cleaners monitor the buffer pool and asynchronously write out changed pages to disk to free space in the buffer pool.

Num IO Servers The current value of the number of input and output servers. This value specifies the number of input and output servers for a database. Input and output servers are used on behalf of the database agents to perform asynchronous input and output operations for utilities such as backup and restore, and to perform prefetch input and output (in which case, they are called prefetchers) operations. Prefetchers read pages from disk into the buffer pool in anticipation of their use. In most situations, these pages are read just before they are needed. However, prefetchers can cause unnecessary input and output operations by reading pages into the buffer pool that might not be used. For example, an application starts reading through a table, and prefetchers read consecutive pages into the buffer pool before the pages are required by the application. Then the application fills the application buffer and stops reading. Meanwhile, the prefetchers already have performed the input and output operations for additional pages and the buffer pool is partially taken up with those pages. To exploit all the input and output devices in the system, a good value for num_ioservers to use is one or two more than the number of physical devices on which the database is established.

Page Cleans for Interval The number of times a page cleaner was invoked for the database (for any reason) during the monitoring interval. Use the returned value to determine how often pages are written to disk by the page cleaners of this database. If this value increases over time, you can define more page cleaners. The number of page cleaners is determined by the number of I/O cleaners configured.

Pages per Prefetch for Interval The number of data pages read per prefetch request for the database during the monitoring interval. Use the returned value to determine the amount of asynchronous input and output in each interaction with the prefetcher. An excessively low returned value when compared to normal operating levels indicates that you need more input and output servers. The more input and output servers that you have, the better your query performance.

Pkg Cache Size The current value in units of 4-KB pages of the package cache size. This value controls the amount of application heap memory to be used for caching static and dynamic SQL statements of a package. You must experiment with the size of the package cache to find the optimal number for this attribute. For example, you can use a smaller package cache size if there is no increase in the number of package cache inserts when you decrease the size of the cache. Decreasing the package cache size frees up system resources for other work. However, increasing the package cache size can improve overall system performance if it results in a decrease of package cache inserts.

Pool Hit Ratio Index Pct for Interval The database index page hit ratio (as a percentage) for the buffer pool during the monitoring interval. The index page hit ratio for the buffer pool indicates the percentage of index page requests for which the database manager did not need to load an index page from disk to service. That is, the index page was already in the buffer pool. The higher the returned value, the lower the frequency of disk input and output, and the faster the performance. If the hit ratio is low compared to normal operating levels, increasing the number of buffer pool pages can improve performance.

Pool Hit Ratio Pct for Interval The overall buffer pool hit ratio as a percentage for the database during the monitoring interval. This hit ratio includes both index and data page activity. The overall buffer pool hit ratio indicates the percentage of page requests for which the database manager did not need to load a page from disk to service. (That is, the page was already in the buffer pool.) The greater the buffer pool hit ratio, the lower the frequency of disk input and output. If the hit ratio is low compared to normal operating levels, increasing the number of buffer pool pages can improve performance. A ratio of zero indicates that pages needed to be read for every request. For a large database, increasing the buffer pool size can have a minimal effect on the buffer pool hit ratio. Such a database can have so large a number of data pages that the statistical chance of a hit is not increased by an increase of the buffer pools. However, even though the data might be too large to fit in the buffer pool, the entire index can fit. In this case, you can refine buffer pool sizes until the overall buffer pool hit ratio stops increasing, and then refine the buffer pool until the buffer pool index hit ratio no longer increases.

Pool IO per Sec The rate (per second) of buffer pool input and output operations for the database. Buffer pool input and output includes all physical data and index pages that go through the buffer pool when read or written. Use the returned value to determine how efficient your data storage device is. A low value indicates the presence of an input and output wait, in which case you must move data to a different device.

Pool Sync Index Reads The number of pool index physical reads minus the pool async index reads.

Pool Total Reads (K) The total number of read requests in thousands that required input output operations to get data pages and index pages into the buffer pool. The value format is an integer. This attribute is the total of the Pool Data Physical Reads and Pool Index Physical Reads attributes. Values that are greater than or equal to 2147483647 are indicated in the portal with the`Value Exceeds Maximum` text , and values that are smaller than -2147483648 are indicated with the `Value Exceeds Minimum` text. The following values are valid:

External value	Internal value
Value Exceeds Maximum	2147483647
Value Exceeds Minimum	-2147483648

Pool Total Writes (K) The total number of write requests in thousands. The value format is an integer. This attribute is the total of the Pool Data Writes and Pool Index Writes attributes. Values that are greater than or equal to 2147483647 are indicated in the portal with the `Value Exceeds Maximum` text , and values that are smaller than -2147483648 are indicated with the `Value Exceeds Minimum` text.

Pri Log Used Pct The percentage of total log space used by the primary log. Use the returned value to help you evaluate the allocated amount of primary log space and refine the log buffer size, log file size, and primary log configuration parameters. The returned value is valid only if circular logging is used.

Pri Log Used Top The maximum bytes of primary logs used. The following values are valid:

External value	Internal value
Value Exceeds Maximum	2147483647
Value Exceeds Minimum	-2147483648

Pri Log Used Top (MB) The maximum bytes of primary logs used in MB. The following values are valid:

External value	Internal value
Value Exceeds Maximum	2147483647
Value Exceeds Minimum	-2147483648

Restore Pending The RESTORE PENDING status in the database during the last monitoring interval.

Rollback Rate for Interval The rate, in rollbacks per second, at which unit-of-work rollbacks were attempted during the monitoring interval. Unit-of-work rollbacks include SQL ROLLBACK statements that are issued from applications and INTERNAL ROLLBACKS that the database manager initiates.

Sec Log Used Pct The percentage of maximum log space used by the secondary log. Use the returned value to show the current dependency on secondary logs. Secondary logs are used when you have circular logging (log retention off) and the primary log files are full.

Sec Log Used Top (MB) The maximum amount of secondary log space in MB that has been used. The value format is an integer. Values that are greater than or equal to 2147483647 are indicated in the portal with the `Value Exceeds Maximum` text , and values that are smaller than -2147483648 are indicated with the `Value Exceeds Minimum` text. The following values are valid:

External value	Internal value
Value Exceeds Maximum	2147483647
Value Exceeds Minimum	-2147483648

Select SQL Pct for Interval The percentage of total SQL statements that were SQL SELECT statements during the monitoring interval. Use the returned value to determine the level of application activity and throughput for the database.

Sequential Detect The current value of the sequential detection flag, which determines if the database manager must perform sequential detection. The database manager can monitor input and output operations. If sequential page reading is occurring, the database manager can activate input and output prefetching. This type of sequential prefetch is known as sequential detection. If this configuration parameter is set to `no`, prefetching takes place only if the database manager determines that it is useful (for example, in table sorts).

Snapshot Time The string date and time when the database system monitor information was collected. Use this attribute to help relate data chronologically if you are saving the results in a file or database for ongoing analysis. The Tivoli Enterprise Portal formats the timestamp value into a date and time string. The internal timestamp value stored in the database is in the format cYYMMDDhhmmss000, which is described in the following table:

Value	Description
C	Century (0 for 20th, 1 for 21st)
YY	Year (last two digits of the year, for example (00 - 99)
MM	Month (01 for January, 02 for February, and so on)
DD	Day (the day of the month, for example 01 - 31)

Value	Description
HH	Hour (the hour of the day in 24-hour format from 00 - 23)
MM	Minute (the minutes of the hour from 00 - 59)
SS	Second (the seconds of the hour (00 - 59)

Snapshot Timestamp The string date and time when the database system monitor information was collected. Use this attribute to help relate data chronologically if you are saving the results in a file or database for ongoing analysis. The Tivoli Enterprise Portal formats the timestamp value into a date and time string. The internal timestamp value stored in the database is in the format cYYMMDDhhmmss000, which is described in the following table:

Value	Description
C	Century (0 for 20th, 1 for 21st)
YY	Year (last two digits of the year, for example (00 - 99)
MM	Month (01 for January, 02 for February, and so on)
DD	Day (the day of the month, for example 01 - 31)
HH	Hour (the hour of the day in 24-hour format from 00 - 23)
MM	Minute (the minutes of the hour from 00 - 59)
SS	Second (the seconds of the hour (00 - 59)

The following special values cause the Tivoli Enterprise Portal to display fixed string messages:

Internal value	Tivoli Enterprise Portal display
0000000000000001	N/A (Not Applicable or Not Available)
0000000000000002	N/C (Not Calculated)
0000000000000003	N/P (Not Present)

Sort Heap The current value in units of 4-KB pages of the sort heap size. This value is the maximum amount of memory that can be allocated as sort heap for each sort within a database. The sort heap is the memory block where data is sorted.

Sort Overflows Pct for Interval The percentage of application sorts that overflowed during the monitoring interval. An overflow occurs when a sort has run out of space in the sort heap and requires disk space for temporary storage. If this percentage is high, you might want to adjust the database configuration by increasing the value of the SORTHEAP configuration parameter. The value format is an integer.

SQL Stmts Rate for Interval The rate, in issued SQL statements per second, at which SQL statements that run during the monitoring interval. The value format is an integer.

System Tablespaces The number of SMS tablespaces in the database. Use the returned value to evaluate the use of SMS tablespaces and their effects on performance. Table data that is read from disk is available in the database buffer pool. Sometimes a data page is freed from the buffer pool before it is used. For SMS tablespaces, when the database manager requests that data page from the file system, the data page might still be in the cache of the file system. Having the page in the cache saves an input and output operation that would otherwise have been required. (For more information, see the DB2 administration documentation for the version of DB2 that you are using.) If you have many SMS tablespaces, you can increase the size of the file system cache to take advantage of this extra buffering.

Tables The number of tables in the database. Use this attribute to track database growth due to an increased number of tables over a period of time.

Tablespaces The number of tablespaces in the database. Use this attribute to track database growth over a period of time.

Tablespaces Long Data The number of tablespaces that store LONG data in the database. Use this attribute to track database growth over a period of time. LONG data can take up a large amount of space in a database.

Total Direct IO Time The total time in milliseconds applied to direct reads and writes for the database. The returned value indicates the amount of time that the database performs direct reads and writes. A high returned value compared to normal operating levels can indicate the presence of an input and output conflict.

Total Log Used The total log space used in bytes. Values that are greater than or equal to 2147483647 are indicated in the portal with the `Value Exceeds Maximum` text , and values that are smaller than -2147483648 are indicated with the `Value Exceeds Minimum` text. The following values are valid:

External value	Internal value
Value Exceeds Maximum	2147483647
Value Exceeds Minimum	-2147483648

Total Log Used (bytes) The total log space used in bytes in the database. The value format is an integer. Values that are greater than or equal to 2147483647 are indicated in the portal with the `Value Exceeds Maximum` text , and values that are smaller than -2147483648 are indicated with the `Value Exceeds Minimum` text. The following values are valid:

External value	Internal value
Value Exceeds Maximum	2147483647
Value Exceeds Minimum	-2147483648

Total log Used Top (bytes) The maximum amount of total log space (in bytes) that has been used. The value format is an integer. Use this attribute to evaluate the amount of primary log space that is allocated. Comparing the value of this attribute with the amount of primary log space that is allocated can help you evaluate the configuration parameter settings. Values that are greater than or equal to 2147483647 are indicated in the portal with the `Value Exceeds Maximum` text , and values that are smaller than -2147483648 are indicated with the `Value Exceeds Minimum` text. The following values are valid:

External value	Internal value
Value Exceeds Maximum	2147483647

External value	Internal value
Value Exceeds Minimum	-2147483648

Total Pool Phys IO The total time in milliseconds applied to physical I/O for the database. A high returned value (as compared to the total number of physical buffer pool input and output operations) can indicate the presence of an input and output wait, which in turn can indicate that you must move data to a different device.

Total Pool Phys Read The total time in milliseconds applied to processing read requests that caused data or index pages to be physically read from disk to the buffer pool for the database. The value is derived through this formula: `pool data p reads + pool index p reads`. The returned value is used to calculate the average pool read time. This average can indicate the presence of an input and output wait, which in turn can indicate that you must move data to a different device.

Total Pool Phys Write The total time in milliseconds for buffer pool physical writes (including asynchronous writes). The value is derived through this formula: `pool data writes + pool index writes`. The returned value is used to calculate the average pool write time. This average can indicate the presence of an input and output wait, which in turn can indicate that you must move data to a different device.

Total Sync IO The total synchronous input and output.

Total Sync IO Time The total time in milliseconds applied to processing requests for synchronous reads or writes for the database. The returned value is the sum of the returned values from the average pool write time (ms) and average pool read time (ms). This time is the amount of time that database agents spend doing synchronous reads and writes.

Triggers The number of triggers defined in the database. Use this attribute to track the use of triggers in the database. There are benefits to using triggers, including faster application development, easier maintenance, and global enforcement of business rules. For more information, see the DB2 administration documentation for the version of DB2 that you are using.

UID SQL Pct for Interval The percentage of total SQL statements that were SQL UPDATE, INSERT, and DELETE statements during the monitoring interval. Use the returned value to determine the level of database data change activity.

User Indexes The number of indexes created by users in the database. Indexes created by SYSIBM are not counted. Use this to track the use of indexes in the database. The use of indexes can improve performance; for example, faster sorting of data. However, indexes can also have adverse effects on performance; for example, each INSERT or DELETE operation performed on a table requires additional updating of each index on that table. For a discussion of this topic, see the DB2 administration documentation for the version of DB2 that you are using.

Views The number of views in the database. Use this attribute to track the use of views in the database. Views can be created to limit access to sensitive data, while allowing more general access to other data. This provides flexibility in the way your programs and end-user queries can look at the table data.

Database02 (KUD_DB2_Database02) attributes

Use these attributes to obtain information about database activities.

Active Hash Joins The total number of hash joins that are currently running and consuming memory.

Active OLAP Funcs The total number of OLAP functions that are currently running and consuming sort heap memory.

Appl ID Oldest Xact The application ID of the application that has the oldest transaction. Use this attribute to which application has the oldest active transaction. This application can be forced to free up log space. If the application take a large amount of log space, examine the application to determine if it can be modified to commit more frequently.

Async Runstats The total number of successful asynchronous RUNSTATS activities that are performed by real-time statistics gathering for all the applications in the database. Values reported by all the database partitions are aggregated together.

Blocks Pending Cleanup The total number of MDC table blocks in the database that are pending asynchronous cleanup following a roll out delete.

Cat Cache Size Top The largest size that is reached by the catalog cache. This attribute indicates the maximum number of bytes the catalog cache required for the workload run against the database since it is activated. If the catalog cache overflows, the value is the largest size reached by the catalog cache during the overflow.

Catalog Partition The number of the catalog node.

Catalog Partition Name The network name of the catalog node.

Data Temp Pool Hit Ratio The data page hit ratio for buffer pools that are located in temporary tablespaces.

DB Name The real name of the host database for which information is collected or to which the application is connected. This name was given to the database when it was created. The value format is a simple text string with a maximum of 60 bytes.

DB Partition The DB2 database partition node number, which can range from 0 to 999. The Aggregated and Current Partition values can be used within a query or situation filter. If a db partition filter is not specified, data is returned for the current database partition. If a db partition filter is set to Aggregated, only aggregated partition data is returned. Historical data collection includes both aggregated and individual partition attribute data. In addition to numeric partition numbers in the 0 to 999 range, the following values are also valid:

External value	Internal value
Aggregated	-1
Current Partition	-2
All	-3

Elapsed Exec Time MS At the DCS statement level, this is the elapsed time (in ms) spent processing an SQL request on a host database server.

Elapsed Exec Time S At the DCS statement level, this is the elapsed time (in seconds) spent processing an SQL request on a host database server.

Instance Name The name of the monitored DB2 instance. The valid format is a text string with a maximum of 60 bytes.

Last Reset Indicates the most recent date and time when the monitor counters are reset for the application issuing the GET SNAPSHOT command. Use this attribute to determine the scope of information returned by the database system monitor.

Log Held by Dirty Pages The amount of log (in bytes) corresponding to the difference between the oldest dirty page in the database and the top of the active log. When the snapshot is taken, this value is calculated based on conditions at the time of that snapshot. Use this element to evaluate the effectiveness of page cleaning for older pages in the buffer pool. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Log Read Time NS The total elapsed time (in ns) that the logger spends reading log data from the disk. Use this attribute with the log reads, num log read io, and num log data found in buffer attributes to determine the following items:
- Whether the current disk is adequate for logging.
- Whether the log buffer size is adequate.

The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Log Read Time S The total elapsed time (in seconds) that the logger spends reading log data from the disk. Use this attribute with the log reads, num log read io, and num log data found in buffer attributes to determine the following items:
- Whether the current disk is adequate for logging.
- Whether the log buffer size is adequate.

The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Log to Redo for Recovery The amount of log (in bytes) that has to be redone for crash recovery.

Log Write Time NS The total elapsed time (in ns) that the logger spends writing log data to the disk. Use this attribute with the log writes and num log write io attributes to determine whether the current disk is adequate for logging. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Log Write Time S The total elapsed time (in seconds) that the logger spends writing log data to the disk. Use this attribute with the log writes and num log write io attributes to determine whether the current disk is adequate for logging. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Min Catalog Cache Size The minimum size of the catalog cache that is required by your workload.

Min Pkg Cache Size The minimum size of the package cache that is required by your workload.

Node Name For new installations of version 6, release 2, the format is instanceid:hostname:UD for all operating systems. The format for version 6, release 1 of the DB2 agent Windows systems is instanceid:hostname:UD; on UNIX and Linux systems, the format is instanceid:hostname.

Num DB Storage Paths The number of automatic storage paths that are associated with this database.

Num Indoubt Trans The number of outstanding indoubt transactions in the database. Indoubt transactions hold log space for uncommitted transactions, which can cause the logs to become full. When the logs are full, further transactions can not be completed. The resolution of this problem involves a manual process of heuristically resolving the indoubt transactions. This attributes provides a count of the number of currently outstanding indoubt transactions that must be heuristically resolved.

Num Log Buffer Full The number of times that agents have to wait for log data to write to disk while copying log records into the log buffer. This value is increased per agent per incident. For example, if two agents attempt to copy log data while the buffer is full, this value is increased by two. Use this attribute to determine if the LOGBUFSZ database configuration parameter needs to be increased. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Num Log Data Found in Buffer The number of times that an agent reads log data from the buffer. Reading log data from the buffer is preferable to reading from the disk because the latter is slower. Use this attribute with the num log read io attribute to determine if the LOGBUFSZ database configuration parameter needs to be increased. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Num Log Part Page IO The number of I/O requests that are issued by the logger for writing partial log data to the disk. Use this attribute with the log writes, log write time, and num log write io attributes to determine if the current disk is adequate for logging. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Num Log Read IO The number of I/O requests that are issued by the logger for reading log data from the disk. Use this attribute with the log reads and log read time attributes to determine if the current disk is adequate for logging. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Num Log Write IO The number of I/O requests that are issued by the logger for writing log data to the disk. Use this attribute with the log writes and log write time attributes to determine if the current disk is adequate for logging. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Num Threshold Violations The number of threshold violations that have taken place in this database since the database was last activated. Use this attribute to determine whether thresholds are effective for this particular application or whether the threshold violations are excessive.

OLAP Func Overflows The number of times that OLAP function data exceeded the available sort heap space. At the database level, use this attribute in conjunction with the total olapfuncs attribute to calculate the percentage of OLAP functions that overflowed to disk. If this percentage is high and the performance of applications using OLAP functions needs to be improved, consider increasing the sort heap size. At the application level, use this attribute to evaluate OLAP function performance for individual applications.

Pkg Cache Num Overflows The number of times that the package cache overflowed the bounds of its allocated memory.

Pkg Cache Size Top The largest size that is reached by the package cache. If the package cache overflowed, this attribute value is the largest size that is reached by the package cache during the overflow. Check the pkg cache num overflows attribute to determine if such a condition occurred.

Pool No Victim Buffer The number of times an agent does not have a preselected victim buffer that is available. Use this attribute to help evaluate whether you have enough page cleaners for a given buffer pool when using proactive page cleaning.

Pool Temp Data L Reads The number of data pages that have been requested from the buffer pool (logical) for temporary tablespaces. In conjunction with the pool temp data p reads attribute, the data page hit ratio for buffer pools located in temporary tablespaces can be calculated using the following formula:

```
1 - (pool temp data p reads / pool temp data l reads)
```

Pool Temp Data P Reads The number of data pages read in from the tablespace containers (physical) for temporary tablespaces.

Pool Temp Hit Ratio The data page and index page hit ratio for buffer pools that are located in temporary tablespaces.

Pool Temp Index L Reads The number of index pages that have been requested from the buffer pool (logical) for temporary tablespaces.

Pool Temp Index P Reads The number of index pages read in from the tablespace containers (physical) for temporary tablespaces.

Pool Temp XDA L Reads The number of pages for XML storage object (XDA) Data that is requested from the buffer pool (logical) for temporary tablespaces.

Pool Temp XDA P Reads The number of pages for XML storage object (XDA) Data that is read in from the tablespace containers (physical) for temporary tablespaces.

Pool XDA L Reads The number of data pages for XML storage objects (XDAs) that are requested from the buffer pool (logical) for regular and large tablespaces.

Pool XDA P Reads The number of data pages for XML storage objects (XDAs) that are read in from the tablespace containers (physical) for regular and large tablespaces.

Pool XDA Writes The number of times that a buffer pool data page for an XML storage object (XDA) is physically written to disk.

Post Shr Threshold Hash Joins The total number of hash joins that were throttled back by the sort memory throttling algorithm. A throttled hash join is a hash join that was granted less memory than requested by the sort memory manager. A hash join is throttled back when the memory allocation from the shared sort heap is close to the limit that is set by the sheapthres_shr database configuration parameter. This throttling significantly reduces the number of overflows over sheapthres_shr limit in a system that is not properly configured. The data reported in this element only reflects hash joins using memory allocated from the shared sort heap.

Post Shr Threshold Sorts The total number of sorts that were throttled back by the sort memory throttling algorithm. A throttled sort is a sort that was granted less memory than requested by the sort memory manager. A sort is throttled back when the memory allocation for sorts is close to the limit that is set by the sheapthres_shr database configuration parameter. This throttling significantly reduces the number of overflows over sheapthres_shr limit in a system that is not properly configured. The data reported by this attribute only reflects sorts using memory allocated from the shared sort heap.

Priv Workspace Num Overflows The number of times that the private workspaces overflowed the bounds of its allocated memory. Use this attribute with the priv workspace size top attribute to determine whether the size of the private workspace needs to be increased to avoid overflowing. Overflows of the private workspace might cause performance degradation and out of memory errors from the other heaps allocated out of agent private memory.

Priv Workspace Section Inserts The number of inserts of SQL sections by applications into the private workspace. The working copy of executable sections are stored in the private workspace. This attribute indicates the number of times when a copy was not available and had to be inserted. At the database level, it is the cumulative total of all inserts for every application across all private workspaces in the database. At the application level, it is the cumulative total of all inserts for all sections in the private workspace for this application.

Priv Workspace Section Lookups Indicates how many times the private workspace was accessed in order to locate a specific section for an application. At the database level, it is the cumulative total of all lookups for every application across all private workspaces in the database. At the application level, it is the cumulative total of all lookups for all sections in the private workspace for this application.

Priv Workspace Size Top The largest size reached by the private workspace.

Rows Read The number of rows that are read from the table. Use this attribute to identify tables with heavy usage, and for which you might want to create additional indexes. This attribute is not the number of rows that are returned to the calling application; it is the number of rows that must be read in order to return the result set.

Shr Workspace Num Overflows The number of times that shared workspaces overflowed the bounds of their allocated memory. Use this attribute with the shr workspace size top attribute to determine whether the size of the shared workspaces need to be increased to avoid overflowing. Overflows of shared workspaces might cause performance degradation and out of memory errors from the other heaps that are allocated out of application shared memory.

Shr Workspace Section Inserts The number of inserts of SQL sections by applications into shared workspaces. The working copy of executable sections are stored in shared workspaces. This attribute indicates the number of times when a copy was not available and had to be inserted. At the database level, it is the cumulative total of all inserts for every application across all shared workspaces in the database. At the application level, it is the cumulative total of all inserts for all sections in the shared workspace for this application.

Shr Workspace Section Lookups Indicates how many times shared workspaces were accessed in order to locate a specific section for an application. At the database level, it is the cumulative total of all lookups for every application across all shared workspaces in the database. At the application level, it is the cumulative total of all lookups for all sections in the shared workspace for this application.

Shr Workspace Size Top The largest size reached by shared workspaces.

Smallest Log Avail Node Indicates the node with the least amount (in bytes) of available log space.

Snapshot Timestamp The date and time when the database system monitor information was collected. Use this attribute to help relate data chronologically if you are saving the results in a file or database for ongoing analysis.

Sort Shrheap Allocated The total amount of shared sort memory allocated in the database.

Sort Shrheap Top The high watermark (in 4KB pages) of the database-wide shared sort memory.

Stats Cache Size The current size of the statistics cache, which is used in a catalog partition to cache statistics information generated by real-time statistics gathering. Use this attribute to determine the size of the current statistics cache.

Stats Fabricate Time The total time (in milliseconds) spent on statistics fabrications by real-time statistics gathering. Statistics fabrication is the statistics collection activity needed to generate statistics during query compilation. If this attribute value is collected at the database level, it represents the total time spent on real-time statistics gathering activities for all the applications running on the database. If this attribute value is collected at the statement level, it represents the time spent on the latest real-time statistics gathering activities for the statement. The times reported by all the database partitions are aggregated together.

Stats Fabrications The total number of statistics fabrications that are performed by real-time statistics during query compilation for all the database applications. Instead of obtaining statistics by scanning data stored in a table or an index, statistics are fabricated based on metadata maintained by the index and data manager. Values reported by all the database partitions are aggregated together.

Sync Runstats The total number of synchronous RUNSTATS activities triggered by real-time statistics gathering for all the applications in the database. This value includes both successful and unsuccessful synchronous RUNSTATS commands. Values reported by all the database partitions are aggregated together.

Sync Runstats Time The total time spent on synchronous RUNSTATS activities triggered by real-time statistics gathering, in milliseconds. The synchronous RUNSTATS activities occur during query compilation. At the database level, this attribute value represents the total time spent on synchronous RUNSTATS activities for all the applications running on the database, triggered by real-time statistics gathering. At the statement level, this attribute value represents the time spent on the latest synchronous RUNSTATS activities for a particular statement, triggered by real-time statistics gathering. Values reported by all the database partitions are aggregated together.

Total Log Available The amount of active log space in the database that is not being used by uncommitted transactions (in bytes). Use this element in conjunction with the total log used attribute to determine whether you need to adjust the following configuration parameters of the monitored DB2 instance to avoid running out of log space:
- logfilsiz
- logprimary
- logsecond

The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Total Log Used Pct The percentage of the log space that is used in the database.

Total OLAP Funcs The total number of OLAP functions that run.

Unread Prefetch Pages The number of pages that the prefetcher read in but are never used. If the value is high, prefetchers are causing unnecessary I/O by reading pages into the buffer pool that will not be used.

DB2 Agent Event (KUD_Agent_Event) attributes

The Agent Event attribute group provides detailed information about predefined and triggered events and can help you determine problems with the monitored database.

Node Name The managed system name of the agent. For new installations of version 7, release 1, the format is instanceid:hostname:UD for all operating systems.

Timestamp The local time on the agent when the event was triggered.

Host Name The hostname of the machine where the DB2 database is hosted.

Instance Name The name of the monitored DB2 instance. The valid format is a text string with a maximum of 20 characters.

DB Name The real name of the database for which information is collected. This name was given to the database when it was created. The value format is a simple text string with a maximum of 60 bytes. Use this attribute to identify the specific database to which the data applies.

Event Level The level of the event. This can be either Error, Warning, Info or Misc.

Category The category of the event.

Subcategory The subcategory of the event.

SQL State The SQL state returned by the DB2 instance.

Error Code The error code returned by the DB2 instance.

Error Message The error message returned by the DB2 instance.

Description The description of the event.

Suggestion The suggestion, detailing how best to proceed once the event has been triggered.

DB2 HADR (KUD_DB2_HADR) attributes

The DB2 HADR attributes provide information about High Availability Disaster Recovery (HADR) configuration and status.

Application Current Connections Indicates the number of applications that are connected to the database.

DB Alias The alias of the database for which information is collected. The value format is a simple text string with a maximum of 60 bytes. Use this attribute to identify the specific database to which the data applies.

Instance Name The name of the monitored DB2 instance (Unicode). The valid format is a text string with a maximum of 60 bytes.

Database Status The status of the monitored database.

DB Location The location of the database.

DB Name The name of the monitored database.

DB Partition The mode of the database partition.

Connect Status The current HADR connection status of the monitored database.

Connect Time Depending on the connection status, the value is the HADR connection time, HADR congestion time, or HADR disconnection time.

Heartbeat The number of missed heartbeats on the HADR connection.

Local Host The name of the local HADR host. The value is displayed as a host name or an IP address.

Local Service The local HADR TCP service. The value is displayed as a service name or a port number.

Log Gap The average gap (in bytes) between the primary log sequence number (LSN) and the standby LSN.

Peer Window The value determines whether the database goes into the disconnected peer state after a connection is lost and how long the database remains in that state.

Peer Window End The time until which a HADR primary database stays in the peer or disconnected peer state.

Primary Log File The name of the current log file on the primary HADR database.

Primary Log LSN The current log position of the primary HADR database. The log sequence number (LSN) is a byte offset in the database log stream.

Primary Log Page The page number in the current log file, indicating the current log position on the primary HADR database.

Remote Host The name of the remote HADR host. The value is displayed as a host name or an IP address.

Remote Instance The name of the remote HADR instance.

Remote Service The remote HADR TCP service. The value is displayed as a service name or a port number.

Role The current HADR role of the monitored database.

Standby Log File The name of the current log file on the standby HADR database.

Standby Log Page The page number in the current log file, indicating the current log position on the standby HADR database.

HADR Standby Log LSN The current log position of the standby HADR database. The log sequence number (LSN) is a byte offset in the database log stream.

State The current HADR state of the database.

Syncmode The current HADR synchronization mode of the database.

Timeout The time after which if no communication is received from the partner, an HADR server determines that the connection between them fails.

Node Name The node name of the database.

Snapshot Timestamp The date and time of the database snapshot.

DCS Database (KUD_DB2_DCS_Database) attributes

The DCS Database attributes provide Direct Connection Service (DCS) database information for the monitored database gateway. You can use this information to monitor DCS database specific attributes, such as DCS connection response times and communication errors.

DB Name The real name of the host database for which information is collected or to which the application is connected. This name was given to the database when it was created. The value format is a simple text string with a maximum of 60 bytes.

DB Partition The DB2 database partition node number, which can range from 0 to 999. The Aggregated and Current Partition values can be used within a query or situation filter. If a db partition filter is not specified, data is returned for the current database partition. If a db partition filter is set to Aggregated, only aggregated partition data is returned. Historical data collection includes both aggregated and individual partition attribute data. In addition to numeric partition numbers in the 0 to 999 range, the following values are also valid:

External value	Internal value
Aggregated	-1
Current Partition	-2
All	-3

GW Comm Errors for Interval The number of times during the monitoring interval that a communication error (SQL30081) occurred while a DCS application tried to connect to a host database, or while it was processing an SQL statement. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

GW Comm Errors for Interval (Superseded) The number of times during the monitoring interval that a communication error (SQL30081) occurred while a DCS application tried to connect to a host database, or while it was processing an SQL statement.

GW Cons Wait Host The current number of connections to host databases that the DB2 Connect gateway is handling, and that are waiting for a reply from the host. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

GW Cons Wait Host (Superseded) The current number of connections to host databases that the DB2 Connect gateway is handling, and that are waiting for a reply from the host.

GW Cur Cons The current number of connections to host databases that the DB2 Connect gateway is handling. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

GW Cur Cons (Superseded) The current number of connections to host databases that the DB2 Connect gateway is handling.

Host Throughput for Interval The host throughput in bytes per second for the monitoring interval. This number represents bytes sent plus the number of bytes received divided by the cumulative host response time. The value format is an integer. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Host Throughput for Interval (Superseded) The host throughput in bytes per second for the monitoring interval. This number represents bytes sent plus the number of bytes received divided by the cumulative host response time. The value format is an integer.

Host Time per Stmt for Interval The host response time (in seconds) over the last interval, including any network time over the last interval, divided by the number of statements attempted over the last interval. The value format is an integer. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Host Time per Stmt for Interval (Superseded) The host response time (in seconds) over the last interval, including any network time over the last interval, divided by the number of statements attempted over the last interval. The value format is an integer.

Instance Name The name of the monitored DB2 instance (Unicode). The valid format is a text string with a maximum of 60 bytes.

Network Time per Stmt The total host response time minus the total statement execution time divided by the total number of attempted statements. The value format is an integer. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Network Time per Stmt (Superseded) The total host response time minus the total statement execution time divided by the total number of attempted statements. The value format is an integer.

Node Name For new installations of version 6, release 2, the format is instanceid:hostname:UD for all operating systems. The format for version 6, release 1 of the DB2 Connect agent on Windows systems is instanceid:hostname:UD; on UNIX and Linux systems, the format is instanceid:hostname.

Recent Conn Rsp Time The elapsed time (in seconds) between the start of connection processing and actual establishment of a connection for the most recent DCS application that connected to this database. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Recent Conn Rsp Time (Superseded) The elapsed time (in seconds) between the start of connection processing and actual establishment of a connection for the most recent DCS application that connected to this database.

Snapshot Timestamp The date and time when the database system monitor information was collected. Use this attribute to help relate data chronologically if you are saving the results in a file or database for ongoing analysis.

Time per Stmt The statement execution time (in seconds) divided by the number of attempted statements. The value format is an integer. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Time per Stmt (Superseded) The statement execution time (in seconds) divided by the number of attempted statements. The value format is an integer.

Diagnostic Log (KUD_DB2_Diagnostic_Log) attributes

This attribute group provides information about log record from a given facility.

Component Name The name of the component that created the message.

DB Name The real name of the database for which information is collected. This name was given to the database when it was created. The value format is a simple text string with a maximum of 60 bytes. Use this attribute to identify the specific database to which the data applies.

Facility A facility is a logical grouping which records relate to. The following values are valid: ALL, MAIN, and OPTSTATS.

Function Name The name of the function that generated the message.

Function String A string provides information about the function that generated the message, including product name, component name, function name, and probe number.

Impact This attribute qualifies the impact of this message from a user's perspective. This clarifies the impact of the message on the business process DB2 is part of. The following values are valid: NONE, UNLIKELY, POTENTIAL, IMMEDIATE, and CRITICAL.

Instance Name The name of the monitored DB2 instance. The valid format is a text string with a maximum of 60 bytes.

Level The severity level of the record. The following values are valid:

Value	Description
C	Critical
E	Error
I	Informational
S	Severe
W	Warning

Message The short description for this record. Only 256 characters are displayed.

Message Number The numeric message number.

Message Type The type of the message.

MSGID The unique message identifier of the message. The msgid is the combination of the message type, message number, and level. For example, ADM7513W.

Node Name For new installations of version 6, release 2, the format is instanceid:hostname:UD for all operating systems. The format for version 6, release 1 of the DB2 agent on Windows systems is instanceid:hostname:UD; on UNIX and Linux systems, the format is instanceid:hostname.

Partition Num The DB2 database partition node number, which can range from 0 to 999. The Aggregated and Current Partition values can be used within a query or situation filter. If a db partition filter is not specified, data is returned for the current database partition. If a db partition filter is set to Aggregated, only aggregated partition data is returned. Historical data collection includes both aggregated and individual partition attribute data. In addition to numeric partition numbers in the 0 to 999 range, the following values are also valid:

External value	Internal value
Aggregated	-1
Current Partition	-2
All	-3

PID The identifier of the operating system process that created this message. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Process Name The name of the operating system process that created this message.

Record Type The type of the record. The following values are valid:

External value	Internal value
All record types	ALL
All diagnostic records	D
All event records	E
Internal diagnostic record	DI
External diagnostic record	DX
Internal event record	EI
External event record	EX

TID The numerical identifier of the thread that created this message. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Timestamp The date and time when the message was created.

Timezone Displacement The difference between UTC (Coordinated Universal Time, formerly known as GMT) and local time at the application server.

Diagnostic Messages (KUD_DB2_Diagnostic_Messages) attributes (Superseded)

The Diagnostic Messages attribute group contains key diagnostic messages, which can be used for troubleshooting.

This attribute group is superseded, and is replaced by the Diagnostic Log attribute group.

Database Name The database name. Data collection performance can be improved with the use of a situation or query filter using this attribute. When specifying a filter with a distinct database name or list of database names, the agent data collector filters the return data prior to data transmission to the Tivoli Enterprise Monitoring Server and Tivoli Enterprise Portal Server.

Full Text of the Message The full text of the message.

Message Timestamp The date and time when the message was recorded.

MSGID The unique message identifier of the message. The msgid is a combination of the message type, message number, and level. For example, ADM7513W.

Node Name For new installations of version 6, release 2, the format is instanceid:hostname:UD for all operating systems. The format for version 6, release 1 of the DB2 agent on Windows systems is instanceid:hostname:UD; on UNIX and Linux systems, the format is instanceid:hostname.

Timezone Displacement The difference between UTC (Coordinated Universal Time, formerly known as GMT) and local time at the application server.

Locking Conflict (KUDLOCKCONFLICT00) attributes

These attributes provide information to identify which applications are waiting for resources and which applications are holding the resources.

Agent ID The application handle, which is a system-wide unique ID for the application. On multi-node systems, where a database is partitioned, this ID is the same on every node where the application might make a secondary connection. Use the application handle to identify an active application (application handle is synonymous with agent ID) uniquely.

Agent ID Holding Lock The application handle of the agent holding a lock for which this application is waiting. The lock monitor group must be turned on to obtain this information. This attribute can help you to determine which applications are in contention for resources.

Appl ID The identifier generated when the application connects to the database at the database manager or when DDCS receives a request to connect to a DRDA database. The identifier is unique across the network. The application ID is displayed in various formats, which depend on the communication protocol between the client and the server system on which the database manager or DDCS are running. Each of the formats consists of three parts separated by periods. Use this ID (known on both the client and server) to correlate the client and server parts of the application.

Appl ID (Unicode) The identifier generated when the application connects to the database at the database manager or when DDCS receives a request to connect to a DRDA database (Unicode).

Appl ID Holding Lock The application ID of the application that is holding a lock on the object that this application is waiting to obtain. The value format is a text string with a maximum of 32 characters. Use this attribute to determine which applications are in contention for resources. Specifically, it can help you to identify the application handle (agent ID) and table ID that are holding the lock. For example: `Local.db2inst1.990212202018`

Appl ID Holding Lock (Unicode) The application ID of the application that is holding a lock on the object that this application is waiting to obtain (Unicode). The value format is a text string with a maximum of 96 bytes.

Appl Name The name of the application running at the client as it is known to the database manager or DB2 Connect. The value format is a text string, with a maximum of 20 characters. For example: `*Local.db2inst1.990212202018`. Use this attribute with the Application ID attribute to relate data items with your application. In a client/server environment, this name is passed from the client to the server to establish the database connection. For DRDA-AS connections, this name is the DRDA external name.

Appl Name (Unicode) The name of the application running at the client as it is known to the database manager or DB2 Connect (Unicode). The value format is a text string, with a maximum of 60 bytes.

Appl Status The status of the application being monitored. The only valid status in LOCK WAITING. The following values are valid:

External value	Internal value
Backing Up Database	Backing_Up_Database
Commit Active	Commit_Active
Compiling SQL Stmt	Compiling_SQL_Stmt
Connect Pending	Connect_Pending
Connected	Connected
Creating Database	Creating_Database
Disconnect Pending	Disconnect_Pending
Loading Database	Loading Database
Lock Waiting	Lock Waiting
I/O Error Waiting	I/O_Error_Waiting
Prepared Transaction	Prepared_Transaction
Quiescing a Tablespace	Quiescing_a_Tablespace
Recompiling Plan	Recompiling_Plan
Request Interrupted	Request_Interrupted
Restarting Database	Restarting_Database
Restoring Database	Restoring_Database
Rollback Active	Rollback_Active
Trans. heuristically aborted	Trans._heuristically_aborted
Trans. heuristically committed	Trans._heuristically_committed
Transaction ended	Transaction_ended
UOW Executing	UOW_Executing
UOW Waiting in the application	UOW_Waiting_in_the_application
Unloading Database	Unloading_Database
UNKNOWN	UNKNOWN

Auth ID The authorization ID of the user who invoked the application that is being monitored. On a DB2 Connect gateway node, this ID is the user authorization ID on the host. The value format is a text string with a maximum of 20 characters. Use this attribute to determine who invoked the application.

Auth ID (Unicode) The authorization ID of the user who invoked the application that is being monitored (Unicode). On a DB2 Connect gateway node, this is the user authorization ID on the host. The value format is a text string with a maximum of 20 bytes.

Client DB Alias The alias defined within the database manager where the database connection request originated. The value format is a text string with a maximum of 20 characters. Use to identify the actual database that the application is accessing. The mapping between this name and Database Name can be done by using the database directories at the client node and the database manager server node. Because different database aliases can have different authentication types, this attribute can also help you determine the authentication type.

Client DB Alias (Unicode) The alias defined within the database manager where the database connection request originated (Unicode). The value format is a text string with a maximum of 60 bytes.

Codepage ID The codepage or CCSID at the node where the application started. For snapshot monitor data, this is the code page at the node where the monitored application started. Use this attribute to analyze problems for remote applications. By using this information, you can ensure that data conversion is supported between the application code page and the database code page (or for DRDA host databases, the host CCSID).

DB Partition The DB2 database partition node number, which can range from 0 to 999. The Aggregated and Current Partition values can be used within a query or situation filter. If a db partition filter is not specified, data is returned for the current database partition. If a db partition filter is set to Aggregated, only aggregated partition data is returned. Historical data collection includes both aggregated and individual partition attribute data. In addition to numeric partition numbers in the 0 to 999 range, the following values are also valid:

External value	Internal value
Aggregated	-1
Current Partition	-2
All	-3

Lock Escalation An indicator of whether a lock request was made as part of a lock escalation. The following values are valid:
- No
- Yes

Use this attribute to better understand the cause of deadlocks. If deadlocks occur that involve applications doing lock escalation, you might want to increase the amount of lock memory or change the percentage of locks that any one application can request.

Lock Mode The type of lock being held. Use the lock mode to determine the source of contention for resources. The following values are valid:

External value	Internal value
Exclusive Lock	Exclusive_Lock
Intent None	Intent_None
Intn Excl Lock	Intn_Excl_Lock
Intn Share Lock	Intn_Share_Lock
No Lock	No_Lock
Share Lock	Share_Lock
Shr Int Ex Lck	Shr_Int_Ex_Lck
Super Excl Lck	Super_Excl_Lck
U-Lock	U-Lock
Unknown	Unknown

This attribute indicates one of the following, depending on the type of monitor information being examined:

- The type of lock another application holds on the object that this application is waiting to lock (for application-monitoring and deadlock-monitoring levels)
- The type of lock held on the object by this application (for object-lock levels).

Lock Object Type The type of object against which the application holds a lock (for object-lock-level information), or the type of object for which the application is waiting to obtain a lock (for application-level and deadlock-level information). The following values are valid:

External value	Internal value
BLOCK LOCK	BLOCK LOCK
EOT LOCK	EOTLOCK
INPLACE REORG	INPLACE REORG
INTERNAL	INTERNAL
INTERNALP LOCK	INTERNALP LOCK
INTERNALV LOCK	INTERNALV LOCK
INTERNALS LOCK	INTERNALS LOCK
INTERNALJ LOCK	INTERNALJ LOCK
INTERNALL LOCK	INTERNALL LOCK
INTERNALC LOCK	INTERNALC LOCK
INTERNALB LOCK	INTERNALB LOCK
INTERNAL O LOCK	INTERNAL O LOCK
INTERNALT LOCK	INTERNALT LOCK
INTERNALQ LOCK	INTERNALQ LOCK
KEYVALUE LOCK	KEYVALUE LOCK
No Lock	No_Lock
ROW	ROW
SYSBOOT LOCK	SYSBOOT LOCK
TABLE	TABLE
TABLESPACE	TABLESPACE
UNKNOWN	UNKNOWN

Use this attribute to help you determine the source of contention for resources.

Lock Wait Start Time The string date and time that the application started waiting to obtain a lock on the object that is currently locked by another application. The value format is CYYMMDDHHMMSSmmm, where the following measures apply:

Value	Description
C	Century (0 for 20th, 1 for 21st)
YY	Year
MM	Month
DD	Day
HH	Hour
MM	Minute
SS	Second

Value	Description
mmm	Millisecond

Use this attribute to help you determine the severity of resource contention.

Lock Wait Start Timestamp The date and time that the application started waiting to obtain a lock on the object that is currently locked by another application. The following special values cause the Tivoli Enterprise Portal to display fixed string messages:

Internal value	Tivoli Enterprise Portal display
0000000000000001	N/A (Not Applicable or Not Available)
0000000000000002	N/C (Not Calculated)
0000000000000003	N/P (Not Present)

Lock Wait Time The total elapsed time (in milliseconds) that a lock was waited for. At the database level, this is the total amount of elapsed time that all applications were waiting for a lock within this database. At the application-connection and transaction levels, this is the total amount of elapsed time that this connection or transaction has waited for a lock to be granted. This attribute might be used with the Lock Waits attribute to calculate the average wait time for a lock. This calculation can be performed at either the database or the application-connection level. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Lock Wait Time (Superseded) The total elapsed time (in milliseconds) that a lock was waited for. At the database level, this is the total amount of elapsed time that all applications were waiting for a lock within this database. At the application-connection and transaction levels, this is the total amount of elapsed time that this connection or transaction has waited for a lock to be granted. This attribute might be used with the Lock Waits attribute to calculate the average wait time for a lock. This calculation can be performed at either the database or the application-connection level.

Locks Held The number of locks currently held. If the monitor information is at the database level, this is the total number of locks currently held by all applications in the database. If it is at the application level, this is the total number of locks currently held by all agents for the application. Usage of this attribute depends on the level of information being returned from the database system monitor. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Locks Held (Superseded) The number of locks currently held. If the monitor information is at the database level, this is the total number of locks currently held by all applications in the database. If it is at the application level, this is the total number of locks currently held by all agents for the application. Usage of this attribute depends on the level of information being returned from the database system monitor.

Node Name For new installations of version 6, release 2, the format is instanceid:hostname:UD for all operating systems. The format for version 6, release 1 of the DB2 agent on Windows systems is instanceid:hostname:UD; on UNIX and Linux systems, the format is instanceid:hostname.

Snapshot Time The string date and time when the database system monitor information was collected. Use this attribute to help relate data chronologically if you are saving the results in a file or database for ongoing analysis. The Tivoli Enterprise Portal formats the timestamp value into a date and time string. The internal timestamp value stored in the database is in the format cYYMMDDhhmmss000, which is described in the following table:

Value	Description
C	Century (0 for 20th, 1 for 21st)
YY	Year (last two digits of the year, for example (00 - 99)
MM	Month (01 for January, 02 for February, and so on)
DD	Day (the day of the month, for example 01 - 31)
HH	Hour (the hour of the day in 24-hour format from 00 - 23)
MM	Minute (the minutes of the hour from 00 - 59)
SS	Second (the seconds of the hour (00 - 59)

Snapshot Timestamp The string date and time when the database system monitor information was collected. Use this attribute to help relate data chronologically if you are saving the results in a file or database for ongoing analysis. The Tivoli Enterprise Portal formats the timestamp value into a date and time string. The internal timestamp value stored in the database is in the format cYYMMDDhhmmss000, which is described in the following table:

Value	Description
C	Century (0 for 20th, 1 for 21st)
YY	Year (last two digits of the year, for example (00 - 99)
MM	Month (01 for January, 02 for February, and so on)
DD	Day (the day of the month, for example 01 - 31)
HH	Hour (the hour of the day in 24-hour format from 00 - 23)
MM	Minute (the minutes of the hour from 00 - 59)
SS	Second (the seconds of the hour (00 - 59)

The following special values cause the Tivoli Enterprise Portal to display fixed string messages:

Internal value	Tivoli Enterprise Portal display
0000000000000001	N/A (Not Applicable or Not Available)
0000000000000002	N/C (Not Calculated)
0000000000000003	N/P (Not Present)

Status Change Time The string date and time the application entered its current status. The value format is CYYMMDDHHMMSSmmm, where the following measures apply:

Value	Description
C	Century (0 for 20th, 1 for 21st)
YY	Year
MM	Month
DD	Day
HH	Hour
MM	Minute
SS	Second
mmm	Millisecond

Use this attribute to determine how long an application has been in its current status. If the application status remains unchanged for a long period of time, the application might have a problem.

Status Change Timestamp The date and time the application entered its current status. The following special values cause the Tivoli Enterprise Portal to display fixed string messages:

Internal value	Tivoli Enterprise Portal display
0000000000000001	N/A (Not Applicable or Not Available)
0000000000000002	N/C (Not Calculated)
0000000000000003	N/P (Not Present)

Table Name The name of the table against which the application is holding locks. The value format is a text string with a maximum of 20 characters. Use this attribute with the Table Schema attribute to help you determine the source of resource contention.

Table Name (Unicode) The name of the table against which the application is holding locks (Unicode). The value format is a text string with a maximum of 60 bytes.

Table Schema The schema of the table that is holding a lock. The value format is a text string with a maximum of 20 characters. Along with the Table Name attribute, use this attribute to help you determine the source of resource contention.

Table Schema (Unicode) The schema of the table against which the application is holding a lock (Unicode). The value format is a text string with a maximum of 60 bytes.

Tablespace Name The name of the tablespace against which the application is holding a lock. The value format is a text string with a maximum of 20 characters. Use this attribute to help you determine the source of resource contention.

Tablespace Name (Unicode) The name of the tablespace against which the application currently holds a lock (Unicode). The value format is a text string with a maximum of 60 bytes.

Log (KUD_DB2_LOG) attributes

The log attribute group provides information about database configuration parameters that are related to archive logs, the number of archive logs, and the size of archived log path.

Arch Retry Delay The archive retry delay on error configuration parameter of the monitored DB2 instance. This parameter specifies the number of seconds to wait after a failed archive attempt before trying to archive the log file again. Subsequent retries only takes affect if the value of the numarchretry database configuration parameter is at least 1. The following values are valid:

External value	Internal value
Value_Exceeds_Maximum	2147483647
Value_Exceeds_Minimum	-2147483647

Backup Pending This parameter indicates whether you need to do a full backup of the database before accessing it. This parameter is only on if the database configuration is changed so that the database moves from being nonrecoverable to recoverable (that is, initially both the logretain and userexit parameters were set to NO, and then either one or both of these parameters is set to YES, and the update to the database configuration is accepted).

Current Active Log The file number of the active log file that the monitored DB2 instance is currently writing. Use this attributes with the first active log and last active log attributes to determine the range of active log files. Knowing the range of active log files helps you determine the disk space required for log files. You can also use this attribute to determine which log files have data to help you identify log files needed for split mirror support. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Current Archive Log The file number of the log file the DB2 instance is currently archiving. If the DB2 instance is not archiving a log file, the value for this element is SQLM_LOGFILE_NUM_UNKNOWN. Use this attribute to determine if there is a problem archiving log files. Such problems include the following two problems:

- Slow archive media
- Archive media that is not available

Current Primary Log Used Percent The percentage of the primary log space that are currently used.

Current Secondary Log Used Percent The percentage of the secondary log space that are currently used.

Database Is Consistent This attribute indicates whether the database is in a consistent state. Valid values are 1 (YES) and 0 (NO).

DB Alias The alias of the database for which information is collected. The value format is a simple text string with a maximum of 60 bytes. Use this attribute to identify the specific database to which the data applies.

DB Name The real name of the database for which information is collected. This name was given to the database when it was created. The value format is a simple text string with a maximum of 60 bytes. Use this attribute to identify the specific database to which the data applies.

DB Partition The DB2 database partition node number, which can range from 0 to 999. The Aggregated and Current Partition values can be used within a query or situation filter. If you do not specify a db partition filter, data is returned for either the current database partition (single partition environment) or the aggregated database partitions (multiple partition environment). If a db partition filter is set to Aggregated, only aggregated partition data is returned. Historical data collection includes both aggregated and individual partition attribute data. In addition to numeric partition numbers in the 0 to 999 range, the following values are also valid:

External value	Internal value
Aggregated	-1
Current Partition	-2
All	-3

Fail Log Path This attribute specifies a path to which the monitored DB2 instance will try to archive log files if the log files cannot be archived to either the primary or the secondary (if set) archive destinations because of a media problem affecting those destinations. This specified path must reference a disk. If there are log files in the directory that is specified by the fail log path attribute, any updates to the fail log path attribute will not take effect immediately. Instead, the update will take effect when all applications disconnect.

Fail Log Path Free Size The amount of free space (in MB) that is available on the file system that is pointed to by the fail log path attribute. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Fail Log Path Total Size The capacity of the file system (in MB) that is pointed to by the fail log path attribute. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

First Active Log The file number of the first active log file. Use this attribute with the last active log and current active log attributes to determine the range of active log files. Knowing the range of active log files helps you determine the disk space required for log files. You can also use this attribute to determine which log files have data to help you identify log files needed for split mirror support. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Instance Name The name of the monitored DB2 instance. The valid format is a text string with a maximum of 60 bytes.

Last Active Log The file number of the last active log file. Use this attribute with the first active log and current active log attributes to determine the range of active log files. Knowing the range of active log files helps you determine the disk space required for log files. You can also use this attribute to determine which log files have data to help you identify log files needed for split mirror support. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Log Arch Meth1 The media type of the primary destination for archived logs.

Log Arch Meth2 The media type of the secondary destination for archived logs.

Log Arch Meth1 Free Size The amount of free space (in MB) of the primary destination for archived logs. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Log Arch Meth2 Free Size The amount of free space (in MB) of the secondary destination for archived logs. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Log Arch Meth1 Total Size The capacity (in MB) of the primary destination for archived logs. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Log Arch Meth2 Total Size The capacity (in MB) of the secondary destination for archived logs. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Log Buffer Size (4KB) This value specifies the amount (in 4KB) of the database heap to use as a buffer for log records before writing these records to disk. It is important that the log buffer can hold the amount of log space used by an average transaction. Otherwise, logging performance decreases and slows the overall system. The valid format is integer.

Log File Size (4KB) This value defines the size (in 4KB) of each primary and secondary log file. The size of these log files limits the number of log records that can be written to them before they become full and a new log file is required. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Log Held by Dirty Pages The amount of log (in bytes) corresponding to the difference between the oldest dirty page in the database and the top of the active log. When the snapshot is taken, this value is calculated based on conditions at the time of that snapshot. Use this element to evaluate the effectiveness of page cleaning for older pages in the buffer pool. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Log Path This value indicates the current path that is used for logging purposes.

Log Path Free Size The amount of the free space (in MB) on the file system that is pointed to by the log path attribute. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Log Path Total Size The capacity of the file system (in MB) that is pointed by the log path attribute. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Log Primary The number of primary log files. The following values are valid:

External value	Internal value
Value_Exceeds_Maximum	2147483647
Value_Exceeds_Minimum	-2147483648

Log Reads The number of log pages that are read from disk by the logger. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Log Read Time The total elapsed time that the logger spends reading log data from the disk. Use this attribute with the log reads, num log read io, and num log data found in buffer attributes to determine the following items:

- Whether the current disk is adequate for logging.
- Whether the log buffer size is adequate.

The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Log Second The number of secondary log files. The following values are valid:

External value	Internal value
Value_Exceeds_Maximum	2147483647
Value_Exceeds_Minimum	-2147483648

Log to Redo for Recovery The amount of log (in bytes) that will have to be redone for crash recovery. When the snapshot is taken, this value is calculated based on conditions at the time of that snapshot. Larger values indicate longer recovery times after a system crash. If the value seems excessive, check the log held by dirty pages attribute to see if page cleaning needs to be tuned. Also check if there are any long running transactions that need to be terminated. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Log Retain The value of the log retain enable configuration parameter. The attribute is deprecated in DB2 Version 9.5, but is used in pre-Version 9.5 data servers and clients. Valid values are OFF = 0, RECOVERY = 1, and CAPTURE = 2.

