CIM Database Model

Presented by
Christina Shaw
and
Todd Guay
Chair Database Working Group
The CIM Database Working Group gratefully acknowledges the generous contributions of the following people and companies:

Linda Ball, Luigi Suardi  
Julie Schott, Andrea Westerinen  
Randy Horman, Will St. Germain IV  
Namik Hrle  
Christina Shaw  
Euan Garden  
Todd Guay (Chair), Sabit Mishra  
Steffen Hulegaard  
Andreas Koppel  
Marco De Luca  
Charlie Jones  
Daniel Forthman

BMC  
CISCO  
IBM  
IBM/Tivoli Systems  
Hewlett-Packard  
Microsoft  
Oracle  
OSA Technologies  
SAP AG  
Telecom Italia Lab  
Vision Solution  
Visionaef
Outline

• Charter
• Goals
• Terminology
• Overview DB Model
  – Conceptual Model and DB Schema
• Use Case: MyDB
• Future Directions
• Questions
• Define model that characterizes common properties and services performed by a database
  – inventory
    • static database properties
    • parameter settings
    • resource limits
    • features available and/or in use
  – behavioral
    • event
    • state
    • rules
    • Methods and attributes

• Common model must be independent of any particular database type or vendor implementation
DB Model Goals

• Provide set of classes and associations for
  – Managing generic database systems

• Provide common classes and associations
  – Across Database organizations and vendor implementations
  – Allow vendors to extend model for their specific content

• Provide content consistent with SNMP RDBMS MIB (RFC 1697) in CIM V2.7
  – Describes common entities and properties
  – Vendor independent relational DBMS
  – Supported by many vendors

• Provide extensions for database management
  – In future CIM versions
Focus for CIM V2.7

- Provide initial common model in a reasonably short timeframe
- Model or reuse existing classes to support basic management
  - such as administration and monitoring
- Incorporate capabilities in widely used IETF standard
  - RFC 1697 (RDBMS SNMP)
- Identify relationships and additions to other Common Models needed to support database management
  - Segue to future work
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Refers to the common database entity; a collection of interrelated data, treated as a unit, which is organized into one or more schemas.</td>
</tr>
<tr>
<td>Database Environment</td>
<td>consists of a database system, one or more databases, and the services that control the administration, usage, monitoring, and maintenance of a database.</td>
</tr>
<tr>
<td>Database Server</td>
<td>is the SNMP RDBMS term for the entity that provides access to the database. In CIM database model terminology, this is the database service.</td>
</tr>
<tr>
<td>Database Service</td>
<td>performs tasks for a database such as providing user access to the information within the database. Database services may be implemented as one or more processes.</td>
</tr>
<tr>
<td>Database System</td>
<td>is represents the application software portion of a database environment.</td>
</tr>
<tr>
<td>Relational Database</td>
<td>management system is a database where the schemas are organized based on the relational model.</td>
</tr>
<tr>
<td>Schema</td>
<td>is a collection of related database objects that reside in a database.</td>
</tr>
</tbody>
</table>
Overview DB Model

- Describes a 3 Level Model for a database environment:
  - Database system software
  - Common database entity
  - Database services
    - Entities that perform tasks for database
    - Such as coordinating user access

- CIM DB model is NOT specific to RDB management
Services and SAPs

- ManagedElement
  - (See Core Model)
- ManagedSystemElement
  - (See Core Model)
- LogicalElement
  - (See Core Model)
- Component
- Service
  - (See Core Model)
- ServiceAccessBySA
- ServiceAccessPoint
  - (See Core Model)
- DatabaseService
  - (See Database Model page (1))
Associations

Dependency (see Core Model)
Antecedent: ref ManagedElement [key]
Dependent: ref ManagedElement [key]

DatabaseAdministrator
Antecedent: ref CommonDatabase [key]
Dependent: ref UserEntity [key]

AssociatedDatabaseSystem
Antecedent: ref DatabaseSystem [1]
Dependent: ref CommonDatabase [key]

ServiceAvailableToElement (see Core Model)
ServiceProvided: ref Service [key]
UserOfService: ref ManagedElement [key]

ServiceAvailableToDatabase
ServiceProvided: ref DatabaseService [key]
UserOfService: ref CommonDatabase [key]
DatabaseSystem Class

SystemAdministrators

*UserEntity
(See User-Security Model)

