Open Software Foundation
Request For Comments: 66.0
January 1996

G. Wilson (IBM)
R. Mandava (IBM)

DCE SNMP SUPPORT — FUNCTIONAL SPECIFICATION

1. INTRODUCTION

The Distributed Computing Environment (DCE) provides a collection of services to manage distributed computing resources. It is a resource manager similar to an operating system. It needs to be manageable just like other resources or resource managers do in a distributed computing systems environment.

Because of the heterogeneous nature of distributed systems environment, to provide integrated, open, and industry-standard mechanisms of system management for DCE is important to achieve interoperability in managing resources and resource managers.

The Simple Network Management Protocol Version 1 (SNMPv1) has been widely used in the Transmission Control Protocol/Internet Protocol (TCP/IP) environment for network management. A distributed computing environment runs on a network. Due to the proliferation of SNMP products in the marketplace, they can also be used for monitoring of other parts of the system in a distributed computing environment. There are significant number of SNMP managers existing in customer environments that can be used to monitor DCE resources and components without introducing a lot of overhead of management framework and requiring new application programs. Therefore, using SNMP for DCE is a natural extension of network management for distributed systems management. This is a first step in satisfying the requirement to provide integrated network and distributed system management. It is recognized that further steps are necessary to fully satisfy that requirement.

The SNMP support in DCE environment consists of two parts: DCE Management Information Base (MIB) and DCE Subagent.

This document defines a portion of the DCE managed objects for use with SNMP. In particular, it describes managed objects used for managing DCE resources and services. The DCE Subagent is the implementation to support the DCE SNMP MIB.

The definition of DCE performance characteristics can be found in [RFC 32] and [RFC 33]. When these two RFCs are finalized and implemented by vendors, this DCE SNMP MIB should contain those performance attributes.

1.1. Changes Since Last Publication

None.

2. TARGET

The DCE SNMP Support provides manageability of the DCE environment through SNMP. It can be used by system administrators and by system management application programmers.

System administrators can use SNMP managers to monitor the health condition of DCE environment.

Application programmers of SNMP managers can query the values of DCE MIB variables, and respond, as needed, to DCE traps.

3. GOALS AND NON-GOALS

The goals of providing DCE SNMP Support include:

(a) To provide integrated network and distributed system management capabilities.

(b) To leverage the existing SNMP management application base.

(c) To provide light-weight management framework for monitoring DCE resources.

(d) To define and standardize important characteristics of DCE core services across all DCE providers to assure consistent management in a heterogeneous network.

(e) To specify the implementation of the DCE Subagent to support DCE SNMP MIB.

Although DCE provides cell-wide views due to its location-transparency, the DCE SNMP MIB only provides views on a per node basis.

Since SNMPv2 is only at the draft-standard level, this RFC defines DCE MIB in SNMPv1 Structure of Management Information (SMI). The DCE SNMP subagent supports SNMPv1 only.

Since SNMPv1 does not provide stringent security, there is limited use of the SNMP ''set'' function. The MIB variables that can be set are the configurable parameters to control the DCE Subagent, not to affect the DCE services.

The MIB definition applies to OSF DCE 1.1.

4. TERMINOLOGY

(a) **SNMP Manager** — The network management application program that allows the network or system administrator to manage resources in the network environment through SNMP.

(b) **SNMP Agent** — A process on the managed node that receives, processes, and responds to SNMP requests from SNMP managers. It also generates asynchronous traps to SNMP managers.

(c) **SNMP Subagent** — A program or process that handles a specific portion of the MIB. It registers with the SNMP agent so the agent knows where to send requests for the variables requested.

(d) **MIB (Management Information Base)** — The information store of the definition and specification of SNMP managed objects.

(e) **Managed Objects** — The abstraction of resources being managed in SNMP. It is primarily used for managed object types.

(f) **Variable** — MIB variable is used primarily for managed object instance, but sometimes it is used interchangeably with Managed Object.

(g) **Trap** — The SNMP notification of an interesting event, sent from SNMP Agent(s) to SNMP Manager(s).

5. REQUIREMENTS

The managed objects in DCE SNMP MIB must be essential for system management. In order to retain ''simplicity'' and ''light-weight'', a minimum set of managed objects is defined.

The portability of DCE subagent on different vendor platforms should be taken into consideration. The agent-subagent protocol should be an open standard.

The extensibility of the DCE subagent to support SNMPv2 should be taken into consideration.

6. FUNCTIONAL DEFINITION

6.1. The SNMP Overview

An SNMP network management system contains three components:

(a) one or more network elements (nodes), each containing an agent;

(b) at least one network management station (NMS); and

(c) a network management protocol, which is used by the station and the agents to exchange management information.

SNMP employs a request-response message exchange model. The SNMP Manager runs on NMS to monitor and control network elements where each network element has an SNMP Agent running. The SNMP Manager communicates with SNMP Agent on each node individually. The SNMP agent is responsible for performing network management functions requested by SNMP Manager.

Management information is the abstraction of managed resources. They are defined as managed objects. The collection of managed objects are called MIB. SNMP Manager accesses managed objects in the MIB through SNMP Agent by management operations.

The SNMP models all management operations as inspections or alterations of managed objects. Thus, a protocol entity on a logically remote host interacts with the agent resident on the network element in order to retrieve or alter variables. The management operations are: Get, Get-Next, and Set. The agent can emit unsolicited messages through Traps.

Management frameworks are to provide a support structure which isolates management application programmers on managing systems and system and application component programmers from having to deal with the complexity of these protocols so that they can focus on making their components more manageable.

Additional managed resources can be defined as MIB extensions for SNMP Managers to work with, without changing the SNMP functional interfaces. This extensibility preserves the simplicity and stability of the protocol. The MIB extension is supported by the subagent which works with the agent and is transparent to the SNMP Managers.

The strategy implicit in the SNMP is that the monitoring of network state at any significant level of detail is accomplished primarily by polling for appropriate information on the part of the manager. A limited number of unsolicited event notifications (traps) guide the timing and focus of the polling.

6.2. System Management Using SNMP

Since SNMP is widely used for network management in the TCP/IP environment, SNMP is chosen to be used for monitoring DCE resources and services.

A principle of SNMP is *simplicity*. It does not provide guaranteed delivery of messages. Therefore, some messages may be lost due to various reasons such as network congestion. The management strategies to deal with those conflicting requirements include:

(a) Polling — The management applications can poll the status of managed resources periodically.

(b) Management By Exceptions — Detect events by receiving SNMP traps, perform trap-directed polling.

(c) Management By Delegation — Delegates proxy, agent or subagents to perform status polling locally.

The enablement of these strategies have been taken into consideration in the definition of DCE management information and in the design of the DCE Subagent.

6.3. SNMP Agent and Subagents on The Managed Host

The enablement of DCE SNMP manageability on a managed DCE host depends on the following:

(a) SNMP Agent running on the host.

(b) SNMP Agent supporting an agent-to-subagent protocol.

The SNMP Agent supports the standard SNMP Protocol Data Unit (PDU) of Get, GetNext, and Set operations as well as event notifications known as Traps. Managed objects are defined in MIB. They are passed in ASN.1 notation. The SNMP Agent handles the encode and decode of the PDU.

In order for SNMP Agent to extend its ability to support additional managed objects in other MIBs, it supports agent-to-subagent protocols. For example, the Internet RFC 1592, ‘‘SNMP Distributed Program Interface Version 2.0’’, is used for the reference implementation.

6.4. DCE Management Information

Although DCE provides location transparency which means the views can be cell-wide as opposed to node-wide, due to the nature of the SNMP model, the DCE SNMP MIB is defined on a per node basis. A new object identifier by the name `SNMP (7)` has been registered under the tree `iso (1) org (3) osf (22) dce(1)`. Under this `iso.org.osf.dce.SNMP` node there are two nodes, namely `MIB (1)` and `subagent (2)`. The node `iso.org.osf.dce.SNMP.MIB` specifies the subtree for the DCE SNMP MIB.

The DCE SNMP MIB consists of the following 19 groups and traps:

(a) Component ID Group

(b) Software Component Information Group

(c) DCE Host Information Group

(d) DCE Server Table Group

(e) DCE Host Server Group

(f) DCE EMS Server Group

(g) DCE Security Server Group

(h) DCE CDS Server Group

(i) DCE CDS Advertiser Group

(j) DCE CDS Clerks Group

(k) DCE CDS Clerk Table Group

(l) DCE CDS Clearinghouse Table Group

(m) DCE CDS Cached Clearinghouse Table Group

(n) DCE GDA Server Group

(o) DCE DTS Entity Group

(p) DCE DTS Server Group

(q) DCE DTS Known Server Table Group

(r) DCE Traps

The descriptions of these 19 groups and traps are given in the following 19 subsections. Each subsection has a common format, namely, the objects or traps of the group are listed, and a description of the meaning of each object or trap is specified, thusly:

(a) `OBJECT #1` (in the case of DCE Traps, an ‘‘object’’ is a ‘‘trap’’)
MEANING of `OBJECT #1`.

(b) `OBJECT #2`
MEANING of `OBJECT #2`.

(c) ... etc. ...
... etc. ...

6.4.1. Component ID group

This group offers base-level identification of the component and represents the minimum amount of information that a component vendor should provide (when meaningful). The group contains the following attributes:

(a) `componentIDGroup`
Group of base-level component information.

(b) `aManufacturer`
The company that produced this component.

(c) `aProduct`
The name of this component or product.

(d) `aVersion`
The version string for this component.

(e) `aSerialNumber`
The serial number for this component.

