CA Management Database

Overview Guide

r1.5

Contents

Chapter 1: Introduction

CA Management Database (MDB) r1.5 is a common enterprise data repository that integrates CA product suites. The MDB provides a unified database schema for the management data stored by all CA products (distributed). Use of the MDB with CA products enables full integration for managing your IT infrastructure.

This Overview contains information you need to know about deploying single and multiple MDBs, administration, such as defining new users, and maintenance activities, such as database backup.

This section contains the following topics:

About CA MDB r1.0.4 and r1.5 Releases

This document contains information that applies to CA MDB r1.0.4 and CA MDB r1.5.

- CA MDB r1.04 supports Ingres, SQL Server, and Oracle database platforms.
- CA MDB r1.5 supports SQL Server and Oracle database platforms only and does not support Ingres.

Note that those sections of this document that refer only to CA MDB r1.0.4 contain CA MDB r1.0.4 in the associated topic headings.

Managing Information Technology

CIOs are challenged to deliver a highly flexible infrastructure with a high degree of automation. This challenge is frequently referred to as *on-demand computing* or *utility computing*. Delivering on-demand computing is essentially an IT management problem. The CIO needs to have advanced management solutions that help to meet three key goals:

- Reduce costs
- Mitigate risks
- Ensure infrastructure availability

Reduce Costs

To become a fully aligned business partner for an enterprise, IT must reduce costs. Total implementation costs are reduced significantly when products do the following:

- Make deployment and administration easier
- Enable better decision making
- Provide the means to automate business processes

IT management software should provide simple and flexible options that can deploy products one module at a time or by functional suite. When data sources are shared, fewer systems need to be administered, managed and monitored; and fewer staff resources are required. Increased visibility and timely access to integrated IT management data help managers make the best possible decisions.

Business process automation may eliminate the costs of manual processes. However, full automation requires complete information, which may be provided best when products share a single data source. For example, a CIO weighing expensive hardware or software purchases may obtain utilization rates for existing hardware and software--and review current lease agreements and contracts--prior to making a purchasing decision.

Mitigate Risks

Enterprises today face increasing risks and exposures in the areas of platform and application management, event management, business processes, records retention, and security. A unified data strategy enables the CIO and IT management staff to ensure that the IT infrastructure is fully secured and protected, and adheres to proven compliance and governance guidelines. Some of the primary approaches to minimizing such risks include:

- Applying consistent security
- Ensuring availability
- Implementing open standards

Managing risks is more difficult when IT environments include multiple heterogeneous databases and proprietary file formats. A common data source and data definitions provide the trace and audit capabilities that are required in complex regulatory environments. Data storage should be based on a relational database management system that provides transaction support, recovery, clustering and high availability. Extensible data definitions provide a way to manage additional proprietary data.

Software products for IT management need to be secure, scalable, robust, and built on mature, proven technologies and architectures. Support for open standards and interfaces, and implementation of best practices such as the Information Technology Infrastructure Library (ITIL), helps mitigate risks.

Ensure Infrastructure Availability

IT downtime means money lost to the business. Keeping the IT infrastructure running and available requires complete information about critical business systems in real time, not from secondary data sources that are extracted and reintegrated later. At the same time, real-time information requirements must not limit the deployment options for IT management products. Organizations should be able to deploy IT management products using either a single data source or multiple data sources.

If multiple sources of data are used, a unified view of information is necessary to ensure an available infrastructure. Highly available systems take advantage of clustered servers so that any failure automatically transitions work to another server in the cluster. Redundancy ensures that business may continue even when some network nodes are lost. IT management data sources hold time-critical information that ensures the availability of the infrastructure. These data sources themselves must be highly available, redundant, and clustered to minimize the business impact of hardware or software failures.

The MDB Approach

The Management Database (MDB) is a common enterprise data repository that integrates CA product suites. The MDB provides a unified database schema for the management data stored by all CA products (mainframe and distributed). Use of the MDB with CA products enables full integration for managing your IT infrastructure.

The MDB integrates management data from all IT disciplines and CA products. Customers may extend the MDB Schema to include additional IT management data from non-CA software products and tools.

Note: The MDB Schema is available from http://ca.com/support. Search the Technical Support knowledge base for more information.

The primary advantages of the MDB approach are:

- Integrated IT management data
- Integrated database administration

Integrated IT Management Data

A unified database schema for IT management data provides increased visibility into the underlying IT organization, including storage, performance, hardware, and software information. This visibility improves IT decision-making and helps to reduce costs and increase utilization of the infrastructure. Using a unified database schema enables fully-informed decisions about hardware, software and storage purchases, as well as provisioning, scheduling, data protection, and more.

The unified MDB schema enables integration of products without additional programming effort. Integration methods that do not include a single database schema require point-to-point programming efforts to access data from disparate sources. Data integration enables rapid development of new product features, because there is no need for programming interfaces to make disparate data available. Without an MDB, data is stored in multiple locations and schemas, making it difficult to integrate and to create new features that take advantage of data relationships.

Example

Without an MDB, it is difficult to determine whether a server located through a discovery process has been backed up, since the data for discovery and storage are maintained by separate products, in separate databases, and on separate servers. With an MDB, the data is always in one integrated schema. Storage and inventory management (discovery) are more tightly integrated. It is also possible to provide this feature to customers.