*System
(See System Model Page (System))

ApplicationSystem
(See Application Model Page (1))

DatabaseSystem
DatabaseSystem Class

- Inherits from Application System
- Does not define additional properties
- Groups application systems that are database systems
- Models database system specific associations
- Represents software application aspects of DB
- Controls organization, retrieval, storage, security, and maintenance
• Includes software inventory information
• User-level software features
• CIM Application Model defines information about:
  – Software installation
  – Files and executable programs
  – Software features
  – Software version
  – Associations between files and programs and database
DatabaseSystem Class

Important properties inherited from CIM_System:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreationClassName</td>
<td>Indicates the name of the class or the subclass used in the creation of an instance; e.g. CIM_DatabaseSystem. Used with Name to uniquely identify instances within the class.</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the database system.</td>
</tr>
<tr>
<td>NameFormat</td>
<td>String; identifies how the database system name was generated.</td>
</tr>
<tr>
<td>PrimaryOwnerContact</td>
<td>String; identifies how the database system owner can be contacted - email address, phone number, pager, etc</td>
</tr>
<tr>
<td>PrimaryOwnerName</td>
<td>Name of the primary owner of the system.</td>
</tr>
<tr>
<td>Roles</td>
<td>String array ; specifies roles that database system plays in IT environment.</td>
</tr>
</tbody>
</table>
DatabaseSystem Class

Important properties inherited from CIM_ManagedElement:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OperationalStatus</td>
<td>The status of the database system.</td>
</tr>
<tr>
<td>InstallDate</td>
<td>The date/time when the database system was first installed.</td>
</tr>
<tr>
<td>Description</td>
<td>A longer textual description of the database system.</td>
</tr>
<tr>
<td>Caption</td>
<td>A short textual description of the database system that can be used for programmatic display in GUI.</td>
</tr>
</tbody>
</table>

Inherited SystemAdministrator Association identifies the user information for the person who installed the database system software.
Represented the relationship between a database system and the databases it controls.
An instance of DatabaseSystem
- May be associated with 0 or more databases
- Instance MUST exist for valid association

AssociatedDatabaseSystem
- May identify the DB_HOME for a specific common database
- Relates a specific common database instance to its system software
- Makes it possible to list the common database instances that are controlled by a DatabaseSystem instance
Database Administrator Association Class

Relates a database to the user entity that administers the database. A database may have 0 or more administrators. An administrator may be responsible for more than 1 database system.
DatabaseService instances identify manageable database service entities.
Database Service

- Processes that performs database tasks
  - Database servers in RFC 1697
- Co-ordinate user access to database
- Tasks may include:
  - user authentication
  - Authorization
  - concurrency control
  - data manipulation
  - integrity verification
  - data recovery
# DatabaseService Class

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>StartupTime</td>
<td>Date/time the service was started. Not valid if service is stopped. Maps to rdbmsSrvInfoStartupTime in RFC 1697</td>
</tr>
<tr>
<td>OperationalStatus</td>
<td>Overrides MSE description. Values/value map inherited from MSE. Operational status of service. OK: service is operational and available for general use, Stopped: service is unavailable and cannot be used, Service: implies administrative state of unavailability, Degraded: service is operating at less than optimal level, Starting: service is in process of becoming operational.</td>
</tr>
<tr>
<td>LastStatusChange</td>
<td>Date/time operational status of service last changed. Maps to rdbmsSrvInfoLastChange in RFC 1697.</td>
</tr>
<tr>
<td>Time</td>
<td></td>
</tr>
<tr>
<td>ConnectionLimit</td>
<td>Max # active inbound connections that can be concurrently open for service. Maps to rdbmsSrvInfoMaxInboundAssociations in RFC 1697.</td>
</tr>
</tbody>
</table>
## DatabaseService Class

### Important properties inherited from CIM_Service:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SystemCreationClassName</td>
<td>CreationClassName for system where database service is running.</td>
</tr>
<tr>
<td>SystemName</td>
<td>Name of system where DB service is running.</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>Name of class or subclass used when the instance was created; e.g. CIM_DatabaseService. Used with Name to uniquely identify instances within the class.</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the database service.</td>
</tr>
<tr>
<td>StartMode</td>
<td>Indicates whether database service startup is manual or automatic.</td>
</tr>
<tr>
<td>Started</td>
<td>Boolean that indicates whether the database service is started or stopped.</td>
</tr>
</tbody>
</table>
DatabaseService Class

