Universal CMDB

Software Version: 10.31

Database Guide

Document Release Date: December 2016
Software Release Date: December 2016
Legal Notices

Warranty
The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Restricted Rights Legend

Copyright Notice
© 2002 - 2016 Hewlett Packard Enterprise Development LP

Trademark Notices
Adobe™ is a trademark of Adobe Systems Incorporated.
Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.
UNIX® is a registered trademark of The Open Group.

Documentation Updates
To check for recent updates or to verify that you are using the most recent edition of a document, go to: https://softwaresupport.hpe.com.
This site requires that you register for an HP Passport and to sign in. To register for an HP Passport ID, click Register on the HP Software Support site or click Create an Account on the HP Passport login page.
You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HPE sales representative for details.

Support
Visit the HPE Software Support site at: https://softwaresupport.hpe.com.
This website provides contact information and details about the products, services, and support that HPE Software offers.
HPE Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support website to:
- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HPE support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and to sign in. Many also require a support contract. To register for an HP Passport ID, click Register on the HPE Support site or click Create an Account on the HP Passport login page.
To find more information about access levels, go to: https://softwaresupport.hp.com/web/softwaresupport/access-levels.

HPSW Integrations Catalog accesses the new HPSW Integrations and Solutions Catalog website. This site enables you to explore HPE Product Solutions to meet your business needs, includes a full list of integrations between HPE Products, as well as a listing of ITIL Processes. The URL for this website is https://softwaresupport.hpe.com/km/KM01702731.
Contents

Part I: Introducing the Database Environment ........................................... 6
  Chapter 1: Introduction to Preparing the Database Environment .............. 7
    Databases in Use Overview .......................................................... 7
    Universal CMDB Sizing ............................................................... 8
    Hardware Requirements ............................................................. 8
    UCMDB Server Connection Retries to the Database ............................ 9
    How to Modify Composite Indexes .................................................. 9
    Large Capacity Planning for UCMDB .............................................. 10
      Large Capacity Planning Overview ........................................... 11
      Configuring the UCMDB Server for Large Capacity ....................... 11
      Configuring the Oracle Database for Large Capacity ..................... 12
      Configuring the Microsoft SQL Database for Large Capacity .......... 12
    System Capacity Test .................................................................... 13

Part II: Deploying the Microsoft SQL Server Database ............................. 15
  Chapter 2: Overview of Microsoft SQL Server Deployment .................... 16
    About Microsoft SQL Server Deployment ......................................... 16
    System Requirements .................................................................... 16
    Installation Prerequisites ............................................................. 17
    Checklist for Support and Certification ........................................... 17
  Chapter 3: Manually Creating the Microsoft SQL Server Databases .......... 19
    Overview of the Microsoft SQL Server Databases ............................ 19
    Manually Creating the Universal CMDB Microsoft SQL Server Database . 19
  Chapter 4: Creating and Configuring the Microsoft SQL Server Database ... 21
    Creating the Database .................................................................... 21
    Configuring the Database ............................................................ 26
    Monitoring the Fragmentation Indexes ............................................ 30
  Chapter 5: Using Windows Authentication to Access the Microsoft SQL
    Server Database ........................................................................... 32
    Enabling Universal CMDB to Work with Windows Authentication ...... 32

Part III: Deploying the Oracle Server Database .................................... 35
  Chapter 6: Overview of Oracle Server Deployment ................................ 36
  About Oracle Server Deployment ..................................................... 36
System Requirements ................................................................. 37
Chapter 7: Manually Creating the Oracle Server Database Schemas .... 38
  Overview of the Oracle Database Schema .................................. 38
  Prerequisites ............................................................................ 38
  Creating Schema Default and Temporary Tablespaces ................. 39
  Creating Schema User Permissions ........................................... 39
  Creating Schemas and Schema Objects ...................................... 40
Chapter 8: Oracle Server Configuration and Sizing Guidelines ............ 41
  Oracle Parameter Settings .......................................................... 41
  Oracle Tablespaces .................................................................. 44
    Locally Managed Tablespaces .................................................. 45
  Oracle Tablespace Settings ....................................................... 45
  Using RAID Configuration ........................................................ 48
  Composite Indexes vs. Non Composite Indexes ........................... 50
  Checklist for Universal CMDB Support and Certification .............. 50
Chapter 9: Real Application Cluster Support ...................................... 53
  About Oracle Real Application Cluster (RAC) ............................... 53
  Single Client Access Name ...................................................... 54
  Client Side Configuration for Oracle RAC ................................... 54
  Create/Connect to an Universal CMDB Database ......................... 57
Part IV: Deploying the PostgreSQL Server Database ............................ 59
Chapter 10: Overview of PostgreSQL Server Deployment ..................... 60
  About PostgreSQL Server Deployment ...................................... 60
  System Requirements ................................................................ 60
  Overview of the PostgreSQL Server Databases ............................ 61
  PostgreSQL Deployment Options ............................................. 61
  Initialize the PostgreSQL Server .............................................. 62
Chapter 11: PostgreSQL Server Configuration .................................... 64
  PostgreSQL Parameter Settings ................................................. 64
  Critical PostgreSQL Server Files ............................................. 65
  How to Configure PostgreSQL on Probe to Accept Local Connections Only ................................................................. 66
    How to Validate If the PostgreSQL Instance Is Listening Only to Localhost? ...................................................... 67
  How to Configure PostgreSQL Log Files Rotation by Size ............ 69
  Always Exclude PostgreSQL Data Folder from Anti-Virus Scan ....... 70
Part I: Introducing the Database Environment
Chapter 1: Introduction to Preparing the Database Environment

This chapter includes:

- Databases in Use Overview ................................................................. 7
- Universal CMDB Sizing ........................................................................ 8
- Hardware Requirements ........................................................................ 8
- UCMDB Server Connection Retries to the Database .............................. 9
- How to Modify Composite Indexes ....................................................... 9
- Large Capacity Planning for UCMDB .................................................... 10

Databases in Use Overview

To work with Universal CMDB, you must set up the CMDB database. The CMDB database is used for storage of configuration information that is gathered from the various Universal CMDB and third-party applications and tools. This information is used when building Universal CMDB views.

You can set up the Universal CMDB databases on a Microsoft SQL Server, on an Oracle Server, or on a PostgreSQL Server, depending on the type of database server used in your organization.

If you are working with a Microsoft SQL Server database, refer to "Deploying the Microsoft SQL Server Database" on page 15.

If you are working with an Oracle Server database, refer to "Deploying the Oracle Server Database" on page 35.

If you are working with a PostgreSQL Server database, refer to "Deploying the PostgreSQL Server Database" on page 59.

Note:

- It is strongly recommended to host database server (Oracle, Microsoft SQL, or PostgreSQL) on a physical machine, and it should be an independent server without other applications running on it.

  Also, if the database server machine is a virtual machine, the resource MUST be dedicated for the database server.
- Database servers must be set to the same time zone, daylight savings settings and time as the Universal CMDB servers.
- For details on working in a non-English language Universal CMDB environment, see the section on non-English locales in the interactive HPE Universal CMDB Deployment Guide.
- The UCMDB server(s) should be located in the same LAN with the database servers (without a proxy and firewalls between them). Otherwise, your system's performance may be impacted.

### Universal CMDB Sizing

Universal CMDB database configuration requirements are dependent on the amount of data, as well as on the runtime load, generated by Universal CMDB. A small Universal CMDB deployment requires a database that can hold up to 1.5 million CMDB objects and links. A large Universal CMDB deployment requires a database containing 1.5 million or more CMDB objects and links.

For more details, see the HPE Universal CMDB Sizing Guide.

### Hardware Requirements

The following table describes the minimum hardware (CPU and memory) requirements recommended for the Universal CMDB Oracle, Microsoft SQL, or PostgreSQL database server:

<table>
<thead>
<tr>
<th>Deployment</th>
<th>Cls and Relationships</th>
<th>Number of Processors</th>
<th>Physical Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>&lt; 2 million</td>
<td>Minimum 1 dual core</td>
<td>8 GB RAM</td>
</tr>
<tr>
<td>Standard</td>
<td>2 - 10 million</td>
<td>Minimum 2 dual core or 1 quad core</td>
<td>8 GB RAM</td>
</tr>
<tr>
<td>Large</td>
<td>10 - 40 million</td>
<td>Minimum 2 dual core or 1 quad core</td>
<td>16 GB RAM</td>
</tr>
<tr>
<td></td>
<td>40 - 60 million</td>
<td>2 x 4 cores</td>
<td>32 GB RAM</td>
</tr>
<tr>
<td></td>
<td>60 - 120 million</td>
<td>2 x 8 cores</td>
<td>64 GB RAM</td>
</tr>
<tr>
<td></td>
<td>&gt; 120 million</td>
<td>2 x 8 cores</td>
<td>128 GB RAM</td>
</tr>
</tbody>
</table>

**Note:** It is strongly recommended to host database server (Oracle, Microsoft SQL, or PostgreSQL) on a **physical machine**, and it should be an independent server without other applications running on it.

Also, if the database server machine is a virtual machine, the resource MUST be dedicated for the database server.
UCMDB Server Connection Retries to the Database

By default UCMDB server tries to connect to the database for 10 times when there is a network problem.

If you need to increase the number of the retries, do the following:

1. Open the `<UCMDB_Server_Home>\bin\wapper.conf` file using a text editor.
2. Locate the following entry in the `wapper.conf` file and increase the number of startup retries to a desired value:

   ```
   # Number of startup retries before stopping the service
   wrapper.max_failed_invocations=10
   ```

3. Save the file.
4. Restart the UCMDB server for the change to take affect.

   In case of HA, restart the UCMDB server cluster.

How to Modify Composite Indexes

A composite index contains multiple key columns. You can invoke the `modifyCompositeIndexes` JMX method to add or remove the `CMDB_ID` column as a key column in the indexes for the specified class. The method then modifies the indexes according to the specified parameters.

**Note:**
- This method only works for a Microsoft SQL or Oracle database.
- When you perform a fresh install of the UCMDB 10.30 (or later), by default there are no composite indexes with the `CMDB_ID` as a key column on Oracle database (ROOT tables), and on Microsoft SQL databases (ROOT and CDM tables). To create the composite indexes with the `CMDB_ID` as a key column, follow the steps described in this procedure.

It is highly recommended that you follow these best practices for running the `modifyCompositeIndexes` method:

- Run the method when the probes are stopped and the database is not heavily used.
- Create a schema dump before running the method.
For environments that execute heavy data-in operations, in order to speed up the INSERT statements, it is recommended to transform the database from composite index to non composite index by invoking the modifyCompositeIndexes method with composite indexes setting to false.

To modify composite indexes:

1. On the UCMDB server, launch the Web browser and enter the following address:
   https://localhost:8443/jmx-console
   You may have to log in with a user name and password.

2. Locate UCMDB:service=Dal Services and click the link to open the JMX MBean View, and then locate the modifyCompositeIndexes method.

