

# Unicenter TNG<sup>®</sup>

---

CA-IDMS<sup>®</sup> Agent  
Using the CA-IDMS Agent  
2.1



Computer Associates™

This documentation and related computer software program (hereinafter referred to as the "Documentation") is for the end user's informational purposes only and is subject to change or withdrawal by Computer Associates International, Inc. ("CA") at any time.

THIS DOCUMENTATION MAY NOT BE COPIED, TRANSFERRED, REPRODUCED, DISCLOSED, OR DUPLICATED, IN WHOLE OR IN PART, WITHOUT THE PRIOR WRITTEN CONSENT OF CA. THIS DOCUMENTATION IS PROPRIETARY INFORMATION OF CA AND PROTECTED BY THE COPYRIGHT LAWS OF THE UNITED STATES AND INTERNATIONAL TREATIES.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO THE END USER OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED OF SUCH LOSS OR DAMAGE.

THE USE OF ANY PRODUCT REFERENCED IN THIS DOCUMENTATION AND THIS DOCUMENTATION IS GOVERNED BY THE END USER'S APPLICABLE LICENSE AGREEMENT.

The manufacturer of this documentation is Computer Associates International, Inc.

Provided with "Restricted Rights" as set forth in 48 C.F.R. Section 12.212, 48 C.F.R. Sections 52.227-19(c)(1) and (2) or DFARS Section 252.227.7013(c)(1)(ii) or applicable successor provisions.

**First Edition, December 2000**

© 2000 Computer Associates International, Inc.  
One Computer Associates Plaza, Islandia, NY 11749  
All rights reserved.

All trademarks, trade names, service marks, or logos referenced herein belong to their respective companies.

# Contents

---

<b>Chapter 1. Welcome!</b>	1-1
1.1 Overview	1-3
1.2 Who Should Read This Guide	1-4
1.3 What an Agent Is	1-5
1.3.1 Role of the MIB	1-5
1.4 What the Unicenter TNG CA-IDMS Agent Is	1-6
1.4.1 The CA-IDMS MIB	1-6
1.4.2 The CA-IDMS Agent	1-7
1.5 Using Unicenter TNG Tools to View CA-IDMS System Health	1-8
1.6 What the Unicenter TNG CA-IDMS Agent Monitors	1-9
1.6.1 Event Monitoring	1-9
1.6.2 Status Monitoring	1-10
1.6.2.1 CPU Monitoring	1-11
1.6.2.2 I/O Monitoring	1-11
1.6.2.3 Buffer Monitoring	1-12
1.6.2.4 Transaction Log Monitoring	1-12
1.6.2.5 Lock Monitoring	1-13
1.6.2.6 Database Monitoring	1-13
1.6.2.7 Workload Monitoring	1-14
1.6.2.8 Memory Monitoring	1-15
1.6.2.9 DC Resources	1-15
1.6.2.10 Network Monitoring	1-16
1.7 Hardware and Software Requirements	1-17
1.7.1 For the Mainframe Platform	1-17
1.7.2 For the PC Platform	1-17
1.8 Where to Find More Information	1-18
<b>Chapter 2. Installing and Configuring Unicenter TNG CA-IDMS Agent</b>	2-1
2.1 Overview	2-3
2.2 Installing, Configuring, and Starting Agent Technology	2-4
2.3 Installing CA-IDMS Agent on the Workstation	2-5
2.4 Installing CA-IDMS Agent on the Mainframe	2-6
2.5 Modifying a CA-IDMS System to Be Monitored	2-7
2.5.1 Increasing Operating System Storage	2-7
2.5.2 Increasing CA-IDMS Region Storage	2-7
2.5.3 Modifying CA-IDMS System Startup JCL	2-7
2.5.4 Specifying SYSAGNT Option File Startup Parameters	2-8
2.5.5 Defining the AGMT Task	2-10
2.5.6 Modify the SYSTEM Statement	2-11
2.5.7 Modify External Security Privileges	2-11
2.6 Starting a CA-IDMS Agent	2-12
2.7 Controlling and Monitoring Agent Execution	2-13
2.7.1.1 AGMT Task Code Syntax	2-13
2.7.1.2 Usage Considerations	2-14
2.8 Configuring the CA-IDMS Agent	2-16
2.8.1 Specifying Agent Configuration Attributes Values	2-16
2.8.2 Setting Initial Configuration Values	2-17

2.8.3	Overriding Trap Destinations and Community Strings	2-19
2.8.4	Tailoring CA-IDMS Event Detection	2-19
2.8.4.1	The IDMSTRAP Table	2-20
2.8.4.2	#TRAP Macro	2-20
<b>Chapter 3.</b>	<b>Using Unicenter TNG with the CA-IDMS Agent</b>	3-1
3.1	Accessing Information About CA-IDMS Resources	3-3
3.2	Viewing CA-IDMS Resources from Node View	3-5
3.2.1	Understanding Node View Resource Colors	3-5
3.2.2	Displaying and Hiding Nodes	3-5
3.2.3	Displaying and Hiding Mini Trees	3-6
3.2.4	Using the Node View Toolbar	3-6
3.2.5	Viewing Information About a Particular CA-IDMS Resource	3-7
3.3	Viewing CA-IDMS Resources from Agent View	3-9
3.3.1	Using the Agent View Toolbar	3-9
3.3.2	Getting More Detailed Information	3-10
3.4	Browsing the CA-IDMS MIB	3-12
3.5	Obtaining Event History Information for a Resource	3-13
3.6	Integration with Unicenter Event Management	3-14
<b>Appendix A.</b>	<b>Enterprise Specific Traps for the CA-IDMS Agent</b>	A-1
A.1	Overview	A-3
A.2	Status Monitor Traps	A-4
A.2.1	CPU-Related Traps	A-4
A.2.2	I/O-Related Traps	A-5
A.2.3	Network-Related Traps	A-5
A.2.4	Buffer-Related Traps	A-5
A.2.5	Log/Journal-Related Traps	A-5
A.2.6	Locking-Related Traps	A-6
A.2.7	Database-Related Traps	A-6
A.2.8	Workload-Related Traps	A-7
A.2.9	Memory-Related Traps	A-8
A.2.10	DC Resource-Related Traps	A-8
A.3	Event Monitor Traps	A-9
<b>Appendix B.</b>	<b>CA-IDMS Agent Messages</b>	B-1
B.1	IDMSACTL Messages	B-3
B.2	IDMSAG00 Messages	B-6

# Chapter 1. Welcome!

---

1.1 Overview	1-3
1.2 Who Should Read This Guide	1-4
1.3 What an Agent Is	1-5
1.3.1 Role of the MIB	1-5
1.4 What the Unicenter TNG CA-IDMS Agent Is	1-6
1.4.1 The CA-IDMS MIB	1-6
1.4.2 The CA-IDMS Agent	1-7
1.5 Using Unicenter TNG Tools to View CA-IDMS System Health	1-8
1.6 What the Unicenter TNG CA-IDMS Agent Monitors	1-9
1.6.1 Event Monitoring	1-9
1.6.2 Status Monitoring	1-10
1.6.2.1 CPU Monitoring	1-11
1.6.2.2 I/O Monitoring	1-11
1.6.2.3 Buffer Monitoring	1-12
1.6.2.4 Transaction Log Monitoring	1-12
1.6.2.5 Lock Monitoring	1-13
1.6.2.6 Database Monitoring	1-13
1.6.2.7 Workload Monitoring	1-14
1.6.2.8 Memory Monitoring	1-15
1.6.2.9 DC Resources	1-15
1.6.2.10 Network Monitoring	1-16
1.7 Hardware and Software Requirements	1-17
1.7.1 For the Mainframe Platform	1-17
1.7.2 For the PC Platform	1-17
1.8 Where to Find More Information	1-18



## 1.1 Overview

Welcome to the Unicenter TNG CA-IDMS Agent, or simply, the CA-IDMS Agent — the technology that makes it easy for you to monitor and manage your CA-IDMS systems from a single Unicenter TNG workstation console. With this product, you can tell at a glance if a CA-IDMS system needs your attention. Using easy-to-understand icons, symbols, and color, you can respond to critical situations that arise. As you drill down for more information, CA-IDMS Agent displays statistics about your CA-IDMS system using graphical displays that let you easily identify problem areas.

## 1.2 Who Should Read This Guide

This guide is for CA-IDMS database administrators and systems administrators, as well as other technical personnel, responsible for configuring and maintaining the CA-IDMS Agent.

## 1.3 What an Agent Is

An agent is an application that supports network management. An agent, such as the CA-IDMS Agent, provides information to a management application, such as Unicenter TNG, with a simplified and standardized view of monitored data. A management protocol that is understood by both managers and agents standardizes management information and is communicated according to a communications protocol. The Unicenter TNG agents use the following management and communication protocols:

- The communications protocol used is the user datagram protocol (UDP) of the transmission control protocol/Internet protocol (TCP/IP) suite.
- The network management protocol is the simple network management protocol (SNMP) designed to run on top of TCP/IP.

### 1.3.1 Role of the MIB

A Management Information Base (MIB) describes the collection of attributes that describe a managed resource. Both the agent and the manager can view the MIB attributes. Each MIB attribute is characterized by:

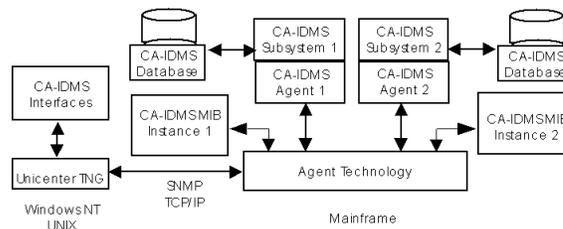
- A name
- A unique object identifier (OID) that conforms to a centrally maintained registration authority
- Type (for example, integer [I] or string [S])
- Access permission (for example, read-only [R], read-write [RW], or not-accessible [NA])
- A detailed description

The MIB is stored in runtime memory, and is populated at runtime by a combination of initial configuration files and SNMP requests. The Unicenter TNG MIB Browser utility lets you view and change MIB attributes at runtime.

## 1.4 What the Unicenter TNG CA-IDMS Agent Is

The CA-IDMS Agent is software that executes as a task in the CA-IDMS region on the mainframe. It detects events that occur in the CA-IDMS system and relays them to the Unicenter TNG console administrator using the Simple Network Management Protocol (SNMP).

This diagram shows the architecture of the CA-IDMS Agent:



The CA-IDMS Agent monitors events and performance-related statistics in the CA-IDMS system. It uses Unicenter TNG Agent Technology for OS/390 (referred to in this manual as Agent Technology) to store these statistics in the MIB and to send SNMP traps to Unicenter TNG when significant events occur or thresholds are exceeded. Unicenter TNG provides interfaces that display the status of the CA-IDMS systems and the information in the MIB.