(f) `aInstallation`
The time and date of the last install of the component on this system.

(g) `aVerify`
The verification level for this component.

6.4.2. Software component information group

This group extends the Component ID group with attributes that further define the software component. The group contains the following:

(a) `softwareCompInfoGroup`
Group of identifying information about this software component.

(b) `aMajorVersion`
Major version of this software component.

(c) `aMinorVersion`
Minor version of this software component.

(d) `aRevision`
Revision of this software component.

(e) `aBuild`
Manufacturer's internal identifier for this compilation.

(f) `aTargetOperatingSystem`
The operating system for which this software component is intended.

(g) `aLanguageEdition`
The language edition of this software component.

(h) `aIdentificationCode`
Stock Keeping Unit (SKU) for this software component.

6.4.3. DCE host information group

This group must be implemented by all managed DCE hosts. It contains the following generic DCE host information:

(a) `dceHostInfoGroup`
The basic information about the DCE host.

(b) `aDCEHostName`
The name of the DCE host.

(c) `aDCEHostCellName`
The name of the primary cell to which this DCE host belong.

(d) `aDCEHostCellNameAliases`
The aliases of the cell to which this DCE host belong.

6.4.4. DCE server table group

This group is a table of entries pertaining the DCE server information on the host. It has the following attributes:

(a) `dceSvrTable`
The table of DCE server information on this host.

(b) `aDCESvrName`
The type of the DCE server.

(c) `aDCESvrState`
The state of this DCE server.

(d) `aDCESvrStateValue`
The value of the state of this DCE server.

6.4.5. DCE host server group

This group contains the information of the Host Daemon on the host. It has the following information:

(a) `dceHostSvrGroup`
The information about the DCE Host daemon on this host.

(b) `aDCEHostSvrPid`
The process ID of the DCE Host daemon.

(c) `aDCEHostSvrUid`
The user ID of the DCE Host daemon.

(d) `aDCEHostSvrGid`
The group ID of the DCE Host daemon.

(e) `aDCEHostSvrInRpcCalls`
The number of RPC calls received by RPC runtime.

(f) `aDCEHostSvrOutRpcCalls`
The number of RPC calls initiated by RPC runtime.

(g) `aDCEHostSvrInRpcPkts`
The number of network packets received by RPC runtime.

(h) `aDCEHostSvrOutRpcPkts`
The number of network packets sent by RPC runtime.

(i) `aDCEHostSvrState`
The state of this DCE Host daemon.

6.4.6. DCE EMS server group

This group contains information about the EMS server on the host. The group contains the following:

(a) `dceEmsSvrGroup`
The information of EMS server on the host.

(b) `aEmsSvrPid`
The process ID of the DCE EMS server.

(c) `aEmsSvrUid`
The user ID of the DCE EMS server.

(d) `aEmsSvrGid`
The group ID of the DCE EMS server.

(e) `aEmsSvrInRpcCalls`
The number of RPC calls received by RPC runtime.

(f) `aEmsSvrOutRpcCalls`
The number of RPC calls initiated by RPC runtime.

(g) `aEmsSvrInRpcPkts`
The number of network packets received by RPC runtime.

(h) `aEmsSvrOutRpcPkts`
The number of network packets sent by RPC runtime.

(i) `aEmsSvrState`
The state and status of this EMS server.

6.4.7. DCE security server group

This group must be implemented by all managed DCE nodes which implement Security Service. The group contains the following:

(a) `dceSecSvrGroup`
The information of Security Server on the host.

(b) `aSecSvrRole`
The role of Security Server: Master or Replica.

(c) `aSecSvrMode`
The mode of Security Server: operational or in maintenance.

(d) `aSecRgyUpdTime`
The localized date and time the replica was last updated.

(e) `aSecRgyUpdSeq`
The sequence number of last update the replica received.

(f) `aSecSvrPid`
The process ID of the DCE Security Server.

(g) `aSecSvrUid`
The user ID of the DCE Security Server.

(h) `aSecSvrGid`
The group ID of the DCE Security Server.

(i) `aSecSvrInRpcCalls`
The number of RPC calls received by RPC runtime.

(j) `aSecSvrOutRpcCalls`
The number of RPC calls initiated by RPC runtime.

(k) `aSecSvrInRpcPkts`
The number of network packets received by RPC runtime.

(l) `aSecSvrOutRpcPkts`
The number of network packets sent by RPC runtime.

(m) `aSecSvrState`
The state of this DCE Security Server.

6.4.8. DCE CDS server group

This group contains the following:

(a) `dceCdsSvrGroup`
The information of CDS Server on the host.

(b) `aCdsSvrUpTime`
The time this server was created.

(c) `aCdsSvrChildUpdFails`
Number of times the server was unable to contact all the clearinghouses that store a replica of a particular child directory's parent directory and apply the child updates that have occurred since last skulk.

(d) `aCdsSvrCrucialReps`
Number of times a user attempted to remove a crucial replica >from this server.

(e) `aCdsSvrMaxSkewTime`
Maximum amount of time that a timestamp on a new or modified entry can vary >from local system time.

(f) `aCdsSvrLookupPathBrokens`
Number of broken connections between clearinghouses on this server and clearinghouses closer to the root.

(g) `aCdsSvrSecFails`
Number of times a server principal for this server was found to have inadequate permissions to perform a request operation.

(h) `aCdsSvrSkulkInitd`
Number of skulks initiated by this CDS Server.

(i) `aCdsSvrSkulkCmpltd`
Number of skulks successfully completed by this CDS Server.

(j) `aCdsSvrReadOps`
Number of read operations directed to this CDS Server.

(k) `aCdsSvrWriteOps`
Number of write operations to this CDS Server.

(l) `aCdsSvrPid`
The process ID of this CDS Server.

(m) `aCdsSvrUid`
The user ID of this CDS Server.

(n) `aCdsSvrGid`
The group ID of this CDS Server.

(o) `aCdsSvrInRpcCalls`
The number of RPC calls received by RPC runtime.

(p) `aCdsSvrOutRpcCalls`
The number of RPC calls initiated by RPC runtime.

(q) `aCdsSvrInRpcPkts`
The number of network packets received by RPC runtime.

(r) `aCdsSvrOutRpcPkts`
The number of network packets sent by RPC runtime.

(s) `aCdsSvrState`
The state of this CDS Server.

6.4.9. DCE CDS advertiser group

This group contains the information about the CDS Advertiser on the host. It has the following attributes:

(a) `dceCdsAdvGroup`
The group of CDS Advertiser.

(b) `aCdsAdvUpTime`
The start time of CDS advertiser.

(c) `aCdsAdvPid`
The process ID of this CDS Advertiser.

(d) `aCdsAdvUid`
The user ID of this CDS Advertiser.

(e) `aCdsAdvGid`
The group ID of this CDS Advertiser.

(f) `aCdsAdvInRpcCalls`
The number of RPC calls received by RPC runtime.

(g) `aCdsAdvOutRpcCalls`
The number of RPC calls initiated by RPC runtime.

(h) `aCdsAdvInRpcPkts`
The number of network packets received by RPC runtime.

(i) `aCdsAdvOutRpcPkts`
The number of network packets sent by RPC runtime.

(j) `aCdsAdvState`
The state of this CDS advertiser.

6.4.10. DCE CDS clerks group

This group contains the information about operations performed by the CDS clerks on the host. It has the following attributes:

(a) `dceCdsClerksGroup`
The group of operations done by the CDS clerks on the host.

(b) `aCdsClerksAuthFails`
The number of times requesting principals failed authentication procedures.

(c) `aCdsClerksCacheBypasses`
The number of requests to read attributes for which the clerks were specifically directed by the requesting application to bypass its own cache.

(d) `aCdsClerksCacheHits`
Total number of read requests directed to the clerks that were satisfied entirely by its cache.

(e) `aCdsClerksMiscOps`
Number of operations other than read and write performed by the clerks.

(f) `aCdsClerksReadOps`
Number of lookup operations performed by the clerks.

(g) `aCdsClerksWriteOps`
Number of requests to modify data processed by the clerks.

6.4.11. DCE CDS clerk table group

This group contains the information about the CDS clerks on the host. It has the following attributes:

(a) `cdsClerkTable`
The group of the CDS clerks on the host.

(b) `aClerkUser`
The user name of a clerk on the host.

(c) `aClerkUid`
The user ID of a clerk on the host.

(d) `aClerkPid`
The process ID of a clerk on the host.

(e) `aClerkUpTime`
The start time of a clerk on the host.

6.4.12. DCE CDS clearinghouse table group

This group contains the information about the CDS clearinghouse on the host. It has the following attributes:

(a) `cdsCHTable`
The table of CDS clearinghouse on the host.

(b) `aChName`
The full name of the clearinghouse.

(c) `aChDataCorrupts`
The number of times that the data corruption event was generated.

(d) `aChEnableCounts`
The number of times that the clearinghouse was enabled since it was last started.

(e) `aChRefReturns`
Number of requests directed to this clearinghouse that resulted in the return of a partial answer instead of satisfying the client's request.

(f) `aChSkulkFails`
The number of times that a skulk of a directory, initiated from this clearinghouse, failed to complete.

(g) `aChEntryMisses`
The number of times the clearinghouse entry missing event was generated.

(h) `aChRootLosts`
The number of times the root lost event was generated.

(i) `aChUpgFails`
The number of times that upgrades failed.

(j) `aChReadOps`
The number of read operations directed to this clearinghouse.

(k) `aChWriteOps`
The number of write operations directed to this clearinghouse.