3. Specify the following parameters of the method:
   - customerId: Specify the Customer ID (leave empty for the default customer)
   - className: Specify the class name (allowed values are root and all)
   - composite indexes: Specify if the method rebuilds the indexes with CMDB_ID as a key column
     - True: CMDB_ID is a key column
     - False: CMDB_ID is not a key column

4. Click Invoke.

   **Note:** If any problem occurs, the operation rolls back and causes no change in the database.

Large Capacity Planning for UCMDB

This section includes:

- Large Capacity Planning Overview
- Configuring the UCMDB Server for Large Capacity
- Configuring the Oracle Database for Large Capacity
- Configuring the Microsoft SQL Database for Large Capacity
- System Capacity Test
Large Capacity Planning Overview

Using the default configuration, HPE Universal CMDB can work with a deployment of more than 25 million CIs and relationships. To work with a larger deployment, you must implement the following configuration:

Depending in the number of CIs and relationships, increase the CMDB heap as follows:

<table>
<thead>
<tr>
<th># CIs and Relationships</th>
<th>Heap Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>≤ 40 million</td>
<td>12 GB</td>
</tr>
<tr>
<td>40 million – 60 million</td>
<td>16 GB</td>
</tr>
<tr>
<td>60 million – 125 million</td>
<td>24 GB</td>
</tr>
<tr>
<td>&gt; 125 million</td>
<td>55 GB</td>
</tr>
</tbody>
</table>

- For capacity planning requirements, see the *HPE Universal CMDB Support Matrix* document.
- For details about the changes you must make to the system configuration to support this capacity, see "Configuring the UCMDB Server for Large Capacity" below.
- For details on how to improve performance, see "Configuring the Oracle Database for Large Capacity" and "Configuring the Microsoft SQL Database for Large Capacity" below.
- For details about the setup used for capacity testing and performance results, see "System Capacity Test" below.

Configuring the UCMDB Server for Large Capacity

For the system to support the desired number of CIs and relationships, update the following parameters on the UCMDB Server:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>CIs and Relationships (million)</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>≤ 40</td>
<td>40 – 60</td>
</tr>
<tr>
<td>wrapper.java.initmemory</td>
<td>1024</td>
<td>2048</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Default</td>
<td>CIs and Relationships (million)</td>
<td>Location</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>---------</td>
<td>---------------------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>≤ 40</td>
<td>40 – 60</td>
</tr>
<tr>
<td>wrapper.java.maxmemory</td>
<td>4096</td>
<td>8192</td>
<td>16384</td>
</tr>
<tr>
<td></td>
<td></td>
<td>wrapper-platform.conf</td>
<td></td>
</tr>
<tr>
<td>dal.object.condition.max.result.size</td>
<td>2000000</td>
<td>50000000</td>
<td>60000000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dal.use.memory.instead.temp.table.high.threshold.oracle</td>
<td>600000</td>
<td>6000000</td>
<td>6000000</td>
</tr>
<tr>
<td>dal.joinf.max.result.size</td>
<td>400000</td>
<td>4000000</td>
<td>4000000</td>
</tr>
</tbody>
</table>

### Configuring the Oracle Database for Large Capacity

When working on a system containing more than 40 million objects and relationships, you can improve performance by increasing the Oracle SGA and PGA to the following suggested sizes:

<table>
<thead>
<tr>
<th>CIs and Relationships</th>
<th>SGA</th>
<th>PGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>40 million – 60 million</td>
<td>22 GB</td>
<td>6 GB</td>
</tr>
<tr>
<td>60 million – 120 million</td>
<td>42 GB</td>
<td>14 GB</td>
</tr>
<tr>
<td>&gt; 120 million</td>
<td>88 GB</td>
<td>24 GB</td>
</tr>
</tbody>
</table>

This improves the performance of both the TQL calculation for several types of TQL queries, as well as for data-in operations performed on the system.

### Configuring the Microsoft SQL Database for Large Capacity

When working on a system containing more than 40 million objects and relationships, you can improve performance by increasing the Microsoft SQL Server Memory to the following suggested sizes:
## CIs and Relationships

<table>
<thead>
<tr>
<th>CIs and Relationships</th>
<th>Microsoft SQL Server Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>40 million – 60 million</td>
<td>28 GB</td>
</tr>
<tr>
<td>&gt; 60 million</td>
<td>56 GB</td>
</tr>
</tbody>
</table>

### System Capacity Test

#### Setup

The system capacity test is conducted for Microsoft SQL Server (with 125 million CIs and relationships) and Oracle Database (with 200 million CIs and relationships) separately, by using the following hardware configurations.

**Microsoft SQL Server**

<table>
<thead>
<tr>
<th>Role</th>
<th>CPU</th>
<th>Memory</th>
<th>OS + 3rd Party SW</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMDB</td>
<td>2 x 4-cores @ 2.67 GHz</td>
<td>32 GB</td>
<td>Microsoft Windows Server 2008 R2 Enterprise Edition x64 SP1</td>
</tr>
</tbody>
</table>
| Database| 2 x 8-cores @ 2.93 GHz       | 64 GB  | • Microsoft Windows Server 2008 R2 Enterprise Edition x64 SP1  
|         |                              |        | • Microsoft SQL Server 2014 - 12.0.2000.8 (x64)          |

**Oracle Database**

<table>
<thead>
<tr>
<th>Role</th>
<th>CPU</th>
<th>Memory</th>
<th>OS + 3rd Party SW</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMDB</td>
<td>2 x E5-2630V3 @ 2.40GHz</td>
<td>32 GB</td>
<td>Red Hat Enterprise Linux 7.2</td>
</tr>
</tbody>
</table>
| Database| 2 x E5-2630V3 @ 2.40GHz      | 97 GB  | • Oracle Linux 7.2                                      
|         |                              |        | • Oracle Database 12c Enterprise Edition x64 - 12.1.0.2 |

The following business flows were tested as part of the system test:

- **TQL Calculation**

  TQLs were divided into sub groups according to the result size (<100, <1000, and <10000), according to the data set that the TQL retrieves, and according to the TQL configuration:

  - Like Condition
  - Like Ignore case
Different number of hierarchies in the TQL results (2-5)
- Compound
- Sub-graph

**Data-in**
The data-in scenario in the system test included insertion, updates, and deletion.

**Enrichments**
Enrichment scenarios included insert, update, and delete.

**Results**
Following the load test in the scenario that includes query execution (2 days), data-in (7 days for Oracle and 10 days for MS SQL), and enrichment execution, the following results were achieved:

- The system was stable throughout the run. No restarts, memory leaks, or any other degradation over time was observed.
- System performance was acceptable.
Part II: Deploying the Microsoft SQL Server Database
Chapter 2: Overview of Microsoft SQL Server Deployment

This chapter includes:

- About Microsoft SQL Server Deployment .............................................................. 16
- System Requirements .............................................................................................. 16
- Installation Prerequisites ....................................................................................... 17
- Checklist for Support and Certification .................................................................. 17

About Microsoft SQL Server Deployment

To deploy Microsoft SQL Server for use with Universal CMDB, you must perform the following procedures:

- **Install and configure Microsoft SQL Server.**
  
  For details on installing and configuring Microsoft SQL Server, see the Microsoft SQL Server documentation for your version of Microsoft SQL Server.

- **Create databases on Microsoft SQL Server.**
  
  You can create Universal CMDB databases manually or you can use the UCMDB Server Configuration Wizard to create the databases ([Start > UCMDB > Start HP Universal CMDB Server Configuration Wizard](start-ucmdb-start-hp-universal-cmdb-server-configuration-wizard)). For details on manually creating databases for Universal CMDB, see "Manually Creating the Microsoft SQL Server Databases" on page 19.

For details on creating a Microsoft SQL database, see "Creating and Configuring the Microsoft SQL Server Database" on page 21.

System Requirements

This section describes the system requirements for working with Microsoft SQL Server in conjunction with Universal CMDB.

This section includes the following topics:

- "Hardware Requirements" on the next page
- "Software Requirements" on the next page
Hardware Requirements

For Universal CMDB hardware sizing guidelines, see “Hardware Requirements” on page 8. For Microsoft SQL Server hardware requirements, refer to the installation guide for your Microsoft SQL Server release for your operating system.

Software Requirements

For details on supported versions of Microsoft SQL Server, see the Server Database Requirements section of the HPE Universal CMDB Support Matrix.

Installation Prerequisites

The following prerequisite must be met before installing Microsoft SQL Server:

- If you want to install Universal CMDB using the installation wizard, you must provide a user account that has database creator privileges. If you are going to create the database manually, supply Universal CMDB with a user account that is a part of db_datareader, db_datawriter and db_ddladmin in the Universal CMDB database.

  **Note:** When working with Microsoft SQL Server, the CMDB collation and the SQL Server collation must be the same.

Checklist for Support and Certification

Information is provided in this section for both supported and certified Microsoft SQL Server options.

The certified options are recommended for working with Universal CMDB. Certified options are rigorously tested by HPE quality assurance personnel. Supported options are those options for which HPE quality assurance personnel have successfully performed basic tests.

<table>
<thead>
<tr>
<th>Subject</th>
<th>Microsoft SQL Server</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Supported</td>
</tr>
<tr>
<td>Instances</td>
<td>Default, Single</td>
</tr>
<tr>
<td>Authentication Mode</td>
<td>Mixed, Windows</td>
</tr>
<tr>
<td>Collation</td>
<td>Case-insensitive. Universal</td>
</tr>
<tr>
<td>Subject</td>
<td>Supported</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CMDB</td>
<td>CMDB does not support binary sort order and case sensitivity. Only case-insensitive order with a combination of accent, kana, or width settings is supported.</td>
</tr>
<tr>
<td></td>
<td>Client: TCP/IP and Named Pipes</td>
</tr>
<tr>
<td>Server Configuration</td>
<td>Defaults, unless instructed otherwise</td>
</tr>
<tr>
<td>Options</td>
<td></td>
</tr>
<tr>
<td>Data File Properties</td>
<td>Manual file growth, or FILEGROWTH less than or equal to 100 MB</td>
</tr>
<tr>
<td>Collation Database</td>
<td>Server default</td>
</tr>
<tr>
<td>Property</td>
<td></td>
</tr>
<tr>
<td>Database Options</td>
<td>Defaults, unless instructed otherwise</td>
</tr>
<tr>
<td>Recovery Model</td>
<td>Any</td>
</tr>
</tbody>
</table>

**Note:** For Turkish, the recommended collation is **Turkish_100_CI_AS**.
Chapter 3: Manually Creating the Microsoft SQL Server Databases

This chapter includes:

Overview of the Microsoft SQL Server Databases ..................................................... 19
Manually Creating the Universal CMDB Microsoft SQL Server Database .................. 19

Overview of the Microsoft SQL Server Databases

Universal CMDB uses the configuration management database (CMDB) for its persistency.

During the Universal CMDB setup, the new database can be set up automatically by the setup procedure, or an existing database can be used. An existing database can either be created manually in advance (for example, due to organization security restrictions), or can be created by a previous installation of the same release of Universal CMDB.

For details about installing the Universal CMDB server, see the interactive HPE Universal CMDB Deployment Guide.

Manually Creating the Universal CMDB Microsoft SQL Server Database

Before creating the database, you must install the Microsoft SQL Database Server according to the Microsoft SQL Server documentation.

If the Universal CMDB database is manually created prior to running the HPE Universal CMDB setup, during the setup procedure, you select the option to use an existing database and enter the relevant data about the existing database for which you are prompted. The data you enter is written to the configuration files that are used by the JDBC driver to connect to the existing database.