The CA-IDMS MIB and CA-IDMS Agent are described more fully below. For more information about Agent Technology, see *Unicenter TNG Agent Technology for OS/390 Getting Started*.

### 1.4.1 The CA-IDMS MIB

The CA-IDMS Management Information Base, or CA-IDMS MIB, resides on the mainframe, and defines the information that the Agent detects and reports on. For example, these lines of code from the CA-IDMS MIB identify an object, `idmsCfgCPUPollInterval`, which defines the collection interval for the CPU monitor:

```
idmsCfgCPUPollInterval          OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The CPU monitor collection interval, in seconds. The
        default is 90,"
    ::= { idmsCfgCPU 2}
```

The CA-IDMS MIB contains two object groups:

- The Configuration Group, which includes startup control values, monitoring values, polling intervals, thresholds, and damping counts.

- The Status Group, which collects statistics of current control values and statistics such as percentages of totals, rates, differences, thresholds, and high water marks.

You can view the MIB contents using Object View in Unicenter TNG WorldView. A MIB Browser is also available from Node View. The unique Object Identifier (OID) for the CA-IDMS MIB is 1.3.6.1.4.1.791.3.9.28 and its name is caiIDMS

## 1.4.2 The CA-IDMS Agent

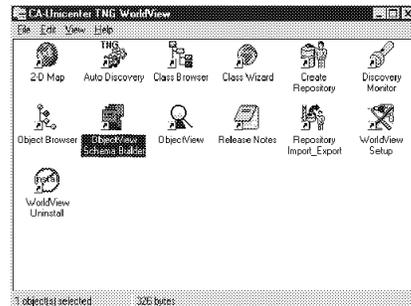
The CA-IDMS Agent consists of:

- The Event Monitor, which sends SNMP traps to the Unicenter TNG Event Manager when it detects an event. An event can be something like a CA-IDMS DC log message or DB status code. Each event is associated with a severity level — critical, warning, and information. You can configure which messages and status codes result in traps by using the IDMSTRAP table. More information about the IDMSTRAP table appears under Tailoring CA-IDMS Event Detection in Chapter 2, "Installing and Configuring CA-IDMS Agent on the Host."
- A Status Monitor, which collects CA-IDMS system statistics at specified intervals and updates the CA-IDMS MIB. The Status Monitor also compares current statistics to threshold values defined in the CA-IDMS MIB and sends a trap when the threshold is exceeded. You can configure the Status Monitor, by setting the polling interval, setting the threshold levels, and defining *damping* counts. A damping count is a way to avoid sending traps when a threshold is exceeded occasionally; instead, the damping count sends a trap when the threshold is exceeded for a specified number of consecutive polling intervals. The Status Monitor also provides the ability to display and cancel active tasks.

For more information about the Event and Status Monitors, see What the CA-IDMS Agent Monitors later in this chapter.

## 1.5 Using Unicenter TNG Tools to View CA-IDMS System Health

Unicenter TNG WorldView provides the tools you need to assess the health of your CA-IDMS systems. Unicenter TNG WorldView is the top-level Unicenter TNG user interface. From WorldView, you can view 2D and 3D maps, topological network structures, and show the state of networks and nodes.



Some of the Unicenter TNG WorldView tools are described below. For detailed descriptions of using these tools with the CA-IDMS Agent, see Chapter 3, "Using Unicenter TNG with the CA-IDMS Agent."

**Object View:** Object View lets you view MIB variables directly. It has a Windows Explorer type interface and allows user-defined "dashboards" where you can view MIB information in the form of graphs and charts.

**Node View:** In Node View, you can view managed objects within a node as a tree. The display follows the structure of the CA-IDMS MIB. The color of each node indicates the state of the managed object.

**MIB Browser:** The MIB Browser is accessible from Node View and is an alternative way to view the CA-IDMS Agent MIB.

**Agent View:** In Agent View, you can view a graphical display of CA-IDMS Agent statistics by group; for example CPU status, I/O status, and so on. You can view detailed statistics in graphs of current values and charts of historical values. Agent View also provides dialogs which let you set monitoring levels, thresholds, and damping counts.

**Event Browser:** The Event Browser is accessible from Node View and displays messages associated with managed object state changes.

## 1.6 What the Unicenter TNG CA-IDMS Agent Monitors

The CA-IDMS Agent is made up of two monitors — the Event Monitor and the Status Monitor. The Event Monitor detects events as they occur. The Status Monitor periodically examines critical statistics within the CA-IDMS system and compares them to thresholds to determine the health of the system.

Although both monitors run within the CA-IDMS address space, they execute as separate subtasks and hence run independently of the CA-IDMS system being monitored.

### 1.6.1 Event Monitoring

The Event Monitor detects:

- Messages written to the CA-IDMS log
- Error statuses set by the database management system
- Events such as system startup and shutdown

Each time it detects one of these events, the Event Monitor issues an SNMP trap. Depending on the severity of the event, the trap may cause the system status to change. Unicenter TNG tools such as World View and Node View display the change in status by changing the color of the icon representing the CA-IDMS system. The Event Monitor also adds a message to a list of messages in a table in the MIB, which you can view through the MIB Browser or Agent View. You can also view these messages through the Event Browser.

You can tailor the Event Monitor to detect specific events for your CA-IDMS system. For example, you can specify:

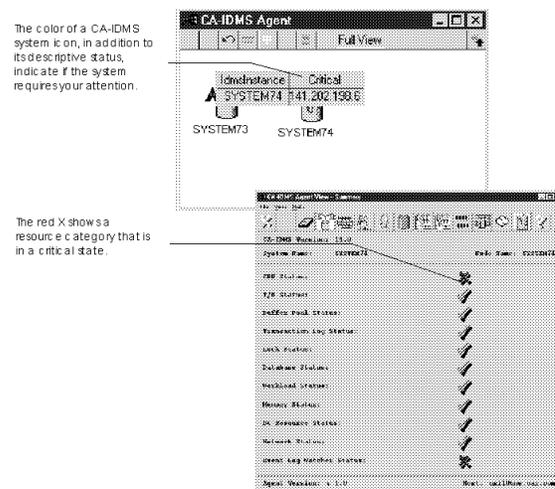
- Which messages and error statuses generate traps and you can also specify an associated severity through the IDMSTRAP table
- The maximum number of messages that will be retained in the message table and the wait interval for the Event Monitor. The wait interval is the maximum number of seconds the Event Monitor will wait for an event to occur before giving up control so that messages from the administrator (SNMP requests) or control messages (from the agentctrl utility) can be processed. Once any control messages have been processed, the Event Monitor will again wait to be posted by the CA-IDMS system.

**Note:** The response time to these requests and control messages will depend on the value of the wait interval. To avoid too long a delay for the response, its value should be less than 30.

## 1.6.2 Status Monitoring

The Status Monitor periodically gathers statistics by examining control blocks within the CA-IDMS region. It then calculates metrics which it uses to assess the health of the system. The Status Monitor compares these metrics to user-specified thresholds which define warning and critical levels.

If a threshold is exceeded, an SNMP trap is generated to inform Unicenter TNG about an exceptional condition. The trap may cause the system status to change, which is reported by changing the color of the CA-IDMS system icon in Unicenter tools such as World View and Node View. Within Node View and Agent View, the particular resource category, for example, I/O usage, that is associated with the metric, is highlighted. You can view the metric and related statistics through the MIB Browser and Agent View.



The Status Monitor captures statistics for these resource categories:

- CPU
- I/O
- Buffer pools
- Transaction logs
- Locks
- Database
- Workload
- Memory
- DC resources
- Network

The Status Monitor monitors each of these categories independently. You can enable and disable monitoring for each. You can also set an individual polling interval for each category; the polling interval determines how frequently statistics are collected.

Each metric within a category has associated warning and critical thresholds that determine the exception levels for a particular CA-IDMS system. By changing these thresholds, you can tailor the monitoring algorithm to match the processing profile of the system. Certain metrics have an associated damping count that is used to eliminate temporary spikes from consideration. By setting a damping count to 5, for example, the CA-IDMS Agent only issues a trap when the associated threshold has been exceeded for at least five consecutive polling intervals.

A brief description of the metrics used within each resource category appears below. In addition to these, many other statistics are gathered and included in the MIB. For detailed information about a specific metric, use the help provided with the CA-IDMS Agent.

### 1.6.2.1 CPU Monitoring



The CA-IDMS Agent uses two metrics to assess CPU activity within a CA-IDMS system, CPU Utilization and CPU Degradation.

**CPU Utilization:** CPU Utilization is a measure of the CPU consumed by the CA-IDMS system. It is calculated as the total system and user mode CPU time as a percentage of the elapsed time. CPU Utilization exceeding 100% may occur if multitasking is in effect. An unusually high CPU Utilization for a system may indicate a runaway task.

**CPU Degradation:** CPU Degradation is a measure of the CPU that is unavailable to the CA-IDMS system due to MVS overhead such as paging. It is calculated as 100 minus the total system, user and dispatch CPU times as a percentage of the time that the CPU was available to the CA-IDMS region. A moderately loaded CA-IDMS system should experience CPU Degradation of 10% or less. A lightly loaded system may appear to experience high CPU Degradation, but this can be ignored.

### 1.6.2.2 I/O Monitoring



The CA-IDMS Agent uses a single metric to assess I/O activity within a CA-IDMS system. The I/O Rate is calculated as the number of file I/Os issued per second. A higher than normal rate may indicate a runaway task.

### 1.6.2.3 Buffer Monitoring



There are three metrics used to assess the health of the buffering system — the number of BCR Waits, the Buffer Miss Percentage and the Forced Write Percentage.

**BCR Wait:** A BCR Wait occurs when a task waits to gain access to a buffer control block which is in use by another task. BCR Waits should never occur unless one or more buffer pools are extremely small. Increasing a small buffer pool by only one or two pages will probably eliminate all BCR Waits. There are no threshold values associated with BCR Waits. If any BCR Wait occurs within an interval, it is considered a warning-level event. You may, however, turn off the monitoring of BCR Waits altogether.

**Buffer Miss Percentage:** The Buffer Miss Percentage is a measure of the effectiveness of the size of the CA-IDMS system's buffer pools. A buffer miss occurs when a requested database page is not in a buffer pool and must be read in from cache or a file. The Buffer Miss Percentage is calculated as the number of buffer misses as a percentage of the number of pages requested. A high Buffer Hit Percentage indicates that the size of one or more buffer pools should be increased.

**Forced Write Percentage:** A forced write occurs when a page must be written to disk in order to free the buffer for use by another database page. The Forced Write Percentage is calculated as the number of forced writes as a percentage of the number of pages written. A high percentage indicates that the size of one or more buffer pools should be increased or that a poor level of clustering has been achieved.