Log Writes The number of log pages written to disk by the logger. Use this attribute with an operating system monitor to quantify the amount of I/O on a device that is attributable to database activity. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Log Write Time The total elapsed time that the logger spends writing log data to the disk. Use this attribute with the log writes and num log write io attributes to determine whether the current disk is adequate for logging. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Mirror Log Path The string that is specified for the mirror log path. The string points to a full qualified path name.

Mirror Log Path Free Size The amount of the free space (in MB) of the file system that is pointed by the mirror log path attribute. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Mirror Log Path Total Size The capacity (in MB) of the file system that is pointed by the mirror log path attribute. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

New Log Path The current value of the newlogpath configuration parameter. A valid value is a text string up to 768 characters in length. You can use the newlogpath configuration parameter to specify a new location for the log files. The specified path does not become the current log path until both of the following conditions are met:

- The database is in a consistent state.
- All users are disconnected from the database.

When the first new connection is made to the database, the database manager moves the logs to this location.

New Log Path Free Size The amount of the free space (in MB) of the file system that is pointed by the new log path attribute. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

New Log Path Total Size The capacity (in MB) of the file system that is pointed by the new log path attribute. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Node Name For new installations of version 6, release 2, the format is instanceid:hostname:UD for all operating systems. The format for version 6, release 1 of the DB2 agent on Windows systems is instanceid:hostname:UD; on UNIX and Linux systems, the format is instanceid:hostname.

Num Arch Retry The number of times that the monitored DB2 instance is to try archiving a log file to the primary or the secondary archive directory before trying to archive log files to the failover directory. This parameter is only used if the failarchpath database configuration parameter is set. If the failarchpath database configuration parameter is not set, DB2 will continuously retry archiving to the primary or the secondary log path. The following values are valid:

External value	Internal value
Value_Exceeds_Maximum	2147483647
Value_Exceeds_Minimum	-2147483647

Num Log Buffer Full The number of times that agents have to wait for log data to write to disk while copying log records into the log buffer. This value is increased per agent per incident. For example, if two agents attempt to copy log data while the buffer is full, this value is increased by two. Use this attribute to determine if the LOGBUFSZ database configuration parameter needs to be increased. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Num Log Data Found in Buffer The number of times that an agent reads log data from the buffer. Reading log data from the buffer is preferable to reading from the disk because the latter is slower. Use this attribute with the num log read io attribute to determine if the LOGBUFSZ database configuration parameter needs to be increased. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Num Log Part Page IO The number of I/O requests that are issued by the logger for writing partial log data to the disk. Use this attribute with the log writes, log write time, and num log write io attributes to determine if the current disk is adequate for logging. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Num Log Read IO The number of I/O requests that are issued by the logger for reading log data from the disk. Use this attribute with the log reads and log read time attributes to determine if the current disk is adequate for logging. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Num Log Write IO The number of I/O requests that are issued by the logger for writing log data to the disk. Use this attribute with the log writes and log write time attributes to determine if the current disk is adequate for logging. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Overflow Log Path The location for DB2 to find log files that are needed for a rollforward operation, and to store active log files that are retrieved from the archive. It also gives a location for finding and storing log files that are needed for using db2ReadLog API.

Overflow Log Path Free Size The amount of free space (in MB) of the file system that is specified by the overflow log path attribute. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Overflow Log Path Total Size The capacity (in MB) of the file system that is specified by the overflow log path attribute. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Primary Log Used Percent The percentage of total log space used by the primary log. Use the returned value to help you evaluate the allocated amount of primary log space and refine the log buffer size, log file size, and primary log configuration parameters. The returned value is valid only if circular logging is used.

Restore Pending This attribute indicates whether a RESTORE PENDING status exists in the database. Valid values are 1(YES) and 0 (NO).

Rollforward Pending This attribute indicates whether the monitored DB2 instance is the rollforward pending status. Valid values are 1(YES) and 0 (NO).

Sec Logs Allocated The total number of secondary log files that are currently being used for the database. Use this attribute with the Secondary Log Used Top and Total Log Used Top attributes to show the current dependency on secondary logs. If this value is consistently high, you might need larger log files, more primary log files, or more frequent COMMIT statements within your application. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Sec Log Used Percent The percentage of maximum log space used by the secondary log. Use the returned value to show the current dependency on secondary logs. Secondary logs are used when you have circular logging (log retention off) and the primary log files are full.

Sec Log Used Top The maximum amount of secondary log space (in bytes) that has been used. Use this attribute with the Secondary Logs Allocated and Total Log Used Top attributes to show the current dependency on secondary logs. If this value is high, you might need larger log files, more primary log files, or more frequent COMMIT statements within your application. Values that are greater than or equal to 9223372036854775807 are indicated with the `Value Exceeds Maximum` text in the portal. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Snapshot Timestamp The date and time when the database system monitor information was collected. Use this attribute to help relate data chronologically if you are saving the results in a file or database for ongoing analysis.

Total Log Available The amount of active log space in the database that is not being used by uncommitted transactions (in bytes). Use this element in conjunction with the total log used attribute to determine whether you need to adjust the following configuration parameters of the monitored DB2 instance to avoid running out of log space:

- logfilsiz
- logprimary
- logsecond

The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Total Log Used The total log space used in bytes in the database. The value format is an integer. Values that are greater than or equal to 9223372036854775807 are indicated with the `Value Exceeds Maximum` text in the portal. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Total Log Used Pct The percentage of the log space that is in used the database. The value is calculated using the following formula:

```
100 * (total_log_used/(total_log_used + total_log_available))
```

Total Log Used Top The maximum amount of total log space (in bytes) that has been used. The value format is an integer. Use this attribute to evaluate the amount of primary log space that is allocated. Comparing the value of this attribute with the amount of primary log space that is allocated can help you evaluate the configuration parameter settings. Values that are greater than or equal to 9223372036854775807 are indicated with the `Value Exceeds Maximum` text in the portal. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

User Exit The value of the user exit enable configuration parameter. The parameter is deprecated in DB2 Version 9.5, but is used in pre-Version 9.5 data servers and clients.

Log Record (KUD_DB2_LOG_RECORD) attributes

The log record attribute group provides historical information about archived logs. This attribute group is only available for DB2 Version 9.1 and later.

Backup ID The backup identifier or unique table identifier.

DB Alias The alias of the database for which information is collected. The value format is a simple text string with a maximum of 60 bytes. Use this attribute to identify the specific database to which the data applies.

DB Name The real name of the database for which information is collected. This name was given to the database when it was created. The value format is a simple text string with a maximum of 60 bytes. Use this attribute to identify the specific database to which the data applies.

DB Partition The DB2 database partition node number, which can range from 0 to 999. The Aggregated and Current Partition values can be used within a query or situation filter. If you do not specify a db partition filter, data is returned for either the current database partition (single partition environment) or the aggregated database partitions (multiple partition environment). If a db partition filter is set to Aggregated, only aggregated partition data is returned. Historical data collection includes both aggregated and individual partition attribute data. In addition to numeric partition numbers in the 0 to 999 range, the following values are also valid:

External value	Internal value
Aggregated	-1
Current Partition	-2
All	-3

Device Type The type identifier of the device that is associated with a logged event. The following values are valid:

External value	Internal value
TSM	A
Client	C
Disk	D
Diskette	K
Local	L
Generated internally by DB2	N
Other	O
Pipe	P
Cursor	Q
Remote fetch data	R
Server	S
Tape	T
Userexit	U

End Timestamp The date and time that a logged event ended.

Entry Status The identifier for the status of an entry in the history file. The following values are valid:

External values	Internal values
Active	A
Deleted	D
Expired	E
Inactive	I
Not yet committed	N
Committed or active	Y

First Log The name of the earliest transaction log that is associated with an event.

Instance Name The name of the monitored DB2 instance. The valid format is a text string with a maximum of 60 bytes.

Last Log The name of the latest transaction log that is associated with an event.

Location The full path of the location to store files, such as backup images or load input file, that are associated with logged events.

Node Name For new installations of version 6, release 2, the format is instanceid:hostname:UD for all operating systems. The format for version 6, release 1 of the DB2 agent on Windows systems is instanceid:hostname:UD; on UNIX and Linux systems, the format is instanceid:hostname.

Object Type The identifier for the target object of an operation. The following values are valid:

External values	Internal values
Full database	D
Tablespace	P
Table	T

Operation The operation identifier. The following values are valid:

External values	Internal values
Add tablespace	A
Backup	B
Load copy	C
Dropped table	D
Rollforward	F
Reorganize table	G
Load	L
Rename tablespace	N
Drop tablespace	O
Quiesce	Q
Restore	R
Alter tablespace	T
Unload	U
Archive logs	X

Operation Type The action identifier of an operation. The following values are valid:

External values	Internal values
Fail archive path	F
Mirror log path	M
Forced truncation via ARCHIVE LOG command	N
Primary log path	P
First log archive method	1
Second log archive method	2

Sequence Number The sequence number. The following value is valid:

External values	Internal values
Value_Exceeds_Maximum	9223372036854775807

Snapshot Timestamp The date and time when the database system monitor information was collected. Use this attribute to help relate data chronologically if you are saving the results in a file or database for ongoing analysis.

Start Timestamp The date and time that a logged event started.

Uniquely Identifies The number that uniquely identifies an entry in the history file.

Network Info (KUD_DB2_IPADDR_TABLE) attributes

The Network Information attribute group provides network information of the monitored DB2 instance.

DB Partition The DB2 database partition node number, which can range from 0 to 999. The Aggregated and Current Partition values can be used within a query or situation filter. If you do not specify a db partition filter, data is returned for either the current database partition (single partition environment) or the aggregated database partitions (multiple partition environment). If a db partition filter is set to Aggregated, only aggregated partition data is returned. Historical data collection includes both aggregated and individual partition attribute data. In addition to numeric partition numbers in the 0 to 999 range, the following values are also valid:

External value	Internal value
Aggregated	-1
Current Partition	-2
All	-3

DB2 Server Name The name of the DB2 server.

IP address The IP address that is used by the DB2 server.

IP Protocol The IP protocol type of the DB2 server.

Listener port The TCP/IP port that the database server uses in communication with a remote client.

Node Name For new installations of version 6, release 2, the format is instanceid:hostname:UD for all operating systems. The format for version 6, release 1 of the DB2 agent on Windows systems is instanceid:hostname:UD; on UNIX and Linux systems, the format is instanceid:hostname.

System Overview (KUD_DB2_System_Overview) attributes

Important: The values of the following attributes are only available when the monitored database is multi-partitioned, or when the monitored database is single-partitioned and the intra-partition parallelism is enabled by setting the intra_parallel parameter to YES.

- Buff Max Used Percent

- Buff Used Percent
- CE Max Used Percent
- CE Used Percent
- MA Max Used Percent
- RB Max Used Percent
- RB Used Percent

Use the System Overview attributes to monitor general information about the DB2 UDB subsystem in your environment. The following traits are included:

- Agent-related high-water marks
- Local and remote connections
- Sorting
- Start time
- Status

Agents Created Empty Pool The number of agents created because the agent pool was empty. It includes the number of agents started at DB2 start up. By using the Agents Assigned From Pool attribute, you can calculate the ratio of the Agents Created Empty Pool attribute to the Agents From Pool attribute. See the Agents From Pool attribute for information about using this attribute. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Agents Created Empty Pool Ratio The percentage of agents that are created because the pool is empty. This ratio is calculated by dividing the value of the Agents Created Empty Pool attribute by the value of the Agents From Pool attribute. Use this attribute to evaluate how often an agent must be created because the pool is empty.

Agents from Pool The number of agents assigned from the pool. Use this attribute with Agents Created Empty Pool attribute to determine how often an agent must be created because the pool is empty. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Agents Registered The number of agents that the database manager registered. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Agents Registered Top The maximum number of agents that the database manager has ever registered, at the same time, since it was started (coordinator agents and subagents). Use this attribute to evaluate the setting of the MAXAGENTS configuration parameter. The number of agents registered at the time the snapshot was taken is recorded by the Agents Registered attribute. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Agents Stolen The number of times that agents are stolen from an application. Agents are stolen when an idle agent associated with an application is reassigned to work on a different application. Use this attribute with the Maximum Number of Associated Agents attribute to evaluate the load that this application places on the system. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Agents Waiting on Token The number of agents waiting for a token so they can run a transaction in the database manager. Use this attribute to evaluate your setting for the MAXCAGENTS configuration parameter. Each application has a dedicated coordinator agent to process database requests (transactions) within the database manager. Each agent must have a token to run a transaction. The maximum number of coordinator agents is limited by the MAXCAGENTS configuration parameter. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Agents Waiting on Token Percent The percentage of agents waiting on a token. The percentage is calculated by dividing the value of the Agents Waiting on Token attribute by the number of local applications that are currently connected to a database (Local Cons attribute). Use this attribute to assess the number of agents

Agents Waiting Top The highest number of agents waiting on a token, at the same time, since the database manager was started. Use this attribute to evaluate the setting of the MAXCAGENTS configuration parameter. In contrast, the Agents Waiting on Token attribute records the number of agents waiting for a token at the time the snapshot was taken. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Appl Support Layer Heap Size The current value (in units of 4-KB pages) of the application support layer heap size. This value is the amount of memory that is allocated for the application support layer heap. This heap is used as a communication buffer between the local application and its associated agent. In addition, this value is used to determine the input and output block size when a blocking cursor is opened. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Buff Free The number of Fast Communication Manager (FCM) buffers that are free in the partitioned database server during the monitoring interval. Use the returned value to determine the current buffer pool utilization. Use this information to refine the configuration of the number of FCM buffers. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Buff Free Bottom The minimum number of free connection entries. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Buff Max Used Percent The percentage of maximum FCM buffers used during processing within the partitioned database server. If the percentage of maximum FCM buffers used is high compared to normal operating levels, you can increase the number of FCM buffers; if the percentage is low compared to normal operating levels, you can decrease the value.

Buff Used Percent The percentage of FCM buffers that are used within the partitioned database server during the monitoring interval. If the percentage of FCM buffers used is high compared to normal operating levels, you can adjust the number of FCM buffers.

CE Free The number of connection entries that are free in the partitioned database server during the monitoring interval. Use the returned value to help determine the current connection entry utilization. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

CE Free Bottom The minimum number of free connection entries. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

CE Max Used Percent The maximum percentage of FCM connection entries used during processing within the partitioned database server. If the percentage of maximum FCM connection entries used is high compared to normal operating levels, you can increase the number of FCM connections; if the percentage is low compared to normal operating levels, you can decrease the value.

CE Used Percent The percentage of FCM connection entries used during processing within the partitioned database server. If the percentage of FCM connection entries used is high compared to normal operating levels, you can increase the number of FCM connections; if the percentage is low compared to normal operating levels, you can decrease the value.

Committed Private Memory The amount of private memory that the instance of the database manager currently has committed at the time of the snapshot. Use this attribute to assess the MIN_PRIV_MEM configuration parameter to ensure that enough private memory is available. This attribute is returned for all platforms, but tuning can be accomplished only on platforms where DB2 uses threads (such as OS/2 and Windows NT systems). Values that are greater than or equal to 9223372036854775807 are indicated with the `Value Exceeds Maximum` text in the portal. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Conn Local Database The number of local databases with current connections to the monitored DB2 instance. This value gives an indication of how many database information records to expect when gathering data at the database level. The applications can be running locally or remotely, and might or might not be executing a unit of work within the database manager. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Connection Status The status of the communication connection between the database partition that is specified by the DB2 Node Number variable and the database partition where this monitor runs. Two nodes can be active, but the connection between them remains inactive unless there is active communication between them.

Cons in Exec Percent The percentage of the maximum number of applications allowed that are connected to a database and processing a unit of work during the monitoring interval.

Coordinating Agents Top The maximum number of coordinating agents working at one time. The MAXCAGENTS configuration parameter determines the number of coordinating agents that can be executing concurrently. If the peak number of coordinating agents results in a workload that is too high for this node, you can reduce the MAXCAGENTS configuration parameter. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

DB Partition The DB2 database partition node number, which can range from 0 to 999. The Aggregated and Current Partition values can be used within a query or situation filter. If you do not specify a db partition filter, data is returned for either the current database partition (single partition environment) or the aggregated database partitions (multiple partition environment). If a db partition filter is set to Aggregated, only aggregated partition data is returned. Historical data collection includes both aggregated and individual partition attribute data. In addition to numeric partition numbers in the 0 to 999 range, the following values are also valid:

External value	Internal value
Aggregated	-1
Current Partition	-2
All	-3

DB2 Available The amount of time (in seconds) the instance has been available since a DB2START command was issued. The value format is an integer. The value is derived through this formula:

```
snapshot time - db2start time
```

The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

DB2 Instance Status The current status of the DB2 instance. The following table shows the valid values:

External value	Internal value	Description
Active	Active	The DB2 instance is currently active.
Inactive/Busy	Inactive/Busy	The DB2 instance is either not running or is too busy to respond.
Quiesce Pending	Quiesce_Pending	A quiesce is pending for the DB2 instance.
Quiesced	Quiesced	The DB2 instance is currently quiesced.
Unknown	Unknown	The DB2 instance is active, but the state is not recognized.

Use this attribute to determine the state of your database manager instance.

DB2 Server Type The type of database manager being monitored. The following values are valid:

External value	Internal value
Client with local databases	Client_with_local_databases
Client/server	Client/server
Host Database Server	Host_Database_Server
MPP	MPP
Requestor	Requestor
Satellite	Satellite
Standalone	Standalone
UNKNOWN	UNKNOWN

DB2 Start Timestamp The date and time that the database manager was started using the DB2START command. Use this attribute with the snapshot time attribute to calculate the elapsed time from the start of the database manager until the snapshot was taken.

DB2 Version The version of the server that is returning the data. For example: 6.1 or 7.1. The data structures used by the monitor might change between releases. As a result, check the version of the data stream to determine whether your applications can process the data. The following values are valid:

External value	Internal value
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
UNKNOWN	UNKNOWN

DBPG Node Status The list of failing local nodes. This is a list of integers, where each integer represents the failed local nodes. Depending on the actual partitions defined in the database partition group, it is not necessarily true that all the nodes defined in the parallel environment are examined. To ensure that all the nodes in the partitioned environment are examined, define a partition group that contains at least one database partition from each of the nodes in the partitioned environment.

FCM Num Anchors The number of FCM message anchors for the DB2 instance during the monitoring interval. Agents use the message anchors to send messages among themselves. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

FCM Num Buffers The number of buffers that are used for internal communications (messages) among the nodes and within the nodes in a DB2 instance during the monitoring interval. You might need to increase the value of this parameter if you have either of the following conditions: multiple logical nodes on a processor, or too many users, nodes, or complex applications that exceed the buffer limit. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

FCM Num Connect The number of FCM connection entries for the DB2 instance during the monitoring interval. Agents use connection entries to pass data among themselves. Use the results from the fcm_num_rqb attribute to help you refine the fcm_num_connect attribute. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

FCM Num Rqb The number of FCM request blocks for the DB2 instance during the monitoring interval. Request blocks are the media through which information is passed between the FCM daemon and an agent. The requirement for request blocks varies according to the number of users on the system, the number of database partition servers in the system, and the complexity of queries that are run. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Gateway Cons Wait Client For host databases being handled by the DB2 Connect gateway, the current number of connections that are waiting for the client to send a request. Because this value can change frequently, take samples at regular intervals over an extended period to obtain a realistic view of gateway usage. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Gateway Cons Wait Host For host databases being handled by the DB2 Connect gateway, the current number of connections that are waiting for a reply from the host. Because this value can change frequently, take samples at regular intervals over an extended period to obtain a realistic view of gateway usage. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Gateway Current Connections The current number of connections to host databases being handled by the DB2 Connect gateway. Use this attribute to help you understand the level of activity at the DB2 Connect gateway and the associated use of system resources. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Gateway Total Connections The total number of connections attempted from the DB2 Connect gateway since the last db2start command or the last reset. Use this attribute to help you understand the level of activity at the DB2 Connect gateway and the associated use of system resources. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Idle Agents The number of agents in the agent pool that are currently unassigned to an application. Use this attribute to set the NUM_POOLAGENTS configuration parameter. By having idle agents available to satisfy requests for agents, you can improve performance. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Instance Name The name of the monitored DB2 instance. The valid format is a text string with a maximum of 60 bytes.

Last Reset Timestamp The date and time that the monitor counters were reset for the application requesting the snapshot. Use this attribute to help you determine the scope of information returned by the database system monitor.

Local Connections The number of local applications that are currently connected to a database within the database manager instance being monitored. By using this number, you can determine the level of concurrent processing occurring in the database manager. This value changes frequently. As a result, you must sample the data at specific intervals over an extended period of time to get a realistic view of system usage. This number includes only applications that were initiated from the same instance as the database manager. The applications are connected, but might or might not be executing a unit of work in the database. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Local Connection Executing The number of local applications that are currently connected to a database within the database manager instance being monitored and are currently processing a unit of work. By using this number, you can determine the level of concurrent processing occurring in the database manager. This value changes frequently. As a result, you must sample the data at specific intervals over an extended period of time to get a realistic view of system usage. This number includes only applications that were initiated from the same instance as the database manager. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

MA Free Bottom The minimum number of free message anchors. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

MA Max Used Percent The maximum number of message anchors used as a percentage.

Max Agent Overflows The number of attempts to create a new agent when the MAXAGENTS configuration parameter had already been reached. If requests to create new agents are received after reaching the MAXAGENTS configuration parameter, the workload for this node might be too high. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Max Agents The current value of the maximum number of existing agents. This value is the maximum number of database manager agents available at any given time to accept application requests. This value limits the total number of applications that can connect to all databases in the DB2 instance at a given time. The value of the maxagents attribute must be the sum of the values of the maxappls attribute in each database that is allowed to be accessed concurrently. Increasing the value of the maxagents attribute can increase resource use because resources for each agent are allocated when the DB2 instance is started. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Max Conc Agents The maximum number of database manager coordinator agents that can concurrently run a database manager transaction in the DB2 instance during the monitoring interval. When this monitor is used with DB2 Universal database servers, the maxcagents value is the default. This value is the maximum number of database manager agents that can be concurrently executing a database manager transaction. Use the maxcagents attributes to control the load on the system during periods of high simultaneous application activity. A value of -1 indicates that the limit is equal to the maximum number of agents (the MAXAGENTS parameter). The maxcagents parameter does not limit the number of applications that can have connections to the database.

Max Coord Agents The maximum number of database manager coordinating agents that can exist on a server in a partitioned or nonpartitioned database environment. One coordinating agent is acquired for each local or remote application that connects to a database or attaches to an instance. Requests that require an instance attachment include CREATE DATABASE, DROP DATABASE, and Database System Monitor commands.

Mon Heap Size The current value (in units of 4-KB pages) of the database system monitor heap size. This value is the amount of memory that is allocated for database system monitor data. A value of zero prevents the database manager from collecting database system monitor data.

Node Name For new installations of version 6, release 2, the format is instanceid:hostname:UD for all operating systems. The format for version 6, release 1 of the DB2 agent on Windows systems is instanceid:hostname:UD; on UNIX and Linux systems, the format is instanceid:hostname.

Piped Sort Hit Ratio Percent for Interval The piped sort hit ratio (as a percentage) for the last monitoring interval. The piped sort hit ratio is the ratio of piped sorts accepted to piped sorts requested.

Piped Sorts Accepted The number of piped sorts that have been accepted. When the number of accepted piped sorts is low compared to the number requested, you can improve sort performance by adjusting one or both of the following configuration parameters:

- SORTHEAP
- SHEAPTHRES

If piped sorts are being rejected, consider decreasing your sort heap or increasing your sort heap threshold. Be aware of the possible implications of these options:

- If you increase the sort heap threshold, more memory might remain allocated for sorting. This can cause the paging of memory to disk.

- If you decrease the sort heap, an extra merge phase (which can slow down the sort) might be required.

The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Piped Sorts Accepted Percent The percentage of piped sorts that have been accepted. The percentage is calculated by dividing the value of the Piped Sorts Accepted attribute by the value of the Piped Sorts Requested attribute. Use this attribute to determine whether the value of the Piped Sorts Accepted attribute is in an acceptable range.

Piped Sorts Rejected for Interval The total number of piped sorts that were rejected during the monitoring interval. In the return phase of sorting, if the sorted information can return directly through the sort heap, it is a piped sort. However, even if the optimizer requests a piped sort, this request is rejected at run time if the total amount of sort heap memory for all sorts on the database is close to exceeding the sheapthres value. If this returned value is high compared to the total number of sorts requested, consider decreasing your sort heap (using the sortheap configuration parameter) or increasing your sort heap threshold (using the sheapthres configuration parameter). However, be aware of the implications of these options. If you increase the sort heap threshold, more memory can remain allocated for sorting, causing the paging of memory to disk. If you decrease the sort heap, you can require an extra merge phase that can slow down the sort. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Piped Sorts Rejected Percent for Interval The percentage of piped sort requests that were rejected for the DB2 instance during the monitoring interval. In the sort return phase, if the sorted information can return directly through the sort heap, it is a piped sort. However, even if the optimizer requests a piped sort, this request is rejected at run time if the total amount of sort heap memory for all sorts on the database is close to exceeding the sheapthres value. If this returned value is high compared to normal operating levels, consider decreasing your sort heap (using the sortheap configuration parameter) or increasing your sort heap threshold (using the sheapthres configuration parameter). However, be aware of the implications of these options. If you increase the sort heap threshold, more memory can remain allocated for sorting, causing the paging of memory to disk. If you decrease the sort heap, you can require an extra merge phase that can slow down the sort.

Piped Sorts Requested The number of piped sorts that have been requested. Because piped sorts might reduce disk I/O, allowing more piped sorts can improve the performance of sort operations and possibly the performance of the overall system. A piped sort is not accepted if the sort heap threshold is exceeded by allocating the requested sort heap. See the Piped Sorts Accepted attribute for more information if piped sorts are being rejected. The SQL EXPLAIN output shows whether the optimizer requested a piped sort. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Post Threshold Hash Joins The total number of times that a hash join heap request was limited due to the concurrent use of shared or private sort heap space. If this value is large (for example, greater than 5% of Hash Join Overflows), you must consider increasing the sort heap threshold. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Post Threshold OLAP Funcs The number of OLAP functions that have requested a sort heap after the sort heap threshold has been exceeded. If the value of this attribute is high, increase the sort heap threshold (sheapthres). The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Post Threshold Sorts The number of sorts that have requested heaps after reaching the sort heap threshold. By modifying the sort heap threshold and sort heap size configuration parameters, you can improve the performance of sort operations or the overall system. If the value of this attribute is high, you can do one of the following actions:
- Increase the sort heap threshold (sheapthres).
- Adjust applications to use fewer or smaller sorts by using SQL query changes.

The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Priority of Agents The current value of the priority of agents. This value is the priority that the operating system scheduler gives to agent and other database manager instance processes and threads. This priority determines how the operating system gives CPU time to the DB2 processes and threads relative to the other processes and threads running on the system. A value of -1 indicates that no special action is taken and the operating system schedules the database manager in the normal way that it schedules all processes and threads. Any other value indicates that the database manager creates its processes and threads with a static priority set to this value.

Product Version The product and version that is running on the DB2 instance. Valid format is PPPVVRRM, where the following measures apply:

Value	Description
PPP	SQL
VV	2-digit version number (with high-order 0 in the case of a 1-digit version)

Value	Description
RR	2-digit release number (with high-order 0 in the case of a 1-digit release)
M	1-digit modification level

Query Heap Size The maximum amount of memory that can be allocated for the query heap within the DB2 instance during the monitoring interval. Use a query heap to store each query in the private memory of the agent. Use the results from the aslheapsz attribute to refine the query heap size.

RB Free The number of request blocks that are free in the partitioned database server during the monitoring interval. Use the returned value with the fcm_num_rqb attribute to determine the current request block utilization. You can use this information to refine the fcm_num_rqb attribute. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

RB Free Bottom The minimum number of free request blocks. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

RB Max Used Percent The percentage of maximum FCM request blocks used during processing within the partitioned database server. If the percentage of maximum FCM request blocks used is high compared to normal operating levels, you can adjust the fcm_num_rqb attribute.

RB Used Percent The percentage of FCM request blocks used within the partitioned database server during the monitoring interval. If the percentage of FCM request blocks used is high compared to normal operating levels, you can adjust the fcm_num_rqb attribute.

Remote Connections The current number of connections initiated from remote clients to the instance of the database manager that is being monitored. This attribute shows the number of connections from remote clients to databases in this instance. This value changes frequently. As a result, you must sample the data at specific intervals over an extended period of time to get a realistic view of system usage. This number does not include applications that were initiated from the same instance as the database manager. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Remote Connections Executing The number of remote applications currently connected to a database and currently processing a unit of work within the database manager instance being monitored. By using this number, you can determine the level of concurrent processing occurring on the database manager. This value changes frequently. As a result, you must sample the data at specific intervals over an extended period of time to get a realistic view of system usage. This number does not include applications that were initiated from the same instance as the database manager. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Req IO Blk The current value (in byte units) of the client input and output block size. This value is the amount of memory that is allocated for the communication buffer between remote applications and their database agents on the database server. When a database client requests a connection to a remote database, this communication buffer is allocated on the client. On the database server, a communication buffer of 32767 bytes is initially allocated, until a connection is established and the server can determine the value of the rqrioblk attribute at the client. In addition to this communication buffer, this parameter is also used to determine the input and output block size at the database client when a blocking cursor is opened.

Snapshot Timestamp The date and time when the database system monitor information was collected. Use this attribute to help relate data chronologically if you are saving the results in a file or database for ongoing analysis.

Sort Heap Allocated The total number of allocated pages of sort heap space for all sorts at the level chosen (database manager or database) and at the time the snapshot was taken. Memory estimates do not include sort heap space. If excessive sorting occurs, add the extra memory (used for the sort heap) to the base memory requirements for running the database manager. Generally, the larger the sort heap, the more efficient the sort. Appropriate use of indexes can reduce the amount of sorting required. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Sort Heap Thres The current value (in units of 4-KB pages) of the sort heap threshold. This value is the maximum amount of memory that the database manager allocates for piped sorts. Piped sorts perform better than non-piped sorts and are used more often. However, their use can affect the performance. The value of the sheapthres attribute must be at least two times the largest sort heap that is defined for any database within the instance. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Sort Heap Used Percent The percentage of the allocated sort heap that the DB2 instance used during the monitoring interval.

Total Buffers Rcvd The total number of FCM buffers received by the database node where this monitor runs. The database node is specified in the DB2_node_number variable. Use the returned value to measure the level of traffic between the node where this monitor runs and another node. If the total number of FCM buffers received from the other node is high compared to normal operating levels, you can redistribute the database or move tables to reduce the internode traffic. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Total Buffers Sent The total number of FCM buffers that are sent from the database node where this monitor runs to the specified node. Use the returned value to measure the level of traffic between the current node where this monitor runs and the specified node. If the total number of FCM buffers sent to the other node is high compared to normal operating levels, you can redistribute the database or move tables to reduce the internode traffic. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

System Overview (KUDINFO00) attributes (Superseded)

Use the System Overview attributes to monitor general information about the DB2 UDB subsystem in your environment. The following traits are included:

- Agent-related high-water marks
- Local and remote connections
- Sorting
- Start time
- Status

This attribute group is superseded. There is a new attribute group with the same name that replaces it.

Agents Created Empty Pool The number of agents created because the agent pool was empty. It includes the number of agents started at DB2 start up. By using the Agents Assigned From Pool attribute, you can calculate the ratio of the Agents Created Empty Pool attribute to the Agents From Pool attribute. See the Agents From Pool attribute for information about using this attribute.

Agents Created Empty Pool Ratio The percentage of agents that are created because the pool is empty. This ratio is calculated by dividing the value of the Agents Created Empty Pool attribute by the value of the Agents From Pool attribute. Use this attribute to evaluate how often an agent must be created because the pool is empty.

Agents from Pool The number of agents assigned from the pool. Use this attribute with Agents Created Empty Pool attribute to determine how often an agent must be created because the pool is empty.

Agents Registered The number of agents registered in the database manager instance that is being monitored (coordinator agents and subagents). Use this attribute to help you evaluate your setting for the MAXAGENTS configuration parameter.

Agents Registered Top The maximum number of agents that the database manager has ever registered, at the same time, since it was started (coordinator agents and subagents). Use this attribute to evaluate the setting of the MAXAGENTS configuration parameter. The number of agents registered at the time the snapshot was taken is recorded by the Agents Registered attribute.

Agents Stolen The number of times that agents are stolen from an application. Agents are stolen when an idle agent associated with an application is reassigned to work on a different application. Use this attribute with the Maximum Number of Associated Agents attribute to evaluate the load that this application places on the system

Agents Waiting on Token The number of agents waiting for a token so they can run a transaction in the database manager. Use this attribute to evaluate your setting for the MAXCAGENTS configuration parameter. Each application has a dedicated coordinator agent to process database requests (transactions) within the database manager. Each agent must have a token to run a transaction. The maximum number of coordinator agents is limited by the MAXCAGENTS configuration parameter.

Agents Waiting on Token Pct The percentage of agents waiting on a token. The percentage is calculated by dividing the value of the Agents Waiting on Token attribute by the number of local applications that are currently connected to a database (Local Cons attribute). Use this attribute to assess the number of agents

Agents Waiting Top The highest number of agents waiting on a token, at the same time, since the database manager was started. Use this attribute to evaluate the setting of the MAXCAGENTS configuration parameter. In contrast, the Agents Waiting on Token attribute records the number of agents waiting for a token at the time the snapshot was taken.

Appl Support Layer Heap Size The current value (in units of 4-KB pages) of the application support layer heap size. This value is the amount of memory that is allocated for the application support layer heap. This heap is used as a communication buffer between the local application and its associated agent. In addition, this value is used to determine the input and output block size when a blocking cursor is opened.

Buff Free The number of Fast Communication Manager (FCM) buffers that are free in the partitioned database server during the monitoring interval. Use the returned value to determine the current buffer pool utilization. Use this information to refine the configuration of the number of FCM buffers.

Buff Free Bottom The minimum number of free connection entries.

Buff Max Used Pct The percentage of maximum FCM buffers used during processing within the partitioned database server. If the percentage of maximum FCM buffers used is high compared to normal operating levels, you can increase the number of FCM buffers; if the percentage is low compared to normal operating levels, you can decrease the value.

Buff Used Pct The percentage of FCM buffers that are used within the partitioned database server during the monitoring interval. If the percentage of FCM buffers used is high compared to normal operating levels, you can adjust the number of FCM buffers.

CE Free The number of connection entries that are free in the partitioned database server during the monitoring interval. Use the returned value to help determine the current connection entry utilization.

CE Free Bottom The minimum number of free connection entries.

CE Max Used Pct The maximum percentage of FCM connection entries used during processing within the partitioned database server. If the percentage of maximum FCM connection entries used is high compared to normal operating levels, you can increase the number of FCM connections; if the percentage is low compared to normal operating levels, you can decrease the value.

CE Used Pct The percentage of FCM connection entries used during processing within the partitioned database server. If the percentage of FCM connection entries used is high compared to normal operating levels, you can increase the number of FCM connections; if the percentage is low compared to normal operating levels, you can decrease the value.

Comm Private Mem The amount (in bytes) of private memory that the instance of the database manager currently has committed at the time of the snapshot. Use this attribute to assess the MIN_PRIV_MEM configuration parameter to ensure that enough private memory is available. This attribute is returned for all platforms, but tuning can be accomplished only on platforms where DB2 uses threads (such as OS/2 and Windows NT systems). Values that are greater than or equal to 2147483647 are indicated with the `Value Exceeds Maximum` text in the portal, and values that are smaller than -2147483648 are indicated with the `Value Exceeds Minimum` text. The following values are valid:

External value	Internal value
Value Exceeds Maximum	2147483647
Value Exceeds Minimum	-2147483648

Comm Private Mem (KB) The amount (in KB) of private memory that the instance of the database manager currently has committed at the time of the snapshot. The value format is an integer. Use this attribute to assess the MIN_PRIV_MEM configuration parameter to ensure that enough private memory is available. This attribute is returned for all platforms, but tuning can be accomplished only on platforms where DB2 uses threads (such as OS/2 and Windows NT systems). Values that are greater than or equal to 2147483647 are indicated with the `Value Exceeds Maximum` text in the portal, and values that are smaller than -2147483648 are indicated with the `Value Exceeds Minimum` text. The following values are valid:

External value	Internal value
Value Exceeds Maximum	2147483647
Value Exceeds Minimum	-2147483648

Conn Local Databases The number of local databases with current connections to the monitored DB2 instance. This value gives an indication of how many database information records to expect when gathering data at the database level. The applications can be running locally or remotely, and might or might not be executing a unit of work within the database manager.

Connection Status The status of the communication connection between the database partition specified by the DB2 Node Number variable and the database partition where this monitor runs. Two nodes can be active, but the connection between them remains inactive unless there is active communication between them.

Cons in Exec Pct The percentage of the maximum number of applications allowed that are connected to a database and processing a unit of work during the monitoring interval.

Coord Agents Top The maximum number of coordinating agents working at one time. The MAXCAGENTS configuration parameter determines the number of coordinating agents that can be executing concurrently. If the peak number of coordinating agents results in a workload that is too high for this node, you can reduce the MAXCAGENTS configuration parameter.

DB2 Avail The amount of time (in seconds) the instance has been available since a DB2START command was issued. The value format is an integer. The value is derived through this formula:

```
snapshot time - db2start time
```

DB2 Start Time The string date and time that the database manager was started using the DB2START command. The valid format is CYYMMDDHHMMSSmmm, where the following measures apply:

Value	Description
C	Century (0 for 20th, 1 for 21st)
YY	Year
MM	Month
DD	Day
HH	Hour
MM	Minute
SS	Second
mmm	Millisecond

Use this attribute with the snapshot time attribute to calculate the elapsed time from the start of the database manager until the snapshot was taken.

DB2 Start Timestamp The date and time that the database manager was started using the DB2START command. The following special values cause the Tivoli Enterprise Portal to display fixed string messages:

Internal value	Tivoli Enterprise Portal display
0000000000000001	N/A (Not Applicable or Not Available)
0000000000000002	N/C (Not Calculated)
0000000000000003	N/P (Not Present)

DB2 Status The current status of the DB2 instance. The following table shows the valid values:

External value	Internal value	Description
Active	Active	The DB2 instance is currently active.
Inactive/Busy	Inactive/Busy	The DB2 instance is either not running or is too busy to respond.
Quiesce Pending	Quiesce_Pending	A quiesce is pending for the DB2 instance.
Quiesced	Quiesced	The DB2 instance is currently quiesced.

External value	Internal value	Description
Unknown	Unknown	The DB2 instance is active, but the state is not recognized.

Use this attribute to determine the state of your database manager instance.

DB Partition The DB2 database partition node number, which can range from 0 to 999. The Aggregated and Current Partition values can be used within a query or situation filter. If you do not specify a db partition filter, data is returned for either the current database partition (single partition environment) or the aggregated database partitions (multiple partition environment). If a db partition filter is set to Aggregated, only aggregated partition data is returned. Historical data collection includes both aggregated and individual partition attribute data. In addition to numeric partition numbers in the 0 to 999 range, the following values are also valid:

External value	Internal value
Aggregated	-1
Current Partition	-2
All	-3

DBPG Node Status The list of failing local nodes. This is a list of integers, where each integer represents the failed local nodes. Depending on the actual partitions defined in the database partition group, it is not necessarily true that all the nodes defined in the parallel environment are examined. To ensure that all the nodes in the partitioned environment are examined, define a partition group that contains at least one database partition from each of the nodes in the partitioned environment.

FCM Num Anchors The number of FCM message anchors for the DB2 instance during the monitoring interval. Agents use the message anchors to send messages among themselves.

FCM Num Buffers The number of buffers that are used for internal communications (messages) among the nodes and within the nodes in a DB2 instance during the monitoring interval. You might need to increase the value of this parameter if you have either of the following conditions: multiple logical nodes on a processor, or too many users, nodes, or complex applications that exceed the buffer limit.

FCM Num Connect The number of FCM connection entries for the DB2 instance during the monitoring interval. Agents use connection entries to pass data among themselves. Use the results from the fcm_num_rqb attribute to help you refine the fcm_num_connect attribute.

FCM Num rqb The number of FCM request blocks for the DB2 instance during the monitoring interval. Request blocks are the media through which information is passed between the FCM daemon and an agent. The requirement for request blocks varies according to the number of users on the system, the number of database partition servers in the system, and the complexity of queries that are run.

GW Cons Wait Client For host databases being handled by the DB2 Connect gateway, the current number of connections that are waiting for the client to send a request. Because this value can change frequently, take samples at regular intervals over an extended period to obtain a realistic view of gateway usage.

GW Cons Wait Host For host databases being handled by the DB2 Connect gateway, the current number of connections that are waiting for a reply from the host. Because this value can change frequently, take samples at regular intervals over an extended period to obtain a realistic view of gateway usage.

GW Cur Cons The current number of connections to host databases being handled by the DB2 Connect gateway. Use this attribute to help you understand the level of activity at the DB2 Connect gateway and the associated use of system resources.

GW Total Cons The total number of connections attempted from the DB2 Connect gateway since the last db2start command or the last reset. Use this attribute to help you understand the level of activity at the DB2 Connect gateway and the associated use of system resources.

Idle Agents The number of agents in the agent pool that are currently unassigned to an application. Use this attribute to set the NUM_POOLAGENTS configuration parameter. By having idle agents available to satisfy requests for agents, you can improve performance.

Instance Name The name of the monitored DB2 instance. The valid format is a text string with a maximum of 20 characters.

Instance Name (Unicode) The name of the monitored DB2 instance (Unicode). The valid format is a text string with a maximum of 60 bytes.

Last Reset The string date and time that the monitor counters were reset for the application requesting the snapshot. The valid format is CYYMMDDHHMMSSmmm, where the following measures apply:

Value	Description
C	Century (0 for 20th, 1 for 21st)
YY	Year
MM	Month
DD	Day
HH	Hour
MM	Minute
SS	Second
mmm	Millisecond

Use this attribute to help you determine the scope of information returned by the database system monitor.

Last Reset Timestamp The date and time that the monitor counters were reset for the application requesting the snapshot. The following special values cause the Tivoli Enterprise Portal to display fixed string messages:

Internal value	Tivoli Enterprise Portal display
0000000000000001	N/A (Not Applicable or Not Available)
0000000000000002	N/C (Not Calculated)
0000000000000003	N/P (Not Present)

Local Cons The number of local applications that are currently connected to a database within the database manager instance being monitored. By using this number, you can determine the level of concurrent processing occurring in the database manager. This value changes frequently. As a result, you must sample the data at specific intervals over an extended period of time to get a realistic view of system usage. This number includes only applications that were initiated from the same instance as the database manager. The applications are connected, but might or might not be executing a unit of work in the database.

Local Cons in Exec The number of local applications that are currently connected to a database within the database manager instance being monitored and are currently processing a unit of work. By using this number, you can determine the level of concurrent processing occurring in the database manager. This value changes frequently. As a result, you must sample the data at specific intervals over an extended period of time to get a realistic view of system usage. This number includes only applications that were initiated from the same instance as the database manager.

MA Free Bottom The minimum number of free message anchors.

MA Max Used Pct The maximum number of message anchors used as a percentage.

Max Agent Overflows The number of attempts to create a new agent when the MAXAGENTS configuration parameter had already been reached. If requests to create new agents are received after reaching the MAXAGENTS configuration parameter, the workload for this node might be too high.

Max Agents The current value of the maximum number of existing agents. This value is the maximum number of database manager agents available at any given time to accept application requests. This value limits the total number of applications that can connect to all databases in the DB2 instance at a given time. The value of the maxagents attribute must be the sum of the values of the maxappls attribute in each database that is allowed to be accessed concurrently. Increasing the value of the maxagents attribute can increase resource use because resources for each agent are allocated when the DB2 instance is started.

Max Conc Agents The maximum number of database manager coordinator agents that can concurrently run a database manager transaction in the DB2 instance during the monitoring interval. When this monitor is used with DB2 Universal database servers, the maxcagents value is the default. This value is the maximum number of database manager agents that can be concurrently executing a database manager transaction. Use the maxcagents attributes to control the load on the system during periods of high simultaneous application activity. A value of -1 indicates that the limit is equal to the maximum number of agents (the MAXAGENTS parameter). The maxcagents parameter does not limit the number of applications that can have connections to the database.

Max Coord Agents The maximum number of database manager coordinating agents that can exist on a server in a partitioned or nonpartitioned database environment. One coordinating agent is acquired for each local or remote application that connects to a database or attaches to an instance. Requests that require an instance attachment include CREATE DATABASE, DROP DATABASE, and Database System Monitor commands.

Mon Heap Size The current value (in units of 4-KB pages) of the database system monitor heap size. This value is the amount of memory that is allocated for database system monitor data. A value of zero prevents the database manager from collecting database system monitor data.

Node Name For new installations of version 6, release 2, the format is instanceid:hostname:UD for all operating systems. The format for version 6, release 1 of the DB2 agent on Windows systems is instanceid:hostname:UD; on UNIX and Linux systems, the format is instanceid:hostname.

Piped Sort Hit Ratio Pct for Interval The piped sort hit ratio (as a percentage) for the last monitoring interval. The piped sort hit ratio is the ratio of piped sorts accepted to piped sorts requested.

Piped Sorts Accepted The number of piped sorts that have been accepted. When the number of accepted piped sorts is low compared to the number requested, you can improve sort performance by adjusting one or both of the following configuration parameters:

- SORTHEAP
- SHEAPTHRES

If piped sorts are being rejected, consider decreasing your sort heap or increasing your sort heap threshold. Be aware of the possible implications of these options:

- If you increase the sort heap threshold, more memory might remain allocated for sorting. This can cause the paging of memory to disk.
- If you decrease the sort heap, an extra merge phase (which can slow down the sort) might be required.

Piped Sorts Accepted Pct The percentage of piped sorts that have been accepted. The percentage is calculated by dividing the value of the Piped Sorts Accepted attribute by the value of the Piped Sorts Requested attribute. Use this attribute to determine whether the value of the Piped Sorts Accepted attribute is in an acceptable range.

Piped Sorts Rejected for Interval The total number of piped sorts that were rejected during the monitoring interval. In the return phase of sorting, if the sorted information can return directly through the sort heap, it is a piped sort. However, even if the optimizer requests a piped sort, this request is rejected at run time if the total amount of sort heap memory for all sorts on the database is close to exceeding the sheapthres value. If this returned value is high compared to the total number of sorts requested, consider decreasing your sort heap (using the sortheap configuration parameter) or increasing your sort heap threshold (using the sheapthres configuration parameter). However, be aware of the implications of these options. If you increase the sort heap threshold, more memory can remain allocated for sorting, causing the paging of memory to disk. If you decrease the sort heap, you can require an extra merge phase that can slow down the sort.

Piped Sorts Rejected Pct for Interval The percentage of piped sort requests that were rejected for the DB2 instance during the monitoring interval. In the sort return phase, if the sorted information can return directly through the sort heap, it is a piped sort. However, even if the optimizer requests a piped sort, this request is rejected at run time if the total amount of sort heap memory for all sorts on the database is close to exceeding the sheapthres value. If this returned value is high compared to normal operating levels, consider decreasing your sort heap (using the sortheap configuration parameter) or increasing your sort heap threshold (using the sheapthres configuration parameter). However, be aware of the implications of these options. If you increase the sort heap threshold, more memory can remain allocated for sorting, causing the paging of memory to disk. If you decrease the sort heap, you can require an extra merge phase that can slow down the sort.

Piped Sorts Requested The number of piped sorts that have been requested. Because piped sorts might reduce disk I/O, allowing more piped sorts can improve the performance of sort operations and possibly the performance of the overall system. A piped sort is not accepted if the sort heap threshold is exceeded by allocating the requested sort heap. See the Piped Sorts Accepted attribute for more information if piped sorts are being rejected. The SQL EXPLAIN output shows whether the optimizer requested a piped sort.

Post Threshold Hash Joins The total number of times that a hash join heap request was limited due to the concurrent use of shared or private sort heap space. If this value is large (for example, greater than 5% of Hash Join Overflows), you must consider increasing the sort heap threshold.

Post Threshold Sorts The number of sorts that have requested heaps after reaching the sort heap threshold. By modifying the sort heap threshold and sort heap size configuration parameters, you can improve the performance of sort operations or the overall system. If the value of this attribute is high, you can do one of the following actions:

- Increase the sort heap threshold (sheapthres).
- Adjust applications to use fewer or smaller sorts by using SQL query changes.

Prdid The product and version that is running on the DB2 instance. Valid format is PPPVVRRM, where the following measures apply:

Value	Description
PPP	SQL
VV	2-digit version number (with high-order 0 in the case of a 1-digit version)
RR	2-digit release number (with high-order 0 in the case of a 1-digit release)
M	1-digit modification level

Priority of Agents The current value of the priority of agents. This value is the priority that the operating system scheduler gives to agent and other database manager instance processes and threads. This priority determines how the operating system gives CPU time to the DB2 processes and threads relative to the other processes and threads running on the system. A value of -1 indicates that no special action is taken and the operating system schedules the database manager in the normal way that it schedules all processes and threads. Any other value indicates that the database manager creates its processes and threads with a static priority set to this value.

Query Heap Size The maximum amount of memory (in units of 4-KB pages) that can be allocated for the query heap within the DB2 instance during the monitoring interval. Use a query heap to store each query in the private memory of the agent. Use the results from the aslheapsz attribute to refine the query heap size.

RB Free The number of request blocks that are free in the partitioned database server during the monitoring interval. Use the returned value with the fcm_num_rqb attribute to determine the current request block utilization. You can use this information to refine the fcm_num_rqb attribute.

RB Free Bottom The minimum number of free request blocks.

RB Max Used Pct The percentage of maximum FCM request blocks used during processing within the partitioned database server. If the percentage of maximum FCM request blocks used is high compared to normal operating levels, you can adjust the fcm_num_rqb attribute.

RB Used Pct The percentage of FCM request blocks used within the partitioned database server during the monitoring interval. If the percentage of FCM request blocks used is high compared to normal operating levels, you can adjust the fcm_num_rqb attribute.

Rem Cons in The current number of connections initiated from remote clients to the instance of the database manager that is being monitored. This attribute shows the number of connections from remote clients to databases in this instance. This value changes frequently. As a result, you must sample the data at specific intervals over an extended period of time to get a realistic view of system usage. This number does not include applications that were initiated from the same instance as the database manager.

Rem Cons in Exec The number of remote applications currently connected to a database and currently processing a unit of work within the database manager instance being monitored. By using this number, you can determine the level of concurrent processing occurring on the database manager. This value changes frequently. As a result, you must sample the data at specific intervals over an extended period of time to get a realistic view of system usage. This number does not include applications that were initiated from the same instance as the database manager.

Req IO Blk The current value (in byte units) of the client input and output block size. This value is the amount of memory that is allocated for the communication buffer between remote applications and their database agents on the database server. When a database client requests a connection to a remote database, this communication buffer is allocated on the client. On the database server, a communication buffer of 32767 bytes is initially allocated, until a connection is established and the server can determine the value of the rqrioblk attribute at the client. In addition to this communication buffer, this parameter is also used to determine the input and output block size at the database client when a blocking cursor is opened.

Server DB2 Type The type of database manager being monitored. The following values are valid:

External value	Internal value
Client with local databases	Client_with_local_databases
Client/server	Client/server
Host Database Server	Host_Database_Server
MPP	MPP
Requestor	Requestor

External value	Internal value
Satellite	Satellite
Standalone	Standalone
UNKNOWN	UNKNOWN

Snapshot Time The string date and time when the database system monitor information was collected. Use this attribute to help relate data chronologically if you are saving the results in a file or database for ongoing analysis. The Tivoli Enterprise Portal formats the timestamp value into a date and time string. The internal timestamp value stored in the database is in the format cYYMMDDhhmmss000, which is described in the following table:

Value	Description
C	Century (0 for 20th, 1 for 21st)
YY	Year (last two digits of the year, for example (00 - 99)
MM	Month (01 for January, 02 for February, and so on)
DD	Day (the day of the month, for example 01 - 31)
HH	Hour (the hour of the day in 24-hour format from 00 - 23)
MM	Minute (the minutes of the hour from 00 - 59)
SS	Second (the seconds of the hour (00 - 59)

Snapshot Timestamp The string date and time when the database system monitor information was collected. Use this attribute to help relate data chronologically if you are saving the results in a file or database for ongoing analysis. The Tivoli Enterprise Portal formats the timestamp value into a date and time string. The internal timestamp value stored in the database is in the format cYYMMDDhhmmss000, which is described in the following table:

Value	Description
C	Century (0 for 20th, 1 for 21st)
YY	Year (last two digits of the year, for example (00 - 99)
MM	Month (01 for January, 02 for February, and so on)
DD	Day (the day of the month, for example 01 - 31)
HH	Hour (the hour of the day in 24-hour format from 00 - 23)
MM	Minute (the minutes of the hour from 00 - 59)
SS	Second (the seconds of the hour (00 - 59)

The following special values cause the Tivoli Enterprise Portal to display fixed string messages:

Internal value	Tivoli Enterprise Portal display
0000000000000001	N/A (Not Applicable or Not Available)
0000000000000002	N/C (Not Calculated)
0000000000000003	N/P (Not Present)

Sort Heap Allocated The total number of allocated pages of sort heap space for all sorts at the level chosen (database manager or database) and at the time the snapshot was taken. Memory estimates do not include sort heap space. If excessive sorting occurs, add the extra memory (used for the sort heap) to the base memory requirements for running the database manager. Generally, the larger the sort heap, the more efficient the sort. Appropriate use of indexes can reduce the amount of sorting required.

Sort Heap Thres The current value (in units of 4-KB pages) of the sort heap threshold. This value is the maximum amount of memory that the database manager allocates for piped sorts. Piped sorts perform better than non-piped sorts and are used more often. However, their use can affect the performance. The value of the Sort Heap Thres attribute must be at least two times the largest sort heap that is defined for any database within the instance.

Sort Heap Used Pct The percentage of the allocated sort heap that the DB2 instance used during the monitoring interval.

Total Buffers Rcvd The total number of FCM buffers received by the database node where this monitor runs. The database node is specified in the DB2_node_number variable. Use the returned value to measure the level of traffic between the node where this monitor runs and another node. If the total number of FCM buffers received from the other node is high compared to normal operating levels, you can redistribute the database or move tables to reduce the internode traffic.

Total Buffers Sent The total number of FCM buffers sent from the database node where this monitor runs to the specified node. Use the returned value to measure the level of traffic between the current node where this monitor runs and the specified node. If the total number of FCM buffers sent to the other node is high compared to normal operating levels, you can redistribute the database or move tables to reduce the internode traffic.

Version The version of the server that is returning the data. For example: 6.1 or 7.1. The data structures used by the monitor might change between releases. As a result, check the version of the data stream to determine whether your applications can process the data. The following values are valid:

External value	Internal value
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
UNKNOWN	UNKNOWN

System Resources (KUD_DB2_System_Resources) attributes

The System Resources attribute group provides statistics about the OS environment in which the DB2 instance is running. This attribute group is only available for DB2 Version 9.5 and later.

Free Physical Memory The amount of free physical memory on the system. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Free Physical Memory (Superseded) The amount of free physical memory on the system.

Free Swap Memory The total amount of the free swap memory on the system. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Free Swap Memory (Superseded) The total amount of the free swap memory on the system.

Free Virtual Memory The total amount of the free virtual memory on the system. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Free Virtual Memory (Superseded) The total amount of the free virtual memory on the system.

Host Name The name of the host that owns the resources.

Node Name For new installations of version 6, release 2, the format is instanceid:hostname:UD for all operating systems. The format for version 6, release 1 of the DB2 agent on Windows systems is instanceid:hostname:UD; on UNIX and Linux systems, the format is instanceid:hostname.

Operating System Name The full name of the operating system.

OS Version The version number of the operating system.

OS Release The release of the operating system. For example, AIX: 4.3 release = 3.

OS Level The maintenance level of the current version and release. For example, LINUX: 2.4.9, level = 9.

Machine Identification The machine hardware identification.