The MDB Schema

The MDB schema provides a unified and extensible model for enterprise IT management. It contains both common tables and product-specific tables that were previously implemented in separate product databases. The MDB serves as the enterprise database for all CA products and acts as the primary reference point. This allows you to write single queries that retrieve data from tables across different products.

The MDB schema includes the database objects used by CA products and their components. These include tables, columns, views, and procedures. The MDB manages operational and transaction data, as well as the analytical data used for intelligence and data mining. Depending on your organization's business needs and structure, you may use a single MDB or multiple MDBs; both approaches utilize the same schema.

Integrated Database Administration

A unified MDB schema means only one database to administer rather than many which makes the management data easier and less expensive to manage. On demand reporting, data mining, and other (non-product) operations that use the MDB become easier. The MDB uses a relational database management system, so its data is available and accessible through the well-established, easy to use Structured Query Language (SQL). In addition to ease of data access, relational database management systems provide a high degree of reliability, availability, and scalability as well as good performance and cluster support. Since the MDB uses a relational database management system, it inherits these advantages.

The MDB and Relational Databases for CA MDB r1.0.4

To store critical operational and IT data, a management database requires a commercial-grade relational database management system (RDBMS). All CA products that use CA MDB r1.0.4 come pre-packaged with a full-featured open source Ingres RDBMS at no additional cost.

What Does the MDB Contain?

CA MDB contains complete, consistent, and manageable definitions for all key IT management data. These definitions are shared by CA product solutions right out of the box, and they also may be extended to other products and third-party tools. MDB information supports CA solutions in the following IT management categories:

- Assets and inventory
- Storage
- Security
- Operations
- Services
- Job scheduling
- Servers and desktops
- Databases
- Application life cycle

This complete and consistent support for the full spectrum of IT management data provides a foundation for the integration of CA management solutions.

Starting with CA MDB r1.5, deployment flexibility is enhanced by installing only the database objects being used in a particular environment.

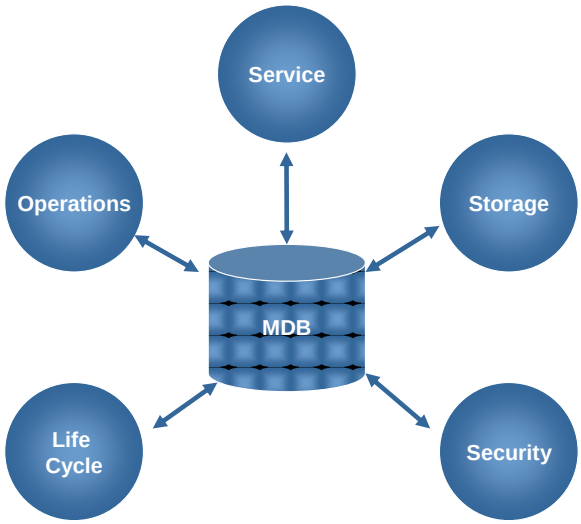
Chapter 2: MDB Deployment

This section contains the following topics:

Single Management Database

Deployment of a single global Management Database to store data for all products is the simplest to implement. This deployment option is shown in the following illustration. In general, this implementation costs the least and is the easiest to manage.

A single integrated view of the underlying IT infrastructure is available with no additional processing required. In this deployment, administration requirements, including establishing security and data availability, occur once for the single Management Database.

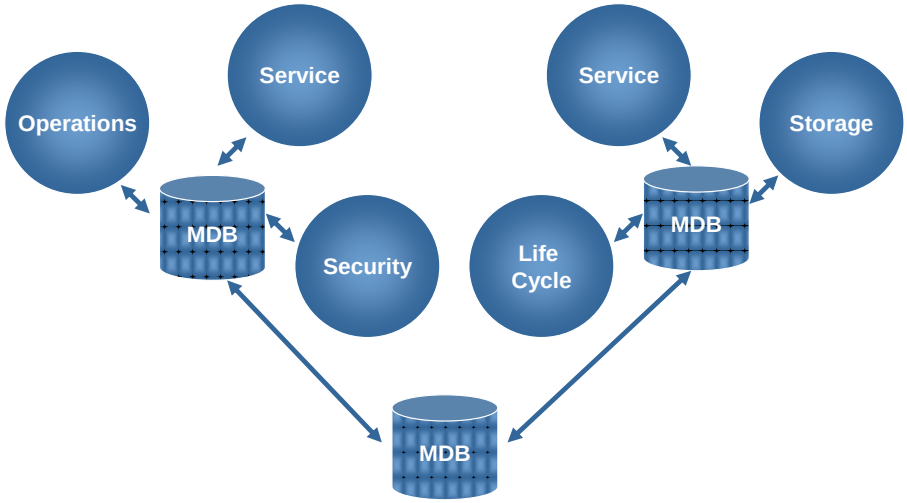


Starting with CA MDB r1.5, installation of database objects relevant to all IT infrastructure areas is no longer required.

Multiple MDBs

Multiple MDB instances may be needed for reasons of size, legal requirements, organizational structure, or geography. In addition, organizations may need to deploy one or more MDBs in IT environments that include non-CA software solutions.