### 1.6.2.4 Transaction Log Monitoring



The CA-IDMS Agent uses Journal Full Percentage to monitor the status of the journaling system and Log Full Percentage to monitor the health of logging.

**Journal Full Percentage:** The Journal Full Percentage is calculated as the number of journal file blocks in use as a percentage of the total number of journal blocks. As journal files fill, they are offloaded. Since a system normally has two or more journal files, the system can write information to one file while another is being offloaded. If the Percentage Full continues an overall upward trend, it may indicate that a long-running update job is not issuing Commits frequently enough or that the offload job is not executing successfully.

**Log Full Percentage:** The Log Full Percentage is calculated as the amount of log space in use as a percentage of log space allocated. It is normal for the log file to fill to the 75% level at which time log images are offloaded by an archive job. If the log

becomes full and remains so for several polling intervals, it may indicate either that the log file is too small or that the log offload did not execute successfully.

### 1.6.2.5 Lock Monitoring



The number of Lock Storage Overflows is the metric used to assess lock management. Each time a lock is granted, one or more lock control blocks must be allocated. CA-IDMS allocates the lock control blocks from a storage area acquired at system startup. If the system needs more storage due to the current number of locks, the Lock Manager acquires more. When that storage is no longer needed, it is freed. While this mechanism enables the CA-IDMS system to continue to function during a period of intensive lock usage, it involves additional CPU overhead. If Lock Storage Overflows occur frequently, you should increase the SYSLOCKS parameter in your CA-IDMS system generation. If an unusually high number of storage overflows occur, it may indicate either a runaway task or a long-running update job that is not issuing Commits frequently enough.

### 1.6.2.6 Database Monitoring



CA-IDMS Agent monitors the following database-related metrics:

- Record Overflow Percentage
- Record Fragment Percentage
- Index Split Rate
- Index Spawns
- SQL Auto-Recompiles

Each of these are described below.

**Record Overflow Percentage:** The Record Overflow Percentage is the number of CALC and VIA record overflows as a percentage of CALC and VIA records stored. A general increase in this metric indicates that one or more areas in the database are becoming full or that the page size of one or more areas is too small to hold all the records targeting to a page.

**Record Fragment Percentage:** The Record Fragment Percentage is the number of record fragments stored as a percentage of CALC and VIA records stored. A general increase in this metric either indicates a page size that is too small, an area becoming full, or insufficient page reserve in one or more areas of the database.

**Index Split Rate:** The Index Split Rate is measured as the average number of index splits per minute in the last interval. A high split rate may indicate that one or more indexes need to be re-organized, potentially increasing the maximum number of entries per index control record.

**Index Spawns:** An Index Spawn occurs when the number of levels within an index increases. The more levels in an index, the more overhead is necessary to search the index for a given entry. An Index Spawn may indicate an index needs to be re-organized with a higher maximum number of entries per index control record.

**SQL Automatic Recompiles:** An automatic recompilation occurs during SQL access either because the definition of the database has changed or the program containing the SQL statements has been recompiled. Typically, auto-recompiles should be avoided in production environments by manually recompiling affected access modules as part of making the change. In a test environment, auto-recompiles occur frequently and should not be considered an exceptional condition.

### 1.6.2.7 Workload Monitoring



Workload monitoring monitors key metrics and displays and cancels active tasks. The following metrics are measured:

- Task Abend Percentage
- Transaction Deadlock Percentage
- Transaction Rollback Percentage
- System Run Unit Overflow Percentage

**Task Abend Percentage:** The Task Abend Percentage is the number of tasks that abnormally terminate as a percentage of the tasks that finish processing. In a production environment, this metric should have a very low value; in fact any task abend may require investigation.

**Transaction Deadlock Percentage:** The Transaction Deadlock Percentage is the number of transactions that deadlocked as a percentage of the transactions that finished processing.

**Transaction Rollback Percentage:** The Transaction Rollback Percentage is the number of transactions that were rolled out as a percentage of the transactions that were terminated. In a production environment, this number should be very small.

**System Run Unit Overflow Percentage:** A system run unit overflow occurs when a new system-managed run unit must be created in order to service a request for such things as message text retrieval or module loading. Normally, these run units are allocated from a pool built at system startup. If, however, all predefined run units are in use, the system must create a new one. The System Run Unit Overflow Percentage

is calculated as the number of overflow allocations as a percentage of the total number of allocations. If this metric is high, you should predefine more run units at system startup.

**Monitoring Tasks:** Workload monitoring lets you display information about tasks that are currently executing within the CA-IDMS system. The CA-IDMS Agent captures task identification information together with summarized resource utilization for each task.

A separate view within Agent View displays active tasks. You can also cancel a task. Since the CA-IDMS Agent is executing as a separate MVS subtask, it can take corrective action such as canceling a task event when you are unable to do so using DCMT commands because the system cannot process new tasks.

### 1.6.2.8 Memory Monitoring



Three components of the CA-IDMS system are monitored under the memory resource category — storage usage, program pool usage, and scratch usage.

**Storage Usage:** Storage Usage is measured in terms of the amount of storage space in use. If this value approaches 100%, the CA-IDMS system may go into a short-on-storage condition, which may cause one or more tasks to fail. The CA-IDMS Agent also monitors whether a short-on-storage condition occurred within the last polling interval.

**Program Pool Usage:** Program Pool Usage is measured in terms of the number of overlays of programs-in-use as a percentage of the number of program loads. Overlays-in-use occur only as a last resort and indicate that the program pool size is too small to support the current processing load.

**Scratch Usage:** Scratch Usage is measured through two metrics — the Percentage of Scratch Pages in Use and the Percentage of Scratch Pages Stolen. If the Percentage of Scratch Pages in Use continues to increase, it indicates that your scratch area is becoming full. The Percentage of Scratch Pages Stolen represents the number of scratch pages re-assigned from one task to another as a percentage of the number of PUT SCRATCH requests. A high percentage indicates that your scratch area is too small to support the scratch activity in your system.

### 1.6.2.9 DC Resources



The CA-IDMS Agent monitors DC resources using the percent of RLEs, RCEs and DPEs in use. Each of these are control blocks used to manage resources that are internal to a CA-IDMS system. CA-IDMS allocates a pool of these control blocks when the system starts up. In Release 14.0 and later, CA-IDMS allocates more control

blocks when the current supply is exhausted, causing a slight decrease in performance. In earlier releases, one or more tasks or the entire system can fail if the supply is exhausted.

### 1.6.2.10 Network Monitoring



The Percentage of Line I/Os in Error is the metric used to monitor the CA-IDMS teleprocessing network. If this value increases, it may indicate a hardware or software problem impairing communication with the CA-IDMS system.

## 1.7 Hardware and Software Requirements

This section describes the hardware and software requirements on the MVS platform and Windows NT platform that you need to run the CA-IDMS Agent.

### 1.7.1 For the Mainframe Platform

On the MVS platform, in addition to the basic requirements for CA-IDMS, you must have the following software installed and running:

- OS/390 Version 1, Release 3 or higher is required for CA-IDMS Release 12.01 up to 14.1, and OS/390 Version 2, Release 5 or higher for CA-IDMS Release 15.0
- OpenEdition in full function mode
- Unicenter TNG Agent Technology for OS/390 Release 2.1 or higher

### 1.7.2 For the PC Platform

On the Windows NT platform, the following hardware and software is required to run the CA-IDMS Agent:

- A Pentium 133 processor or greater
- At least 64 megabytes of memory (128 megabytes recommended)
- At least one gigabyte of hard disk storage
- Windows NT 4.0 with Service Pack 3
- Microsoft SQL Server, Version 6.5 with Service Pack 2 (required for WorldView DSM system only)
- Unicenter TNG, Release 2.1 or higher

## 1.8 Where to Find More Information

The following table summarizes where you can find related documentation files.

Document	Location
Unicenter TNG documentation in Winhelp format, including the following documents referenced in this guide: <ul style="list-style-type: none"> <li>■ <i>Unicenter TNG Concepts Guide</i></li> <li>■ <i>Unicenter TNG SDK Programmers Guide</i></li> </ul>	Unicenter TNG installation CD, Online Books folder
Unicenter TNG documentation in PDF format including the following documents referenced in this guide: <ul style="list-style-type: none"> <li>■ <i>Unicenter TNG Concepts Guide</i></li> <li>■ <i>Unicenter TNG SDK Programmers Guide</i></li> </ul>	Unicenter TNG installation CD (Release 2.2 only)
<i>Unicenter TNG Agent Technology for OS/390 Getting Started</i> (in both Winhelp and PDF formats)	<ul style="list-style-type: none"> <li>■ For Unicenter TNG Release 2.1 — Unicenter TNG Agent Technology Documentation Diskette</li> <li>■ For Unicenter TNG Release 2.2 — Unicenter TNG installation CD</li> </ul>
CA-IDMS Documentation in IBM BookManager format, including the following documents referenced in this guide: <ul style="list-style-type: none"> <li>■ <i>CA-IDMS System Operations</i></li> <li>■ <i>CA-IDMS System Generation</i></li> </ul>	CA-IDMS Documentation CD

In addition, Computer Associates provides comprehensive, context-sensitive help for all its Unicenter TNG agents. To use help, simply click a field and press F1 to receive detailed field-level help about MIB attributes and agent configuration values.

# Chapter 2. Installing and Configuring Unicenter TNG CA-IDMS Agent

---

2.1 Overview	2-3
2.2 Installing, Configuring, and Starting Agent Technology	2-4
2.3 Installing CA-IDMS Agent on the Workstation	2-5
2.4 Installing CA-IDMS Agent on the Mainframe	2-6
2.5 Modifying a CA-IDMS System to Be Monitored	2-7
2.5.1 Increasing Operating System Storage	2-7
2.5.2 Increasing CA-IDMS Region Storage	2-7
2.5.3 Modifying CA-IDMS System Startup JCL	2-7
2.5.4 Specifying SYSAGNT Option File Startup Parameters	2-8
2.5.5 Defining the AGMT Task	2-10
2.5.6 Modify the SYSTEM Statement	2-11
2.5.7 Modify External Security Privileges	2-11
2.6 Starting a CA-IDMS Agent	2-12
2.7 Controlling and Monitoring Agent Execution	2-13
2.7.1.1 AGMT Task Code Syntax	2-13
2.7.1.2 Usage Considerations	2-14
2.8 Configuring the CA-IDMS Agent	2-16
2.8.1 Specifying Agent Configuration Attributes Values	2-16
2.8.2 Setting Initial Configuration Values	2-17
2.8.3 Overriding Trap Destinations and Community Strings	2-19
2.8.4 Tailoring CA-IDMS Event Detection	2-19
2.8.4.1 The IDMSTRAP Table	2-20
2.8.4.2 #TRAP Macro	2-20



## 2.1 Overview

This chapter tells you:

- How to install the CA-IDMS Agent on the mainframe and on the Windows NT workstation
- How to start the CA-IDMS Agent and control its execution

To monitor your CA-IDMS systems through Unicenter TNG, you must first install and configure the CA-IDMS Agent in the following sequence:

1. Install, configure, and start Agent Technology on the mainframe
2. Install Unicenter TNG and the CA-IDMS Agent on a Windows NT workstation
3. Install CA-IDMS Agent on the mainframe
4. Modify the CA-IDMS system to be monitored
5. Start the CA-IDMS Agent
6. Configure the CA-IDMS Agent

The following sections describe each step in more detail.