Pct of CPU Used The percentage of the CPU that is used on the system.

Pct of Physical Memory Used The percentage of the physical memory that is used on the system.

Pct of Physical Memory Used (Superseded) The percentage of the physical memory that is used on the system.

Pct of Swap Memory Used The percentage of the swap memory that is used on the system.

Pct of Swap Memory Used (Superseded) The percentage of the swap memory that is used on the system.

Pct of Virtual Memory Used The percentage of the virtual memory that is used on the system.

Pct of Virtual Memory Used (Superseded) The percentage of the virtual memory that is used on the system.

Total Physical Memory The total amount of the physical memory on the system. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Total Physical Memory (Superseded) The total amount of the physical memory on the system.

Total Swap Memory The total amount of swap memory on the system. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Total Swap Memory (Superseded) The total amount of the swap memory on the system.

Total Virtual Memory The total amount of the virtual memory on the system. The following value is valid:

External value	Internal value
Value_Exceeds_Maximum	9223372036854775807

Total Virtual Memory (Superseded) The total amount of the virtual memory on the system.

Table (KUD_DB2_Table) attributes

The Table attribute group provides information to monitor table-specific attributes, such as row read and row write rates.

DB Name The database name. Data collection performance can be improved with the use of a situation or query filter using this attribute. When specifying a filter with a distinct database name or list of database names, the agent data collector filters the return data prior to data transmission to the Tivoli Enterprise Monitoring Server and Tivoli Enterprise Portal Server.

DB Partition The DB2 database partition node number, which can range from 0 to 999. The `Aggregated` and `Current Partition` values can be used within a query or situation filter. When specifying a filter with a distinct database partition value or list of values, the monitoring agent returns data for the requested partitions. If you do not specify a db partition filter value or if you specify a db partition filter value that is not valid, the return data is for the current partition. If a db partition filter is set to `Aggregated`, only aggregated partition data is returned. Historical data collection includes both aggregated and individual partition attribute data. The following values are valid:

External value	Internal value
Aggregated	-1
Current Partition	-2
All	-3

Instance Name The name of the monitored DB2 instance. The valid format is a text string with a maximum of 60 bytes.

Node Name For new installations of version 6, release 2, the format is instanceid:hostname:UD for all operating systems. The format for version 6, release 1 of the DB2 agent on Windows systems is instanceid:hostname:UD; on UNIX and Linux systems, the format is instanceid:hostname.

Reorg Needed Indicates whether the table, its indexes, or both need to be reorganized, and is calculated using DB2 monitoring data that is generated when the DB2 RUNSTATS utility is run.

Important: The RUNSTATS utility collects statistics on tables and indexes, and can affect system performance as it is collecting the statistics. For this reason, RUNSTATS is not automatically run.

The following values are valid:

External value	Internal value	Description
None	NN	Reorganization is not needed.
Table Data	YN	Reorganization of the table data is needed.
Table Index	NY	Reorganization of the table indexes is needed.
Table Data and Table Index	YY	Reorganization of the table data and indexes is needed.

Rows Read Rate for Interval The rate (per second) at which rows were read from the table during the monitoring interval.

Rows Write Rate for Interval The rate (per second) at which rows were changed (inserted, deleted, or updated) in the table during the monitoring interval.

Snapshot Timestamp The date and time when the database system monitor information was collected. Use this attribute to help relate data chronologically if you are saving the results in a file or database for ongoing analysis.

Table Name The name of the table for which the information is collected. The value format is a text string with a maximum of 60 bytes. Data collection performance can be improved with the use of a situation or query filter using this attribute. When specifying a filter with a distinct table name or list of table names, the agent data collector filters the return data prior to data transmission to the Tivoli Enterprise Monitoring Server and Tivoli Enterprise Portal Server.

Table Schema The schema of the table for which the information is collected. The value format is a text string with a maximum of 60 bytes.

Tablespace (KUD_DB2_Tablespace) attributes

The Tablespace attributes provide information to monitor page size and usage characteristics for a tablespace.

Avg Direct Read Time The time (in milliseconds) for performing the direct reads for the tablespace. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Direct Write Time The time (in milliseconds) for performing the direct writes for the tablespace. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Pool I/O Time The average time (in milliseconds) for performing buffer pool input and output operations (reading or writing) for the tablespace. A high average time can indicate the existence of an input and output conflict. In this case, you might need to move data to a different device. The returned value includes the time applied to asynchronous input and output operations (which are performed by prefetchers and page cleaners). The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Pool Read Time The average time (in milliseconds) for processing read requests that caused data or index pages to be physically read from disk to buffer pool for the tablespace. A high average time generally indicates the existence of an input and output conflict. In this case, you might need to move data to a different device. The returned value includes the time applied to asynchronous read operations that are performed by prefetchers. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Pool Write Time The average time (in milliseconds) for processing write requests that caused data or index pages to be physically written from buffer pool to disk for the tablespace. A high average time generally indicates the existence of an input and output conflict. In this case, you might need to move data to a different device. The returned value includes the time applied to asynchronous write operations that are performed by page cleaners. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Sectors Read The average number of sectors that are read for this tablespace for each direct read. Direct reads do not use the buffer pool, and so result in poor performance because the data is physically read from disk each time. If you are using system monitors to track input and output for the device, this returned value helps you distinguish database input and output from non-database input and output. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Sectors Written The average number of sectors that are written for this tablespace for each direct read. Direct writes do not use the buffer pool, and so result in poor performance because the data is physically written from disk each time. If you are using system monitors to track input and output for the device, this returned value helps you distinguish database input and output from non-database input and output. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Sync Data Read Time The average time (in milliseconds) for synchronous data reads for the tablespace. Use the returned value to analyze the input and output work being performed for the tablespace. Synchronous read operations are performed by database manager agents. Asynchronous reads are performed by prefetchers, which read data pages from disk into the buffer pool in anticipation of their use. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Sync Data Write Time The average time (in milliseconds) for synchronous data writes for the tablespace. Use the returned value to analyze the input and output work being performed for the tablespace. Synchronous write operations are performed by database manager agents. Asynchronous writes are performed by page cleaners, which write out changed pages to disk and free up space in the buffer pool. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Avg Sync I/O Time The average time (in milliseconds) for synchronous input and output operations for the tablespace. Use the returned value to analyze the input and output work being performed for the tablespace. Synchronous input and output operations are performed by database manager agents. Asynchronous input and output operations are performed by prefetchers (reads) and page cleaners (writes). In general, asynchronous input and output helps your applications run faster. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Container Name Indicates the location of the container. Value format is a simple text string with a maximum of 768 characters.

DB Name Represents the real name of the database for which information is collected or to which the application is connected. This name was given to the database when it was created. Value format is a simple text string with a maximum of 96 characters.

DB Partition The DB2 database partition node number, which can range from 0 to 999. The `Aggregated` and `Current Partition` values can be used within a query or situation filter. When specifying a filter with a distinct database partition value or list of values, the monitoring agent returns data for the requested partitions. If you do not specify a db partition filter value or if you specify a db partition filter value that is not valid, the return data is for the current partition. If a db partition filter is set to `Aggregated`, only aggregated partition data is returned. Historical data collection includes both aggregated and individual partition attribute data. The following values are valid:

External value	Internal value
Aggregated	-1
Current Partition	-2
All	-3

Direct Read Reqs The number of requests to perform a direct read from disk of one or more sectors of data for the database. The returned value is used in calculating the returned value for the average number of sectors read per direct read for the tablespace. Direct reads are performed in units, the smallest being a 512-byte sector. They are used while the system is doing any of the following operations: reading LONG VARCHAR columns, reading LOB columns, or performing a backup. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Direct Read Time The time (in milliseconds) for performing the direct reads for the tablespace since the first connection. The returned value is used in calculations for the average direct read time (ms). A high average time can indicate an input and output conflict. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Direct Reads The number of requests to perform a direct read from disk of one or more sectors of data for the tablespace since the first connection. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Direct Write Reqs The number of requests to perform a direct write to disk of one or more sectors of data for the database. The returned value is used in calculating the returned value for the average number of sectors written per direct write. Direct writes are performed in units, the smallest being a 512-byte sector. They are used while the system is doing any of the following operations: writing LONG VARCHAR columns, writing LOB columns, performing a restore, or performing a load. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Direct Write Time The time (in milliseconds) for performing the direct writes for the tablespace since the first connection. The returned value is used in calculations for the average direct write time (ms). A high average time can indicate an input and output conflict. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Direct Writes The number of direct writes to disk for the tablespace since the first connection. The returned value is used in calculating the returned value for the average number of sectors written per direct write. Direct writes are performed in units, the smallest being a 512-byte sector. They are used while the system is doing any of the following operations: writing LONG VARCHAR columns, writing LOB columns, performing a restore, or performing a load. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Estore Read/Write Ratio The ratio (as a percentage) of pages (data plus index) copied from extended storage to pages copied to extended storage within the tablespace. When a page is transferred from extended storage to the buffer pool, you save a system input and output call. However, you still incur the cost of attaching to the extended memory segment, copying the page, and detaching from the segment. Use the returned value to determine if you would benefit from using extended storage. The higher the ratio, the more likely you are to benefit. In general, extended storage is particularly useful if input and output activity is very high on your system.

Extent Size Indicates the extent size. A valid value is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Files Closed The total number of closed files for the tablespace since the first database connection. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Free Pages Represents the number of free pages associated with the database. A valid value is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Instance Name The name of the monitored DB2 instance. The valid format is a text string with a maximum of 60 bytes.

Node Name For new installations of version 6, release 2, the format is instanceid:hostname:UD for all operating systems. The format for version 6, release 1 of the DB2 agent on Windows systems is instanceid:hostname:UD; on UNIX and Linux systems, the format is instanceid:hostname.

Num Containers Indicates the number of containers used. A valid value is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Object ID Represents the identifier for the object. A valid value is an integer.

Page Size Indicates the page size. A valid value is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pending Free Pages Represents the number of pending free pages associated with the database. A valid value is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Async Data Read Reqs The number of asynchronous data read requests. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Async Data Reads The number of data pages read asynchronously into the buffer pool for the tablespace. Compare the returned value with number of synchronous reads to gain insight into how well the prefetchers are working. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Async Data Writes The number of times a buffer pool data page was physically written asynchronously to disk for the tablespace. Compare the returned value with the number of synchronous writes to gain insight into how well the page cleaners are working. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Async Index Reads The number of index pages read asynchronously into the buffer pool by a prefetcher within the tablespace. By comparing the ratio of asynchronous to synchronous reads, you can determine how well the prefetchers are working. Asynchronous reads are performed by database manager prefetchers. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Async Index Read Reqs The number of asynchronous index read requests. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Async Index Writes The number of times a buffer pool index page was written asynchronously to disk for the tablespace. Subtract the returned value from the buffer pool index writes to calculate the number of synchronous index writes. By comparing the number of asynchronous index writes to synchronous index writes, you can gain insight into how well the buffer pool page cleaners are performing. This ratio can be helpful when you are refining the num_iocleaners configuration parameter. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Async Read Time The total time (in milliseconds) that database manager prefetchers spent reading data into the buffer pool for the tablespace. Compare the returned value to the synchronous read time to understand where input and output time is being spent. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Async Write Time The total time (in milliseconds) that database manager page cleaners spent writing data or index pages from the buffer pool to disk for the tablespace. Compare the returned value to the synchronous write time to understand where input and output time is being spent. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Data from Estore The number of buffer pool data pages copied from extended storage within the tablespace to the buffer pool. Required pages are copied from extended storage to the buffer pool if they are not in the buffer pool but are in extended storage. This copying can incur the cost of connecting to the shared memory segment but saves the cost of a disk read. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Data L Reads The number of logical read requests for data pages that went through the buffer pool for the tablespace since the connection occurred. The returned value includes requests for data that is already in the buffer pool or read from disk into the buffer pool to fulfill the request. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Data P Reads The number of read requests requiring input and output to get data pages into the buffer pool for the tablespace since the first connection. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Data Reads The number of read requests to get data pages into the buffer pool for the tablespace. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Data to Estore The number of buffer pool data pages copied to extended storage for the tablespace. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Data Writes The number of times that a buffer pool data page was physically written to disk for the tablespace. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Hit Percent The percent buffer pool hit ratio (data plus index).

Pool Hit Ratio for Interval The overall buffer pool hit ratio (as a percentage) for the database during the monitoring interval. This hit ratio includes both index and data page activity. The overall buffer pool hit ratio indicates the percentage of page requests for which the database manager did not need to load a page from disk to service. (That is, the page was already in the buffer pool.) The greater the buffer pool hit ratio, the lower the frequency of disk input and output. If the hit ratio is low compared to normal operating levels, increasing the number of buffer pool pages can improve performance. A ratio of zero indicates that pages needed to be read for every request. For a large database, increasing the buffer pool size can have a minimal effect on the buffer pool hit ratio. Such a database can have so large a number of data pages that the statistical chance of a hit is not increased by an increase of the buffer pools. However, even though the data might be too large to fit in the buffer pool, the entire index can fit. In this case, you can refine buffer pool sizes until the overall buffer pool hit ratio stops increasing, and then refine the buffer pool until the buffer pool index hit ratio no longer increases.

Pool Index Hit Ratio Percent for Interval The percent buffer pool index hit ratio for the monitoring interval.

Pool Index from Estore The number of buffer pool index pages copied from extended storage for the tablespace. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Index L Reads The number of logical read requests for index pages that went through the buffer pool for the tablespace since the connection. The returned value includes requests for index pages that are already in the buffer pool or read from disk into the buffer pool to fulfill the request. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Index P Reads The number of physical read requests to get index pages into the buffer pool for the tablespace. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Index to Estore The number of buffer pool index pages copied to extended storage within the tablespace. Pages are copied from the buffer pool to extended storage when they are selected as victim pages. This copying is required to make space for new pages in the buffer pool. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Index Writes The number of times that a buffer pool index page was physically written to disk for the tablespace since the first connection. If the returned value is high compared to the buffer pool index physical reads, you can improve performance by increasing the available buffer pool space. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool I/O per Sec The rate (per second) for buffer pool input and output for the tablespace. Buffer pool input and output includes all physical data and index pages that go through the buffer pool when read or written. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Read Time The time (in milliseconds) spent reading data from the buffer pool to disk for the tablespace since the first connection. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Sync Data Reads The number of buffer pool synchronous reads. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Sync Data Writes The number of buffer pool synchronous writes. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Sync Index Reads The number of buffer pool synchronous index reads. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Sync Index Writes The number of buffer pool synchronous index writes. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Pool Write Time The time (in milliseconds) spent writing data from the buffer pool to disk for the tablespace since the first connection. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Prefetch Percent for Interval The percentage of asynchronous read requests that were satisfied for a tablespace during the last monitoring interval.

Prefetch Reqs for Interval The number of prefetch requests for the tablespace during the monitoring interval. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Prefetch Size Indicates the prefetch size. A valid value is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Space Used DMS Table Percent The percentage of space used in the Database Managed Space (DMS) tablespace. Use the returned value to determine if the tablespace needs more space.

Space Used SMS Table The number of bytes allocated to the System Managed Space (SMS) tablespace. Use the returned value to determine whether the number of bytes used by the SMS tablespace is excessive in relation to the file system on which the tablespace is established. Values that are greater than or equal to 9223372036854775807 are indicated with the `Value Exceeds Maximum` text in the portal. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Sync Read Time The time (in milliseconds) applied to synchronous reads for the tablespace. Compare the returned value to the buffer pool async read time to understand where input and output time for this tablespace is used. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Sync Write Time The time (in milliseconds) spent synchronously writing data to disk from the buffer pool for the tablespace. Compare the returned value to the value returned by the buffer pool async write time to understand where input and output time for this tablespace is used. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Tablespace ID Represents the identifier for the tablespace. A valid value is an integer. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Tablespace Name Represents the tablespace name of DB2. Value format is a simple text string with a maximum of 96 characters.

Tablsespace Status The status of the tablespace. DB2 sets this value, which corresponds to the DB2 tablespace_state element in the DB2 tablespace_nodeinfo Snapshot logical data group. This element contains a hexadecimal value that indicates the current tablespace state. The externally visible state of a tablespace is composed of the hexadecimal sum of certain state values. For example, if the state is `quiesced: EXCLUSIVE` and `Load pending`, the value is 0x0004 + 0x0008, which is 0x000c. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Tablespace Status Name The comma-delimited tablespace state name that corresponds to the tablespace status (TBSP STATUS) attribute. The following table shows the text that is used, depending upon the bit setting of tbsp status:

Hexadecimal Value	Decimal Value	Text
0x0	0	NORMAL
0x1	1	QUIESCED_SH
0x2	2	QUIESCED_UP
0x4	4	QUIESCED_EX
0x8	8	LOAD_PENDING
0x10	16	DELETE_PENDING
0x20	32	BACKUP_PENDING
0x40	64	RF_PROGRESS
0x80	128	RF_PENDING
0x100	256	RESTORE_PENDING
0x200	512	DISABLE_PENDING
0x400	1024	REORG_PROGRESS
0x800	2048	BACKUP_PROGRESS
0x1000	4096	STORAGE_UNDEFINED
0x2000	8192	RESTORE_PROGRESS
0x4000	16384	OFFLINE
0x8000	32768	DROP_PENDING

Hexadecimal Value	Decimal Value	Text
0x2000000	33554432	STORDEF
0x4000000	67108864	STORDEF_FINAL
0x8000000	134217728	STORDEF_CHANGED
0x10000000	268435456	DMS_REBALANCE
0x20000000	536870912	TBS_DELETE
0x40000000	1073741824	TBS_CREATE

Tablespace Type Represents the tablespace type of DB2. Value format is a simple text string with a maximum of 32 characters. The following values are valid:

External value	Internal value
System managed space	System_managed_space
Database managed space	Database_managed_space
UNKNOWN	UNKNOWN

Total Direct I/O Time The total time (in milliseconds) for direct reads to and writes from the tablespace. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Total I/O Percent The percentage total of I/O.

Total Pages Represents the number of total pages available associated with the database. A valid value is an integer. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Total Pool I/O Time The total pool I/O time. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Total Pool P Read Time The time (in milliseconds) spent reading data and index pages from the buffer pool to the disk for the tablespace since the first connection. The value format is an integer. This attribute is the same as the pool read time attribute. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Total Pool P Write Time The total pool physical write time. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Total Sync I/O The total number of synchronous reads and writes for both data and index pages for the tablespace. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Total Sync I/O Time The total time (in milliseconds) for processing requests for synchronous reads or writes within the tablespace. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Usable Pages Represents the number of usable pages associated with the database. A valid value is an integer. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Used Pages Represents the total number of used pages. A valid value is an integer. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Version Indicates the version number of DB2.

Tablespace (KUDTABSPACE) attributes (Superseded)

The Tablespace attributes provide information to monitor page size and usage characteristics for a tablespace. This attribute group is superseded. There is a new attribute group with the same name that replaces it.

Avg Direct Read Time The time (in milliseconds) for performing the direct reads for the tablespace.

Avg Direct Write Time The time (in milliseconds) for performing the direct writes for the tablespace.

Avg Pool IO Time The average time (in milliseconds) for performing buffer pool input and output operations (reading or writing) for the tablespace. A high average time can indicate the existence of an input and output conflict. In this case, you might need to move data to a different device. The returned value includes the time applied to asynchronous input and output operations (which are performed by prefetchers and page cleaners).

Avg Pool Read Time The average time (in milliseconds) for processing read requests that caused data or index pages to be physically read from disk to buffer pool for the tablespace. A high average time generally indicates the existence of an input and output conflict. In this case, you might need to move data to a different device. The returned value includes the time applied to asynchronous read operations that are performed by prefetchers.

Avg Pool Write Time The average time (in milliseconds) for processing write requests that caused data or index pages to be physically written from buffer pool to disk for the tablespace. A high average time generally indicates the existence of an input and output conflict. In this case, you might need to move data to a different device. The returned value includes the time applied to asynchronous write operations that are performed by page cleaners.

Avg Sect Read The average number of sectors that are read for this tablespace for each direct read. Direct reads do not use the buffer pool, and so result in poor performance because the data is physically read from disk each time. If you are using system monitors to track input and output for the device, this returned value helps you distinguish database input and output from non-database input and output.

Avg Sect Written The average number of sectors that are written for this tablespace for each direct read. Direct writes do not use the buffer pool, and so result in poor performance because the data is physically written from disk each time. If you are using system monitors to track input and output for the device, this returned value helps you distinguish database input and output from non-database input and output.

Avg Sync Data Read Time The average time (in milliseconds) for synchronous data reads for the tablespace. Use the returned value to analyze the input and output work being performed for the tablespace. Synchronous read operations are performed by database manager agents. Asynchronous reads are performed by prefetchers, which read data pages from disk into the buffer pool in anticipation of their use.

Avg Sync Data Write Time The average time (in milliseconds) for synchronous data writes for the tablespace. Use the returned value to analyze the input and output work being performed for the tablespace. Synchronous write operations are performed by database manager agents. Asynchronous writes are performed by page cleaners, which write out changed pages to disk and free up space in the buffer pool.

Avg Sync IO Time The average time (in milliseconds) for synchronous input and output operations for the tablespace. Use the returned value to analyze the input and output work being performed for the tablespace. Synchronous input and output operations are performed by database manager agents. Asynchronous input and output operations are performed by prefetchers (reads) and page cleaners (writes). In general, asynchronous input and output helps your applications run faster.

Container Name Indicates the location of the container. Value format is a simple text string with a maximum of 32 characters.

Container Name (Unicode) Indicates the location of the container (Unicode). Value format is a simple text string with a maximum of 768 characters.

DB Name Represents the real name of the database for which information is collected or to which the application is connected. This name was given to the database when it was created. Value format is a simple text string with a maximum of 32 characters.

DB Name (Unicode) Represents the real name of the database for which information is collected or to which the application is connected (Unicode). This name was given to the database when it was created. Value format is a simple text string with a maximum of 96 characters.

DB Partition The DB2 database partition node number, which can range from 0 to 999. The `Aggregated` and `Current Partition` values can be used within a query or situation filter. When specifying a filter with a distinct database partition value or list of values, the monitoring agent returns data for the requested partitions. If you do not specify a db partition filter value or if you specify a db partition filter value that is not valid, the return data is for the current partition. If a db partition filter is set to `Aggregated`, only aggregated partition data is returned. Historical data collection includes both aggregated and individual partition attribute data. The following values are valid:

External value	Internal value
Aggregated	-1
Current Partition	-2
All	-3

Direct Read Reqs The number of requests to perform a direct read from disk of one or more sectors of data for the database. The returned value is used in calculating the returned value for the average number of sectors read per direct read for the tablespace. Direct reads are performed in units, the smallest being a 512-byte sector. They are used while the system is doing any of the following operations: reading LONG VARCHAR columns, reading LOB columns, or performing a backup.

Direct Read Time The time (in milliseconds) for performing the direct reads for the tablespace since the first connection. The returned value is used in calculations for the average direct read time (ms). A high average time can indicate an input and output conflict.

Direct Reads The number of requests to perform a direct read from disk of one or more sectors of data for the tablespace since the first connection.

Direct Write Reqs The number of requests to perform a direct write to disk of one or more sectors of data for the database. The returned value is used in calculating the returned value for the average number of sectors written per direct write. Direct writes are performed in units, the smallest being a 512-byte sector. They are used while the system is doing any of the following operations: writing LONG VARCHAR columns, writing LOB columns, performing a restore, or performing a load.

Direct Write Time The time (in milliseconds) for performing the direct writes for the tablespace since the first connection. The returned value is used in calculations for the average direct write time (ms). A high average time can indicate an input and output conflict.

Direct Writes The number of direct writes to disk for the tablespace since the first connection. The returned value is used in calculating the returned value for the average number of sectors written per direct write. Direct writes are performed in units, the smallest being a 512-byte sector. They are used while the system is doing any of the following operations: writing LONG VARCHAR columns, writing LOB columns, performing a restore, or performing a load.

Estore RW Ratio The ratio (as a percentage) of pages (data plus index) copied from extended storage to pages copied to extended storage within the tablespace. When a page is transferred from extended storage to the buffer pool, you save a system input and output call. However, you still incur the cost of attaching to the extended memory segment, copying the page, and detaching from the segment. Use the returned value to determine if you would benefit from using extended storage. The higher the ratio, the more likely you are to benefit. In general, extended storage is particularly useful if input and output activity is very high on your system.

Extent Size Indicates the extent size. A valid value is an integer.

Files Closed The total number of closed files for the tablespace since the first database connection.

Free Pages Represents the number of free pages associated with the database. A valid value is an integer.

Instance Name (Unicode) The name of the monitored DB2 instance (Unicode). The valid format is a text string with a maximum of 60 bytes.

Node Name For new installations of version 6, release 2, the format is instanceid:hostname:UD for all operating systems. The format for version 6, release 1 of the DB2 agent on Windows systems is instanceid:hostname:UD; on UNIX and Linux systems, the format is instanceid:hostname.

Num Containers Indicates the number of containers used. A valid value is an integer.

Object ID Represents the identifier for the object. A valid value is an integer.

Page Size Indicates the page size. A valid value is an integer.

Pending Free Pages Represents the number of pending free pages associated with the database. A valid value is an integer.

Pool Async Data Read Reqs The number of asynchronous data read requests.

Pool Async Data Reads The number of data pages read asynchronously into the buffer pool for the tablespace. Compare the returned value with number of synchronous reads to gain insight into how well the prefetchers are working.

Pool Async Data Writes The number of times a buffer pool data page was physically written asynchronously to disk for the tablespace. Compare the returned value with the number of synchronous writes to gain insight into how well the page cleaners are working.

Pool Async Index Read Reqs The number of asynchronous index read requests.

Pool Async Index Reads The number of index pages read asynchronously into the buffer pool by a prefetcher within the tablespace. By comparing the ratio of asynchronous to synchronous reads, you can determine how well the prefetchers are working. Asynchronous reads are performed by database manager prefetchers.

Pool Async Index Writes The number of times a buffer pool index page was written asynchronously to disk for the tablespace. Subtract the returned value from the buffer pool index writes to calculate the number of synchronous index writes. By comparing the number of asynchronous index writes to synchronous index writes, you can gain insight into how well the buffer pool page cleaners are performing. This ratio can be helpful when you are refining the num_iocleaners configuration parameter.

Pool Async Read Time The total time (in milliseconds) that database manager prefetchers spent reading data into the buffer pool for the tablespace. Compare the returned value to the synchronous read time to understand where input and output time is being spent.

Pool Async Write Time The total time (in milliseconds) that database manager page cleaners spent writing data or index pages from the buffer pool to disk for the tablespace. Compare the returned value to the synchronous write time to understand where input and output time is being spent.

Pool Data from Estore The number of buffer pool data pages copied from extended storage within the tablespace to the buffer pool. Required pages are copied from extended storage to the buffer pool if they are not in the buffer pool but are in extended storage. This copying can incur the cost of connecting to the shared memory segment but saves the cost of a disk read.

Pool Data L Reads The number of logical read requests for data pages that went through the buffer pool for the tablespace since the connection occurred. The returned value includes requests for data that is already in the buffer pool or read from disk into the buffer pool to fulfill the request.

Pool Data P Reads The number of read requests requiring input and output to get data pages into the buffer pool for the tablespace since the first connection.

Pool Data Reads The number of read requests to get data pages into the buffer pool for the tablespace.

Pool Data to Estore The number of buffer pool data pages copied to extended storage for the tablespace.

Pool Data Writes The number of times that a buffer pool data page was physically written to disk for the tablespace.

Pool Hit Pct The percent buffer pool hit ratio (data plus index).

Pool Hit Ratio for Interval The overall buffer pool hit ratio (as a percentage) for the database during the monitoring interval. This hit ratio includes both index and data page activity. The overall buffer pool hit ratio indicates the percentage of page requests for which the database manager did not need to load a page from disk to service. (That is, the page was already in the buffer pool.) The greater the buffer pool hit ratio, the lower the frequency of disk input and output. If the hit ratio is low compared to normal operating levels, increasing the number of buffer pool pages can improve performance. A ratio of zero indicates that pages needed to be read for every request. For a large database, increasing the buffer pool size can have a minimal effect on the buffer pool hit ratio. Such a database can have so large a number of data pages that the statistical chance of a hit is not increased by an increase of the buffer pools. However, even though the data might be too large to fit in the buffer pool, the entire index can fit. In this case, you can refine buffer pool sizes until the overall buffer pool hit ratio stops increasing, and then refine the buffer pool until the buffer pool index hit ratio no longer increases.

Pool Index Hit Pct for Interval The percent buffer pool index hit ratio for the monitoring interval.

Pool Index from Estore The number of buffer pool index pages copied from extended storage for the tablespace.

Pool Index L Reads The number of logical read requests for index pages that went through the buffer pool for the tablespace since the connection. The returned value includes requests for index pages that are already in the buffer pool or read from disk into the buffer pool to fulfill the request.

Pool Index P Reads The number of physical read requests to get index pages into the buffer pool for the tablespace.

Pool Index to Estore The number of buffer pool index pages copied to extended storage within the tablespace. Pages are copied from the buffer pool to extended storage when they are selected as victim pages. This copying is required to make space for new pages in the buffer pool.

Pool Index Writes The number of times that a buffer pool index page was physically written to disk for the tablespace since the first connection. If the returned value is high compared to the buffer pool index physical reads, you can improve performance by increasing the available buffer pool space.

Pool IO per Sec The rate (per second) for buffer pool input and output for the tablespace. Buffer pool input and output includes all physical data and index pages that go through the buffer pool when read or written.

Pool Read Time The time (in milliseconds) spent reading data from the buffer pool to disk for the tablespace since the first connection.

Pool Sync Data Reads The number of buffer pool synchronous data reads.

Pool Sync Data Writes The number of buffer pool synchronous data writes.

Pool Sync Index Reads The number of buffer pool synchronous index reads.

Pool Sync Index Writes The number of buffer pool synchronous index writes.

Pool Write Time The time (in milliseconds) spent writing data from the buffer pool to disk for the tablespace since the first connection.

Prefetch Pct for Interval The percentage of asynchronous read requests that were satisfied for a tablespace during the last monitoring interval.

Prefetch Reqs for Interval The number of prefetch requests for the tablespace during the monitoring interval.

Prefetch Size Indicates the prefetch size. A valid value is an integer.

Space Used DMS Table Pct The percentage of space used in the Database Managed Space (DMS) tablespace. Use the returned value to determine if the tablespace needs more space.

Space Used SMS Table The number of bytes allocated to the System Managed Space (SMS) tablespace. Use the returned value to determine whether the number of bytes used by the SMS tablespace is excessive in relation to the file system on which the tablespace is established. Values that are greater than or equal to 2147483647 are indicated with the `Value Exceeds Maximum` text in the portal, and values that are smaller than -2147483648 are indicated with the `Value Exceeds Minimum` text. The following values are valid:

External value	Internal value
Value Exceeds Maximum	2147483647
Value Exceeds Minimum	-2147483648

Space Used SMS Table (MB) The number of MB allocated to the System Managed Space (SMS) tablespace. Use the returned value to determine whether the number of bytes used by the SMS tablespace is excessive in relation to the file system on which the tablespace is established. Values that are greater than or equal to 2147483647 are indicated with the `Value Exceeds Maximum` text in the portal, and values that are smaller than -2147483648 are indicated with the `Value Exceeds Minimum` text. The following values are valid:

External value	Internal value
Value Exceeds Maximum	2147483647
Value Exceeds Minimum	-2147483648

Sync Read Time The time (in milliseconds) applied to synchronous reads for the tablespace. Compare the returned value to the buffer pool async read time to understand where input and output time for this tablespace is used.

Sync Write Time The time (in milliseconds) spent synchronously writing data to disk from the buffer pool for the tablespace. Compare the returned value to the value returned by the buffer pool async write time attribute to understand where input and output time for this tablespace is used.

Tablespace ID Represents the identifier for the tablespace. A valid value is an integer.

Tablespace Name Represents the tablespace name of DB2. Value format is a simple text string with a maximum of 32 characters.

Tablespace Name (Unicode) Represents the tablespace name of DB2 (Unicode). Value format is a simple text string with a maximum of 768 bytes.

Tablespace Status The status of the tablespace. DB2 sets this value, which corresponds to the DB2 tablespace_state element in the DB2 tablespace_nodeinfo Snapshot logical data group. This element contains a hexadecimal value that indicates the current tablespace state. The externally visible state of a tablespace is composed of the hexadecimal sum of certain state values. For example, if the state is `quiesced: EXCLUSIVE` and `Load pending`, the value is 0x0004 + 0x0008, which is 0x000c.

Tablespace Status Name The comma-delimited tablespace state name that corresponds to the tablespace status (TBSP STATUS) attribute. The following table shows the text that is used, depending upon the bit setting of tbsp status:

Hexadecimal Value	Decimal Value	Text
0x0	0	NORMAL
0x1	1	QUIESCED_SH
0x2	2	QUIESCED_UP
0x4	4	QUIESCED_EX
0x8	8	LOAD_PENDING
0x10	16	DELETE_PENDING
0x20	32	BACKUP_PENDING
0x40	64	RF_PROGRESS
0x80	128	RF_PENDING
0x100	256	RESTORE_PENDING
0x200	512	DISABLE_PENDING
0x400	1024	REORG_PROGRESS
0x800	2048	BACKUP_PROGRESS
0x1000	4096	STORAGE_UNDEFINED
0x2000	8192	RESTORE_PROGRESS
0x4000	16384	OFFLINE
0x8000	32768	DROP_PENDING
0x2000000	33554432	STORDEF
0x4000000	67108864	STORDEF_FINAL
0x8000000	134217728	STORDEF_CHANGED
0x10000000	268435456	DMS_REBALANCE
0x20000000	536870912	TBS_DELETE
0x40000000	1073741824	TBS_CREATE

Tablespace Type Represents the tablespace type of DB2. Value format is a simple text string with a maximum of 32 characters. The following values are valid:

External value	Internal value
System managed space	System_managed_space
Database managed space	Database_managed_space
UNKNOWN	UNKNOWN

Total Direct IO Time The total time (in milliseconds) for direct reads to and writes from the tablespace.

Total IO Pct The percentage total of I/O.

Total Pages Represents the number of total pages available associated with the database. A valid value is an integer.

Total Pool IO Time The total pool I/O time.

Total Pool P Read Time The time (in milliseconds) spent reading data and index pages from the buffer pool to the disk for the tablespace since the first connection. The value format is an integer. This attribute is the same as the pool read time attribute.

Total Pool P Write Time The total pool physical write time.

Total Sync IO The total number of synchronous reads and writes for both data and index pages for the tablespace.

Total Sync IO Time The total time (in milliseconds) for processing requests for synchronous reads or writes within the tablespace.

Usable Pages Represents the number of usable pages associated with the database. A valid value is an integer.

Used Pages Represents the total number of used pages. A valid value is an integer.

Version Indicates the version number of DB2. A valid value is an integer.

DB2 Tablespace Auto-resize (KUD_Tablespace_Auto_Resize) attributes

The DB2 Tablespace Auto-resize attributes provide information about the monitoring of the automatic resizing of tablespaces.

Node Name The node name of the database.

Timestamp The local time at the agent when the data was collected.

Host Name The hostname of the machine where the DB2 database is hosted.

Instance Name The name of the monitored DB2 instance (Unicode). The valid format is a text string with a maximum of 60 bytes.

DB Name The name of the monitored database.

DB Partition The DB2 database partition node number, which can range from 0 to 999. The Aggregated and Current Partition values can be used within a query or situation filter. If a db partition filter is not specified, data is returned for the current database partition. If a db partition filter is set to Aggregated, only aggregated partition data is returned. Historical data collection includes both aggregated and individual partition attribute data. In addition to numeric partition numbers in the 0 to 999 range, the following values are also valid:

External value	Internal value
Aggregated	-1
Current Partition	-2
All	-3

TBSP Name Represents the tablespace name of DB2. Value format is a simple text string with a maximum of 96 characters.

TBSP ID Represents the identifier for the tablespace. A valid value is an integer. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Total Pages Represents the number of total pages available associated with the database. A valid value is an integer. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Usable Pages Represents the number of usable pages associated with the database. A valid value is an integer. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Used Pages Represents the total number of used pages. A valid value is an integer. The following value is valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Free Pages Represents the number of free pages associated with the database. A valid value is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Initial Size Represents the initial tablespace size in bytes associated with the database. A valid value is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Current Size Represents the current tablespace size in bytes associated with the database. A valid value is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Max Size Represents the maximum tablespace size in bytes associated with the database. A valid value is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Increase Size Represents the size in bytes of the increase of the tablespace associated with the database. A valid value is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

Increase Size Pct Represents the size as a percentage to which an auto-resize tablespace increases before reaching its maximum size and more space is required. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	2147483647

Last Resize Time Represents the time at which the last tablespace resize was successfully executed.

Last Resize Failed Indicates whether the last resize attempted succeeded.

Rebalance Mode Represents whether a forward or reverse rebalance is taking place. A valid value is an integer.

Prefetch Size Represents the maximum number of pages the prefetcher gets from the disk at a time. A valid value is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	9223372036854775807

TBSP Utilization (%) Represents the percentage of storage consumption for each DMS tablespace. . A valid value is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	2147483647

Used Allocated Pct Represents the percentage of the allocated size that has been used. A valid value is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	2147483647

Used Max Pct Represents the percentage of the maximum size that has been used. A valid value is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	2147483647

Used Disk Pct Represents the percentage of the disk size that has been used. A valid value is an integer. The following value is also valid:

External value	Internal value
Value Exceeds Maximum	2147483647

Auto Resize Enabled Indicates whether automatic resizing is enabled for the tablespace

Using Auto Storage Indicates whether the tablespace was created as an automatic storage tablespace.

Disk capacity planning for historical data

Disk capacity planning for a monitoring agent is a prediction of the amount of disk space to be consumed for each attribute group whose historical data is being collected. Required disk storage is an important factor to consider when you are defining data collection rules and your strategy for historical data collection.

Calculate expected disk space consumption by multiplying the number of bytes per instance by the expected number of instances, and then multiplying that product by the number of samples. Table 9 on page 312 provides the following information required to calculate disk space for the DB2 agent:

- *DB table name* is the table name that is displayed in the warehouse database, if the attribute group is configured to be written to the warehouse.
- *Bytes per instance (agent)* is an estimate of the record length for each row or instance written to the agent disk for historical data collection. This estimate can be used for agent disk space planning purposes.
- *Bytes per instance (warehouse)* is an estimate of the record length for detailed records written to the warehouse database, if the attribute group is configured to be written to the warehouse. Detailed records are those that have been uploaded from the agent for long-term historical data collection. This estimate can be used for warehouse disk space planning purposes.
- *Bytes per summarized instance (warehouse)* is an estimate of the record length for aggregate records written to the warehouse database, if the attribute group is configured to be written to the warehouse. Aggregate records are created by the Summarization agent for attribute groups that have been configured for summarization. This estimate can be used for warehouse disk space planning purposes.
- *Expected number of instances* is a guideline that can be different for each attribute group, because it is the number of instances of data that the agent returns for a given attribute group, and depends upon the application environment that is being monitored. For example, if your attribute group is monitoring each processor on your workstation and you have a dual processor workstation, the number of instances is 2.

The IBM Tivoli Monitoring Installation and Setup Guide contains formulas that can be used to estimate the amount of disk space used at the agent and in the warehouse database for historical data collection of an attribute group.

Table 9. Capacity planning for historical data

Attribute group	DB table name	Bytes per instance (agent)	Bytes per instance (warehouse)	Bytes per summarized instance (warehouse)	Expected number of instances
Application00 (group 00)	KUDAPPL00	3802	3957	6924	Number of applications using the DB2 instance
Application (group 00) (superseded)	KUD2649700	3104	31245202	5202	Number of applications using the DB2 instance
Application (group 00, Unicode application attributes) (superseded)	KUD2649900	3638	3658	5736	Number of applications using the DB2 instance
Application (group 01)	KUDAPPL01	634	862	2552	Number of applications using the DB2 instance
Application (group 01) (superseded)	KUD2649800	626	700	2136	Number of applications using the DB2 instance
Apply Program	KUDAPPLYPM	316	336	640	Number of apply program processes with the DB2 instance
Apply Subscription	KUDAPPLYSN	414	434	738	Number of apply program subscription sets with the DB2 instance
Buffer Pool	KUDBPOOL	1354	1401	2880	Number of buffer pools per database (default of 1)
Buffer Pool Data (superseded)	KUD4177600	1490	1510	2526	Number of buffer pools per database (default of 1)
Database (group 00)	KUDDBASE00	1850	2032	5166	Number of databases with the DB2 instance
Database (group 00) (superseded)	KUD3437500	1850	1870	4260	Number of databases with the DB2 instance
Database (group 01)	KUDDBASE01	1646	2179	7090	Number of databases with the DB2 instance
Database (group 01) (superseded)	KUD3437600	1650	1670	5404	Number of databases with the DB2 instance
Database02	KUDDBASE02	806	907	2646	Number of databases with the DB2 instance
DCS Database	KUDDCSDB	270	289	650	Number of DCS databases with the DB2 instance
Diagnostic Log	KUDDIAGLOG	1675	1695	1729	
Diagnostic Messages	KUDMESSAGE	N/A	N/A	N/A	This attribute group is available for historical data collection
Locking Conflict	KUD5214100	1022	1024	1246	Number of applications using the DB2 instance

Table 9. Capacity planning for historical data (continued)

Attribute group	DB table name	Bytes per instance (agent)	Bytes per instance (warehouse)	Bytes per summarized instance (warehouse)	Expected number of instances
Log	KUDLOG	4926	5081	6540	
Log Record	KUDLOGREC	1074	N/A	N/A	
Network Info	KUDIPADDR	166	186	218	
System Overview	KUDSYSINFO	1714	2112	5366	One per DB2 instance
System Overview (superseded)	KUD4238000	1590	1610	4270	One per DB2 instance
System Resources	KUDSYSRES	652	753	2152	
Table	KUDTABLE	304	378	596	Number of tables per database
Tablespace	KUDTBLSPC	1754	1963	4726	Number of tablespaces per database (default of 4)
Tablespace (superseded)	KUDTABSPC	1802	1802	3918	Number of tablespaces per database (default of 4)
Tablespace Auto Resize	KUDRESIZ	848	900	1100	Number of tablespaces per database (default of 4)

For more information about historical data collection, see the *IBM Tivoli Monitoring Administrator's Guide*.

Chapter 6. Situations reference

This chapter contains an overview of situations, references for detailed information about situations, and descriptions of the predefined situations included in this monitoring agent.

About situations

A situation is a logical expression involving one or more system conditions. Situations are used to monitor the condition of systems in your network. You can manage situations from the Tivoli Enterprise Portal by using the Situation editor.

The IBM Tivoli Monitoring agents that you use to monitor your system environment are shipped with a set of predefined situations that you can use as-is or you can create new situations to meet your requirements. Predefined situations contain attributes that check for system conditions common to many enterprises.

Using predefined situations can improve the speed with which you can begin using the DB2 agent. You can examine and, if necessary, change the conditions or values being monitored by a predefined situation to those best suited to your enterprise.

Important: The predefined situations provided with this monitoring agent are not read-only. Do not edit these situations, and save them. Software updates write over any of the changes that you make to these situations. Instead, clone the situations that you want to change to suit your enterprise.

You can display predefined situations and create your own situations using the Situation editor. The left frame of the Situation editor initially lists the situations associated with the Navigator item that you selected. When you click a situation name or create a new situation, the right frame opens with the following tabs:

Formula
Condition being tested

Distribution
List of managed systems (operating systems, subsystems, or applications) to which the situation can be distributed.

Expert Advice
Comments and instructions to be read in the event workspace

Action
Command to be sent to the system

Until Duration of the situation

More information about situations

The *IBM Tivoli Monitoring User's Guide* contains more information about predefined and custom situations and how to use them to respond to alerts.

For a list of the predefined situations for this monitoring agent and a description of each situation, see Predefined situations.

Predefined situations

This monitoring agent contains the following predefined situations, which are organized alphabetically:

- UDB_Agent_DB_Standby
- UDB_Agent_DM_Down
- UDB_Agent_Insufficient_Auth
- UDB_Agent_WaitToken_High
- UDB_Agent_WaitToken_High_2
- UDB_Agents_Stolen_High
- UDB_Agents_Stolen_High2
- UDB_Appl_BP_Hit_Ratio_Low
- UDB_Appl_BP_Hit_Ratio_Low_2
- UDB_Appl_BP_Hit_Ratio_Low_610
- UDB_Appl_CatCache_Hit_Low
- UDB_Appl_CatCache_Hit_Low _2
- UDB_Appl_CatCache_Hit_Low_610
- UDB_Appl_HJoinOflws_High
- UDB_Appl_HJoinOflws_High _2
- UDB_Appl_HJoinOflws_High_610
- UDB_Appl_HJoinSmOflw_High
- UDB_Appl_HJoinSmOflw_High_2
- UDB_Appl_HJoinSmOflw_High_610
- UDB_Appl_Lock_Warning
- UDB_Appl_Lock_Warning_2
- UDB_Appl_Lock_Warning_610
- UDB_Appl_PkgCache_Hit_Low
- UDB_Appl_PkgCache_Hit_Low_2
- UDB_Appl_PkgCache_Hit_Low_610
- UDB_Appl_SQL_Fail_High
- UDB_Appl_SQL_Fail_High_2
- UDB_Appl_SQL_Fail_High_610
- UDB_Appl_Wait_Lock
- UDB_Appl_Wait_Lock_2
- UDB_Appl_Wait_Lock_610
- UDB_BP_DrtyPg_Steal_Clns
- UDB_BP_DrtyPg_Steal_Clns_2
- UDB_BP_DrtyPg_thresh_Clns
- UDB_BP_DrtyPg_thrsh_Clns_2
- UDB_BP_Hit_Ratio_Low
- UDB_BP_Hit_Ratio_Low_2
- UDB_Buff_Max_Used_Pct_Crit
- UDB_Buff_Max_Used_Pct_Crit _2
- UDB_Buff_Max_Used_Pct_Warn
- UDB_Buff_Max_Used_Pct_Warn_2

- UDB_Buff_Used_Pct_Crit
- UDB_Buff_Used_Pct_Crit_2
- UDB_Buff_Used_Pct_Warn
- UDB_Buff_Used_Pct_Warn_2
- UDB_Ce_Max_Used_Pct_Crit
- UDB_Ce_Max_Used_Pct_Crit_2
- UDB_Ce_Max_Used_Pct_Warn
- UDB_Ce_Max_Used_Pct_Warn_2
- UDB_Ce_Used_Pct_Crit
- UDB_Ce_Used_Pct_Crit_2
- UDB_Ce_Used_Pct_Warn
- UDB_Ce_Used_Pct_Warn_2
- UDB_Customized_SQL_Failed
- UDB_Database_Lock_Warning
- UDB_Database_Lock_Warning_2
- UDB_DB_BP_Hit_Ratio_Low
- UDB_DB_BP_Hit_Ratio_Low_2
- UDB_DB_Cat_Cache_Hit_Ratio_Crit
- UDB_DB_Cat_Cache_Hit_Ratio_Crit_2
- UDB_DB_Cat_Cache_Hit_Ratio_Warn
- UDB_DB_Cat_Cache_Hit_Ratio_Warn_2
- UDB_DB_Cur_Cons_Pct_Crit
- UDB_DB_Cur_Cons_Pct_Crit_2
- UDB_DB_Cur_Cons_Pct_Warn
- UDB_DB_Cur_Cons_Pct_Warn_2
- UDB_DB_Dlk_Rb_Pct_For_Int_Crit
- UDB_DB_Dlk_Rb_Pct_For_Int_Crt_2
- UDB_DB_Dlk_Rb_Pct_For_Int_Warn
- UDB_DB_Dlk_Rb_Pct_For_Int_Wrn_2
- UDB_DB_File_Closed_High
- UDB_DB_File_Closed_High_2
- UDB_DB_Int_Deadlock_Rb_Pct_Crit
- UDB_DB_Int_Deadlock_Rb_Pct_Crit_2
- UDB_DB_Int_Deadlock_Rb_Pct_Warn
- UDB_DB_Int_Deadlock_Rb_Pct_Warn_2
- UDB_DB_Invalid_Pkgs_Crit
- UDB_DB_Invalid_Pkgs_Crit_2
- UDB_DB_Invalid_Pkgs_Warn
- UDB_DB_Invalid_Pkgs_Warn_2
- UDB_DB_Invalid_Sys_Pkgs_Crit
- UDB_DB_Invalid_Sys_Pkgs_Crit_2
- UDB_DB_Invalid_Sys_Pkgs_Warn
- UDB_DB_Invalid_Sys_Pkgs_Warn_2
- UDB_DB_Invalid_Triggers_Crit
- UDB_DB_Invalid_Triggers_Crit_2

- UDB_DB_Invalid_Triggers_Warn
- UDB_DB_Invalid_Triggers_Warn_2
- UDB_DB_Lock_Waits_Pct_Crit
- UDB_DB_Lock_Waits_Pct_Crit_2
- UDB_DB_Lock_Waits_Pct_Warn
- UDB_DB_Lock_Waits_Pct_Warn_2
- UDB_DB_Pool_Hit_Idx_Pct_Crit
- UDB_DB_Pool_Hit_Idx_Pct_Crit_2
- UDB_DB_Pool_Hit_Idx_Pct_Warn
- UDB_DB_Pool_Hit_Idx_Pct_Warn_2
- UDB_DB_Pool_Hit_Ratio_Pct_Crit
- UDB_DB_Pool_Hit_Ratio_Pct_Crit_2
- UDB_DB_Pool_Hit_Ratio_Pct_Warn
- UDB_DB_Pool_Hit_Ratio_Pct_Warn_2
- UDB_DB_Pri_Log_Used_Pct_Crit
- UDB_DB_Pri_Log_Used_Pct_Crit_2
- UDB_DB_Pri_Log_Used_Pct_Warn
- UDB_DB_Pri_Log_Used_Pct_Warn_2
- UDB_DB_Sec_Log_Used_Pct_Crit
- UDB_DB_Sec_Log_Used_Pct_Crit_2
- UDB_DB_Sec_Log_Used_Pct_Warn
- UDB_DB_Sec_Log_Used_Pct_Warn_2
- UDB_DB_Sort_Overflow_High
- UDB_DB_Sort_Overflow_High_2
- UDB_DB_SQL_Fail_High
- UDB_DB_SQL_Fail_High_2
- UDB_DB_SQL_Stmts_Fail_Pct_Crit
- UDB_DB_SQL_Stmts_Fail_Pct_Crit_2
- UDB_DB_SQL_Stmts_Fail_Pct_Warn
- UDB_DB_SQL_Stmts_Fail_Pct_Wrn_2
- UDB_DB_Status_Warn
- UDB_HADR_Con_Status_Congest
- UDB_HADR_Con_Status_Disconnect
- UDB_HADR_Primary_Down
- UDB_HADR_Primary_Status_Warn
- UDB_HADR_Primary_Status_Warn_2
- UDB_HADR_Standby_Down
- UDB_HADR_Stopped_Warn
- UDB_Log_Archive_Info
- UDB_Log_Diag_Msg_Crit
- UDB_Log_Disk_Space_Crit
- UDB_Log_FailArch_Path_Warn
- UDB_Log_Overflow_Path_Warn
- UDB_Log_Pri_Log_Used_Pct_Crit
- UDB_Log_Pri_Log_Used_Pct_Warn

- UDB_Log_Sec_Log_Used_Pct_Crit
- UDB_Log_Sec_Log_Used_Pct_Warn
- UDB_Ma_Max_Used_Pct_Crit
- UDB_Ma_Max_Used_Pct_Crit_2
- UDB_Ma_Max_Used_Pct_Warn
- UDB_Ma_Max_Used_Pct_Warn_2
- UDB_Max_Agent_Overflows_High
- UDB_Max_Agent_Overflows_High_2
- UDB_Pip_Sort_Hit_Ratio_Pct_Crit
- UDB_Pip_Sort_Hit_Rat_Pct_Crt_2
- UDB_Pip_Sort_Hit_Ratio_Pct_Warn
- UDB_Pip_Sort_Hit_Rat_Pct_Wrn_2
- UDB_Piped_Sorts_Rej_Pct_Crit
- UDB_Piped_Sorts_Rej_Pct_Crit_2
- UDB_Piped_Sorts_Rej_Pct_Warn
- UDB_Piped_Sorts_Rej_Pct_Warn_2
- UDB_Post_Threshold_Sorts_High
- UDB_Post_Threshold_Sorts_High_2
- UDB_Rb_Max_Used_Pct_Crit
- UDB_Rb_Max_Used_Pct_Crit_2
- UDB_Rb_Max_Used_Pct_Warn
- UDB_Rb_Max_Used_Pct_Warn_2
- UDB_Rb_Used_Pct_Crit
- UDB_Rb_Used_Pct_Crit_2
- UDB_Rb_Used_Pct_Warn
- UDB_Rb_Used_Pct_Warn_2
- UDB_Status_Warning
- UDB_Status_Warning_2
- UDB_TS_Sp_Used_DMS_Tab_Pct_Crit
- UDB_TS_Sp_Used_DMS_Tab_Pct_Crt_2
- UDB_TS_Sp_Used_DMS_Tab_Pct_Warn
- UDB_TS_Sp_Used_DMS_Tb_Pct_Wrn_2
- UDB_TS_Status_Warn
- UDB_TS_Status_Warn_2
- UDB_TS_Utilization_Crit
- UDB_TS_Utilization_Warn

The remaining sections of this chapter contain descriptions of each of these predefined situations. The situations are organized alphabetically.