Important properties inherited from CIM_ManagedElement:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InstallDate</td>
<td>The date/time when the database system was first installed.</td>
</tr>
<tr>
<td>Description</td>
<td>A longer textual description of the database system.</td>
</tr>
<tr>
<td>Caption</td>
<td>A short textual description of the database system that can be used for programmatic display in GUI.</td>
</tr>
</tbody>
</table>
• Represents the logical database entity
• Describes vendor and database organization agnostic properties
• Names a specific, manageable organized body of related information
• Instances of CIM_CommonDatabase span database organizations
• Supports hierarchical, relational, object-oriented, mixed, and other database model implementations
• Each separately manageable database, regardless of how it was created must have an instance in the CIM_CommonDatabase class
CommonDatabase Properties

- **CreationClassName**  [key]
- **Name**  [key]
- **DatabaseVersion**
  - Version number for the database.
  - When not relevant, this value must be set to NULL
- **SizeAllocated**
  - Estimated amount of disk space (in units) reserved for DB
- **SizeUnits**
  - Values: Bytes, Kilobytes, Megabytes, Gigabytes, Terabytes
- **LastBackUp**
  - Date and time of latest complete or partial backup of DB
  - Should be set to all zeros if DB has never been backed up
Important properties inherited from CIM_ManagedElement:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OperationalStatus</td>
<td>The status of the database system.</td>
</tr>
<tr>
<td>InstallDate</td>
<td>The date/time when the database system was first installed.</td>
</tr>
<tr>
<td>Description</td>
<td>A longer textual description of the database system.</td>
</tr>
<tr>
<td>Caption</td>
<td>A short textual description of the database system that can be used for programmatic display in GUI.</td>
</tr>
</tbody>
</table>
ServiceAvailableToDatabase Association Class

Relates database services to databases. A database may have 0 or more database services. A database service may serve 0 or more databases.
ServiceAvailableToDatabase Association Class

• Relationship between a database and its database services is
  – determined by the architecture of the vendor implementation

• May be used to identify
  – number of database services providing access to a specific database
  – (by association) status of the service
  – (by association) host where the service is running

• For a specific database service
  – may be used to determine the databases being served
ServiceAvailableToDatabase
Association Class Properties

• **AvailableState**
  - Current state DB service wrt its ability to access a specific DB
    • ACTIVE: Service is actively using DB
    • AVAILABLE: Service is waiting for a task to perform
    • RESTRICTED: Service is less than completely available for use
    • UNAVAILABLE: DB is not accessible from this service
    • OTHER: Any service states

• **OtherAvailableState**
  - Describes meaning when AvailableState = Other

• **AvailableTime**
  - Time the DB was made active by this service
  - If AvailableState is not active, then the value of this property MUST be set to zero
Database Parameters

• CIM Database Model was designed to offer flexibility
  – To accommodate different implementation styles
  – Allow configuration settings on the database only
  – Allow configuration settings on the database service only
  – Allow configuration settings on both

• DatabaseParameter subclasses ScopedSettingData
  – Abstract class
  – Specifies database and service configuration parameters
  – Common across DB models and vendor implementations
**SNMPDatabaseParameter**

- Extends CIM_DatabaseParameter
- Defines parameters as a name value pair
  - SNMP RDBMS MIB defines DB and service parameters as name/value pair combinations
  - Rather than as named properties of the class
  - Provides consistent mapping between SNMP and CIM
- SNMP RDBMS MIB defines generic tables
  - Rows of named database or service configuration entries.
- A parameter entry contains:
  - parameter name
  - description
  - value
- rdbmsDbParamTable and rdbmsSrvParamTable contain more information on the SNMP definitions
Database Parameters

- **CIM_Policy**
  - Establishes the parameter context for a specific DB or DB service
  - Using the CIM_ElementProfile association

- **Future database model versions may create new subclasses of DatabaseParameter**
  - To specify properties for common configuration parameters
DatabaseResourceStatics

ManagedElement
(See Core Model)

Dependency

* StatisticalData
  (See Core Model)

* RelatedStatistics

ElementStatisticalData
(See Core Model)

DatabaseResourceStatistic
Current: uint64
Limit: uint64
Highwater: uint64
Failures: uint32