## 2.2 Installing, Configuring, and Starting Agent Technology

Agent Technology is a common component of all mainframe agents. It executes in the OpenEdition environment and acts as a bridge between the mainframe agents and Unicenter TNG manager. Before an agent can be started, you must install, configure, and start Agent Technology on the mainframe, as described in *Unicenter TNG Agent Technology for OS/390 Getting Started*.

As part of configuring Agent Technology, you define default trap destinations and community strings. Default trap destinations specify where SNMP traps are sent unless overridden for a specific agent. Before Unicenter TNG can successfully monitor an agent, trap destinations must either be defined to Agent Technology or they must be specified as part of a configuration set used by the Agent. Initially, it is easier to use default destinations for all agents, and override them later if necessary.

Similarly, Agent Technology is automatically installed with two community strings that are the defaults for all agents. Community strings define which Unicenter TNG managers can access and update MIB information. The default community strings may be overridden for an individual agent through the use of a configuration set. We recommend that you use the defaults until you are more experienced with Unicenter TNG.

For more information on defining default trap destinations and community strings, see *Unicenter TNG Agent Technology for OS/390 Getting Started*. For information on overriding these for an agent, see Configuring the CA-IDMS Agent later in this chapter.

## 2.3 Installing CA-IDMS Agent on the Workstation

To install CA-IDMS Agent on a Windows NT workstation, you first must install Unicenter TNG. Run the install program on the CA-IDMS Agent Workstation Components diskette for Unicenter TNG 2.1. For later releases, the CA-IDMS Workstation Components are included on the Unicenter TNG CD.

Once Unicenter TNG has been installed, you must start its associated services and run auto-discovery to locate the agents that are available for monitoring. For more information on these steps, refer to the *Unicenter TNG Concepts Guide*.

Before you run auto-discovery to locate a CA-IDMS Agent you must first start Agent Technology on the mainframe. If you don't, you must run auto-discovery again. The CA-IDMS Agent will be detected automatically once it has been started.

## 2.4 Installing CA-IDMS Agent on the Mainframe

You must install the CA-IDMS Agent on the mainframe before it can be used to monitor a CA-IDMS system. For detailed installation instructions, refer to the PIB accompanying the tape.

## 2.5 Modifying a CA-IDMS System to Be Monitored

In order to activate a CA-IDMS Agent for a CA-IDMS system, you must make the following changes:

- Increase the storage returned to the operating system at CA-IDMS system startup
- Increase the storage available to the CA-IDMS region
- Add new DD statements to the startup JCL
- Define the AGMT task code and program
- Modify the system generation SYSTEM statement to collect task statistics
- Modify the external security for the CA-IDMS system

### 2.5.1 Increasing Operating System Storage

Unless the CA-IDMS system currently returns extra space to the operating system at startup, you may need to increase this amount. To do so, assemble an RHDCPARM macro specifying a larger value for the FREESTG parameter. The agent requires an additional 600K over your current specification. For more information on assembling the RHDCPARM macro, refer to *CA-IDMS System Operations* viewable on the CA-IDMS Documentation CD.

**Note:** If there is insufficient storage, one or both of the monitors will fail to start. Look in the SYSALOG and SYSALGS log files for the following message:

```
Fail to pthread create: 112
```

### 2.5.2 Increasing CA-IDMS Region Storage

You may need to increase the storage available to the CA-IDMS region by specifying a larger REGION parameter value on the EXEC card in the CA-IDMS system startup JCL. The agent requires an additional 1,200K over your existing core requirements. For more information about CA-IDMS system startup, see *CA-IDMS System Generation* viewable on the CA-IDMS Documentation CD.

**Note:** If there is insufficient storage, one or both of the monitors will fail to start. Look in the SYSALOG and SYSALGS log files for the following message:

```
Fail to pthread create: 112
```

### 2.5.3 Modifying CA-IDMS System Startup JCL

You must add additional DD statements to the CA-IDMS system startup JCL. The presence or absence of the SYSAGNT DD statement determines whether the CA-IDMS Agent will be used to monitor the CA-IDMS system and therefore whether the Agent is started automatically by the CA-IDMS system. You need additional DD statements to log status information during Agent execution, to provide Agent Technology environment information and to enable TCP/IP communications by the Agent.

**Required DD Statements:** The following DD statements are required:

```
//SYSAGNT DD *          CA-IDMS Agent Option File
/*
//SYSALOG DD SYSOUT=*   Log file for the event monitor
//SYSALGS DD SYSOUT=*   Log file for the status monitor
//ENVFILE DD DISP=SHR,DSN=<Agent-Technology-environment-file>
//SYSTCPD DD DISP=SHR,DSN=<TCP/IP-dataset-name>
```

where:

- The SYSAGNT file, is the CA-IDMS Agent Option File that contains parameters to control agent execution. The parameters are described later under Specifying SYSAGNT Option File Startup Parameters.
- The ENVFILE file is the Agent Technology environment file. It is created during Agent Technology installation, and is saved in the corresponding SRCLIB with the ENVFILE member name.
- The SYSTCPD file identifies the dataset that contains the TCP/IP control information. The dataset name is typically TCPIP.TCPIP.DATA, but can be assigned a different name when TCP/IP is installed. This DD statement is required only if the dataset name is not the default.

**Optional DD Statements:** The following DD statements are optional. They are used only if additional information is required for debugging purposes:

```
//SYSADBG DD SYSOUT=*   (Event monitor debug file)
//SYSADBS DD SYSOUT=*   (Status monitor debug file)
```

**Additional STEPLIB:** Add a DD statement for the Agent Technology load library to the STEPLIB concatenation for your system startup. This library was installed as part of the Agent Technology.

### 2.5.4 Specifying SYSAGNT Option File Startup Parameters

SYSAGNT Option File parameters let you specify startup values to the CA-IDMS Agent. The Option File is a card image file with a DDNAME of SYSAGNT. It must be included in the startup JCL for any CA-IDMS system monitored by CA-IDMS Agent.

Since there are defaults for all of the options that can be specified, you may not need to include anything in the file; however, the DD statement itself is required. The first time you start an agent, the only option that you will likely use is SYSTEM\_NAME.

Specify parameters in this file as "keyword = value" pairs. The parameters can start in any column and can contain any number of spaces between the tokens:

Parameter	Description
SYSTEM_NAME = <i>system-name</i>	Specifies the CA-IDMS system name as it will be known within Unicenter TNG. The default is the value of the DCNAME parameter if the system is a member of a Data Sharing Group, or SYSTnnnn where nnnn is the DC system version number, in the other cases.
CANCEL_TASK = YES/NO	Specifies whether the CA-IDMS Agent is permitted to cancel tasks. By specifying NO, you can prohibit anyone, including those with the ability to change MIB attributes, from canceling tasks through the CA-IDMS Agent. The default is YES.
CONFIG_SET = <i>config-set-name</i>	<p>Specifies a configuration set name, which optionally supplies initial values for configuration attributes, community strings and trap destinations. By default, the initial values for:</p> <ul style="list-style-type: none"> <li>■ Configuration variables are the values used by the Agent the last time it ran</li> <li>■ Community strings are values set by Agent Technology</li> <li>■ Trap destinations default to those established by Agent Technology</li> </ul> <p>For more information, refer to <i>Configuring the CA-IDMS Agent</i> later in this chapter.</p>
USE_CONFIG = DEFAULT/ ALWAYS	Specifies whether the specified configuration set should always be used to initialize configuration attributes or used only if no configuration information exists within the Object Store. DEFAULT is the default. For more information, refer to <i>Setting Initial Configuration Values</i> later in this chapter.

Parameter	Description
LOG_LEVEL = <i>n</i>	<p>Specifies the level of messages to be written to the Agent log files. <i>n</i> must be a number between 0 and 7; the default is 3. The meaning of the LOG_LEVEL values are:</p> <ul style="list-style-type: none"><li>■ 0 — Log fatal level messages only</li><li>■ 1 — Log critical and fatal messages only</li><li>■ 2 — Log warning, critical, and fatal messages only</li><li>■ 3 — Log information, warning, critical, and fatal messages only</li><li>■ 4 — Log debug, information, warning, critical, and fatal messages only</li><li>■ 5 — Log debug1, debug, information, warning, critical, and fatal messages only</li><li>■ 6 — Log debug2, debug1, debug, information, warning, critical, and fatal messages only</li><li>■ 7 — Log all messages</li></ul>

### 2.5.5 Defining the AGMT Task

The IDMS Agent runs the Event Monitor and the Status Monitor as two separate OS/390 subtasks. These subtasks are attached automatically when the CA-IDMS system is started. The AGMT program and task code let a system administrator display the status of these subtasks and provide the ability to stop and restart them.

Before using this task, it and its related program must be defined to the CA-IDMS system using the following system generation definitions:

```
ADD PROGRAM IDMSAGMT
    LANGUAGE ASSEMBLER
    NOPROTECT
.
ADD TASK AGMT
    INPUT
    INVOKES PROGRAM IDMSAGMT
    LOCATION ANY
.
```

For more information about the ADD PROGRAM statement and ADD TASK statement, see *CA-IDMS System Generation*, viewable on the CA-IDMS Documentation CD.

## 2.5.6 Modify the SYSTEM Statement

Make sure that the SYSTEM statement in your system generation specifies TASK COLLECT in the STATISTICS clause. This is necessary for Agent View to display the correct details in the CPU window.

## 2.5.7 Modify External Security Privileges

Modify the user ID's external security privileges that the CA-IDMS system executes under to provide the ability to sign on to OMVS as a member of AWGROUP, or whatever group Agent Technology was installed under.

**Note:** This user ID cannot be a UID 0 user.

Set the Home directory to the directory Agent Technology was installed in, typically /agent.

## 2.6 Starting a CA-IDMS Agent

Once you have made the necessary changes to the CA-IDMS system to be monitored, the CA-IDMS Agent will start automatically the next time you start the system and begin to monitor the health of your CA-IDMS system.