UDB_Agent_DB_Standby

Issues a warning alert if the monitored database is an HADR standby database and you issue an SQL command to the database. This situation has the following formula:

```
Error_Code *EQ -1776
```

UDB_Agent_DM_Down

Issues a warning alert if the database manager is not running. This situation has the following formula:

```
Error_Code *EQ -1032
```

UDB_Agent_Insufficient_Auth

Issues a critical alert if the user ID that is used to run the DB2 agent does not have DB2 SYSADM authority. SYSADM authority is required for the agent to turn on all monitoring switches. This situation has the following formula:

```
Error_Code *EQ -1092
```

UDB_Agent_WaitToken_High_2

Issues a warning alert if the DB2 UDB server experiences more than 20 agents waiting for a token. This situation has the following formula:

```
UDB_waiting_on_token GT 20
```

UDB_Agent_WaitToken_High

(Superseded) Issues a warning alert if the DB2 UDB server experiences more than 20 agents waiting for a token. This situation has the following formula:

```
UDB_waiting_on_token GT 20
```

UDB_Agents_Stolen_High_2

Issues a warning alert if the DB2 UDB server experiences more than 50 stolen agents. This situation has the following formula:

```
agents_stolen GT 50
```

UDB_Agents_Stolen_High

(Superseded) Issues a warning alert if the DB2 UDB server experiences more than 50 stolen agents. This situation has the following formula:

```
agents_stolen GT 50
```

UDB_Appl_BP_Hit_Ratio_Low_2

Issues a warning alert if an application experiences a buffer pool hit ratio below 50%. This situation has the following formula:

```
pool_hit_ratio LT 50
```

UDB_Appl_BP_Hit_Ratio_Low

(Superseded) Issues a warning alert if an application experiences a buffer pool hit ratio below 50%. This situation has the following formula:

```
pool_hit_ratio LT 50
```

UDB_Appl_BP_Hit_Ratio_Low_610

(Superseded) Issues a warning alert if an application experiences a buffer pool hit ratio below 50%. This situation has the following formula:

```
pool_hit_ratio LT 50
```

UDB_Appl_CatCache_Hit_Low_2

Issues a warning alert if an application experiences a catalog cache hit ratio that is lower than 50%. This situation has the following formula:

```
cat_cache_hit_ratio LT 50
```

UDB_Appl_CatCache_Hit_Low

(Superseded) Issues a warning alert if an application experiences a catalog cache hit ratio that is lower than 50%. This situation has the following formula:

```
cat_cache_hit_ratio LT 50
```

UDB_Appl_CatCache_Hit_Low_610

(Superseded) Issues a warning alert if an application experiences a catalog cache hit ratio that is lower than 50%. This situation has the following formula:

```
cat_cache_hit_ratio LT 50
```

UDB_Appl_HJoinOflws_High_2

Issues a warning alert if an application experiences more than 20 hash join overflows. This situation has the following formula:

```
hash_join_overflows GT 20
```

UDB_Appl_HJoinOflws_High

(Superseded) Issues a warning alert if an application experiences more than 20 hash join overflows. This situation has the following formula:

```
hash_join_overflows GT 20
```

UDB_Appl_HJoinOflws_High_610

(Superseded) Issues a warning alert if an application experiences more than 20 hash join overflows. This situation has the following formula:

```
hash_join_overflows GT 20
```

UDB_Appl_HJoinSmOflw_High_2

Issues a warning alert if an application experiences more than 20% hash join small overflows. This situation has the following formula:

```
hash_join_small_overflows GT 20
```

UDB_Appl_HJoinSmOflw_High

(Superseded) Issues a warning alert if an application experiences more than 20% hash join small overflows. This situation has the following formula:

```
hash_join_small_overflows GT 20
```

UDB_Appl_HJoinSmOflw_High_610

(Superseded) Issues a warning alert if an application experiences more than 20% hash join small overflows. This situation has the following formula:

```
hash_join_small_overflows GT 20
```

UDB_Appl_Lock_Warning_2

Issues a warning alert if a monitored application experiences one or more of the following conditions:

- More than five deadlocks
- More than five lock timeouts
- More than twenty lock waits

This situation has the following formula:

```
deadlocks GT 5
```

or

```
lock_timeouts GT 5
```

or

```
lockk_waits GT 20
```

UDB_Appl_Lock_Warning

(Superseded) Issues a warning alert if a monitored application experiences one or more of the following conditions:

- More than five deadlocks
- More than five lock timeouts
- More than twenty lock waits

This situation has the following formula:

```
deadlocks GT 5
```

or

```
lock_timeouts GT 5
```

or

```
lockk_waits GT 20
```

UDB_Appl_Lock_Warning_610

(Superseded) Issues a warning alert if a monitored application experiences one or more of the following conditions:

- More than five deadlocks
- More than five lock timeouts
- More than twenty lock waits

This situation has the following formula:

```
deadlocks GT 5
```

or

```
lock_timeouts GT 5
```

or

```
lockk_waits GT 20
```

UDB_Appl_PkgCache_Hit_Low_2

Issues a warning alert if an application experiences a package cache hit ratio that is lower than 50%. This situation has the following formula:

```
pkg_cache_hit_ratio LT 50
```

UDB_Appl_PkgCache_Hit_Low

(Superseded) Issues a warning alert if an application experiences a package cache hit ratio that is lower than 50%. This situation has the following formula:

```
pkg_cache_hit_ratio LT 50
```

UDB_Appl_PkgCache_Hit_Low_610

(Superseded) Issues a warning alert if an application experiences a package cache hit ratio that is lower than 50%. This situation has the following formula:

```
pkg_cache_hit_ratio LT 50
```

UDB_Appl_SQL_Fail_High_2

Issues a warning alert if an application experiences a failure rate for SQL statements that is greater than 50 percent. This situation has the following formula:

```
failed_sql_stmts_pct GT 50
```

UDB_Appl_SQL_Fail_High

(Superseded) Issues a warning alert if an application experiences a failure rate for SQL statements that is greater than 50 percent. This situation has the following formula:

```
failed_sql_stmts_pct GT 50
```

UDB_Appl_SQL_Fail_High_610

(Superseded) Issues a warning alert if an application experiences a failure rate for SQL statements that is greater than 50 percent. This situation has the following formula:

```
failed_sql_stmts_pct GT 50
```

UDB_Appl_Wait_Lock_2

Issues a warning alert if an application is waiting for a lock. This situation has the following formula:

```
agent_id_holding_lk NE 0
```

UDB_Appl_Wait_Lock

(Superseded) Issues a warning alert if an application is waiting for a lock. This situation has the following formula:

```
agent_id_holding_lk NE 0
```

UDB_Appl_Wait_Lock_610

(Superseded) Issues a warning alert if an application is waiting for a lock. This situation has the following formula:

```
agent_id_holding_lk NE 0
```

UDB_BP_DrtyPg_Steal_Clns_2

Issues a warning alert if a monitored database invokes an asynchronous page cleaner more than 20% of the time. This situation has the following formula:

```
pool_drty_pg_steal_clns GT 20
```

UDB_BP_DrtyPg_Steal_Clns

(Superseded) Issues a warning alert if a monitored database invokes an asynchronous page cleaner more than 20% of the time. This situation has the following formula:

```
pool_drty_pg_steal_clns GT 20
```

UDB_BP_DrtyPg_thrsh_Clns_2

Issues a warning alert if a monitored database invokes an asynchronous page cleaner more than 20% of the time because the buffer pool reached the dirty page threshold criterion for the database. This situation has the following formula:

```
pool_drty_pg_thrsh_clns GT 20
```

UDB_BP_DrtyPg_thresh_Clns

(Superseded) Issues a warning alert if a monitored database invokes an asynchronous page cleaner more than 20% of the time because the buffer pool reached the dirty page threshold criterion for the database. This situation has the following formula:

```
pool_drty_pg_thrsh_clns GT 20
```

UDB_BP_Hit_Ratio_Low_2

Issues a warning alert if the buffer pool hit ratio falls below 50% (pool_hit_ratio LT 50). This situation has the following formula:

```
pool_hit_ratio LT 50
```

UDB_BP_Hit_Ratio_Low

(Superseded) Issues a warning alert if the buffer pool hit ratio falls below 50% (pool_hit_ratio LT 50). This situation has the following formula:

```
pool_hit_ratio LT 50
```

UDB_Buff_Max_Used_Pct_Crit_2

Issues a critical alert if the percentage maximum FCM buffers used exceeds the critical threshold. This situation has the following formula:

```
buff_max_used_pct GT 95
```

UDB_Buff_Max_Used_Pct_Crit

(Superseded) Issues a critical alert if the percentage maximum FCM buffers used exceeds the critical threshold. This situation has the following formula:

```
buff_max_used_pct GT 95
```

UDB_Buff_Max_Used_Pct_Warn_2

Issues a warning alert if the percentage maximum FCM buffers used exceeds the warning threshold. This situation has the following formula:

```
buff_max_used_pct GT 80 and LT 96
```

UDB_Buff_Max_Used_Pct_Warn

(Superseded) Issues a warning alert if the percentage maximum FCM buffers used exceeds the warning threshold. This situation has the following formula:

```
buff_max_used_pct GT 80 and LT 96
```

UDB_Buff_Used_Pct_Crit_2

Issues a critical alert if the percentage of FCM buffers currently used exceeds the critical threshold. This situation has the following formula:

```
buff_used_pct GT 95
```

UDB_Buff_Used_Pct_Crit

(Superseded) Issues a critical alert if the percentage of FCM buffers currently used exceeds the critical threshold. This situation has the following formula:

```
buff_used_pct GT 95
```

UDB_Buff_Used_Pct_Warn_2

Issues a warning alert if the percentage of FCM buffers currently used exceeds the warning threshold. This situation has the following formula:

```
buff_used_pct GT 80 and LT 96
```

UDB_Buff_Used_Pct_Warn

(Superseded) Issues a warning alert if the percentage of FCM buffers currently used exceeds the warning threshold. This situation has the following formula:

```
buff_used_pct GT 80 and LT 96
```

UDB_Ce_Max_Used_Pct_Crit_2

Issues a critical alert if the percentage of FCM connection entries currently used exceeds the critical threshold. This situation has the following formula:

```
ce_max_used_pct GT 95
```

UDB_Ce_Max_Used_Pct_Crit

(Superseded) Issues a critical alert if the percentage of FCM connection entries currently used exceeds the critical threshold. This situation has the following formula:

```
ce_max_used_pct GT 95
```

UDB_Ce_Max_Used_Pct_Warn_2

Issues a warning alert if the percentage maximum FCM connection entries used exceeds the warning threshold. This situation has the following formula:

```
ce_max_used_pct GT 80 and LT 96
```

UDB_Ce_Max_Used_Pct_Warn

(Superseded) Issues a warning alert if the percentage maximum FCM connection entries used exceeds the warning threshold. This situation has the following formula:

```
ce_max_used_pct GT 80 and LT 96
```

UDB_Ce_Used_Pct_Crit_2

Issues a critical alert if the percentage of FCM connection entries currently used exceeds the critical threshold. This situation has the following formula:

```
ce_used_pct GT 95
```

UDB_Ce_Used_Pct_Crit

(Superseded) Issues a critical alert if the percentage of FCM connection entries currently used exceeds the critical threshold. This situation has the following formula:

```
ce_used_pct GT 95
```

UDB_Ce_Used_Pct_Warn_2

Issues a warning alert if the percentage maximum FCM connection entries used exceeds the warning threshold. This situation has the following formula:

```
ce_used_pct GT 95
```

UDB_Ce_Used_Pct_Warn

(Superseded) Issues a warning alert if the percentage maximum FCM connection entries used exceeds the warning threshold. This situation has the following formula:

```
ce_used_pct GT 95
```

UDB_Customized_SQL_Failed

Issues a warning alert if the execution of a customized SQL statement fails. This situation has the following formula:

```
Last_Execution_Error_Code NE 0
```

UDB_Database_Lock_Warning_2

Issues a warning alert if the monitored database experiences one or more of the following conditions:

- More than 10 deadlocks
- More than 10 lock timeouts
- More than 20 lock waits.

This situation has the following formula:

```
deadlocks GT 10
```

or

```
lock_timeouts GT 10
```

or

```
lock_waits GT 20
```

UDB_Database_Lock_Warning

(Superseded) Issues a warning alert if the monitored database experiences one or more of the following conditions:

- More than 10 deadlocks
- More than 10 lock timeouts
- More than 20 lock waits.

This situation has the following formula:

```
deadlocks GT 10
```

or

```
lock_timeouts GT 10
```

or

```
lock_waits GT 20
```

UDB_DB_BP_Hit_Ratio_Low_2

Issues a warning alert if a database buffer pool hit ratio falls below 65%. This situation has the following formula:

```
pool_hit_ratio LT 65
```

UDB_DB_BP_Hit_Ratio_Low

(Superseded) Issues a warning alert if a database buffer pool hit ratio falls below 65%. This situation has the following formula:

```
pool_hit_ratio LT 65
```

UDB_DB_Cat_Cache_Hit_Rat_Crit_2

Issues a critical alert if the percentage catalog cache hit ratio drops below the critical threshold. This situation has the following formula:

```
cat_cache_hit_ratio LT 80
```

UDB_DB_Cat_Cache_Hit_Ratio_Crit

(Superseded) Issues a critical alert if the percentage catalog cache hit ratio drops below the critical threshold. This situation has the following formula:

```
cat_cache_hit_ratio LT 80
```

UDB_DB_Cat_Cache_Hit_Rat_Warn_2

Issues a warning alert if the percentage catalog cache hit ratio drops below the warning threshold. This situation has the following formula:

```
cat_cache_hit_ratio LT 95 and GT 79
```

UDB_DB_Cat_Cache_Hit_Ratio_Warn

(Superseded) Issues a warning alert if the percentage catalog cache hit ratio drops below the warning threshold. This situation has the following formula:

```
cat_cache_hit_ratio LT 95 and GT 79
```

UDB_DB_Cur_Cons_Pct_Crit_2

Issues a critical alert if the internal deadlock rollbacks percent for interval exceeds the critical threshold. This situation has the following formula:

```
cur_cons_pct GT 95
```

UDB_DB_Cur_Cons_Pct_Crit

(Superseded) Issues a critical alert if the internal deadlock rollbacks percent for interval exceeds the critical threshold. This situation has the following formula:

```
cur_cons_pct GT 95
```

UDB_DB_Cur_Cons_Pct_Warn_2

Issues a warning alert if the percentage of connections used exceeds the warning threshold. This situation has the following formula:

```
cur_cons_pct GT 60 and LT 96
```

UDB_DB_Cur_Cons_Pct_Warn

(Superseded) Issues a warning alert if the percentage of connections used exceeds the warning threshold. This situation has the following formula:

```
cur_cons_pct GT 60 and LT 96
```

UDB_DB_Dlk_Rb_Pct_For_Int_Crt_2

Issues a critical alert if the internal deadlock rollbacks percent for interval exceeds the critical threshold. This situation has the following formula:

```
Int_deadlock_rollbacks_pct_for_int GT 95
```

UDB_DB_Dlk_Rb_Pct_For_Int_Crit

(Superseded) Issues a critical alert if the internal deadlock rollbacks percent for interval exceeds the critical threshold. This situation has the following formula:

```
Int_deadlock_rollbacks_pct_for_int GT 95
```

UDB_DB_Dlk_Rb_Pct_For_Int_Wrn_2

Issues a warning alert if the internal deadlock rollbacks percent for interval exceeds the warning threshold. This situation has the following formula:

```
Int_deadlock_rollbacks_pct_for_int GT 80 and LT 96
```

UDB_DB_Dlk_Rb_Pct_For_Int_Warn

(Superseded) Issues a warning alert if the internal deadlock rollbacks percent for interval exceeds the warning threshold. This situation has the following formula:

```
Int_deadlock_rollbacks_pct_for_int GT 80 and LT 96
```

UDB_DB_File_Closed_High_2

Issues a warning alert if the number of files closed for a database exceeds 500 files. This situation has the following formula:

```
files_closed GT 500
```

UDB_DB_File_Closed_High

(Superseded) Issues a warning alert if the number of files closed for a database exceeds 500 files. This situation has the following formula:

```
files_closed GT 500
```

UDB_DB_Int_Ddlck_Rb_Pct_Crit_2

Issues a critical alert if the percentage of internal rollbacks due to internal deadlocks exceeds the critical threshold. This situation has the following formula:

```
Int_deadlock_rollbacks_pct GT 95
```

UDB_DB_Int_Deadlock_Rb_Pct_Crit

(Superseded) Issues a critical alert if the percentage of internal rollbacks due to internal deadlocks exceeds the critical threshold. This situation has the following formula:

```
Int_deadlock_rollbacks_pct GT 95
```

UDB_DB_Int_Ddlck_Rb_Pct_Warn_2

Issues a warning alert if the percentage of internal rollbacks due to internal deadlocks exceeds the warning threshold. This situation has the following formula:

```
Int_deadlock_rollbacks_pct GT 70 and LT 96
```

UDB_DB_Int_Deadlock_Rb_Pct_Warn

(Superseded) Issues a warning alert if the percentage of internal rollbacks due to internal deadlocks exceeds the warning threshold. This situation has the following formula:

```
Int_deadlock_rollbacks_pct GT 70 and LT 96
```

UDB_DB_Invalid_Pkgs_Crit_2

Issues a critical alert if the number of all packages exceeds the critical threshold. This situation has the following formula:

```
invalid_pkgs GT 20
```

UDB_DB_Invalid_Pkgs_Crit

(Superseded) Issues a critical alert if the number of all packages exceeds the critical threshold. This situation has the following formula:

```
invalid_pkgs GT 20
```

UDB_DB_Invalid_Pkgs_Warn_2

Issues a warning alert if the number of all packages exceeds the warning threshold. This situation has the following formula:

```
invalid_pkgs GT 1 and invalid_pkgs LT 21
```

UDB_DB_Invalid_Pkgs_Warn

(Superseded) Issues a warning alert if the number of all packages exceeds the warning threshold. This situation has the following formula:

```
invalid_pkgs GT 1 and invalid_pkgs LT 21
```

UDB_DB_Invalid_Sys_Pkgs_Crit_2

Issues a critical alert if the number of not valid SYSTEM packages exceeds the critical threshold. This situation has the following formula:

```
invalid_sys_pkgs GT 20
```

UDB_DB_Invalid_Sys_Pkgs_Crit

(Superseded) Issues a critical alert if the number of not valid SYSTEM packages exceeds the critical threshold. This situation has the following formula:

```
invalid_sys_pkgs GT 20
```

UDB_DB_Invalid_Sys_Pkgs_Warn_2

Issues a warning alert if the number of not valid SYSTEM packages exceeds the warning threshold. This situation has the following formula:

```
invalid_sys_pkgs GT 1 and LT 21
```

UDB_DB_Invalid_Sys_Pkgs_Warn

(Superseded) Issues a warning alert if the number of not valid SYSTEM packages exceeds the warning threshold. This situation has the following formula:

```
invalid_sys_pkgs GT 1 and LT 21
```

UDB_DB_Invalid_Triggers_Crit_2

Issues a critical alert if the number of not valid triggers exceeds the critical threshold. This situation has the following formula:

```
invalid_triggers GT 20
```

UDB_DB_Invalid_Triggers_Crit

(Superseded) Issues a critical alert if the number of not valid triggers exceeds the critical threshold. This situation has the following formula:

```
invalid_triggers GT 20
```

UDB_DB_Invalid_Triggers_Warn_2

Issues a warning alert if the number of not valid triggers exceeds the warning threshold. This situation has the following formula:

```
invalid_triggers GT 1 and LT 21
```

UDB_DB_Invalid_Triggers_Warn

(Superseded) Issues a warning alert if the number of not valid triggers exceeds the warning threshold. This situation has the following formula:

```
invalid_triggers GT 1 and LT 21
```

UDB_DB_Lock_Waits_Pct_Crit_2

Issues a critical alert if the percentage application in lock wait exceeds the critical threshold. This situation has the following formula:

```
lock_waits_pct GT 85
```

UDB_DB_Lock_Waits_Pct_Crit

(Superseded) Issues a critical alert if the percentage application in lock wait exceeds the critical threshold. This situation has the following formula:

```
lock_waits_pct GT 85
```

UDB_DB_Lock_Waits_Pct_Warn_2

Issues a warning alert if the percentage application in lock wait exceeds the warning threshold. This situation has the following formula:

```
lock_waits_pct GT 80 and LT 86
```

UDB_DB_Lock_Waits_Pct_Warn

(Superseded) Issues a warning alert if the percentage application in lock wait exceeds the warning threshold. This situation has the following formula:

```
lock_waits_pct GT 80 and LT 86
```

UDB_DB_Pool_Hit_Idx_Pct_Crit_2

Issues a warning alert if the percentage buffer pool hit ratio (index) falls below the critical threshold. This situation has the following formula:

```
pool_hit_ratio_index_pct_for_int LT 80
```

UDB_DB_Pool_Hit_Idx_Pct_Crit

(Superseded) Issues a warning alert if the percentage buffer pool hit ratio (index) falls below than the critical threshold. This situation has the following formula:

```
pool_hit_ratio_index_pct_for_int LT 80
```

UDB_DB_Pool_Hit_Idx_Pct_Warn_2

Issues a warning alert if the percentage buffer pool hit ratio (index) falls below the warning threshold. This situation has the following formula:

```
pool_hit_ratio_index_pct_for_int LT 95 and GT 79
```

UDB_DB_Pool_Hit_Idx_Pct_Warn

(Superseded) Issues a warning alert if the percentage buffer pool hit ratio (index) falls below the warning threshold. This situation has the following formula:

```
pool_hit_ratio_index_pct_for_int LT 95 and GT 79
```

UDB_DB_Pool_Hit_Rat_Pct_Crit_2

Issues a critical alert if the percentage buffer pool hit ratio (data plus index) falls below the critical threshold. This situation has the following formula:

```
pool_hit_ratio_pct_for_int LT 80
```

UDB_DB_Pool_Hit_Ratio_Pct_Crit

(Superseded) Issues a critical alert if the percentage buffer pool hit ratio (data plus index) falls below the critical threshold. This situation has the following formula:

```
pool_hit_ratio_pct_for_int LT 80
```

UDB_DB_Pool_Hit_Rat_Pct_Warn_2

Issues a warning alert if the percentage buffer pool hit ratio (data plus index) falls below than the warning threshold. This situation has the following formula:

```
pool_hit_ratio_pct_for_int LT 95 and GT 79
```

UDB_DB_Pool_Hit_Ratio_Pct_Warn

(Superseded) Issues a warning alert if the percentage buffer pool hit ratio (data plus index) falls below than the warning threshold. This situation has the following formula:

```
pool_hit_ratio_pct_for_int LT 95 and GT 79
```

UDB_DB_Pri_Log_Used_Pct_Crit_2

Issues a critical alert if the percentage used in primary log exceeds the critical threshold. This situation has the following formula:

```
pri_log_used_pct GT 95
```

UDB_DB_Pri_Log_Used_Pct_Crit

(Superseded) Issues a critical alert if the percentage used in primary log exceeds the critical threshold. This situation has the following formula:

```
pri_log_used_pct GT 95
```

UDB_DB_Pri_Log_Used_Pct_Warn_2

Issues a warning alert if the percentage used in primary log exceeds the warning threshold. This situation has the following formula:

```
pri_log_used_pct GT 80 and LT 96
```

UDB_DB_Pri_Log_Used_Pct_Warn

(Superseded) Issues a warning alert if the percentage used in primary log exceeds the warning threshold. This situation has the following formula:

```
pri_log_used_pct GT 80 and LT 96
```

UDB_DB_Sec_Log_Used_Pct_Crit_2

Issues a critical alert when the percentage used in secondary log exceeds the critical threshold. This situation has the following formula:

```
sec_log_used_pct GT 95
```

UDB_DB_Sec_Log_Used_Pct_Crit

(Superseded) Issues a critical alert when the percentage used in secondary log exceeds the critical threshold. This situation has the following formula:

```
sec_log_used_pct GT 95
```

UDB_DB_Sec_Log_Used_Pct_Warn_2

Issues a warning alert when the percentage used in secondary log exceeds the warning threshold. This situation has the following formula:

```
sec_log_used_pct GT 80 and LT 96
```

UDB_DB_Sec_Log_Used_Pct_Warn

(Superseded) Issues a warning alert when the percentage used in secondary log exceeds the warning threshold. This situation has the following formula:

```
sec_log_used_pct GT 80 and LT 96
```

UDB_DB_Sort_Overflow_High_2

Issues a warning alert if a monitored database experiences more than 30% sort overflows. This situation has the following formula:

```
sort_overflows_pct GT 30
```

UDB_DB_Sort_Overflow_High

(Superseded) Issues a warning alert if a monitored database experiences more than 30% sort overflows. This situation has the following formula:

```
sort_overflows_pct GT 30
```

UDB_DB_SQL_Fail_High_2

Issues a warning alert if a monitored database experiences more than 40% SQL statement failures. This situation has the following formula:

```
sql_stmts_failed_pct GT 40
```

UDB_DB_SQL_Fail_High

(Superseded) Issues a warning alert if a monitored database experiences more than 40% SQL statement failures. This situation has the following formula:

```
sql_stmts_failed_pct GT 40
```

UDB_DB_Sql_Stmts_Fail_Pct_Crt_2

Issues a warning alert if a monitored database experiences more than 95% SQL statement failures. This situation has the following formula:

```
sql_stmts_failed_pct GT 95
```

UDB_DB_Sql_Stmts_Fail_Pct_Crt

(Superseded) Issues a warning alert if a monitored database experiences more than 95% SQL statement failures. This situation has the following formula:

```
sql_stmts_failed_pct GT 95
```

UDB_DB_Sql_Stmts_Fail_Pct_Wrn_2

Issues a warning alert if a monitored database experiences more than 80% but less than 96% SQL statement failures. This situation has the following formula:

```
sql_stmts_failed_pct GT 80 LT 96
```

UDB_DB_Sql_Stmts_Fail_Pct_Warn

(Superseded) Issues a warning alert if a monitored database experiences more than 80% but less than 96% SQL statement failures. This situation has the following formula:

```
sql_stmts_failed_pct GT 80 LT 96
```

UDB_DB_Status_Warn

Issues a warning alert if the monitored database is not active. This situation has the following formula:

```
dbase_status NE Active
```

UDB_HADR_Con_Status_Congest

Issues a warning alert if the connection between the primary database and standby database is congested. This situation has the following formula:

```
hadr_connect_status EQ Congested
```

UDB_HADR_Con_Status_Disconnect

Issues a critical alert if the connection between the primary database and standby database is lost. This situation has the following formula:

```
hadr_connect_status EQ Disconnected
```

UDB_HADR_Primary_Down

Issues a critical alert if the primary database is inactive. This situation has the following formula:

```
hadr_role EQ Primary AND db_status EQ Inactive
```

UDB_HADR_Primary_Status_Warn

(Superseded) Issues a warning alert if the primary database is inactive. This situation has the following formula:

```
hadr_role EQ Primary AND db_status NE Active
```

UDB_HADR_Primary_Status_Warn_2

Issues a warning alert if the primary database is inactive. This situation has the following formula:

```
hadr_role EQ Primary AND db_status NE Active AND db_status NE Inactive
```

UDB_HADR_Standby_Down

Issues a critical alert if the standby database is inactive. This situation has the following formula:

```
hadr_role EQ Standby AND db_status EQ Inactive
```

UDB_HADR_Stopped_Warn

Issues a warning alert if the HADR role changes from primary or standby to standard. This situation has the following:

```
hadr_role EQ Standard
```

UDB_Log_Archive_Info

Issues information when the database log mode is circular, not archive mode. This situation has the following formula:

```
logarchmeth1 EQ OFF and logarchmeth2 EQ OFF
```

UDB_Log_Diag_Msg_Crit

Issues a critical alert if the severity level of the log record is critical. This situation has the following formula:

```
level EQ Critical
```

UDB_Log_Disk_Space_Crit

Issues a critical alert if the free space of any log related directory is less than 10 MB. This situation has the following formula:

```
new_log_path_free_size GT 0 and LT 10240000
or
log_path_free_size GT 0 and LT 10240000
or
mirror_log_path_free_size GT 0 and LT 10240000
or
logarchmeth1_free_size GT 0 and LT 10240000
or
logarchmeth2_free_size GT 0 and LT 10240000
```

UDB_Log_FailArch_Path_Warn

Issues a warning alert if the free space of the failover log archive path is less than 10 MB. This situation has the following formula:

```
fail_log_path_free_size GT 0 and LT 10240000
```

UDB_Log_Overflow_Path_Warn

Issues a warning alert if the free space of the overflow log path is less than 10 MB. This situation has the following formula:

```
overflow_log_path_freesize GT 0 and LT 10240000
```

UDB_Log_Pri_Log_Used_Pct_Crit

Issues a critical alert if the percentage of primary log usage is higher than 95%. This situation has the following formula:

```
pri_log_used_pct GT 95
```

UDB_Log_Pri_Log_Used_Pct_Warn

Issues a warning alert if the percentage of primary log usage is higher than 80%, but is lower than 96%. This situation has the following formula:

```
pri_log_used_pct GT 80 and pri_log_used_pct LT 96
```

UDB_Log_Sec_Log_Used_Pct_Crit

Issues a critial alert if the percentage of secondary log usage is higher than 95%. This situation has the following formula:

```
sec_log_used_pct GT 95
```

UDB_Log_Sec_Log_Used_Pct_Warn

Issues a warning alert if the percentage of secondary log usage is higher than 80%, but is lower than or equal to 95%. This situation has the following formula:

```
sec_log_used_pct GT 80 and sec_log_used_pct LT 96
```

UDB_Ma_Max_Used_Pct_Crit_2

Issues a critical alert if the percentage maximum FCM message anchors used exceeds the critical threshold. This situation has the following formula:

```
ma_max_used_pct GT 95
```

UDB_Ma_Max_Used_Pct_Crit

(Superseded) Issues a critical alert if the percentage maximum FCM message anchors used exceeds the critical threshold. This situation has the following formula:

```
ma_max_used_pct GT 95
```

UDB_Ma_Max_Used_Pct_Warn_2

Issues a warning alert if the percentage maximum FCM message anchors used exceeds the warning threshold. This situation has the following formula:

```
ma_max_used_pct GT 80 and LT 96
```

UDB_Ma_Max_Used_Pct_Warn

(Superseded) Issues a warning alert if the percentage maximum FCM message anchors used exceeds the warning threshold. This situation has the following formula:

```
ma_max_used_pct GT 80 and LT 96
```

UDB_Max_Agent_Overflows_High_2

Issues a warning alert if the UDB server experiences more than 50 max agent overflows. This situation has the following formula:

```
max_agent_overflows GT 50
```

UDB_Max_Agent_Overflows_High

(Superseded) Issues a warning alert if the UDB server experiences more than 50 max agent overflows. This situation has the following formula:

```
max_agent_overflows GT 50
```

UDB_Pip_Sort_Hit_Rat_Pct_Crt_2

Issues a critical alert if the percentage piped sort hits ratio exceeds the critical threshold. This situation has the following formula:

```
piped_sort_hit_ratio_pct_for_int GT 70 and LT 96
```

UDB_Pip_Sort_Hit_Ratio_Pct_Crit

(Superseded) Issues a critical alert if the percentage piped sort hits ratio exceeds the critical threshold. This situation has the following formula:

```
piped_sort_hit_ratio_pct_for_int GT 70 and LT 96
```

UDB_Pip_Sort_Hit_Rat_Pct_Wrn_2

Issues a warning alert if the percentage piped sort hits ratio exceeds the warning threshold. This situation has the following formula:

```
piped_sort_hit_ratio_pct_for_int LT 80
```

UDB_Pip_Sort_Hit_Ratio_Pct_Warn

(Superseded) Issues a warning alert if the percentage piped sort hits ratio exceeds the warning threshold. This situation has the following formula:

```
piped_sort_hit_ratio_pct_for_int LT 80
```

UDB_Piped_Sorts_Rej_Pct_Crit_2

Issues a critical alert if the percentage of piped sorts rejected exceeds the critical threshold. This situation has the following formula:

```
piped_sorts_rejected_pct_for_int GT 95
```

UDB_Piped_Sorts_Rej_Pct_Crit

(Superseded) Issues a critical alert if the percentage of piped sorts rejected exceeds the critical threshold. This situation has the following formula:

```
piped_sorts_rejected_pct_for_int GT 95
```

UDB_Piped_Sorts_Rej_Pct_Warn_2

Issues a warning alert if the percentage of piped sorts rejected exceeds the warning threshold. This situation has the following formula:

```
piped_sorts_rejected_pct_for_int GT 80 and LT 96
```

UDB_Piped_Sorts_Rej_Pct_Warn

(Superseded) Issues a warning alert if the percentage of piped sorts rejected exceeds the warning threshold. This situation has the following formula:

```
piped_sorts_rejected_pct_for_int GT 80 and LT 96
```

UDB_Post_Threshold_Sorts_High_2

Issues a warning alert if the UDB server experiences more than 20 post-threshold sorts. This situation has the following formula:

```
Post_Threshold_Sorts GT 20
```

UDB_Post_Threshold_Sorts_High

(Superseded) Issues a warning alert if the UDB server experiences more than 20 post-threshold sorts. This situation has the following formula:

```
Post_Threshold_Sorts GT 20
```

UDB_Rb_Max_Used_Pct_Crit_2

Issues a critical alert if the percentage maximum FCM request blocks used exceeds the critical thresholds. This situation has the following formula:

```
rb_max_used_pct GT 95
```

UDB_Rb_Max_Used_Pct_Crit

(Superseded) Issues a critical alert if the percentage maximum FCM request blocks used exceeds the critical thresholds. This situation has the following formula:

```
rb_max_used_pct GT 95
```

UDB_Rb_Max_Used_Pct_Warn_2

Issues a warning alert if the percentage maximum FCM request blocks used exceeds the warning thresholds. This situation has the following formula:

```
rb_max_used_pct GT 80 and LT 96
```

UDB_Rb_Max_Used_Pct_Warn

(Superseded) Issues a warning alert if the percentage maximum FCM request blocks used exceeds the warning thresholds. This situation has the following formula:

```
rb_max_used_pct GT 80 and LT 96
```

UDB_Rb_Used_Pct_Crit_2

Issues a critical alert if the percentage of FCM request blocks currently used exceeds the critical threshold. This situation has the following formula:

```
rb_used_pct GT 95
```

UDB_Rb_Used_Pct_Crit

(Superseded) Issues a critical alert if the percentage of FCM request blocks currently used exceeds the critical threshold. This situation has the following formula:

```
rb_used_pct GT 95
```

UDB_Rb_Used_Pct_Warn_2

Issues a warning alert if the percentage of FCM request blocks currently used exceeds the warning threshold. This situation has the following formula:

```
rb_used_pct GT 80 and LT 96
```

UDB_Rb_Used_Pct_Warn

(Superseded) Issues a warning alert if the percentage of FCM request blocks currently used exceeds the warning threshold. This situation has the following formula:

```
rb_used_pct GT 80 and LT 96
```

UDB_Status_Warning_2

Issues a warning alert if the status of the monitored UDB instance is other than active. This situation has the following formula:

```
db2_status NE active
```

UDB_Status_Warning

(Superseded) Issues a warning alert if the status of the monitored UDB instance is other than active. This situation has the following formula:

```
db2_status NE active
```

This situation runs automatically at the startup of the agent.

UDB_TS_Sp_Usd_DMS_Tb_Pct_Crt_2

Issues a critical alert if the percentage of spaced used in the DMS tablespace exceeds the critical threshold. This situation has the following formula:

```
space_used_dms_table_pct GT 95
```

UDB_TS_Sp_Used_DMS_Tab_Pct_Crit

(Superseded) Issues a critical alert if the percentage of spaced used in the DMS tablespace exceeds the critical threshold. This situation has the following formula:

```
space_used_dms_table_pct GT 95
```

UDB_TS_Sp_Usd_DMS_Tb_Pct_Wrn_2

Issues a warning alert if the percentage of spaced used in the DMS tablespace exceeds the warning threshold. This situation has the following formula:

```
space_used_dms_table_pct GT 80 and LT 96
```

UDB_TS_Sp_Used_DMS_Tab_Pct_Warn

(Superseded) Issues a warning alert if the percentage of spaced used in the DMS tablespace exceeds the warning threshold. This situation has the following formula:

```
space_used_dms_table_pct GT 80 and LT 96
```

UDB_TS_Status_Warn_2

Issues a warning alert if the tablespace status is not normal. This situation has the following formula:

```
tbsp_status GT 0
```

UDB_TS_Status_Warn

(Superseded) Issues a warning alert if the tablespace status is not normal. This situation has the following formula:

```
tbsp_status GT 0
```

UDB_TS_Utilization_Crit

The tablespace utilization is measured as the percentage of space consumed. A critical alert is issued when that figure reaches a specific threshold. This situation has the following formula:

```
TBSP_Utilization GT 95
```

UDB_TS_Utilization_Warn

Issues a warning alert if the tablespace utilization percentage is in a specific range. This situation has the following formula:

```
TBSP_Utilization GT 80 and LT 96
```

Chapter 7. Take Action commands reference

This chapter contains an overview of Take Action commands, references for detailed information about Take Action commands, and descriptions of the Take Action commands included in this monitoring agent.

About Take Action commands

Take Action commands can be run from the desktop or included in a situation or a policy.

When included in a situation, the command executes when the situation becomes true. A Take Action command in a situation is also referred to as reflex automation. When you enable a Take Action command in a situation, you automate a response to system conditions. For example, you can use a Take Action command to send a command to restart a process on the managed system or to send a text message to a cell phone.

Advanced automation uses policies to perform actions, schedule work, and automate manual tasks. A policy comprises a series of automated steps called activities that are connected to create a workflow. After an activity is completed, Tivoli Enterprise Portal receives return code feedback, and advanced automation logic responds with subsequent activities prescribed by the feedback.

More information about Take Action commands

For more information about working with Take Action commands, see the *IBM Tivoli Monitoring User's Guide*.

For a list of the Take Action commands for the DB2 agent and a description of each command, see Predefined Take Action commands.

Predefined Take Action commands

This monitoring agent contains the following Take Action commands:

- Backup Database
- Rebind All Packages
- Rebind Package
- Reorg Table
- Run Statistics
- Start DB2
- Stop DB2
- Update Database Configuration
- Update DB Manager Configuration

Important: The user ID of the process running the command is used for authentication against the DB2 server.

The remaining sections of this chapter contain descriptions of each of these Take Action commands, which are listed alphabetically. The following information is provided following the description of each Take Action command:

Authorization role
The required authorization role, plus any required DB2 permissions

Arguments
List of arguments, if any, for the Take Action with a short description and default value for each one

Destination systems
Where the command is to be executed: on the Managed System (monitoring agent) where the agent runs or on the Managing System (Tivoli Enterprise Monitoring Server) to which it is connected

Usage notes
Additional relevant notes for using the Take Actions

Remember: If you need to set an argument value of a Take Action command with quotes, input the value using the following format:

`\"`*`argument_value`*`\"`

For example, if you want to set an argument value as `"Linux_Disk"`, you must input `\"Linux_Disk\"`.

Backup Database action

Creates a backup copy of a database.

Authorization role

DB2 permissions are **sysmaint**, **sysctrl**, or **sysadm**.

Arguments

db_instance_name
Specifies the name of the DB2 instance. Enter a text string with a maximum of 60 bytes.

db_name
Specifies the name of the database to be backed up. This is the name the database was given when created. Enter a text string with a maximum of 60 bytes.

on_line_switch
Enter YES or NO to specify an online backup.

TSM_switch
Enter YES or NO to indicate whether to use Tivoli Storage Manager managed output (formerly ADSM) for the backup.

TSM_Sessions
Specifies the number of I/O sessions to be used with the Tivoli Storage Manager managed output. The default value is 2. If you are not using managed output, specify any number (it will be ignored).

target_device
Specifies a directory or tape device name to which the backup is written. If you specify a directory, you must enter the full path of the directory. The directory must already exist. The default for UNIX hosts is the /TMPDIR directory. There is no default for Windows hosts.

number_of_buffers
: Specifies the number of buffers to use during the backup process. A typical value is 1.

buffer_size
: Specifies the number of pages for the buffer that is used when building the backup image. The minimum size is 16 pages and the default is 1024.

parallelism
: Specifies the number of buffer manipulators to spawn during the restore process. The default is 1.

libname
: Specifies the name of the shared library that contains the vendor backup and restore input and output functions to be used. This variable can contain the full path and file name of the library. If the full path is not specified, the path is set to the directory that contains the user exit program. This variable is optional.

table_space
: Specifies one or more tablespaces to back up. If no tablespace is specified, all tablespaces are backed up. To specify more than one tablespace, separate them using commas.

Destination system

Managed system

Usage notes

Because this task can take longer to complete than the default time of 60 seconds, you need to set the timeout value to a larger value, like 600 seconds.

If a database becomes damaged or corrupted, it can be returned to the state of the backed up copy. If a successfully restored database was enabled for roll-forward recovery at the time of the backup, it can be returned to the state that it was in prior to the occurrence of damage. The backup might be directed to hard disk, diskette, tape, Tivoli Storage Manager utility, or to other vendor products enabled for DB2.

For more information, see the BACKUP DATABASE CLP command in the DB2 command reference information for the version of DB2 that you are using.

Rebind All Packages action

Re-creates all packages stored in the database without needing a bind file.

Authorization role

DB2 permissions are **sysadm** or **dbadm**.

Arguments

db_instance_name
: Specifies the name of the DB2 instance. Enter a text string with a maximum of 60 bytes.

db_name
: Specifies the name of the database. This is the name the database was given when created. Enter a text string with a maximum of 60 bytes.

log_file_name
Specifies the path (optional) and the file name (mandatory) to be used for recording errors that result from the package revalidation procedure.

Destination system

Managed system

Usage notes

The Rebind Package command does not automatically commit the transaction following a successful rebind. This task uses the CLP REBIND command to attempt the revalidation of all packages in a database. You can allow package revalidation to occur implicitly when the packages are first used.

For more information, see the db2rbind system command in the DB2 command reference for the version of DB2 that you are using.

Rebind Package action

Re-creates a package stored in the database without needing a bind file.

Authorization role

DB2 permissions are **sysadm** or **dbadm**.

Arguments

db_instance_name
Specifies the name of the DB2 instance. Enter a text string with a maximum of 60 bytes. .

db_name
Specifies the name of the database. This is the name the database was given when created. Enter a text string with a maximum of 60 bytes.

package_name_schema
Specifies the qualifier for the package name. For example, if the qualified name of a package is USERID.PACK1, the schema is USERID. If you specify _default_, the current user ID is used.

package_name
Specifies the unqualified package name. For example, if the qualified name of a package is USERID.PACK1, the name is PACK1.

Destination system

Managed system

Usage notes

The Rebind Package command does not automatically commit the transaction following a successful rebind. The user must explicitly commit the transaction. This enables "what if" analysis, in which the user updates certain statistics, and then tries to rebind the package to see what changes. It also permits multiple rebinds within a unit of work.

The Rebind Package command commits the transaction if auto-commit is enabled.

For more information, see the REBIND CLP command in the DB2 command reference for the version of DB2 that you are using.

Reorg Table action

Reorganizes a table by reconstructing the rows to eliminate fragmented data and by compacting information.

Authorization role

DB2 permissions are **sysmaint**, **sysctrl**, **sysadm**, or **dbadm**.

Arguments

db_instance_name
: Specifies the name of the DB2 instance. Enter a text string with a maximum of 60 bytes.

db_name
: Specifies the name of the database. This is the name the database was given when created. Enter a text string with a maximum of 60 bytes.

table_schema
: Specifies the qualifier for the table name. For example, if the qualified name of a table is USERID.TABLE1, the schema is USERID.

table_name
: Specifies the unqualified table name. For example, if the qualified name of a table is USERID.TABLE1, the name is TABLE1.

index_schema
: Specifies the qualifier for the index name.

index_name
: Specifies the unqualified index name.

table_space
: Specifies the unqualified tablespace name.

Destination system

Managed system

Usage notes

Because this task might take longer than the 60-second default time to complete, set the timeout value to a larger value, such as 600 seconds.

Tables that have been modified so many times that data is fragmented and access performance is noticeably slow are candidates for reorganization. Be sure to complete all database operations and release all locks before running the Reorg Table command. After reorganizing a table, use the Run Statistics Take Action command to update the table statistics, and the Rebind Take Action command to rebind the packages that use this table.

If the reorganization is not successful, do not delete temporary files. The database manager uses these files to recover the database.

For more information, see the REORG TABLE CLP command in the DB2 command reference information for the version of DB2 that you are using.

Run Statistics action

Updates statistics about the physical characteristics of a table and the associated indexes.

Authorization role

DB2 permissions are **sysmaint**, **sysctrl**, **sysadm**, or **dbadm**.

Arguments

db_instance_name
: Specifies the name of the DB2 instance. Enter a text string with a maximum of 60 bytes.

db_name
: Specifies the name of the database. This is the name the database was given when created. Enter a text string with a maximum of 60 bytes.

table_schema
: Specifies the qualifier for the table name. For example, if the qualified name of a table is USERID.TABLE1, the schema is USERID.

table_name
: Specifies the unqualified table name. For example, if the qualified name of a table is USERID.TABLE1, the name is TABLE1.

dist_switch
: Enter Yes or No to indicate whether you want distribution statistics calculated.

index_switch
: Enter one of the following responses:

  Yes Updates statistics on the indexes.

  No Does not update statistics on the indexes.

  Index Specifies to update statistics on a specific index that is defined in the index-schema and index-name variables.

index_only_switch
: Enter Yes or No to indicate whether you want to update statistics on the indexes only.

  If the index argument is Yes and index_only_switch is No, statistics on both the table and its indexes are updated.

  If index is No and index_only_switch is No, statistics on the table only are updated.

detail_switch
: Enter Yes or No to indicate whether you want extended index statistics calculated when you are updating statistics on the indexes.

share_switch
: Enter Change or Reference to indicate how other users can access the table while statistics are calculated:

  Change
  : Other users can read from and write to the table while statistics are calculated.

  Reference
  : Other users can only read from the table while statistics are calculated.

index_schema
: Specifies the qualifier for the index name. This variable is required if you specify the Index parameter for the index argument.

index_name

Specifies the unqualified index name. This variable is required if you specify the Index parameter for the index argument.

Destination system

Managed system

Usage notes

Run this command when a table has had many updates, or after reorganizing a table. The statistics updated by this task include number of records, number of pages, and average record length. The optimizer uses these statistics when determining access paths to the data.

After the command is run, note the following item:

- A COMMIT must be issued to release the locks.
- To allow new access plans to be generated, the packages that reference the target table must be rebound.

After statistics are updated, you can create new access paths to the table by rebinding the packages.

Collect statistics that apply only to the tables before you create indexes. By running the table statistics first, you ensure that the indexes statistics are not overlaid by table statistics.

For more information, see the RUNSTATS CLP command in the DB2 command reference information for the version of DB2 that you are using.

Start DB2 action

Starts the DB2 instance and allocates resources.

Authorization role

DB2 permissions are **sysmaint**, **sysctrl**, or **sysadm**.

Arguments

db_instance_name

The name of the DB2 instance. Enter a text string with a maximum of 60 bytes.

db_partition

The DB2 database partition node number, which can range from 0 to 999.

Destination system

Managed system

Usage notes

After this task starts, the database manager instance runs until you stop it, even if all programs that were using it have ended.

For more information when you are running UNIX, see the db2start system command in the DB2 command reference for the version of DB2 that you are using.

Stop DB2 action

Stops the DB2 instance and releases resources.

Authorization role

DB2 permissions are **sysmaint**, **sysctrl**, or **sysadm**.

Arguments

db_instance_name
The name of the DB2 instance. Enter a text string with a maximum of 60 bytes.

db_partition
The DB2 database partition node number, which can range from 0 to 999.

Destination system

Managed system

Usage notes

This task does not stop DB2 if any applications are connected to databases. If there are no database connections, but there are instance attachments, the Stop DB2 Take Action command forces the instance attachments and stops DB2.

Stop DB2 can be run on a DB2 server or on other managed nodes. After DB2 stops, a successful completion message is sent to the standard output device. If an error occurs, processing stops, and an error message is sent to the standard output device.

For more information when you are running UNIX, see the db2stop system command in the DB2 command reference information for the version of DB2 that you are using.

Update Database Configuration action

Modifies individual entries in a specific database configuration file.

Authorization role

DB2 permissions are **sysmaint**, **sysctrl**, or **sysadm**.

Arguments

db_instance_name
Specifies the name of the DB2 instance. Enter a text string with a maximum of 60 bytes.

dbname
Specifies the name of the database. This is the name the database was given when created. Enter a text string with a maximum of 60 bytes.

keyword1
Select the name of the database configuration parameter you want to update. You must specify from one to five keyword-value pairs using **keyword1** to **keyword5** arguments. Enter the word "None" to specify an empty keyword.

value1
Specifies the new value for the configuration parameter specified in the corresponding keyword. Enter the word "None" to specify an empty keyword.

Destination system

Managed system

Usage notes

Changes to the database configuration file become effective only after they are loaded into memory. All applications must disconnect from the database before changes can be loaded. If an error occurs, the database configuration file does not change.

The database configuration file cannot be updated if the checksum is invalid. If this problem occurs, the database must be restored to reset the database configuration file.

For more information, see the UPDATE DATABASE CONFIGURATION CLP command in the DB2 command reference information for the version of DB2 that you are using.

Update DB Manager Configuration action

Modifies individual entries in the database manager configuration file.

Authorization role

DB2 permission is **sysadm**.

Arguments

db_instance_name
Specifies the name of the DB2 instance. Enter a text string with a maximum of 60 bytes.

keyword1
Specifies the name of the database manager configuration parameter you want to update. You can specify from one to five keyword-value pairs using **keyword1** to **keyword5** arguments. Enter the word "None" to specify an empty keyword.

value1
Specifies the new value for the configuration parameter specified in the corresponding keyword. Enter the word "None" to specify an empty keyword.

Destination system

Managed system

Usage notes

Changes to the database manager configuration file become effective only after they are loaded into memory. For a server configuration, parameter changes are loaded into memory when the Start DB2 Take Action command is run. For a client configuration parameter, parameter changes are loaded into memory when the application is restarted. If an error occurs, the database manager configuration file does not change.

The database manager configuration file cannot be updated if the checksum is invalid. If this error occurs, the database manager must be reinstalled to reset the database manager configuration file.

For more information, see the UPDATE DATABASE MANAGER CONFIGURATION CLP command in the DB2 command reference information for the version of DB2 that you are using.

Chapter 8. Policies reference

This chapter contains an overview of policies and references for detailed information about policies.

About policies

Policies are an advanced automation technique for implementing more complex workflow strategies than you can create through simple automation.

A *policy* is a set of automated system processes that can perform actions, schedule work for users, or automate manual tasks. You use the Workflow Editor to design policies. You control the order in which the policy executes a series of automated steps, which are also called activities. Policies are connected to create a workflow. After an activity is completed, Tivoli Enterprise Portal receives return code feedback and advanced automation logic responds with subsequent activities prescribed by the feedback.

Important: For monitoring agents that provide predefined policies, predefined policies are not read-only. Do not edit these policies and save over them. Software updates will write over any of the changes that you make to these policies. Instead, clone the policies that you want to change to suit your enterprise.

More information about policies

For more information about working with policies, see the *IBM Tivoli Monitoring User's Guide*.

For information about using the Workflow Editor, see the *IBM Tivoli Monitoring Administrator's Guide* or the Tivoli Enterprise Portal online help.

For a list of the policies for the monitoring agent and a description of each policy, see Predefined policies.

Predefined policies

There are no predefined policies for this monitoring agent.

Chapter 9. Troubleshooting

This chapter explains how to troubleshoot the Tivoli Composite Application Manager Agent for DB2. Troubleshooting, is the process of determining why a certain product is malfunctioning.

Tip: You can resolve some problems by ensuring that your system matches the system requirements listed in "Prerequisite Checking" on page 8.

This chapter provides agent-specific troubleshooting information. See the *IBM Tivoli Monitoring Troubleshooting Guide* for general troubleshooting information. Also see "Support information" on page 384 for other problem-solving options.

Gathering product information for IBM Software Support

Before contacting IBM Software Support about a problem you are experiencing with this product, gather the following information that relates to the problem:

Table 10. Information to gather before contacting IBM Software Support

Information type	Description
Log files	Collect trace log files from failing systems. Most logs are located in a `logs` subdirectory on the host computer. See "Trace logging" on page 353 for lists of all trace log files and their locations. See the *IBM Tivoli Monitoring User's Guide* for general information about the IBM Tivoli Monitoring environment.
DB2 Universal Database™ information	• Version number and patch level • Sample application data file (if monitoring a file)
Operating system	Operating system version number and patch level
Messages	Messages and other information displayed on the screen
Version numbers for IBM Tivoli Monitoring	Version number of the following members of the monitoring environment: • IBM Tivoli Monitoring. Also provide the patch level, if available. • Tivoli Composite Application Manager Agent for DB2
Screen captures	Screen captures of incorrect output, if any.
(*UNIX only*) Core dump files	If the system stops on UNIX systems, collect core dump file from *install_dir*/bin directory, where *install_dir* is the directory path where you installed the DB2 agent.

You can use the pdcollect tool to collect the most commonly used information from a system. This tool gathers log files, configuration information, version information, and other data. See the "pdcollect tool" section in the "Tools" chapter of the IBM Tivoli Monitoring Troubleshooting Guide for more information about using this tool.

Built-in troubleshooting features

The primary troubleshooting feature in the Tivoli Composite Application Manager Agent for DB2 is logging. *Logging* refers to the text messages and trace data generated by the Tivoli Composite Application Manager Agent for DB2. Messages and trace data are sent to a file.

Trace data captures transient information about the current operating environment when a component or application fails to operate as designed. IBM Software

Support personnel use the captured trace information to determine the source of an error or unexpected condition. See “Trace logging” on page 353 for more information.

Problem classification

The following types of problems might occur with the DB2 agent:
- Installation and configuration
- General usage and operation
- Display of monitoring data
- Take Action commands

This chapter provides symptom descriptions and detailed workarounds for these problems, and also describes the logging capabilities of the DB2 agent. See the *IBM Tivoli Monitoring Troubleshooting Guide* for general troubleshooting information.

Agent upgrade and restart using non-root

The DB2 agent can run using a non-root user ID on UNIX and Linux systems. This can be done by running the **itmcmd agent start** command while logged in as a non-root user, and this can be done remotely by deploying the agent using the **Run As** option on the GUI or using the **_UNIX_STARTUP_.Username** option on the **tacmd addSystem** command line. If the agent is running using a non-root user ID, and then the agent is upgraded, restarted remotely, restarted as a result of a system reboot, or the **itmcmd agent start** is run using the root user ID, the DB2 agent subsequently runs as the root user. To confirm the user ID that the DB2 agent is using, run the following command:

```
install_dir/bin/cinfo -r
```

If the agent is using root, and that is not the desired user ID, use the following steps to restart the agent:
1. Log in as root.
2. Run the **itmcmd agent stop** command.
3. Log in (or 'su') to the user ID that you want the agent to run as.
4. Run the **itmcmd agent start** command.

If the agent was running as root because of a system reboot, edit the startup file using the following steps so that the appropriate user ID is used the next time the system is rebooted:
1. Look at *install_dir*/registry/AutoStart, and get *NUM*.
2. Edit the autostart for your operating system:

   The location of the startup file is platform dependent as follows.
   - AIX®: /etc/rc.itm*NUM*
   - HP-UX: /sbin/init.d/ITMAgents*NUM*
   - Linux: /etc/init.d/ITMAgents*NUM*
   - Solaris: /etc/init.d/ITMAgents*NUM*
3. Add or modify entries for your operating system using the following command:

```
/usr/bin/su - user
-c "install_dir/bin/itmcmd agent
-h install_dir
-o instancename
start product_code"
```

Where:

user User ID that used to start the process

instancename
Name of the database instance

install_dir
Name of the directory

product_code
Two-character product code for the agent, for example, ud for the DB2 agent

Examples:

- For AIX, add entries with the following format:

```
su - USER -c "/opt/IBM/ITM/bin/itmcmd agent
-o INSTANCE start ud"
```

Where:

USER Name of the user

INSTANCE
Name of the database instance

- For Linux, HP_UX, and Solaris, add entries with the following format:

```
/bin/su - USER -c "/opt/IBM/ITM/bin/itmcmd agent
-o INSTANCE start ud >/dev/null 2>&1"
```

Where:

USER Name of the user

INSTANCE
Name of the database instance

4. Repeat Steps 1 through 3 for each instance of the DB2 agent that was stopped.
5. Save the file.

Trace logging

Trace logs capture information about the operating environment when component software fails to operate as intended. The principal log type is the RAS (Reliability, Availability, and Serviceability) trace log. These logs are in the English language only. The RAS trace log mechanism is available for all components of IBM Tivoli Monitoring. Most logs are located in a `logs` subdirectory on the host computer. See the following sections to learn how to configure and use trace logging:

- "Principal trace log files" on page 355
- "Examples: using trace logs" on page 357
- "Setting RAS trace parameters" on page 357

Important: The documentation refers to the RAS facility in IBM Tivoli Monitoring as "RAS1".

IBM Software Support uses the information captured by trace logging to trace a problem to its source or to determine why an error occurred. The default configuration for trace logging, such as whether trace logging is enabled or disabled and trace level, depends on the source of the trace logging. Trace logging is always enabled.

Overview of log file management

Table 11 on page 355 provides the names, locations, and descriptions of RAS1 log files. The log file names adhere to the following naming convention:

Windows systems:

hostname_product_instance_program_timestamp-nn.log

UNIX and Linux systems

hostname_product_program_timestamp-nn.log

where:

- *hostname* is the host name of the computer on which the monitoring component is running.
- *product* is the two-character product code. For DB2 agent, the product code is ud.
- *instance* is the name of the database instance that is being monitored.
- *program* is the name of the program being run.
- *timestamp* is an eight-character hexadecimal timestamp representing the time at which the program started.
- *nn* is a rolling log suffix. See “Examples of trace logging” for details of log rolling.

Examples of trace logging

For example, if a DB2 Universal Database instance "dbinst02" is running on computer "server01", the RAS log file for the DB2 agent might be named as follows:

```
server01_ud_dbinst02_kudcma_437fc59-01.log
```

Important: These examples show log names on Windows systems, where the program is named **kudcma**.