CommonDatabaseStatistics
SizeUsed: uint32

DatabaseServiceStatistics
LastActivity: datetime
ActiveConnections: uint64
CumulativeConnections: uint64
RejectedConnections: uint64
CompletedTransactions: uint64
DiskReads: uint64
DiskWrites: uint64
LogicalReads: uint64
LogicalWrites: uint64
PageReads: uint64
PageWrites: uint64
DiskSpaceUnavailable: uint64
RequestsHandled: uint64
RequestsReceived: uint64
RequestsSent: uint64
HighwaterConnections: uint64
• Focuses primarily on statistics defined in RFC 1697
• No additional classes or associations added for DB software modeling
• Future DB model release Model may support specific database software capabilities
• Three primary classes of statistics:
  – resource statistics
  – common database statistics
  – database service statistics.
  – Each class inherits from CIM_StatisticalData
    • InstanceID, Name, Description, Caption
DatabaseResourceStatistics

Provides statistics for resources that have limits enforced by DB. There is one instance of this class for each limited DB resource. E.g, the DB may impose a limit on number of locks, or amount of disk space that can be allocated for a DB partition. Maps to the rdbmsLimitedResourceTable in RFC 1697

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current</td>
<td>Current value of the limited resource. rdbmsDbLimitedResourceCurrent in RFC 1697</td>
</tr>
<tr>
<td>Limit</td>
<td>Maximum value a DB resource can attain. E.g, if a resource is defined to limit number of DB locks, the limit might be set to 10,000 locks. This means that no more than 10,000 locks can be held at one time for DB usage. Maps to rdbmsDbLimitedResourceLimit in RFC 1697</td>
</tr>
<tr>
<td>Highwater</td>
<td>Maximum value for DB resource measured from when DB service was started. A precise value for ‘when’ in non-determinate. Maps to rdbmsDbLimitedResourceHighwater in RFC 1697</td>
</tr>
<tr>
<td>Failures</td>
<td>Count of number of times the DB resource limit would have been exceeded if the resource was allowed to be consumed beyond the limit. Maps to rdbmsDbLimitedResourceFailures in RFC 1697</td>
</tr>
</tbody>
</table>

DB resources specified in DatabaseResourceStatistics are associated to system resources through the CIM_StatisticsForDatabaseResources association.
## DatabaseServiceStatics

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LastActivity</td>
<td>Date/time most recent inbound connection was made to service. All zeros indicates no inbound connections since service started. Maps to rdbmsSrvInfoLastInboundActivity in RFC 1697.</td>
</tr>
<tr>
<td>ActiveConnections</td>
<td>Number of active inbound connections that are currently using service.</td>
</tr>
<tr>
<td>CumulativeConnections</td>
<td>Total number of inbound connections to the service since it started.</td>
</tr>
<tr>
<td>RejectedConnections</td>
<td>Total number of inbound connections rejected by service since it started.</td>
</tr>
<tr>
<td>CompletedTransactions</td>
<td>Total number of transactions completed by a commit or abort. Some database operations, like read-only queries, may not create a transaction.</td>
</tr>
<tr>
<td>DiskReads</td>
<td>Total number of DB file reads issued by the service since it started.</td>
</tr>
<tr>
<td>DiskWrites</td>
<td>Total number of DB file writes issued by service since it started.</td>
</tr>
<tr>
<td>LogicalReads</td>
<td>Total number of logical DB file reads issued by service since it started.</td>
</tr>
<tr>
<td>LogicalWrites</td>
<td>Total number of logical DB file writes issued by service since it started. A logical write is a count of the number of times that parts of DB files have been marked dirty to indicate that they need to be written to disk.</td>
</tr>
</tbody>
</table>

DMTF 2002 Developers' Conference June 10-13, 2002  Page 40
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PageReads</td>
<td>Total number of DB pages read by the service since it was started.</td>
</tr>
<tr>
<td>PageWrites</td>
<td>Total number of DB pages written by the service since it was started.</td>
</tr>
<tr>
<td>DiskSpaceUnavailable</td>
<td>Total number of times DB requested disk space and it was not available since the service started.</td>
</tr>
<tr>
<td>RequestsHandled</td>
<td>Total number of requests received by service since it started.</td>
</tr>
<tr>
<td>RequestsReceived</td>
<td>Total number of receive operations made by service during request processing since it started.</td>
</tr>
<tr>
<td>RequestsSent</td>
<td>Total number of send operations made by the service during request processing since it started.</td>
</tr>
<tr>
<td>HighwaterConnections</td>
<td>Maximum number of active inbound connections concurrently using the service since it started.</td>
</tr>
</tbody>
</table>
• Modeling RFC 1697 traps
• No new modeling required
  – Out of Space
    • CIM.InstModification event where object is CIM.DatabaseResource type Memory and property is OutOfSpace
  – State Change
    • CIM.InstModification event
Use Case: MyDB