6.4.13. DCE CDS cached clearinghouse table group

This group has a table of entries of cached clearinghouse on the host. It contains the following:

(a) `cdsCachedCHTable`
The table of cached CDS clearinghouse on the host.

(b) `aCachedCHName`
The name of the cached clearinghouse.

(c) `aCachedCHUpTime`
The date and time at which the clearinghouse was added to the cache.

(d) `aCachedCHMiscOps`
The number of operations other than read and write the clerk has performed on the clearinghouse represented by the cache entry.

(e) `aCachedCHReadOps`
The number of lookup operations the clerk has performed on the clearinghouse represented by the cache entry.

(f) `aCachedCHWriteOps`
The number of write operations the clerk has sent to the clearinghouse represented by the cache entry.

6.4.14. DCE global directory agent server group

This group must be implemented by all managed DCE nodes which implement GDA. The group contains the following:

(a) `dceGdaSvrGroup`
The group of Global Directory Agent.

(b) `aGdaSvrPid`
The process ID of the DCE GDA.

(c) `aGdaSvrUid`
The user ID of this DCE GDA.

(d) `aGdaSvrGid`
The group ID of this DCE GDA.

(e) `aGdaSvrInRpcCalls`
The number of RPC calls received by RPC runtime.

(f) `aGdaSvrOutRpcCalls`
The number of RPC calls initiated by RPC runtime.

(g) `aGdaSvrInRpcPkts`
The number of network packets received by RPC runtime.

(h) `aGdaSvrOutRpcPkts`
The number of network packets sent by RPC runtime.

(i) `aGdaSvrState`
The state of this Global Directory Agent.

6.4.15. DCE DTS entity group

This group must be implement by all managed DCE nodes which implement DTS. The group contains the following:

(a) `dceDtsEntityGroup`
The information of the DTS entity on this host.

(b) `aDtsdCurrTime`
Current time on the DTS node.

(c) `aDtsdBadProtocols`
The number of times the local node failed to process a received message containing an incompatible protocol version.

(d) `aDtsdBadTimeReps`
The number of times the local node failed to process a received message containing an incompatible timestamp format.

(e) `aDtsdInsufRes`
The number of times the node has been unable to allocate virtual memory.

(f) `aDtsdLocalNotIntersects`
The number of times the node's time interval failed to intersect with the computed interval of the servers.

(g) `aDtsdSyncCmpltd`
The number of times the node successfullt synchronized time.

(h) `aDtsdSysErrors`
The number of times a DTS operation detected a system error.

(i) `aDtsdTooFewSvrs`
The number of times a node failed to synchronize because it couldn't contact required minimum number of servers.

(j) `aDtsdGlobalTo`
The amount of time the node waits for a response to a WAN synchronization request before sending another request or declaring a global to be unavailable.

(k) `aDtsdLocalTo`
The amount of time the node waits for a response to a synchronization request before sending another request or declaring a server to be unavailable.

(l) `aDtsdPid`
The process ID of this DCE DTS entity.

(m) `aDtsdUid`
The user ID of this DCE DTS entity.

(n) `aDtsdGid`
The group ID of this DCE DTS entity.

(o) `aDtsdInRpcCalls`
The number of RPC calls received by RPC runtime.

(p) `aDtsdOutRpcCalls`
The number of RPC calls initiated by RPC runtime.

(q) `aDtsdInRpcPkts`
The number of network packets received by RPC runtime.

(r) `aDtsdOutRpcPkts`
The number of network packets sent by RPC runtime.

(s) `aDtsdState`
The state of this DTS entity.

6.4.16. DCE DTS server group

This group contains the information of local or global DTS servers on this host. It has the following attributes:

(a) `dceDtsSvrGroup`
The group of this DTS entity on the host.

(b) `aDtsSvrRole`
The role of the node as a DTS Server, local or global.

(c) `aDtsSvrCourierRole`
A server's interaction with the set of global servers.

(d) `aDtsSvrDiffEpochs`
The number of times the node received time response messages that had epoch number different.

(e) `aDtsSvrNoGlobals`
The number of times the clock could not contact any global clocks.

(f) `aDtsSvrNotResponds`
The number of times the clock could not contact a specific global clock.

(g) `aDtsSvrProviderErrors`
The number of times the external time-provider signaled a failure or the node was unable to access the time-provider.

6.4.17. DCE DTS known server table group

This group contains the information of a list of local or global DTS servers known by this DTS entity. It has the following attributes:

(a) `dtsKnownSvrTable`
The table of local or global DTS Servers known by this DTS entity.

(b) `aDtsKnownSvrName`
The name of the known DTS Server by this host.

(c) `aLastPolled`
The time that the known DTS Server was last polled.

(d) `aLastObsSkew`
The last observed time difference of the known DTS Server.

(e) `aUsedInLastSync`
The time that the DTS Server was last synchronized.

(f) `aLastObsTime`
The time that the known DTS Server was last observed.

(g) `aDtsSvrProto`
Transport protocol of the known DTS Server.

6.4.18. DCE traps

Various events emitted by the monitored DCE servers are routed to this subagent via the DCE Event Management System daemon. These are translated into traps and, then, sent to the SNMP agent. The traps emitted by this subagent contain the following information:

(a) `aDCEEventType`
The type of the DCE event.

(b) `aDCEEventText`
The text associated with the event.

(c) `aDCEEventSeverity`
The severity of the event.

(d) `aDCECellName`
The name of the DCE cell where the event occurred.

(e) `aDCEHostName`
The name of the host where the event occurred.

(f) `aDCETime`
The time when the event occurred.

(g) `aDCEProgram`
The program name where the event occurred.

(h) `aDCEComponent`
The name of the component where the event occurred.

(i) `aDCESubComponent`
The name of the subcomponent where the event occurred.

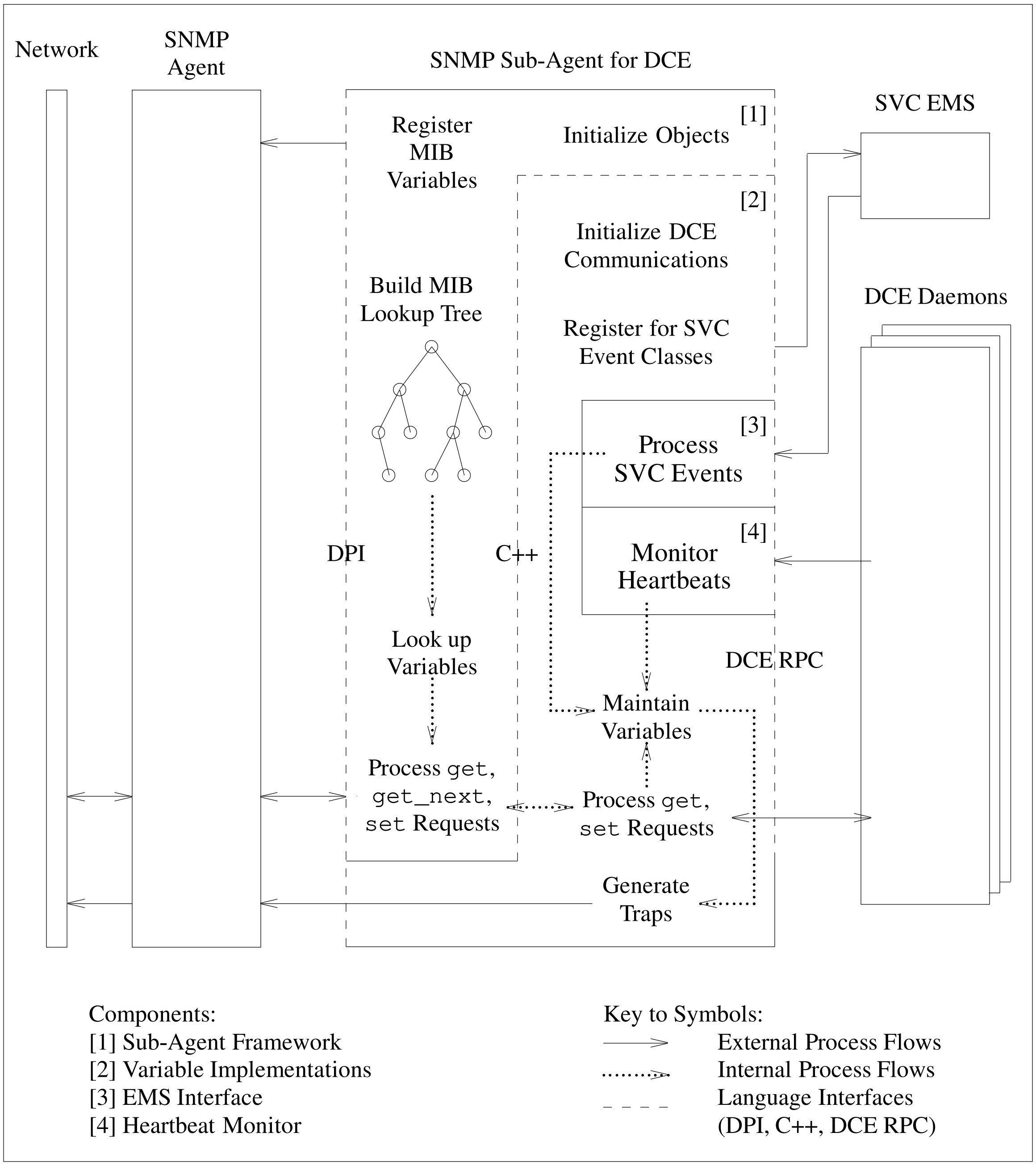
(j) `aDCEThreadId`
The ID of the thread where the event occurred.