For long-running programs, the *nn* suffix is used to maintain a short history of log files for that startup of the program. For example, the kudcma program might have a series of log files as follows:

```
server01_ud_dbinst02_kudcma_437fc59-01.log
server01_ud_dbinst02_kudcma_437fc59-02.log
server01_ud_dbinst02_kudcma_437fc59-03.log
```

As the program runs, the first log (*nn*=01) is preserved because it contains program startup information. The remaining logs "roll." In other words, when the set of numbered logs reach a maximum size, the remaining logs are overwritten in sequence. Each time a program is started, a new timestamp is assigned to maintain a short program history. For example, if the DB2 agent is started twice, it might have log files as follows:

```
server01_ud_dbinst02_kudcma_437fc59-01.log
server01_ud_dbinst02_kudcma_437fc59-02.log
server01_ud_dbinst02_kudcma_437fc59-03.log

server01_ud_dbinst02_kudcma_537fc59-01.log
server01_ud_dbinst02_kudcma_537fc59-02.log
server01_ud_dbinst02_kudcma_537fc59-03.log
```

Each program that is started has its own log file. For example, the DB2 agent would have agent logs in this format:

```
server01_ud_dbinst02_kudcma_437fc59-01.log
```

Other logs, such as logs for collector processes and Take Action commands, have a similar syntax, as in the following example:

```
server01_ud_dbinst02_kudBackupDatabase_447fc59-01.log
```

where **kudBackupDatabase** is the name of a program.

Remember: When you communicate with IBM Software Support, you must capture and send the RAS1 log that matches any problem occurrence that you report.

Principal trace log files

Table 11 contains locations, file names, and descriptions of trace logs that can help determine the source of problems with agents.

Table 11. Trace log files for troubleshooting agents

System where the log is located	File name and path	Description
On the computer that hosts the DB2 agent See "Definitions of variables" on page 356 for descriptions of the variables in the file names in column two.	The RAS1 log files are named *hostname*_ud_*instance_program_timestamp*-*nn*.log and are located in the following path: • **On Windows systems:** *install_dir*\tmaitm6\logs • **On UNIX and Linux systems:** *install_dir*/logs **Important:** File names for RAS1 logs include a hexadecimal time stamp. **Also on UNIX and Linux systems, a log with a decimal time stamp is provided:** *hostname*_ud_*timestamp*.log and *hostname*_ud_*timestamp*.pid*nnnnn* in the *install_dir/logs* path, where *nnnnn* is the process ID number.	Traces activity of the DB2 agent. **Tip:** Other logs, such as logs for collector processes and Take Action commands (if available), have a similar syntax and are located in this directory path.
	The *.LG0 file is located in the following path: • **On Windows systems:** *install_dir*\tmaitm6\logs • **On UNIX and Linux systems:** *install_dir*/logs	A new version of this file is generated every time the agent is restarted. IBM Tivoli Monitoring generates one backup copy of the *.LG0 file with the tag .LG1. View .LG0 to learn the following details regarding the current monitoring session: • Status of connectivity with the monitoring server. • Situations that were running. • The success or failure status of Take Action commands.

Table 11. Trace log files for troubleshooting agents (continued)

System where the log is located	File name and path	Description
On the Tivoli Enterprise Monitoring Server See "Definitions of variables" for descriptions of the variables in the file names in column two.	**On UNIX and Linux systems:** The `candle_installation.log` file in the *`install_dir`*`/logs` path. > **On Windows systems:** The file in the *`install_dir`*`\InstallITM` path. Unlike RAS1 log files, the name of the file displays a *decimal* time stamp. *	Provides details about products that are installed. **Important:** Trace logging is enabled by default. A configuration step is not required to enable this tracing.
	The `Warehouse_Configuration.log` file is located in the following path on Windows systems: *`install_dir`*`\InstallITM`.	Provides details about the configuration of data warehousing for historical reporting.
	The RAS1 log file is named *`hostname`*`_ms_`*`timestamp-nn`*`.log` and is located in the following path: • **On Windows systems:** *`install_dir`*`\logs` • **On UNIX and Linux systems:** *`install_dir`*`/logs` **Important:** File names for RAS1 logs include a hexadecimal time stamp **Also on UNIX and Linux systems, a log with a decimal time stamp is provided:** *`hostname`*`_ms_`*`timestamp`*`.log` and *`hostname`*`_ms_`*`timestamp`*`.pid`*`nnnnn`* in the *`install_dir/logs`* path, where *nnnnn* is the process ID number.	Traces activity on the monitoring server.
On the Tivoli Enterprise Portal Server See "Definitions of variables" for descriptions of the variables in the file names in column two.	The RAS1 log file is named *`hostname`*`_cq_`*`timestamp-nn`*`.log` and is located in the following path: • **On Windows systems:** *`install_dir`*`\logs` • **On UNIX and Linux systems:** *`install_dir`*`/logs` **Important:** File names for RAS1 logs include a hexadecimal time stamp **Also on UNIX and Linux systems, a log with a decimal time stamp is provided:** *`hostname`*`_cq_`*`timestamp`*`.log` and *`hostname`*`_cq_`*`timestamp`*`.pid`*`nnnnn`* in the *install_dir/logs* path, where *nnnnn* is the process ID number.	Traces activity on the portal server.
	The `TEPS_ODBC.log` file is located in the following path on Windows systems: *`install_dir`*`\InstallITM`.	When you enable historical reporting, this log file traces the status of the Warehouse Proxy agent.

Definitions of variables for RAS1 logs:
- *hostname* is the host name of the computer on which the agent is running.
- *install_dir* represents the directory path where you installed the IBM Tivoli Monitoring component. *install_dir* can represent a path on the computer that hosts the monitoring server, the DB2 agent, or the portal server.
- *product* is the two character product code. For DB2 agent, the product code is ud.
- *instance* refers to the name of the database instance that you are monitoring.
- *program* is the name of the program being run.
- *timestamp* is an eight-character hexadecimal time stamp representing the time at which the program started.
- *nn* is a rolling log suffix. See "Examples of trace logging" on page 354 for details of log rolling.

See the *IBM Tivoli Monitoring Installation and Setup Guide* for more information on the complete set of trace logs that are maintained on the monitoring server.

Examples: using trace logs

Typically IBM Software Support applies specialized knowledge to analyze trace logs to determine the source of problems. However, you can open trace logs in a text editor to learn some basic facts about your IBM Tivoli Monitoring environment.

Example one

This excerpt shows the typical .LG0 log for a failed connection between a monitoring agent and a monitoring server with the host name **server1a**:

```
(Thursday, August 11, 2005, 08:21:30-{94C}kdcl0cl.c,105,"KDCL0_ClientLookup") status=1c020006,
    "location server unavailable", ncs/KDC1_STC_SERVER_UNAVAILABLE
(Thursday, August 11, 2005, 08:21:35-{94C}kraarreg.cpp,1157,"LookupProxy") Unable to connect to
    broker at ip.pipe:: status=0, "success", ncs/KDC1_STC_OK
(Thursday, August 11, 2005, 08:21:35-{94C}kraarreg.cpp,1402,"FindProxyUsingLocalLookup") Unable
    to find running CMS on CT_CMSLIST <IP.PIPE:#server1a>
```

Example two

The following excerpts from the trace log *for the monitoring server* show the status of an agent, identified here as "Remote node." The name of the computer where the agent is running is **SERVER5B**:

```
(42C039F9.0000-6A4:kpxreqhb.cpp,649,"HeartbeatInserter") Remote node SERVER5B:KUD is ON-LINE.
. . .
(42C3079B.0000-6A4:kpxreqhb.cpp,644,"HeartbeatInserter") Remote node SERVER5B:KUD is OFF-LINE.
```

Key points regarding the preceding excerpt:

- The monitoring server appends the **KUD** product code to the server name to form a unique name (SERVER5B:KUD) for this instance of DB2 agent. This unique name enables you to distinguish multiple monitoring products that might be running on **SERVER5B**.
- The log shows when the agent started (ON-LINE) and later stopped (OFF-LINE) in the environment.
- For the sake of brevity an ellipsis (...) represents the series of trace log entries that were generated while the agent was running.
- Between the ON-LINE and OFF-LINE log entries, the agent was communicating with the monitoring server.
- The ON-LINE and OFF-LINE log entries are always available in the trace log. All trace levels that are described in “Setting RAS trace parameters” provide these entries.

On Windows systems, you can use the following alternate method to view trace logs:

1. To play the Manage Tivoli Enterprise Monitoring Services window, click **Start > Programs > IBM Tivoli Monitoring > Manage Tivoli Monitoring Services**.
2. Right-click a component, and select **Advanced > View Trace Log**. The program displays the Select Log File window that lists the RAS1 logs for the DB2 agent.
3. Select a log file from the list and click **OK**. You can also use this viewer to access remote logs.

   Important: The viewer converts time stamps in the logs to a readable format.

Setting RAS trace parameters

Objective

Pinpoint a problem by setting detailed tracing of individual components of the DB2 agent and modules.

Background Information

DB2 agent uses RAS1 tracing and generates the logs described in Table 11 on page 355. The default RAS1 trace level is ERROR.

RAS1 tracing has control parameters to manage to the size and number of RAS1 logs. Use the procedure described in this section to set the parameters.

Remember: The **KBB_RAS1_LOG** parameter also provides for the specification of the log file directory, log file name, and the inventory control file directory and name. Do not modify these values, or log information can be lost.

Before you begin

See "Overview of log file management" on page 354 to ensure that you understand log rolling and can reference the correct log files when you are managing log file generation.

After you finish

Monitor the size of the **logs** directory. Default behavior can generate a total of 45 MB to 60 MB for each agent that is running on a computer. For example, each database instance that you monitor could generate 45 MB to 60 MB of log data. See the "Procedure" section to learn how to adjust file size and numbers of log files to prevent logging activity from occupying too much disk space.

Regularly prune log files other than the RAS1 log files in the **logs** directory. Unlike the RAS1 log files which are pruned automatically, other log types can grow indefinitely, for example, the logs in Table 11 on page 355 that include a process ID number (PID).

Consider using collector trace logs (described in Table 11 on page 355) as an additional source of troubleshooting information.

Remember: The **KDC_DEBUG** setting and the Maximum error tracing setting can generate a large amount of trace logging. Use them only temporarily, while you are troubleshooting problems. Otherwise, the logs can occupy excessive amounts of hard disk space.

Procedure

Specify RAS1 trace options in the *install_dir*`\tmaitm6\KUDENV` file on Windows systems or the *install_dir*`/config/ud.ini` file on UNIX and Linux systems. Use one of the following methods to modify trace options:

- **Manually edit the configuration file to set trace logging**
  1. Open the trace options file:
     - On Windows systems, open the *install_dir*`\tmaitm6\KUDENV` file.
     - On UNIX and Linux systems, open the `/`*install_dir*`/config/ud.ini` file.
  2. Edit the line that begins with **KBB_RAS1=** to set trace logging preferences.

     For example, if you want detailed trace logging, set the Maximum Tracing option (Windows, UNIX and Linux systems):

     ```
     KBB_RAS1=ERROR (UNIT:KUD ALL) (UNIT:CSS ALL) (UNIT:KRA ALL) (UNIT:IRADB2 ALL)
     ```
  3. Edit the line that begins with **KBB_RAS1_LOG=** to manage the generation of log files:
     - Edit the following parameters to adjust the number of rolling log files and their size.

- **MAXFILES**: the total number of files that are to be kept for all startups of a given program. Once this value is exceeded, the oldest log files are discarded. Default value is 9.
- **LIMIT**: the maximum size, in megabytes (MB) of a RAS1 log file. Default value is 5.

– IBM Software Support might guide you to modify the following parameters:
  - **COUNT**: the number of log files to keep in the rolling cycle of one program startup. Default value is 3.
  - **PRESERVE**: the number of files that are not to be reused in the rolling cycle of one program startup. Default value is 1.

Remember: The **KBB_RAS1_LOG** parameter also provides for the specification of the log file directory, log file name, and the inventory control file directory and name. Do not modify these values or log information can be lost.

4. Restart the DB2 agent so that your changes take effect.

- (*Windows only*) **Alternate method to edit trace logging parameters:**
  1. Open the Manage Tivoli Enterprise Monitoring Services window.
  2. Right-click the icon of the DB2 agent whose logging you want to modify.
  3. Select **Advanced** > **Edit Trace Parms**.
  4. Edit the line that begins with **KBB_RAS1=** to set trace logging preferences.

     For example, if you want detailed trace logging, set the Maximum Tracing option (Windows, UNIX and Linux systems):

     ```
     KBB_RAS1=ERROR (UNIT:KUD ALL) (UNIT:CSS ALL) (UNIT:KRA ALL) (UNIT:IRADB2 ALL)
     ```
  5. Edit the line that begins with **KBB_RAS1_LOG=** to manage the generation of log files:
     – Edit the following parameters to adjust the number of rolling log files and their size.
       - **MAXFILES**: the total number of files that are to be kept for all startups of a given program. Once this value is exceeded, the oldest log files are discarded. Default value is 9.
       - **LIMIT**: the maximum size, in megabytes (MB) of a RAS1 log file. Default value is 5.

     – IBM Software Support might guide you to modify the following parameters:
       - **COUNT**: the number of log files to keep in the rolling cycle of one program startup. Default value is 3.
       - **PRESERVE**: the number of files that are not to be reused in the rolling cycle of one program startup. Default value is 1.

     Remember: The **KBB_RAS1_LOG** parameter also provides for the specification of the log file directory, log file name, and the inventory control file directory and name. Do not modify these values or log information can be lost.
  6. (*Optional*) Click Y (Yes) in the **KDC_DEBUG Setting** menu to log information that can help you diagnose communications and connectivity problems between the DB2 agent and the monitoring server.

     Remember: The **KDC_DEBUG** setting and the Maximum error tracing setting can generate a large amount of trace logging. Use them only temporarily, while you are troubleshooting problems. Otherwise, the logs can occupy excessive amounts of hard disk space.

7. Save and close the file. You see a message reporting a restart of the DB2 agent so that your changes take effect.

Problems and workarounds

The following sections provide symptoms and workarounds for problems that might occur with the DB2 agent:

- "Installation and configuration troubleshooting" on page 360
- "Agent troubleshooting" on page 369
- "Tivoli Enterprise Portal troubleshooting" on page 376
- "Workspace troubleshooting" on page 379
- "Troubleshooting for remote deployment" on page 377
- "Situation troubleshooting" on page 380
- "Take Action command troubleshooting" on page 384

Tip: You can resolve some problems by ensuring that your system matches the system requirements listed in Chapter 2, "Agent-specific installation and configuration for the monitoring agent," on page 7.

This chapter provides agent-specific troubleshooting information. See the *IBM Tivoli Monitoring Troubleshooting Guide* for general troubleshooting information.

Installation and configuration troubleshooting

This section provides tables that show solutions for the following types of installation, configuration, and uninstallation problems:

- Operating system problems
- Problems with database applications

Table 12. Problems and solutions for installation and configuration for agents that run on UNIX and Linux systems

Problem	Solution

After installing IBM Tivoli Monitoring 6.2.3 fix pack 1 and then installing the DB2 agent you attempt to run the following command
`./cinfo -t ud`
 	
This results in an error message containing the following line:
`grep: can't open jraix523.ver`
	There is currently no solution to this problem at this time.

When you upgrade to IBM Tivoli Monitoring, you might need to apply fix packs to Candle, Version 350, agents.	Fix packs for Candle, Version 350, are delivered as each monitoring agent is upgraded to IBM Tivoli Monitoring. **Important:** The IBM Tivoli Monitoring download image or CD provides application fix packs for the monitoring agents that are installed from that CD (for example, the agents for operating systems such as Windows, Linux, UNIX, and i5/OS®). The upgrade software for other agents is located on the download image or CDs for that specific monitoring agent, such as the agents for database applications. If you do not upgrade the monitoring agent to IBM Tivoli Monitoring, the agent continues to work. However, you must upgrade to have all the functionality that IBM Tivoli Monitoring offers.

Table 12. Problems and solutions for installation and configuration for agents that run on UNIX and Linux systems (continued)

Problem	Solution
Presentation files and customized OMEGAMON® DE screens for Candle monitoring agents need to be upgraded to a new Linux on z/Series system.	The upgrade from version 350 to IBM Tivoli Monitoring handles export of the presentation files and the customized OMEGAMON DE screens.
The following message is displayed in the installation log for some Windows agents when upgrading from Tivoli OMEGAMON V350: `<REPLACELINE> Pair missing 1=[KBB_RAS1=ERROR] no 2, skipped`.	There is no workaround. The previous value of KBB_RAS1 from the OMEGAMON V350 agent is used, preserving prior customer settings for this variable. The problem has no adverse effect on the installation or subsequent operation of the monitoring agent .
Non-ASCII characters entered into the configuration window for the monitoring agent do not show up or are not the correct characters.	Enter only ASCII characters into these fields.
During the command-line installation, you choose to install a component that is already installed, and you see the following warning: `WARNING - you are about to install the SAME version of "`*`component`*`"` where *component* is the name of the component that you are attempting to install. **Important:** This problem affects UNIX and Linux command-line installations. If you monitor only Windows environments, you see this problem if you choose to install a product component (for example, a monitoring server) on a UNIX or Linux system.	You must exit and restart the installation process. You cannot return to the list where you selected components to install. When you run the installer again, do not attempt to install any component that is already installed.
While installing the agent from a CD, the following message is displayed and you are not able to continue the installation: `install.sh warning: unarchive of "/cdrom/unix/cienv1.tar" may have failed`	This error is caused by low disk space. Although the install.sh script indicates that it is ready to install the agent software, the script considers the size of *all* tar files, not the size of all the files that are contained within the tar file.Run the **df -k** command to check whether the file systems have enough space to install agents.

Table 12. Problems and solutions for installation and configuration for agents that run on UNIX and Linux systems (continued)

Problem	Solution

You see an error similar to the following example:
 	
`db2inst3@aix7%> ./bin/itmcmd agent -o db2inst3 start ud`
`CandleAgent : installer level 400 / 100. find: 0652-023 Cannot open file /home/ITM/images.`
`find: 0652-023 Cannot open file /home/ITM/images.`
`find: 0652-023 Cannot open file /home/ITM/images.`
`find: 0652-023 Cannot open file /home/ITM/images.`
`CandleAgent : running aix513 jre.`
`Starting agent...`
`Agent Started...`
`db2inst3@aix7%>`
	When the product has been installed using the root user account, you can change from the root user account to a different user account without re-installing the product.
	
	Use the **CHMOD** command to update the privileges for specific directories in the installation path for IBM Tivoli Monitoring. For example, you might use the root user account to install the DB2 agent and attempt to run the agent as the DB2 instance owner.
	Important: The error (in the "Problem" column) is occurring only for the `images` directory. The DB2 agent runs correctly, in spite of the error message. To avoid getting the `Cannot open file...` message, run the `chmod -R 755` *`directory`* command, where *directory* is the directory specified in the error.

Cannot locate the **KDCB0_HOSTNAME** setting.	Go to *`install_dir`*`/config` and edit the corresponding **.ini** file. Set the **KDCB0_HOSTNAME** parameter followed by the IP address. If you use multiple network interface cards (NICs), give the Primary IP address of the network interface.
The DB2 agent repeatedly restarts.	You can collect data to analyze this problem as follows: 1. Access the *`install_dir`*`/config/ud.ini` file, which is described in "Setting RAS trace parameters" on page 357. 2. Add the following line: `KBB_SIG1=trace –dumpoff`
Agents in the monitoring environment use different communication protocols. For example, some agents have security enabled and others do not.	Configure both the monitoring server and the Warehouse Proxy server to accept multiple protocols, as described in the *IBM Tivoli Monitoring Installation and Setup Guide*.
Creating a firewall partition file: The partition file enables an agent to connect to the monitoring server through a firewall.	**How it works:** When the agents start, they search KDCPARTITION.TXT for the following matches: • An entry that matches the partition name **OUTSIDE** • An entry that also includes a valid external address. For more information, see the *IBM Tivoli Monitoring Installation and Setup Guide*.
You see the following error: `Hub not registered with location broker. Error-code 1195.`	Confirm that the password within the Tivoli Enterprise Monitoring Server is correct.
The DB2 agent is started and running but not displaying data in the Tivoli Enterprise Portal.	Confirm that application support for this monitoring agent has been added to the Tivoli Enterprise Monitoring Server, and confirm that you have configured the Tivoli Enterprise Portal Server as described in the *IBM Tivoli Monitoring Installation and Setup Guide.* Also, see *Exploring IBM Tivoli Monitoring* to learn about overall installation and configuration issues.

Table 12. Problems and solutions for installation and configuration for agents that run on UNIX and Linux systems (continued)

Problem	Solution
You successfully upgraded from an OMEGAMON monitoring agent to IBM Tivoli Monitoring, Version 6.2.0. However, when you configure historical data collection, you see an error message that includes, `Attribute name may be invalid, or attribute file not installed for warehouse agent`.	Copy the attribute files (kud.atr) for the upgraded monitoring agent to the *install_dir*`/tables/TEMS_name/ATTRLIB` directory on the computer where you have installed the Warehouse Proxy agent. The Warehouse Proxy agent must be able to access the short attribute names for tables and columns. That way, if the longer versions of these names exceed the limits of the Warehouse database, the shorter names can be substituted.
The DB2 agent does not start in a non-ASCII environment.	Check the agent configuration to ensure that all the values are correctly represented. To view these parameters, go to the Manage Tivoli Enterprise Monitoring Services window, select the agent template, and choose the Configure using defaults. From the resulting window, select and edit the database instance to view its parameters.
Text for configuration functions is displayed in English instead of native languages when installing and configuring the DB2 agent. For example, when using the **itmcmd config** command on a UNIX or Linux system.	None. You must complete configuration of the DB2 agent using English.
The system is experiencing high CPU usage.	**Agent process:** View the CPU usage of the **kuddb2** process on a UNIX or Linux system. If CPU usage seems to be excessive, recycle the DB2 agent. **Network Cards:** The network card configurations can decrease the performance of a system. Each of the stream of packets that a network card receives (assuming it is a broadcast or destined for the under-performing system) must generate a CPU interrupt and transfer the data through the I/O bus. If the network card in question is a bus-mastering card, work can be off-loaded and a data transfer between memory and the network card can continue without using CPU processing power. Bus-mastering cards are generally 32-bit and are based on PCI or EISA bus architectures.
You successfully upgraded from an earlier version of IBM Tivoli Composite Application Manager Agent for DB2 to IBM Tivoli Composite Application Manager Agent for DB2 , Version 6.2.2. However, in the History Collection Configuration window, the names of the attribute groups that you enabled historical data collection with the earlier version changed to the following format: `KUD00_`*tablename*.	Move your mouse to the name of an attribute group, and the fly-over text displays the attribute group name. You can rename these attribute groups using the **Name** field.

Table 13. Problems and solutions for installation and configuration on Windows

Problem	Solution
When you upgrade to IBM Tivoli Monitoring, you might need to apply fix packs to Candle, Version 350, agents.	Fix packs for Candle, Version 350, are delivered as each monitoring agent is upgraded to IBM Tivoli Monitoring. **Important:** The IBM Tivoli Monitoring download image or CD provides application fix packs for the monitoring agents that are installed from that CD (for example, the agents for operating systems such as Windows, Linux, UNIX, and i5/OS). The upgrade software for other agents is located on the download image or CDs for that specific monitoring agent, such as the agents for database applications. If you do not upgrade the monitoring agent to IBM Tivoli Monitoring, the agent continues to work. However, you must upgrade to have all the functionality that IBM Tivoli Monitoring offers.
Presentation files and customized OMEGAMON DE screens for Candle monitoring agents need to be upgraded to a new Linux on z/Series system.	The upgrade from version 350 to IBM Tivoli Monitoring handles export of the presentation files and the customized OMEGAMON DE screens.
Diagnosing problems with product browse settings.	When you have problems with browse settings, perform the following steps: 1. Click on **Start > Programs > IBM Tivoli Monitoring > Manage Tivoli Monitoring Services**. The Manage Tivoli Enterprise Monitoring Services window is displayed. 2. Right-click the Windows agent and select **Browse Settings**. A text window is displayed. 3. Click **Save As** and save the information in the text file. If requested, you can forward this file to IBM Software Support for analysis.
A message similar to "Unable to find running CMS on CT_CMSLIST" in the log file is displayed.	If a message similar to "Unable to find running CMS on CT_CMSLIST" is displayed in the Log file, the agent is not able to connect to the monitoring server. Confirm the following points: • Is the computer that hosts the monitoring server running and connected to the network? • Do multiple network interface cards (NICs) exist on the system? • If multiple NICs exist on the system, find out which one is configured for the monitoring server. Ensure that you specify the correct host name and port settings for communication in the IBM Tivoli Monitoring environment.
The system is experiencing high CPU usage.	**Agent process:** View the CPU usage of the **KUDCMA** process on Windows systems. If CPU usage seems to be excessive, recycle the DB2 agent. **Network Cards:** The network card configurations can decrease the performance of a system. Each of the stream of packets that a network card receives (assuming it is a broadcast or destined for the under-performing system) must generate a CPU interrupt and transfer the data through the I/O bus. If the network card in question is a bus-mastering card, work can be off-loaded and a data transfer between memory and the network card can continue without using CPU processing power. Bus-mastering cards are generally 32-bit and are based on PCI or EISA bus architectures.
Text for configuration functions is displayed in English instead of native languages when installing and configuring the DB2 agent. For example, when using the Manage Tivoli Enterprise Monitoring Services GUI on a Windows system.	None. You must complete configuration of the DB2 agent using English.

Table 13. Problems and solutions for installation and configuration on Windows (continued)

Problem	Solution
You successfully upgraded from an OMEGAMON monitoring agent to IBM Tivoli Monitoring, Version 6.2.0. However, when you configure historical data collection, you see an error message that includes, `Attribute name may be invalid, or attribute file not installed for warehouse agent`	Copy the attribute files (kud.atr) for the upgraded monitoring agent to *install_dir*`\tmaitm6\attrlib` on the computer where you have installed the Warehouse Proxy agent. The Warehouse Proxy agent must be able to access the short attribute names for tables and columns. That way, if the longer versions of these names exceed the limits of the Warehouse database, the shorter names can be substituted.
You successfully upgraded from an earlier version of Tivoli Composite Application Manager Agent for DB2 to Tivoli Composite Application Manager Agent for DB2 , Version 6.2.2. However, in the History Collection Configuration window, the names of the attribute groups that you enabled historical data collection with the earlier version changed to the following format: `KUD00_`*tablename*.	Move your mouse to the name of an attribute group, and the fly-over text displays the attribute group name. You can rename these attribute groups using the **Name** field.

Table 13. Problems and solutions for installation and configuration on Windows (continued)

Problem	Solution
When the DB2 agent is installed using group deployment, DB2 deploygroup was run multiple times. The group deployment starts and completes successfully, but there were multiple entries in the Deploy Status Summary workspace on the Tivoli Enterprise Portal. When the command tried to install multiple times, the additional installations were queued and then were in failed state though the agent was deployed successfully. **Note:** • When the bundle group contains a single bundle and the deploy group contains more than one member (managed system of the same type as AIX or Linux), the deployment is successful on both systems. • When the bundle group contains more than one bundle and the deploy group contains single or multiple members, the deployment will be executed on each group member (managed system) depending on the members present in the bundle group and deploy group. • The command creates a transaction for each XX bundle for each target system; the bundle matching the operating system for the deployment member is processed successfully; and remaining transactions were in a queued or failed state .	There is no solution at this time.

The following table shows problems and solutions for the DB2 agent.

Table 14. Problems and solutions for installation and configuration of the DB2 agent

Problem	**Solution**
ITM6.2.3 adds a new functionality for Automatic Agent support files Data Synchronization called Self Describing Agent (SDA). This automatic product support installation feature helps to eliminate errors that might occur from the inconsistent installation of application data on the IBM Tivoli Monitoring server. It also avoids the need to recycle the hub Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, or remote Tivoli Enterprise Monitoring Server to perform a product installation. However when SDA is enabled certain situations available with the DB2 agent are no longer displayed and the DB2 node in the History Collection configuration is also no longer displayed.	Restart the Tivoli Enterprise Portal Server.

The procedure for launching the DB2 agent: Trace Parameters window in "Setting RAS trace parameters" on page 357 fails.	This problem occurs when the trace options are missing from the configuration file. You can correct the problem by doing the following steps:
	1. Edit a text file with the following path name:
	• On Windows systems: `install_dir\tmaitm6\logs\KUDRAS1`
	• On UNIX and Linux systems: `/install_dir/config/ud.ini`
	2. Paste the following configuration setting in the file:
	• On Windows systems:
	`KBB_RAS1=ERROR ^> C:\IBM\ITM\tmaitm6\logs\KUDRAS1.LOG`
	Remember: If you installed the product in a directory path other than the default, use that directory path instead of `C:\IBM\ITM`.
	• On UNIX and Linux systems:
	`KBB_RAS1=ERROR`
	The default installation path on UNIX is `/opt/IBM/ITM`.
	3. Save your changes.
	4. Repeat the "Setting RAS trace parameters" on page 357 procedure. Now the Tivoli Enterprise Monitoring Server: Trace Parameters window is displayed.

After running the agent successfully, you reinstall the agent software, and collection of monitoring data stops.	Consult the list of supported versions in Chapter 2, "Agent-specific installation and configuration for the monitoring agent," on page 7. Confirm that you are running a valid version of the target application. If you are monitoring a supported version of the database application, gather log files and other information and contact IBM Software Support, as described in "Gathering product information for IBM Software Support" on page 351.
Error counts are displayed in the Alert summary report in the Tivoli Enterprise Portal, however, error messages are not displayed in the Alert detail report.	Check the time stamp for the reports. If you have set up historical data collection for Alert summary report, set up historical data collection for the Alert detail report, too.

Table 14. Problems and solutions for installation and configuration of the DB2 agent (continued)

Problem	Solution
An error window is displayed with the message `"Service executable not found. Cannot configure service."`	This problem occurs when non-ASCII characters are entered for the DB2 instance name. Delete the configured Tivoli Enterprise Monitoring Server instance and recreate an instance specifying an instance name with ASCII characters only.
After you add the agent bundle to the Tivoli Enterprise Monitoring Server, if the IBM Tivoli Composite Application Manager Agent for DB2 is still not listed in the Select a Monitoring Agent window.	Deploy the agent through the command line.

Table 15. General problems and solutions for uninstallation

Problem	Solution
On Windows systems, uninstallation of IBM Tivoli Monitoring fails to uninstall the entire environment.	Confirm that the following problems do not exist: • Ensure that you are the only user who is logging into the computer where you are performing an uninstallation operation. If another user is performing operations during an uninstall process, the uninstall process fails. • Be sure that you follow the general uninstallation process described in the *IBM Tivoli Monitoring Installation and Setup Guide*: 1. Uninstall monitoring agents first, as in the following examples: – Uninstall a single monitoring agent for a specific database. —OR— – Uninstall all instances of a monitoring product, such as IBM Tivoli Monitoring for Databases. 2. Uninstall IBM Tivoli Monitoring.
The way to remove inactive managed systems (systems whose status is OFFLINE) from the Navigator tree in the portal is not obvious.	Use the following steps to remove, but not uninstall, an offline managed system from the Navigator tree: 1. Click the Enterprise icon in the Navigator tree. 2. Right-click, and click **Workspace** > **Managed System Status**. 3. Right-click the offline managed system, and select **Clear offline entry**. If you also want to uninstall the DB2 agent, use the procedure described in the *IBM Tivoli Monitoring Installation and Setup Guide*.
There is no configuration command or menu option to remove an instance of the DB2 agent that was created and configured on a UNIX or Linux system.	Use the following steps to delete an instance of the DB2 agent: 1. Stop the DB2 agent. 2. Open the *install_dir* directory. 3. Open the config directory. 4. Open the .ConfigData directory. 5. Edit the kudenv file, removing all lines beginning with *INSTANCE*. 6. Save the file, and exit.

Unique names for monitoring components

IBM Tivoli Monitoring might not be able to generate a unique name for monitoring components because of the truncation of names that the product automatically generates.

IBM Tivoli Monitoring automatically creates a name for each monitoring component by concatenating the subsystem name, host name, and product code separated by colons (*subsystem_name*:*hostname*:UD).

Important: When you monitor a multinode system, such as a database, IBM Tivoli Monitoring adds a subsystem name to the concatenated name, typically a database instance name.

The length of the name that IBM Tivoli Monitoring generates is limited to 32 characters. Truncation can result in multiple components having the same 32-character name. If this problem happens, shorten the *hostname* portion of the name as follows:

1. Open the configuration file for the DB2 agent, which is located in the following path:
   - **On Windows systems:** `install_dir\tmaitm6\KUDCMA_instance.ini`.
   - **On UNIX and Linux systems:** `install_dir/config/ud.ini`.
2. Find the line that begins with **CTIRA_HOSTNAME=**.
3. Type a new name for host name that is a unique, shorter name for the host computer. The final concatenated name including the subsystem name, new host name, and UD, cannot be longer than 32 characters.

   Remember: You must ensure that the resulting name is unique with respect to any existing monitoring component that was previously registered with the Tivoli Enterprise Monitoring Server.
4. Save the file.
5. Restart the agent.
6. If you do not find the files mentioned in Step 1, perform the workarounds listed in the next paragraph.

If you do not find the files mentioned in the preceding steps, uninstall the DB2 agent, and install the DB2 agent again.

Agent troubleshooting

This section lists problems that might occur with agents.

This chapter provides agent-specific troubleshooting information. See the *IBM Tivoli Monitoring Troubleshooting Guide* for general troubleshooting information.

Table 16. Agent problems and solutions for the DB2 agent

Problem	Solution
Execution of deprecated situations results in a "Problem" status.	The defined query formula is too long. You must split your query formulae into smaller queries.

Table 16. Agent problems and solutions for the DB2 agent (continued)

Problem	Solution

DB2 databases are visible in TADDM but not visible in TBSM.	Remove the comments for the following section of the `CDM_TO_TBSM4x_MAP_Templates.xml` file:
	`<--`
	`<template primary='BSM_DB2Database' >`
	`<othertemplate name='SCR_RetrieveDependentObjectsTemplate'/>`
	`<othertemplate name='SCR_ServiceComponentRawStatusTemplate'/>`
	`<othertemplate name='BSM_Database'/>`
	`<othertemplate name='BSM_Application'/>`
	`<cdmclass name='cdm:app.db.db2.Db2Database'/>`
	`</template>`
	`-->`
	
	Note: For version 6.1 of TBSM the following additional steps are required to first export the file for editing and then to import it after you are done removing the comments from the above section:
	• To extract the file from the TBSM artifact store use the `getArtifact` command
	• To import the edited file back into the system use the `putArtifact` command.
	• To apply the changes use the `utils.sh` command.

You have completed all the installation and configuration task for the DB2 agent. You then start the agent and find that some HADR related attributes are not available. For example attributes accessible through the Database Table workspace.	There is no solution to this problem at this time.
When you edit the configuration for an existing monitoring agent, the values displayed are not correct.	The original configuration settings might include non-ASCII characters. These values were stored incorrectly and result in the incorrect display. Enter new values using only ASCII characters.
Column names have different formats in table views for the DB2 agent.	Some attributes for the DB2 agent have length restrictions that cause the short column name to be used if the modified column name becomes too long.

Table 16. Agent problems and solutions for the DB2 agent (continued)

Problem	Solution
A problem can arise when you run multiple agents on one computer and want them to communicate with multiple monitoring servers, as described in this example: • Agents are running on a computer and communicating with a Tivoli Enterprise Monitoring Server, called **TEMS1**. • You install a new agent on the same computer and you want this agent to communicate with a different monitoring server, called **TEMS2**. • When you configure the new agent to communicate with **TEMS2**, all the existing agents are re-configured to communicate with **TEMS2**.	You must reconfigure the previously existing agents to restore their communication connection with **TEMS1**. For example, you can right-click the row for a specific agent in the Manage Tivoli Enterprise Monitoring Services, and select **Reconfigure**. See the *IBM Tivoli Monitoring Installation and Setup Guide* for more information on reconfiguration. If you plan to install and run multiple instances of the DB2 agent on one computer, these instances can use the same network interface, because they run as different processes. However, if you want to have two DB2 agent instances on the same computer or want to run two instances of each agent, install two-network adapters. Each instance is configured for the host-specific address so they can be recognized in the configuration settings.
The DB2 agent is installed and running normally. After rebooting the computer where the Tivoli Enterprise Monitoring Server was running, or restarting the system that hosts the Tivoli Enterprise Monitoring Server, the agent is not online. However, when you use `CandleAgent start`, the agent starts and continues running.	This problem can occur when the agent is installed locally using a non-root user, or when the agent is installed remotely using the **Run As** option on the GUI or using the **_UNIX_STARTUP_.Username** option on the **tacmd addSystem** command line. Verify whether you have used a non-root user to install the DB2 agent. Manually start the DB2 agent using the correct user ID. See "Agent upgrade and restart using non-root" on page 352 for additional information.

You receive the following error when stopping the DB2 agent when running with a local non-root UID on a UNIX or Linux Server:
 	
`./itmcmd agent -o db2itm61 stop ud`
`Stopping agent...`
`Unable to stop agent or process...`
`PID: 10642 attempt return code: 2`
	This message indicates that the UID that you are using does not have the necessary operating system privileges to stop the DB2 agent. Change to the UID that is running the DB2 agent, and run the stop command again. Use the following command on the UNIX or Linux Server:
	
	`install_dir/bin/cinfo -r`
	
	Use the UID that is returned as logged into the UNIX or Linux server to stop the DB2 agent.
	
	See "Permissions for starting and stopping the DB2 agent on UNIX and Linux systems" on page 19 for more information.

Table 16. Agent problems and solutions for the DB2 agent (continued)

Problem	Solution
A configured and running instance of the DB2 agent is not displayed in the Tivoli Enterprise Portal, but other instances of the DB2 agent on the same system are displayed in the portal.	Tivoli Monitoring products use Remote Procedure Call (RPC) to define and control product behavior. RPC is the mechanism that allows a client process to make a subroutine call (such as GetTimeOfDay or ShutdownServer) to a server process somewhere in the network. Tivoli processes can be configured to use TCP/UDP, TCP/IP, SNA, and SSL as the desired protocol (or delivery mechanism) for RPCs. "IP.PIPE" is the name given to Tivoli TCP/IP protocol for RPCs. The RPCs are socket-based operations that use TCP/IP ports to form socket addresses. IP.PIPE implements virtual sockets and multiplexes all virtual socket traffic across a single physical TCP/IP port (visible from the netstat command). A Tivoli process derives the physical port for IP.PIPE communications based on the configured, well-known port for the HUB Tivoli Enterprise Monitoring Server. (This well-known port or BASE_PORT is configured using the 'PORT:' keyword on the KDC_FAMILIES / KDE_TRANSPORT environment variable and defaults to '1918'.) The physical port allocation method is defined as (BASE_PORT + 4096*N) where N=0 for a Tivoli Enterprise Monitoring Server process and N={1, 2, ..., 15} for a non-Tivoli Enterprise Monitoring Server. Two architectural limits result as a consequence of the physical port allocation method: • No more than one Tivoli Enterprise Monitoring Server reporting to a specific Tivoli Enterprise Monitoring Server HUB can be active on a system image. • No more that 15 IP.PIPE processes can be active on a single system image. A single system image can support any number of Tivoli Enterprise Monitoring Server processes (address spaces) provided that each Tivoli Enterprise Monitoring Server on that image reports to a different HUB. By definition, there is one Tivoli Enterprise Monitoring Server HUB per monitoring Enterprise, so this architecture limit has been simplified to one Tivoli Enterprise Monitoring Server per system image. No more that 15 IP.PIPE processes or address spaces can be active on a single system image. With the first limit expressed above, this second limitation refers specifically to Tivoli Enterprise Monitoring Agent processes: no more that 15 agents per system image. This limitation can be circumvented (at current maintenance levels, IBM Tivoli Monitoring V6.1 Fix Pack 4 and later) if the Tivoli Enterprise Monitoring Agent process is configured to use EPHEMERAL IP.PIPE. (This is IP.PIPE configured with the 'EPHEMERAL:Y' keyword in the KDC_FAMILIES / KDE_TRANSPORT environment variable). There is no limitation to the number of ephemeral IP.PIPE connections per system image. If ephemeral endpoints are used, the Warehouse Proxy Agent is accessible from the Tivoli Enterprise Monitoring Server associated with the agents using ephemeral connections either by running the Warehouse Proxy Agent on the same computer or by using the Firewall Gateway feature. (The Firewall Gateway feature relays the Warehouse Proxy Agent connection from the Tivoli Enterprise Monitoring Server computer to the Warehouse Proxy Agent computer if the Warehouse Proxy Agent cannot coexist on the same computer.)

Table 16. Agent problems and solutions for the DB2 agent (continued)

<table>
<tr><th>Problem</th><th>Solution</th></tr>
<tr><td>Agent unable to connect: The agent is started, but no reports are displayed on Tivoli Enterprise Monitoring Server. The log file includes the following error:

<code>Unable to find running CMS on CMSLIST or Endpoint unavailable</code></td><td>This error message means that the agent is not able to connect to the computer where the Tivoli Enterprise Monitoring Server is running. The reason might be any one of the following reasons:
Computer where the Tivoli Enterprise Monitoring Server is running is down
Ping the computer where the Tivoli Enterprise Monitoring Server is running and make sure that it is up and running.
Tivoli Enterprise Monitoring Server is not running
If the Tivoli Enterprise Monitoring Server is not running, recycle the Tivoli Enterprise Monitoring Server and verify whether the agent is connecting.
Multiple NIC Cards on the computer where the Tivoli Enterprise Monitoring Server is running.
If multiple NICs are installed on the computer where the Tivoli Enterprise Monitoring Server is running, identify the Primary NIC and use the hostname or IP address.

Verify that the Tivoli Enterprise Monitoring Server has been configured with the IP address or hostname of the Primary NIC.

If you are using hostname, make sure that <code>/etc/hosts</code> has a valid entry for the host name and IP address of the Primary NIC.

On the Tivoli Enterprise Monitoring Server set the KDCB0_HOSTNAME variable to the primary IP address of the computer. Use the same address to configure the agent.

To connect to the Tivoli Enterprise Monitoring Server, configure the agent with the IP address of the Primary NIC or host name of the computer where the Tivoli Enterprise Monitoring Server is running.

While configuring the agent, make sure that the port number that you are connecting to is correct. If you are not using the default port number, make sure that you are using the same port number used in Tivoli Enterprise Monitoring Server. For more information, see the IBM Tivoli Monitoring Installation and Setup Guide.
Agent is behind the Firewall
If you use a Firewall, identify whether you have any one of the following scenarios:
• Hub monitoring server INSIDE, and agents OUTSIDE
• Hub and remote monitoring servers INSIDE, agents OUTSIDE
• Hub monitoring server INSIDE, remote monitoring server, and agents OUTSIDE

See Creating a firewall partition file for information about the KDC_PARTITION file that enables communication across a firewall. For additional information, see the IBM Tivoli Monitoring Installation and Setup Guide.
Connecting to the monitoring server through a Virtual Private Network (VPN)
In some cases, the agent or a remote monitoring server needs to connect to the hub monitoring server through a VPN. You must configure the communication channel (pipe) to be ephemeral, as in the following example:

<code>KDC_FAMILIES=ip.pipe port:port_number ephemeral:y ip use:n sna use:</code></td></tr>
</table>

Table 16. Agent problems and solutions for the DB2 agent (continued)

Problem	Solution
When there are multiple instances of the DB2 agent in the Tivoli Enterprise Portal, the status of the DB2 Server for instances other than the first one is shown as "InactiveBusy", and very little data is displayed in the other workspaces.	For the DB2 agent to collect data for all of your databases, you must catalog all of the databases for all of your instances.
Attributes do not allow non-ASCII input in the Situation Editor.	None. Any attribute that does not include "(Unicode)" might support only ASCII characters. For example, "Attribute (Unicode)" supports unicode, but "Attribute" without "(Unicode)" might only support ASCII characters.
Tivoli Enterprise Console events from IBM Tivoli Monitoring V6.2 for IBM Tivoli Monitoring v5.x migrated situations have parsing errors in the IBM Tivoli Monitoring server.	1. Ensure that you have IBM Tivoli Monitoring V6.2 Event Synchronization installed on your Tivoli Enterprise Console server. 2. Obtain updated baroc files from the DB2 agent events. The following updated baroc files are in Tivoli Enterprise Monitoring Server in *install_dir*/CMS/TECLIB/itm5migr: • DB2_Event.baroc • DB2Agents.baroc • DB2ApplyReplication.baroc • DB2BufferPool.baroc • DB2BufferPoolExtStorage.baroc • DB2CatalogCache.baroc • DB2CpuUtilization.baroc • DB2Cursor.baroc • DB2DatabaseStatus.baroc • DB2DirectIO.baroc • DB2FCMActivity.baroc • DB2HostThroughput.baroc • DB2InstanceStatus.baroc • DB2Locks.baroc • DB2LockWaits.baroc • DB2Logging.baroc • DB2PackageCache.baroc • DB2ReplicationCapture.baroc • DB2SAPTablespaceUsageStatus.baroc • DB2Sorts.baroc • DB2SQLStatementActivity.baroc • DB2TableApplyReplication.baroc
You are receiving Tivoli Business Service Manager events that cannot be associated because application_oid and application_class are not set.	This problem is caused by IBM Tivoli Monitoring V6.2 sending Tivoli Enterprise Console events for IBM Tivoli Monitoring 5.x migrated situations. These events are not able to set the cited slot values. Replace the DB2_send_to_TBSM.sh script on the Tivoli Enterprise Console server with the version of this file from the Tivoli Enterprise Monitoring Server in *install_dir*/CMS/TECLIB/itm5migr.
From the Tivoli Enterprise Portal, stop an agent instance that was started with the DB2 instance owner, and start the agent instance again, the agent is running with the root user ID instead of the DB2 instance owner.	Follow the reconfiguration steps in 18 to change the user ID with which the agent instance runs to the DB2 instance owner. Recycle the agent instance for the changes to take effect.

Table 16. Agent problems and solutions for the DB2 agent (continued)

Problem	**Solution**

In the kudagent log file, the following message is repeatedly displayed:
`(4AAE723A.0000-4:khdxhist.cpp,`
`3673,"readAheadRow") History file`
`</opt/IBM/ITMIST/li6263/ud/hist/`
`db2inst8/KUDDIAGLOG.old> error -`
`last row is truncated,`
`expected 1705 bytes,`
`read 342 bytes, retries = 0.`
`(4AAE723A.0001-4:khdxhist.cpp,2642,`
`"copyHistoryFile")`
`Found 1 corrupted rows`
`for "KUDDIAGLOG".`
`Rows were skipped during copying.`
`(4AAE723A.0002-4:kraahbin.cpp,`
`498,"WriteRow")`
`Samples = 10,`
`timestamp = 1090914123636`
`(4AAE723A.0003-4:kraahbin.cpp,`
`530,"WriteRow")`
`Wrote 1 rows history data,`
`UADVISOR_KUD00_KUDDIAGLOG`
`KUD00.KUDDIAGLOG,`
`<3633327809,2452621247>`
 	
Check the size of the KUDDIAGLOG historical file, and the file size exceeds the file size limitation of the file system. For example, 2G on xLinux RH4. This problem also causes high CPU usage, and might lead to agent crash.
	Do the following steps to solve this problem:
	1. Remove the historical file whose size exceeds the file size limitation.
	2. Change the history collection configuration to reduce the historical file size:
	• Change the value of the Collection Interval attribute to a larger collection interval. This can be used to reduce the amount of data that is kept on disk by collecting less historical data. For example, from the default value of 15 minutes to 1 hour.
	• Change the value of the Warehouse Interval attribute to a shorter retention period. This can be used to reduce the amount of data that is kept on disk after a successful upload to the warehouse is performed. For example, from the default value of 1 day to 1 hour.

The DB2 agent on a UNIX or Linux system does not automatically start when the UNIX or Linux system reboots.	The problem occurs because the line that is used to start the agent is missing from the start up file. To solve this problem, add the following line in the start up file:
	`/usr/bin/su - ` *`dbinstancename`*
	`-c "`*`ITM_InstalDir`*`/bin/itmcmd agent [-h `*`ITM_InstalDir`*`]`
	` -o `*`dbinstancename`*` start ud"`

32 bit DB2 agents cannot be started on a 64 bit AIX 5.3 TL5 system, and a core dumping occurs in the $CANDLEHOME/tmp directory.	To solve this problem, upgrade the libc fileset to version 5.3.8.1 or later. If you do not want to upgrade the libc fileset, see the optional solution at http://www-01.ibm.com/support/docview.wss?rs=654&context=SSTFYB&dc=DB560&dc=DB520&uid=swg21366661&loc=en_US&cs=UTF-8&lang=en&rss=ct654tivoli

The DB2 agent fails to monitor 64 bit DB2 databases on Windows systems, and the following error is found:
`(iradb2agentmain.cpp,236,`
`"LoadDb2Version")`
` WARNING!!!Windows Registry`
`reading of`
`Version value failed!!`
	This problem occurs because the DB2 registry settings are not correct.
	
	To solve this problem, set the registry setting of the HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\IBM\DB2 entry to "DB2INSTPROF"="C:\DOCUMENTS AND SETTINGS\ALL USERS\APPLICATION DATA\IBM\DB2\DB2"

Table 16. Agent problems and solutions for the DB2 agent (continued)

Problem	Solution

You have chosen to install the DB2 Agent on a windows 64 bit machine in a directory other than the default directory. You attempt to open the Agent log and the following message is displayed:
`4DE4524F.0005-A94:kudprop.cpp,315`
`,"CPropFile::needLoad")`
` Property file C:\IBM\ITMIST\`
`TMAITM6_x64\`
`db2lib.properties`
` does not exist or not accessable.:`
` 2 No such file or directory.`
	This problem occurs because the db2lib.properties file is missing from the *CandleHome*`\TMAITM6_x64\` directory.
	
	To solve this problem create a new text file and name it *db2lib.properties*, and then place it in the *CandleHome*`\TMAITM6_x64\` directory.

A memory leak occurs with a long-running DB2 agent.	The memory leak is for the DB2 server, and is caused when the monitored DB2 server is one of the following versions: • DB2 V91 Fix Pack 7 • DB2 V95 Fix Pack 5 Use a different version of the DB2 server to eliminate the memory leak.

You have configured a new instance of the DB2 instance to monitor a database instance. You then launch the Tivoli Enterprise Portal client to view workspace information. However when you try to open a workspace no data is displayed and instead a message window opens with the following error message:
`SQL1097N  The node name was`
` not found`
` in the node directory.`
` SQLSTATE=42720'`
 	
'The agent log also contains the same message.
	This situation results when in the process of configuring the new agent instance you enter an incorrect name for the database instance that you would like to monitor. The agent therefore attempts to monitor a non existent database instance. To resolve this problem you must re-configure the agent instance and enter the correct database instance name.

Tivoli Enterprise Portal troubleshooting

Table 17 on page 377 lists problems that might occur with the Tivoli Enterprise Portal. This chapter provides agent-specific troubleshooting information. See the *IBM Tivoli Monitoring Troubleshooting Guide* for general troubleshooting information.

Table 17. Tivoli Enterprise Portal problems and solutions

Problem	Solution
Historical data collection is unavailable because of incorrect queries in the Tivoli Enterprise Portal.	The column, Sort By, Group By, and First/Last functions are not compatible with the historical data collection feature. Use of these advanced functions will make a query ineligible for historical data collection. Even if data collection has been started, you cannot use the time span feature if the query for the chart or table includes any column functions or advanced query options (Sort By, Group By, First / Last). To ensure support of historical data collection, do not use the Sort By, Group By, or First/Last functions in your queries. See the *IBM Tivoli Monitoring Administrator's Guide* the Tivoli Enterprise Portal online Help for information on the Historical Data Collection function.
When you use a long process name in the situation, the process name is truncated.	Truncation of process names in the portal display is the expected behavior. 64 bytes is the maximum name length.
In the Network Info workspace, the value of the **listener port** column is 0 or -1.	Set the svcename parameter as db2c_*Instance* in the database manager configuration of the corresponding DB2 database instance. Where *Instance* is the name of the DB2 agent instance. Add the following line in the services file of the DB2 database instance: `db2c_Instance PortNumber/tcp` The services file is located at the following place: • Windows systems: `%SystemRoot%\system32\drivers\etc\services` • UNIX and Linux systems: `/etc/services` After these changes take effect, the specified *PortNumber* is displayed in the Network Info workspace.

Troubleshooting for remote deployment

Table 18 on page 378 lists problems that might occur with remote deployment. This section provides information about troubleshooting remote deployment of the DB2 agent. See the *IBM Tivoli Monitoring Troubleshooting Guide* for general troubleshooting information.

This section describes problems and solutions for remote deployment and removal of agent software Agent Remote Deploy:

Table 18. Remote deployment problems and solutions

Problem	**Solution**

Remote deploying an instance of the agent on a Windows 64 bit system in a ITM623FP1 environment using the command line failed with the following error message: `KDY0005E The component UD is not installed on AC. The Agent HOSTNAME bundle requires that this component be present for the installation to proceed.`	1. On the TEMS machine where the DB2 agent bundle was added, add the KAC bundle version 062301000 into the TEMS depot from ITM 6.2.3 FP1 from the WINDOWS images. For example:
	• TEMS on Windows :
	`C:\IBM\ITM> tacmd addbundles -t AC -i C:\images\ITM623FP1\WINDOWS\Deploy`
	• TEMS on Linux :
	`/opt/IBM/ITM/bin/tacmd addbundles -t AC -i /images/ITM623FP1/WINDOWS/Deploy`
	2. (Optional) check if there is an orphaned process named `SETUP.EXE` running on the OS agent machine and kill the process.
	3. run the tacmd command to remote deploy the DB2 agent again.

After upgrading the agent from version 6.22 fix pack 1 to version 7.1 on a Linux or a UNIX system, you restart remote instances of the agent which connect to ITM version 6.22 fix pack 2 components. The remote instances of the agent are restarted using the root ID which does not have the authority to connect to DB2 instances.	You have to define the correct user ID's for the agent instances in the kcirunas.cfg file. For more information, see http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/topic/com.ibm.itm.doc_6.2.3/itm623_install226.htm
While you are using the remote deployment feature to install DB2 agent, an empty command window is displayed on the target computer. This problem occurs when the target of remote deployment is a Windows computer. (See the *IBM Tivoli Monitoring Installation and Setup Guide* for more information on the remote deployment feature.)	Do not close or modify this window. It is part of the installation process and will be dismissed automatically.
The removal of a monitoring agent fails when you use the remote removal process in the Tivoli Enterprise Portal desktop or browser.	This problem might happen when you attempt the remote removal process immediately after you have restarted the Tivoli Enterprise Monitoring Server. You must allow time for the DB2 agent to refresh its connection with the Tivoli Enterprise Monitoring Server before you begin the remote removal process.
You have configured and attempted to deploy a remote instance of the agent and the operation has failed. However the Agent Event workspace does not display information explaining why the deployment operation failed.	There are a number of reasons that could result in this problem, to solve the problem you can attempt the following: • On Unix or Linux systems, verify that the DB2 instance to be monitored and the username entered as the DB2 instance owner under **Run As** both exist on the remote machine. • On Windows systems, verify that you are not trying to deploy a 32 bit agent instance to monitor a 64 bit DB2 instance and that no agent instance of the same name is already deployed to the remote machine. If this problem persists, check the ITM logs for more information.

Workspace troubleshooting

Table 19 shows problems that might occur with workspaces. This chapter provides agent-specific troubleshooting information. See the *IBM Tivoli Monitoring Troubleshooting Guide* for general troubleshooting information.