- Vendor neutral demonstration of CIM DB model
- Assumes providers are supplied for each class
- Provides consistent management content across the enterprise
- Ensures management content available for MyDB product
- For all aspects of product lifecycle.
  - Database Software installation
  - Database creation
  - Database operation
CIM Object Manager discovers and populates the CIM schema via class providers. The MyDB vendor includes a provider for MyDB product to create an instance of CIM_DatabaseSystem during database software installation.

**Contents of CIM_DatabaseSystem:**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreationClassName</td>
<td>CIM_DatabaseSystem</td>
</tr>
<tr>
<td>Name</td>
<td>MyDB_V10Home</td>
</tr>
<tr>
<td>PrimaryOwnerName</td>
<td>Cynthia Smith</td>
</tr>
<tr>
<td>PrimaryOwnerContact</td>
<td>Email:<a href="mailto:cynthia.smith@customer.com">cynthia.smith@customer.com</a>; phone @(603) 123-4567; pager: (603) 123-7654</td>
</tr>
<tr>
<td>OperationalStatus</td>
<td>3</td>
</tr>
<tr>
<td>InstallDate</td>
<td>28-May-2002 11:21.02</td>
</tr>
<tr>
<td>Caption</td>
<td>The software home location where MyDB product is installed.</td>
</tr>
</tbody>
</table>
Additional classes from the core, application and other CIM schemas are populated during the Installation process to provide complete management content. Here is an example of the CIM_Product class to illustrate what one of the supporting classes would look like.

**Contents of CIM_Product:**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>MyDB</td>
</tr>
<tr>
<td>IdentifyingNumber</td>
<td>1</td>
</tr>
<tr>
<td>Vendor</td>
<td>MyDB Company Inc</td>
</tr>
<tr>
<td>Version</td>
<td>V1.0</td>
</tr>
<tr>
<td>SKU?Number</td>
<td>11233</td>
</tr>
<tr>
<td>WarrantyStartDate</td>
<td>28-May-2002 11:21.02</td>
</tr>
<tr>
<td>WarrantyDuration</td>
<td>1</td>
</tr>
<tr>
<td>Caption</td>
<td>MyDB Database Software Version 1.0</td>
</tr>
</tbody>
</table>
An instance of CIM_CommonDatabase is created to represent the logical database entity during the create database operation.

**Contents of CIM_CommonDatabase:**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreationClassName</td>
<td>CIM_CommonDatabase</td>
</tr>
<tr>
<td>Name</td>
<td>\golem\usr\database\mydb\golem.bin</td>
</tr>
<tr>
<td>DatabaseVersion</td>
<td>1.0</td>
</tr>
<tr>
<td>SizeAllocated</td>
<td>1.0</td>
</tr>
<tr>
<td>SizeUnits</td>
<td>3</td>
</tr>
<tr>
<td>OperationalStatus</td>
<td>10</td>
</tr>
<tr>
<td>LastBackup</td>
<td>0</td>
</tr>
<tr>
<td>InstallDate</td>
<td>31-May-2002 10:16.22</td>
</tr>
<tr>
<td>Caption</td>
<td>The Golem database contains Greek Mythology.</td>
</tr>
</tbody>
</table>
MyDB Creation

- MyDB would probably extend CommonDatabase for vendor specific properties
- Useful management information in CommonDatabase
  - Database is used to store info about Greek Mythology
  - It is database version 1.0
  - It has an allocated size of 10MB
  - It has not been backed up
  - It was created on May 31
  - The database is not running
  - CreationClassName and Name uniquely identify the DB in the enterprise
- CIM_AssociatedDatabaseSystem instance created
  - Relates new DB to the DatabaseSystem software instance
  - that created and controls it
An instance of CIM_DatabaseService is created during the create database operation representing MyDB product’s single service.