Creating the Database

The database administrator should create a Microsoft SQL Server database for the CMDB data. No scripts are necessary to create the objects to populate this database.

To create a database, you must have CREATE DATABASE permissions. To connect to an existing database, the login account with which you are connecting must be mapped to dbo in the database.
Note: Members of the sysadmin server role automatically have CREATE DATABASE permissions, and are also mapped to dbo in all databases. A database owner is automatically mapped to dbo in the database.

For details on creating the database manually, see "Creating and Configuring the Microsoft SQL Server Database" on page 21.

Note: When the Universal CMDB setup procedure automatically creates the database, it uses two file groups for each database; one to hold the system tables and one for the application data.
Chapter 4: Creating and Configuring the Microsoft SQL Server Database

This chapter includes:

- Creating the Database .......................................................... 21
- Configuring the Database ...................................................... 26
- Monitoring the Fragmentation Indexes ...................................... 30

Creating the Database

This section includes the following topics:

"Database Permissions" below

"Database File Layout" on the next page

"System Databases" on page 25

Database Permissions

To create a database, you must have CREATE DATABASE permissions. To grant CREATE DATABASE permissions to a user, the user’s login must first be mapped to a database user in the master database.

**Note:** Universal CMDB login accounts should be mapped to dbo in the database. Members of the sysadmin server role automatically have CREATE DATABASE permissions, and are also mapped to dbo in all databases. A database owner is automatically mapped to dbo in the database.

To check whether a user has CREATE DATABASE permissions, log in to Management Studio with the login account of the user whose permissions you want to check, and run the following:

```
USE master
IF PERMISSIONS() & 1 = 1
PRINT 'User has CREATE DATABASE permissions.'
```

To check whether a user has enough permissions in the database, log in to Management Studio with the login account of the user whose mapping you want to check. Change the database context to the required database, and run the following command:
```sql
SELECT CASE WHEN IS_MEMBER ('db_owner')=1
    OR IS_SRVROLEMEMBER ('sysadmin')=1
    OR (IS_MEMBER ('db_ddladmin') = 1 AND
        IS_MEMBER ('db_datareader')=1 AND
        IS_MEMBER ('db_datawriter')=1 AND
        IS_MEMBER ('dbdenydatareader')=0 AND
        IS_MEMBER ('dbdenydatawriter')=0 )
     THEN 'User has enough permissions'
 ELSE 'User does not have enough permissions'
END
```

Database File Layout

When you create a database, it must consist of at least one data file (with an .mdf extension) and one transaction log file (with an .ldf extension). You can optionally create additional data files (.ndf), as well as additional log files (.ldf).

To enhance performance, you may want to create several data files. Microsoft SQL Server stripes the data among the data files, so that if you do not have RAID controllers that stripe your data, you can spread the data files over several regular physical disks and, in this way, have the data striped. The log, however, is read sequentially, so that there is no performance gain in adding more log files. An additional log file should be created on a different disk when your existing log is out of disk space.

Data and Log Placement

**Caution:**

- It is recommended not to place data or log files on the same disk that stores the page (swap) file.
- It is recommended that you place the data and log files on separate disk subsystems.
- **Log files.** Changes are not flushed to the database until they are written to the log, and the log architecture dictates serial writes, so it is advisable that there be as little interference as possible with the log activity. It is usually sufficient to place the log on a RAID 1 system because of the serial writes to the log. If you have processes reading from the log (for example, triggers accessing the inserted and deleted views which are formed from the log records or transactional replication), or several log files for different databases, consider placing the log file(s) on a RAID 0+1 (striped mirror) system.
- **Data files.** Data files should be placed on a RAID 0+1 system for optimal performance.
File and Database Properties

When you create a database you can specify the following properties for each file (.mdf, .ndf, .ldf):

- **NAME.** The logical file name which you can use later when you want to alter one of the properties.

- **FILENAME.** The physical file path and name. Make sure the destination directory is not compressed (right-click the directory in Windows Explorer, select Advanced, and verify that the Compression check box is not selected).

- **SIZE.** The initial file size.

- **MAXSIZE.** Determines the maximum size to which the file can grow. If this argument is omitted, or if you specify Unlimited, the file can grow until the disk is full.

- **FILEGROWTH.** The automatic growth increment of the file. This argument can be specified as either a percentage of the existing file size, or as a fixed size.

An autogrowth operation invoked by a modification sent by a client that timed out cannot be completed successfully. This means that the next time a client sends a modification, the autogrowth process starts at the beginning and may also time out.

To avoid this problem, it is recommended that you either expand the files manually every time the database nearly reaches full capacity (for example, 20 percent free), or set the growth increment to a fixed size that takes less time to be allocated than the client’s timeout setting. Using a small growth increment is not recommended because it increases file system fragmentation. On the other hand, if you use a very large increment, modifications sent by clients might incur connection timeouts while waiting for the automatic expansion to finish. For large databases, a percentage growth increment can lead to exponential growth of the database and should be avoided.

For more information on this problem, refer to Microsoft Knowledge Base Article - 305635 (http://support.microsoft.com/kb/305635).

- **ALLOW_SNAPSHOT_ISOLATION** and **READ_COMMITTED_SNAPSHOT.** Set the database properties ON for **ALLOW_SNAPSHOT_ISOLATION** and **READ_COMMITTED_SNAPSHOT** with the following script:

```
ALTER DATABASE [{dbName}] SET ALLOW_SNAPSHOT_ISOLATION ON
ALTER DATABASE [{dbName}] SET SINGLE_USER WITH ROLLBACK IMMEDIATE
ALTER DATABASE [{dbName}] SET READ_COMMITTED_SNAPSHOT ON
ALTER DATABASE [{dbName}] SET MULTI_USER
```

**Note:**

Database Guide
Chapter 4: Creating and Configuring the Microsoft SQL Server Database

HPE Universal CMDB (10.31) Page 23 of 71
The above database properties MUST be set for UCMDB database even if the database is not on AlwaysOn.

If the database is set on AlwaysOn, the above database properties must be set before you add the database on AlwaysOn.

The tempdb Database Settings

The frequent expansion of the tempdb system database can affect the database’s performance, especially in large Microsoft SQL Server installations. The size of the tempdb, therefore, should be large enough to avoid the need for early expansion. Its growth increment should be large enough to avoid fragmentation, yet not too large to expand in a reasonable amount of time. Create the tempdb with a minimum, initial size of 1 GB and with a growth increment of 50 MB. The tempdb database should be striped across several disks, ideally on a RAID 0+1 controller. It is recommended to move the tempdb database to its own set of disks.

To ensure that there is enough disk space for the tempdb to grow during times of heavy usage (for example, when aggregating or sorting data), it is recommended that you leave at least 20 GB free disk space on the drive where the tempdb is located.

File Groups

File groups are logical groupings of data files. Each of the following objects can be placed in its own file group unit:

- A table’s data
- A table’s large objects (text, ntext, image columns)
- An index

Data is inserted proportionally into all files belonging to the file group in which the object is stored, according to the amount of free space in each file. The .mdf file is placed in a file group called PRIMARY, which is marked as Default when the database is created (the default file group for objects when no file group is specified). If you do not place other data files (.ndf files) in their own file groups, they are also placed in the PRIMARY file group. Note that you can change the Default file group later on.

File groups can be used for performance tuning or maintenance. For details, see Microsoft SQL Server Books Online at http://www.microsoft.com/downloads.

Following is an example that demonstrates how to use file groups for maintenance:

- **Partial Restoring.** Microsoft SQL Server does not support the restoration of a single table. Even if you place a single table in a file group, you cannot restore a file group to a point in time earlier than
the rest of the data. Instead, you must apply all log file backups in order to synchronize the file group with the rest of the data. Microsoft SQL Server supports partial restoration to a database with a different name. A partial restoration allows you to restore a single file group, and supports point-in-time restoration. However, you must restore the PRIMARY file group because it contains the SYSTEM tables.

To be able to restore a single table to a point in time if a logical error occurs, you need to design the file groups in your database as follows:

- Ensure that the .mdf file is the only file in the PRIMARY file group.
- Place each large table in its own file group.
- Place all small tables in a separate file group.

System Databases

The following system databases are especially important for the smooth performance of Microsoft SQL Server:

- **tempdb.** Numerous Microsoft SQL Server activities—such as creating local and global temporary tables, creating work tables behind the scenes to spool intermediate query execution results, and sorting—implicitly or explicitly use the tempdb system database.

  If your system is not configured properly, the tempdb database can become a performance bottleneck, so it is very important to determine the tempdb database’s original size correctly.

  For more information on setting database sizes, see "The tempdb Database Settings" on the previous page.

  To move tempdb’s files, use the ALTER DATABASE tempdb MODIFY FILE command, and restart Microsoft SQL Server.

- **master, msdb, model.** These databases, although crucial for the operation of Microsoft SQL Server, are smaller than tempdb because they store only meta data.

  It is strongly recommended to use a fault tolerant disk—ideally, RAID 1—for these databases.

  **Note:** For Universal CMDB certification, place system databases on fault tolerant disks. It is recommended to use RAID 1 disks.

To check the database’s properties, run the following:

```
EXEC sp_helpdb <database name>
```
Configuring the Database

Once you have created the necessary databases, you can add new files to the databases, change some of the existing database file properties, and set the database configuration options appropriately.

This section includes the following topics:

"Database File Configuration"
"Database Configuration Options"
"Composite Indexes vs. Non Composite Indexes"

Database File Configuration

You can change certain database file properties, as well as add or drop files using either of the following methods:

- use the Properties dialog box in Management Studio
- use the ALTER DATABASE command (for details, see Microsoft SQL Server Books Online at http://www.microsoft.com/downloads)

Adding Files

Data files can be added to an existing file group in a database, or to a new file group. There are no special restrictions or requirements.

Dropping Files

To drop a file, you must first empty it using the DBCC SHRINKFILE command’s EMPTYFILE option, which transmits the file data to all the other files in the file group. Once you empty the file, you can use the ALTER DATABASE <database name> DROP FILE command to drop it.

Changing File Properties

You can change the size-related properties for all databases, as well as the filename property for the tempdb database (this takes effect after you restart Microsoft SQL Server). The SIZE, MAXSIZE, and FILEGROWTH properties can be changed using the ALTER DATABASE tempdb MODIFY FILE command. Note that the SIZE property can only be enlarged.