Table 19. Workspace problems and solutions

Problem	Solution
The name of the attribute does not display in a bar chart or graph view.	When a chart or graph view that includes the attribute is scaled to a small size, a blank space is displayed instead of a truncated name. To see the name of the attribute, expand the view of the chart until there is sufficient space to display all characters of the attribute's name.
At the bottom of each view, you see the following Historical workspace KFWITM220E error: **Request failed during execution**.	Ensure that you configure all groups that supply data to the view. In the Historical Configuration view, ensure that data collection is started for all groups that supply data to the view.
You start collection of historical data but the data cannot be seen.	Managing options for historical data collection: • Basic historical data collection populates the Warehouse with raw data. This type of data collection is turned off by default. See Chapter 2, "Agent-specific installation and configuration for the monitoring agent," on page 7 for information on managing this feature including how to set the interval at which data is collected. By setting a more frequent interval for data collection you reduce the load on the system incurred every time data is uploaded. • You use the Summarization and Pruning monitoring agent to collect specific amounts and types of historical data. Be aware that historical data is not displayed until the Summarization and Pruning monitoring agent begins collecting the data. By default, this agent begins collection at 2 AM daily. At that point, data is visible in the workspace view. See the IBM Tivoli Monitoring Administrator's Guide to learn how to modify the default collection settings.
No data is displayed in the Database workspace.	The Database workspace only shows the active databases, which are the databases for which an active connection exists. The live external connection might be the result of an application connecting or the result of a db2 connect. Also, monitoring only occurs for active databases. When the connection becomes inactive, the database disappears from the Database workspace, and monitoring stops for this database.
DB2 workspaces become empty with a busy hourglass for a while, when the DB2 agent is monitoring a partitioned DB2 database.	This problem occurs because a DB2 partition is unavailable, and there isDB2 time-out and retry activities. When a partition is unavailable, the duration after which the error status is returned is determined by the Fast Communications Manager (FCM) settings, which include the max_connretries parameter (Node connection retries) with the default value of 5 retries, and the conn_elapse parameter (Connection elapse time) with the default value of 10 seconds. When the default values of the two parameters are used, there is a 50-second delay for each DB2 API call while there are a lot of API requests for the agent to make. In this period of time, the workspaces are empty with a busy hourglass. To reduce the empty workspace time, you can set a smaller value for the max_connretries parameter or the conn_elapse parameter.

Situation troubleshooting

This section provides information about both general situation problems and problems with the configuration of situations. See the *IBM Tivoli Monitoring Troubleshooting Guide* for more information about troubleshooting for situations.

General situation problems

Table 20 lists problems that might occur with specific situations.

Table 20. Specific situation problems and solutions

Problem	Solution
You want to change the appearance of situations when they are displayed in a Workspace view.	1. Right-click an item in the Navigation tree, and click **Situations**. 2. In the Situation Editor window, select the situation that you want to modify. 3. Use the **Status** pull-down menu in the lower right of the window to set the status and appearance of the Situation when it triggers. **Important:** This status setting is not related to severity settings in IBM Tivoli Enterprise Console.
Monitoring activity requires too much disk space.	Check the RAS trace logging settings that are described in "Setting RAS trace parameters" on page 357. For example, trace logs grow rapidly when you apply the **ALL** logging option.
Monitoring activity requires too many system resources.	Table 21 on page 381 describes the performance impact of specific attribute groups. If possible, decrease your use of the attribute groups that require greater system resources.
A formula that uses mathematical operators appears to be incorrect. For example, if you were monitoring Linux, a formula that calculates when **Free Memory** falls under 10 percent of **Total Memory** does not work: `LT #'Linux_VM_Stats.Total_Memory' / 10`	This formula is incorrect because situation predicates support only logical operators. Your formulas cannot have mathematical operators. **Tip:** The Situation Editor provides alternatives to math operators. Regarding the example, you can select **% Memory Free** attribute and avoid the need for math operators.
If you are running a version of the DB2 agent prior to 6.2, and you choose to alter the views to include a new attribute, data for this attribute is not displayed and you see a blank column in this view.	To enable Unicode and other features, upgrade the DB2 agent to IBM Tivoli Monitoring, Version 6.2.0.

You see the 'Unable to get attribute name' error in the Tivoli Enterprise Monitoring Server log after creating a situation.	Ensure that the agent attribute files are installed on the Tivoli Enterprise Monitoring Server.
	
	The following example shows a typical log entry when you have this problem:
	`(4320916A.0049-F60:kfaottev.c,1572,"Translate_ResultBuffer") \`
	`  Unable to get attribute name for tablename/column \`
	`  <UAG524400.UA4>. Ignored.`

Situations are triggered in the Tivoli Enterprise Monitoring Server, but events for the situation are not sent to the Tivoli Enterprise Console server. The Tivoli Enterprise Monitoring Server is properly configured for event forwarding, and events for many other situations are sent to the event server.	None. This is a limitation of the Tivoli Enterprise Monitoring Server event forwarding function. Situations that only monitor other situations do not send events to the event server. This condition can occur when a situation is only monitoring the status of other situations. The event forwarding function requires an attribute group reference in the situation in order to determine the correct event class to use in the event. When the situation only monitors other situations, no attribute groups are defined and the event class cannot be determined. Because the event class cannot be determined, no event is sent.

Consider performance impact of each attribute group: Table 21 lists the impact on performance (high, medium, or low) of each attribute group. The multiple-instance attributes have been classified at the lowest level. That is, the performance overhead will increase if you do not specify compare values for one or more key values.

When you want to prevent impact on performance by any of the attribute groups listed in Table 21 you must avoid referencing that attribute group, as suggested in this list:

- Disable the attribute group.
- Never select workspaces that reference the attribute group.
- Disable situations that reference the attribute group by using the "Undistributed situations" option in the Situation Editor.
- Disable historical reporting that references the attribute group.
- Avoid using the "Auto Refresh" refresh feature in a Workspace because this option causes a refresh of data for all attribute groups.

See the *IBM Tivoli Monitoring User's Guide* for additional information on controlling attribute group usage.

Table 21. Performance impact by attribute group

Attribute group	High	Medium	Low
Application00 (KUD_DB2_Application00	✓		
Application00 (KUDDB2APPLGROUP00) (Superseded)	✓		
Application00U (KUDDB2APPLGROUP00_U) (Superseded)	✓		
Application01 (KUD_DB2_Application01)	✓		
Application01 (KUDDB2APPLGROUP01) (Superseded)	✓		
Apply Program (KUD_DB2_Apply_Program)		✓	
Apply Subscription (KUD_DB2_Apply_Subscription)		✓	
Buffer Pool (KUD_DB2_Buffer_Pool)		✓	
Buffer Pool (KUDBUFFERPOOL00) (Superseded)		✓	
Database00 (KUD_DB2_Database00)	✓		
Database00 (KUDDBASEGROUP00) (Superseded)	✓		
Database01 (KUD_DB2_Database01)	✓		
Database01 (KUDDBASEGROUP01) (Superseded)	✓		
Database02 (KUD_DB2_Database02)	✓		
DCS Database (KUD_DB2_DCS_Database)			✓
Diagnostic Log (KUD_DB2_Diagnostic_Log)			✓
Diagnostic Messages (KUD_DB2_Diagnostic_Messages) (Superseded)			✓
Locking Conflict (KUDLOCKCONFLICT00)		✓	

Table 21. Performance impact by attribute group (continued)

Attribute group	High	Medium	Low
Log (KUD_DB2_LOG)	✓		
Log Record (KUD_DB2_LOG_RECORD)		✓	
Network Info (KUD_DB2_IPADDR_TABLE)			✓
System Overview (KUD_DB2_System_Overview)		✓	
System Overview (KUDINFO00) (Superseded)		✓	
System Resources (KUD_DB2_System_Resources)			✓
Table (KUD_DB2_Table)	✓		
Tablespace (KUD_DB2_Tablespace)		✓	
Tablespace (KUDTABSPACE) (Superseded)		✓	

Problems with configuration of situations

Table 22 lists problems that might occur with situations.

This section provides information for troubleshooting for agents. Be sure to consult the *IBM Tivoli Monitoring Troubleshooting Guide* for more general troubleshooting information.

Table 22. Problems with configuring situations that you solve in the Situation Editor

Problem	Solution
Remember: To get started with the solutions in this section, perform these steps: 1. Launch the Tivoli Enterprise Portal. 2. Click **Edit > Situation Editor**. 3. In the tree view, choose the agent whose situation you want to modify. 4. Choose the situation in the list. The Situation Editor view is displayed.	
The situation for a specific agent is not visible in the Tivoli Enterprise Portal.	Open the Situation Editor. Access the All managed servers view. If the situation is absent, confirm that application support for DB2 agent has been added to the monitoring server. If not, add application support to the server, as described in the *IBM Tivoli Monitoring Installation and Setup Guide.*
The monitoring interval is too long.	Access the Situation Editor view for the situation that you want to modify. Check the **Sampling interval** area in the **Formula** tab. Adjust the time interval as needed.
The situation did not activate at startup.	Recycle the situation as follows: 1. Right-click the situation and choose **Stop Situation**. 2. Right-click the situation and choose **Start Situation**. **Tip:** You can permanently avoid this problem by placing a check mark in the **Run at Startup** option of the Situation Editor view for a specific situation.
The situation is not displayed.	Click the **Action** tab and check whether the situation has an automated corrective action. This action can occur directly or through a policy. The situation might be resolving so quickly that you do not see the event or the update in the graphical user interface.
An Alert event has not occurred even though the predicate has been properly specified.	Check the logs, reports, and workspaces.
A situation fires on an unexpected managed object.	Confirm that you have distributed and started the situation on the correct managed system.

Table 22. Problems with configuring situations that you solve in the Situation Editor (continued)

Problem	Solution
The product did not distribute the situation to a managed system.	Click the **Distribution** tab and check the distribution settings for the situation.
The situation does not fire. Incorrect predicates are present in the formula that defines the situation. For example, the managed object shows a state that normally triggers a monitoring event, but the situation is not true because the wrong attribute is specified in the formula.	In the **Formula** tab, analyze predicates as follows: 1. Click the *fx* icon in the upper-right corner of the Formula area. The Show formula window is displayed. a. Confirm the following details in the **Formula** area at the top of the window: • The attributes that you intend to monitor are specified in the formula. • The situations that you intend to monitor are specified in the formula. • The logical operators in the formula match your monitoring goal. • The numerical values in the formula match your monitoring goal. b. (*Optional*) Click the **Show detailed formula** check box in the lower left of the window to see the original names of attributes in the application or operating system that you are monitoring. c. Click **OK** to dismiss the Show formula window. 2. (*Optional*) In the Formula area of the **Formula** tab, temporarily assign numerical values that will immediately trigger a monitoring event. The triggering of the event confirms that other predicates in the formula are valid. **Remember:** After you complete this test, you must restore the numerical values to valid levels so that you do not generate excessive monitoring data based on your temporary settings.

Table 23. Problems with configuration of situations that you solve in the Workspace area

Problem	Solution
Situation events are not displayed in the Events Console view of the workspace.	Associate the situation with a workspace. The situation does not need to be displayed in the workspace. It is sufficient that the situation is associated with any workspace.
You do not have access to a situation.	**Important:** You must have administrator privileges to perform these steps. 1. Select **Edit** > **Administer Users** to access the Administer Users window. 2. In the Users area, select the user whose privileges you want to modify. 3. In the Permissions tab, Applications tab, and Navigator Views tab, select the permissions or privileges that correspond to the user's role. 4. Click **OK**.
A managed system seems to be offline.	1. Select Physical View and highlight the Enterprise Level of the navigator tree. 2. Select **View** > **Workspace** > **Managed System Status** to see a list of managed systems and their status. 3. If a system is offline, check network connectivity and status of the specific system or application.

Table 24. Problems with configuration of situations that you solve in the Manage Tivoli Enterprise Monitoring Services window

Problem	Solution
After an attempt to restart the agents in the Tivoli Enterprise Portal, the agents are still not running.	For UNIX, NetWare, or Windows systems, log on to the applicable system and perform the appropriate queries.

Table 24. Problems with configuration of situations that you solve in the Manage Tivoli Enterprise Monitoring Services window (continued)

Problem	Solution
The Tivoli Enterprise Monitoring Server is not running.	Check the system status and check the appropriate IBM Tivoli Monitoring logs.
The managed objects you created are firing on incorrect managed systems.	Check the managed system distribution on both the situation and the managed object settings sheets.
For agents that can have multiple subnodes, such as database agents:	
The icon is incorrect.	Check the icon assignments in the template.
The situation is not assigned to a state in the template.	Check the situation assignments in the template of the associated managed object.
You assigned the situation to an incorrect state in the template.	Check the State settings sheet for the template.

Take Action command troubleshooting

Table 25 lists general problems that might occur with Take Action commands. When each Take Action command runs it generates the log file listed in Table 11 on page 355. This chapter provides agent-specific troubleshooting information. See the *IBM Tivoli Monitoring Troubleshooting Guide* for general troubleshooting information.

Table 25. Take Action commands problems and solutions

Problem	Solution
Some Take Action commands require several minutes to complete.	Allow several minutes. If you do not see a pop-up message advising you of completion, try to run the command manually. Consult the log file for the Take Action command, `name.log`, where *name* is the name of the Take Action command.
Take Action commands work only when all the argument values are specified.	Set the values of the arguments that you do not need as `None`. Do not left any argument value blank.

Support information

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

Online

The following sites contain troubleshooting information:

- Go to the IBM Software Support site at http://www.ibm.com/software/support/probsub.html and follow the instructions.
- Go to the IBM Tivoli Distributed Monitoring and Application Management Wiki at http://www.ibm.com/developerworks/wikis/display/tivolimonitoring/Home. Feel free to contribute to this wiki.

IBM Support Assistant

The IBM Support Assistant (ISA) is a free local software serviceability workbench that helps you resolve questions and problems with IBM software products. The ISA provides quick access to support-related information and serviceability tools for problem determination. To install the ISA software, go to http://www.ibm.com/software/support/isa.

Appendix A. Upgrading for warehouse summarization

The DB2 agent made changes to the warehouse collection and summarization characteristics for some agent attribute groups. These changes correct and improve the way warehouse data is summarized, producing more meaningful historical reports. This appendix explains those changes and the implications to your warehouse collection and reporting.

Warehouse summarization is controlled on a per-table basis. How the rows in each table are summarized is determined by a set of attributes in each table that are designated as primary keys. There is always one primary key representing the monitored resource, and data is minimally summarized based on this value. For all agents, this primary key is represented internally by the column name, ORIGINNODE; however, the external attribute name varies with each monitoring agent.

One or more additional primary keys are provided for each attribute group to further refine the level of summarization for that attribute group. For example, in an OS agent disk attribute group, a primary key might be specified for the logical disk name that allows historical information to be reported for each logical disk in a computer.

Tables in the warehouse

For a monitoring agent, there are two main types of warehouse tables:

- Raw tables:

  These tables contain the raw information reported by a monitoring agent and written to the warehouse by the Warehouse Proxy agent. Raw tables are named for the attribute group that they represent, for example, KUDBUFFERPOOL00.
- Summary tables:

  These tables contain summarized information based on the raw tables and written to the warehouse by the Summarization and Pruning agent. Summarization provides aggregation results over various reporting intervals, for example, hours, or days. Summary table names are based on the raw table name with an appended suffix, for example, KUDBUFFERPOOL00_H, KUDBUFFERPOOL00_D.

Effects on summarized attributes

When tables are summarized in the warehouse, the summary tables and summary views are created to include additional columns to report summarization information. Table 26 contains a list of the time periods and the suffixes for the summary tables and views.

Table 26. Time periods and suffixes for summary tables and views

Data collection time period	Summary table suffixes	Summary view suffixes
Hourly	_H	_HV
Daily	_D	_DV
Weekly	_W	_WV
Monthly	_M	_MV

Table 26. Time periods and suffixes for summary tables and views (continued)

Data collection time period	Summary table suffixes	Summary view suffixes
Quarterly	_Q	_QV
Yearly	_Y	_YV

Table 27 shows the expansion to summary columns of some of the most commonly used attribute types.

Table 27. Additional columns to report summarization information

Attribute name	Aggregation type	Additional summarization columns
MyGauge	GAUGE	MIN_MyGauge MAX_MyGauge SUM_MyGauge AVG_MyGauge
MyCounter	COUNTER	TOT_MyCounter HI_MyCounter LO_MyCounter LAT_MyCounter
MyProperty	PROPERTY	LAT_Property

These additional columns are provided only for attributes that are not primary keys. In the cases when an existing attribute is changed to be a primary key, the Summarization and Pruning agent no longer creates summarization values for the attributes, but the previously created column names remain in the table with any values already provided for those columns. These columns cannot be deleted from the warehouse database, but as new data is collected, these columns will not contain values. Similarly, when the primary key for an existing attribute has its designation removed, that attribute has new summarization columns automatically added. As new data is collected, it is used to populate these new column values, but any existing summarization records do not have values for these new columns.

The overall effect of these primary key changes is that summarization information is changing. If these changes result in the old summarization records no longer making sense, you can delete them. As a part of warehouse upgrade, summary views are dropped. The views will be recreated by the Summarization and Pruning agent the next time it runs. Dropping and recreating the views ensure that they reflect the current table structure.

Upgrading your warehouse with limited user permissions

The IBM Tivoli Monitoring warehouse agents (Warehouse Proxy and Summarization and Pruning agents) can dynamically adjust warehouse table definitions based on attribute group and attribute information being loaded into the warehouse. These types of table changes must be done for this monitoring agent for one or both of the following conditions:

- The DB2 agent has added new attributes to an existing attribute group and that attribute group is included in the warehouse.
- The DB2 agent has added a new attribute group and that attribute group is included in the warehouse.

For the warehouse agents to automatically modify the warehouse table definitions, they must have permission to alter warehouse tables. You might not have granted these agents these permissions, choosing instead to manually define the raw tables and summary tables needed for the DB2 agents. Or, you might have granted these permissions initially, and then revoked them after the tables were created.

You have two options to effect the required warehouse table changes during the upgrade process:

- Grant the warehouse agents temporary permission to alter tables

  If using this option, grant the permissions, start historical collection for all the desired tables, allow the Warehouse Proxy agent to add the new data to the raw tables, and allow the Summarization and Pruning agent to summarize data for all affected tables. Then, remove the permission to alter tables
- Make the warehouse table updates manually

  If using this option, you must determine the table structures for the raw and summary tables. If you manually created the tables in the earlier warehouse definition, you already have a methodology and tools to assist you in this effort. You can use a similar technique to update and add new tables for this warehouse migration.

  For a method of obtaining raw table schema, see the IBM Redbook, *Tivoli Management Services Warehouse and Reporting*, January 2007, SG24-7290. The chapter that explains warehouse tuning includes a section on creating data tables manually.

Types of table changes

The following types of table changes affect warehouse summarization:

Case 1 - New attribute added to an attribute group and defined as a primary key.

Case 2 - Existing attribute defined as a primary key or had primary key designation removed. Case 2 does not apply to the DB2 agent.

Case 3 - Moving some tables from 4K tablespaces to 8K tablespaces when using DB2 as the warehouse database.

Case 1 and Case 2 are primary key changes. In both cases, new summarization records do not match existing summarized data:

- A new attribute is added to an attribute group, and the attribute is defined as a primary key:

  New summarization records provide more accurate summarization or greater granularity than previous records. Existing summarization records are still available but contain less granular detail if default values are not assigned for the new primary keys.
- An existing attribute is defined as a primary key or the primary key designation is removed:

  If a new key is added, the new summarization records provide more accurate summarization or greater granularity than previous records. If an existing key is removed, the new summarization records provide less granularity than previous records, but with the intent of providing more meaningful summarization. Existing summarization records are still available.

Case 3 requires that you move some tables from 4K tablespaces to 8K tablespaces when using DB2 as the warehouse database to avoid errors during summarization and pruning processing.

Table summary

Table 28 provides information to help you determine the effects of primary key and warehouse changes for this DB2 agent. The table shows each attribute group, the current primary keys (in addition to ORIGINNODE) for the attribute group, primary keys that were removed, and whether this table is being included in warehouse reporting.

Table 28. Primary key and warehouse changes for the DB2 agent

Attribute group	Current primary key attributes	Removed primary key attributes	Warehoused
Buffer Pool (KUDBUFFERPOOL00)	db_partition db_name bp_id	None	Yes
Application00U (KUDDB2APPLGROUP00_U)	db_partition appl_id	None	Yes
Application01 (KUDDB2APPLGROUP01)	db_partition appl_name	None	Yes
Database (KUDDBASEGROUP00)	db_partition db_name	None	Yes
Database (KUDDBASEGROUP01)	db_partition db_name	None	Yes
System Overview (KUDINFO00)	db_partition instance_name	None	Yes
Locking Conflict (KUDLOCKCONFLICT00)	db_partition appl_id_holding_lk appl_id	None	Yes
Tablespace (KUDTABSPACE)	db_partition TABLESPACE_NAME DB_NAME	None	Yes
Apply Program (KUD_DB2_Apply_Program)	apply_qualifier	None	Yes
Apply Subscription (KUD_DB2_Apply_Subscription)	target_owner target_table db_name_target	None	Yes
DCS Database (KUD_DB2_DCS_Database)	db_partition db_name_U	None	Yes
Table (KUD_DB2_Table)	table_name_U db_partition db_name_U	None	Yes

Upgrading your warehouse for primary key and tablespace changes

Upgrading your warehouse includes making the following types of changes:

- Case 1 - New attribute is added and is designated as a primary key
  - New attribute and a default value must be added to the raw table and the summarization tables.

    If the attribute group name is not too large for the underlying database, the table name corresponds to the attribute group name. If the attribute group name is too long, a short name is used. The mapping of attribute group names to table names is stored in the WAREHOUSEID table.
  - Case-1 scripts that perform the following actions are provided to assist in this change:
    - Alter existing raw tables
    - Alter existing summary tables
    - Drop existing summary views

- These changes must be done before the DB2 agent is started and begins exporting data to the Warehouse Proxy agent.
- Case 2 - Existing attributes are changed to either add or remove primary key designation.
  - Existing data is of limited value and must be deleted.
  - Case-2_Truncate scripts that perform the following actions are provided to assist in this change:
    - Remove all records from existing summary tables, preserving existing table definitions
    - Delete the raw data marker allowing raw data to be resummarized
  - Case-2_Drop scripts that perform the following actions are provided to assist in this change:
    - Drop existing summary views
    - Drop existing summary tables
    - Delete the raw data marker allowing raw data to be resummarized
  - These changes are optional, but result in more accurate summarized information.
- Case 3 - Move tables from 4K tablespace to 8K tablespace for selected agents
  - Special processing for selected agents, to move tables from a 4K tablespace to an 8K tablespace.
  - Individual scripts are provided for each summary table to be changed.

Affected attribute groups and supporting scripts

Table 29 shows the attribute groups and summary tables affected for this DB2 agent, the names of the SQL scripts provided to assist in the upgrade process, the types of warehouse databases for which the scripts must be run, and the types of changes (cases) to which the scripts apply.

Table 29. Scripts for affected attribute groups and summary tables for the DB2 agent

Attribute group or summary table	File	DB2	Oracle	MS SQL Server	Case 1	Case 2	Case 3
Buffer Pool (KUDBUFFERPOOL00)	kud_61migr_DB2_Agent_Case-1.sql	X	X	X	X		
Application00U (KUDDB2APPLGROUP00_U)	kud_61migr_DB2_Agent_Case-1.sql	X	X	X	X		
Application01 (KUDDB2APPLGROUP01)	kud_61migr_DB2_Agent_Case-1.sql	X	X	X	X		
Database (KUDDBASEGROUP00)	kud_61migr_DB2_Agent_Case-1.sql	X	X	X	X		
KUDDBASEGROUP01_D[1]	kud_61migr_KUDDBASEGROUP01_D.sql	X					X
KUDDBASEGROUP01_H[1]	kud_61migr_KUDDBASEGROUP01_H.sql	X					X
KUDDBASEGROUP01_M[1]	kud_61migr_KUDDBASEGROUP01_M.sql	X					X
KUDDBASEGROUP01_Q[1]	kud_61migr_KUDDBASEGROUP01_Q.sql	X					X
KUDDBASEGROUP01_W[1]	kud_61migr_KUDDBASEGROUP01_W.sql	X					X

Table 29. Scripts for affected attribute groups and summary tables for the DB2 agent (continued)

Attribute group or summary table	File	DB2	Oracle	MS SQL Server	Case 1	Case 2	Case 3
KUDDBASEGROUP01_Y[1]	kud_61migr_ KUDDBASEGROUP01_Y.sql	X					X
Database (KUDDBASEGROUP01)	kud_61migr_DB2_Agent_ Case-1.sql		X	X	X		
System Overview (KUDINFO00)	kud_61migr_DB2_Agent_ Case-1.sql	X	X	X	X		
Locking Conflict (KUDLOCKCONFLICT00)	kud_61migr_DB2_Agent_ Case-1.sql	X	X	X	X		
Tablespace (KUDTABSPACE)	kud_61migr_DB2_Agent_ Case-1.sql	X	X	X	X		
[1] Summary table							

The following types of warehouse objects are affected by these scripts. Review the scripts before running them:

- Case-1.sql

  These scripts affect raw tables, summary tables, and summary views.
- Case-2_Drop.sql

  These scripts affect the summary tables, summary views, and the Summarization and Pruning agent WAREHOUSEMARKER table.
- Case-2_Truncate.sql

  These scripts affect the summary tables and the Summarization and Pruning agent WAREHOUSEMARKER table.

Procedures

The warehouse can be hosted on any of three databases: DB2, Oracle, or Microsoft SQL Server. There are different sets of script files for each type of database. These scripts are provided as part of the DB2 agent Tivoli Enterprise Portal Server support file installation. After installing the Tivoli Enterprise Portal Server support files for the DB2 agent, the files are located on the Tivoli Enterprise Portal Server computer in *install_dir*/CNPS/SQLLIB/WAREHOUSE. There is a subdirectory for each type of database: DB2 for DB2, Oracle for Oracle, and SQLServer for Microsoft SQL Server.

The scripts provide commands for all affected tables and views. If you do not have summarization enabled for some periods, for example, quarterly or yearly, you will not have the corresponding summary tables (_Q, _Y) and summary views (_QV, _YV) in your warehouse database. If you run the scripts that are provided, the database reports errors for these missing objects. The scripts continue to run the remaining commands. Similarly, if you rerun the scripts, all commands are attempted. If the objects do not exist, or the command cannot be run (especially for the ALTER commands), the scripts continue processing the remaining commands.

DB2 warehouse database procedure

1. Stop *all* running Warehouse Proxy agent instances and the Summarization and Pruning agent.
2. Back up your warehouse database.

3. Copy the scripts from the Tivoli Enterprise Portal Server in one of the following directories to a temporary directory on the system where the warehouse database is located:
   - Windows systems:

     install_dir\CNPS\SQLLIB\WAREHOUSE\DB2
   - UNIX and Linux systems:

     install_dir/*arch*/cq/sqllib/WAREHOUSE/DB2
4. On the system where the warehouse database is located, change to the directory where you placed the script files in Step 3. Then, connect to the warehouse database through the DB2 command line with a user ID that has the authorization to alter and load tables and drop views. Run commands based on the following example to connect, set the schema, and save the script to an output file:

   ```
   db2 connect to WAREHOUS user ITMUSER using ITMPASS
   db2 set current schema="ITMUSER"
   db2 -tv -z log/script.sql.log -f script.sql
   ```

   These parameters are used in the example:
   - WAREHOUS is the database name.
   - ITMUSER is the user name used by the Warehouse Proxy agent.
   - ITMPASS is the password used by the Warehouse Proxy agent.
   - *script.sql* is the name of the script file. See Table 29 on page 389 for the script file names.
   - *script.sql*.log is the name of the output file.

   Important: You might receive error messages like the following two messages from the DB2 database:
   - `SQL0204N "schema name.table name" is an undefined name. SQLSTATE=42704`

     This message indicates that the table named *table name* does not exist and cannot be altered or dropped. This happens if you do not have warehousing or summarization enabled for the given table. For example if you only have hourly and daily summarization enabled, you see this message for the weekly, monthly, quarterly, and yearly summarization tables because these tables do not exist.
   - `SQL3304N The table does not exist.`

     This message indicates that the table does not exist and cannot be loaded. This happens if you do not have warehousing or summarization enabled for the given table. For example if you only have hourly and daily summarization enabled, you see this message for the weekly, monthly, quarterly, and yearly summarization tables because these tables do not exist.

Oracle warehouse database procedure

1. Stop *all* running Warehouse Proxy agent instances and the Summarization and Pruning agent.
2. Back up your warehouse database.
3. Copy the scripts from The Tivoli Enterprise Portal Server in one of the following directories to a temporary directory on the system where the warehouse database is located:
   - Windows systems:

install_dir\CNPS\SQLLIB\WAREHOUSE\Oracle

- UNIX and Linux systems:

  install_dir/*arch*/cq/sqllib/WAREHOUSE/Oracle

4. On the system where the warehouse database is located, change to the directory where you placed the script files in Step 3. Then, connect to the warehouse database through the Oracle command line with the same user that the Warehouse Proxy agent uses to connect to the warehouse, and run the script. To run the script, the user ID must have authorization to alter tables and drop views, or to drop tables when using Case 2 Drop, or truncate tables when using Case 2 Truncate. The output is saved to a file named *script name*.log. Run the following command:

```
sqlplus ITMUSER/ITMPASS@WAREHOUS @script.sql
```

These parameters are used in the example:

- WAREHOUS is the connect identifier.
- ITMUSER is the user name used by the Warehouse Proxy agent.
- ITMPASS is the password used by the Warehouse Proxy agent.
- *script.sql* is the name of this script file. See Table 29 on page 389 for the script file names.

Important: You might receive error messages such as the following one from Oracle database:

```
ORA-00942: table or view does not exist
```

This message indicates that the table does not exist and cannot be altered, dropped, or truncated. This happens if you do not have warehousing or summarization enabled for the given table. For example if you only have hourly and daily summarization enabled, you see this message for the weekly, monthly, quarterly, and yearly summarization tables because these tables do not exist.

MS SQL warehouse database procedure

1. Stop *all* running Warehouse Proxy agent instances and the Summarization and Pruning agent.
2. Back up your warehouse database.
3. Copy the scripts from the Tivoli Enterprise Portal Server in the one of the following directories to a temporary directory on the system where the warehouse database is located:
   - Windows systems:

     install_dir\CNPS\SQLLIB\WAREHOUSE\SQLServer
   - UNIX and Linux systems:

     install_dir/*arch*/cq/sqllib/WAREHOUSE/SQLServer
4. On the system where the warehouse database is located, change to the directory where you placed the script files in Step 3. Then, connect to the warehouse database through the SQL Server command line with the same user that the Warehouse Proxy agent uses to connect to the warehouse, and run the script. To run the script, the user ID must have authorization to alter tables and drop views, or to drop tables when using Case 2 Drop, or truncate tables when using Case 2 Truncate. The output is saved to a file named *script name*.log. Run the following command:

```
osql -I -S SQLHOST[\SQLINST] -U ITMUSER -P ITMPASS -d WAREHOUS
   -m-1 -n -o log/script.sql.log -i script.sql
```

These parameters are used in the example:

- WAREHOUS is the database name.
- ITMUSER is the user name used by the Warehouse Proxy agent.
- ITMPASS is the password used by the Warehouse Proxy agent.
- *script.sql* is the name of this script file.
- SQLHOST is the SQL server name.
- SQLINST is the optional SQL instance name.

Important: You might receive error messages from the SQL Server such as the following example:

```
Msg 4902, Level 16, State 1, Server ENTERPRISE,
Line 1 Cannot find the object "table name"
because it does not exist or you do not have permissions.
```

This message indicates that the table named *table name* does not exist and cannot be dropped or truncated. This happens if you do not have warehousing or summarization enabled for the given table. For example if you only have hourly and daily summarization enabled, you see this message for the weekly, monthly, quarterly, and yearly summarization tables because these tables do not exist.

Appendix B. DB2 agent Workspaces

Additional information about the following predefined workspaces is provided here:

- Application Top Ten Summary workspace
- Buffer Pool workspace
- Database workspace

Application Top Ten Summary workspace

The following scenarios describe specific uses for the Application Top Summary workspace and its linked workspaces.

Scenario 1: Appropriate value for LOCK TIMEOUT parameter

The LOCKTIMEOUT configuration parameter specifies the number of seconds that an application waits to obtain a lock. By specifying a maximum value, you can avoid global deadlocks for applications.

If you set this parameter to 0, an application does not wait for a lock. In this case, if a lock is not available at the time of the request, the application receives notification of this immediately.

If you set this parameter to -1, lock timeout detection is turned off. In this case an application waits for a lock (if one is not available at the time of the request) until the lock is granted or a deadlock occurs.

Set the value to detect quickly any waits that are occurring because of an abnormal situation, such as a transaction that is stalled. Set the LOCKTIMEOUT parameter high enough that valid lock requests do not time out because of peak workloads. However, the LOCKTIMEOUT configuration parameter can be set too high, which causes the system to experience too few lock timeouts. In this case, applications might wait excessively to obtain a lock.

You can use the Application Top Summary workspace to help track the number of times an application (connection) experienced a lock timeout. By using the Application Lock Activity workspace, you can view the Lock_Timeout attribute in addition to other lock-related attributes. This attribute indicates the number of times that a request to lock an object timed out instead of being granted. A high value for the Lock_Timeout attribute can be caused by:

- The value of the LOCKTIMEOUT configuration parameter being too low
- An application (transaction) that is holding one or more locks for an extended period.
- A concurrency problem, that can be caused by lock escalations (from the row-level to a table-level).

The Lock_Timeout attribute can help you adjust the setting for the LOCKTIMEOUT configuration parameter. If the number of lock timeouts becomes excessive when compared to normal operating levels, you might have an application that is holding locks for long durations. In this case, this attribute might indicate that you need to analyze some of the other attributes related to locks and deadlocks to determine if you have an application problem.

Scenario 2: Establishing an appropriate amount of sort memory

Online transaction processing (OLTP) applications must not perform large sort operations. Large sort operations are very costly in terms of CPU, I/O, and elapsed time; as a result, sorts can slow down an OLTP application. The default SORTHEAP size is 1 MB (256 4KB pages), which is adequate for most situations.

You can use the information in the Application Top Summary workspace to help you track the number of sort overflows. In the Application Summary workspace, you can view information about the number of sort overflows and the sort overflow percentage. Additionally, you can use the Application Sort/Hash Join Activity workspace to find information about the total number of sorts, the average sort time, the number of sort overflows, and the percentage of sorts that cause an overflow condition.

Sort overflows indicate that large sorts are occurring. If the number of sort overflows represents greater than 3% of sorts, an application might experience serious, unexpected sort problems. You must identify the SQL statements that are causing the sorts and modify the SQL, indexes, or clustering to reduce the cost of the sorts.

Buffer Pool workspace

The following scenario describes a specific use for the Buffer Pool workspace.

Scenario: Monitoring buffer pool efficiency

Database performance and tuning always start with buffer pool efficiency. The buffer pool hit ratio indicates the percentage of time that the database manager did not need to load a page from disk in order to satisfy a page request. That is, the page was already in the buffer pool. The greater the buffer pool hit ratio, the lower the frequency of disk I/O.

If the buffer pool hit ratio is low, the database will experience excessive I/O activity. If this is the case, consider enlarging the buffer pool size for frequently accessed tables or placing the indexes into a separate buffer pool. Buffer pools that are too small result in excessive, unnecessary, physical I/O. Buffer pools that are too large put a system at risk for operating system paging activity.

You can use the information displayed in the Buffer Pool workspace to evaluate many of the characteristics of buffer pool activity. In the associated Buffer Pool Detail workspace, you can evaluate the values of the various attributes related to buffer pool hit ratios, asynchronous and synchronous I/O activity, and extended store and non-buffer pool I/O activity.

Armed with this information, you can identify aspects of buffer pool activity that are outside normal operating levels and take corrective action.

Database workspace

The following scenarios describe specific uses for the Database workspace and its related workspaces.

Scenario 1: Determining the maximum number of open files

The MAXFILOP parameter specifies the maximum number of database files that any single database agent can have open at the same time. If opening a file might cause this value to be exceeded, DB2 closes a file already in use by this agent. If the value of MAXFILOP is too small, DB2 encounters the overhead of opening and closing files so that the system does not exceed this limit. The overhead can become excessive and cause performance degradation. SQL response time can slow considerably. You can monitor the opening and closing of files by using the Database workspace, Database Summary workspace, and Database I/O Activity workspace. In the Buffer Pool Activity area, you can determine the value of the Files Closed attribute.

The Files Closed attribute can help you determine the best value for the MAXFILOP configuration parameter. If the number of files being closed exceeds the norm in your environment, consider increasing the value of MAXFILOP parameter until the opening and closing reaches an acceptable level.

Scenario 2: Setting the value of the LOCKTIMEOUT parameter

The LOCKTIMEOUT configuration parameter specifies the number of seconds that an application will wait to obtain a lock. By specifying a maximum value, you can avoid global deadlocks for applications.

If you set this parameter to 0, an application does not wait for a lock. In this case, if a lock is not available at the time of the request, the application receives notification of this immediately.

If you set this parameter to -1, lock timeout detection is turned off. In this case an application waits for a lock (if one is not available at the time of the request) until the lock is granted or a deadlock occurs.

Set the value to detect quickly any waits that occur because of an abnormal situation, such as a transaction that is stalled. Set the LOCKTIMEOUT parameter high enough that valid lock requests do not time out because of peak workloads. However the LOCKTIMEOUT configuration parameter can be set too high, which causes the system to experience too few lock timeouts. In this case, applications might wait excessively to obtain a lock.

You can use the information in the Database workspace to help you track the number of times that a database detected a timeout situation for all applications that were connected. Use the Database Summary workspace and Database Lock Activity workspace. The Database Lock Activity workspace displays the Lock_Timeout attribute in addition to other lock-related attributes.

The Lock_Timeout attribute indicates the number of times that a request to lock an object timed-out instead of being granted. A high value for this attribute can be caused by:

- The value of the LOCKTIMEOUT configuration parameter being too low
- An application (transaction) that is holding one or more locks for an extended period.
- A concurrency problem, that might be caused by lock escalations (from the row-level to a table-level).

The Lock_Timeout attribute can help you adjust the setting for the LOCKTIMEOUT configuration parameter. If the number of lock timeouts becomes excessive when compared to normal operating levels, one or more applications might be holding locks for long durations. In this case, the Lock_Timeout attribute might indicate that you need to analyze some of the other attributes related to locks and deadlocks to determine if you have an application problem.

Appendix C. IBM Tivoli Enterprise Console event mapping

EIF events specify an event class and the event data is specified as name value pairs that identify the name of an event slot and the value for the slot. An event class can have subclasses. IBM(r)Tivoli(r) Monitoring provides the base event class definitions and a set of base slots that are included in all monitoring events. Agents extend the base event classes to define subclasses that include agent-specific slots. For DB2 agent events, the event classes correspond to the agent attribute groups, and the agent-specific slots correspond to the attributes in the attribute group.

A description of the event slots for each event class is provided in this topic. The situation editor in the Tivoli(r) Enterprise Portal can be used to perform custom mapping of data to EIF slots instead of using the default mapping described in this topic. For more information about EIF slot customization, see the Tivoli(r) Enterprise Portal User's Guide.

Tivoli Enterprise Console(r) requires that event classes and their slots are defined in BAROC (Basic Recorder of Objects in C) files. Each agent provides a BAROC file that contains event class definitions for the agent and is installed on the Tivoli(r) Enterprise Monitoring Server in the `TECLIB` directory `install_dir/cms/TECLIB` for Windows systems and`install_dir/tables/TEMS_hostname/TECLIB` for UNIX systems) when application support for the agent is installed. The BAROC file for the agent and the base BAROC files provided with Tivoli(r) Monitoring must also be installed onto the Tivoli Enterprise Console(r). For details, see "Setting up event forwarding to Tivoli Enterprise Console(r)" in the Tivoli Enterprise Console(r).

Each of the event classes is a child of KUD_Base and is defined in the`kud.baroc` (version 7.1) file. The KUD_Base event class can be used for generic rules processing for any event from the DB2 agent.

Each of the event classes is a child of KUD_Base. The KUD_Base event class can be used for generic rules processing for any event from the DB2 agent.

Table 30. Overview of attribute groups to event classes and slots

Event slots	IBM Tivoli Enterprise Console event class
KUDINFO00 attribute group • node_name: STRING • db2_status: STRING • db2_status_enum: STRING • db2start_time: STRING • last_reset: STRING • snapshot_time: STRING • prdid: STRING • instance_name: STRING • instance_name_enum: STRING • version: STRING • version_enum: STRING • server_db2_type: STRING • server_db2_type_enum: STRING • sort_heap_allocated: INTEGER • sort_heap_allocated_enum: STRING • post_threshold_sorts: INTEGER • post_threshold_sorts_enum: STRING • piped_sorts_requested: INTEGER • piped_sorts_requested_enum: STRING • piped_sorts_accepted: INTEGER • piped_sorts_accepted_enum: STRING • piped_sorts_accepted_pct: INTEGER • piped_sorts_accepted_pct_enum: STRING • rem_cons_in: INTEGER • rem_cons_in_enum: STRING • rem_cons_in_exec: INTEGER • rem_cons_in_exec_enum: STRING • local_cons: INTEGER • local_cons_enum: STRING • local_cons_in_exec: INTEGER • local_cons_in_exec_enum: STRING • con_local_dbases: INTEGER • con_local_dbases_enum: STRING • agents_registered: INTEGER • agents_registered_enum: STRING • agents_waiting_on_token: INTEGER • agents_waiting_on_token_enum: STRING • agents_waiting_on_token_pct: INTEGER • agents_waiting_on_token_pct_enum: STRING • agents_from_pool: INTEGER • agents_from_pool_enum: STRING • agents_created_empty_pool: INTEGER • agents_created_empty_pool_enum: STRING • agents_created_empty_pool_ratio: INTEGER • agents_created_empty_pool_ratio_enum: STRING • coord_agents_top: INTEGER • coord_agents_top_enum: STRING • max_agent_overflows: INTEGER • max_agent_overflows_enum: STRING (Continued on the next page.)	ITM_KUDINFO00

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• agents_stolen: INTEGER • agents_stolen_enum: STRING • agents_registered_top: INTEGER • agents_registered_top_enum: STRING • agents_waiting_top: INTEGER • agents_waiting_top_enum: STRING • comm_private_mem: INTEGER • comm_private_mem_enum: STRING • idle_agents: INTEGER • idle_agents_enum: STRING • gw_total_cons: INTEGER • gw_total_cons_enum: STRING • gw_cur_cons: INTEGER • gw_cur_cons_enum: STRING • gw_cons_wait_host: INTEGER • gw_cons_wait_host_enum: STRING • gw_cons_wait_client: INTEGER • gw_cons_wait_client_enum: STRING • post_threshold_hash_joins: INTEGER • piped_sort_hit_ratio_pct_for_int: INTEGER • agentpri: INTEGER • agentpri_enum: STRING • aslheapsz: INTEGER • aslheapsz_enum: STRING • fcm_num_anchors: INTEGER • fcm_num_anchors_enum: STRING • fcm_num_buffers: INTEGER • fcm_num_buffers_enum: STRING • fcm_num_connect: INTEGER • fcm_num_connect_enum: STRING • fcm_num_rqb: INTEGER • fcm_num_rqb_enum: STRING • maxagents: INTEGER • maxagents_enum: STRING • max_coordagents: INTEGER • max_coordagents_enum: STRING • maxcagents: INTEGER • maxcagents_enum: STRING • mon_heap_sz: INTEGER • mon_heap_sz_enum: STRING • query_heap_sz: INTEGER • query_heap_sz_enum: STRING • rqrioblk: INTEGER • rqrioblk_enum: STRING • piped_sorts_rejected_pct_for_int: INTEGER • piped_sorts_rejected_pct_for_int_enum: STRING • piped_sorts_rejected_pct_for_int: INTEGER • piped_sorts_rejected_pct_for_int_enum: STRING • sheapthres: INTEGER • sheapthres_enum: STRING (Continued on the next page.)	ITM_KUDINFO00 (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• connection_status: INTEGER • connection_status_enum: STRING • buff_free: INTEGER • buff_free_enum: STRING • ce_free: INTEGER • ce_free_enum: STRING • rb_free: INTEGER • rb_free_enum: STRING • connection_status: INTEGER • connection_status_enum: STRING • buff_free: INTEGER • buff_free_enum: STRING • ce_free: INTEGER • ce_free_enum: STRING • rb_free: INTEGER • rb_free_enum: STRING • buff_free_bottom: INTEGER • buff_free_bottom_enum: STRING • ce_free_bottom: INTEGER • ce_free_bottom_enum: STRING • ma_free_bottom: INTEGER • ma_free_bottom_enum: STRING • rb_free_bottom: INTEGER • rb_free_bottom_enum: STRING • buf_used_pct: INTEGER • buf_used_pct_enum: STRING • rb_used_pct: INTEGER • rb_used_pct_enum: STRING • ce_used_pct: INTEGER • ce_used_pct_enum: STRING • buff_max_used_pct: INTEGER • buff_max_used_pct_enum: STRING • ce_max_used_pct: INTEGER • ce_max_used_pct_enum: STRING • ma_max_used_pct: INTEGER • ma_max_used_pct_enum: STRING • rb_max_used_pct: INTEGER • rb_max_used_pct_enum: STRING • total_buffers_rcvd: INTEGER • total_buffers_rcvd_enum: STRING • total_buffers_sent: INTEGER • total_buffers_sent_enum: STRING • piped_sorts_rejected_for_int: INTEGER • piped_sorts_rejected_for_int_enum: STRING • dbpg_node_status: STRING • instance_name_u: STRING • instance_name_u_enum: STRING • comm_private_mem_enum: STRING • db2_avail: INTEGER • db2start_time_timestamp: STRING • last_reset_timestamp: STRING • snapshot_time_timestamp: STRING • comm_private_mem_kb: INTEGER • comm_private_mem_kb_enum: STRING (Continued on the next page.)	ITM_KUDINFO00 (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• cons_in_exec_pct: INTEGER • cons_in_exec_pct_enum: STRING • db_partition: STRING • db_partition_enum: STRING • sort_heap_used_pct: INTEGER • sort_heap_used_pct_enum: STRING	ITM_KUDINFO00 (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
KUDDBASEGROUP00 attribute group • node_name: STRING • db_name: STRING • input_db_alias: STRING • db_path: STRING • dbase_status: STRING • dbase_status_enum: STRING • db_conn_time: STRING • last_backup: STRING • total_cons: INTEGER • total_cons_enum: STRING • appls_cur_cons: INTEGER • appls_cur_cons_enum: STRING • snapshot_time: STRING • int_rollbacks: INTEGER • int_rollbacks_enum: STRING • int_deadlock_rollbacks: INTEGER • int_deadlock_rollbacks_enum: STRING • rollback_sql_stmts: INTEGER • rollback_sql_stmts_enum: STRING • failed_sql_stmts: INTEGER • failed_sql_stmts_enum: STRING • prefetch_wait_time: INTEGER • prefetch_wait_time_enum: STRING • agents_top: INTEGER • agents_top_enum: STRING • coord_agents_top: INTEGER • coord_agents_top_enum: STRING • db_location: STRING • db_location_enum: STRING • server_platform: STRING • connections_top: INTEGER • connections_top_enum: STRING • db_heap_top: INTEGER • db_heap_top_enum: STRING • locks_held: INTEGER • locks_held_enum: STRING • lock_waits: INTEGER • lock_waits_enum: STRING • lock_wait_time: INTEGER • lock_wait_time_enum: STRING • lock_list_in_use: INTEGER • lock_list_in_use_enum: STRING • deadlocks: INTEGER • deadlocks_enum: STRING • lock_escals: INTEGER • lock_escals_enum: STRING • x_lock_escals: INTEGER • x_lock_escals_enum: STRING • locks_waiting: INTEGER • locks_waiting_enum: STRING • lock_timeouts: INTEGER • lock_timeouts_enum: STRING • avg_lock_wait_time: INTEGER • avg_lock_wait_time_enum: STRING (Continued on the next page.)	ITM_KUDDBASEGROUP00

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• avg_lock_wait_time: INTEGER • avg_lock_wait_time_enum: STRING • sort_heap_allocated: INTEGER • sort_heap_allocated_enum: STRING • total_sorts: INTEGER • total_sorts_enum: STRING • total_sort_time: INTEGER • total_sort_time_enum: STRING • sort_overflows: INTEGER • sort_overflows_enum: STRING • active_sorts: INTEGER • active_sorts_enum: STRING • avg_sort_time: INTEGER • avg_sort_time_enum: STRING • sort_overflows_pct: INTEGER • sort_overflows_pct_enum: STRING • pool_data_l_reads: INTEGER • pool_data_l_reads_enum: STRING • pool_data_p_reads: INTEGER • pool_data_p_reads_enum: STRING • pool_data_writes: INTEGER • pool_data_writes_enum: STRING • pool_index_l_reads: INTEGER • pool_index_l_reads_enum: STRING • pool_index_p_reads: INTEGER • pool_index_p_reads_enum: STRING • pool_index_writes: INTEGER • pool_index_writes_enum: STRING • pool_read_time: INTEGER • pool_read_time_enum: STRING • pool_write_time: INTEGER • pool_write_time_enum: STRING • files_closed: INTEGER • files_closed_enum: STRING • pool_async_index_reads: INTEGER • pool_async_index_reads_enum: STRING • pool_data_to_estore: INTEGER • pool_data_to_estore_enum: STRING • pool_index_to_estore: INTEGER • pool_index_to_estore_enum: STRING • pool_index_from_estore: INTEGER • pool_index_from_estore_enum: STRING • pool_data_from_estore: INTEGER • pool_data_from_estore_enum: STRING • pool_async_data_reads: INTEGER • pool_async_data_reads_enum: STRING • pool_async_data_writes: INTEGER • pool_async_data_writes_enum: STRING • pool_async_index_writes: INTEGER • pool_async_index_writes_enum: STRING • pool_async_read_time: INTEGER • pool_async_read_time_enum: STRING • pool_async_write_time: INTEGER • pool_async_write_time_enum: STRING (Continued on the next page.)	ITM_KUDDBASEGROUP00 (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• pool_async_data_read_reqs: INTEGER • pool_async_data_read_reqs_enum: STRING • pool_lsn_gap_clns: INTEGER • pool_lsn_gap_clns_enum: STRING • pool_drty_pg_steal_clns: INTEGER • pool_drty_pg_steal_clns_enum: STRING • pool_drty_pg_thrsh_clns: INTEGER • pool_drty_pg_thrsh_clns_enum: STRING • pool_total_reads: INTEGER • pool_total_reads_enum: STRING • pool_hit_ratio: INTEGER • pool_hit_ratio_enum: STRING • avg_pool_read_time: INTEGER • avg_pool_read_time_enum: STRING • pool_total_writes: INTEGER • pool_total_writes_enum: STRING • avg_pool_write_time: INTEGER • avg_pool_write_time_enum: STRING • pool_sync_data_reads: INTEGER • pool_sync_data_reads_enum: STRING • pool_sync_index_reads: INTEGER • pool_sync_index_reads_enum: STRING • pool_sync_read: INTEGER • pool_sync_read_enum: STRING • pool_sync_data_writes: INTEGER • pool_sync_data_writes_enum: STRING • pool_sync_index_writes: INTEGER • pool_sync_index_writes_enum: STRING • pool_sync_write: INTEGER • pool_sync_write_enum: STRING • pool_sync_read_time: INTEGER • pool_sync_read_time_enum: STRING • avg_sync_read_time: INTEGER • avg_sync_read_time_enum: STRING • pool_sync_write_time: INTEGER • pool_sync_write_time_enum: STRING • avg_sync_write_time: INTEGER • avg_sync_write_time_enum: STRING • avg_data_page_read_per_async_req: INTEGER • avg_data_page_read_per_async_req_enum: STRING • direct_reads: INTEGER • direct_reads_enum: STRING • direct_writes: INTEGER • direct_writes_enum: STRING • direct_read_reqs: INTEGER • direct_read_reqs_enum: STRING • direct_write_reqs: INTEGER • direct_write_reqs_enum: STRING • direct_read_time: INTEGER • direct_read_time_enum: STRING • direct_write_time: INTEGER • direct_write_time_enum: STRING (Continued on the next page.)	ITM_KUDDBASEGROUP00 (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• commit_sql_stmts: INTEGER • commit_sql_stmts_enum: STRING • dynamic_sql_stmts: INTEGER • dynamic_sql_stmts_enum: STRING • static_sql_stmts: INTEGER • static_sql_stmts_enum: STRING • select_sql_stmts: INTEGER • select_sql_stmts_enum: STRING • ddl_sql_stmts: INTEGER • ddl_sql_stmts_enum: STRING • uid_sql_stmts: INTEGER • uid_sql_stmts_enum: STRING • total_sql_stmts: INTEGER • total_sql_stmts_enum: STRING • sql_stmts_failed_pct: INTEGER • sql_stmts_failed_pct_enum: STRING • sql_stmts_rollback_pct: INTEGER • sql_stmts_rollback_pct_enum: STRING • appl_section_lookups: INTEGER • appl_section_lookups_enum: STRING • appl_section_inserts: INTEGER • appl_section_inserts_enum: STRING • binds_precompiles: INTEGER • binds_precompiles_enum: STRING • total_hash_joins: INTEGER • total_hash_joins_enum: STRING • total_hash_loops: INTEGER • total_hash_loops_enum: STRING • hash_join_overflows: INTEGER • hash_join_overflows_enum: STRING • hash_join_small_overflows: INTEGER • hash_join_small_overflows_enum: STRING • rows_deleted: INTEGER • rows_deleted_enum: STRING • rows_inserted: INTEGER • rows_inserted_enum: STRING • rows_updated: INTEGER • rows_updated_enum: STRING • rows_selected: INTEGER • rows_selected_enum: STRING • total_sec_cons: INTEGER • total_sec_cons_enum: STRING • num_assoc_agents: INTEGER • num_assoc_agents_enum: STRING • catalog_node_name: STRING • catalog_node_name_enum: STRING • sec_log_used_top: INTEGER • sec_log_used_top_enum: STRING • tot_log_used_top: INTEGER • tot_log_used_top_enum: STRING • sec_logs_allocated: INTEGER • sec_logs_allocated_enum: STRING • log_reads: INTEGER • log_reads_enum: STRING (Continued on next page.)	ITM_KUDDBASEGROUP00 (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• log_writes: INTEGER • log_writes_enum: STRING • pkg_cache_lookups: INTEGER • pkg_cache_lookups_enum: STRING • pkg_cache_inserts: INTEGER • pkg_cache_inserts_enum: STRING • pkg_cache_hit_ratio: INTEGER • pkg_cache_hit_ratio_enum: STRING • cat_cache_lookups: INTEGER • cat_cache_lookups_enum: STRING • cat_cache_inserts: INTEGER • cat_cache_inserts_enum: STRING • cat_cache_overflows: INTEGER • cat_cache_overflows_enum: STRING • cat_cache_hit_ratio: INTEGER • cat_cache_hit_ratio_enum: STRING • cat_cache_heap_full: INTEGER • cat_cache_heap_full_enum: STRING • db_name_u: STRING • input_db_alias_u: STRING • db_path_u: STRING • db_partition: STRING • db_partition_enum: STRING • instance_name_u: STRING	ITM_KUDDBASEGROUP00 (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
KUDDB2APPLGROUP00 attribute group • node_name: STRING • agent_id: INTEGER • appl_id: STRING • appl_status: STRING • appl_status_enum: STRING • snapshot_time: STRING • appl_name: STRING • auth_id: STRING • client_prdid: STRING • db_name: STRING • execution_id: STRING • corr_token: STRING • client_platform: STRING • client_platform_enum: STRING • client_protocol: STRING • client_protocol_enum: STRING • locks_held: INTEGER • locks_held_enum: STRING • lock_waits: INTEGER • lock_waits_enum: STRING • lock_wait_time: INTEGER • lock_wait_time_enum: STRING • lock_escals: INTEGER • lock_escals_enum: STRING • x_lock_escals: INTEGER • x_lock_escals_enum: STRING • deadlocks: INTEGER • deadlocks_enum: STRING • uow_lock_wait_time: INTEGER • uow_lock_wait_time_enum: STRING • lock_timeouts: INTEGER • lock_timeouts_enum: STRING • avg_lock_waittime: INTEGER • avg_lock_waittime_enum: STRING • agent_id_holding_lk: INTEGER • agent_id_holding_lk_enum: STRING • appl_id_holding_lk: STRING • appl_id_holding_lk_enum: STRING • lock_mode: STRING • lock_mode_enum: STRING • lock_object_type: STRING • lock_object_type_enum: STRING • lock_wait_start_time: STRING • table_name: STRING • table_schema: STRING • tablespace_name: STRING • pool_data_l_reads: INTEGER • pool_data_l_reads_enum: STRING • pool_data_p_reads: INTEGER • pool_data_p_reads_enum: STRING • pool_data_writes: INTEGER • pool_data_writes_enum: STRING • pool_index_l_reads: INTEGER • pool_index_l_reads_enum: STRING (Continued on next page.)	ITM_KUDDB2APPLGROUP00 (deprecated)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• pool_index_p_reads: INTEGER • pool_index_p_reads_enum: STRING • pool_index_writes: INTEGER • pool_index_writes_enum: STRING • pool_read_time: INTEGER • pool_read_time_enum: STRING • pool_write_time: INTEGER • pool_write_time_enum: STRING • pool_data_to_estore: INTEGER • pool_data_to_estore_enum: STRING • pool_index_to_estore: INTEGER • pool_index_to_estore_enum: STRING • pool_index_from_estore: INTEGER • pool_index_from_estore_enum: STRING • pool_data_from_estore: INTEGER • pool_data_from_estore_enum: STRING • pool_total_reads: INTEGER • pool_total_reads_enum: INTEGER • pool_hit_ratio: INTEGER • pool_hit_ratio_enum: STRING • avg_pool_read_time: INTEGER • avg_pool_read_time_enum: STRING • pool_total_writes: INTEGER • pool_total_writes_enum: STRING • avg_pool_write_time: INTEGER • avg_pool_write_time_enum: STRING • appl_idle_time: INTEGER • appl_idle_time_enum: STRING • agent_usr_cpu_time: STRING • agent_sys_cpu_time: STRING • appl_con_time: STRING • conn_complete_time: STRING • prefetch_wait_time: INTEGER • prefetch_wait_time_enum: STRING • direct_reads: INTEGER • direct_reads_enum: STRING • direct_writes: INTEGER • direct_writes_enum: STRING • direct_read_reqs: INTEGER • direct_read_reqs_enum: STRING • direct_write_reqs: INTEGER • direct_write_reqs_enum: STRING • direct_read_time: INTEGER • direct_read_time_enum: STRING • direct_write_time: INTEGER • direct_write_time_enum: STRING • stmt_type: STRING • stmt_type_enum: STRING • stmt_operation: STRING • stmt_operation_enum: STRING • cursor_name: STRING • creator: STRING • package_name: STRING • total_sql_stmt: INTEGER • total_sql_stmt_enum: STRING (Continued on the next page)	ITM_KUDDB2APPLGROUP00 (deprecated) (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• failed_sql_stmts: INTEGER • failed_sql_stmts_enum: STRING • int_rollbacks: INTEGER • int_rollbacks_enum: STRING • int_deadlock_rollbacks: INTEGER • int_deadlock_rollbacks_enum: STRING • rollback_sql_stmts: INTEGER • rollback_sql_stmts_enum: STRING • failed_sql_stmts_pct: INTEGER • failed_sql_stmts_pct_enum: STRING • total_sorts: INTEGER • total_sorts_enum: STRING • total_sort_time: INTEGER • total_sort_time_enum: STRING • sort_overflows: INTEGER • sort_overflows_enum: STRING • avg_sort_time: INTEGER • avg_sort_time_enum: STRING • sort_overflows_pct: INTEGER • sort_overflows_pct_enum: STRING • commit_sql_stmts: INTEGER • commit_sql_stmts_enum: STRING • dynamic_sql_stmts: INTEGER • dynamic_sql_stmts_enum: STRING • static_sql_stmts: INTEGER • static_sql_stmts_enum: STRING • select_sql_stmts: INTEGER • select_sql_stmts_enum: STRING • ddl_sql_stmts: INTEGER • ddl_sql_stmts_enum: STRING • uid_sql_stmts: INTEGER • uid_sql_stmts_enum: STRING • binds_precompiles: INTEGER • binds_precompiles_enum: STRING • section_number: INTEGER • section_number_enum: STRING • stmt_start: STRING • stmt_stop: STRING • uow_log_space_used: INTEGER • uow_log_space_used_enum: STRING • uow_comp_status: STRING • uow_comp_status_enum: STRING • prev_uow_stop_time: STRING • uow_start_time: STRING • uow_stop_time: STRING • int_auto_rebinds: INTEGER • int_auto_rebinds_enum: STRING • int_rows_deleted: INTEGER • int_rows_deleted_enum: STRING • int_rows_updated: INTEGER • int_rows_updated_enum: STRING • int_commits: INTEGER • int_commits_enum: STRING • int_rows_inserted: INTEGER • int_rows_inserted_enum: STRING (Continued on the next page.)	ITM_KUDDB2APPLGROUP00 (deprecated) (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• rows_deleted: INTEGER • rows_deleted_enum: STRING • rows_inserted: INTEGER • rows_inserted_enum: STRING • rows_updated: INTEGER • rows_updated_enum: STRING • rows_selected: INTEGER • rows_selected_enum: STRING • rows_read: INTEGER • rows_read_enum: STRING • rows_written: INTEGER • rows_written_enum: STRING • open_loc_curs: INTEGER • open_loc_curs_enum: STRING • open_loc_curs_blk: INTEGER • open_loc_curs_blk_enum: STRING • open_rem_curs: INTEGER • open_rem_curs_enum: STRING • open_rem_curs_blk: INTEGER • open_rem_curs_blk_enum: STRING • rej_curs_blk: INTEGER • rej_curs_blk_enum: STRING • acc_curs_blk: INTEGER • acc_curs_blk_enum: STRING • client_pid: INTEGER • client_pid_enum: STRING • country_code: INTEGER • country_code_enum: STRING • pkg_cache_lookups: INTEGER • pkg_cache_lookups_enum: STRING • pkg_cache_inserts: INTEGER • pkg_cache_inserts_enum: STRING • pkg_cache_hit_ratio: INTEGER • pkg_cache_hit_ratio_enum: STRING • cat_cache_lookups: INTEGER • cat_cache_lookups_enum: STRING • cat_cache_inserts: INTEGER • cat_cache_inserts_enum: STRING • cat_cache_overflows: INTEGER • cat_cache_overflows_enum: STRING • cat_cache_inserts: INTEGER • cat_cache_inserts_enum: STRING • cat_cache_overflows: INTEGER • cat_cache_overflows_enum: STRING • cat_cache_heap_full: INTEGER • cat_cache_heap_full_enum: STRING • cat_cache_hit_ratio: INTEGER • cat_cache_hit_ratio_enum: STRING • total_hash_joins: INTEGER • total_hash_joins_enum: STRING • total_hash_loops: INTEGER • total_hash_loops_enum: STRING • hash_join_overflows: INTEGER • hash_join_overflows_enum: STRING (Continued on the next page.)	ITM_KUDDB2APPLGROUP00 (deprecated) (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• hash_join_small_overflows: INTEGER • hash_join_small_overflows_enum: STRING • query_cost_estimate: INTEGER • query_cost_estimate_enum: STRING • query_card_estimate: INTEGER • query_card_estimate_enum: STRING • degree_parallelism: INTEGER • degree_parallelism_enum: STRING • stmt_text: STRING	ITM_KUDDB2APPLGROUP00 (deprecated) (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
KUDLOCKCONFLICT00 attribute group • node_name: STRING • agent_id: INTEGER • agent_id_enum: STRING • appl_id: STRING • appl_name: STRING • appl_status: STRING • appl_status_enum: STRING • snapshot_time: STRING • auth_id: STRING • client_db_alias: STRING • status_change_time: STRING • codepage_id: INTEGER • codepage_id_enum: STRING • locks_held: INTEGER • locks_held_enum: STRING • lock_wait_time: INTEGER • lock_wait_time_enum: STRING • agent_id_holding_lk: INTEGER • agent_id_holding_lk_enum: STRING • appl_id_holding_lk: STRING • table_name: STRING • table_schema: STRING • tablespace_name: STRING • lock_escalation: STRING • lock_escalation_enum: STRING • lock_wait_start_time: STRING • lock_mode: STRING • lock_mode_enum: STRING • lock_object_type: STRING • lock_object_type_enum: STRING • appl_id_u: STRING • appl_name_u: STRING • auth_id_u: STRING • client_db_alias_u: STRING • appl_id_holding_lk_u: STRING • table_name_u: STRING • table_schema_u: STRING • tablespace_name_u: STRING • snapshot_time_timestamp: STRING • status_change_time_timestamp: STRING • lock_wait_start_time_timestamp: STRING • db_partition: STRING • db_partition_enum: STRING • locks_held_64: REAL • locks_held_64_enum: STRING • lock_wait_time_64: REAL • lock_wait_time_64_enum: STRING	ITM_KUDLOCKCONFLICT00