Contents of CIM_DatabaseService:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SystemCreationClassName</td>
<td>CIM_System</td>
</tr>
<tr>
<td>SystemName</td>
<td>GolemHost</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>CIM_DatabaseService</td>
</tr>
<tr>
<td>Name</td>
<td>Golem</td>
</tr>
<tr>
<td>StartupTime</td>
<td>0</td>
</tr>
<tr>
<td>OperationalStatus</td>
<td>10</td>
</tr>
<tr>
<td>LastStatusChangeTime</td>
<td>0</td>
</tr>
<tr>
<td>ConnectionLimit</td>
<td>1</td>
</tr>
<tr>
<td>StartMode</td>
<td>Manual</td>
</tr>
<tr>
<td>Started</td>
<td>0</td>
</tr>
<tr>
<td>InstallDate</td>
<td>31-May-2002 10:16.22</td>
</tr>
<tr>
<td>Caption</td>
<td>The Golem database service</td>
</tr>
</tbody>
</table>
MyDB Operation

- MyDB is targeted for small installations
  - With only a couple of users
- A few parameters control how the DB is configured
  - Number of database buffers
  - Database page size
  - Maximum number of users
- Only the number of buffers is configurable at the DB Service level (operational).
- MyDB company extends CIM_DatabaseParameter
  - To have more strongly typed parameter definitions than
  - name/value pairs offered by SNMPDatabaseParameter class
Example of CIM_MyDB_DatabaseParameter:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InstanceID</td>
<td>MyDBGUID (GUID represents a unique ID within a source vendor)</td>
</tr>
<tr>
<td>Name</td>
<td>\golem\usr\database\mydb\golem.bin</td>
</tr>
<tr>
<td>Caption</td>
<td>Database Parameter settings for the Golem Database</td>
</tr>
<tr>
<td>NumberBuffers</td>
<td>100</td>
</tr>
<tr>
<td>PageSize</td>
<td>4</td>
</tr>
<tr>
<td>MaximumUsers</td>
<td>1</td>
</tr>
</tbody>
</table>
Example of CIM_MyDB_DatabaseServiceParameter:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InstanceID</td>
<td>MyDBGUID (GUID represents a unique ID within a source vendor)</td>
</tr>
<tr>
<td>Name</td>
<td>\golem\usr\database\mydb\golem.bin</td>
</tr>
<tr>
<td>Caption</td>
<td>Database Service settings for the Golem Database</td>
</tr>
<tr>
<td>NumberBuffers</td>
<td>100</td>
</tr>
</tbody>
</table>
MyDB Operation

- 2 CIM_ScopedSetting association instances relate
  - Golem database to its possible parameter settings
  - Golem database service to its possible service settings
- 2 CIM_ElementSettingData associations updated
  - IsDefault set to 1
  - IsCurrent set to 0
- For future changes to DB or service parameter values:
  - Create new instances of CIM_DatabaseParameter and
  - CIM_ElementSettingData
    - IsDefault set to 2
    - IsCurrent set to 1
- CIM_ElementProfile instances relate
  - Golem database to its parameter settings
  - Golem database service to its service settings
  - Separate profiles can be used to represent default and current
Use the CIM Database Schema to manage the database!

- What DB services are running on system GolemHost?
- Which databases on the system GolemHost are open?
- Find the configuration parameter settings for DB X.
- How many connections are currently active on system X?
- What services provide access to database X?

Stumped? Answers can be found in the CIM Database Model White Paper.
Future Directions

- Common database management workflows such as BAR
- Database clusters
- Modeling relational database entities for SQL and schema
- Leveraging other modeling efforts such as OMG's CWM meta-model in the database warehouse domain
- Working in conjunction with the interoperability working group to deliver database vendor independent SQL DDL to hold CIM information.
- Extending the common database support for fault and event management, inventory, resource management, performance management, maintenance operations, and upgrade.
Want to join us?

June 13th
1:15 - 4pm PST
Room 3
Back Up Slides
• Database MIB (RFC 1697): 1994
  – Covers product, database, and server
    • ~55 metrics spanning performance statistics and configuration
    • 2 events – state change, out of space

• Pluses
  – Wide adoption
  – Simple, low overhead
  – Allows consistent access to database management data across vendors

• Negatives
  – Most content provided in vendor private MIB
  – Information is not organized into an object-oriented model