Syntax Parameter	Description
STOP Status monitor	Stops the Status Monitor
IMMEDIATE	Stops and detaches a monitor without waiting.
NOWAIT	Issues a request to stop a monitor without waiting for a detach to occur.
Display IDMSTRAP table	Displays the contents of the current IDMSTRAP table.
Reload IDMSTRAP table	Reloads the IDMSTRAP table, allowing changes to become effective without recycling the CA-IDMS system.

### 2.7.1.2 Usage Considerations

**Displaying Monitor Status:** When you use the DISPLAY MONITORS option, AGMT displays the status of either or both monitors. The monitors are identified by the name of their associated programs:

- IDMSAEVM — The Event Monitor
- IDMSAPRM — The Status Monitor

A monitor is displayed only if it has been attached as a subtask. A subtask can be attached, but not active. The Display Monitors function only shows attached subtasks. An *active* subtask is one that has not ended and is still running or is in a wait state. An *ended* subtask is not running or waiting, but is still attached. AGMT reports that a subtask has ended normally or abnormally based on data that OS/390 returns at subtask termination time.

Once you stop an agent using the AGMT STOP command, the Agent no longer appears on the status display. If, on the other hand, the Agent terminates for other reasons, like the termination of Agent Technology, the monitors continue to appear on the status display because their subtasks have not yet been detached.

**Monitor Interactions:** The Event Monitor must be active in order for the Status Monitor to run. If the Event Monitor has ended, you cannot start the Status Monitor. If the Status Monitor is running, you cannot stop the Event Monitor.

**Stopping Monitors:** A delay of several seconds is normal when waiting for a response from an AGMT STOP command. To stop a monitor, AGMT first sends a request to the monitor to shut itself down. When the monitor ends, AGMT detaches the subtask. If the monitor does not end within 10 seconds, AGMT stops waiting and ends itself without detaching the subtask.

Use the STOP IMMEDIATE option only when the subtask is not responding to a STOP command. The IMMEDIATE option does not wait for the monitor to end, but

detaches the monitor subtask immediately. The monitor simply stops, regardless of what it may have been doing.

Use the `STOP NOWAIT` option when you don't want to wait for the subtask to end. The `NOWAIT` option sends a request to the subtask to stop without waiting or detaching. Leaving a subtask attached causes no harm and any attached subtasks automatically detach at system shutdown. A subtask is also detached if the monitor is restarted.

**Reloading the IDMSTRAP table:** The `IDMSTRAP` table permits user-tailoring of the messages and database error statuses that results in traps being generated by the agent. `AGMT` allows the contents of this table to be viewed and changed by using the `DISPLAY` and `RELOAD IDMSTRAP TABLE` commands, respectively. There is no need to stop and restart the agent to make changes effective.

## 2.8 Configuring the CA-IDMS Agent

You can configure the CA-IDMS Agent to meet the needs of your environment. There are three basic types of information that can be specified:

- Values for such things as thresholds, polling intervals, etc.
- Trap destinations and community strings
- Events that result in SNMP traps

Thresholds, polling intervals and other such values are MIB attributes that can be examined and set through Unicenter TNG interfaces such as Agent View and MIB Browser. Initial values for these attributes may come from one of several places depending on the initialization strategy chosen for the Agent.

Trap destinations and community strings have defaults established through Agent Technology, which you can override for a specific agent through use of a configuration set specified at agent startup.

Events that result in SNMP traps are specified through the IDMSTRAP table, which can be tailored to meet the needs of a specific system.

Each of these facilities is discussed in more detail in this section.

### 2.8.1 Specifying Agent Configuration Attributes Values

You can tailor the CA-IDMS Agent to monitor CA-IDMS systems of widely different execution profiles. For example, it may be common on one system to have an extremely heavy I/O rate, while on another system, such a rate would signal an unusual event that should be investigated. You can configure the Agent to handle both systems, treating the high I/O rate on one as normal, while flagging it as a warning condition on the other.

To tailor an agent for a particular system, you set different values for configuration attributes in the CA-IDMS MIB. These are attributes to which designated workstations have read/write access.

The initial values for configuration attributes are established during agent startup based on the initialization strategy in effect for the Agent. Subsequently, the values of the configuration attributes can be changed through Unicenter TNG tools such as Agent View and MIB Browser.

When you use the tools to make changes, the changes remain in effect until the Agent is shutdown. They may also remain in effect when the Agent is restarted depending on the agent initialization strategy you chose. For more information, see Setting Initial Configuration Values later in this section.

**MIB Attributes:** There are two groups in the MIB that contain configuration attributes:

- **idmsConfigGroup** — The configuration attributes in this group are used only during CA-IDMS Agent initialization and then only if the startup control option indicates that they should be used.
- **idmsStatusGroup** — The configuration attributes in this group control the execution of the CA-IDMS Agent once it is executing. If any of these values are changed, they immediately impact the Agent. The most recent **idmsStatusGroup** information is saved in the Object Store and can be used to initialize the Agent the next time it is started.

Each of these groups contains subgroups that correspond to the resource categories monitored by the Agent. Each also has a subgroup for controlling Event Monitoring. The following table outlines the various groups that contain configuration attributes:

Category	idmsConfigGroup	idmsStatusGroup
Events	idmsCfgEvent	idmsStatEvent
CPU	idmsCfgCPU	idmsStatCPU
I/O	idmsCfgIO	idmsStatIO
Buffers	idmsCfgBuffer	idmsStatBuffer
Trans-action logs	idmsCfgTransLog	idmsStatTransLog
Locking	idmsCfgLock	idmsStatLock
Database	idmsCfgDatabase	idmsStatDatabase
Workload	idmsCfgWorkload	idmsStatWorkload
Memory	idmsCfgMemory	idmsStatMemory
DC Resources	idmsCfgDCResources	idmsStatDCResources
Network	idmsCfgNetwork	idmsStatNetwork

For more information on the types of configuration attributes that the CA-IDMS Agent supports, see What the Agent Monitors in Chapter 1. For more information about specific attributes and their meanings, refer to the Agent View help or the MIB listing viewable through the MIB Browser.

## 2.8.2 Setting Initial Configuration Values

When the CA-IDMS Agent begins executing, it establishes the initial values for the configuration attributes that control its subsequent execution. The potential sources of these values are:

- Those retained from the previous agent execution
- Corresponding attributes in the **idmsConfigGroup** in the MIB

- A configuration set
- Defaults established by the CA-IDMS Agent

Which source is used is determined both by options specified through the SYSAGNT Option File and on current attribute values in the idmsCfgEvent group in Object Store. If not directed otherwise, the CA-IDMS Agent will retain the values from its previous execution.

When the Agent starts, it first looks in the SYSAGNT Option File to see if a configuration set has been named and if so, what the USE\_CONFIG option indicates. If no configuration set has been specified, or if the USE\_CONFIG option specifies DEFAULT, then the Agent will look at attributes in the idmsCfgEvent group to determine how to initialize the configuration attributes. If, however, the USE\_CONFIG option indicates ALWAYS, all configuration attribute values will be retrieved from the named configuration set. If it doesn't exist, the attribute values will be set to defaults established by the CA-IDMS Agent.

If no configuration set was named or if the USE\_CONFIG option indicates DEFAULT, the CA-IDMS Agent will retrieve the idmsCfgEvtGroup from Object Store and determine how to proceed based on the value of the idmsCfgEvtStartupControl attribute.

Value of idmsCfgEvtStartupControl	Description
1	Retain the values as they were when the Agent shutdown
2	Reset the values to those in the idmsConfigGroup
3	Reset the values to those in the configuration set named in idmsCfgEvtConfigSet. If no configuration set name is supplied, use the one specified in the Option File.

If the Agent attempts to use a configuration set for initialization purposes and it isn't found, or if no configuration set name was provided and no configuration information is present in Object Store, the values of the configuration attributes will be set to defaults established by the CA-IDMS agent. These defaults are documented in the descriptions for the associated MIB attributes.

For more information on defining configuration sets and loading them into Object Store, refer to the *Unicenter TNG SDK Programmers Guide* and the *Unicenter TNG Agent Technology for OS/390 Getting Started*.

### 2.8.3 Overriding Trap Destinations and Community Strings

Trap destinations determine where SNMP traps are sent. Community strings restrict access to MIB information to designated workstations.

Agent Technology establishes defaults for trap destinations and community strings, but these can be overridden for a specific agent by including the overrides in a configuration set. If you name a configuration set in the SYSAGNT Option File, CA-IDMS agent will load the configuration set at startup. Any trap destinations or community strings that were specified in the configuration set will override the system-wide defaults.

For more information on defining configuration sets and loading them into Object Store, refer to the *Unicenter TNG SDK Programmers Guide* and the *Unicenter TNG Agent Technology for OS/390 Getting Started*. Refer to the same manuals for information on trap destinations and community strings.

### 2.8.4 Tailoring CA-IDMS Event Detection

**For DC Messages:** The Event Monitor sends SNMP trap messages when it detects an important event happening within a CA-IDMS/DC system. Most of these traps are triggered by inspecting DC log messages and determining if they are important enough to cause an SNMP trap to be sent to the CA-IDMS Unicenter manager.

The following table identifies DC message severity levels and their associated default SNMP trap type:

DC Message Severity Level	SNMP Trap Type
0, 1, 2	None
3, 4, and 5	Warning
6, 7, 8	Critical

If you want to narrow the selection of DC messages that trigger SNMP traps, you can filter the selection using the IDMSTRAP table. You can also use the IDMSTRAP table to generate SNMP traps for internal DC messages that have severity levels 0, 1, or 2; some of these messages are important and should generate an SNMP trap.

**For DB Status Codes:** SNMP traps can also be triggered by IDMS status codes if you add the status codes to the IDMSTRAP table.

### 2.8.4.1 The IDMSTRAP Table

The IDMSTRAP table is generated by the IDMSTRAP assembler program and #TRAP macro, both of which are distributed in source with the CA-IDMS Agent. The distributed IDMSTRAP table contains entries for all potentially serious event messages that have a severity less than 3 and therefore would not otherwise generate a trap. You can configure this table by:

- Adding new entries for your own DC messages.
- Removing entries so that a trap is not generated for a message with a severity less than 3.
- Modifying the severity of the generated trap by changing the trap id associated with a message.
- Preventing traps for messages with severity level 3 or higher
- Overriding the default trap severity for messages with severity level 3 or higher

The following are the TRAPIDs that are supported by the #TRAP macro.

Trap ID	Description
TRAPID=0000	Do not send an SNMP trap.
TRAPID=8000	Send an SNMP informational trap.
TRAPID=8001	Send an SNMP warning trap.
TRAPID=8002	Send an SNMP critical trap.