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
KUDBUFFERPOOL00 attribute group • node_name: STRING • bp_id: STRING • bp_name: STRING • input_db_alias: STRING • db_name: STRING • db_path: STRING • pool_data_l_reads: INTEGER • pool_data_l_reads_enum: STRING • pool_data_p_reads: INTEGER • pool_data_p_reads_enum: STRING • pool_data_writes: INTEGER • pool_data_writes_enum: STRING • pool_index_l_reads: INTEGER • pool_index_l_read:_enum: STRING • pool_index_p_reads: INTEGER • pool_index_p_reads_enum: STRING • pool_index_writes: INTEGER • pool_index_writes_enum: STRING • pool_read_time: INTEGER • pool_read_time_enum: STRING • pool_write_time: INTEGER • pool_write_time_enum: STRING • files_closed: INTEGER • files_closed_enum: STRING • pool_async_index_reads: INTEGER • pool_async_index_reads_enum: STRING • pool_data_to_estore: INTEGER • pool_data_to_estore_enum: STRING • pool_index_to_estore: INTEGER • pool_index_to_estore_enum: STRING • pool_index_from_estore: INTEGER • pool_index_from_estore_enum: STRING • pool_data_from_estore: INTEGER • pool_data_from_estore_enum: STRING • pool_async_data_reads: INTEGER • pool_async_data_reads_enum: STRING • pool_async_data_writes: INTEGER • pool_async_data_writes_enum: STRING • pool_async_index_writes: INTEGER • pool_async_index_writes_enum: STRING • pool_async_read_time: INTEGER • pool_async_read_time_enum: STRING • pool_async_write_time: INTEGER • pool_async_write_time_enum: STRING • pool_async_data_read_reqs: INTEGER • pool_async_data_read_reqs_enum: STRING • pool_total_reads: INTEGER • pool_total_reads_enum: STRING • pool_hit_ratio: INTEGER • pool_hit_ratio_enum: STRING • avg_pool_read_time: INTEGER • avg_pool_read_time_enum: STRING • pool_total_writes: INTEGER • pool_total_writes_enum: STRING (Continued on the next page.)	ITM_KUDBUFFERPOOL00

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• avg_pool_write_time: INTEGER • avg_pool_write_time_enum: STRING • pool_sync_data_reads: INTEGER • pool_sync_data_reads_enum: STRING • pool_sync_index_reads: INTEGER • pool_sync_index_reads_enum: STRING • pool_sync_read: INTEGER • pool_sync_read_enum: STRING • pool_sync_data_writes: INTEGER • pool_sync_data_writes_enum: STRING • pool_sync_index_writes: INTEGER • pool_sync_index_writes_enum: STRING • pool_sync_write: INTEGER • pool_sync_write_enum: STRING • pool_sync_read_time: INTEGER • pool_sync_read_time_enum: STRING • avg_sync_read_time: INTEGER • avg_sync_read_time_enum: STRING • pool_sync_write_time: INTEGER • pool_sync_write_time_enum: STRING • avg_sync_write_time: INTEGER • avg_sync_write_time_enum: STRING • avg_data_page_read_per_async_req: INTEGER • avg_data_page_read_per_async_req_enum: STRING • direct_reads: INTEGER • direct_reads_enum: STRING • pool_total_reads_k: STRING • pool_total_reads_k_enum: STRING • pool_total_writes_k: STRING • pool_total_writes_k_enum: STRING • db_partition: STRING • db_partition_enum: STRING • direct_writes: INTEGER • direct_writes_enum: STRING • direct_read_reqs: INTEGER • direct_read_reqs_enum: STRING • direct_write_reqs: INTEGER • direct_write_reqs_enum: STRING • direct_read_time: INTEGER • direct_read_time_enum: STRING • direct_write_time: INTEGER • direct_write_time_enum: STRING • avg_direct_read_time: INTEGER • avg_direct_read_time_enum: STRING • avg_direct_write_time: INTEGER • avg_direct_write_time_enum: STRING • bp_name_u: STRING • input_db_alias_u: STRING • db_name_u: STRING • db_path_u: STRING (Continued on the next page.)	ITM_KUDBUFFERPOOL00 (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
KUDTABSPACE attribute group • node_name: STRING • db_name: STRING • version: INTEGER • version_enum: STRING • tablespace_name: STRING • tablespace_type: STRING • extent_size: INTEGER • extent_size_enum: STRING • page_size: INTEGER • page_size_enum: STRING • prefetch_size: INTEGER • prefetch_size_enum: STRING • num_containers: INTEGER • num_containers_enum: STRING • container_name: STRING • object_id: INTEGER • object_id_enum: STRING • tablespace_id: INTEGER • tablespace_id_enum: STRING • total_pages: INTEGER • total_pages_enum: STRING • usable_pages: INTEGER • usable_pages_enum: STRING • used_pages: INTEGER • used_pages_enum: STRING • pending_free_pages: INTEGER • pending_free_pages_enum: STRING • free_pages: INTEGER • free_pages_enum: STRING • direct_reads: INTEGER • direct_reads_enum: STRING • direct_writes: INTEGER • direct_writes_enum: STRING • direct_read_reqs: INTEGER • direct_read_reqs_enum: STRING • direct_write_reqs: INTEGER • direct_write_reqs_enum: STRING • pool_data_writes: INTEGER • pool_data_writes_enum: STRING • pool_async_data_reads: INTEGER • pool_async_data_reads_enum: STRING • pool_async_data_writes: INTEGER • pool_async_data_writes_enum: STRING • pool_async_index_reads: INTEGER • pool_async_index_reads_enum: STRING • pool_async_index_writes: INTEGER • pool_async_index_writes_enum: STRING • pool_async_read_time: INTEGER • pool_async_read_time_enum: STRING • pool_async_write_time: INTEGER • pool_async_write_time_enum: STRING • pool_data_l_reads: INTEGER • pool_data_l_reads_enum: STRING • pool_data_from_estore: INTEGER • pool_data_from_estore_enum: STRING (Continued on the next page.)	ITM_KUDTABSPACE

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• pool_data_to_estore: INTEGER • pool_data_to_estore_enum: STRING • pool_data_p_reads: INTEGER • pool_data_p_reads_enum: STRING • pool_index_l_reads: INTEGER • pool_index_l_reads_enum: STRING • pool_index_from_estore: INTEGER • pool_index_from_estore_enum: STRING • pool_index_to_estore: INTEGER • pool_index_to_estore_enum: STRING • pool_index_p_reads: INTEGER • pool_index_p_reads_enum: STRING • pool_index_writes: INTEGER • pool_index_writes_enum: STRING • direct_write_time: INTEGER • direct_write_time_enum: STRING • direct_read_time: INTEGER • direct_read_time_enum: STRING • files_closed: INTEGER • files_closed_enum: STRING • pool_read_time: INTEGER • pool_read_time_enum: STRING • pool_write_time: INTEGER • pool_write_time_enum: STRING • pool_async_data_read_reqs: INTEGER • pool_async_data_read_reqs_enum: STRING • pool_data_reads: INTEGER • pool_data_reads_enum: STRING • pool_sync_idx_writes: INTEGER • pool_sync_idx_writes_enum: STRING • pool_sync_reads: INTEGER • pool_sync_reads_enum: STRING • pool_sync_writes: INTEGER • pool_sync_writes_enum: STRING • avg_sect_read: INTEGER • avg_sect_read_enum: STRING • avg_sect_written: INTEGER • avg_sect_written_enum: STRING • avg_direct_read_time: INTEGER • avg_direct_read_time_enum: STRING • avg_direct_write_time: INTEGER • avg_direct_write_time_enum: STRING • avg_pool_io_time: INTEGER • avg_pool_io_time_enum: STRING • avg_pool_read_time: INTEGER • avg_pool_read_time_enum: STRING • avg_pool_write_time: INTEGER • avg_pool_write_time_enum: STRING • avg_sync_data_read_time: INTEGER • avg_sync_data_read_time_enum: STRING • avg_sync_data_write_time: INTEGER • avg_sync_data_write_time_enum: STRING (Continued on the next page.)	ITM_KUDTABSPACE (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• avg_sync_io_time: INTEGER • avg_sync_io_time_enum: STRING • pool_io_per_sec: INTEGER • pool_io_per_sec_enum: STRING • estore_rw_ratio: INTEGER • estore_rw_ratio_enum: STRING • pool_hit_pct: INTEGER • pool_hit_pct_enum: STRING • pool_idx_hit_pct_for_int: INTEGER • pool_idx_hit_pct_for_int_enum: STRING • prefetch_pct_for_int: INTEGER • prefetch_pct_for_int_enum: STRING • total_io_pct: INTEGER • total_io_pct_enum: STRING • prefetch_reqs_for_int: INTEGER • prefetch_reqs_for_int_enum: STRING • sync_read_time: INTEGER • sync_read_time_enum: STRING • sync_write_time: INTEGER • sync_write_time_enum: STRING • total_direct_io_time: INTEGER • total_direct_io_time_enum: STRING • total_pool_io_time: INTEGER • total_pool_io_time_enum: STRING • total_pool_p_read_time: INTEGER • total_pool_p_read_time_enum: STRING • total_pool_p_write_time: INTEGER • total_pool_p_write_time_enum: STRING • total_sync_io: INTEGER • total_sync_io_enum: STRING • total_sync_io_time: INTEGER • total_sync_io_time_enum: STRING • pool_async_index_read_reqs: INTEGER • pool_async_index_read_reqs_enum: STRING • space_used_dms_table_pct: INTEGER • space_used_dms_table_pct_enum: STRING • space_used_sms_table: INTEGER • space_used_sms_table_enum: STRING • tbsp_status: INTEGER • tbsp_status_enum: STRING • db_name_u: STRING • tablespace_name_u: STRING • container_name_u: STRING • space_used_sms_table_mb: STRING • space_used_sms_table_mb_enum: STRING • db_partition: STRING • db_partition_enum: STRING • pool_hit_ratio_pct_for_int: INTEGER • pool_hit_ratio_pct_for_int_enum: STRING • instance_name_u: STRING • tbsp_state_name: STRING	ITM_KUDTABSPACE (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
KUDDBASEGROUP01 attribute group • db_name: STRING • app_ctl_heap_sz: INTEGER • applheapsz: INTEGER • applheapsz_enum: STRING • avg_appls: INTEGER • avg_appls_enum: STRING • buffpage: INTEGER • buffpage_enum: STRING • catalogcache_sz: INTEGER • catalogcache_sz_enum: STRING • chngpgs_thresh: INTEGER • chngpgs_thresh_enum: STRING • dbheap: INTEGER • dbheap_enum: STRING • locklist: INTEGER • locklist_enum: STRING • logbufsz: INTEGER • logbufsz_enum: STRING • logprimary: INTEGER • logprimary_enum: STRING • maxappls: INTEGER • maxappls_enum: STRING • maxlocks: INTEGER • maxlocks_enum: STRING • mincommit: INTEGER • mincommit_enum: STRING • newlogpath: STRING • num_iocleaners: INTEGER • num_iocleaners_enum: STRING • pckcachesz: INTEGER • pckcachesz_enum: STRING • restore_pending: INTEGER • restore_pending_enum: STRING • seqdetect: INTEGER • seqdetect_enum: STRING • sortheap: INTEGER • sortheap_enum: STRING • appls_in_db2: INTEGER • appls_in_db2_enum: STRING • deadlocks_for_int: INTEGER • deadlocks_for_int_enum: STRING • int_auto_rebinds: INTEGER • int_auto_rebinds_enum: STRING • int_commits: INTEGER • int_commits_enum: STRING • int_rows_deleted: INTEGER • int_rows_deleted_enum: STRING • int_rows_inserted: INTEGER • int_rows_inserted_enum: STRING • int_rows_updated: INTEGER • int_rows_updated_enum: STRING • log_io_for_int: INTEGER • log_io_for_int_enum: STRING • cur_cons_pct: INTEGER • cur_cons_pct_enum: STRING (Continued on the next page.)	ITM_KUDDBASEGROUP01

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• lock_list_in_use_pct: INTEGER • lock_list_in_use_pct_enum: STRING • sec_log_used_pct: INTEGER • sec_log_used_pct_enum: STRING • total_log_used: INTEGER • total_log_used_enum: STRING • pri_log_used_pct: INTEGER • pri_log_used_pct_enum: STRING • page_cleans_for_interval: INTEGER • page_cleans_for_interval_enum: STRING • pages_per_prefetch_for_int: INTEGER • pages_per_prefetch_for_int_enum: STRING • tot_sync_io: INTEGER • tot_sync_io_enum: STRING • pool_sync_index_reads: INTEGER • pool_sync_index_reads_enum: STRING • avg_pool_async_data_reads: INTEGER • avg_pool_async_data_reads_enum: STRING • avg_pool_async_data_writes: INTEGER • avg_pool_async_data_writes_enum: STRING • avg_pool_writes_per_read: INTEGER • avg_pool_writes_per_read_enum: STRING • avg_sect_read_per_direct_read: INTEGER • avg_sect_read_per_direct_read_enum: STRING • avg_sect_written_per_direct_write: INTEGER • avg_sect_written_per_direct_write_enum: STRING • avg_direct_read_time: INTEGER • avg_direct_read_time_enum: STRING • avg_direct_write_time: INTEGER • avg_direct_write_time_enum: STRING • avg_pool_io_time: INTEGER • avg_pool_io_time_enum: STRING • avg_sync_io_time: INTEGER • avg_sync_io_time_enum: STRING • avg_pages_per_cleaner_for_int: INTEGER • avg_pages_per_cleaner_for_int_enum: STRING • pool_io_per_sec: INTEGER • pool_io_per_sec_enum: STRING • estore_rw_ratio_for_int: INTEGER • estore_rw_ratio_for_int_enum: STRING • pool_hit_ratio_pct_for_int: INTEGER • pool_hit_ratio_pct_for_int_enum: STRING • pool_hit_ratio_index_pct_for_int: INTEGER • pool_hit_ratio_index_pct_for_int_enum: STRING (Continued on the next page)	ITM_KUDDBASEGROUP01 (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• tot_direct_io_time: INTEGER • tot_direct_io_time_enum: STRING • tot_pool_phys_io: INTEGER • tot_pool_phys_io_enum: STRING • tot_pool_phys_read: INTEGER • tot_pool_phys_read_enum: STRING • tot_pool_phys_write: INTEGER • tot_pool_phys_write_enum: STRING • tot_sync_io_time: INTEGER • tot_sync_io_time_enum: STRING • db_cap_err: INTEGER • db_cap_err_enum: STRING • db_cap_lag: INTEGER • db_cap_lag_enum: STRING • db_cap_prun: INTEGER • db_cap_prun_enum: STRING • avg_locks_held: INTEGER • avg_locks_held_enum: STRING • lock_waits_for_int: INTEGER • lock_waits_for_int_enum: STRING • lock_waits_pct: INTEGER • lock_waits_pct_enum: STRING • int_deadlock_rollbacks_pct_for_int: INTEGER • int_deadlock_rollbacks_pct_for_int_enum: STRING • db_tablespaces: INTEGER • db_tablespaces_enum: STRING • event_monitors: INTEGER • event_monitors_enum: STRING • invalid_triggers: INTEGER • invalid_triggers_enum: STRING • system_tablespaces: INTEGER • system_tablespaces_enum: STRING • tables: INTEGER • tables_enum: STRING • tablespaces: INTEGER • tablespaces_enum: STRING • tablespaces_long_data: INTEGER • tablespaces_long_data_enum: STRING • triggers: INTEGER • triggers_enum: STRING • views: INTEGER • views_enum: STRING • user_indexes: INTEGER • user_indexes_enum: STRING • commit_stmts_per_sec: INTEGER • commit_stmts_per_sec_enum: STRING • ddl_sql_pct_for_int: INTEGER • ddl_sql_pct_for_int_enum: STRING • select_sql_pct_for_int: INTEGER • select_sql_pct_for_int_enum: STRING • uid_sql_pct_for_int: INTEGER • uid_sql_pct_for_int_enum: STRING (Continued on the next page)	ITM_KUDDBASEGROUP01 (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• invalid_pkgs: INTEGER • invalid_pkgs_enum: STRING • invalid_sys_pkgs: INTEGER • invalid_sys_pkgs_enum: STRING • pri_log_used_top: INTEGER • pri_log_used_top_enum: STRING • int_deadlock_rollbacks_pct: INTEGER • int_deadlock_rollbacks_pct_enum: STRING • num_ioservers: INTEGER • num_ioservers_enum: STRING • snapshot_time: STRING • node_name: STRING • db_name_u: STRING • newlogpath_u: STRING • snapshot_time_timestamp: STRING • db_conn_time_timestamp: STRING • last_backup_timestamp: STRING • lock_list_in_use_kb: INTEGER • lock_list_in_use_kb_enum: STRING • pool_total_reads_k: INTEGER • pool_total_reads_k_enum: STRING • pool_total_writes_k: INTEGER • pool_total_writes_k_enum: STRING • sec_log_used_top_mb: INTEGER • sec_log_used_top_mb_enum: STRING • tot_log_used_top_mb: INTEGER • tot_log_used_top_mb_enum: STRING • total_log_used_mb: INTEGER • total_log_used_mb_enum: STRING • db_partition: STRING • db_partition_enum: STRING • instance_name_u: STRING • lock_timeouts_for_int: INTEGER • lock_timeouts_for_int_enum: STRING • rollback_rate_for_int: INTEGER • rollback_rate_for_int_enum: STRING • sql_stmts_rate_for_int: INTEGER • sql_stmts_rate_for_int_enum: STRING • days_since_last_backup: INTEGER • days_since_last_backup_enum: STRING • sort_overflows_pct_for_int: INTEGER • sort_overflows_pct_for_int_enum: STRING • failed_sql_stmts_pct_for_int: INTEGER • failed_sql_stmts_pct_for_int_enum: STRING • avg_lock_escal_con_for_int: INTEGER • avg_lock_escal_con_for_int_enum: STRING • deadlock_rollbacks_pct: INTEGER • deadlock_rollbacks_pct_enum: STRING • lock_escalation_for_int: INTEGER • lock_escalation_for_int_enum: STRING • pri_log_used_top_mb: INTEGER; • pri_log_used_top_mb_enum: STRING;	ITM_KUDDBASEGROUP01 (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
KUDDB2APPLGROUP01 attribute group • snapshot_time: STRING • appl_name: STRING • db_name: STRING • agents_stolen: INTEGER • agents_stolen_enum: STRING • app_work_load: INTEGER • app_work_load_enum: STRING • associated_agents_top: INTEGER • avg_sect_read_per_direct_read: INTEGER • avg_sect_read_per_direct_read_enum: STRING • avg_sect_written_per_direct_read: INTEGER • avg_sect_written_per_direct_read_enum: STRING • pool_index_hit_ratio_pct_for_int: INTEGER • pool_index_hit_ratio_pct_for_int_enum: STRING • pkg_cache_hit_pct: INTEGER • pkg_cache_hit_pct_enum: STRING • tot_pool_io_time: INTEGER • tot_pool_io_time_enum: STRING • lock_escalation_for_int: INTEGER • lock_escalation_for_int_enum: STRING • deadlocks_for_int: INTEGER • deadlocks_for_int_enum: STRING • stmts_sorts: INTEGER • stmts_sorts_enum: STRING • appl_section_inserts: INTEGER • appl_section_inserts_enum: STRING • binds_precompiles: INTEGER • binds_precompiles_enum: STRING • ddl_sql_pct_for_int: INTEGER • ddl_sql_pct_for_int_enum: STRING • uid_sql_pct_for_int: INTEGER • uid_sql_pct_for_int_enum: STRING • sql_reqs_since_commit: INTEGER • sql_reqs_since_commit_enum: STRING • node_name: STRING • appl_name_u: STRING • db_name_u: STRING • snapshot_time_timestamp: STRING • lock_wait_start_time_timestamp: STRING • appl_con_time_timestamp: STRING • conn_complete_time_timestamp: STRING • stmt_start_timestamp: STRING • stmt_stop_timestamp: STRING • prev_uow_stop_time_timestamp: STRING • uow_start_time_timestamp: STRING • uow_stop_time_timestamp: STRING • pool_total_reads_k: INTEGER • pool_total_reads_k_enum: STRING (Continued on the next page)	ITM_KUDDB2APPLGROUP01

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• pool_total_writes_k: INTEGER • pool_total_writes_k_enum: STRING • uow_log_space_used_mb: INTEGER • uow_log_space_used_mb_enum: STRING • appl_id_u: STRING • db_partition: STRING • db_partition_enum: STRING • instance_name_u: STRING • appl_section_lookups: INTEGER • appl_section_lookups_enum: STRING • agent_usr_cpu_time: REAL • agent_usr_cpu_time_enum: STRING • agent_sys_cpu_time: REAL • agent_sys_cpu_time_enum: STRING • lock_list_in_use_pct: INTEGER • lock_list_in_use_pct_enum: STRING • lock_wait_time_for_int: INTEGER • lock_wait_time_for_int_enum: STRING • open_curs: INTEGER • open_curs_enum: STRING • open_curs_blk: INTEGER • open_curs_blk_enum: STRING • pool_hit_ratio_pct_for_int: INTEGER • pool_hit_ratio_pct_for_int_enum: STRING • total_sorts_for_int: INTEGER • total_sorts_for_int_enum: STRING	ITM_KUDDB2APPLGROUP01 (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
KUDDB2APPLGROUP00_U attribute group • node_name: STRING • agent_id: INTEGER • agent_id_enum: STRING • appl_id: STRING • appl_status: STRING • appl_status_enum: STRING • snapshot_time: STRING • appl_name_u: STRING • auth_id_u: STRING • client_prdid: STRING • db_name_u: STRING • execution_id_u: STRING • corr_token_u: STRING • client_platform: STRING • client_platform_enum: STRING • client_protocol: STRING • client_protocol_enum: STRING • locks_held: INTEGER • locks_held_enum: STRING • lock_waits: INTEGER • lock_waits_enum: STRING • lock_wait_time: INTEGER • lock_wait_time_enum: STRING • lock_escals: INTEGER • lock_escals_enum: STRING • x_lock_escals: INTEGER • x_lock_escals_enum: STRING • deadlocks: INTEGER • deadlocks_enum: STRING • uow_lock_wait_time: INTEGER • uow_lock_wait_time_enum: STRING • lock_timeouts: INTEGER • lock_timeouts_enum: STRING • avg_lock_waittime: INTEGER • avg_lock_waittime_enum: STRING • agent_id_holding_lk: INTEGER • agent_id_holding_lk_enum: STRING • appl_id_holding_lk_u: STRING • lock_mode: STRING • lock_mode_enum: STRING • lock_object_type: STRING • lock_object_type_enum: STRING • lock_wait_start_time: STRING • table_name_u: STRING • table_schema_u: STRING • tablespace_name_u: STRING • pool_data_l_reads: INTEGER • pool_data_l_reads_enum: STRING • pool_data_p_reads: INTEGER • pool_data_p_reads_enum: STRING • pool_data_writes: INTEGER • pool_data_writes_enum: STRING • pool_index_l_reads: INTEGER • pool_index_l_reads_enum: STRING (Continued on the next page.)	ITM_KUDDB2APPLGROUP00_U

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• pool_index_p_reads: INTEGER • pool_index_p_reads_enum: STRING • pool_index_writes: INTEGER • pool_index_writes_enum: STRING • pool_read_time: INTEGER • pool_read_time_enum: STRING • pool_write_time: INTEGER • pool_write_time_enum: STRING • pool_data_to_estore: INTEGER • pool_data_to_estore_enum: STRING • pool_index_to_estore: INTEGER • pool_index_to_estore_enum: STRING • pool_index_from_estore: INTEGER • pool_index_from_estore_enum: STRING • pool_data_from_estore: INTEGER • pool_data_from_estore_enum: STRING • pool_total_reads: INTEGER • pool_total_reads_enum: STRING • pool_hit_ratio: INTEGER • pool_hit_ratio_enum: STRING • avg_pool_read_time: INTEGER • avg_pool_read_time_enum: STRING • pool_total_writes: INTEGER • pool_total_writes_enum: STRING • avg_pool_write_time: INTEGER • avg_pool_write_time_enum: STRING • appl_idle_time: INTEGER • appl_idle_time_enum: STRING • agent_usr_cpu_time: STRING • agent_sys_cpu_time: STRING • appl_con_time: STRING • conn_complete_time: STRING • prefetch_wait_time: INTEGER • prefetch_wait_time_enum: STRING • direct_reads: INTEGER • direct_reads_enum: STRING • direct_writes: INTEGER • direct_writes_enum: STRING • direct_read_reqs: INTEGER • direct_read_reqs_enum: STRING • direct_write_reqs: INTEGER • direct_write_reqs_enum: STRING • direct_read_time: INTEGER • direct_read_time_enum: STRING • direct_write_time: INTEGER • direct_write_time_enum: STRING • stmt_type: STRING • stmt_type_enum: STRING • stmt_operation: STRING • stmt_operation_enum: STRING • cursor_name_u: STRING • creator_u: STRING • package_name_u: STRING • total_sql_stmt: INTEGER • total_sql_stmt_enum: STRING (Continued on the next page.)	ITM_KUDDB2APPLGROUP00_U (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• failed_sql_stmts: INTEGER • failed_sql_stmts_enum: STRING • int_rollbacks: INTEGER • int_rollbacks_enum: STRING • int_deadlock_rollbacks: INTEGER • int_deadlock_rollbacks_enum: STRING • rollback_sql_stmts: INTEGER • rollback_sql_stmts_enum: STRING • failed_sql_stmts_pct: INTEGER • failed_sql_stmts_pct_enum: STRING • total_sorts: INTEGER • total_sorts_enum: STRING • total_sort_time: INTEGER • total_sort_time_enum: STRING • sort_overflows: INTEGER • sort_overflows_enum: STRING • avg_sort_time: INTEGER • avg_sort_time_enum: STRING • sort_overflows_pct: INTEGER • sort_overflows_pct_enum: STRING • commit_sql_stmts: INTEGER • commit_sql_stmts_enum: STRING • dynamic_sql_stmts: INTEGER • dynamic_sql_stmts_enum: STRING • static_sql_stmts: INTEGER • static_sql_stmts_enum: STRING • select_sql_stmts: INTEGER • select_sql_stmts_enum: STRING • ddl_sql_stmts: INTEGER • ddl_sql_stmts_enum: STRING • uid_sql_stmts: INTEGER • uid_sql_stmts_enum: STRING • binds_precompiles: INTEGER • binds_precompiles_enum: STRING • section_number: INTEGER • section_number_enum: STRING • stmt_start: STRING • stmt_stop: STRING • uow_log_space_used: INTEGER • uow_log_space_used_enum: STRING • uow_comp_status: STRING • uow_comp_status_enum: STRING • prev_uow_stop_time: STRING • uow_start_time: STRING • uow_stop_time: STRING • int_auto_rebinds: INTEGER • int_auto_rebinds_enum: STRING • int_rows_deleted: INTEGER • int_rows_deleted_enum: STRING • int_rows_updated: INTEGER • int_rows_updated_enum: STRING • int_commits: INTEGER • int_commits_enum: STRING • int_rows_inserted: INTEGER • int_rows_inserted_enum: STRING (Continued on the next page.)	ITM_KUDDB2APPLGROUP00_U (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• rows_deleted: INTEGER • rows_deleted_enum: STRING • rows_inserted: INTEGER • rows_inserted_enum: STRING • rows_updated: INTEGER • rows_updated_enum: STRING • rows_selected: INTEGER • rows_selected_enum: STRING • rows_read: INTEGER • rows_read_enum: STRING • rows_written: INTEGER • rows_written_enum: STRING • open_loc_curs: INTEGER • open_loc_curs_enum: STRING • open_loc_curs_blk: INTEGER • open_loc_curs_blk_enum: STRING • open_rem_curs: INTEGER • open_rem_curs_enum: STRING • open_rem_curs_blk: INTEGER • open_rem_curs_blk_enum: STRING • rej_curs_blk: INTEGER • rej_curs_blk_enum: STRING • acc_curs_blk: INTEGER • acc_curs_blk_enum: STRING • client_pid: INTEGER • client_pid_enum: STRING • country_code: INTEGER • country_code_enum: STRING • pkg_cache_lookups: INTEGER • pkg_cache_lookups_enum: STRING • pkg_cache_inserts: INTEGER • pkg_cache_inserts_enum: STRING • pkg_cache_hit_ratio: INTEGER • pkg_cache_hit_ratio_enum: STRING • cat_cache_lookups: INTEGER • cat_cache_lookups_enum: STRING • cat_cache_inserts: INTEGER • cat_cache_inserts_enum: STRING • cat_cache_overflows: INTEGER • cat_cache_overflows_enum: STRING • cat_cache_heap_full: INTEGER • cat_cache_heap_full_enum: STRING • cat_cache_hit_ratio: INTEGER • cat_cache_hit_ratio_enum: STRING • total_hash_joins: INTEGER • total_hash_joins_enum: STRING • total_hash_loops: INTEGER • total_hash_loops_enum: STRING • hash_join_overflows: INTEGER • hash_join_overflows_enum: STRING • hash_join_small_overflows: INTEGER • hash_join_small_overflows_enum: STRING • query_cost_estimate: INTEGER • query_cost_estimate_enum: STRING (Continued on the next page.)	ITM_KUDDB2APPLGROUP00_U (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• query_card_estimate: INTEGER • query_card_estimate_enum: STRING • degree_parallelism: INTEGER • degree_parallelism_enum: STRING • stmt_text_u: STRING • db_partition: STRING • db_partition_ENUM: STRING	ITM_KUDDB2APPLGROUP00_U (Continued)
KUD_DB2_Apply_Program attribute group • node_name: STRING • instance_name_u: STRING • db_name_u: STRING • apply_qualifier: STRING • apply_id: STRING • apply_status: INTEGER • apply_status_enum: STRING • tot_apply_sub_fail: INTEGER • tot_apply_sub_fail_enum: STRING • tot_apply_sub_lag: INTEGER • tot_apply_sub_lag_enum: STRING • snapshot_time_timestamp: STRING • tot_apply_sub_fail_64: REAL • tot_apply_sub_fail_64_enum: STRING • tot_apply_sub_lag_64: REAL • tot_apply_sub_lag_64_enum: STRING	KUD_DB2_Apply_Program
KUD_DB2_Table attribute group • node_name: STRING • instance_name_u: STRING • db_name_u: STRING • db_partition: STRING • db_partition_enum: STRING • table_name_u: STRING • table_schema_u: STRING • reorg_needed: STRING • reorg_needed_enum: STRING • rows_read_rate_for_int: INTEGER • rows_read_rate_for_int_enum: STRING • rows_write_rate_for_int: INTEGER • rows_write_rate_for_int_enum: STRING • snapshot_time_timestamp: STRING	KUD_DB2_Table

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
KUD_DB2_DCS_Database attribute group • node_name: STRING • instance_name_u: STRING • db_name_u: STRING • db_partition: STRING • db_partition_enum: STRING • gw_cur_cons: INTEGER • gw_cur_cons_enum: STRING • gw_cons_wait_host: INTEGER • gw_cons_wait_host_enum: STRING • gw_comm_errors_for_int: INTEGER • gw_comm_errors_for_int_enum: STRING • recent_con_rsp_time: REAL • recent_con_rsp_time_enum: STRING • network_time_per_stmt: INTEGER • network_time_per_stmt_enum: STRING • host_throughput_for_int: INTEGER • host_throughput_for_int_enum: STRING • host_time_per_stmt_for_int: INTEGER • host_time_per_stmt_for_int_enum: STRING • time_per_stmt: INTEGER • time_per_stmt_enum: STRING • snapshot_time_timestamp: STRING • gw_cur_cons_64: REAL • gw_cur_cons_64_enum: STRING • gw_cons_wait_host_64: REAL • gw_cons_wait_host_64_enum: STRING • gw_comm_errors_for_int_64: REAL • gw_comm_errors_for_int_64_enum: STRING • recent_con_rsp_time_64: REAL • recent_con_rsp_time_64_enum: STRING • network_time_per_stmt_64: REAL • network_time_per_stmt_64_enum: STRING • host_throughput_for_int_64: REAL • host_throughput_for_int_64_enum: STRING • host_time_per_stmt_for_int_64: REAL • host_time_per_stmt_for_int_64_enum: STRING • time_per_stmt_64: REAL • time_per_stmt_64_enum: STRING	KUD_DB2_DCS_Database

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
KUD_DB2_Apply_Subscription attribute group • node_name: STRING • instance_name_u: STRING • db_name_target: STRING • target_table: STRING • target_owner: STRING • apply_num_reqs_refresh: INTEGER • apply_num_reqs_refresh_enum: STRING • apply_sub_lag_time: INTEGER • apply_sub_lag_time_enum: STRING • apply_sub_status: INTEGER • apply_sub_status_enum: STRING • apply_id: STRING • snapshot_time_timestamp: STRING • apply_num_reqs_refresh_64: REAL • apply_num_reqs_refresh_64_enum: STRING • apply_sub_lag_time_64: REAL • apply_sub_lag_time_64_enum: STRING	KUD_DB2_Apply_Subscription
KUD_DB2_Diagnostic_Messages attribute group • node_name: STRING • msgid: STRING • msgtext: STRING • timestamp: STRING • timezone: INTEGER • dbname: STRING	ITM_KUD_DB2_Diagnostic_Messages

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
KUD_DB2_System_Resources attribute group • node_name: STRING • host_name: STRING • os_name: STRING • os_version: STRING • total_physical_mem: INTEGER • total_physical_mem_enum: STRING • free_physical_mem: INTEGER • free_physical_mem_enum: STRING • used_physical_mem_pct: INTEGER • used_physical_mem_pct_enum: STRING • total_swap_mem: INTEGER • total_swap_mem_enum: STRING • free_swap_mem: INTEGER • free_swap_mem_enum: STRING • used_swap_mem_pct: INTEGER • used_swap_mem_pct_enum: STRING • total_virtual_mem: INTEGER • total_virtual_mem_enum: STRING • free_virtual_mem: INTEGER • free_physical_mem_enum: STRING • free_virtual_mem_enum: STRING • used_virtual_mem_pct: INTEGER • used_virtual_mem_pct_enum: STRING • cpu_usage_pct: INTEGER • cpu_usage_pct_enum: STRING • total_physical_mem_64: REAL • total_physical_mem_64_enum: STRING • free_physical_mem_64: REAL • free_physical_mem_64_enum: STRING • total_swap_mem_64: REAL • total_swap_mem_64_enum: STRING • free_swap_mem_64: REAL • free_swap_mem_64_enum: STRING • total_virtual_mem_64: REAL • total_virtual_mem_64_enum: STRING • free_virtual_mem_64: REAL • free_virtual_mem_64_enum: STRING • used_physical_mem_percent: REAL • used_physical_mem_percent_enum: STRING • used_swap_mem_percent: REAL • used_swap_mem_percent_enum: STRING • used_virtual_mem_percent: REAL • used_virtual_mem_percent_enum: STRING	ITM_KUD_DB2_System_Resources

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
KUD_DB2_IPADDR_TABLE attribute group • node_name: STRING • server_name: STRING • ip_address: STRING • ip_type: INTEGER • ip_type_enum: STRING • listener_port: INTEGER • listener_port_enum: STRING • db_partition: STRING • db_partition_enum: STRING	ITM_KUD_DB2_IPADDR_TABLE

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
KUD_DB2_System_Overview attribute group • node_name: STRING • db2_status: STRING • db2_status_enum: STRING • last_reset: STRING • snapshot_time: STRING • prdid: STRING • instance_name: STRING • instance_name_enum: STRING • version: STRING • version_enum: STRING • server_db2_type: STRING • server_db2_type_enum: STRING • sort_heap_allocated: REAL • sort_heap_allocated_enum: STRING • post_threshold_sorts: REAL • post_threshold_sorts_enum: STRING • piped_sorts_requested: REAL • piped_sorts_requested_enum: STRING • piped_sorts_accepted: REAL • piped_sorts_accepted_enum: STRING • piped_sorts_accepted_pct: REAL • piped_sorts_accepted_pct_enum: STRING • rem_cons_in: REAL • rem_cons_in_enum: STRING • rem_cons_in_exec: REAL • rem_cons_in_exec_enum: STRING • local_cons: REAL • local_cons_enum: STRING • local_cons_in_exec: REAL • local_cons_in_exec_enum: STRING • con_local_dbases: REAL • con_local_dbases_enum: STRING • agents_registered: REAL • agents_registered_enum: STRING • agents_waiting_on_token: REAL • agents_waiting_on_token_enum: STRING • agents_waiting_on_token_pct: REAL • agents_waiting_on_token_pct_enum: STRING • agents_from_pool: REAL • agents_from_pool_enum: STRING • agents_created_empty_pool: REAL • agents_created_empty_pool_enum: STRING • agents_created_empty_pool_ratio: REAL • agents_created_empty_pool_ratio_enum: STRING • coord_agents_top: REAL • coord_agents_top_enum: STRING • max_agent_overflows: REAL • max_agent_overflows_enum: STRING • agents_stolen: REAL • agents_stolen_enum: STRING (Continued on the next page.)	ITM_KUD_DB2_System_Overview

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• agents_registered_top: REAL • agents_registered_top_enum: STRING • agents_waiting_top: REAL • agents_waiting_top_enum: STRING • comm_private_mem: REAL • comm_private_mem_enum: STRING • idle_agents: REAL • idle_agents_enum: STRING • gw_total_cons: REAL • gw_total_cons_enum: STRING • gw_cur_cons: REAL • gw_cur_cons_enum: STRING • gw_cons_wait_host: REAL • gw_cons_wait_host_enum: STRING • gw_cons_wait_client: REAL • gw_cons_wait_client_enum: STRING • post_threshold_hash_joins: REAL • post_threshold_hash_joins_enum: STRING • piped_sort_hit_ratio_pct_for_int: REAL • piped_sort_hit_ratio_pct_for_int_enum: STRING • agentpri: INTEGER • agentpri_enum: STRING • aslheapsz: REAL • aslheapsz_enum: STRING • fcm_num_anchors: REAL • fcm_num_anchors_enum: STRING • fcm_num_buffers: REAL • fcm_num_buffers_enum: STRING • fcm_num_connect: REAL • fcm_num_connect_enum: STRING • fcm_num_rqb: REAL • fcm_num_rqb_enum: STRING • maxagents: REAL • maxagents_enum: STRING • max_coordagents: INTEGER • max_coordagents_enum: STRING • maxcagents: INTEGER • maxcagents_enum: STRING • mon_heap_sz: INTEGER • mon_heap_sz_enum: STRING • query_heap_sz: INTEGER • query_heap_sz_enum: STRING • rqrioblk: INTEGER • rqrioblk_enum: STRING • piped_sorts_rejected_pct_for_int: REAL • piped_sorts_rejected_pct_for_int_enum: STRING • sheapthres: REAL • sheapthres_enum: STRING (Continued on the next page.)	ITM_KUD_DB2_System_Overview (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• connection_status: INTEGER • connection_status_enum: STRING • buff_free: REAL • buff_free_enum: STRING • ce_free: REAL • ce_free_enum: STRING • rb_free: REAL • rb_free_enum: STRING • buff_free_bottom: REAL • buff_free_bottom_enum: STRING • ce_free_bottom: REAL • ce_free_bottom_enum: STRING • ma_free_bottom: REAL • ma_free_bottom_enum: STRING • rb_free_bottom: REAL • rb_free_bottom_enum: STRING • buf_used_pct: REAL • buf_used_pct_enum: STRING • rb_used_pct: REAL • rb_used_pct_enum: STRING • ce_used_pct: REAL • ce_used_pct_enum: STRING • buff_max_used_pct: REAL • buff_max_used_pct_enum: STRING • ce_max_used_pct: REAL • ce_max_used_pct_enum: STRING • ma_max_used_pct: REAL • ma_max_used_pct_enum: STRING • rb_max_used_pct: REAL • rb_max_used_pct_enum: STRING • total_buffers_rcvd: REAL • total_buffers_rcvd_enum: STRING • total_buffers_sent: REAL • total_buffers_sent_enum: STRING • piped_sorts_rejected_for_int: REAL • piped_sorts_rejected_for_int_enum: STRING • dbpg_node_status: STRING • db2_avail: REAL • db2_avail_enum: STRING • cons_in_exec_pct: REAL • cons_in_exec_pct_enum: STRING • db_partition: STRING • db_partition_enum: STRING • sort_heap_used_pct: REAL • sort_heap_used_pct_enum: STRING • post_threshold_olap_funcs: REAL • post_threshold_olap_funcs_enum: STRING	ITM_KUD_DB2_System_Overview (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
KUD_DB2_Application00 attribute group • node_name: STRING • agent_id: INTEGER • agent_id_enum: STRING • appl_id: STRING • appl_status: STRING • appl_status_enum: STRING • snapshot_time: STRING • appl_name: STRING • auth_id: STRING • client_prdid: STRING • db_name: STRING • execution_id: STRING • corr_token: STRING • client_platform: STRING • client_platform_enum: STRING • client_protocol: STRING • client_protocol_enum: STRING • locks_held: REAL • locks_held_enum: STRING • lock_waits: REAL • lock_waits_enum: STRING • lock_wait_time: REAL • lock_wait_time_enum: STRING • lock_escals: REAL • lock_escals_enum: STRING • x_lock_escals: REAL • x_lock_escals_enum: STRING • deadlocks: REAL • deadlocks_enum: STRING • uow_lock_wait_time: REAL • uow_lock_wait_time_enum: STRING • lock_timeouts: REAL • lock_timeouts_enum: STRING • avg_lock_waittime: REAL • avg_lock_waittime_enum: STRING • agent_id_holding_lk: INTEGER • agent_id_holding_lk_enum: STRING • appl_id_holding_lk: STRING • lock_mode: STRING • lock_mode_enum: STRING • lock_object_type: STRING • lock_object_type_enum: STRING • lock_wait_start_time: STRING • table_name: STRING • table_schema: STRING • tablespace_name: STRING • pool_data_l_reads: REAL • pool_data_l_reads_enum: STRING • pool_data_p_reads: REAL • pool_data_p_reads_enum: STRING • pool_data_writes: REAL • pool_data_writes_enum: STRING • pool_index_l_reads: REAL • pool_index_l_reads_enum: STRING (Continued on the next page.)	ITM_KUD_DB2_Application00

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• pool_index_p_reads: REAL • pool_index_p_reads_enum: STRING • pool_index_writes: REAL • pool_index_writes_enum: STRING • pool_read_time: REAL • pool_read_time_enum: STRING • pool_write_time: REAL • pool_write_time_enum: STRING • pool_data_to_estore: REAL • pool_data_to_estore_enum: STRING • pool_index_to_estore: REAL • pool_index_to_estore_enum: STRING • pool_index_from_estore: REAL • pool_index_from_estore_enum: STRING • pool_data_from_estore: REAL • pool_data_from_estore_enum: STRING • pool_total_reads: REAL • pool_total_reads_enum: STRING • pool_hit_ratio: REAL • pool_hit_ratio_enum: STRING • avg_pool_read_time: REAL • avg_pool_read_time_enum: STRING • pool_total_writes: REAL • pool_total_writes_enum: STRING • avg_pool_write_time: REAL • avg_pool_write_time_enum: STRING • appl_idle_time: REAL • appl_idle_time_enum: STRING • appl_con_time: STRING • conn_complete_time: STRING • direct_reads: REAL • direct_reads_enum: STRING • direct_writes: REAL • direct_writes_enum: STRING • direct_read_reqs: REAL • direct_read_reqs_enum: STRING • direct_write_reqs: REAL • direct_write_reqs_enum: STRING • direct_read_time: REAL • direct_read_time_enum: STRING • direct_write_time: REAL • direct_write_time_enum: STRING • stmt_type: STRING • stmt_type_enum: STRING • stmt_operation: STRING • stmt_operation_enum: STRING • cursor_name: STRING • creator: STRING • package_name: STRING • total_sql_stmt: REAL • total_sql_stmt_enum: STRING • failed_sql_stmts: REAL • failed_sql_stmts_enum: STRING • int_rollbacks: REAL • int_rollbacks_enum: STRING (Continued on the next page.)	ITM_KUD_DB2_Application00 (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• int_deadlock_rollbacks: REAL • int_deadlock_rollbacks_enum: STRING • rollback_sql_stmts: REAL • rollback_sql_stmts_enum: STRING • failed_sql_stmts_pct: REAL • failed_sql_stmts_pct_enum: STRING • total_sorts: REAL • total_sorts_enum: STRING • total_sort_time: REAL • total_sort_time_enum: STRING • sort_overflows: REAL • sort_overflows_enum: STRING • avg_sort_time: REAL • avg_sort_time_enum: STRING • sort_overflows_pct: REAL • sort_overflows_pct_enum: STRING • commit_sql_stmts: REAL • commit_sql_stmts_enum: STRING • dynamic_sql_stmts: REAL • dynamic_sql_stmts_enum: STRING • static_sql_stmts: REAL • static_sql_stmts_enum: STRING • select_sql_stmts: REAL • select_sql_stmts_enum: STRING • ddl_sql_stmts: REAL • ddl_sql_stmts_enum: STRING • uid_sql_stmts: REAL • uid_sql_stmts_enum: STRING • binds_precompiles: REAL • binds_precompiles_enum: STRING • int_auto_rebinds: REAL • int_auto_rebinds_enum: STRING • int_rows_deleted: REAL • int_rows_deleted_enum: STRING • int_rows_updated: REAL • int_rows_updated_enum: STRING • int_commits: REAL • int_commits_enum: STRING • int_rows_inserted: REAL • int_rows_inserted_enum: STRING • rows_deleted: REAL • rows_deleted_enum: STRING • rows_inserted: REAL • rows_inserted_enum: STRING • rows_updated: REAL • rows_updated_enum: STRING • rows_selected: REAL • rows_selected_enum: STRING • rows_read: REAL • rows_read_enum: STRING • rows_written: REAL • rows_written_enum: STRING • open_loc_curs: REAL • open_loc_curs_enum: STRING (Continued on the next page.)	ITM_KUD_DB2_Application00 (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• open_loc_curs_blk: REAL • open_loc_curs_blk_enum: STRING • open_rem_curs: REAL • open_rem_curs_enum: STRING • open_rem_curs_blk: REAL • open_rem_curs_blk_enum: STRING • rej_curs_blk: REAL • rej_curs_blk_enum: STRING • acc_curs_blk: REAL • acc_curs_blk_enum: STRING • client_pid: INTEGER • client_pid_enum: STRING • country_code: INTEGER • country_code_enum: STRING • pkg_cache_lookups: REAL • pkg_cache_lookups_enum: STRING • pkg_cache_inserts: REAL • pkg_cache_inserts_enum: STRING • pkg_cache_hit_ratio: REAL • pkg_cache_hit_ratio_enum: STRING • cat_cache_lookups: REAL • cat_cache_lookups_enum: STRING • cat_cache_inserts: REAL • cat_cache_inserts_enum: STRING • cat_cache_overflows: REAL • cat_cache_overflows_enum: STRING • cat_cache_heap_full: REAL • cat_cache_heap_full_enum: STRING • cat_cache_hit_ratio: REAL • cat_cache_hit_ratio_enum: STRING • total_hash_joins: REAL • total_hash_joins_enum: STRING • total_hash_loops: REAL • total_hash_loops_enum: STRING • hash_join_overflows: REAL • hash_join_overflows_enum: STRING • hash_join_small_overflows: REAL • hash_join_small_overflows_enum: STRING • query_cost_estimate: REAL • query_cost_estimate_enum: STRING • query_card_estimate: REAL • query_card_estimate_enum: STRING • degree_parallelism: REAL • degree_parallelism_enum: STRING • stmt_text: STRING • db_partition: STRING • db_partition_enum: STRING	ITM_KUD_DB2_Application00 (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
KUD_DB2_Application01 attribute group • snapshot_time: STRING • appl_name: STRING • db_name: STRING • agents_stolen: REAL • agents_stolen_enum: STRING • app_work_load: REAL • app_work_load_enum: STRING • associated_agents_top: REAL • associated_agents_top_enum: STRING • avg_sect_read_per_direct_read: REAL • avg_sect_read_per_direct_read_enum: STRING • avg_sect_written_per_direct_write: REAL • avg_sect_written_per_direct_write_enum: STRING • pool_index_hit_ratio_pct_for_int: REAL • pool_index_hit_ratio_pct_for_int_enum: STRING • pkg_cache_hit_pct: REAL • pkg_cache_hit_pct_enum: STRING • tot_pool_io_time: REAL • tot_pool_io_time_enum: STRING • lock_escalation_for_int: REAL • lock_escalation_for_int_enum: STRING • deadlocks_for_int: REAL • deadlocks_for_int_enum: STRING • stmts_sorts: REAL • stmts_sorts_enum: STRING • appl_section_inserts: REAL • appl_section_inserts_enum: STRING • ddl_sql_pct_for_int: REAL • ddl_sql_pct_for_int_enum: STRING • uid_sql_pct_for_int: REAL • uid_sql_pct_for_int_enum: STRING • sql_reqs_since_commit: REAL • sql_reqs_since_commit_enum: STRING • node_name: STRING • appl_id: STRING • db_partition: STRING • db_partition_enum: STRING • instance_name: STRING • appl_section_lookups: REAL • appl_section_lookups_enum: STRING • agent_usr_cpu_time: REAL • agent_usr_cpu_time_enum: STRING • agent_sys_cpu_time: REAL • agent_sys_cpu_time_enum: STRING • lock_list_in_use_pct: REAL • lock_list_in_use_pct_enum: STRING • lock_wait_time_for_int: REAL • lock_wait_time_for_int_enum: STRING (Continued on the next page.)	ITM_KUD_DB2_Application01