To shrink the file, use the DBCC SHRINKFILE command. For details and recommendations concerning file properties, see "Creating the Database" on page 21.
Database Configuration Options

Each database contains a set of configurable options that determine its behavior. You can view or change the database options using any one of the following utilities:

- the Options tab in the Management Studio’s Properties dialog box
- the EXEC sp_dboptions stored procedure
- the ALTER DATABASE <database name> SET command

**Note:** Not all of the database configuration options are available in this dialog box.

The following table lists, in alphabetical order, the default configuration options, as well as the configuration settings required for Universal CMDB certification:

<table>
<thead>
<tr>
<th>Configuration Option</th>
<th>Description</th>
<th>Default</th>
<th>Universal CMDB Certification in Microsoft SQL Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANSI NULL default (see note below)</td>
<td>Specifies whether the database columns are defined as NULL or NOT NULL, by default</td>
<td>Not set</td>
<td>Not set</td>
</tr>
<tr>
<td>ANSI PADDING</td>
<td>Controls the way the column stores values shorter than the defined size of the column and the way the column stores values that have trailing blanks in char, varchar, binary, and varbinary data.</td>
<td>OFF</td>
<td>ON</td>
</tr>
<tr>
<td>Auto close</td>
<td>Specifies whether the database shuts down after its resources are freed and all users exit</td>
<td>Not set</td>
<td>Not set</td>
</tr>
</tbody>
</table>

**Note:** If set, it may take a long time for the database to allocate resources every time a user...
<table>
<thead>
<tr>
<th>Configuration Option</th>
<th>Description</th>
<th>Default</th>
<th>Universal CMDB Certification in Microsoft SQL Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto create statistics</td>
<td>Specifies whether missing statistics required by a query for optimization are built automatically during optimization</td>
<td>Set</td>
<td>Set</td>
</tr>
<tr>
<td>Auto shrink</td>
<td>Specifies whether the database is automatically shrunk every hour, leaving 25% of free space</td>
<td>Not set</td>
<td>Not set</td>
</tr>
<tr>
<td>Auto update statistics</td>
<td>Specifies whether out-of-date statistics required by a query for optimization are built automatically during optimization</td>
<td>Set</td>
<td>Set</td>
</tr>
<tr>
<td>Compatibility level</td>
<td>The version of Microsoft SQL Server that the database appears to be (for the application)</td>
<td>The same version as the release installed</td>
<td>The same version as the release installed</td>
</tr>
<tr>
<td>Read only</td>
<td>Database is read only</td>
<td>Not set (READ_WRITE)</td>
<td>READ_WRITE</td>
</tr>
<tr>
<td>Recovery</td>
<td>The database recovery model determines the recovery capabilities by controlling the amount of bulk operation logging (such as Select into, Bulk, Insert, Create index, LOB manipulation). The higher the recovery model, the</td>
<td>Full</td>
<td>Full (unless you are certain that the lower recovery capabilities are sufficient for your system)</td>
</tr>
<tr>
<td>Configuration Option</td>
<td>Description</td>
<td>Default</td>
<td></td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td><strong>Universal CMDB Certification in Microsoft SQL Server</strong></td>
<td>higher the recovery capabilities. However, the amount of logging also increases, which may affect performance.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Recursive triggers</td>
<td>Specifies whether recursive triggers are supported</td>
<td>Not set</td>
<td>Not set</td>
</tr>
<tr>
<td>Restrict access</td>
<td>Only single users or members of the db_owner, dbcreator, or sysadmin groups can access the database.</td>
<td>Not set (MULTI_USER)</td>
<td>MULTI_USER</td>
</tr>
<tr>
<td>Torn page detection</td>
<td>Specifies whether incomplete pages can be detected</td>
<td>Set</td>
<td>Set</td>
</tr>
<tr>
<td>Truncate log on checkpoint</td>
<td>Automatically marks inactive portions of log for reuse on checkpoint</td>
<td>Not set</td>
<td>N/A</td>
</tr>
<tr>
<td>Use quoted identifiers</td>
<td>Specifies whether the Microsoft SQL Server enforces ANSI rules regarding quotation marks. Select this option to specify that double quotation marks be used only for identifiers, such as column and table names. Note that character strings must be enclosed in single quotation marks.</td>
<td>Not set</td>
<td>Not set</td>
</tr>
</tbody>
</table>

**Note:**

Not all ANSI options can be set using Management Studio. The ANSI database configuration options include: ANSI_NULLS, ANSI_NULL_DEFAULT, ANSI_PADDING, ANSI_WARNINGS, ARITHABORT, CONCAT_NULL_YIELDS_NULL, NUMERIC_ROUNDABORT, and QUOTED_IDENTIFIER.

The options you set may not take effect, since these options can also be set at a higher level. For example, if the session option QUOTED_IDENTIFIER was turned on, the equivalent database configuration option is irrelevant. Some tools or database interfaces turn certain session options on or off, so that the database configuration options never take effect.
The following table summarizes the characteristics of each recovery model:

<table>
<thead>
<tr>
<th>Model/Support</th>
<th>Allows Log Backup</th>
<th>Allows Point-in-Time/Log Mark Restoration</th>
<th>Allows Backup Log when Data Crashes (Saves changes until the crash point)</th>
<th>Amount of Bulk Operation Logging (can affect the performance of bulk operations)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Minimal</td>
</tr>
<tr>
<td>Bulk Logged</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Minimal</td>
</tr>
<tr>
<td>Full</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Full</td>
</tr>
</tbody>
</table>

To check your database’s properties, run the command:

EXEC sp_helpdb <database name>

For information on SQL databases, see Microsoft SQL Server Books Online at http://www.microsoft.com/downloads.

Composite Indexes vs. Non Composite Indexes

For environments that execute heavy data-in operations, in order to speed up the INSERT statements, it is recommended to transform the database from composite index to non composite index.

You can achieve this by invoking the modifyCompositeIndexes JMX method with composite indexes setting to false. For detailed instructions, see "How to Modify Composite Indexes".

Monitoring the Fragmentation Indexes

Due to the fact that UCMDB is an OLTP application and performs many insert, update, and delete operations each day, its indexes might become fragmented. The index fragmentation could be even higher if discovery jobs that run also modify the data.

To determine the indexes’ fragmentation, run the following script every day. This script could be scheduled as part of the SQL job. The result of the script contains all the indexes having average fragmentation larger than 15 percent. When you have data from several days, you should compare the results and determine if the fragmentation index continue to increase day by day.

If the indexes continue to become more fragmented, you should rebuild these indexes each day. You must rebuild or reorganize the indexes when no discovery, enrichment, or other jobs that modify the data are running.
To determinate the fragmentation indexes

You must run this script in the context of the UCMDB database. In this example, replace CMDB_DATABASE with the actual name of the client's UCMDB database.

```sql
USE CMDB_DATABASE
select
    OBJECT_NAME(b.object_id) as TblName,
    CAST(a.object_id as varchar) as object_id,
    CAST(a.index_id as varchar) as index_id,
    Coalesce(b.name,'') as IndexName,
    CAST(Coalesce(a.avg_fragmentation_in_percent,'') as varchar) as PercentFragment,
    CAST(CAST(Coalesce(a.fragment_count,'') as bigint) as varchar) as TotalFrags,
    CAST(Coalesce(a.avg_fragment_size_in_pages,'') as varchar) as PagesPerFrag,
    CAST(Coalesce(a.page_count,'') as bigint) as NumPages,
    Coalesce(a.index_type_desc,'') as IndexType,
    CAST(Coalesce(a.index_depth,'') as varchar) as index_depth,
    CAST(Coalesce(a.index_level,'') as varchar) as index_level,
    CAST(Coalesce(a.avg_page_space_used_in_percent,'') as varchar) as avg_page_space_used_in_percent,
    CAST(CAST(Coalesce(a.record_count,'') as bigint) as varchar) as record_count
FROM sys.dm_db_index_physical_stats(DB_ID('CMDB_DATABASE'),NULL, NULL, NULL, 'SAMPLED') AS a
    JOIN sys.indexes AS b
    ON a.object_id = b.object_id AND a.index_id = b.index_id
WHERE a.avg_fragmentation_in_percent > 15 and record_count > 100000
ORDER BY OBJECT_NAME(b.object_id),a.index_id;
```
Chapter 5: Using Windows Authentication to Access the Microsoft SQL Server Database

Unless configured otherwise, Universal CMDB uses Microsoft SQL Server authentication to access the Microsoft SQL Server databases. However, Windows authentication can also be used.

This chapter describes how to enable Universal CMDB to use Windows authentication to access Microsoft SQL Server databases.

**Note:** The authentication procedure described in this chapter is relevant for Configuration Manager as well as Universal CMDB.

This chapter includes:

- Enabling Universal CMDB to Work with Windows Authentication ............................................ 32

Enabling Universal CMDB to Work with Windows Authentication

You can enable Universal CMDB to use Windows authentication instead of Microsoft SQL Server authentication to access the Universal CMDB database.

To enable Universal CMDB to use Windows authentication to access a Microsoft SQL database, you must:

- Configure the Microsoft SQL Server to use Windows authentication.
- Launch the Universal CMDB Server service on all the Universal CMDB servers with a Windows user that has the necessary permissions to access the Microsoft SQL database.
- Run the UCMDB Server Configuration utility to create or connect to a Microsoft SQL database and specify the use of Windows authentication.

This section includes the following topics:

"Configuring Microsoft SQL Server to Use Windows Authentication" on the next page
"Launching the UCMDB Server service with a Windows User " on the next page
"Creating or Connecting to a Microsoft SQL Database Using Windows Authentication" on page 34
Configuring Microsoft SQL Server to Use Windows Authentication

To configure Microsoft SQL Server to use Windows authentication, perform the following steps:

1. In the SQL Server Enterprise Manager, select Security > Logins, right-click and choose New Login.
2. Enter the desired domain account, including the domain name, in the following pattern: DOMAIN\USER (for example, MY_DOMAIN\MY_ACCOUNT).
3. In the Server Roles tab, select System Administrators and click OK.

Launching the UCMDB Server service with a Windows User

By default, the UCMDB Server service is run as a system service. If you have configured your Microsoft SQL Server to use Windows authentication, you must change the user running the UCMDB Server service to the same Windows user you defined for the Microsoft SQL Server in order to enable the service user to access the database.

To change the UCMDB Server service user:

1. Disable Universal CMDB (Start > Programs >HP UCMDB > Stop HP UCMDB Server).
2. In Microsoft’s Services window, double-click UCMDB_Server. The UCMDB Server Properties (Local Computer) dialog box opens.
3. Click the Log On tab.
4. Select **This account** and browse to choose the user you previously defined for your Microsoft SQL Server.

5. Enter the selected user’s Windows password and confirm this password.

6. Click **Apply** to save your settings and **OK** to close the dialog box.

7. Enable Universal CMDB (**Start** > **Programs** > **HP UCMDB** > **Start HP UCMDB Server**).

Creating or Connecting to a Microsoft SQL Database Using Windows Authentication

You create or connect to a database using the UCMDB Server Configuration utility. To create or connect to a Microsoft SQL database using Windows authentication, you must select this option within the UCMDB Server Configuration utility. For details on using the utility, see the section about accessing the servers in the interactive **HPE Universal CMDB Deployment Guide**.
Part III: Deploying the Oracle Server Database
Chapter 6: Overview of Oracle Server Deployment

This chapter includes:

- About Oracle Server Deployment ................................................................. 36
- System Requirements .................................................................................... 37

About Oracle Server Deployment

To deploy Oracle Server for use with Universal CMDB, perform the following procedures:

**Note:** Oracle 12C must be installed without container database (CDB). UCMDB does not support Oracle 12C with CDB.

- **Install Oracle Server.**
  
  For details on Oracle software installation, refer to the installation guide in the documentation for your specific Oracle platform. For software installation options, see "Oracle Server Configuration and Sizing Guidelines" on page 41.

- **Build a database on Oracle Server to store Universal CMDB data.**
  
  For instance configuration and sizing guidelines, see "Oracle Server Configuration and Sizing Guidelines" on page 41. For details on database instance installation, refer to the installation guide in the documentation for your specific Oracle platform.