The MDB enables a federated approach that uses multiple databases or data instances. To gain a single integrated view, data must be retrieved or accessed from multiple data instances as shown in the following illustration. Supporting multiple MDBs may require additional planning and administration.



In this scenario, one or more products access different MDBs. Each of the following approaches provides a single integrated view (for reporting purposes) of the underlying IT infrastructure:

- **Distributed query support**, provided by the relational database being used, enables a federated approach. With distributed query, multiple MDBs, each with their own management information, are segmented and implemented by function/geography and organization. Distributed queries allow data from different databases (MDBs) to appear as if it were from one database.
- **Replication**, available by the relational database being used and built into some of CA's management products directly, enables a federated approach. With replication, each function/geography or organization uses its own MDB for management information. Data from each MDB is replicated to a central database server with an MDB that provides a single, integrated view of the company's IT infrastructure. In a replicated approach, the central database server might also serve as an MDB supporting one or more CA products.

Planning for Availability

The data stored in the MDB is critical and is required for the operation of CA solutions that manage your IT infrastructure. It is important that you plan for availability in case of any eventuality that would put your IT infrastructure and the functionality of critical business processes at risk. Without such planning, the MDB could become a single point of failure. You may use several features of the underlying relational database management software to help protect against failures. These include the use of failover capabilities, cluster support, custom replication, and the performance of database backups on a regular basis.

Planning for availability must be an integral part of the implementation of your CA solutions.

Hardware

A planned and designed Management Database (MDB) environment provides a quality, high performing system with extendable growth capability. Lack of or incorrect architecture decisions may cause poor performance, lack of scalability, poor reliability, or poor availability. To prevent such problems, the following key areas should be considered prior to MDB deployment:

- Processor Technology (in database servers a minimum of two is preferred)
- Storage and I/O subsystems (in database servers a minimum of two I/O channels is recommended)
- Network Bandwidth and Availability (in database servers utilize backbone networks for linking servers to each other, where possible separate out client to server traffic from server to server traffic)
- Dedicated or shared server MDB deployment

Database Backup

The MDB must be backed up at regular intervals. In the event that disaster recovery is needed or a failure occurs, the MDB may be restored.

Replication

Replication capabilities enable you to move data from one database instance to another. By moving data to a secondary database instance on a regular basis, it is possible to recover from a disk crash, because processing may be resumed on a secondary database instance. The secondary database instance may be different from the original based on the intervals for which replication occurs.

Note: This is a custom solution and must be built for the specific CA products in use at your site.

Cluster Support

Active cluster support allows a system's workload to be spread across the nodes of a cluster. If one node fails, the remaining nodes continue processing to avoid loss of service. Relational database management systems exploit this capability and automatically balance the load across the nodes in the cluster.

Failover Support

Failover support enables the relational database management system to operate on a cluster as a distributed application, providing transparent access to an MDB that resides on shared storage devices. If there is a database or hardware failure on one of the nodes, processing continues uninterrupted using the remaining nodes.

Chapter 3: MDB Administration

This section contains the following topics:

Security Rules (CA MDB r1.0.4)

Enforcement of security rules for access to the MDB is provided by the underlying relational database management system (RDBMS).

Database objects in the MDB include tables, views, procedures, and rules. Typically, a CA product utilizes a specific set of database objects (such as tables, indexes, views, constraints and so forth) to expose information to the user. While there may be many shared objects between products, the MDB security model ensures that the user of a CA product may only access the database objects for that product. For example, a customer using CA's Unicenter Service Desk product may only access tables used by that product and may not access tables for another product.

The MDB security model restricts access to the database objects of a product by using underlying RDBMS user groups. User groups (Ingres), or roles (Oracle SQL Server) are created for each product or for each product application component. Users of a product, or users created for the product's component, are assigned to a user group as their default. Any number of users may be associated with a user group. A product's access to data, procedures, and rules is restricted by establishing user groups of database objects and associating database users with the groups.

The MDB is created with many user groups already defined. Products specify these groups when they connect to the database. If you choose to use external tools such as report writers to access MDB data, you may limit the report writer user's MDB access by using these user groups as part of the database connection information.

Define New Users

User definitions are required for access to the MDB. Some CA products create users automatically during installation; others require that you create users manually. Consult specific CA product documentation to determine what is required.

If you use an external tool such as a report writer for creating new applications to access MDB data, you may need to create new users for that purpose.

If you access database objects used by an existing CA product, any new database users you create must be added to an existing database access group for the CA product.

If you access new database objects, or database objects from several different products, you must create a new user group for the database and add the user to that group.

Maintenance

Regardless of database implementation, regular maintenance activities are required to ensure optimal performance and responsiveness for an MDB.

Maintenance activities such as database backup, system catalog reorganization, and file and disk space monitoring for optimization are essential for best performance. These activities must occur regardless of the number or variety of CA products that access an instance of the MDB. In addition, table reorganization and statistics are also useful for performance purposes.

Most databases provide for maintenance using command line utilities and SQL commands. These features allow sites to use command files and a job scheduler to automate the maintenance process.

More Information

For more information about specific maintenance utilities and commands, consult your database documentation.