(k) `aDCEFile`
The name of the file where the event occurred.

(l) `aDCELine`
The line number where the event occurred.

6.5. DCE Subagent

The SNMP Manager communicates with the SNMP Agent on each managed DCE host (see Figure). The SNMP Agent uses the agent-to-subagent protocol to communicate with DCE Subagent, which in turn invokes the DCE APIs to access DCE management information.



The agent works with the subagent through agent-to-subagent protocol. There are more than one way to implement a subagent. SMUX is one protocol often used on Unix platforms. DPI has been widely used by IBM and other vendors. The agent-to-subagent protocol used in IBM DCE products is DPI.

The DCE Subagent invokes the DCE APIs to access information of DCE resources or services and return the results to SNMP Agent. It is expected that the DCE Subagent accesses local information only.

The DCE Subagent registers with the DCE Event Management Service (EMS) as an event consumer to receive DCE event notifications of severity `SVC_FATAL` and `SVC_ERROR`. It also receives events that are specified in the file `/opt/dcelocal/etc/snmptrap.tbl`. This file contains message indices for the DCE events that can be of any severity (including `SVC_NOTICE` and `SVC_NOTICE_VERBOSE`). Additional events of interest can be added to this file by the administrator. Once these events are received through event filters by the Subagent from the EMS daemon, the Subagent converts these into SNMP traps and sends them to the SNMP agent, which, in turn, will route the traps to the listening manager applications. The mechanism to set up event filters, register and listen for events is described in the OSF RFC 67.0.

6.5.1. Subagent external interfaces

The DCE Subagent does not provide external functional interfaces to end users. However, there are interfaces provided to system administrators or management applications to configure, unconfigure, start and stop the Subagent. The SNMP Managers use standard SNMP requests to access DCE management information.

6.5.1.1. Configuration

The DCE SNMP Subagent can be configured by the command `mkdceagt`. It initially ensures that the Subagent is not already configured and that the EMS daemon is configured. Then it establishes the specified cell administration credentials, configures the DCE SNMP Subagent, and adds the DCE MIB to the SNMP MIB database.

Its command line interface is:

`mkdceagt` *`CellAdminId CellAdminPassword`* `[` *`TrapDestinationHost`* `... ]`

where:

(a) *`CellAdminId`* specifies the DCE account name for the cell administration (required argument; typically `cell_admin`).

(b) *`CellAdminPassword`* specifies the DCE password for the cell administration account, `CellAdminId` (required argument; typically `_dce_`).

(c) *`TrapDestinationHost`* specifies the hostname of the host to which SNMP traps should be sent. Multiple hostnames can be specified, in which case, traps will be sent to all these hosts upon generation. This argument is optional, and if not specified, traps will be sent to the local host.

6.5.1.2. Unconfiguration

The DCE Subagent can be unconfigured by the command `rmdceagt`. This command unconfigures the DCE SNMP Subagent. It initially ensures that the Subagent is not already unconfigured. Then it removes the DCE MIB from the SNMP MIB database. Subsequently, it establishes the cell administration credentials, and unconfigures the DCE SNMP Subagent.

Its command line interface is:

`rmdceagt` *`CellAdminId CellAdminPassword`*

where:

(a) *`CellAdminId`* specifies the DCE account name for the cell administration (required argument; typically `cell_admin`).

(b) *`CellAdminPassword`* specifies the DCE password for the cell administration account, `CellAdminId` (required argument; typically `_dce_`).

6.5.1.3. Starting

The DCE Subagent can be started by its command line interface. Its command line interface is:

`dceagtd [ -h` *`hostname`* `-c` *`community`* `-d` *`debuglevel`* `-shm -p` *`poll`* `-s` *`scope`* `-i` *`info`* `]`

where:

(a) `-p` is for local polling interval in seconds (integer greater than 0; default: 60).

(b) `-s` is for the scope of polling on DCE services (integer, range 0 to 15; default: 0).

(c) `-i` is for the level of DCE return information (integer, range 0 to 4; default: 0).

(d) `-h` is for the host to send requests to (hostname conforming to TCP/IP naming conventions; default: local host).

(e) `-c` is for the community (default: `dcesnmp`).

(f) `-shm` is for connecting over shared memory (default: TCP).

(g) `-d` is for the debug level (integer, range 0 to 9; default: 0).

If the configurable parameters are not given on the command line, the DCE Subagent will use pre-defined default values. Values for some of these parameters (`-p`, `-s` and `-i`) can also be set using Set request operations from the management applications, provided the selected community allows write access to the Subagent.

The DCE Subagent can also be started by the command `agtstart`. This command checks to see if the Subagent is not already running, the EMS daemon is running and the Subagent is configured. If these conditions are met, it starts the DCE SNMP Subagent.

Its command line interface is:

```
agtstart
```

The system administrator has to be the root user to start this subagent.

6.5.1.4. Stopping

The DCE Subagent can be stopped by the command `agtstop`. Its command line interface is:

```
agtstop
```

The DCE Subagent can also be stopped by Ctrl-Break or Ctrl-C.

6.5.2. Subagent content

The design of subagent is similar to that of SNMP Agent. Efforts have been made for ease of migration to future subagent.

The functional interfaces used between agent and subagent are provided by agent-to-subagent protocol. That includes:

(a) Initialization.

(b) Opening of agent-to-subagent connection.

(c) Registration of DCE management information.

(d) Waiting for request messages from SNMP Agent.

(e) Conveying response to SNMP Agent.

(f) Sending traps to SNMP Agent.

(g) Closing of agent-to-subagent connection.

Logically, the subagent can be structured into the following components, as described in the implementation specifications:

(a) Control Module.

(b) Object Handler.

(c) Heartbeat Monitor.

(d) Trap Generator.

6.5.3. DCE subagent design

The subagent has four main functions:

(a) Interface to the SNMP Agent via the Internet RFC 1592.

(b) Interface to DCE resources (services and databases) through DCE APIs to perform functions to retrieve or modify management information base (MIB) of the resources monitored.

(c) Interface to DCE servers through DCE APIs to periodically monitor the servers and report the change of status through SNMP traps to the SNMP Agent.

(d) Interface to the Event Management System as an event consumer to listen for event notifications coming from various DCE servers and applications and convert these into SNMP traps and send the traps to the SNMP Agent.

When the subagent is started, it will try to establish the connection with the SNMP Agent. Then, it will perform initialization to allocate memory for internal data structures. Then it will spawn threads to establish and maintain a login context and to manage the key in the keytab file. The subagent will also spawn threads for event receiving and heartbeat monitoring.

After initialization, the subagent enters a main loop. This main loop waits for incoming service requests from the SNMP Agent. When a request arrives, the message is decoded and the appropriate functions are performed to serve the request.

6.5.3.1. Control module

The subagent Control Module consists of the following functions:

(a) Initialization of environments.

(b) Handling of input arguments.

(c) Loading in the management information.

(d) Establishing a DCE login context and spawning threads to maintain it and to manage the key.

(e) Initialization of the connection to SNMP Agent via Internet RFC

(f) Spawning a thread to monitor heartbeats of the DCE servers.

(g) Spawning a thread to listen for event notifications.

(h) Main Loop.

(i) System-dependent functions.

The system-dependent library of SNMP agent can be used by the subagent.

6.5.3.2. Object handler

This module takes as input a pointer to the request data structure, which contains the VarBind list, operation, requestID, and some other elements.

It finds the variables in the MIB tree, makes sure the operation requested is allowed to perform on this variable, and performs the function, giving it the VarBind(s) and operation as parameters.

On return, the function informs the Object Handler, whether the operation was carried out successfully, and what the result is. The value part of the appropriate VarBind will have the value of the variable requested.

6.5.3.3. Heartbeat monitor

The Heartbeat Monitor periodically polls the DCE daemons to ensure that they are still up and running. When one is found to be down, the Heartbeat Monitor informs the Object Handler which will call the Trap Generator. The polling scope and interval can be controlled by variables in the DCE MIB. The administrator can issue SNMP request to set the value of these variables on demand.

6.5.3.4. Trap Generator

When the DCE Subagent detects interesting events either through polling or through receiving DCE event notifications, it will pass the information to the SNMP Agent to generate SNMP traps to the registered SNMP Managers. It initially sets up event filters with Event Management System to receive events of severity `SVC_ERROR`, `SVC_FATAL` and specific events as specified in the file `/opt/dcelocal/etc/snmptrap.tbl`. It then registers with the EMS and listens for the EMS events.

7. SNMP DATA

The SNMP users or applications use SNMP PDU to communicate with SNMP agent. The SNMP agent decodes the message and put the information into the PDU data structures of agent-to-subagent protocol. For further information, see Internet RFC 1157 (SNMPv1).

8. USER INTERFACES

There is no direct user interface to the DCE Subagent to query and set MIB objects. However, SNMP users communicate with SNMP agent which in turn interfaces with the Subagent. There are command line interfaces provided to configure, unconfigure, and start the Subagent, which are explained in the section Subagent External Interfaces.

9. API'S

The implementation of DCE subagent does not provide any external APIs. Users or application programs use SNMP APIs to communicate with SNMP agent which routes the requests to the Subagent and then receives the responses back.

10. REMOTE INTERFACES

There are no remote interfaces for DCE subagent. All remote SNMP requests are sent to the SNMP agent on the node.

11. MANAGEMENT INTERFACES

The external management interface provided by the subagent are the config, unconfigure, start and stop functions. The command line interfaces for these functions are explained in the section Subagent External Interfaces.