### 2.8.4.2 #TRAP Macro

The #TRAP macro is coded for each entry in the IDMSTRAP table. The #TRAP parameters are MSG=xxxxxxx,TRAPID=nnnn where xxxxxxxx is either the actual DC message number, or an IDMS status code. IDMS status codes can be entered either as a 4 digit major/minor code prefixed with 'STAT' or as a 2 digit minor code prefixed with 'STATUS'. For example IDMS status 0970 would be coded as STAT0970, and any IDMS status code with a minor code of 11 would be coded as STATUS11.

Here is an example of an IDMSTRAP table:

```
TITLE 'IDMSTRAP - CA-IDMS Unicenter TRAP table filter'
* IDMSTRAP NORENT EP=TRAPEP1
#TRAP MSG=STATUS11,TRAPID=8002 AREA full
#TRAP MSG=STATUS60,TRAPID=8002 AREA corruption
#TRAP MSG=STAT0970,TRAPID=8001 AREA will not READY
#TRAP MSG=STATUS71,TRAPID=8001 Definition error
#TRAP MSG=DC200007,TRAPID=8002 3005 - Area not available
#TRAP MSG=DC205006,TRAPID=0000 Don't TRAP this message
#TRAP MSG=DC205007,TRAPID=8002 READ ERROR
#TRAP MSG=DC205008,TRAPID=8002 WRITE ERROR
#TRAP , This null #TRAP must be last
END TRAPEP1
```

**Note:** The IDMSTRAP table must end with a #TRAP macro that has no parameters.



# Chapter 3. Using Unicenter TNG with the CA-IDMS Agent

---

3.1	Accessing Information About CA-IDMS Resources	3-3
3.2	Viewing CA-IDMS Resources from Node View	3-5
3.2.1	Understanding Node View Resource Colors	3-5
3.2.2	Displaying and Hiding Nodes	3-5
3.2.3	Displaying and Hiding Mini Trees	3-6
3.2.4	Using the Node View Toolbar	3-6
3.2.5	Viewing Information About a Particular CA-IDMS Resource	3-7
3.3	Viewing CA-IDMS Resources from Agent View	3-9
3.3.1	Using the Agent View Toolbar	3-9
3.3.2	Getting More Detailed Information	3-10
3.4	Browsing the CA-IDMS MIB	3-12
3.5	Obtaining Event History Information for a Resource	3-13
3.6	Integration with Unicenter Event Management	3-14



## 3.1 Accessing Information About CA-IDMS Resources

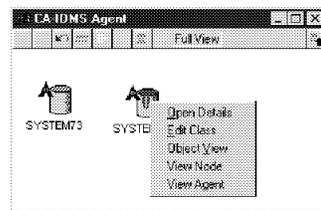


Unicenter TNG accesses information about CA-IDMS resources via the CA-IDMS Agent. To view this information from Unicenter TNG, follow the steps provided below:

- Open Unicenter TNG WorldView and select 2-D map. The Managed Objects window appears.
- Drill-down to locate a CA-IDMS system icon that represents the CA-IDMS system being monitored.



The CA-IDMS icon reflects the state of the CA-IDMS resources being monitored, as well as the state of the CA-IDMS Agent. After you view the general condition of all CA-IDMS subsystems, you can determine the reason for the status of the host icon. To obtain more information about the status of the host icon, right-click. A pop-up menu appears, with these menu options:



When you right-click a CA-IDMS system icon, you can access:

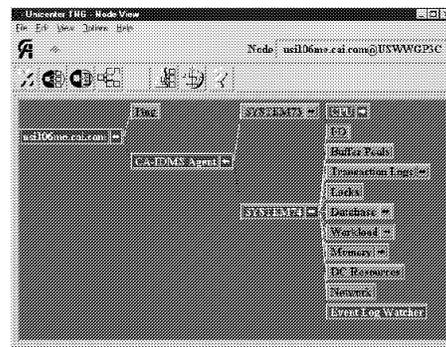
- Object View, which displays the MIB in an Explorer-type interface.
- Node View, which displays the status of all the CA-IDMS resources being reported on by this CA-IDMS Agent. More information about Node View is provided in the next section.
- Agent View, which displays the status and graphical details of all the CA-IDMS resources being reported on by this CA-IDMS Agent. More information about the CA-IDMS Agent implementation of Agent View can be found in the CA-IDMS Agent View online help file and later in this chapter.

You can also click **Open Details** to access the **Managed Object Notebook** for the CA-IDMS managed object, which allows you to view and modify information about the CA-IDMS managed object. The **Edit Class** menu option opens the **Unicenter TNG Class Wizard**, which lets you edit the attributes of the CA-IDMS class of managed objects.

For more information about the **Managed Object Notebook**, the **Unicenter TNG Class Wizard**, and **Object Viewer**, see the **Unicenter TNG documentation**.

## 3.2 Viewing CA-IDMS Resources from Node View

To view information about CA-IDMS resources from Node View, right-click the CA-IDMS system icon and select View Node from the pop-up menu. Node View displays the CA-IDMS system icon, as well as a tree of all the CA-IDMS resources being monitored.



### 3.2.1 Understanding Node View Resource Colors

When you navigate to the CA-IDMS system resource, note its color. Its color reflects the health of the CA-IDMS resources being monitored. The meanings for the colors that the CA-IDMS system resources can reflect are:

Icon Color	Meaning
Red	Indicates a critical or warning resource state.
Orange	Indicates a critical or warning resource state that has been acknowledged.
Green	Indicates a normal resource state.
Lime Green	Indicates a repaired resource state.
Blue	Indicates an unknown resource state.

To access information about a resource, right-click the resource icon. A pop-up menu appears, which contains options you can select to view more information. Each option is discussed in detail later in this chapter.

### 3.2.2 Displaying and Hiding Nodes

The Node View screen displays nodes representing the host, and all discovered agents running on the selected machine. Subnodes for the CA-IDMS Agent are displayed showing the status of all monitored CA-IDMS resources. For more information on these nodes, refer to Accessing Information about CA-IDMS Resources later in this chapter. You can collapse all displayed resource nodes by clicking on the left arrow on the CA-IDMS node, which contains the id of the CA-IDMS system being

monitored. Once you collapse a set of subnodes, a right arrow appears on the node into which they have been collapsed. To redisplay all collapsed nodes, click the right arrow.

### 3.2.3 Displaying and Hiding Mini Trees

The Display/Hide Mini Tree option can be used to more readily identify problem resources (that is, those that are yellow or red). Click a resource icon on the mini-tree to align that icon with its corresponding resource on the main window. By aligning the mini-tree and resource column in this way, you can readily select more information about a resource by using the pop-up menu available from the resource column.

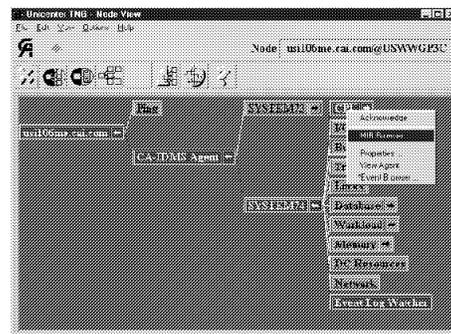
### 3.2.4 Using the Node View Toolbar

The Node View screen contains a toolbar that can be used to modify and populate the viewable objects on the screen. The following table outlines how to use the toolbar's options as they occur from left to right.

Toolbar Option	Purpose
	Exits Node View and returns to the previous display.
	Displays or hides a mini-tree of all agent resources.
	Displays a panel that describes the Node View object.
	Redisplays all nodes, if you have previously collapsed the mini-tree of resource objects.
	Displays all agent nodes if you have previously collapsed these objects in the display.
	Opens Unicenter TNG DSM View, which is discussed in detail in the Unicenter TNG documentation.
	Rediscovered all displayable agent resources. If you think that information is missing from the Node View display, click this icon to communicate with the agents running on the host, and refresh the information. If your map contains a large number of resource objects, rediscovering all resource objects may take some time due to the volume of information transported between mainframe and workstation across the TCP/IP link.
	Provides you with information about Node View.

### 3.2.5 Viewing Information About a Particular CA-IDMS Resource

You can view detailed information about each resource displayed on the Node View map. If you select a resource, and then right-click, a pop-up menu appears:

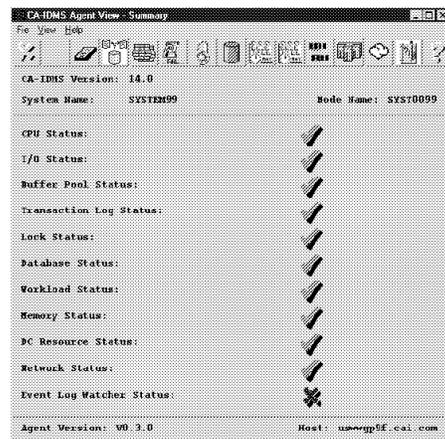


You can select the following options from this pop-up menu:

<b>Option</b>	<b>Description</b>
Acknowledge	If a particular node is in an abnormal state, acknowledges that you are aware of the problem and changes the state to an acknowledged state.
MIB Browser	Allows you to browse all information contained in the CA-IDMS MIB.
Properties	Opens Unicenter TNG DSM View, which shows all objects managed by this particular Unicenter TNG workstation.
View Agent	Opens Agent View, which displays the status and graphical details of all the CA-IDMS resources being reported on by this CA-IDMS Agent.
Event Browser	Opens the Event Browser, which displays a history of all the state changes for a particular resource. You can sort the historical output and filter the information displayed. The Reason field in the Filter option displays messages indicating the cause of the resource state change.

## 3.3 Viewing CA-IDMS Resources from Agent View

Agent View displays the status of CA-IDMS system resources in a graphical display, letting you quickly determine problem areas. To open Agent View, chose the View Agent menu option that appears when you right-click the CA-IDMS system icon in WorldView or in Node View.



The meaning of the Agent View status icons is:

Icon	Description
	The resource status is critical.
	The resource status is normal.
	The resource status is warning.
	The resource status is not known.

### 3.3.1 Using the Agent View Toolbar

To display information about different resource categories for a CA-IDMS system, click a toolbar button. The toolbar options are described below:

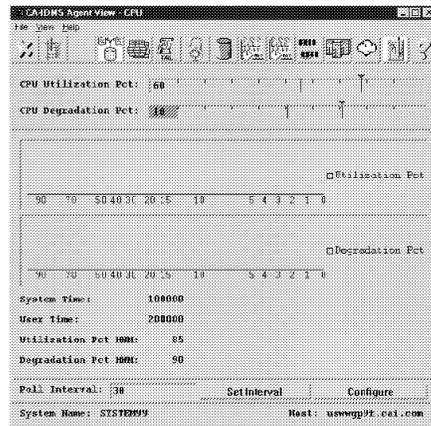
Toolbar Option	Purpose
	Exits Agent View.
	Displays the Agent View Summary screen
	Displays details of the CPU utilization and degradation.
	Displays details of the I/O rate.
	Displays details of the buffer pool status, such as the percentage of buffer misses and forced writes.
	Displays details of the transaction log and journal file status.
	Displays details of lock storage overflows.
	Displays details of the database-related status information, such as index splits and spawns, record fragmentation or overflow, and SQL auto recompiles.
	Displays details of task abends, transaction rollbacks, and system run unit overflows.
	Displays active tasks.
	Displays details of the storage pool usage, program pool usage, and scratch usage.
	Displays details of the DC resource status.
	Displays details of the line I/O status for the CA-IDMS teleprocessing network.
	Displays details of the Event Log Watcher status.