Schema Upgrades

The MDB schema is upgraded when patches are applied or when an MDB is already installed and a newer MDB release is installed by another CA product.

Prior to any action that will upgrade the MDB schema, the database should be backed up by a database administrator. The backup provides the ability to restore in case an error occurs during the upgrade process.

Extend the MDB

Customers and third parties may extend the MDB schema to include additional IT management data. New database objects (for example, tables, views, and procedures) can be created that incorporate the IT management data from CA products and also provide additional data, features, and capabilities.

If a CA product already provides functionality for extending its data model, then that product's own functionality should be used. These CA products provide support for identifying MDB extensions and upgrading them for subsequent product releases.

The standard method for extending the MDB is to add new database objects (for example, tables, views, and procedures) as needed.

Important! To ensure the integrity of software products and data, customers must not change any standard database objects that were provided with the base MDB or add new objects to the “mdbadmin” domain as this may prohibit successful future upgrades if a duplicate object name is found.

For details about creating a schema, owner IDs, and database objects, consult your database documentation.

Chapter 4: Ingres MDB Considerations (CA MDB r1.0.4)

The information in this chapter applies to CA MDB r1.0.4.

- CA MDB r1.04 supports Ingres, SQL Server, and Oracle database platforms.
- CA MDB r1.5 supports SQL Server and Oracle database platforms only and does not support Ingres.

Note that those sections that refer specifically to CA MDB r1.0.4 contain CA MDB r1.0.4 in the topic headings.

This section contains the following topics:

Microsoft Windows Paging File for Ingres (CA MDB r1.0.4)

When an Ingres MDB is installed running under Microsoft Windows, you should have a minimum Microsoft Windows paging file size of two (2) gigabytes or one and a half times the RAM in the computer, whichever is greater.

Ingres Memory Pre-Allocation (CA MDB r1.0.4)

To create an MDB, Ingres pre-allocates most of the resources that it requires. The Management Database often uses hundreds of megabytes of virtual memory and several hundred thousand handles. The exact amount of resources depends upon the number and sizes of the page caches used. The default (medium) MDB configuration uses approximately one (1) gigabyte of virtual memory and under Microsoft Windows 350,000 to 500,000 handles.

Ingres NFC Unicode (CA MDB r1.0.4)

For Ingres, the Management Database is created as a Unicode Normalization Form C (NFC) database. This allows Unicode data to be stored and manipulated by defining columns as Unicode data types (that is, nchar, nvarchar, and long nvarchar).

If a Unicode collation name is not specified, the Unicode database is created with the default collation sequence “udefault.” The Unicode database created in this manner (createdb -n) uses NFC for normalization of Unicode strings for processing and storage. NFC results from the canonical decomposition of a Unicode string, followed by the replacement of all decomposed sequences by primary composites, where possible.

For more information, see section A1.2 of http://www.unicode.org.

Ingres MDB Owner Name (CA MDB r1.0.4)

The owner of any MDB database (the MDB DBA) is *mdbadmin*. When connected to an Ingres MDB, this user name is returned by the following Ingres SQL statement:

SELECT dbmsinfo('dba')

Set Ingres Configuration Parameters (CA MDB r1.0.4)

To simplify the Ingres configuration for the CA products that use the Management Database, the *setupmdb* utility is provided to set the II_MDB_SIZE parameter to one of the following values:

- small
- medium
- large

When you install a CA product, one of these settings is established automatically. Do not make any changes unless instructed to do so by the CA product documentation or CA technical support.

If you install a second CA product that uses an existing Management Database, you can use the configuration parameter utility to increase the Ingres configuration setting. The utility only allows increasing the setting. For example, you may go from small to medium, but not from medium to small.

The table below provides guidelines for specifying the Ingres MDB requirements and the machine sizes required to support them.

Size	Connections	Number of CPUs (suggested minimum)	Memory (required)
Small	Up to 100	1	2 Gigabytes
Medium	Up to 500	1	2 Gigabytes
Large	Up to 1000	2	4 Gigabytes

Note: The Large data model is not supported on Microsoft Windows 2000 systems.

II_MDB_SIZE should be set to one of the values in the previous table. You must ensure that the server hosting the Management Database has sufficient memory and number of processors. The configuration utility must be run on the server where Ingres is installed and you must shut Ingres down before running the configuration utility. For Ingres shutdown instructions, see the *Ingres Getting Started* guide. The new setting takes effect the next time Ingres is started.

Example

The setupmdb utility may be found on Microsoft Windows in the <II_SYSTEM>\ingres\MDB subdirectory or on Linux in $II_SYSTEM/ingres/mdb.

```
setupmdb -II_MDB_SIZE=medium -reinitcfg
```

In this example, the reinitcfg flag re-initializes the II_MDB_SIZE configuration setting to medium. Since the configuration size can only be increased, this example can only change a small to medium. If the previous setting was medium or larger, this example has no effect.

Ingres configuration settings are stored in the <II_SYSTEM>\ingres\files\config.dat (Microsoft Windows) file or $II_SYSTEM/ingres/files/config.dat (Linux) file. A backup copy of the previous file is saved.

Define Ingres User IDs (CA MDB r1.0.4)

To connect to an Ingres MDB, you must possess a valid Ingres user ID and an associated valid user ID on the operating system that hosts the Ingres database server.