12. RESTRICTIONS AND LIMITATIONS

The DCE Subagent uses the Distributed Program Interface (DPI) 2.0 which is explained in the Internet RFC 1592.

13. OTHER COMPONENT DEPENDENCIES

The DCE subagent depends on the existence of SNMP agent and the agent-to-subagent support on the managed DCE nodes.

14. COMPATIBILITY

The Subagent works with the hosts running DCE 1.1 or later.

15. STANDARDS

The DCE SNMP Support follows the Internet IETF standards on defining the MIB and on specifying the subagent implementation complying with SNMPv1:

(a) RFC 1155, ''Structure and Identification of Management Information for TCP/IP-based Internets''.

(b) RFC 1157, ''A Simple Network Management Protocol''.

(c) RFC 1227, ''SNMP MUX Protocol and MIB''.

(d) RFC 1592, ‘‘SNMP Distributed Program Interface Version 2.0’’.

16. OPEN ISSUES

The SNMP agent-to-subagent interface is an issue for discussion. Since SMUX has been deprecated by IETF, DPI 2.0 is a prime candidate for this purpose.

The DFS managed objects will be modelled in future.

APPENDIX A. DCE MIB DEFINITIONS

```
IBMDCE-MIB DEFINITIONS ::= BEGIN

-- This component represents the base configuration of a DCE system
-- produced by OSF members.  The groups defined are as follows:
--       the Component ID group,
--       the Software Component Information group,
--       the DCE Host Information group,
--       the DCE Server Table group,
--       the DCE Host Server group,
--       the DCE EMS Server group,
--       the DCE Security Server group,
--       the DCE CDS Server group,
--       the DCE CDS Advertiser group,
--       the DCE CDS Clerks group,
--       the DCE CDS Clerk Table group,
--       the DCE CDS Clearinghouse Table group,
--       the DCE CDS Cached Clearinghouse Table group,
--       the DCE GDA Server group,
--       the DCE DTS Entity group,
--       the DCE DTS Server group,
--       the DCE DTS Known Server Table group, and
--       the DCE Traps.

IMPORTS
     Counter, enterprises
             FROM RFC1155-SMI
     OBJECT-TYPE
             FROM RFC-1212
     DisplayString
             FROM RFC1213-MIB
     TRAP-TYPE
             FROM RFC-1215;

Date                ::= OCTET STRING (SIZE (50))
osf                 OBJECT IDENTIFIER ::= { org 22 }
dce                 OBJECT IDENTIFIER ::= { osf  1 }
SNMP                OBJECT IDENTIFIER ::= { dce 7 }
MIB                 OBJECT IDENTIFIER ::= { SNMP 1 }
subagent            OBJECT IDENTIFIER ::= { SNMP 2 }

-- Component ID Group ===============================================

componentIDGroup OBJECT IDENTIFIER ::= {MIB 1}

aManufacturer OBJECT-TYPE
```

```
        SYNTAX          DisplayString (SIZE (0..64))
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The name of the manufacturer that produces
                        this component."
        ::= {componentIDGroup 1}

aProduct OBJECT-TYPE
        SYNTAX          DisplayString (SIZE (0..64))
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The name of the product for this component."
        ::= {componentIDGroup 2}

aVersion OBJECT-TYPE
        SYNTAX          DisplayString (SIZE (0..64))
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The version number of this component."
        ::= {componentIDGroup 3}

aSerialNumber OBJECT-TYPE
        SYNTAX          DisplayString (SIZE (0..64))
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The serial number for this instance of this
                        component."
        ::= {componentIDGroup 4}

aInstallation OBJECT-TYPE
        SYNTAX          Date
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The installation date of this component.
                        The Service Layer assigns this value, the
                        manufacturer should leave it blank."
        ::= {componentIDGroup 5}

aVerify OBJECT-TYPE
        SYNTAX          INTEGER
--      {
--              vAnErrorOccurred;CheckStatusCode                (0),
--              vThisComponentDoesNotExist                      (1),
--              vTheVerifyIsNotSupported                        (2),
--              vReserved                                       (3),
--              vComponent'sFunctionalityUntested               (4),
--              vComponent'sFunctionalityUnknown                (5),
--              vComponentIsNotFunctioningCorrectly             (6),
--              vComponentFunctionsCorrectly                    (7)
--      }
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "A code that provides a level of verification
                        that the component is still installed and
                        working."
```

```
        ::= {componentIDGroup 6}

-- Software Component Information Group =============================

softwareCompInfoGroup OBJECT IDENTIFIER ::= {MIB 2}

aMajorVersion OBJECT-TYPE
        SYNTAX          DisplayString (SIZE (0..32))
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "Major version of this software component."
        ::= {softwareCompInfoGroup 1}

aMinorVersion OBJECT-TYPE
        SYNTAX          DisplayString (SIZE (0..32))
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "Minor version of this software component."
        ::= {softwareCompInfoGroup 2}

aRevision OBJECT-TYPE
        SYNTAX          DisplayString (SIZE (0..32))
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "Revision of this software component."
        ::= {softwareCompInfoGroup 3}

aBuild OBJECT-TYPE
        SYNTAX          DisplayString (SIZE (0..32))
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "Manufacturer's internal identifier for this
                        compilation."
        ::= {softwareCompInfoGroup 4}

aTargetOperatingSystem OBJECT-TYPE
        SYNTAX          INTEGER
--      {
--              vOther                                          (0),
--              vDos                                            (1),
--              vMacos                                          (2),
--              vOs2                                            (3),
--              vUnix                                           (4),
--              vWin16                                          (5),
--              vWin32                                          (6),
--              vOpenvms                                        (7)
--      }
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The operating system for which this software
                        component is intended."
        ::= {softwareCompInfoGroup 5}

aLanguageEdition OBJECT-TYPE
        SYNTAX          DisplayString (SIZE (0..32))
```

```
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "This identifies the language edition of this
                        component."
        ::= {softwareCompInfoGroup 6}

aIdentificationCode OBJECT-TYPE
        SYNTAX          DisplayString (SIZE (0..32))
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "Stock Keeping Unit (SKU) or Part Number for
                        this software component."
        ::= {softwareCompInfoGroup 7}

-- DCE Host Information Group =======================================

dceHostInfoGroup OBJECT IDENTIFIER ::= {MIB 3}

aDCEHostName OBJECT-TYPE
        SYNTAX          DisplayString (SIZE (0..64))
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "An administratively-assigned name for this
                        managed DCE host."
        ::= {dceHostInfoGroup 1}

aDCEHostCellName OBJECT-TYPE
        SYNTAX          DisplayString (SIZE (0..255))
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "An administratively-assigned name for the
                        DCE cell to which this managed DCE host
                        belongs."
        ::= {dceHostInfoGroup 2}

aDCEHostCellNameAliases OBJECT-TYPE
        SYNTAX          DisplayString (SIZE (0..255))
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The aliases the host believes are defined
                        for the cell it belongs to."
        ::= {dceHostInfoGroup 3}

-- DCE  Server Table ================================================

dceSvrTable OBJECT-TYPE
        SYNTAX          SEQUENCE OF DCEHostSvrEntry
        ACCESS          not-accessibleMIB
        STATUS          mandatory
        DESCRIPTION     "A list of server entries configured on this
                        managed DCE host."
        ::= {MIB 4}

dceHostSvrEntry OBJECT-TYPE
        SYNTAX          DCEHostSvrEntry
```

```
        ACCESS          not-accessible
        STATUS          mandatory
        DESCRIPTION     ""
--      INDEX           {aDCESvrName}
        ::= {dceSvrTable 1}

DCEHostSvrEntry ::= SEQUENCE {
        aDCESvrName                                    DisplayString,
        aDCESvrState                                   DisplayString,
        aDCESvrStateValue                              INTEGER
}

aDCESvrName OBJECT-TYPE
        SYNTAX          DisplayString (SIZE (0..32))
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The type of this DCE server in this entry."
        ::= {dceHostSvrEntry 1}

aDCESvrState OBJECT-TYPE
        SYNTAX          DisplayString (SIZE (0..16))
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The state information of DCE Server on the
                        host."
        ::= {dceHostSvrEntry 2}

aDCESvrStateValue OBJECT-TYPE
        SYNTAX          INTEGER
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The state value of DCE Server on the host."
        ::= {dceHostSvrEntry 3}

-- DCE Host Server Group ============================================

dceHostSvrGroup OBJECT IDENTIFIER ::= {MIB 5}

aHostSvrPid OBJECT-TYPE
        SYNTAX          INTEGER
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The process ID of this DCE server on the
                        host."
        ::= {dceHostSvrGroup 1}

aHostSvrUid OBJECT-TYPE
        SYNTAX          INTEGER
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The POSIX identifier of the user the server
                        should be run as on the host."
        ::= {dceHostSvrGroup 2}

aHostSvrGid OBJECT-TYPE
```

```
        SYNTAX          INTEGER
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The POSIX identifier of the primary group
                        the server should be run as on the host."
        ::= {dceHostSvrGroup 3}

aHostSvrInRpcCalls OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of RPC calls received by RPC
                        runtime."
        ::= {dceHostSvrGroup 4}

aHostSvrOutRpcCalls OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of RPC calls initiated by RPC
                        runtime."
        ::= {dceHostSvrGroup 5}

aHostSvrInRpcPkts OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of RPC packets received by RPC
                        runtime."
        ::= {dceHostSvrGroup 6}

aHostSvrOutRpcPkts OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of RPC packets initiated by RPC
                        runtime."
        ::= {dceHostSvrGroup 7}

aHostSvrState OBJECT-TYPE
        SYNTAX          INTEGER
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The state information of DCED Server on the
                        host."
        ::= {dceHostSvrGroup 8}

-- DCE EMS Server Group =============================================

dceEmsSvrGroup OBJECT IDENTIFIER ::= {MIB 6}

aEmsSvrPid OBJECT-TYPE
        SYNTAX          INTEGER
        ACCESS          read-only
        STATUS          mandatory
```

```
        DESCRIPTION     "The process ID of this server."
        ::= {dceEmsSvrGroup 1}

aEmsSvrUid OBJECT-TYPE
        SYNTAX          INTEGER
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The POSIX identifier of the user the server
                        should be run as on the host."
        ::= {dceEmsSvrGroup 2}

aEmsSvrGid OBJECT-TYPE
        SYNTAX          INTEGER
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The POSIX identifier of the primary group
                        the server should be run as on the host."
        ::= {dceEmsSvrGroup 3}

aEmsSvrInRpcCalls OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of RPC calls received by RPC
                        runtime."
        ::= {dceEmsSvrGroup 4}

aEmsSvrOutRpcCalls OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of RPC calls initiated by RPC
                        runtime."
        ::= {dceEmsSvrGroup 5}

aEmsSvrInRpcPkts OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of RPC packets received by RPC
                        runtime."
        ::= {dceEmsSvrGroup 6}

aEmsSvrOutRpcPkts OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of RPC packets initiated by RPC
                        runtime."
        ::= {dceEmsSvrGroup 7}

aEmsSvrState OBJECT-TYPE
        SYNTAX          INTEGER
        ACCESS          read-only
        STATUS          mandatory
```

```
        DESCRIPTION     "The state information of EMS Server on the
                        host."
        ::= {dceEmsSvrGroup 8}

-- DCE Security Server Group =========================================

dceSecSvrGroup OBJECT IDENTIFIER ::= {MIB 7}

aSecSvrRole OBJECT-TYPE
        SYNTAX          INTEGER
--      {
--              vUnknown                                        (0),
--              vMastersecsvr                                   (1),
--              vReplica                                        (2)
--      }
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The role of the Security server."
        ::= {dceSecSvrGroup 1}

aSecSvrMode OBJECT-TYPE
        SYNTAX          INTEGER
--      {
--              vUnknown                                        (0),
--              vOperation                                      (1),
--              vMaintenance                                    (2)
--      }
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The mode of the Security server."
        ::= {dceSecSvrGroup 2}

aSecRgyUpdTime OBJECT-TYPE
        SYNTAX          Date
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The localized date and time the replica was
                        last updated."
        ::= {dceSecSvrGroup 3}

aSecRgyUpdSeq OBJECT-TYPE
        SYNTAX          INTEGER
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The sequence number of last update the
                        replica received."
        ::= {dceSecSvrGroup 4}

aSecSvrPid OBJECT-TYPE
        SYNTAX          INTEGER
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The process ID of this server."
        ::= {dceSecSvrGroup 5}
```

```
aSecSvrUid OBJECT-TYPE
        SYNTAX          INTEGER
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The POSIX identifier of the user the server