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• open_curs: REAL • open_curs_enum: STRING • open_curs_blk: REAL • open_curs_blk_enum: STRING • pool_hit_ratio_pct_for_int: REAL • pool_hit_ratio_pct_for_int_enum: STRING • total_sorts_for_int: REAL • total_sorts_for_int_enum: STRING • prefetch_wait_time: REAL • prefetch_wait_time_enum: STRING • prev_uow_stop_time: STRING • section_number: REAL • section_number_enum: STRING • stmt_start: STRING • stmt_stop: STRING • uow_comp_status: STRING • uow_comp_status_enum: STRING • uow_log_space_used: REAL • uow_log_space_used_enum: STRING • uow_start_time: STRING • uow_stop_time: STRING	ITM_KUD_DB2_Application01 (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
KUD_DB2_Buffer_Pool attribute group • node_name: STRING • bp_id: STRING • bp_name: STRING • input_db_alias: STRING • db_name: STRING • db_path: STRING • pool_data_l_reads: REAL • pool_data_l_reads_enum: STRING • pool_data_p_reads: REAL • pool_data_p_reads_enum: STRING • pool_data_writes: REAL • pool_data_writes_enum: STRING • files_closed: REAL • files_closed_enum: STRING • pool_async_index_reads: REAL • pool_async_index_reads_enum: STRING • pool_data_to_estore: REAL • pool_data_to_estore_enum: STRING • pool_index_to_estore: REAL • pool_index_to_estore_enum: STRING • pool_index_from_estore: REAL • pool_write_time_enum: STRING • pool_index_from_estore_enum: STRING • pool_data_from_estore: REAL • pool_data_from_estore_enum: STRING • pool_async_data_reads: REAL • pool_async_data_writes: REAL • pool_async_data_writes_enum: STRING • pool_async_index_writes: REAL • pool_async_index_writes_enum: STRING • pool_async_read_time: REAL • pool_async_read_time_enum: STRING • pool_async_write_time: REAL • pool_async_write_time_enum: STRING • pool_async_data_read_reqs: REAL • pool_async_data_read_reqs_enum: STRING • pool_total_reads: REAL • pool_total_reads_enum: STRING • pool_hit_ratio: REAL • pool_hit_ratio_enum: STRING • avg_pool_read_time: REAL • avg_pool_read_time_enum: STRING • pool_total_writes: REAL • pool_total_writes_enum: STRING • avg_pool_write_time: REAL • avg_pool_write_time_enum: STRING • pool_sync_data_reads: REAL • pool_sync_data_reads_enum: STRING • pool_sync_index_reads: REAL • pool_sync_index_reads_enum: STRING • pool_sync_read: REAL • pool_sync_read_enum: STRING • pool_sync_data_writes: REAL • pool_sync_data_writes_enum: STRING (Continued on the next page.)	ITM_KUD_DB2_Buffer_Pool

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• pool_sync_index_writes: REAL • pool_sync_index_writes_enum: STRING • pool_sync_write: REAL • pool_sync_write_enum: STRING • pool_sync_read_time: REAL • pool_sync_read_time_enum: STRING • avg_sync_read_time: REAL • avg_sync_read_time_enum: STRING • pool_sync_write_time: REAL • pool_sync_write_time_enum: STRING • avg_sync_write_time: REAL • avg_sync_write_time_enum: STRING • avg_data_page_read_per_async_req: REAL • avg_data_page_read_per_async_req_enum: STRING • direct_reads: REAL • direct_reads_enum: STRING • direct_writes: REAL • direct_writes_enum: STRING • direct_read_reqs: REAL • direct_read_reqs_enum: STRING • direct_write_reqs: REAL • direct_write_reqs_enum: STRING • direct_read_time: REAL • direct_read_time_enum: STRING • direct_write_time: REAL • direct_write_time_enum: STRING • avg_direct_read_time: REAL • avg_direct_read_time_enum: STRING • avg_direct_write_time: REAL • avg_direct_write_time_enum: STRING • db_partition: STRING • db_partition_enum: STRING • snapshot_time: STRING	ITM_KUD_DB2_Buffer_Pool (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
KUD_DB2_Database00 attribute group • node_name: STRING • instance_name: STRING • db_name: STRING • input_db_alias: STRING • db_path: STRING • dbase_status: STRING • dbase_status_enum: STRING • db_conn_time: STRING • last_backup: STRING • total_cons: REAL • total_cons_enum: STRING • appls_cur_cons: REAL • appls_cur_cons_enum: STRING • snapshot_time: STRING • int_rollbacks: REAL • int_rollbacks_enum: STRING • int_deadlock_rollbacks: REAL • int_deadlock_rollbacks_enum: STRING • rollback_sql_stmts: REAL • rollback_sql_stmts_enum: STRING • failed_sql_stmts: REAL • failed_sql_stmts_enum: STRING • agents_top: REAL • agents_top_enum: STRING • coord_agents_top: REAL • coord_agents_top_enum: STRING • db_location: STRING • db_location_enum: STRING • server_platform: STRING • connections_top: REAL • connections_top_enum: STRING • locks_held: REAL • locks_held_enum: STRING • lock_waits: REAL • lock_waits_enum: STRING • lock_wait_time: REAL • lock_wait_time_enum: STRING • lock_list_in_use: REAL • lock_list_in_use_enum: STRING • deadlocks: REAL • deadlocks_enum: STRING • lock_escals: REAL • lock_escals_enum: STRING • x_lock_escals: REAL • x_lock_escals_enum: STRING • locks_waiting: REAL • locks_waiting_enum: STRING • lock_timeouts: REAL • lock_timeouts_enum: STRING • avg_lock_wait_time: REAL • avg_lock_wait_time_enum: STRING • sort_heap_allocated: REAL • sort_heap_allocated_enum: STRING • total_sorts: REAL • total_sorts_enum: STRING (Continued on the next page.)	ITM_KUD_DB2_Database00

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• total_sort_time: REAL • total_sort_time_enum: STRING • sort_overflows: REAL • sort_overflows_enum: STRING • active_sorts: REAL • active_sorts_enum: STRING • avg_sort_time: REAL • avg_sort_time_enum: STRING • sort_overflows_pct: REAL • sort_overflows_pct_enum: STRING • pool_data_l_reads: REAL • pool_data_l_reads_enum: STRING • pool_data_p_reads: REAL • pool_data_p_reads_enum: STRING • pool_data_writes: REAL • pool_data_writes_enum: STRING • pool_index_l_reads: REAL • pool_index_l_reads_enum: STRING • pool_index_p_reads: REAL • pool_index_p_reads_enum: STRING • pool_index_writes: REAL • pool_index_writes_enum: STRING • pool_read_time: REAL • pool_read_time_enum: STRING • pool_write_time: REAL • pool_write_time_enum: STRING • files_closed: REAL • files_closed_enum: STRING • pool_async_index_reads: REAL • pool_async_index_reads_enum: STRING • pool_data_to_estore: REAL • pool_data_to_estore_enum: STRING • pool_index_to_estore: REAL • pool_index_to_estore_enum: STRING • pool_index_from_estore: REAL • pool_index_from_estore_enum: STRING • pool_data_from_estore: REAL • pool_data_from_estore_enum: STRING • pool_async_data_reads: REAL • pool_async_data_reads_enum: STRING • pool_async_data_writes: REAL • pool_async_data_writes_enum: STRING • pool_async_index_writes: REAL • pool_async_index_writes_enum: STRING • pool_async_read_time: REAL • pool_async_read_time_enum: STRING • pool_async_write_time: REAL • pool_async_write_time_enum: STRING • pool_async_data_read_reqs: REAL • pool_async_data_read_reqs_enum: STRING • pool_lsn_gap_clns: REAL • pool_lsn_gap_clns_enum: STRING • pool_drty_pg_steal_clns: REAL • pool_drty_pg_steal_clns_enum: STRING • pool_drty_pg_thrsh_clns: REAL • pool_drty_pg_thrsh_clns_enum: STRING	ITM_KUD_DB2_Database00 (Continued

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• pool_total_reads: REAL • pool_total_reads_enum: STRING • pool_hit_ratio: REAL • pool_hit_ratio_enum: STRING • avg_pool_read_time: REAL • avg_pool_read_time_enum: STRING • pool_total_writes: REAL • pool_total_writes_enum: STRING • avg_pool_write_time: REAL • avg_pool_write_time_enum: STRING • pool_sync_data_reads: REAL • pool_sync_data_reads_enum: STRING • pool_sync_index_reads: REAL • pool_sync_index_reads_enum: STRING • direct_reads: REAL • direct_reads_enum: STRING • direct_writes: REAL • direct_writes_enum: STRING • direct_read_reqs: REAL • direct_read_reqs_enum: STRING • direct_write_reqs: REAL • direct_write_reqs_enum: STRING • direct_read_time: REAL • direct_read_time_enum: STRING • direct_write_time: REAL • direct_write_time_enum: STRING • commit_sql_stmts: REAL • commit_sql_stmts_enum: STRING • dynamic_sql_stmts: REAL • dynamic_sql_stmts_enum: STRING • static_sql_stmts: REAL • static_sql_stmts_enum: STRING • select_sql_stmts: REAL • select_sql_stmts_enum: STRING • ddl_sql_stmts: REAL • ddl_sql_stmts_enum: STRING • uid_sql_stmts: REAL • uid_sql_stmts_enum: STRING • total_sql_stmts: REAL • total_sql_stmts_enum: STRING • sql_stmts_failed_pct: REAL • sql_stmts_failed_pct_enum: STRING • sql_stmts_rollback_pct: REAL • sql_stmts_rollback_pct_enum: STRING • total_hash_joins: REAL • total_hash_joins_enum: STRING • total_hash_loops: REAL • total_hash_loops_enum: STRING • hash_join_overflows: REAL • hash_join_overflows_enum: STRING • hash_join_small_overflows: REAL • hash_join_small_overflows_enum: STRING • rows_deleted: REAL • rows_deleted_enum: STRING	ITM_KUD_DB2_Database00 (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• rows_inserted: REAL • rows_inserted_enum: STRING • rows_updated: REAL • rows_updated_enum: STRING • rows_selected: REAL • rows_selected_enum: STRING • total_sec_cons: REAL • total_sec_cons_enum: STRING • num_assoc_agents: REAL • num_assoc_agents_enum: STRING • catalog_node_name: STRING • sec_log_used_top: REAL • sec_log_used_top_enum: STRING • tot_log_used_top: REAL • tot_log_used_top_enum: STRING • sec_logs_allocated: REAL • sec_logs_allocated_enum: STRING • log_reads: REAL • log_reads_enum: STRING • log_writes: REAL • log_writes_enum: STRING • pkg_cache_lookups: REAL • pkg_cache_lookups_enum: STRING • pkg_cache_inserts: REAL • pkg_cache_inserts_enum: STRING • pkg_cache_hit_ratio: REAL • pkg_cache_hit_ratio_enum: STRING • cat_cache_lookups: REAL • cat_cache_lookups_enum: STRING • cat_cache_inserts: REAL • cat_cache_inserts_enum: STRING • cat_cache_overflows: REAL • cat_cache_overflows_enum: STRING • cat_cache_hit_ratio: REAL • cat_cache_hit_ratio_enum: STRING • cat_cache_heap_full: REAL • cat_cache_heap_full_enum: STRING • db_partition: STRING • db_partition_enum: STRING	ITM_KUD_DB2_Database00 (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
KUD_DB2_Database01 attribute group • db_name: STRING • app_ctl_heap_sz: INTEGER • app_ctl_heap_sz_enum: STRING • applheapsz_enum: STRING • avg_appls: INTEGER • avg_appls_enum: STRING • buffpage: REAL • buffpage_enum: STRING • catalogcache_sz: INTEGER • catalogcache_sz_enum: STRING • chngpgs_thresh: INTEGER • chngpgs_thresh_enum: STRING • dbheap: REAL • dbheap_enum: STRING • locklist: REAL • locklist_enum: STRING • logbufsz: INTEGER • logbufsz_enum: STRING • logprimary: INTEGER • logprimary_enum: STRING • maxappls: INTEGER • maxappls_enum: STRING • maxlocks: INTEGER • maxlocks_enum: STRING • mincommit: INTEGER • mincommit_enum: STRING • newlogpath: STRING • num_iocleaners: INTEGER • num_iocleaners_enum: STRING • pckcachesz: REAL • pckcachesz_enum: STRING • restore_pending: INTEGER • restore_pending_enum: STRING • seqdetect: INTEGER • seqdetect_enum: STRING • sortheap: REAL • sortheap_enum: STRING • appls_in_db2: REAL • appls_in_db2_enum: STRING • deadlocks_for_int: REAL • deadlocks_for_int_enum: STRING • int_auto_rebinds: REAL • int_auto_rebinds_enum: STRING • int_commits: REAL • int_commits_enum: STRING • int_rows_deleted: REAL • int_rows_deleted_enum: STRING • int_rows_inserted: REAL • int_rows_inserted_enum: STRING • int_rows_updated: REAL • int_rows_updated_enum: STRING • log_io_for_int: REAL • log_io_for_int_enum: STRING • cur_cons_pct: REAL • cur_cons_pct_enum: STRING (Continued on the next page.)	ITM_KUD_DB2_Database01

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• tot_pool_phys_io: REAL • tot_pool_phys_io_enum: STRING • tot_pool_phys_read: REAL • tot_pool_phys_read_enum: STRING • tot_pool_phys_write: REAL • tot_pool_phys_write_enum: STRING • tot_sync_io_time: REAL • tot_sync_io_time_enum: STRING • db_cap_err: REAL • db_cap_err_enum: STRING • db_cap_lag: REAL • db_cap_lag_enum: STRING • db_cap_prun: REAL • db_cap_prun_enum: STRING • avg_locks_held: REAL • avg_locks_held_enum: STRING • lock_waits_for_int: REAL • lock_waits_for_int_enum: STRING • lock_waits_pct: REAL • lock_waits_pct_enum: STRING • int_deadlock_rollbacks_pct_for_int: REAL • int_deadlock_rollbacks_pct_for_int_enum: STRING • db_tablespaces: REAL • db_tablespaces_enum: STRING • event_monitors: REAL • event_monitors_enum: STRING • invalid_triggers: REAL • invalid_triggers_enum: STRING • system_tablespaces: REAL • system_tablespaces_enum: STRING • tables: REAL • tables_enum: STRING • tablespaces: REAL • tablespaces_enum: STRING • tablespaces_long_data: REAL • tablespaces_long_data_enum: STRING • triggers: REAL • triggers_enum: STRING • views: REAL • views_enum: STRING • user_indexes: REAL • user_indexes_enum: STRING • commit_stmts_per_sec: REAL • commit_stmts_per_sec_enum: STRING • ddl_sql_pct_for_int: REAL • ddl_sql_pct_for_int_enum: STRING • select_sql_pct_for_int: REAL • select_sql_pct_for_int_enum: STRING • uid_sql_pct_for_int: REAL • uid_sql_pct_for_int_enum: STRING • invalid_pkgs: REAL • invalid_pkgs_enum: STRING (Continued on the next page.)	ITM_KUD_DB2_Database01 (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• invalid_sys_pkgs: REAL • invalid_sys_pkgs_enum: STRING • pri_log_used_top: REAL • pri_log_used_top_enum: STRING • int_deadlock_rollbacks_pct: INTEGER • int_deadlock_rollbacks_pct_enum: STRING • num_ioservers: INTEGER • num_ioservers_enum: STRING • snapshot_time: STRING • node_name: STRING • db_partition: STRING • db_partition_enum: STRING • instance_name: STRING • lock_timeouts_for_int: REAL • lock_timeouts_for_int_enum: STRING • rollback_rate_for_int: REAL • rollback_rate_for_int_enum: STRING • sql_stmts_rate_for_int: REAL • sql_stmts_rate_for_int_enum: STRING • days_since_last_backup: REAL • days_since_last_backup_enum: STRING • sort_overflows_pct_for_int: REAL • sort_overflows_pct_for_int_enum: STRING • failed_sql_stmts_pct_for_int: REAL • failed_sql_stmts_pct_for_int_enum: STRING • avg_lock_escal_con_for_int: REAL • avg_lock_escal_con_for_int_enum: STRING • deadlock_rollbacks_pct: REAL • deadlock_rollbacks_pct_enum: STRING • lock_escalation_for_int: REAL • lock_escalation_for_int_enum: STRING • curr_pri_log_used_pct: REAL • curr_pri_log_used_pct_enum: STRING • curr_sec_log_used_pct: REAL • curr_sec_log_used_pct_enum: STRING • prefetch_wait_time: REAL • db_heap_top: REAL • db_heap_top_enum: STRING • pool_sync_read: REAL • pool_sync_read_enum: STRING • pool_sync_data_writes: REAL • pool_sync_data_writes_enum: STRING • pool_sync_index_writes: REAL • pool_sync_index_writes_enum: STRING • pool_sync_write: REAL • pool_sync_write_enum: STRING • pool_sync_read_time: REAL • pool_sync_read_time_enum: STRING • avg_sync_read_time: REAL • avg_sync_read_time_enum: STRING (Continued on the next page.)	ITM_KUD_DB2_Database01 (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• pool_sync_write_time: REAL • pool_sync_write_time_enum: STRING • avg_sync_write_time: REAL • avg_sync_write_time_enum: STRING • avg_data_page_read_per_async_req: REAL • avg_data_page_read_per_async_req_enum: STRING • appl_section_lookups: REAL • appl_section_lookups_enum: STRING • appl_section_inserts: REAL • appl_section_inserts_enum: STRING • binds_precompiles: REAL • binds_precompiles_enum: STRING	ITM_KUD_DB2_Database01 (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
KUD_DB2_Tablespace attribute group • node_name: STRING • db_name: STRING • tablespace_name: STRING • tablespace_type: STRING • tablespace_type_enum: STRING • extent_size: REAL • extent_size_enum: STRING • page_size: REAL • page_size_enum: STRING • prefetch_size: REAL • prefetch_size_enum: STRING • num_containers: REAL • num_containers_enum: STRING • container_name: STRING • object_id: INTEGER • object_id_enum: STRING • tablespace_id: REAL • tablespace_id_enum: STRING • total_pages: REAL • total_pages_enum: STRING • usable_pages: REAL • usable_pages_enum: STRING • used_pages: REAL • used_pages_enum: STRING • pending_free_pages: REAL • pending_free_pages_enum: STRING • free_pages: REAL • free_pages_enum: STRING • direct_reads: REAL • direct_reads_enum: STRING • direct_writes: REAL • direct_writes_enum: STRING • direct_read_reqs: REAL • direct_read_reqs_enum: STRING • direct_write_reqs: REAL • direct_write_reqs_enum: STRING • pool_data_writes: REAL • pool_data_writes_enum: STRING • pool_async_data_reads: REAL • pool_async_data_reads_enum: STRING • pool_async_data_writes: REAL • pool_async_data_writes_enum: STRING • pool_async_index_reads: REAL • pool_async_index_reads_enum: STRING • pool_async_index_writes: REAL • pool_async_index_writes_enum: STRING • pool_async_read_time: REAL • pool_async_read_time_enum: STRING • pool_async_write_time: REAL • pool_async_write_time_enum: STRING • pool_data_l_reads: REAL • pool_data_l_reads_enum: STRING • pool_data_from_estore: REAL • pool_data_from_estore_enum: STRING (Continued on the next page.)	ITM_KUD_DB2_Tablespace

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• pool_data_to_estore: REAL • pool_data_to_estore_enum: STRING • pool_data_p_reads: REAL • pool_data_p_reads_enum: STRING • pool_index_l_reads: REAL • pool_index_l_reads_enum: STRING • pool_index_from_estore: REAL • pool_index_from_estore_enum: STRING • pool_index_to_estore: REAL • pool_index_to_estore_enum: STRING • pool_index_p_reads: REAL • pool_index_p_reads_enum: STRING • pool_index_writes: REAL • pool_index_writes_enum: STRING • direct_write_time: REAL • direct_write_time_enum: STRING • direct_read_time: REAL • direct_read_time_enum: STRING • files_closed: REAL • files_closed_enum: STRING • pool_read_time: REAL • pool_read_time_enum: STRING • pool_write_time: REAL • pool_write_time_enum: STRING • pool_async_data_read_reqs: REAL • pool_async_data_read_reqs_enum: STRING • pool_data_reads: REAL • pool_data_reads_enum: STRING • pool_sync_idx_reads: REAL • pool_sync_idx_reads_enum: STRING • pool_sync_idx_writes: REAL • pool_sync_idx_writes_enum: STRING • pool_sync_data_reads: REAL • pool_sync_data_reads_enum: STRING • pool_sync_data_writes: REAL • pool_sync_data_writes_enum: STRING • avg_sect_read: REAL • avg_sect_read_enum: STRING • avg_sect_written: REAL • avg_sect_written_enum: STRING • avg_direct_read_time: REAL • avg_direct_read_time_enum: STRING • avg_direct_write_time: REAL • avg_direct_write_time_enum: STRING • avg_pool_io_time: REAL • avg_pool_io_time_enum: STRING • avg_pool_read_time: REAL • avg_pool_read_time_enum: STRING (Continued on the next page.)	ITM_KUD_DB2_Tablespace (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• avg_pool_write_time: REAL • avg_pool_write_time_enum: STRING • avg_sync_data_read_time: REAL • avg_sync_data_read_time_enum: STRING • avg_sync_data_write_time: REAL • avg_sync_data_write_time_enum: STRING • avg_sync_io_time: REAL • avg_sync_io_time_enum: STRING • pool_io_per_sec: REAL • pool_io_per_sec_enum: STRING • estore_rw_ratio_enum: STRING • estore_rw_ratio_enum: STRING • pool_hit_pct: REAL • pool_hit_pct_enum: STRING • pool_idx_hit_pct_for_int: REAL • pool_idx_hit_pct_for_int_enum: STRING • prefetch_pct_for_int: REAL • prefetch_pct_for_int_enum: STRING • total_io_pct: REAL • total_io_pct_enum: STRING • prefetch_reqs_for_int: REAL • prefetch_reqs_for_int_enum: STRING • sync_read_time: REAL • sync_read_time_enum: STRING • sync_write_time: REAL • sync_write_time_enum: STRING • total_direct_io_time: REAL • total_direct_io_time_enum: STRING • total_pool_io_time: REAL • total_pool_io_time_enum: STRING • total_pool_p_read_time: REAL • total_pool_p_read_time_enum: STRING • total_pool_p_write_time: REAL • total_pool_p_write_time_enum: STRING • total_sync_io: REAL • total_sync_io_enum: STRING • total_sync_io_time: REAL • total_sync_io_time_enum: STRING • pool_async_index_read_reqs: REAL • pool_async_index_read_reqs_enum: STRING • space_used_dms_table_pct: REAL • space_used_dms_table_pct_enum: STRING • space_used_sms_table: REAL • space_used_sms_table_enum: STRING • tbsp_status: REAL • tbsp_status_enum: STRING • db_partition: STRING • db_partition_enum: STRING • pool_hit_ratio_pct_for_int: REAL • pool_hit_ratio_pct_for_int_enum: STRING • instance_name: STRING • tbsp_state_name: STRING	ITM_KUD_DB2_Tablespace (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
KUD_DB2_Database02 attribute group • node_name: STRING • instance_name: STRING • db_name: STRING • snapshot_time: STRING • db_partition: STRING • db_partition_enum: STRING • log_held_by_dirty_pages: REAL • log_held_by_dirty_pages_enum: STRING • log_to_redo_for_recovery: REAL • log_to_redo_for_recovery_enum: STRING • num_log_buffer_full: REAL • num_log_buffer_full_enum: STRING • num_log_data_found_in_buffer: REAL • num_log_data_found_in_buffer_enum: STRING • num_log_part_page_io: REAL • num_log_part_page_io_enum: STRING • num_log_read_io: REAL • num_log_read_io_enum: STRING • num_log_write_io: REAL • num_log_write_io_enum: STRING • total_log_available: REAL • total_log_available_enum: STRING • appl_id_oldest_xact: REAL • appl_id_oldest_xact_enum: STRING • log_read_time_ns: REAL • log_read_time_ns_enum: STRING • log_read_time_s: REAL • log_read_time_s_enum: STRING • log_write_time_ns: REAL • log_write_time_ns_enum: STRING • log_write_time_s: REAL • log_write_time_s_enum: STRING • pool_no_victim_buffer: REAL • pool_no_victim_buffer_enum: STRING • num_db_storage_paths: REAL • num_db_storage_paths_enum: STRING • num_indoubt_trans: REAL • num_indoubt_trans_enum: STRING • cat_cache_size_top: REAL • cat_cache_size_top_enum: STRING • unread_prefetch_pages: REAL • unread_prefetch_pages_enum: STRING • pool_temp_data_l_reads: REAL • pool_temp_data_l_reads_enum: STRING • pool_temp_data_p_reads: REAL • pool_temp_data_p_reads_enum: STRING • pool_temp_index_l_reads: REAL • pool_temp_index_l_reads_enum: STRING • pool_temp_index_p_reads: REAL • pool_temp_index_p_reads_enum: STRING • smallest_log_avail_node: REAL • smallest_log_avail_node_enum: STRING (Continued on the next page.)	ITM_KUD_DB2_Database02

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• shr_workspace_size_top: REAL • shr_workspace_size_top_enum: STRING • shr_workspace_num_overflows: REAL • shr_workspace_num_overflows_enum: STRING • shr_workspace_section_lookups: REAL • shr_workspace_section_lookups_enum: STRING • shr_workspace_section_inserts: REAL • shr_workspace_section_inserts_enum: STRING • priv_workspace_size_top: REAL • priv_workspace_size_top_enum: STRING • priv_workspace_num_overflows: REAL • priv_workspace_num_overflows_enum: STRING • priv_workspace_section_lookups: REAL • priv_workspace_section_lookups_enum: STRING • priv_workspace_section_inserts: REAL • priv_workspace_section_inserts_enum: STRING • pool_temp_hit_ratio: REAL • pool_temp_hit_ratio_enum: STRING • data_temp_pool_hit_ratio: REAL • data_temp_pool_hit_ratio_enum: STRING • min_catalog_cache_size: REAL • min_catalog_cache_size_enum: STRING • catalog_partition: INTEGER • catalog_partition_enum: STRING • catalog_partition_name: STRING • elapsed_exec_time_ms: REAL • elapsed_exec_time_ms_enum: STRING • elapsed_exec_time_s: REAL • elapsed_exec_time_s_enum: STRING • last_reset: STRING • pkg_cache_num_overflows: REAL • pkg_cache_num_overflows_enum: STRING • pkg_cache_size_top: REAL • pkg_cache_size_top_enum: STRING • min_pkg_cache_size: REAL • min_pkg_cache_size_enum: STRING • rows_read: REAL • rows_read_enum: STRING • sort_shrheap_allocated: REAL • sort_shrheap_allocated_enum: STRING • sort_shrheap_top: REAL • sort_shrheap_top_enum: STRING • active_hash_joins: REAL • active_hash_joins_enum: STRING (Continued on the next page.)	ITM_KUD_DB2_Database02 (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• post_shrthreshold_hash_joins: REAL • post_shrthreshold_hash_joins_enum: STRING • post_shrthreshold_sorts: REAL • post_shrthreshold_sorts_enum: STRING • pool_temp_xda_l_reads: REAL • pool_temp_xda_l_reads_enum: STRING • pool_temp_xda_p_reads: REAL • pool_temp_xda_p_reads_enum: STRING • pool_xda_l_reads: REAL • pool_xda_l_reads_enum: STRING • pool_xda_p_reads: REAL • pool_xda_p_reads_enum: STRING • pool_xda_writes: REAL • pool_xda_writes_enum: STRING • num_threshold_violations: REAL • num_threshold_violations_enum: STRING • active_olap_funcs: REAL • active_olap_funcs_enum: STRING • olap_func_overflows: REAL • olap_func_overflows_enum: STRING • total_olap_funcs: REAL • total_olap_funcs_enum: STRING • async_runstats: REAL • async_runstats_enum: STRING • stats_cache_size: REAL • stats_cache_size_enum: STRING • stats_fabricate_time: REAL • stats_fabricate_time_enum: STRING • stats_fabrications: REAL • stats_fabrications_enum: STRING • sync_runstats: REAL • sync_runstats_enum: STRING • sync_runstats_time: REAL • sync_runstats_time_enum: STRING • blocks_pending_cleanup: REAL • blocks_pending_cleanup_enum: STRING • total_log_used_pct: REAL • total_log_used_pct_enum: STRING	ITM_KUD_DB2_Database02 (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
KUD_DB2_LOG attribute group • node_name: STRING • instance_name: STRING • db_name: STRING • db_alias: STRING • snapshot_time: STRING • db_partition: STRING • db_partition_enum: STRING • backup_pending: INTEGER • backup_pending_enum: STRING • database_is_consistent: INTEGER • database_is_consistent_enum: STRING • rollforward_pending: INTEGER • rollforward_pending_enum: STRING • restore_pending: INTEGER • restore_pending_enum: STRING • logbufsz: INTEGER • logbufsz_enum: STRING • logfilsiz: REAL • logfilsiz_enum: STRING • logprimary: INTEGER • logprimary_enum: STRING • logsecond: INTEGER • logsecond_enum: STRING • newlogpath: STRING • newlogpath_size: REAL • newlogpath_size_enum: STRING • newlogpath_freesize: REAL • newlogpath_freesize_enum: STRING • logpath: STRING • logpath_size: REAL • logpath_size_enum: STRING • logpath_freesize: REAL • logpath_freesize_enum: STRING • overflowlogpath: STRING • overflow_log_path_size: REAL • overflow_log_path_size_enum: STRING • overflow_log_path_freesize: REAL • overflow_log_path_freesize_enum: STRING • mirror_log_path: STRING • mirror_log_path_size: REAL • mirror_log_path_size_enum: STRING • mirror_log_path_freesize: REAL • mirror_log_path_freesize_enum: STRING • logretain: INTEGER • logretain_enum: STRING • userexit: INTEGER • userexit_enum: STRING • logarchmeth1: STRING • logarchmeth1_size: REAL • logarchmeth1_size_enum: STRING • logarchmeth1_freesize: REAL • logarchmeth1_freesize_enum: STRING (Continued on the next page.)	ITM_KUD_DB2_LOG

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• logarchmeth2: STRING • logarchmeth2_size: REAL • logarchmeth2_size_enum: STRING • logarchmeth2_freesize: REAL • logarchmeth2_freesize_enum: STRING • failarchpath: STRING • failarchpath_size: REAL • failarchpath_size_enum: STRING • failarchpath_freesize: REAL • failarchpath_freesize_enum: STRING • numarchretry: INTEGER • numarchretry_enum: STRING • archretrydelay: INTEGER • archretrydelay_enum: STRING • sec_log_used_top: REAL • sec_log_used_top_enum: STRING • tot_log_used_top: REAL • tot_log_used_top_enum: STRING • sec_logs_allocated: REAL • sec_logs_allocated_enum: STRING • log_reads: REAL • log_reads_enum: STRING • log_writes: REAL • log_writes_enum: STRING • uow_log_space_used: REAL • uow_log_space_used_enum: STRING • total_log_used: REAL • total_log_used_enum: STRING • total_log_available: REAL • total_log_available_enum: STRING • log_held_by_dirty_pages: REAL • log_held_by_dirty_pages_enum: STRING • log_to_redo_for_recovery: REAL • log_to_redo_for_recovery_enum: STRING • log_write_time: REAL • log_write_time_enum: STRING • log_read_time: REAL • log_read_time_enum: STRING • num_log_write_io: REAL • num_log_write_io_enum: STRING • num_log_read_io: REAL • num_log_read_io_enum: STRING • num_log_part_page_io: REAL • num_log_part_page_io_enum: STRING • num_log_buffer_full: REAL • num_log_buffer_full_enum: STRING • num_log_data_found_in_buffer: REAL • num_log_data_found_in_buffer_enum: STRING • sec_log_used_pct: REAL • sec_log_used_pct_enum: STRING • total_log_used_pct: REAL • total_log_used_pct_enum: STRING • pri_log_used_pct: REAL • pri_log_used_pct_enum: STRING (Continued on the next page.)	ITM_KUD_DB2_LOG (Continued)

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
• curr_pri_log_used_pct: REAL • curr_pri_log_used_pct_enum: STRING • curr_sec_log_used_pct: REAL • curr_sec_log_used_pct_enum: STRING • first_active_log: REAL • first_active_log_enum: STRING • last_active_log: REAL • last_active_log_enum: STRING • current_active_log: REAL • current_active_log_enum: STRING • current_archive_log: REAL • current_archive_log_enum: STRING	ITM_KUD_DB2_LOG (Continued)
KUD_DB2_LOG_RECORD attribute group • node_name: STRING • instance_name: STRING • db_name: STRING • db_alias: STRING • db_partition: STRING • db_partition_enum: STRING • snapshot_time: STRING • eid: REAL • eid_enum: STRING • seqnum: REAL • seqnum_enum: STRING • start_time: STRING • end_time: STRING • first_log: STRING • last_log: STRING • backup_id: STRING • operation: STRING • operation_enum: STRING • operation_type: STRING • operation_type_enum: STRING • object_type: STRING • object_type_enum: STRING • location: STRING • device_type: STRING • device_type_enum: STRING • entry_status: STRING • entry_status_enum: STRING	ITM_KUD_DB2_LOG_RECORD

Table 30. Overview of attribute groups to event classes and slots (continued)

Event slots	IBM Tivoli Enterprise Console event class
KUD_DB2_Diagnostic_Log attribute group • node_name: STRING • instance_name: STRING • db_name: STRING • db_partition: STRING • db_partition_enum: STRING • facility: STRING • rectype: STRING • rectype_enum: STRING • timestamp: STRING • timezone: INTEGER • timezone_enum: STRING • impact: STRING • pid: REAL • pid_enum: STRING • process_name: STRING • tid: REAL • tid_enum: STRING • funcation_string: STRING • component_name: STRING • funcation_name: STRING • msgid: STRING • message_type: STRING • message_number: REAL • message_number_enum: STRING • level: STRING • message: STRING	ITM_KUD_DB2_Diagnostic_Log

Appendix D. Discovery Library Adapter for the DB2 agent

This appendix contains information about the Discovery Library Adapter (DLA) for the Tivoli Composite Application Manager Agent for DB2.

About the DLA

The Tivoli Management Services DLA discovers resources and relationships and creates a Discovery Library Book file. The Book follows the Discovery Library IdML schema and is used to populate the Configuration Management Database (CMDB) and Tivoli Business System Management products. The Tivoli Management Services DLA discovers DB2 resources. For all DB2 systems that are active and online at the Tivoli Enterprise Portal Server, information is included in the discovery book for those resources. The Tivoli Management Services DLA discovers active resources. It is run on demand and can be run periodically to discover resources that were not active during previous discoveries.

The DLA discovers DB2 components (DB2 system, DB2 server instance, database).

More information about DLAs

The following sources contain additional information about using the DLA program with all monitoring agents:

- The *IBM Tivoli Monitoring Administrator's Guide* contains information about using the Tivoli Management Services Discovery Library Adapter.
- For information about using a DLA with Tivoli Application Dependency Discovery Manager (TADDM), see the information center at http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.taddm.doc_7.1/cmdb_welcome.html.

DB2 DLA data model class types represented in CDM

This section contains information about how the various source application data objects map to classes in the Common Data Model (CDM) for the DB2 agent.

The following information is provided for each class:

Relationships
: CDM relationships (hierarchical) between currently identified model objects

CDM attributes, agent attributes, descriptions, and examples
: CDM and agent attributes that are required to create an instance of a resource, descriptions of the attributes, and examples of the attributes

ComputerSystem class

The ComputerSystem class represents the system where the DB2 agent can run.

CDM class name
: sys.ComputerSystem

Relationships
- contains

Source: INODESTS.HOSTADDR-ComputerSystem

Target: KUDIPADDR.IPADDR-IpInterface

Example: contains source="9.123.111.206-ComputerSystem" target="9.123.111.206-IpInterface"

CDM attributes, agent attributes, descriptions, and examples

- CDM attribute: Id
  Agent attribute: INODESTS.HOSTADDR-ComputerSystem
  Description: The host address of the workstation where the db2 agent instance is running.
  Example: 9.123.111.206-ComputerSystem
- CDM attribute: Name
  Agent attribute: KUDSYSRES.HOSTNAME
  Description: The name of the computer system.
  Example: CMWIN206
- CDM attribute: Signature
  Agent attribute: part of INODESTS.HOSTADDR
  Description: The IP address of the system.
  Example: 9.27.142.54
- CDM attribute: Type
  Agent attribute: ComputerSystem
  Description: A string used to appropriately render the desired icon in the TADDM User Interface.
  Example: ComputerSystem
- CDM attribute: Fqdn
  Agent attribute: part of INODESTS.HOSTADDR
  Description: The fully qualified hostname of the computer system.
  Example: CMWIN206

IpInterface class

The IpInterface class represents Layer 3 IP endpoint. Layer 3 is the Organization of Standardization (OSI) Network Layer.

CDM class name
net.IpInterface

Relationships

- contains

  Source: INODESTS.HOSTADDR-ComputerSystem

  Target: KUDIPADDR.IPADDR-IpInterface

  Example: contains source="9.123.111.206-ComputerSystem" target="9.123.111.206-IpInterface"

CDM attributes, agent attributes, descriptions, and examples

- CDM attribute: Id
  Agent attribute: KUDIPADDR.IPADDR-IpInterface
  Description: The host address of the workstation where the DB2 agent instance is running.
  Example: 9.123.111.206-IpInterface
- CDM attribute: Label
  Agent attribute: IpInterface
  Description: The label of the IP interface.
  Example: IpInterface

Db2System class

The Db2System class represents an IBM DB2 System.

CDM class name
app.db.db2.Db2System

Relationships

- runsOn

  Source: KUDIPADDR.IPADDR-Db2System

  Target: INODESTS.HOSTADDR-ComputerSystem

  Example: runsOn source=9.123.111.206-Db2System target=9.123.111.206-ComputerSystem

CDM attributes, agent attributes, descriptions, and examples

- CDM attribute: Id
  Agent attribute: KUDIPADDR.IPADDR-Db2System
  Description: The host address of the workstation where the DB2 agent instance is running.
  Example: 9.123.111.206-Db2System
- CDM attribute: Label
  Agent attribute: KUDIPADDR.SVR_NAME
  Description: A user-defined string that is used to identify the DB2 system.
  Example: cmwin206.cn.ibm.com

Db2Instance class

The Db2Instance class represents an instance within a DB2 System.

CDM class name
app.db.db2.Db2Instance

Relationships

- contains

  Source: KUDIPADDR.IPADDR-Db2System

  Target: KUDIPADDR.IPADDR-Db2Instance-KUDLOG.INAME

  Example: contains source="9.123.111.206-Db2System" Target="9.123.111.206-Db2Instance-DB2"

CDM attributes, agent attributes, descriptions, and examples

- CDM attribute: Id
  Agent attribute: KUDIPADDR.IPADDR-Db2Instance-KUDLOG.INAME
  Description: The host address of the workstation where the DB2 agent instance is running and the name of the DB2 agent instance.
  Example: 9.123.111.206-Db2Instance-DB2
- CDM attribute: KeyName
  Agent attribute: KUDLOG.INAME
  Description: The name of the DB2 agent instance.
  Example: DB2
- CDM attribute: Name
  Agent attribute: KUDLOG.INAME
  Description: The name of the DB2 agent instance.
  Example: DB2
- CDM attribute: Label

Agent attribute: Db2Instance
Description: A user-defined string that is used to identify the DB2 instance.
Example: Db2Instance

BindAddress class

The BindAddress class represents a combination of IP address and port.

CDM class name
net.BindAddress

Relationships

- accessedVia

  Source: KUDIPADDR.IPADDR-Db2Instance-KUDLOG.INAME

  Target: KUDIPADDR.IPADDR-BindAddress-KUDLOG.INAME

  Example: accessedVia source="9.123.111.206-Db2Instance-DB2" target="9.123.111.206-BindAddress-DB2"
- bindsTo

  Source: KUDIPADDR.IPADDR-IpInterface

  Target: KUDIPADDR.IPADDR-IpV4Address

  Example: bindsTo source="9.123.111.206-IpInterface" target="9.123.111.206-IpV4Address"
- bindsTo

  Source: KUDIPADDR.IPADDR-BindAddress-KUDLOG.INAME

  Target: KUDIPADDR.IPADDR-IpV4Address

  Example: bindsTo source="9.123.111.206-Db2Instance-DB2" target="9.123.111.206-IpV4Address"
- bindsAsPrimary

  Source: KUDIPADDR.IPADDR-BindAddress-KUDLOG.INAME

  Target: KUDIPADDR.IPADDR-IpV4Address

  Example: bindsTo source="9.123.111.206-Db2Instance-DB2" target="9.123.111.206-IpV4Address"

CDM attributes, agent attributes, descriptions, and examples

- CDM attribute: Id
  Agent attribute: KUDIPADDR.IPADDR-BindAddress-KUDLOG.INAME
  Description: The host address of the workstation where the DB2 agent instance is running and the name of the agent instance.
  Example: 9.123.111.206-BindAddress-DB2
- CDM attribute: Path
  Agent attribute: None
  Description: This attribute value is none.
- CDM attribute: PortNumber
  Agent attribute: KUDIPADDR.PORT
  Description: The port number of the host address.
  Example: 50000
- CDM attribute: Label
  Agent attribute: BindAddress
  Description: The label of this address
  Example: BindAddress

IpV4Address class

The IpV4Address class represents an IP V4 address.

CDM class name
net.IpV4Address

Relationships

- accessedVia

  Source: KUDIPADDR.IPADDR-Db2Instance-KUDLOG.INAME

  Target: KUDIPADDR.IPADDR-BindAddress-KUDLOG.INAME

  Example: accessedVia source="9.123.111.206-Db2Instance-DB2" target="9.123.111.206-BindAddress-DB2"
- bindsTo

  Source: KUDIPADDR.IPADDR-IpInterface

  Target: KUDIPADDR.IPADDR-IpV4Address

  Example: bindsTo source="9.123.111.206-IpInterface" target="9.123.111.206-IpV4Address"
- bindsTo

  Source: KUDIPADDR.IPADDR-BindAddress-KUDLOG.INAME

  Target: KUDIPADDR.IPADDR-IpV4Address

  Example: bindsTo source="9.123.111.206-Db2Instance-DB2" target="9.123.111.206-IpV4Address"
- bindsAsPrimary

  Source: KUDIPADDR.IPADDR-BindAddress-KUDLOG.INAME

  Target: KUDIPADDR.IPADDR-IpV4Address

  Example: bindsTo source="9.123.111.206-Db2Instance-DB2" target="9.123.111.206-IpV4Address"

CDM attributes, agent attributes, descriptions, and examples

- CDM attribute: Id
  Agent attribute: KUDIPADDR.IPADDR-IpV4Address
  Description: The host address of the workstation where the DB2 agent instance is running.
  Example: 9.123.111.206-IpV4Address
- CDM attribute: DotNotation
  Agent attribute: KUDIPADDR.IPADDR
  Description: An IPv4 address in string form.
  Example: 9.123.111.206
- CDM attribute: Label
  Agent attribute: KUDIPADDR.IPADDR
  Description: The label of this address
  Example: 9.123.111.206

Db2Database class

The Db2Database class epresents an instance of the DB2 Database.

CDM class name
app.db.db2.Db2Database

Relationships

- contains

  Source: KUDIPADDR.IPADDR-Db2Instance-KUDLOG.INAME

Target: KUDIPADDR.IPADDR-Database-KUDLOG.DBNM

Example: contains source="9.123.111.206-Db2Instance-DB2" target="9.123.111.206-Database-MYDB6"

CDM attributes, agent attributes, descriptions, and examples

- CDM attribute: Id
  Agent attribute: KUDIPADDR.IPADDR-Database-KUDLOG.DBNM
  Description: The host address of the workstation where the monitored database instance is running and the name of the database instance.
  Example: 9.123.111.206-Database-MYDB6
- CDM attribute: Name
  Agent attribute: KUDLOG.DBNM
  Description: The name of the database.
  Example: MYDB6
- CDM attribute: Alias
  Agent attribute: KUDLOG.DBA
  Description: The alias of the database.
  Example: MYDB6
- CDM attribute: Label
  Agent attribute: Db2Database
  Description: The label of DB2 databases
  Example: Db2Database

IpV6Address

The IpV6Address represents an IP V6 address.

CDM class name
net.IpV6Address

Relationships

- bindsTo

  Source: KUDIPADDR.IPADDR-IpInterface

  Target: com.ibm.tivoli.monitoring.tmsdla.IPv6Addr-IpV6Address
- bindsTo

  Source: com.ibm.tivoli.monitoring.tmsdla.IPv6Addr-BindAddress-KUDLOG.INAME

  Target: com.ibm.tivoli.monitoring.tmsdla.IPv6Addr-IpV6Address

CDM attributes, agent attributes, descriptions, and examples

- CDM attribute: Id
  Agent attribute: com.ibm.tivoli.monitoring.tmsdla.IPv6Addr-IpV6Address
  Description: The host address of the workstation where the DB2 agent instance is running.
  Example: :135.75.43.52-IpV6Address
- CDM attribute: DotNotation
  Agent attribute: com.ibm.tivoli.monitoring.tmsdla.IPv6Addr
  Description: An IPv6 address in string form.
  Example: :135.75.43.52
- CDM attribute: Label
  Agent attribute: com.ibm.tivoli.monitoring.tmsdla.IPv6Addr
  Description: The label of this IP address.
  Example: :135.75.43.52

TMSAgent

The TMSAgent class represents the Tivoli Monitoring Services Agent.

CDM class name
app.TMSAgent

Relationships

- runsOn

  Source: INODESTS.ORIGINNODE-TMSAgent

  Target: NODESTS.HOSTADDR-ComputerSystem

  Example: runsOn source="db2inst1:tivpc029:UD-TMSAgent" target="9.123.98.138-ComputerSystem"

- monitors

  Source: INODESTS.ORIGINNODE-TMSAgent

  Target: NODESTS.HOSTADDR-Db2Instance-{first part of INODESTS.ORIGINNODE}

  Example: monitors source="db2inst1:tivpc029:UD-TMSAgent" target="9.123.98.138-Db2Instance-db2inst1"

CDM attributes, agent attributes, descriptions, and examples

- CDM attribute: Id
  Agent attribute: INODESTS.ORIGINNODE-TMSAgent
  Description: The name of the DB2 agent.
  Example: DB2:CMWIN206:UD-TMSAgent
- CDM attribute: ManagedObjectName
  Agent attribute: p@INODESTS.ORIGINNODE
  Description: The name of the IBM Tivoli Monitoring (ITM) component that provides data for the management of the DB2 agent instance.
  Example: p@DB2:CMWIN206:UD
- CDM attribute: SoftwareVersion
  Agent attribute: INODESTS.VERSION
  Description: The version of the DB2 agent instance.
  Example: 06.20.00
- CDM attribute: ProductCode
  Agent attribute: INODESTS.PRODUCT
  Description: The product code of DB2 agent.
  Example: UD
- CDM attribute: Affinity
  Agent attribute: INODESTS.AFFINITIES
  Description: The affinity of DB2 agent.
  Example: 0000000000000000000W000000000000W0000yw0a7
- CDM attribute: Name
  Agent attribute: INODESTS.ORIGINNODE-DB2Agent
  Description: The name of the DB2 agent.
  Example: db2inst1:tivsun15:UD-DB2Agent
- CDM attribute: Label
  Agent attribute: INODESTS.ORIGINNODE-DB2Agent
  Description: The label of the DB2 agent.
  Example: db2inst1:tivsun15:UD-DB2Agent

Appendix E. DB2 agent data collection

The DB2 agent collects non-aggregated monitor data from single database partitions and aggregated monitor data across all partitions in a database.

DB2 system monitor

Data collected by the DB2 system monitor is organized into logical groups, with the data collection for each group being controlled by a set of DB2 monitor switches. These monitor switches enable or disable the collection of the logical groups of monitored data for the system monitor.

Because the DB2 agent collects attribute data from the DB2 system monitor, the DB2 agent requires dependent monitor switches to be enabled while the agent is running. To that end, the DB2 agent automatically enables the required set of monitor switches during agent startup. Because of DB2 data collection constraints, monitor switch enablement is at the database manager level. When the DB2 agent is stopped, the monitor switch settings are restored to their previous values.

Table 31 lists the DB2 monitor switches that are automatically enabled by the DB2 agent and the associated information whose collection is controlled by that monitor switch.

Table 31. DB2 monitor switches

Monitor switch	Monitor information
BUFFERPOOL	Number of reads and writes, time taken
LOCK	Lock wait times, deadlocks
SORT	Number of heaps used, sort performance
STATEMENT	Start and stop time, statement identification
TABLE	Measure of activity (rows read or written)
TIMESTAMP	Timestamps
UOW	Start and end times, completion status

Partitioned environments

In a partitioned DB2 environment, many attributes are available as either raw, non-aggregated data from individual database partitions or as aggregated data across all database partitions. The DB2 agent can monitor this data for a specific single partition, the current partition, or an aggregation across all database partitions. All attribute groups that contain the db_partition attribute are capable of returning partition-specific data.

By default, the workspaces and situations provided by the DB2 agent request data for the current partition. To change this default behavior, you must create a query filter or situation filter using the db_partition attribute. Set the filter value to one or more of the desired partitioning numbers (0 – 999), current partition, or partition aggregation. To monitor all partitions, set filter value to "!= Aggregated", and all the partition data except the aggregated data is covered. An empty filter results in the default behavior. When an attribute group containing a db_parition attribute is enabled for historical data collection, data is collected for each individual partition and the aggregation; this behavior cannot be configured.

Appendix F. Documentation library

This appendix contains information about the publications related to the DB2 agent. These publications are listed in the following categories:

- DB2 agent library
- Prerequisite publications
- Related publications

See the *IBM Tivoli Monitoring, OMEGAMON XE, and Composite Application Manager products: Documentation Guide*, SC23-8816, for information about accessing and using publications. You can find the *Documentation Guide* in the Information Center at http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/topic/com.ibm.docguide.doc/DocGuide.htm.

DB2 agent library

There is one document specific to the DB2 agent: *IBM Tivoli Composite Application Manager Agent for DB2 User's Guide*. This user's guide provides agent-specific reference and troubleshooting information for configuring and using the DB2 agent.

Use the configuration chapter in this guide with the *IBM Tivoli Monitoring Installation and Setup Guide* to set up the software.

Use the information in this guide with the *IBM Tivoli Monitoring User's Guide* to monitor DB2 Universal Database resources.

Prerequisite publications

To use the information in this publication effectively, you must have some prerequisite knowledge, which you can obtain from the following IBM Tivoli Monitoring publications:

- *Exploring IBM Tivoli Monitoring*
- *IBM Tivoli Monitoring Administrator's Guide*
- *IBM Tivoli Monitoring Agent Builder User's Guide*
- *IBM Tivoli Monitoring Command Reference*
- *IBM Tivoli Monitoring Installation and Setup Guide*
- *IBM Tivoli Monitoring: Messages*
- *IBM Tivoli Monitoring Migration Toolkit User's Guide*
- *IBM Tivoli Monitoring Troubleshooting Guide*
- *IBM Tivoli Monitoring: Upgrading from Tivoli Distributed Monitoring*
- *Tivoli Enterprise Portal User's Guide*
- *IBM Tivoli Monitoring: i5/OS Agent User's Guide*
- *IBM Tivoli Monitoring: Linux OS Agent User's Guide*
- *IBM Tivoli Monitoring Universal Agent User's Guide*
- *IBM Tivoli Universal Agent API/Command Programming Reference Guide*
- *IBM Tivoli Monitoring: UNIX Log OS Agent User's Guide*
- *IBM Tivoli Monitoring: UNIX OS Agent User's Guide*

- *IBM Tivoli Monitoring: Upgrading from V5.1.2*
- *IBM Tivoli Monitoring: Windows OS Agent User's Guide*

Related publications

The following documents also provide useful information:

- *IBM Tivoli Enterprise Console Adapters Guide*
- *IBM Tivoli Enterprise Console Event Integration Facility User's Guide*
- *IBM Tivoli Enterprise Console Reference Manual*
- *IBM Tivoli Enterprise Console Rule Builder's Guide*

Other sources of documentation

You can also obtain technical documentation about Tivoli Monitoring and OMEGAMON XE products from the following sources:

- IBM Tivoli Open Process Automation Library (OPAL)

  http://www.ibm.com/software/tivoli/opal

  OPAL is an online catalog that contains integration documentation and other downloadable product extensions. This library is updated daily.
- Redbooks

  http://www.redbooks.ibm.com/

  IBM Redbooks®, Redpapers, and Redbooks Technotes provide information about products from platform and solution perspectives.
- Technotes

  You can find Technotes through the IBM Software Support Web site at http://www.ibm.com/software/support/probsub.html, or more directly through your product Web site, which contains a link to Technotes (under **Solve a problem**).

  Technotes provide the latest information about known product limitations and workarounds.

Appendix G. Accessibility

Accessibility features help users with physical disabilities, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in this product enable users to do the following things:

- Use assistive technologies, such as screen-reader software and digital speech synthesizer, to hear what is displayed on the screen. Consult the product documentation of the assistive technology for details on using those technologies with this product.
- Operate specific or equivalent features using only the keyboard.
- Magnify what is displayed on the screen.

In addition, the product documentation was modified to include the following features to aid accessibility:

- All documentation is available in both HTML and convertible PDF formats to give the maximum opportunity for users to apply screen-reader software.
- All images in the documentation are provided with alternative text so that users with vision impairments can understand the contents of the images.

Navigating the interface using the keyboard

Standard shortcut and accelerator keys are used by the product and are documented by the operating system. Refer to the documentation provided by your operating system for more information.

Magnifying what is displayed on the screen

You can enlarge information on the product windows using facilities provided by the operating systems on which the product is run. For example, in a Microsoft Windows environment, you can lower the resolution of the screen to enlarge the font sizes of the text on the screen. Refer to the documentation provided by your operating system for more information.

Appendix H. 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:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 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 is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating systems. 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 system 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.

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

Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. 2005-2012. All rights reserved.

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

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information? at www.ibm.com/legal/copytrade.shtml.

Adobe and the Adobe log are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Intel, Intel logo, and Intel Xeon, are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

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

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

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

Other company, product, or service names may be trademarks or service marks of others.

Index

Special characters

A

B

C

D

E

F

G

H

I

L

M

N

U

W

Printed in USA

SC27-2491-01