- **Create one or more Oracle tablespaces to store Universal CMDB data.**
  
  For details, see "Oracle Tablespaces" on page 44.

- **Create an Oracle user schema for Universal CMDB schemas.**
  
  You can create Universal CMDB user schemas manually, or you can use the UCMDB Server Configuration Wizard to create the schemas for you (**Start > UCMDB > Start HP Universal CMDB Server Configuration Wizard**). For details on creating an Oracle user schema for Universal CMDB, see "Manually Creating the Oracle Server Database Schemas" on page 38.
System Requirements

This section describes the system requirements for working with Oracle Server in conjunction with Universal CMDB.

This section includes the following topics:

"Hardware Requirements" below

"Software Requirements" below

Hardware Requirements

For Universal CMDB hardware sizing guidelines, see "Universal CMDB Sizing" on page 8.

For Oracle hardware requirements, refer to the installation guide for your specific Oracle platform. Additional information is also available in the Oracle software distribution media as well as the online Oracle documentation. For Oracle documentation, refer to:


Software Requirements

For details on supported versions of Oracle Server, see the Server Database Requirements section of the HPE Universal CMDB Support Matrix.
Chapter 7: Manually Creating the Oracle Server Database Schemas

This chapter includes:

Overview of the Oracle Database Schema ................................................................. 38
Prerequisites .................................................................................................................. 38
Creating Schema Default and Temporary Tablespaces .............................................. 39
Creating Schema User Permissions ............................................................................. 39
Creating Schemas and Schema Objects ...................................................................... 40

Overview of the Oracle Database Schema

Universal CMDB uses the configuration management database (CMDB) database schema for its persistency.

During the Universal CMDB setup, new schemas can be set up automatically by the setup procedure, or already existing schemas can be used. Existing schemas can either have been created manually in advance, or can have been created by a previous installation of the same release of Universal CMDB.

If you choose to have the schema set up automatically by the setup procedure, you must have administrator permissions. Usually, the System account credentials are used. If you are unable to submit database administrator connection parameters (for example, due to organization security restrictions), manually set up the schema in advance.

For details about installing the Universal CMDB server, see the interactive HPE Universal CMDB Deployment Guide.

Prerequisites

Before creating the database schema, the Oracle Database Server must be installed according to the Oracle documentation. For further details on installing and configuring the Oracle Database Server, see "Oracle Server Configuration and Sizing Guidelines" on page 41.

If the schema is manually created prior to running the Universal CMDB setup, during the setup procedure, select the option to use an existing schema and enter the relevant data about the existing
schema for which you are prompted. The data you enter is written to the configuration files that are used by the JDBC driver to connect to the existing schema.

Creating Schema Default and Temporary Tablespaces

The database administrator should create at least one Temporary tablespace in the database instance hosting the Universal CMDB schema, and at least one tablespace to serve as the default tablespace for the Universal CMDB schema.

The tablespace specification should be made according to your storage policy, taking into account RAID configuration, available disks and storage devices.

For further details on tablespace file sizes and recommended settings, see "Oracle Tablespace Settings" on page 45.

Creating Schema User Permissions

The database administrator should create an Oracle schema user with the database permissions required by the Universal CMDB application.

The following database permissions must be granted to Universal CMDB Oracle schema users. These permissions are also used by the Universal CMDB installer to create a new Oracle user:

- Roles
  - Connect
- Privileges
  - CREATE TABLE
  - CREATE VIEW
  - CREATE SEQUENCE
  - CREATE TRIGGER
  - CREATE PROCEDURE
  - UNLIMITED TABLESPACE
  - ALTER USER ${user} DEFAULT ROLE ALL
  - CREATE TYPE
EXECUTE ON DBMS_LOB
EXECUTE ON DBMS_STATS

Note:
- Universal CMDB supports any user with higher permissions. For Universal CMDB certification, use an Oracle user that has the exact Oracle permissions described above.
- The last two permissions (EXECUTE ON DBMS_LOB and EXECUTE ON DBMS_STATS) are granted by default.

Creating Schemas and Schema Objects

The database administrator should create a schema for CMDB persistency.

Create a user schema with the permissions described in "Creating Schema User Permissions" on the previous page. No scripts are necessary to create the objects to populate these schemas.
Chapter 8: Oracle Server Configuration and Sizing Guidelines

This chapter contains guidelines for the Oracle database configuration and storage settings that should be used when working with Oracle Server and Universal CMDB. Note that the recommended settings differ according to the size of your Universal CMDB deployment. For details, see "Universal CMDB Sizing" on page 8.

This chapter includes:

- Oracle Parameter Settings ................................................................. 41
- Oracle Tablespaces ........................................................................ 44
- Oracle Tablespace Settings .............................................................. 45
- Using RAID Configuration ............................................................... 48
- Composite Indexes vs. Non Composite Indexes ............................... 50
- Checklist for Universal CMDB Support and Certification ................ 50

Oracle Parameter Settings

The following table describes the recommended values for a number of Oracle database initialization parameters, when working with the Universal CMDB database server:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Universal CMDB Deployment</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Small</td>
<td>Large</td>
</tr>
<tr>
<td>DB_BLOCK_SIZE</td>
<td>8K</td>
<td>8K-16K</td>
</tr>
<tr>
<td>DB_CACHE_ADVICE</td>
<td>ON</td>
<td>ON</td>
</tr>
<tr>
<td>SGA_TARGET</td>
<td>1 GB</td>
<td>4 GB and higher</td>
</tr>
<tr>
<td>MEMORY_TARGET</td>
<td>2 GB</td>
<td>5 GB and higher</td>
</tr>
<tr>
<td>LOG_BUFFER</td>
<td>1 MB</td>
<td>5 MB</td>
</tr>
<tr>
<td>Parameter Name</td>
<td>Universal CMDB Deployment</td>
<td>Remarks</td>
</tr>
<tr>
<td>----------------------------------------------------</td>
<td>---------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>DB_FILE_MULTIBLOCK_READ_COUNT</td>
<td>Oracle default value</td>
<td></td>
</tr>
<tr>
<td>PROCESSES</td>
<td>200</td>
<td>400 Add an additional 100 as a safety net</td>
</tr>
<tr>
<td>SECTIONS</td>
<td>225</td>
<td>445 (1.1 * PROCESSES) + 5</td>
</tr>
<tr>
<td>OPTIMIZER_INDEX_COST_ADJ parameter value</td>
<td>100</td>
<td>100 Affects performance</td>
</tr>
<tr>
<td>TIMED_STATISTICS</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>LOG_CHECKPOINT_INTERVAL</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LOG_CHECKPOINT_TIMEOUT</td>
<td>0; or greater than or equal to 1800</td>
<td>0; or greater than or equal to 1800</td>
</tr>
<tr>
<td>OPTIMIZER_MODE</td>
<td>ALL_ROWS</td>
<td>ALL_ROWS</td>
</tr>
<tr>
<td>CURSOR_SHARING</td>
<td>Exact</td>
<td>Exact</td>
</tr>
<tr>
<td>OPEN_CURSORS</td>
<td>800</td>
<td>800</td>
</tr>
<tr>
<td>COMPATIBLE</td>
<td>The same as the release installed</td>
<td>The same as the release installed</td>
</tr>
<tr>
<td>SQL_TRACE</td>
<td>False, True</td>
<td>False</td>
</tr>
<tr>
<td>UNDO_MANAGEMENT</td>
<td>Auto</td>
<td>Auto</td>
</tr>
<tr>
<td>UNDO</td>
<td>Oracle</td>
<td>Oracle</td>
</tr>
<tr>
<td>Parameter Name</td>
<td>Universal CMDB Deployment</td>
<td>Remarks</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>Small</td>
<td>Large</td>
</tr>
<tr>
<td>RETENTION</td>
<td>default</td>
<td>default</td>
</tr>
<tr>
<td>RECYCLEBIN</td>
<td>Off</td>
<td>Off</td>
</tr>
<tr>
<td>NLS_LENGTH_SEMANTICS</td>
<td>BYTE</td>
<td>BYTE</td>
</tr>
<tr>
<td>NLS_COMP</td>
<td>BINARY</td>
<td>BINARY</td>
</tr>
<tr>
<td>NLS_SORT</td>
<td>BINARY</td>
<td>BINARY</td>
</tr>
<tr>
<td>WORKAREA_SIZE_POLICY</td>
<td>AUTO</td>
<td>AUTO</td>
</tr>
<tr>
<td>PGA_AGGREGATE_TARGET</td>
<td>400 MB</td>
<td>1 GB and higher</td>
</tr>
<tr>
<td>STATISTICS_LEVEL</td>
<td>TYPICAL</td>
<td>TYPICAL</td>
</tr>
<tr>
<td>OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES</td>
<td>FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td>AUDIT_TRAIL</td>
<td>NONE</td>
<td>NONE</td>
</tr>
<tr>
<td>CURSOR_SPACE_FOR_TIME</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>USE_STORED_OUTLINES</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>BLANK_TRIMMING</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>FIXED_DATE</td>
<td>Not set</td>
<td>Not set</td>
</tr>
<tr>
<td>Parameter Name</td>
<td>Universal CMDB Deployment</td>
<td>Remarks</td>
</tr>
<tr>
<td>----------------</td>
<td>---------------------------</td>
<td>---------</td>
</tr>
<tr>
<td></td>
<td>Small</td>
<td>Large</td>
</tr>
<tr>
<td>_PARTITION_LARGE_EXTENTS</td>
<td>FALSE</td>
<td>FALSE</td>
</tr>
</tbody>
</table>

Note the following:

- **SGA_TARGET.** Setting this parameter configures Oracle to automatically determine the size of the buffer cache (db_cache_size), shared pool (shared_pool_size), large pool (large_pool_size), java pool (java_pool_size), and streams pool (streams_pool_size).

  The value configured for SGA_TARGET sets the total size of the SGA components.

  When SGA_TARGET is set (that is, it's value is not 0), and one of the above pools is also set to a non-zero value, the pool value is used as the minimum value for that pool.

- **MEMORY_TARGET.** In Oracle 11g, Automatic Memory Management enables the entire instance memory to be automatically managed and tuned by the instance. The instance memory contains the System Global Area (SGA) and the Program Global Area (PGA). In Oracle 11g, MEMORY_TARGET is the only required memory parameter to set, however it is recommended to set SGA_TARGET or PGA_AGGREGATE_TARGET as well to avoid frequent resizing of the SGA and PGA components. The values entered for SGA_TARGET and PGA_AGGREGATE_TARGET serve as minimum values.

  **Note:** HPE does not recommend to collect statistics for any temporary tables.

**Oracle Tablespaces**

An Oracle tablespace is an Oracle object that is a logical container of database objects, for example, tables, indexes, and so forth. When working with Universal CMDB, you must create one or more dedicated default tablespaces for your Universal CMDB user schema. You may also want to create a dedicated temporary tablespace for Universal CMDB. To create a tablespace, you must provide specific operating system files that physically represent the tablespace, as well as extent parameters.
When mapping operating system files, there is an option to make the file auto-extendable. This feature is supported by Universal CMDB, but not certified for use with Universal CMDB, since it can cause the system to consume all available disk space.