                        should be run as on the host."
        ::= {dceSecSvrGroup 6}

aSecSvrGid OBJECT-TYPE
        SYNTAX          INTEGER
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The POSIX identifier of the primary group
                        the server should be run as on the host."
        ::= {dceSecSvrGroup 7}

aSecSvrInRpcCalls OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of RPC calls received by RPC
                        runtime."
        ::= {dceSecSvrGroup 8}

aSecSvrOutRpcCalls OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of RPC calls initiated by RPC
                        runtime."
        ::= {dceSecSvrGroup 9}

aSecSvrInRpcPkts OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of RPC packets received by RPC
                        runtime."
        ::= {dceSecSvrGroup 10}

aSecSvrOutRpcPkts OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of RPC packets initiated by RPC
                        runtime."
        ::= {dceSecSvrGroup 11}

aSecSvrState OBJECT-TYPE
        SYNTAX          INTEGER
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The state information of the DCE Security
                        Server."
        ::= {dceSecSvrGroup 12}
```

```
-- DCE CDS Server Group ============================================

dceCdsSvrGroup OBJECT IDENTIFIER ::= {MIB 8}

aCdsSvrUpTime OBJECT-TYPE
        SYNTAX          Date
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The time this server was created."
        ::= {dceCdsSvrGroup 1}

aCdsSvrChildUpdFails OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "Number of times the server was unable to
                        contact all the clearinghouses that store a
                        replica of a particular child directory's
                        parent directory and apply the child updates
                        that have occurred since last skulk."
        ::= {dceCdsSvrGroup 2}

aCdsSvrCrucialReps OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "Number of times a user attempted to remove a
                        crucial replica from from this server."
        ::= {dceCdsSvrGroup 3}

aCdsSvrMaxSkewTime OBJECT-TYPE
        SYNTAX          INTEGER
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "Maximum amount of time that a timestamp on a
                        new or modified entry can vary from local
                        system time."
        ::= {dceCdsSvrGroup 4}

aCdsSvrLookupPathBrokens OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "Number of broken connections between
                        clearinghouses on this server and
                        clearinghouses closer to the root."
        ::= {dceCdsSvrGroup 5}

aCdsSvrSecFails OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "Number of times a server principal for this
                        server was found to have inadequate
                        permissions to perform a request operation."
```

```
        ::= {dceCdsSvrGroup 6}

aCdsSvrSkulkInitd OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "Number of skulks initiated by this CDS
                        server."
        ::= {dceCdsSvrGroup 7}

aCdsSvrSkulkCmpltd OBJECT-TYPE
        SYNTAX          INTEGER
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "Number of skulks successfully completed by
                        this CDS server."
        ::= {dceCdsSvrGroup 8}

aCdsSvrReadOps OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "Number of read operations directed to this
                        CDS server."
        ::= {dceCdsSvrGroup 9}

aCdsSvrWriteOps OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "Number of write operations to this CDS
                        server."
        ::= {dceCdsSvrGroup 10}

aCdsSvrPid OBJECT-TYPE
        SYNTAX          INTEGER
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The process ID of this server."
        ::= {dceCdsSvrGroup 11}

aCdsSvrUid OBJECT-TYPE
        SYNTAX          INTEGER
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The POSIX identifier of the user the server
                        should be run as on the host."
        ::= {dceCdsSvrGroup 12}

aCdsSvrGid OBJECT-TYPE
        SYNTAX          INTEGER
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The POSIX identifier of the primary group
                        the server should be run as on the host."
```

```
        ::= {dceCdsSvrGroup 13}

aCdsSvrInRpcCalls OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of RPC calls received by RPC
                        runtime."
        ::= {dceCdsSvrGroup 14}

aCdsSvrOutRpcCalls OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of RPC calls initiated by RPC
                        runtime."
        ::= {dceCdsSvrGroup 15}

aCdsSvrInRpcPkts OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of RPC packets received by RPC
                        runtime."
        ::= {dceCdsSvrGroup 16}

aCdsSvrOutRpcPkts OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of RPC packets initiated by RPC
                        runtime."
        ::= {dceCdsSvrGroup 17}


aCdsSvrState OBJECT-TYPE
        SYNTAX          INTEGER
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The state information of this CDS Server."
        ::= {dceCdsSvrGroup 18}

-- DCE CDS Advertiser Group ==========================================

dceCdsAdvGroup OBJECT IDENTIFIER ::= {MIB 9}

aCdsAdvUpTime OBJECT-TYPE
        SYNTAX          Date
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The start time of CDS advertiser."
        ::= {dceCdsAdvGroup 1}

aCdsAdvPid OBJECT-TYPE
        SYNTAX          INTEGER
```

```
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The process ID of CDS advertiser."
        ::= {dceCdsAdvGroup 2}

aCdsAdvUid OBJECT-TYPE
        SYNTAX          INTEGER
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The POSIX identifier of the user the server
                        should be run as on the host."
        ::= {dceCdsAdvGroup 3}

aCdsAdvGid OBJECT-TYPE
        SYNTAX          INTEGER
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The POSIX identifier of the primary group
                        the server should be run as on the host."
        ::= {dceCdsAdvGroup 4}

aCdsAdvInRpcCalls OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of RPC calls received by RPC
                        runtime."
        ::= {dceCdsAdvGroup 5}

aCdsAdvOutRpcCalls OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of RPC calls initiated by RPC
                        runtime."
        ::= {dceCdsAdvGroup 6}

aCdsAdvInRpcPkts OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of RPC packets received by RPC
                        runtime."
        ::= {dceCdsAdvGroup 7}

aCdsAdvOutRpcPkts OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of RPC packets initiated by RPC
                        runtime."
        ::= {dceCdsAdvGroup 8}

aCdsAdvState OBJECT-TYPE
        SYNTAX          INTEGER
```

```
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The state information of DCE CDS
                        Advertiser."
        ::= {dceCdsAdvGroup 9}

-- DCE CDS Clerks Group ============================================

dceCdsClerksGroup OBJECT IDENTIFIER ::= {MIB 10}

aCdsClerksAuthFails OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of times requesting principals
                        failed authentication procedures."
        ::= {dceCdsClerksGroup 1}

aCdsClerksCacheBypasses OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of requests to read attributes
                        for which the clerks were specifically
                        directed by the requesting application to
                        bypass its own cache."
        ::= {dceCdsClerksGroup 2}

aCdsClerksCacheHits OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "Total number of read requests directed to
                        the clerks that were satisfied entirely by
                        its cache."
        ::= {dceCdsClerksGroup 3}

aCdsClerksMiscOps OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "Number of operations other than Read and
                        Write performed by the clerks."
        ::= {dceCdsClerksGroup 4}

aCdsClerksReadOps OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "Number of lookup operations performed by the
                        clerks."
        ::= {dceCdsClerksGroup 5}

aCdsClerksWriteOps OBJECT-TYPE
        SYNTAX          Counter
```

```
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "Number of requests to modify data processed
                        by the clerks."
        ::= {dceCdsClerksGroup 6}

-- CDS Clerk Table ==================================================

cdsClerkTable OBJECT-TYPE
        SYNTAX          SEQUENCE OF CdsClerkEntry
        ACCESS          not-accessible
        STATUS          mandatory
        DESCRIPTION     "A list of CDS clerks on the host."
        ::= {MIB 11}

cdsClerkEntry OBJECT-TYPE
        SYNTAX          CdsClerkEntry
        ACCESS          not-accessible
        STATUS          mandatory
        DESCRIPTION     ""
--      INDEX           {aClerkPid}
        ::= {cdsClerkTable 1}

CdsClerkEntry ::= SEQUENCE {
        aClerkUser                                    DisplayString,
        aClerkUid                                     INTEGER,
        aClerkPid                                     INTEGER,
        aClerkUpTime                                  Date
}

aClerkUser OBJECT-TYPE
        SYNTAX          DisplayString (SIZE (0..64))
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The user name of a clerk on the host."
        ::= {cdsClerkEntry 1}

aClerkUid OBJECT-TYPE
        SYNTAX          INTEGER
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The user ID of a clerk on the host."
        ::= {cdsClerkEntry 2}

aClerkPid OBJECT-TYPE
        SYNTAX          INTEGER
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The process ID of a clerk on the host."
        ::= {cdsClerkEntry 3}

aClerkUpTime OBJECT-TYPE
        SYNTAX          Date
        ACCESS          read-only
        STATUS          mandatory
```

```
        DESCRIPTION     "The start time of a clerk on the host."
        ::= {cdsClerkEntry 4}
-- CDS Clearinghouse Table ==========================================

cdsCHTable OBJECT-TYPE
        SYNTAX          SEQUENCE OF CdsCHEntry
        ACCESS          not-accessible
        STATUS          mandatory
        DESCRIPTION     "A list of clearinghouses on the host."
        ::= {MIB 12}

cdsCHEntry OBJECT-TYPE
        SYNTAX          CdsCHEntry
        ACCESS          not-accessible
        STATUS          mandatory
        DESCRIPTION     ""
--      INDEX           {aChName}
        ::= {cdsCHTable 1}

CdsCHEntry ::= SEQUENCE {
        aChName                                                 DisplayString,
        aChDataCorrupts                                         Counter,
        aChEnableCounts                                         Counter,
        aChDisableCounts                                        Counter,
        aChRefReturns                                           Counter,
        aChSkulkFails                                           Counter,
        aChEntryMisses                                          Counter,
        aChRootLosts                                            Counter,
        aChUpgFails                                             Counter,
        aChReadOps                                              Counter,
        aChWriteOps                                             Counter
}

aChName OBJECT-TYPE
        SYNTAX          DisplayString (SIZE (0..255))
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The full name of the clearinghouse."
        ::= {cdsCHEntry 1}

aChDataCorrupts OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of times that the data corruption
                        event was generated "
        ::= {cdsCHEntry 2}

aChEnableCounts OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of times that the clearinghouse
                        was enabled since it was last started."
        ::= {cdsCHEntry 3}
```

```
aChDisableCounts OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of times that the clearinghouse
                        was disabled since it was last started."
        ::= {cdsCHEntry 4}

aChRefReturns OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of requests directed to this
                        clearinghouse that resulted in the return of
                        a partial answer instead of satisfying the
                        client’s request."
        ::= {cdsCHEntry 5}

aChSkulkFails OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of times that a skulk of a
                        directory, initiated from this clearinghouse,
                        failed to complete."
        ::= {cdsCHEntry 6}

aChEntryMisses OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of times the clearinghouse entry
                        missing event was generated."
        ::= {cdsCHEntry 7}

aChRootLosts OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of times the root lost event was
                        generated."
        ::= {cdsCHEntry 8}

aChUpgFails OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of times that upgrades failed."
        ::= {cdsCHEntry 9}

aChReadOps OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of read operations directed to
```

```
                        this clearinghouse."
        ::= {cdsCHEntry 10}

aChWriteOps OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of write operations directed to
                        this clearinghouse."
        ::= {cdsCHEntry 11}

-- CDS Cached Clearinghouse Table ===================================

cdsCachedCHTable OBJECT-TYPE
        SYNTAX          SEQUENCE OF CdsCachedCHEntry
        ACCESS          not-accessible
        STATUS          mandatory
        DESCRIPTION     "A list of cached clearinghouses on the
                        host."
        ::= {MIB 13}

cdsCachedCHEntry OBJECT-TYPE
        SYNTAX          CdsCachedCHEntry
        ACCESS          not-accessible
        STATUS          mandatory
        DESCRIPTION     ""
--      INDEX           {aCachedCHName}
        ::= {cdsCachedCHTable 1}

CdsCachedCHEntry ::= SEQUENCE {
        aCachedCHName                           DisplayString,
        aCachedCHUpTime                         Date,
        aCachedCHMiscOps                        Counter,
        aCachedCHReadOps                        Counter,