The Management Database is created with all of the pre-defined Ingres user IDs required by CA products to access the database. A user ID is not accessible until an operating system user ID with the same user name is created on the database server with the MDB. The operating system user IDs are not created until the product is installed. If you have one CA product accessing the Management Database, the database server will have defined only the operating system user IDs required for that product. Ingres user IDs are defined without passwords, because Ingres authenticates the operating system user definition before checking whether the user is defined to Ingres.

Some product installations create operating system users when the product is installed. In such cases, product installation should prompt for the password to be used in creating the OS user ID. Whenever possible, it should refrain from creating the password automatically, or (if created automatically) it should be a strong password.

Important! The Ingres user ID mdbadmin is the administrative owner of all MDB database objects. This user name should not have a corresponding operating system user ID; if it did, it would expose the Management Database to administration and access from remote sources. This user name should not be used by any product to access objects in the Management Database, as it is for CA internal use only. If such an operating system user is found, it should be investigated and ultimately removed.

Enable an Existing User to Administer the CA MDB for r1.0.4

An existing user may be given the ability to administer the MDB by impersonating the MDB administrative user (mdbadmin). This is accomplished by using the following SQL commands:

```
alter user username with default_privileges=(createdb,security,operator,maintain_locations,maintain_users) \g

grant db_admin on database mdb to user username \g
```

These statements need to be issued from the database server by the operating system user that installed the Ingres MDB. In addition, the command line for the SQL command is:

```
sql iidbdb –u$ingres
```

Note: The Ingres Visual DBA tool may be used to create users, alter users and add grants.

For information about creating users with administrative capabilities, see the *Ingres Database Administrator Guide.*

For information about the create user, alter user, and grant commands, see the *Ingres SQL Reference.*

Create a New User to Administer the CA MDB for r1.0.4

A new Ingres user may be created who is able to impersonate the MDB administrative user (mdbadmin). This impersonation is valid for all Ingres commands that support the specification of a user, typically those using a -u parameter. The Ingres SQL command is a good example of this.

Other commands such as the Ingres checkpoint command (ckpdb) also support the -u parameter.

To create an administrative user

1. Create an operating system user ID for the Ingres user to be defined. The OS user ID must conform to Ingres user naming conventions.
2. Create an Ingres user with the following default privileges (createdb, security, operator, maintain_locations, maintain_users). This may be done by using accessdb, the SQL command or Visual DBA.

   To create a user, use the following SQL commands:

   ```
   create user username with default privileges=(createdb,security,operator,maintain_locations,maintain_users)
   \g

   grant db_admin on database mdb to user username \g
   ```

   These statements need to be issued from the database server by the OS user that installed Ingres/MDB. In addition the command line for the SQL command is:

   ```
   sql iidbdb -u$ingres
   ```

   Note: The connection in this case is made to the Ingres installation database, not the MDB.

Access to Ingres MDB Objects (CA MDB r1.0.4)

When a new Ingres user ID is created, that user does not automatically have access to MDB objects. One method of granting access to a specific group of database objects is to assign a predefined user group. For example, a newly created Ingres user ID could be assigned to a Service Desk user group to limit access only to those database objects permitted to the Service Desk group and to no other objects.

If an Ingres user ID is created to access database objects for reporting purposes, the new user must be granted access to the appropriate database objects (tables and views); this is accomplished using Ingres grant statements.

Ingres MDB Case Sensitivity and Collations (CA MDB r1.0.4)

An Ingres MDB is created as a Unicode Normal Form C database and by default all columns are defined as case sensitive. Unicode Table columns that require case-insensitive sorting or searching may be assigned a case-insensitive collation sequence when the MDB is created.

How to Maintain an Ingres MDB (CA MDB r1.0.4)

The following activities are recommended for an Ingres-based Management Database:

- Monitor file size
- Monitor disk space
- Reorganize tables
- Collect statistics
- Reorganize the system catalog
- Back up the database

Monitoring file size and disk space, reorganizing system catalogs, and backing up the MDB database are generic maintenance activities. These activities need to occur regardless of the number or variety of CA products accessing an instance of the MDB. Table reorganization and statistics gathering are also necessary for performance purposes.

Command line utilities and SQL commands are used to perform these maintenance functions. Using a job scheduler, you can create command files that automate the housekeeping processes.

Note: Ingres provides a Visual DBA utility for administering Ingres databases, including MDBs. Most functions that are available from command line utilities and SQL statements are also available from the Ingres Visual DBA utility. For more information, see your Ingres documentation.

Remove the Ingres MDB (CA MDB r1.0.4)

The MDB should not be removed except in extreme cases.

Before removing an MDB, make sure the data stored in the database is no longer needed and that all products that were accessing the database have been uninstalled or configured to work with another MDB. Once an MDB is removed the data is lost.

Note: You should make a backup of the MDB before removing it, especially if more than one product has shared it.