### 3.3.2 Getting More Detailed Information

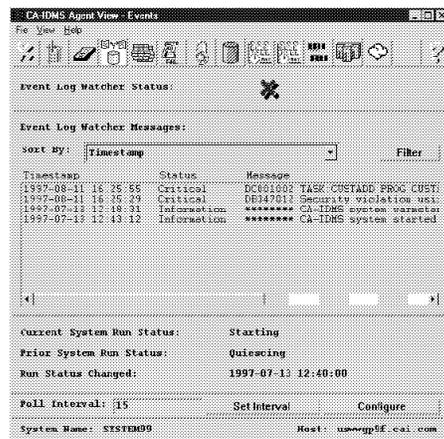
To get more detailed information about a resource, click the toolbar button for the resource. The following screen shot shows an example of the graphical display that Agent View provides.



In this screen, the CPU Utilization Percentage has reached the warning threshold, which is indicated by the graph bar color (yellow) and the graph itself:



This screen shot shows the messages produced by the Event Log Watcher. You can sort the event messages in different ways — for example, by status or timestamp.

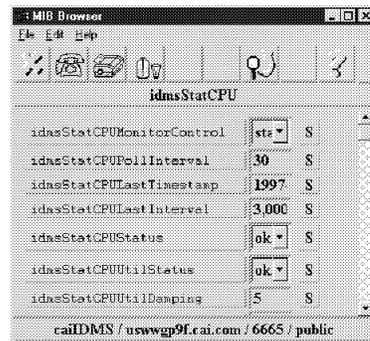


## 3.4 Browsing the CA-IDMS MIB

The MIB Browser allows you to browse all information contained in the CA-IDMS MIB. To access the MIB Browser, select it when you right-click a CA-IDMS system resource in Node View. The caiIDMS MIB appears, organized by its three main groups — the configuration group, status group, and polling group:



Keep clicking the right-facing arrow next to a group category to display more detailed information about the MIB attributes:

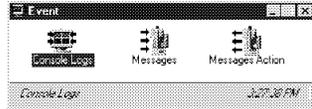




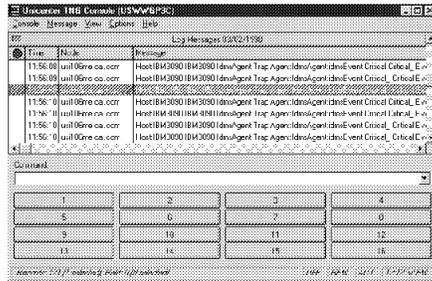
## 3.6 Integration with Unicenter Event Management

The CA-IDMS Agent sends messages in a standard format to the Unicenter Enterprise Manager Event Console Log.

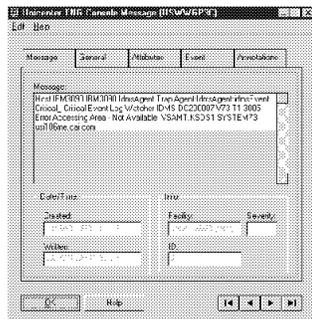
Messages and actions can be defined using the full power of Unicenter Event Management.



The Unicenter Event Console Log displays CA-IDMS DC log information:



Detailed message text from the CA-IDMS Agent Event Monitor displays in the Unicenter Event Console Log:



# Appendix A. Enterprise Specific Traps for the CA-IDMS Agent

---

- A.1 Overview . . . . . A-3
- A.2 Status Monitor Traps . . . . . A-4
  - A.2.1 CPU-Related Traps . . . . . A-4
  - A.2.2 I/O-Related Traps . . . . . A-5
  - A.2.3 Network-Related Traps . . . . . A-5
  - A.2.4 Buffer-Related Traps . . . . . A-5
  - A.2.5 Log/Journal-Related Traps . . . . . A-5
  - A.2.6 Locking-Related Traps . . . . . A-6
  - A.2.7 Database-Related Traps . . . . . A-6
  - A.2.8 Workload-Related Traps . . . . . A-7
  - A.2.9 Memory-Related Traps . . . . . A-8
  - A.2.10 DC Resource-Related Traps . . . . . A-8
- A.3 Event Monitor Traps . . . . . A-9



## A.1 Overview

This appendix identifies the traps that can be sent by the CA-IDMS Agent. There are two general types of traps that can be sent:

- One type is sent by the Status monitor when a condition arises that causes the status of one of the resource categories to change.
- The other type is sent by the Event Monitor when it detects an event for which a trap is required.

## A.2 Status Monitor Traps

The Status Monitor issues a trap when it changes the status of a resource category. The trap informs Unicenter TNG that a change has occurred in the state of the CA-IDMS system.

Each status group within the MIB contains one or more status attributes. The Status Monitor changes the value of one of these attributes because it detects a condition (such as a threshold exceeded) which warrants a new status. When it changes the value of the status attribute, it also sends a trap. Each status attribute has four associated traps, one for each of the four possible values that can be assigned to the status attribute.

Whenever the Status Monitor sends a trap, the following attributes are included in the VARBIND list associated with the trap:

- The status attribute whose value is being changed (the new value is sent)
- A text string, indicating what status attribute was changed and why. The corresponding MIB attribute is `idmsStatEvtMsg` in group `idmsStatEvent`.

The following tables list the traps that can be sent by the Status Monitor, the status attribute with which they are associated and the value to which the status is being changed:

### A.2.1 CPU-Related Traps

Trap ID	Trap Name	Status Attribute	Attribute Value
1	<code>idmsCPUDegrUnknown</code>	<code>idmsStatCPUDegrStatus</code>	1 - Unknown
2	<code>idmsCPUDegrOK</code>	<code>idmsStatCPUDegrStatus</code>	2 - OK
3	<code>idmsCPUDegrWarning</code>	<code>idmsStatCPUDegrStatus</code>	3 - Warning
4	<code>idmsCPUDegrCritical</code>	<code>idmsStatCPUDegrStatus</code>	4 - Critical
5	<code>idmsCPUUtilUnknown</code>	<code>idmsStatCPUUtilStatus</code>	1 - Unknown
6	<code>idmsCPUUtilOK</code>	<code>idmsStatCPUUtilStatus</code>	2 - OK
7	<code>idmsCPUUtilWarning</code>	<code>idmsStatCPUUtilStatus</code>	3 - Warning
8	<code>idmsCPUUtilCritical</code>	<code>idmsStatCPUUtilStatus</code>	4 - Critical

## A.2.2 I/O-Related Traps

Trap ID	Trap Name	Status Attribute	Attribute Value
9	idmsIOUnknown	idmsStatIOStatus	1 - Unknown
10	idmsIOOK	idmsStatIOStatus	2 - OK
11	idmsIOWarning	idmsStatIOStatus	3 - Warning
12	idmsIOCritical	idmsStatIOStatus	4 - Critical

## A.2.3 Network-Related Traps

Trap ID	Trap Name	Status Attribute	Attribute Value
13	idmsNetworkUnknown	idmsStatNetworkStatus	1 - Unknown
14	idmsNetworkOK	idmsStatNetworkStatus	2 - OK
15	idmsNetworkWarning	idmsStatNetworkStatus	3 - Warning
16	idmsNetworkCritical	idmsStatNetworkStatus	4 - Critical

## A.2.4 Buffer-Related Traps

Trap ID	Trap Name	Status Attribute	Attribute Value
17	IdmsBufferUnknown	idmsStatBufStatus	1 - Unknown
18	idmsBufferOK	idmsStatBufStatus	2 - OK
19	idmsBufferWarning	idmsStatBufStatus	3 - Warning
20	idmsBufferCritical	idmsStatBufStatus	4 - Critical

## A.2.5 Log/Journal-Related Traps

<b>Trap ID</b>	<b>Trap Name</b>	<b>Status Attribute</b>	<b>Attribute Value</b>
21	idmsLogUnknown	idmsStatLogStatus	1 - Unknown
22	idmsLogOK	idmsStatLogStatus	2 - OK
23	idmsLogWarning	idmsStatLogStatus	3 - Warning
24	idmsLogCritical	idmsStatLogStatus	4 - Critical
25	idmsJnlUnknown	idmsStatJnlStatus	1 - Unknown
26	idmsJnlOK	idmsStatJnlStatus	2 - OK
27	idmsJnlWarning	idmsStatJnlStatus	3 - Warning
28	idmsJnlCritical	idmsStatJnlStatus	4 - Critical

### A.2.6 Locking-Related Traps

<b>Trap ID</b>	<b>Trap Name</b>	<b>Status Attribute</b>	<b>Attribute Value</b>
29	idmsLockUnknown	idmsStatLockStatus	1 - Unknown
30	idmsLockOK	idmsStatLockStatus	2 - OK
31	idmsLockWarning	idmsStatLockStatus	3 - Warning
32	idmsLockCritical	idmsStatLockStatus	4 - Critical

### A.2.7 Database-Related Traps

<b>Trap ID</b>	<b>Trap Name</b>	<b>Status Attribute</b>	<b>Attribute Value</b>
33	idmsSpaceUnknown	idmsStatDBSpaceStatus	1 - Unknown
34	idmsSpaceOK	idmsStatDBSpaceStatus	2 - OK
35	idmsSpaceWarning	idmsStatDBSpaceStatus	3 - Warning
36	idmsSpaceCritical	idmsStatDBSpaceStatus	4 - Critical
37	idmsIndexUnknown	idmsStatIndexStatus	1 - Unknown
38	idmsIndexOK	idmsStatIndexStatus	2 - OK
39	idmsIndexWarning	idmsStatIndexStatus	3 - Warning
40	idmsIndexCritical	idmsStatIndexStatus	4 - Critical
41	idmsSQLUnknown	idmsStatSQLStatus	1 - Unknown
42	idmsSQLOK	idmsStatSQLStatus	2 - OK
43	idmsSQLWarning	idmsStatSQLStatus	3 - Warning
44	idmsSQLCritical	idmsStatSQLStatus	4 - Critical