        aCachedCHWriteOps                       Counter
}

aCachedCHName OBJECT-TYPE
        SYNTAX          DisplayString (SIZE (0..255))
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The name of the cached clearinghouse."
        ::= {cdsCachedCHEntry 1}

aCachedCHUpTime OBJECT-TYPE
        SYNTAX          Date
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The date and time at which the clearinghouse
                        was added to the cache."
        ::= {cdsCachedCHEntry 2}

aCachedCHMiscOps OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
```

```
        STATUS          mandatory
        DESCRIPTION     "The number of operations other than Read and
                        Write the clerk has performed on the
                        clearinghouse represented by the cache
                        entry."
        ::= {cdsCachedCHEntry 3}

aCachedCHReadOps OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of lookup operations the clerk
                        has performed on the clearinghouse
                        represented by the cache entry."
        ::= {cdsCachedCHEntry 4}

aCachedCHWriteOps OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of Write operations the clerk has
                        sent to the clearinghouse represented by the
                        cache entry."
        ::= {cdsCachedCHEntry 5}

-- DCE GDA Server Group =============================================

dceGdaSvrGroup OBJECT IDENTIFIER ::= {MIB 14}

aGdaSvrPid OBJECT-TYPE
        SYNTAX          INTEGER
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The process ID of Global Directory Agent."
        ::= {dceGdaSvrGroup 1}

aGdaSvrUid OBJECT-TYPE
        SYNTAX          INTEGER
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The POSIX identifier of the user the GDA
                        should be run as on the host."
        ::= {dceGdaSvrGroup 2}

aGdaSvrGid OBJECT-TYPE
        SYNTAX          INTEGER
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The POSIX identifier of the primary group
                        the GDA should be run as on the host."
        ::= {dceGdaSvrGroup 3}

aGdaSvrInRpcCalls OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
```

```
        STATUS          mandatory
        DESCRIPTION     "The number of RPC calls received by RPC
                        runtime."
        ::= {dceGdaSvrGroup 4}

aGdaSvrOutRpcCalls OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of RPC calls initiated by RPC
                        runtime."
        ::= {dceGdaSvrGroup 5}

aGdaSvrInRpcPkts OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of RPC packets received by RPC
                        runtime."
        ::= {dceGdaSvrGroup 6}

aGdaSvrOutRpcPkts OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of RPC packets initiated by RPC
                        runtime."
        ::= {dceGdaSvrGroup 7}

aGdaSvrState OBJECT-TYPE
        SYNTAX          INTEGER
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The state information of Global directory
                        Agent."
        ::= {dceGdaSvrGroup 8}

-- DCE DTS Entity Group ==============================================

dceDtsEntityGroup OBJECT IDENTIFIER ::= {MIB 15}

aDtsdCurrTime OBJECT-TYPE
        SYNTAX          Date
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "Current time on the node."
        ::= {dceDtsEntityGroup 1}

aDtsdBadProtocols OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of times the local node failed to
                        process a received message containing an
                        incompatible protocol version."
```

```
        ::= {dceDtsEntityGroup 2}

aDtsdBadTimeReps OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of times the local node failed to
                        process a received message containing an
                        incompatible timestamp format."
        ::= {dceDtsEntityGroup 3}

aDtsdInsufRes OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of times the node has been unable
                        to allocate virtual memory."
        ::= {dceDtsEntityGroup 4}

aDtsdLocalNotIntersects OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of times the node's time interval
                        failed to intersect with the computed
                        interval of the servers."
        ::= {dceDtsEntityGroup 5}

aDtsdSyncCmpltd OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of times the node successfully
                        synchronized time."
        ::= {dceDtsEntityGroup 6}

aDtsdSysErrors OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of times a DTS process detected a
                        system error."
        ::= {dceDtsEntityGroup 7}

aDtsdTooFewSvrs OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of times a node failed to
                        synchronize because it couldn't contact
                        required minimum number of servers."
        ::= {dceDtsEntityGroup 8}

aDtsdGlobalTo OBJECT-TYPE
        SYNTAX          Date
```

```
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The amount of time the node waits for a
                        response to a WAN synchronization request
                        before sending another request or declaring a
                        global server to be unavailable."
        ::= {dceDtsEntityGroup 9}

aDtsdLocalTo OBJECT-TYPE
        SYNTAX          Date
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The amount of time the node waits for a
                        response to a synchronization request before
                        sending another request or declaring a server
                        to be unavailable."
        ::= {dceDtsEntityGroup 10}

aDtsdPid OBJECT-TYPE
        SYNTAX          INTEGER
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The process ID of DTS entity."
        ::= {dceDtsEntityGroup 11}
aDtsdUid OBJECT-TYPE
        SYNTAX          INTEGER
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The POSIX identifier of the user the DTS
                        entity should be run as on the host."
        ::= {dceDtsEntityGroup 12}

aDtsdGid OBJECT-TYPE
        SYNTAX          INTEGER
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The POSIX identifier of the primary group
                        the DTS entity should be run as on the host."
        ::= {dceDtsEntityGroup 13}

aDtsdInRpcCalls OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of RPC calls received by RPC
                        runtime."
        ::= {dceDtsEntityGroup 14}

aDtsdOutRpcCalls OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of RPC calls initiated by RPC
                        runtime."
        ::= {dceDtsEntityGroup 15}
```

```
aDtsdInRpcPkts OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of RPC packets received by RPC
                        runtime."
        ::= {dceDtsEntityGroup 16}

aDtsdOutRpcPkts OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of RPC packets initiated by RPC
                        runtime."
        ::= {dceDtsEntityGroup 17}

aDtsdState OBJECT-TYPE
        SYNTAX          INTEGER
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The state information of this DTS entity."
        ::= {dceDtsEntityGroup 18}

-- DCE DTS Server Group ==========================================

dceDtsSvrGroup OBJECT IDENTIFIER ::= {MIB 16}

aDtsSvrRole OBJECT-TYPE
        SYNTAX          Counter
--      {
--              vUnknown                                        (0),
--              vDtslocalsvr                                    (1),
--              vDtsglobalsvr                                   (2)
--      }
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The role of the DTS server, local or
                        global."
        ::= {dceDtsSvrGroup 1}

aDtsSvrCourierRole OBJECT-TYPE
        SYNTAX          INTEGER
--      {
--              vUnknown                                        (0),
--              vCourier                                        (1),
--              vBackupcourier                                  (2),
--              vNoncourier                                     (3)
--      }
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "A server's interaction with the set of
                        global servers."
        ::= {dceDtsSvrGroup 2}

aDtsSvrDiffEpochs OBJECT-TYPE
```

```
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of times the node received time
                        response messages from servers or clerks that
                        had epoch numbers different from its own."
        ::= {dceDtsSvrGroup 3}


aDtsSvrNoGlobals OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of times the clock could not
                        contact any global clock."
        ::= {dceDtsSvrGroup 4}

aDtsSvrNotResponds OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of times the clock could not
                        contact a specific global clock."
        ::= {dceDtsSvrGroup 5}

aDtsSvrProviderErrors OBJECT-TYPE
        SYNTAX          Counter
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The number of times the external
                        time-provider signaled a failure or the node
                        was unable to access the time-provider."
        ::= {dceDtsSvrGroup 6}

dtsKnownSvrTable OBJECT-TYPE
        SYNTAX          SEQUENCE OF DtsKnownSvrEntry
        ACCESS          not-accessible
        STATUS          mandatory
        DESCRIPTION     "A list of DTS local or global servers known
                        by this DTS entity."
        ::= {MIB 17}

dtsKnownSvrEntry OBJECT-TYPE
        SYNTAX          DtsKnownSvrEntry
        ACCESS          not-accessible
        STATUS          mandatory
        DESCRIPTION     ""
--      INDEX           {aDtsKnownSvrName}
        ::= {dtsKnownSvrTable 1}

DtsKnownSvrEntry ::= SEQUENCE {
        aDtsKnownSvrName                                DisplayString,
        aLastPolled                                     Date,
        aLastObsSkew                                    Date,
        aUsedInLastSync                                 Date,
```

```
        aLastObsTime                                       Date,
        aDtsSvrProto                                       INTEGER
}

aDtsKnownSvrName OBJECT-TYPE
        SYNTAX          DisplayString (SIZE (0..255))
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The name of the DTS server known by this
                        host."
        ::= {dtsKnownSvrEntry 1}

aLastPolled OBJECT-TYPE
        SYNTAX          Date
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The time that the known DTS Server was last
                        polled."
        ::= {dtsKnownSvrEntry 2}
aLastObsSkew OBJECT-TYPE
        SYNTAX          Date
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The last observed time difference of the
                        known DTS server."
        ::= {dtsKnownSvrEntry 3}

aUsedInLastSync OBJECT-TYPE
        SYNTAX          Date
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The time that the known DTS server was last
                        synchronized."
        ::= {dtsKnownSvrEntry 4}

aLastObsTime OBJECT-TYPE
        SYNTAX          Date
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The time that the known DTS server was last
                        observed."
        ::= {dtsKnownSvrEntry 5}

aDtsSvrProto OBJECT-TYPE
        SYNTAX          INTEGER
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "The transport protocol of the known DTS
                        server."
        ::= {dtsKnownSvrEntry 6}

-- DCE Traps  =======================================================

dceTrapsGroup OBJECT IDENTIFIER ::= {MIB 18}
```

```
aEventType  OBJECT-TYPE
        SYNTAX          DisplayString
        ACCESS          not-accessible
        STATUS          mandatory
        DESCRIPTION     "Type of the event."
        ::= {dceTrapsGroup 1}

aEventText  OBJECT-TYPE
        SYNTAX          DisplayString
        ACCESS          not-accessible
        STATUS          mandatory
        DESCRIPTION     "Text of the event."
        ::= {dceTrapsGroup 2}

aEventSeverity  OBJECT-TYPE
        SYNTAX          DisplayString
        ACCESS          not-accessible
        STATUS          mandatory
        DESCRIPTION     "Severity of the event."
        ::= {dceTrapsGroup 3}

aCellName  OBJECT-TYPE
        SYNTAX          DisplayString
        ACCESS          not-accessible
        STATUS          mandatory
        DESCRIPTION     "Name of the cell where the event occurred."
        ::= {dceTrapsGroup 4}

aHostName  OBJECT-TYPE
        SYNTAX          DisplayString
        ACCESS          not-accessible
        STATUS          mandatory
        DESCRIPTION     "Name of the host where the event occurred."
        ::= {dceTrapsGroup 5}

aTime  OBJECT-TYPE
        SYNTAX          DisplayString
        ACCESS          not-accessible
        STATUS          mandatory
        DESCRIPTION     "Time when the event occurred"
        ::= {dceTrapsGroup 6}

aProgram  OBJECT-TYPE
        SYNTAX          DisplayString
        ACCESS          not-accessible
        STATUS          mandatory
        DESCRIPTION     "Name of the program where the event
                        occurred."
        ::= {dceTrapsGroup 7}

aComponent  OBJECT-TYPE
        SYNTAX          DisplayString
        ACCESS          not-accessible
        STATUS          mandatory
        DESCRIPTION     "Name of the component where the event
```

```
                        occurred."
        ::= {dceTrapsGroup 8}

aSubComponent  OBJECT-TYPE
        SYNTAX          DisplayString
        ACCESS          not-accessible
        STATUS          mandatory
        DESCRIPTION     "Name of the subcomponent where the event
                        occurred."
        ::= {dceTrapsGroup 9}

aThreadId  OBJECT-TYPE
        SYNTAX          Integer
        ACCESS          not-accessible
        STATUS          mandatory
        DESCRIPTION     "Id of the thread where the event occurred."
        ::= {dceTrapsGroup 10}

aFile  OBJECT-TYPE
        SYNTAX          DisplayString
        ACCESS          not-accessible
        STATUS          mandatory
        DESCRIPTION     "Name of the file where the event occurred."
        ::= {dceTrapsGroup 11}

aLine  OBJECT-TYPE
        SYNTAX          Integer
        ACCESS          not-accessible
        STATUS          mandatory
        DESCRIPTION     "Line number where the event occurred"
        ::= {dceTrapsGroup 12}

END
```