To remove an Ingres MDB

1. On the Ingres server, destroy the database using the following command-line command:

   ```
   destroydb mdb –umdbadmin
   ```

2. Issue the following commands to remove MDB information from the Ingres configuration file:

   ```
   iiremres -v ii.%HOST%.mdb.mdb_dbname
   iiremres -v ii.%HOST%.mdb.mdb.version.major
   iiremres -v ii.%HOST%.mdb.mdb.version.minor
   iiremres -v ii.%HOST%.mdb.mdb.version.build
   iiremres -v ii.%HOST%.mdb.mdb.description
   iiremres -v ii.%HOST%.mdb.mdb.size
   ```

3. Remove the signature file. This file is located in the mdb subdirectory of the Ingres files directory. It is named:

   ```
   mdb.signature.txt
   ```

4. Remove OS users from the database server that were created by CA products for use with the MDB.

Chapter 5: Microsoft SQL Server MDB Considerations

This section contains the following topics:

Connections

Microsoft SQL Server supports setting a limit on the number of concurrent user connections.

Install Microsoft SQL Server to use unlimited concurrent user connections.

For best performance with more than 255 concurrent connections, increase the "maximum worker threads" setting.

Management Database Owner

The MDB is owned by the Microsoft Windows user who ran the MDB installation.

Tempdb

Microsoft SQL Server uses the tempdb database as a scratch area for MDB temporary tables, sorting, subqueries, and so forth.

The size of the tempdb database should be increased based on available disk space and expected usage. Microsoft SQL Server adjusts the size incrementally over time, but each adjustment causes a performance hit.

Partitions

Microsoft SQL Server supports partitioning of data files among multiple disk drives to ensure recovery and improved performance.

Important! You can not install the MDB into a Microsoft SQL Server instance where the default data files are created on a raw partition.

For best results:

- The Microsoft SQL Server MDB data and log directories should reside on separate disk drives for better recovery operations.
- The *tempdb* database should reside in a separate partition for improved performance.

Security

Microsoft SQL Server supports Microsoft Windows and Microsoft SQL Server authentication.

Microsoft SQL Server must be installed using mixed mode (both Microsoft SQL Server and Microsoft Windows authentication).

Configuration

Microsoft SQL Server supports various configuration options. The only configuration option set during the MDB installation is recursive triggers.

The Microsoft SQL Server configuration options should not be modified from their default values.

For More Information, see *Microsoft SQL Server Books Online* (**sp_configure**).

Environment

Microsoft SQL Server can be installed on the same server as other applications but consider the following:

- Installing Microsoft SQL Server on a dedicated computer may improve performance.
- Depending on the number of processors, memory, and disks on your computer, a dedicated computer may not be required. Network latency between the application and the SQL Server may in fact degrade application performance.

Case Sensitivity

The SQL Server instance that hosts the MDB can either be case sensitive or case insensitive. The MDB is created as a case insensitive database.

If the SQL Server instance is defined as case sensitive, all connection requests to the MDB must specify the database name (mdb) in lower case. If the database instance is defined as case insensitive it does not matter what case is used in the database name when connecting to the MDB.

SQL Server Collations

SQL Server collations determine how character data is compared and sorted. Collations are composed of a language designator and a sorting style. The two different sorting styles are binary and composite.

Binary is a sorting style that sorts data by binary value. Each character, upper and lower case, has a different binary value. The binary value of these characters, however, may not match the dictionary order for that language. Collations that use the binary sorting style contain the term BIN.

Composite sorting style is denoted by a term to indicate sensitivity for case (CI/CS), accent (AI/AS), kana (KI/KS), and width (WI/WS). Minimally, a composite sorting style term contains a sensitivity designation for case and for accent. For example, a composite sorting style term of CI_AS signifies case insensitivity and accent sensitivity. A term of **CI_AI_KI_WI** signifies case insensitivity, accent insensitivity, kana insensitivity, and width insensitivity.

SQL Server Installation and MDB Collations

The collation sequence for the MDB is set based on the default collation sequence of the SQL Instance. The default collation sequence for the instance is specified when SQL Server is installed. There are many collations that can be selected including those provided by SQL Server and those that come with Microsoft Windows.

Some of the available collation sequences do not support kana or width sensitivity. If such a collation is in effect when the MDB is created, case and accent sensitive values are set for columns but since kana and width sensitive characters are not available in the collation they are not set.

Setting the MDB Database Collation

The SQL Server MDB is created as a case insensitive database. This allows database object names (such as table names, view names, column names, and so forth) to be referred to in upper, lower or mixed case. The MDB is also created as kana insensitive and width insensitive. The accent sensitivity of the MDB is inherited from the SQL Server instance and therefore may vary by installation.

During MDB creation the language and accent sensitivity of the database server are determined by examining the database collation of the SQL Server instance. The collation established for the MDB is a combination of the language of the SQL Server instance, the accent sensitivity of the instance and a case insensitive setting. If the database server does not have an accent sensitivity setting (if it uses a binary sort), then the MDB will use accent sensitive as a part of its collation setting.

For example, if the SQL Server instance has a collation of Latin1_General_CS_AI, then the MDB will have a collation of Latin1_General_CI_AI. Similarly, if the instance has a collation of Korean_90_CS_AS_KS_WS, then the collation of the MDB will be Korean_90_CI_AS. Because MDB collations do not specify width or kana sensitivity, they are implicitly width and kana insensitive.