**Locally Managed Tablespaces**

A locally managed tablespace is a feature introduced in Oracle8i. Prior to Oracle8i, all tablespaces were dictionary-managed tablespaces. A tablespace that manages its extents locally can have either uniform extent sizes, or variable extent sizes that are determined automatically by the system. When you create the tablespace, the **uniform** or **autoallocate** (system-managed) option specifies the type of allocation.

For system-managed extents, Oracle determines the optimal size of extents, with a minimum extent size of 64 KB. This is the default extent size for permanent tablespaces.

For uniform extents, you can specify an extent size, or use the default size, which is 1 MB. Temporary tablespaces that manage their extents locally can only use this type of allocation.

Note that the NEXT, PCTINCREASE, MINEXTENTS, MAXEXTENTS, and DEFAULT STORAGE storage parameters are not valid for extents that are managed locally.

All data and temporary tablespaces should be locally managed when working with Universal CMDB.

For information on locally managing temporary tablespace using TEMPFILE, see “Temporary Tablespace Settings” on page 47.

**Oracle Tablespace Settings**

This section describes the storage settings and file sizing guidelines for data tablespaces, temporary tablespaces, redo logs, and undo tablespaces.

This section includes the following topics:

- "Data Tablespace Settings" on the next page
- "System Tablespace Settings" on the next page
- "Temporary Tablespace Settings" on page 47
- "Redo Log Settings" on page 47
- "Undo Segment Settings" on page 48
Data Tablespace Settings

The following table specifies the recommended sizes for Universal CMDB tablespaces:

<table>
<thead>
<tr>
<th>Tablespace</th>
<th>Universal CMDB Deployment</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMDB</td>
<td>Small 5 GB</td>
<td>Large 60 GB</td>
</tr>
</tbody>
</table>

**Note:** The data in the table is only relevant when the following parameters are assigned their default values:

- the Oracle parameter `deferred_segment_creation` is set to `true` (only relevant for Oracle 11g R2)
- the tablespace setting `INITIAL_EXTENT` is set to **65,536**

Data Tablespace Default Storage Settings

The storage settings for data tablespaces should be:

- Locally managed tablespace
- Automatic segment space management
- Automatic local extent management

The following command can be used to create a data tablespace:

```
CREATE TABLESPACE <tablespace name> DATAFILE '<data file full path>' REUSE SIZE <file size>
EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;
```

For example:

```
CREATE TABLESPACE APPTBS DATAFILE 
'/ORADATA/ORCL/APPTBS1.DBF' REUSE SIZE 1024M
EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;
```

System Tablespace Settings

The following table specifies the recommended settings for system tablespaces:
The System Tablespace storage default settings should be:

- Locally managed tablespace
- Segment space management:
  - SYSAUX: Automatic
  - SYSTEM: Manual
- Automatic local extent management

Temporary Tablespace Settings

The following table specifies the recommended settings for temporary tablespaces:

<table>
<thead>
<tr>
<th>Tablespace</th>
<th>Universal CMDB Deployment</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Small</td>
<td>Large</td>
</tr>
<tr>
<td>TEMP</td>
<td>1 GB</td>
<td>10 GB</td>
</tr>
<tr>
<td>TEMP storage</td>
<td>Uniform allocation: 2 MB</td>
<td>Uniform allocation: 2 MB</td>
</tr>
<tr>
<td>settings</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Redo Log Settings

The following table specifies the recommended settings for redo log files:
### Setting

<table>
<thead>
<tr>
<th>Setting</th>
<th>Universal CMDB Deployment</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Small</td>
<td>Large</td>
</tr>
<tr>
<td>Redo log file size</td>
<td>100 MB</td>
<td>200 MB - 500 MB</td>
</tr>
<tr>
<td>Minimum number of groups</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Minimum number of members per group</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

### Undo Segment Settings

The following table specifies the recommended Undo settings:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Universal CMDB Deployment</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Small</td>
<td>Large</td>
</tr>
<tr>
<td>Undo tablespace size</td>
<td>1 GB</td>
<td>10 GB</td>
</tr>
<tr>
<td>UNDO_MANAGEMENT parameter</td>
<td>AUTO</td>
<td>Oracle default value</td>
</tr>
<tr>
<td>UNDO_RETENTION parameter</td>
<td>Oracle default value</td>
<td></td>
</tr>
</tbody>
</table>

The Undo Tablespace storage default settings should be:

- Locally managed tablespace
- Automatic segment space management
- Segment space management in undo tablespaces is manual

### Using RAID Configuration

The use of RAID is transparent to Oracle. All the features specific to RAID configurations are handled by the operating system and not by Oracle.

The use of RAID devices differs according to the Oracle file type. Data files and archive logs can be placed on RAID devices, since they are accessed randomly. Redo logs should not be put on RAID devices, since they are accessed sequentially and performance is enhanced by having the disk drive
head near the last write location. However, mirroring of redo log files is strongly recommended by Oracle.

RAID is much easier to use than the Oracle techniques for data placement and striping.

Note the following RAID configuration recommendations:

- RAID usually impacts write operations more than read operations. This is especially true where parity needs to be calculated (RAID 3, RAID 5, and so forth).
- You can place online or archived redo log files on RAID 1 devices. Do not use RAID 5. In addition, place TEMP tablespace data files on RAID 1 devices, instead of RAID 5, because the streamed write performance of distributed parity (RAID 5) is not as good as that of simple mirroring (RAID 1).
- Swap space can be used on RAID devices without affecting Oracle.

The following table describes the RAID devices and types to be used with each Oracle file type:

<table>
<thead>
<tr>
<th>RAID</th>
<th>Type of RAID</th>
<th>Control File</th>
<th>Database File</th>
<th>Redo Log File/Temporary</th>
<th>Archive File</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Striping</td>
<td>Avoid</td>
<td>OK</td>
<td>Avoid</td>
<td>Avoid</td>
</tr>
<tr>
<td>1</td>
<td>Shadowing</td>
<td>OK</td>
<td>OK</td>
<td>Recommended</td>
<td>Recommended</td>
</tr>
<tr>
<td>0+1</td>
<td>Striping + Shadowing</td>
<td>OK</td>
<td>Recommended</td>
<td>Avoid</td>
<td>Avoid</td>
</tr>
<tr>
<td>3</td>
<td>Striping with static parity</td>
<td>OK</td>
<td>Avoid when this data file involves heavy write operation</td>
<td>Avoid</td>
<td>Avoid</td>
</tr>
<tr>
<td>5</td>
<td>Striping with rotating parity</td>
<td>OK</td>
<td>Avoid when this data file involves heavy write operation</td>
<td>Avoid</td>
<td>Avoid</td>
</tr>
</tbody>
</table>

**Note:**

- RAID 0 does not provide protection against failures. It requires a strong backup strategy.
- RAID 0+1 is recommended for database files because it avoids hot spots and provides the best possible performance during a disk failure. The disadvantage of RAID 0+1 is its costly configuration.
- Use the highest RPM disks for temporary/redu logs. Use as many controllers as you can in the array, and ensure that you place the redo log groups on different controllers.
Composite Indexes vs. Non Composite Indexes

For environments that execute heavy data-in operations, in order to speed up the INSERT statements, it is recommended to transform the database from composite index to non composite index.

You can achieve this by invoking the modifyCompositeIndexes JMX method with composite indexes setting to false. For detailed instructions, see "How to Modify Composite Indexes".

Checklist for Universal CMDB Support and Certification

Information is provided in this section for both supported and certified Oracle options.

The certified options are recommended for working with Universal CMDB. Certified options are rigorously tested by HPE quality assurance personnel. Supported options are those options for which HPE quality assurance personnel have successfully performed basic tests.

<table>
<thead>
<tr>
<th>Option</th>
<th>Supported</th>
<th>Recommended</th>
<th>Remarks</th>
<th>For more information, see</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle edition</td>
<td>Standard, Enterprise</td>
<td>Enterprise</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dedicated Universal CMDB server</td>
<td>Not necessary</td>
<td>Not necessary. It is recommended to dedicate an instance for Universal CMDB.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Use of multiple Oracle instances</td>
<td>Yes</td>
<td>No</td>
<td>The configuration of all instances must match in a certified environment.</td>
<td>&quot;System Requirements&quot; on page 37</td>
</tr>
<tr>
<td>Use of non-default port</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Undo management</td>
<td>Automatic; Manual</td>
<td>Automatic</td>
<td>Set UNDO_MANAGEMENT parameter to AUTO in a certified environment</td>
<td></td>
</tr>
<tr>
<td>Option</td>
<td>Supported</td>
<td>Recommended</td>
<td>Remarks</td>
<td></td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------</td>
<td>-------------</td>
<td>-------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Oracle Shared Servers connection method</td>
<td>Yes</td>
<td>No</td>
<td>Universal CMDB uses a connection pool architecture. Use the dedicated server connection method in a certified environment</td>
<td></td>
</tr>
<tr>
<td>Oracle replication</td>
<td>No full support</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operating System file compression</td>
<td>No</td>
<td>No</td>
<td>Not supported by Oracle; Causes abnormal behavior and affects performance</td>
<td></td>
</tr>
<tr>
<td>Database control files required</td>
<td>Greater than or equal to 2</td>
<td></td>
<td>Preferable on different disks.</td>
<td></td>
</tr>
<tr>
<td>Redo log groups</td>
<td>Greater than or equal to 3</td>
<td>4</td>
<td>Oracle enables the software mirroring of redo log files. This is achieved by creating at least two members of redo log in each group. Members of the same group should reside on different disks.</td>
<td></td>
</tr>
<tr>
<td>Character set</td>
<td>WE8ISO8859P1; UTF8,AL32UTF8</td>
<td>AL32UTF8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPEN_CURSORS</td>
<td>800</td>
<td>800</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Working in archive log mode</td>
<td>True; False</td>
<td>True</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Autoextend option in tablespace files</td>
<td>Yes</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Locally managed data tablespace</td>
<td>Yes</td>
<td>Yes</td>
<td>&quot;Locally Managed Tablespaces&quot; on page 45</td>
<td></td>
</tr>
<tr>
<td>Tablespace extent management</td>
<td>Local uniform for TEMP tablespace</td>
<td>Local uniform for TEMP tablespace</td>
<td>&quot;Oracle Tablespace Settings&quot; on</td>
<td></td>
</tr>
</tbody>
</table>

"Oracle Tablespace Settings" on
## Automatic Segment Space Management Tablespace (ASSM)

<table>
<thead>
<tr>
<th>Option</th>
<th>Supported</th>
<th>Recommended</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic Segment Space Management Tablespace (ASSM)</td>
<td>Yes</td>
<td>Yes</td>
<td>page 45</td>
</tr>
</tbody>
</table>

For more information, see page 45.
Chapter 9: Real Application Cluster Support

This chapter includes:

- About Oracle Real Application Cluster (RAC) ................................................................. 53
- Single Client Access Name ........................................................................................................ 54
- Client Side Configuration for Oracle RAC ............................................................................. 54
- Create/Connect to an Universal CMDB Database ................................................................. 57

Note: The information in this chapter applies to advanced users only.

About Oracle Real Application Cluster (RAC)

A cluster is a collection of interconnected servers that appear as one server to the end user and to applications. Oracle Real Application Cluster (RAC) is Oracle’s solution for high availability, scalability, and fault tolerance. It is based on clustered servers that share the same storage.

Oracle RAC is a single Oracle database installed on a cluster of hardware servers. Each server runs an instance of the database and all the instances share the same database files.