REFERENCES

[RFC 1155] Internet RFC, ‘‘Structure and Identification of Management Information for TCP/IP-based Internets’’.

[RFC 1227] Internet RFC, ‘‘SNMP MUX Protocol and MIB’’.

[RFC 1592] Internet RFC, ‘‘SNMP Distributed Program Interface Version 2.0’’.

[RFC 32] R. Friedrich, ‘‘Requirements for Performance Instrumentation of DCE RPC and CDS Services’’, June 1993.

[RFC 33] R. Friedrich, S. Saunders, G. Zaidenweber, D. Bachmann, S. Blumson, ‘‘Standardized Performance Instrumentation and Interface Specification for Monitoring DCE-Based Applications’’, July 1995.

[RFC 47] J. Bowe, D. Mackey, R. Salz, P. Wang, DCE-RFC 47.3, ‘‘DCED: The DCE Host Daemon’’, April 1994.

[RFC 67] R. Cohen, G. Wilson, OSF RFC 67.0, ‘‘Event Management System’’, January 1996.

AUTHORS' ADDRESSES

Gregg Wilson
IBM Corporation
11400 Burnet Road, Zip 9132
Austin, TX 78758
USA
Internet email: gwilson@austin.ibm.com IBM
Telephone: +1-512-838-2244

Ravi Mandava
IBM Corporation
11400 Burnet Road, Zip 9132
Austin, TX 78758
USA
Internet email: ravikm@austin.ibm.com
Telephone: +1-512-838-8254