Setting MDB Column Collations

Data in MDB columns may have either case insensitive or case sensitive collations. Columns that are defined as case insensitive inherit the accent sensitivity of the SQL Server instance; they will also be kana and width insensitive. Columns that are defined as case sensitive are also accent sensitive.

Microsoft SQL Server Maintenance

Extended updates to the MDB will eventually cause reduced Microsoft SQL Server performance. Microsoft SQL Server should receive the following periodic maintenance:

- **Monitor indexes** - Fragmented indexes should be rebuilt using Microsoft SQL Server facilities.
- **Monitor data files** - Operating system data files should be de-fragmented as necessary.
- **Monitor the transaction log -** The size of the transaction log automatically expands and shrinks. For best performance:
  - Increase the initial size of the transaction log file based on estimated usage.
  - Use a specific growth increment amount such as 100 Megabytes, instead of a percentage increment.
  - Disable automatic transaction log file shrinkage and manually shrink the transaction log files as determined from your monitoring efforts.
  - De-fragment the transaction log file as necessary.
- **Monitoring disk space** - Disk space should be kept at least 20% free.
- **Backup and restore** - The following should be included in your backup and recovery plan:
  - Database and transaction log files should be scheduled for regular backups.
  - The master database should be included in the backup plan.
  - Multiple simultaneous backup devices may be used to improve performance.

For More Information, see *Microsoft SQL Server Books Online* (*Database Maintenance Plan Wizard, DBCC SHOWCONTIG*, and *DBCC_INDEXDEFRAG*).

Remove the SQL Server MDB

The MDB should not be removed except in extreme cases.

Before removing an MDB, make sure the data stored in the database is no longer needed and that all products that were accessing the database have been uninstalled or configured to work with another MDB. Once an MDB is removed the data is lost.

Note: You should make a backup of the MDB before removing it, especially if more than one product has shared it.

To remove a SQL Server Management Database

1. Using the SQL Server Enterprise manager, remove the MDB database.
2. Remove the signature file located in the folder that was specified in the MDB_TARGET_DIR parameter of the installation. The file is named:

   mdb.signature.txt

Chapter 6: Oracle Management Database Considerations

This section contains the following topics:

Storage Considerations

Oracle includes features that can maximize the efficiency of your file storage system. Consideration of these features when implementing your Oracle MDB may be beneficial.

Stripe And Mirror Everything (S.A.M.E.)

The Oracle 10g Automatic Storage Management (ASM) is based on this initiative. The main objective of S.A.M.E. offers maximum configuration flexibility and performance tuning by using large I/O and minimize random access by minimizing length of head movement.

For best I/O:

- Stripe using disk technology instead of the database.
- Mirror data.
- Use outer edge of disk for hot data.
- Place hot data on the outer edge of the disk and colder on the inner.

RAID

RAID 5 is powerful and inexpensive, but best avoided when configuring Oracle databases. An alternative to RAID 5 is RAID 0, commonly known as disk mirroring.

For optimal performance, the MDB should not be placed on RAID 5 storage subsystem. For best RAID performance, use hardware RAID 1+0. RAID 1+0 is preferred because of the heavy write penalty associated when using RAID 5 and RAID 1 (Mirroring) provide faster READ I/O. If RAID 1+0 (strip + mirror) is not available, then RAID 0+1 (Mirror of strip) would be another alternative.

File Systems

An Oracle MDB is comprised of several files which store user data, database meta data, and even information to recover from a failure. As such you should take into consideration the type of storage sub-system you will be locating the files on. There are the following options:

- File System - Creates database files managed by your operating system's file system.

  Note: File System is the choice recommended for MDB deployment.

- Automatic Storage Management (ASM) - ASM allows automatic stripe-and-mirror everywhere approach for automatically load balancing the disk I/O sub-system. Automatic Storage Management requires a separate instance to configure and manage disks groups.
- Raw Devices - This method is primarily used in Oracle Real Application Clusters environments.

Optimum Flexible Architecture (OFA)

The MDB follows the Oracle Optimum Flexible Architecture (OFA) for layout. An OFA implementation is the standard that defines how to set up Oracle Databases across all platforms in a consistent manner.

Log Files

Place Redo Logs and Archive Logs separate from data files to assist with performance and recovery purposes. Keep key table datafiles on different disks (or separate arrays) and controllers than the corresponding index datafiles if not using ASM.

Oracle Database Installation, Configuration, and Deployment

The Oracle user that is used to run the MDB installation must have database administrator privileges assigned.

Database Block Size

Larger block sizes result in more efficient I/O activity at potential expense of less efficient cache utilization and greater strain on the I/O system. An 8KB block size should be utilized in your MDB Oracle deployment. Selecting a block size other than 8KB requires advanced knowledge and should only be done when absolutely required.

Tablespaces

Installation allows you to use existing tablespaces or have new ones created for the MDB. If tablespaces are created, a tablespace for data and a tablespace for indexes will be created for optimal performance.

The system defined defaults for temporary and undo tablespaces are used.

Note: Since the MDB schema is predetermined, customers are advised not to alter or add additional tablespaces or tablespaces datafiles without consulting CA support or CA services.

Using Existing Tablespaces for the MDB

If you choose to use existing tablespaces for the MDB, then the tablespaces require a minimum of 200 megabytes of available disk space If this amount of space is not available the creation of the MDB will fail.