For more details about Oracle RAC, refer to the Oracle Clusterware Guide and the Oracle Real Application Clusters Administration and Deployment Guide in the Oracle documentation set of your release.

In this chapter, the following Oracle RAC example is used:

- Oracle RAC database name: UCMDBRAC
- Machine names: Server1, Server2
- On each machine there is an Oracle instance of UCMDBRAC:
  - SID on Server1: RAC1
  - SID on Server2: RAC2
- On each machine there is a virtual IP (Server1-Vip and Server2-Vip):
  - Server1-Vip is assigned to Server1
  - Server2-Vip is assigned to Server2
The virtual IP is in addition to the static IP assigned to the machine.

- The listeners on both servers are listening on the default port 1521 and support the database service UCMDBRAC.

![Diagram showing network configuration with SCAN method]

**Single Client Access Name**

In release 11g, Oracle introduced the Single Client Access Name (SCAN), as a preferred access method for clients connecting to the RAC. In this method clients are not required to configure individual nodes in the RAC, rather they use a single virtual IP known as the SCAN or the SCAN VIP.

The SCAN is a single network name defined for the cluster either in your organization’s Domain Name Server (DNS) or in the Grid Naming Service (GNS) that rotates between several IP addresses reflecting multiple listeners in the cluster. The SCAN eliminates the need to change clients when nodes are added to or removed from the cluster.

The SCAN and its associated IP addresses provide a stable name for clients to use for connections, independent of the nodes that make up the cluster. Database server SCAN addresses, virtual IP addresses, and public IP addresses must all be on the same subnet.

In Universal CMDB, when using Oracle RAC, it is recommended to use the SCAN method.

**Client Side Configuration for Oracle RAC**

Universal CMDB uses DataDirect’s JDBC driver to connect to regular Oracle databases and to Oracle RAC databases.
When using a RAC database, install the servers without creating the relevant UCMDB database schemas (for a description of UCMDB schemas, see "Introduction to Preparing the Database Environment" on page 7). After completing the client and server side RAC configuration, create the schemas as described in "Create/Connect to an Universal CMDB Database" on page 57.

Make the following changes in Universal CMDB's configuration files before you create the management database or connect to an existing one on Oracle RAC:

1. On the UCMDB Server, create the file `<Universal CMDB root directory>\UCMDBServer\conf\ucmdb-tnsnames.ora`.

The format of `ucmdb-tnsnames.ora` is the same as the Oracle tnsnames.ora network configuration file:

```sql
<DB service name> =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = <first instance virtual ip> )
       (PORT = <first instance's listener port>))
      (ADDRESS = (PROTOCOL = TCP)(HOST = <second instance virtual ip> )
       (PORT = <second instance's listener port>))
     (... entry for each instance...)
     (LOAD_BALANCE = on)
     (FAILOVER = on)
    )
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = <DB service name>)
    )
  )
```

where:

- `<DB service name>` is the name of a service the listeners support. It is the same one used in the CONNECT_DATA part.

- `ADDRESS_LIST` contains an address entry for each node in the RAC environment. In the case of Oracle 11g using SCAN, it contains only the SCAN virtual IP. The address contains all the details needed for connecting to the node:
  - `HOST` contains the virtual-IP for that instance. It is important to use the virtual IP and not the static IP of the node for faster failure detection.
  - `PORT` is the port on which the listener is configured to listen on that given node.
  - `FAILOVER` set to `on` allows the driver to try to connect to another node after failing to connect
to one of the nodes. The connection attempts continue until a connection is successfully established to one of the nodes or until none of the nodes can be reached.

An attempt to connect to another node takes place only if the connection to the current node fails. If the connection is refused by the node (for example, communication to the node was established, but the connection was rejected), no attempt is made to connect to another node.

**Caution:** Failover is for connection attempts only. If a connection fails during a database transaction, there is no failover of the transaction to another machine to continue the transaction.

- **LOAD_BALANCE** set to on instructs the driver to distribute connection requests between the nodes to prevent overloading any single node. The order in which the driver accesses the nodes is random.
- **SERVER** is the connection mode you want to use. There are two possible connection modes: dedicated and shared. Configure this according the Server configuration you support.
- **SERVICE_NAME** is the name of a service that the listeners support. It is the same one you gave in <DB service name>.

In the above example, **ucmdb-tnsnames.ora** would be configured as:

```sql
UCMDBRAC =
(DESCRIPTION =
 (ADDRESS_LIST =
 (ADDRESS = (PROTOCOL = TCP)(HOST = Server1-Vip)(PORT = 1521))
 (ADDRESS = (PROTOCOL = TCP)(HOST = Server2-Vip)(PORT = 1521))
 (LOAD_BALANCE = on)
 (FAILOVER = on)
 )
 (CONNECT_DATA =
 (SERVER = DEDICATED)
 (SERVICE_NAME = UCMDBBSMRAC)
 )
)
```

2. On the UCMDB Server, open the directory **<Universal CMDB root directory>\UCMDBServer\conf**. Locate the **jdbc.properties** file.
   a. Find the line starting with **cmdb.url**.
   b. Replace this with the following line:

```java
cmdb.url=
jdbc:mercury:oracle:TNSNamesFile=<HP Universal CMDB root
```
where `<SERVICE_NAME>` is the entry in `ucmdb-tnsnames.ora`, equivalent to the RAC service name.

Note that each back slash (`\`) in the path of TNSNamesFile is doubled.

c. If the file does not exist, create an empty `jdbc.properties` file under the above folder and add the following entry:

```
Oracle = dдорacle
cmdb.url = jdbc:mercury:oracle:TNSNamesFile=<HP Universal CMDB root directory>\UCMDBServer\conf\ucmdb-tnsnames.ora;TNSServerName=<SERVICE_NAME>
```

where `<SERVICE_NAME>` is the entry in `ucmdb-tnsnames.ora`, equivalent to the RAC service name.

d. If your server is running the UNIX or LINUX operating system, replace all the double back slashes with single backslashes.

**Create/Connect to an Universal CMDB Database**

When creating a new database schema or connecting to an existing schema in the UCMDB Server Configuration Wizard, fill all the parameters as follows:

- **Oracle Machine.** One of the Virtual IPs. In Oracle 11g, the SCAN virtual IP can be used.
- **Port.** The local listener port on the Oracle machine or the port of the SCAN listener.
- **SID.** The service name of the database.
- **Schema name and schema password.** The name and password of the existing user schema, or the name that you are giving the new user schema (for example, `UCMDB_schema`).

If you are creating a new user schema, you need the following additional information:

- **Admin user name and admin password (to connect as an administrator).** The name and password of a user with administrative permissions on Oracle Server (for example, a System user).
- **Default tablespace.** The name of the default tablespace you created for the user schema. For details on creating an Universal CMDB tablespace, see "Manually Creating the Oracle Server Database Schemas" on page 38".
- **Temporary tablespace.** The name of the temporary tablespace you assigned to the user schema. The default Oracle temporary tablespace is **temp**.

In the example, the parameters would be:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Machine</td>
<td>Server1-Vip</td>
</tr>
<tr>
<td>Port</td>
<td>1521</td>
</tr>
<tr>
<td>SID</td>
<td>UCMDBRAC</td>
</tr>
</tbody>
</table>
Part IV: Deploying the PostgreSQL Server Database
Chapter 10: Overview of PostgreSQL Server Deployment

This chapter includes:

- About PostgreSQL Server Deployment .................................................................................. 60
- System Requirements .............................................................................................................. 60
- Overview of the PostgreSQL Server Databases .................................................................. 61
- PostgreSQL Deployment Options ......................................................................................... 61
- Initialize the PostgreSQL Server .......................................................................................... 62

About PostgreSQL Server Deployment

**Caution:** Embedded PostgreSQL is not supported on Enterprise deployments of UCMDB.

To deploy PostgreSQL Server for use with Universal CMDB, you must perform the following procedures:

- **Install and configure PostgreSQL Server.**
  
  For details on installing and configuring PostgreSQL Server, see the PostgreSQL Server documentation.

- **Create databases on PostgreSQL Server.**
  
  You create Universal CMDB databases using the UCMDB Server Configuration Wizard.

System Requirements

This section describes the system requirements for working with PostgreSQL Server in conjunction with Universal CMDB.

This section includes the following topics:

- "Hardware Requirements" on the next page
- "Software Requirements" on the next page
Hardware Requirements

For Universal CMDB hardware sizing guidelines, see “Hardware Requirements” on page 8. For PostgreSQL Server hardware requirements, refer to the installation guide for your PostgreSQL Server release for your operating system.

Software Requirements

For details on supported versions of PostgreSQL Server, see the Server Database Requirements section of the *HPE Universal CMDB Support Matrix*.

Overview of the PostgreSQL Server Databases

Universal CMDB uses the configuration management database (CMDB) for its persistency.

During the Universal CMDB setup, the new database can be set up automatically by the setup procedure, or an existing database can be used. An existing database can either be created manually in advance (for example, due to organization security restrictions), or can be created by a previous installation of the same release of Universal CMDB.

For details about installing the Universal CMDB server, see the interactive *HPE Universal CMDB Deployment Guide*.

PostgreSQL Deployment Options

There are two options for deploying PostgreSQL Server with Universal CMDB:

- **Embedded.** PostgreSQL is part of the Universal CMDB installation.
- **External.** PostgreSQL installed on a separate server.

The embedded deployment option is appropriate for small capacity UCMDB installations. In this option, the following configuration files are used:

- `postgresql.conf`
- `ucmdbpg.conf`

Both files are located in the `UCMDB\UCMDBServer\PostgreSQL\data` directory. PostgreSQL parameters in the `ucmdbpg.conf` file overwrite parameters in the `postgresql.conf` file.
Note: In the external deployment option, only the `postgresql.conf` file is used by default.

Initialize the PostgreSQL Server

If you choose not to configure the UCMDB Server with the PostgreSQL database at the time of UCMDB installation, you can initialize the PostgreSQL database and create the database service according to the following instructions:

Note: Initializing the PostgreSQL database and creating the database service is necessary only if you want to use the local PostgreSQL as an alternative to Oracle/MSSQL. It is not mandatory to use the local PostgreSQL; the UCMDB Server functions properly without it.

For Windows:

Note: In the following instructions, replace `C:\hp\UCMDB\UCMDBServer\` with the appropriate path.

1. Download and install `vcredist_x64.exe` from the Microsoft website.
2. Run: `C:\hp\UCMDB\UCMDBServer\PostgreSQL\pgInitDb.bat username password port`
   
   The username, password, and port must be chosen by the user. Verify that the port is free.
3. Run:
   
   ```
   C:\hp\UCMDB\UCMDBServer\PostgreSQL\pgsql\bin\pg_ctl.exe register -N UCMDB_Server_DB -D "C:\hp\UCMDB\UCMDBServer\PostgreSQL\pgsql\data".
   ```
4. Run: `sc description UCMDB_Server_DB "HP UCMDB Database"`
5. Run: `net start UCMDB_Local_DB`.
6. Run the configuration wizard, and provide the username, password, and port from above.

For Linux:

Note: In the following instructions, replace `/opt/hp/UCMDB/UCMDBServer/` with the appropriate path.