### A.2.8 Workload-Related Traps

<b>Trap ID</b>	<b>Trap Name</b>	<b>Status Attribute</b>	<b>Attribute Value</b>
45	idmsTaskUnknown	idmsStatTasksStatus	1 - Unknown
46	idmsTaskOK	idmsStatTasksStatus	2 - OK
47	idmsTaskWarning	idmsStatTasksStatus	3 - Warning
48	idmsTaskCritical	idmsStatTasksStatus	4 - Critical
49	idmsTransUnknown	idmsStatTransStatus	1 - Unknown
50	idmsTransOK	idmsStatTransStatus	2 - OK
51	idmsTransWarning	idmsStatTransStatus	3 - Warning
52	idmsTransCritical	idmsStatTransStatus	4 - Critical
53	idmsSystemRUUnknown	idmsStatSysRUStatus	1 - Unknown
54	idmsSystemRUOK	idmsStatSysRUStatus	2 - OK
55	idmsSystemRUWarning	idmsStatSysRUStatus	3 - Warning
56	idmsSystemRUCritical	idmsStatSysRUStatus	4 - Critical

### A.2.9 Memory-Related Traps

Trap ID	Trap Name	Status Attribute	Attribute Value
57	idmsStgPoolUnknown	idmsStatStgPoolStatus	1 - Unknown
58	idmsStgPoolOK	idmsStatStgPoolStatus	2 - OK
59	idmsStgPoolWarning	idmsStatStgPoolStatus	3 - Warning
60	idmsStgPoolCritical	idmsStatStgPoolStatus	4 - Critical
61	idmsPgmPoolUnknown	idmsStatPgmPoolStatus	1 - Unknown
62	idmsPgmPoolOK	idmsStatPgmPoolStatus	2 - OK
63	idmsPgmPoolWarning	idmsStatPgmPoolStatus	3 - Warning
64	idmsPgmPoolCritical	idmsStatPgmPoolStatus	4 - Critical
65	idmsScratchUnknown	idmsStatScratchStatus	1 - Unknown
66	idmsScratchOK	idmsStatScratchStatus	2 - OK
67	idmsScratchWarning	idmsStatScratchStatus	3 - Warning
68	idmsScratchCritical	idmsStatScratchStatus	4 - Critical

### A.2.10 DC Resource-Related Traps

Trap ID	Trap Name	Status Attribute	Attribute Value
69	idmsDCResourceUnknown	idmsStatDCResourceStatus	1 - Unknown
70	idmsDCResourceOK	idmsStatDCResourceStatus	2 - OK
71	idmsDCResourceWarning	idmsStatDCResourceStatus	3 - Warning
72	idmsDCResourceCritical	idmsStatDCResourceStatus	4 - Critical

## A.3 Event Monitor Traps

The Event Monitor issues a trap when an event occurs, such as a message being written to the CA-IDMS log file. The Event Monitor sends only one of three possible traps, one for each of the possible severity levels with which the event is associated — OK, Warning, Critical.

Whenever the Event Monitor sends a trap, the CA-IDMS Agent sends a text string describing the event which occurred as the VARBIND list associated with the trap. The MIB attribute associated with the text string is `idmsStatEvtMsg` in group `idmsStatEvent`.

The following table lists the traps that can be sent by the Event Monitor and their corresponding severity.

Trap ID	Trap Name	Severity
8000	<code>idmsEventInfo</code>	Informational
8001	<code>idmsEventWarning</code>	Warning
8002	<code>idmsEventCritical</code>	Critical



# Appendix B. CA-IDMS Agent Messages

---

B.1 IDMSACTL Messages	.....	B-3
B.2 IDMSAG00 Messages	.....	B-6



## B.1 IDMSACTL Messages

### **AG001001 Starting IDMSAEVM Event Monitor**

**Reason:** Indicates the IDMSAEVM Event Monitor is starting.

**Module:** IDMSACTL

**Severity:** 0

### **AG001002 Starting IDMSAPRM Status Monitor**

**Reason:** Indicates the IDMSAPRM Status Monitor is starting.

**Module:** IDMSACTL

**Severity:** 0

### **AG001003 Shutdown the IDMSAEVM Event Monitor**

**Reason:** Indicates the IDMSAEVM Event Monitor is shutting down.

**Module:** IDMSACTL

**Severity:** 0

### **AG001004 Shutdown the IDMSAPRM Status Monitor**

**Reason:** Indicates the IDMSAPRM Status Monitor is shutting down.

**Module:** IDMSACTL

**Severity:** 0

### **AG001005 IDMSAEVM Event Monitor has started**

**Reason:** Indicates the IDMSAEVM Event Monitor has started.

**Module:** IDMSEVNT

**Severity:** 0

### **AG001006 IDMSAPRM Status Monitor has started**

**Reason:** Indicates the IDMSAPRM Status Monitor has started.

**Module:** IDMSPERF

**Severity:** 0

**AG001007 IDMSACTL called at normal shutdown**

**Reason:** Indicates IDMSACTL called at normal shutdown.

**Module:** IDMSACTL

**Severity:** 0

**AG001008 IDMSACTL called to stop IDMSAEVM monitor**

**Reason:** Indicates IDMSACTL called to stop the IDMSAEVM monitor.

**Module:** IDMSACTL

**Severity:** 0

**AG001009 IDMSACTL called to stop IDMSAPRM monitor**

**Reason:** Indicates IDMSACTL called to stop the IDMSAPRM monitor.

**Module:** IDMSACTL

**Severity:** 0

**AG001020 Invalid call to the IDMSACTL module**

**Reason:** Invalid call to the IDMSACTL module. This is an internal error.

**Module:** IDMSACTL

**Severity:** 0

**AG001021 Error LOADING the IDMSATRP module**

**Reason:** Could not LOAD the IDMSATRP module.

**Module:** IDMSACTL

**Severity:** 0

**AG001022 Error LOADING the IDMSTRAP module**

**Reason:** Could not LOAD the IDMSTRAP module.

**Module:** IDMSACTL

**Severity:** 0

**AG001023 Error ATTACHING the IDMSAEVM Monitor**

**Reason:** Could not ATTACH the IDMSAEVM Event Monitor.

**Module:** IDMSACTL

**Severity:** 0

**AG001024 Error ATTACHING the IDMSAPRM Monitor**

**Reason:** Could not ATTACH the IDMSAPRM Status Monitor.

**Module:** IDMSACTL

**Severity:** 0

**AG001025 IDMSACTL called at abend shutdown**

**Reason:** Indicates IDMSACTL called at abend shutdown.

**Module:** IDMSACTL

**Severity:** 0

## B.2 IDMSAG00 Messages

### **AG002001 Attach wait error**

**Reason:** WAIT call returns non-zero return code.

**Module:** IDMSAG00

**Severity:** 0

### **AG002002 No available SCA**

**Reason:** An attach of an agent subtask failed because there was no available SCA control block for the new task.

**Module:** IDMSAG00

**Severity:** 0

### **AG002003 Subtask already attached**

**Reason:** A new attach of an agent subtask failed because the subtask was already attached.

**Module:** IDMSAG00

**Severity:** 0

### **AG002004 IDENTIFY for <entry-label> failed RC=<return-code>**

**Reason:** An IDENTIFY call failed with the specified return code for the named entry point label.

**Module:** IDMSAG00

**Severity:** 0

### **AG002005 AG002005 SET\_DUB\_DEFAULT failed RC=<return-code> REASON=<reason-code>**

**Reason:** An open edition SET\_DUB\_DEFAULT called failed with the specified return code and reason code.

**Module:** IDMSAG00

**Severity:** 0

### **AG002006 DUB default set to subtask process**

**Reason:** An informational message issued when the SET\_DUB\_DEFAULT command is successfully issued. This should only occur once during the life of a DC system.

**Module:** IDMSAG00

**Severity:** 0

**AG002007 LOAD of <module-name> failed. RC=<return-code>  
ABCODE=<abend-code>**

**Reason:** A LOAD call failed for the named module with the specified return and abend codes. An abend code of 00000806 means the module was not found in STEPLIB.

**Module:** IDMSAG00

**Severity:** 0

**AG002008 GETMAIN FAILED SIZE=<size-of-request> RC=<return-code>**

**Reason:** A GETMAIN call failed. The size requested and return code are displayed. Usually this means there is insufficient available storage in the region to satisfy the request.

**Module:** IDMSAG00

**Severity:** 0

**AG002009 LOAD of <module-name> failed. RC=<return-code>**

**Reason:** A DC OS00 LOAD request failed. The module name and return code are displayed.

**Module:** IDMSAG00

**Severity:** 0

**AG002010 Arrived in AG00 subtask**

**Reason:** An informational message indicating when control arrives at the entry point of an agent subtask. Currently this code is not used.

**Module:** IDMSAG00

**Severity:** 0

**AG002011 Subtask <subtask-name> has terminated <how>.  
RC=<TCB-completion-code>**

**Reason:** An informational message stating the named subtask has ended and how.

The TCB completion code is a 4-byte hex value. If the high byte is zero, the subtask is considered to have ended normally. If non-zero, it is considered abnormal. The next 12-bits are the system completion code and the last 12-bits are the user completion code. See the TCBCMP field in the TCB DSECT.

**Module:** IDMSAG00

**Severity:** 0

**AG002012 Unknown subtask terminated**

**Reason:** The subtask termination exit was called but it was unable to find an SCA for the task and therefore was unable to identify it.

**Module:** IDMSAG00

**Severity:** 0

**AG002013 Called function** *<name>* **parm** *<name>* **TCB=***<hex-addr>*  
**RC=***<return-code>*

**Reason:** An informational message issued when certain functions are called, what the parameter was, what the return code was, and on what TCB it was called from.

**Module:** IDMSAG00

**Severity:** 0

**AG002014 Current TCB=***<hex-address>*

**Reason:** An informational message stating the address of the current TCB of the code that issues the message.

**Module:** IDMSAG00

**Severity:** 0

**AG002015 CV not authorized to start agent services**

**Reason** Indicates that the CV is not authorized to start agent services.

**Module:** IDMSAG00

**Severity:** 0

**AG002016 Status monitor waiting on event monitor to start**

**Reason** Indicates that the status monitor is waiting for the event monitor to start. The event monitor must be running for the status monitor to run. If the status monitor needs to wait for the event monitor this message is issued.

**Module:** IDMSAG00

**Severity:** 0

**AG002017 Status monitor wait complete, starting initialization**

**Reason** Indicates that the status monitor wait is complete and initialization is starting. If the status monitor was waiting on the event monitor, this message will be issued when the wait is completed.

**Module:** IDMSAG00 :

**Severity:** 0

**AG002018 Entering AG00 INIT code - <program-name>**

**Reason** An informational message indicating when an agent subtask's initialization code is entered.

**Module:** IDMSAG00

**Severity:** 0