Existing tablespaces should have archive logging enabled so that online backups can be taken, archive audit log analysis may be performed, and data recovery options such as complete and point-in-time media are available.

Tablespaces Created by MDB Installation

When the MDB is installed, tablespaces are created with archive logging enabled. Tablespaces are created using the EXTENT MANAGEMENT LOCAL clause, enabling automate extent management. Tablespaces are also created with the SEGMENT SPACE MANAGEMENT AUTO clause enabling Automatic Segment Space Management.

System Temp Tablespace

The MDB uses a system temp tablespace to store temporary tables. At least 50 megabytes of space should be available for this purpose.

Database File Management

AUTOEXTEND is set to enable data files to grow. Proper monitoring of the operating system and disk space should be maintained to insure the space is always available for database growth.

Note: Products such as CA Unicenter Database Performance Management and CA Unicenter NSM may be utilized to automate and monitor such work.

Global Database Name, System Identifier (SID)

The Global Database Name is the full name of the database which uniquely identifies it from other Oracle databases. The global database name is of the form database_name.database_domain as in omdbprod1.us.ourcompany.com. The database name portion (omdbprod1) designates this as the first production Oracle MDB within the enterprise. The database domain portion (us.ourcompany.com) specifies the domain in which the MDB is located. Together database name and domain make up the Global Database Name.

Processes Parameter

The default value for this parameter is 150 which is acceptable for the majority of MDB deployments.

Character Sets

Choose from one of the following options:

- Choose Default if you only need the MDB to support the language currently used by the operating system.
- Choose Unicode (AL32UTF8) if your MDB needs to support multiple languages.

Connection Mode

When setting up connections, the Oracle MDB Dedicated Server Mode is recommended so there is a dedicated server process for each user process. Dedicated Server Mode is used when the number of total clients is small, or with persistent long-running requests to the database.

Redo Log Files

A database has a minimum of two redo log groups. A typical configuration has three redo log groups. Multiplexing of the redo log files is highly recommended.

MDB Database Users

When the MDB is created, a user named MDBADMIN is created. MDBADMIN is the user that owns the MDB database.

An Oracle user with SYSDBA privileges is required to create the MDB. This user's password should be specified when creating the MDB. If no user is specified when the MDB is created, the SYS user will be used and its password will be required. However, use of the SYS and SYSTEM accounts should be avoided when creating or making changes to the MDB schema.

The Oracle user you use to run the MDB installation must have the following database administrator privileges:

- The dba role (connect sys as sysdba; grant dba to installation_user;)
- The sysdba role (connect sys as sysdba; grant sysdba to installation_user;)
- The privilege to grant the mdbadmin user various privileges to the system tables and views (connect sys as sysdba; grant all privileges on "sys". TABLE_NAME" to installation_user with grant option;)
- The values to be assumed by TABLE_NAME are:

  COL$, DBA_CONSTRAINTS, DBA_CONS_COLUMNS, DBA_INDEXES, DBA_IND_COLUMNS, DBA_OBJECTS, DBA_OBJECT_TABLES, DBA_REGISTRY, DBA_TABLES, DBA_TABLESPACES, DBA_TAB_COLUMNS, DBA_TAB_PRIVS, DBA_VIEWS, DBMS_REGISTRY, EXPDEPACT$, EXPDEPOBJ$, EXPPKGACT$, EXPPKGOBJ$, KOPM$, OBJ$, TS$, USER$

Security Considerations

To ensure that your MDB is secure

- Change the passwords for all MDB administrative-level accounts routinely.
- Keep the MDB in a secure location. Make sure the database is in a directory owned by a secure user who has read-only permissions. Any users who have access to a saved database may potentially restore it on another system and view the data contained.

Archive Logging

Upon installation of the MDB, archive logging is enabled allowing you to:

- Perform online backups so you will have complete and point-in-time media data recovery options.
- Perform archive audit log analysis.

Note: You may utilize the Unicenter Database Analyzer for Oracle to comprehensively audit, recover, and analyze the Oracle log.

Undo and Database Backup

Changes made by transactions to the database are stored using undo data. When you install the database, undo tablespace and UNDO_MANAGEMENT = AUTO should be set. This enables undo functionality should the need arise in a recovery situation.

Remove the Oracle MDB

The MDB should not be removed except in extreme cases.

Before removing an MDB, make sure the data stored in the database is no longer needed and that all products that were accessing the database have been uninstalled or configured to work with another MDB. Once an MDB is removed the data is lost.

Note: You should make a backup of the MDB before removing it, especially if more than one product has shared it.

To remove an Oracle MDB

1. Drop the mdb_data and mdb_index tablespaces.

   For example:

   ```
   DROP TABLESPACE MDB_DATA INCLUDING CONTENTS AND DATAFILES
   CASCADE CONSTRAINTS;
   DROP TABLESPACE MDB_INDEX INCLUDING CONTENTS AND DATAFILES
   CASCADE CONSTRAINTS;
   ```

2. Drop the MDBADMIN user. For example:

   ```
   DROP USER MDBADMIN CASCADE
   ```

3. Drop the users that were created by the CA products that were using the MDB.
4. Drop the roles that were created by the MDB installation.

Index

T

U