1. Run: `find /opt/hp/UCMDB/UCMDBServer/PostgreSQL -type f \( -name "*.sh" -or -name "*.sql" -or -name "*.conf" -or -name "*.cnf" \) -exec dos2unix \{} \`
2. Create the `postgres_server` group using: `/usr/sbin/groupadd -f postgres_server`
3. Create the postgres_server user using: 
/usr/sbin/useradd -r -g postgres_server postgres_server

4. Run: rpm -qa | grep -ic postgres

   **Note:** If there are other versions of the Postgres RPM package installed on the system, it is recommended to remove them as they may cause conflicts.

5. Run:
   
   ```
   cd /opt/hp/UCMDB/UCMDBServer/PostgreSQL
   chmod -R 770 .
   chown -R postgres_server .
   chgrp -R postgres_server .
   ```

6. Run: runuser -l postgres_server -c
   ```
   "/opt/hp/UCMDB/UCMDBServer/PostgreSQL/pgInitDB.sh username password port"
   ```
   The username, password, and port must be chosen by the user. Verify that the port is free.

7. Edit /opt/hp/UCMDB/UCMDBServer/PostgreSQL/pgsql/postgresql.server and replace the string _install_dir_ with the path to the installation folder (for example, 
   `/opt/hp/UCMDB/UCMDBServer/`).

8. Run:
   
   ```
   cd /opt/hp/UCMDB/UCMDBServer/PostgreSQL/pgsql
   chmod a+rx ./postgresql.server
   ```

9. Copy /opt/hp/UCMDB/UCMDBServer/PostgreSQL/pgsql/postgresql.server to /etc/init.d/ and rename it to postgresql_server.

10. Run:
    
    ```
    chkconfig --add postgresql_server
    /sbin/service postgresql_server start
    ```

11. Run the configuration wizard, and provide the username, password and port from above.
Chapter 11: PostgreSQL Server Configuration

This chapter contains guidelines for the database configuration and storage settings that should be used when working with PostgreSQL Server and Universal CMDB.

This chapter includes:

- PostgreSQL Parameter Settings ................................................................. 64
- Critical PostgreSQL Server Files ............................................................... 65
- How to Configure PostgreSQL on Probe to Accept Local Connections Only .................... 66
- How to Configure PostgreSQL Log Files Rotation by Size .................................... 69
- Always Exclude PostgreSQL Data Folder from Anti-Virus Scan ............................ 70

PostgreSQL Parameter Settings

The following table describes the recommended (non-default) values for a number of PostgreSQL database initialization parameters, when working with the Universal CMDB database server:

<table>
<thead>
<tr>
<th>Category</th>
<th>Parameter Name</th>
<th>Universal CMDB Deployment</th>
<th>Defaults and Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory</td>
<td>shared_buffers</td>
<td>Small 1024 MB</td>
<td>Large 4096 MB</td>
</tr>
<tr>
<td></td>
<td>work_mem</td>
<td>25 MB</td>
<td>50 MB</td>
</tr>
<tr>
<td></td>
<td>maintenance_work_mem</td>
<td>256 MB</td>
<td>340 MB</td>
</tr>
<tr>
<td>Planner</td>
<td>effective_cache_size</td>
<td>4096 MB</td>
<td>8192 MB</td>
</tr>
<tr>
<td>Checkpoint (WAL)</td>
<td>checkpoint_segments</td>
<td>32</td>
<td>64</td>
</tr>
<tr>
<td></td>
<td>checkpoint_timeout</td>
<td>15 minutes</td>
<td>20 minutes</td>
</tr>
<tr>
<td></td>
<td>checkpoint_completion_target</td>
<td>0.9</td>
<td>0.9</td>
</tr>
</tbody>
</table>
### Critical PostgreSQL Server Files

Note the following types of files that are relevant when working with PostgreSQL Server with Universal CMDB. It is strongly recommended for these file types to be located on different disks from each other and from the main database files (the installation files):

- **Data (PGDATA)**

  All the data needed for a database is stored within the data directory, commonly referred to as PGDATA (after the name of the environment variable that can be used to define it). A common location for PGDATA is `/var/lib/pgsql/data`.

  The PGDATA directory contains several subdirectories and control files. In addition to these required items, the cluster configuration files PostgreSQL.conf, pg_hba.conf, and pg_ident.conf are...
traditionally stored in PGDATA (although in PostgreSQL 8.0 and later, it is possible to keep them elsewhere).

During the PostgreSQL installation, select the option to store the PGDATA files in a designated path.

- **Write-Ahead Logging (WAL)**

Write-Ahead Logging (WAL) is a standard method for ensuring data integrity. WAL’s central concept is that changes to data files (where tables and indexes reside) must be written only after those changes have been logged, that is, after log records describing the changes have been flushed to permanent storage. This procedure eliminates the need to flush data pages to disk on every transaction commit, because in the event of a problem, any changes that have not been applied to the data pages can be redone from the log records. (This is roll-forward recovery, also known as REDO.)

You can ensure that these files are located on different disks by moving the pg_xlog directory to another location. To do this, follow this procedure:

a. Shut down the server.

b. Create a symbolic link from the original location in the main data directory to the new location, using the following command:

   ```
   mklink /j "C:\Program Files\PostgreSQL\9.2\data\pg_xlog" "E:\pg_xlog".
   ```


How to Configure PostgreSQL on Probe to Accept Local Connections Only

By default the PostgreSQL instance running on the probe does not need to listen to requests from the network. However, you can perform this additional configuration to bind the PostgreSQL instance on the probe to localhost.

To do so,

1. On the probe, open the following file using a text editor:

   ```
   hp\UCMDB\DataFlowProbe\conf\postgresql.conf
   ```

2. Locate and modify the following file from:
listen_addresses = '*'

to:

listen_addresses = 'localhost'

3. Save the file.

4. Restart the probe.

   To restart the probe properly,
   
   a. Open services.msc.
   
   b. Stop the UCMDB Probe service.
   
   c. Stop the UCMDB_Probe_DB service.
   
   d. Start the UCMDB_Probe_DB service.
   
   e. Start the UCMDB Probe service.

How to Validate If the PostgreSQL Instance Is Listening Only to Localhost?

Validate from a remote PostgreSQL server

To validate that the change to the postgresql.conf file is taking effect successfully, you can run the following PostgreSQL client command from a remote server with PostgreSQL installed:

```
psql -username="mamuser" --host=<Target DataFlowProbe Host or IP>
```

When PostgreSQL is listening only to localhost, you will receive a "Connection refused" message.

Example outputs:

**PostgreSQL listening to localhost:**

```
C:\hp\UCMDB\DataFlowProbe\pgsql\bin>psql --username="mamuser" --host=10.10.15.220

psql: could not connect to server: Connection refused (0x0000274D/10061)
Is the server running on host "10.10.15.220" and accepting
```
TCP/IP connections on port 5432?

PostgreSQL listening to any host (known as "*" in the postgresql.conf file):

C:\hp\UCMDB\DataFlowProbe\pgsql\bin>psql --username="mamuser" --host=10.10.15.220

psql: FATAL: no pg_hba.conf entry for host "10.10.20.254", user "mamuser", database "mamuser", SSL off

The FATAL error indicates simply that the pg_hba.conf file is not properly set up, but for the purpose of this test it is sufficient to understand that the connection was successful from a remote host. This would indicate the change to listen_addresses in postgresql.conf was NOT applied properly per the instructions above. Please check those steps again.

Validate using Telnet

If you do not have another PostgreSQL system with psql installed, you can try to Telnet to the PostgreSQL port from a different computer to confirm if the change was applied correctly:

1. Install Telnet.
   a. Go to Start > Control Panel > Turn Windows features on or off.
      The Windows Features dialog opens.
   b. Select the check box for Telnet Client.
   c. Click OK.
2. Test connection.
   a. Open the Command Prompt window by going to Start > Run > cmd and clicking OK.
   b. From the Command Prompt window, run the following:

      telnet <target host/IP and port>

      For example, telnet 10.10.15.220 5432.

      If you receive a message like: "telnet: connect to address <address you specified>: Connection refused", it indicates that PostgreSQL is properly configured to refuse remote connections.
How to Configure PostgreSQL Log Files Rotation by Size

The log rotation by size on PostgreSQL keeps all the log files, and then the disk space could become full.

To configure rotation of the PostgreSQL log files by size:

1. Locate and open the PostgreSQL configuration file using a text editor:
   - For embedded PostgreSQL: `<UCMDB_Server_Home>\PostgreSQL\ucmdbpg.conf`
   - For external PostgreSQL: `<UCMDB_Server_Home>\PostgreSQL\data\postgresql.conf`

2. Copy and paste the following configuration into the file, under the CUSTOMIZED OPTIONS section:

   ```
   # These are only used if logging_collector is on:
   log_directory = 'pg_log' # directory where log files are written, can be absolute or relative to PGDATA
   log_filename = 'postgresql-%a.log' # log file name pattern, can include strftime() escapes
   log_file_mode = 0600 # creation mode for log files, begin with 0 to use octal notation
   log_truncate_on_rotation = on # If on, an existing log file with the
                                 # same name as the new log file will be truncated rather than appended to.
                                 # But such truncation only occurs on
                                 # time-driven rotation, not on restarts
                                 # or size-driven rotation. Default is
                                 # off, meaning append to existing files
                                 # in all cases.
   log_rotation_age = 1440 # Automatic rotation of logfiles will
                           # happen after that time. 0 disables.
   log_rotation_size = 0 # Automatic rotation of logfiles will
                        # happen after that much log output.
                        # 0 disables
   ```

3. Save the file.

4. Restart PostgreSQL for the changes to take effect.

**Recommended usage of the options for the backup period:**
For backup period of only 24 hours, just use %H: `log_filename = 'postgresql-%H.log'`, set `log_rotation_age = 60`.

One day could be too short for future investigation and troubleshooting, because the log files will be overwritten after one day.

For backup period of one week, use %a-%w: `log_filename = 'postgresql-%a.log'`, and set `log_rotation_age = 1440`.

The log files will be overwritten after one week. %a is the day of the week and %w is the week of the month.

For backup period of one month, use %d: `log_filename = 'postgresql-%d.log'`, and set `log_rotation_age = 1440`.

The log file will be overwritten after a month. %d is the day of the month.

For old log files like "31", you can right click order by modified date on Windows, or ls -t on Linux, then you can know this "31" is this month’s or last month’s.

Note that this configuration could increase the space used by the PostgreSQL log files.

Always Exclude PostgreSQL Data Folder from Anti-Virus Scan

When a third-party Anti-Virus software scans the PostgreSQL server data folder, it could cause the PostgreSQL data tables to corrupt.

To resolve the issue, always make sure that the PostgreSQL install directory is added into the anti-virus software exclusion list. The exclusion of data files will not introduce any potential security risk.

If you need to run weekly-based scan, monitor the `probeerror.log` file, and if a database error shows up (for example, a database error related to the Discovery_result table), do the following:

1. Clean the probe log folder.
2. Run Clear Probe Results Cache from UCMDB UI > Data Flow Management > Universal Discovery > Discovery Modules/Jobs to clean the problematic table.

This should resolve the issue.
Send documentation feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

**Feedback on Database Guide (Universal CMDB 10.31)**

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to cms-doc@hpe.com.

We appreciate your feedback!