

Oracle® Fusion Middleware

Creating Schemas with the Repository Creation Utility



12c (12.2.1.3.0)

E95489-03

February 2021

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Oracle Fusion Middleware Creating Schemas with the Repository Creation Utility, 12c (12.2.1.3.0)

E95489-03

Copyright © 2016, 2021, Oracle and/or its affiliates.

Primary Author: Oracle Corporation

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Intended Audience	vii
Documentation Accessibility	vii
Related Documents	vii
Conventions	viii

1 About the Repository Creation Utility

Verifying RCU System and Database Requirements	1-1
RCU Supported Platforms	1-1
Finding a Certified Database	1-2
Database Requirements for RCU	1-2
Additional Requirements for IBM DB2 Databases	1-2
Preparing for Schema Creation	1-3
About System Load and Product Load	1-3
Granting Permissions to Users for Querying System Load Objects	1-5
About Custom Prefixes	1-5
About the Service Table Schema	1-6
Planning Your Schema Creation	1-7
Organizing Schemas on a Single Database for a Single Domain	1-7
Organizing Schemas on Multiple Databases for a Single Domain	1-8
Organizing Schemas on a Single Database for Multiple Domains	1-9
Organizing Schemas on Multiple Databases for Multiple Domains	1-9
Integrating Components Using Declarative XML	1-10
Using RCU with Java Access Bridge (Windows Only)	1-10
Install Java Access Bridge	1-11
Configure RCU to Use Java Access Bridge	1-11

2 Obtaining and Running Repository Creation Utility

Obtaining RCU	2-1
Starting RCU	2-2
Creating Schemas	2-2

Creating Schemas as a User with Full SYS or SYSDBA Privileges	2-3
Creating Schemas as a User With Limited Database Privileges	2-4
Verifying Schema Version Numbers	2-7
Dropping Schemas	2-8
About Dropping Shared Tablespaces	2-9
Dropping Schemas and Deleting Datafiles (Windows Only)	2-9

3 Running Repository Creation Utility from the Command Line

Command-Line Syntax and Parameters	3-2
RCU Modes Available from the Command Line	3-2
RCU Operations Available from the Command Line	3-3
RCU Command Line Parameters	3-3
Using the -silent Command	3-9
Using the -validate Parameter to Verify Prerequisites	3-10
Using a Wallet to Store Passwords	3-12
Using the -interactive Command	3-13
Using Response Files	3-15
About Response Files	3-16
Creating a RCU Response File from the Graphical Interface	3-16
Creating a RCU Response File from the Command Line	3-16
Performing RCU Operations by Using a Response File	3-17
Creating a Repository from the Command Line	3-18
Generating a System Load Script From the Command Line	3-19
Loading Data Into the Repository From the Command Line	3-20
Dropping a Repository from the Command Line	3-20
Generating a List of Components in an Oracle Home from the Command Line	3-21
Generating a List of Schemas in a Database from the Command Line	3-21
RCU Environment Variables	3-22

A Repository Creation Utility Screens

Welcome	A-2
Create Repository	A-2
Create Repository	A-2
Drop Repository	A-3
Database Connection Details	A-3
Connection Credentials for Oracle Databases and Oracle Databases with Edition-Based Redefinition	A-4
Connection Credentials for Microsoft SQL Server Databases	A-6
Connection Credentials for IBM DB2 Databases	A-6
Select Components (for Create Operation)	A-6

Select Components (for Drop Operation)	A-8
Schema Passwords	A-8
Custom Variables	A-9
Custom Variables for Oracle Data Integrator	A-10
Custom Variables for Oracle SOA Suite	A-11
Custom Variable for Oracle WebCenter Portal Analytics	A-11
Custom Variables for Oracle GoldenGate Studio	A-11
Map Tablespaces	A-12
Default Tablespace Mappings	A-13
Changing Default and Temporary Tablespaces	A-13
Viewing and Changing Additional Tablespaces	A-13
Managing Tablespaces and Datafiles	A-13
Adding, Modifying, and Removing Tablespaces	A-14
Adding, Modifying, and Removing Datafiles	A-14
Encrypting Tablespaces	A-15
Summary (for Create Operation)	A-16
Summary (for Drop Operation)	A-17
Completion Summary (for Create Operation)	A-18
Completion Summary (for Drop Operation)	A-19

B **About Repository Creation Utility Schemas, IDs, and Tablespaces**

Repository Creation Utility Schemas, IDs, and Tablespaces	B-1
---	-----

C **Extending Repository Creation Utility to Configure Custom Application Repositories**

RCU Integration Options	C-1
RCU JDBC Engine Compliant SQL*Plus Scripts	C-2
Pure JDBC Scripts	C-2
SQL*Plus Scripts	C-3
External Processes	C-3
Java Code Using JavaAction	C-4
RCU Configuration Files	C-4
XML DTDs Defined by RCU	C-5
Component Descriptor Configuration File	C-5
Repository Configuration File	C-7
Master List of Supported Components	C-7
Storage Attributes Configuration File	C-9
Component Repository Configuration File	C-10
Component List Configuration File	C-17

Soft-Prerequisite Support	C-29
Default Tablespaces Configuration File	C-29
RCU Script Writing Guidelines	C-30
Guidelines for RCU JDBC Engine Compliant SQL*Plus Scripts	C-31
Guidelines for Pure JDBC Scripts	C-31
Guidelines for SQL*Plus Scripts	C-32
Guidelines for SQL Server-Based Scripts	C-32

D Troubleshooting Repository Creation Utility

General Troubleshooting Tips	D-1
RCU Log Files	D-2
RCU Fails to Connect with ORA-604 ORA-1882 Error Codes	D-3
SOAINFRA Schema Creation Fails on a Hardened Database	D-3
Need More Help?	D-4

Preface

The *Creating Schemas with the Repository Creation Utility* book contains overview information and usage instructions for Oracle Repository Creation Utility (RCU).

- [Intended Audience](#)
- [Documentation Accessibility](#)
- [Related Documents](#)
- [Conventions](#)

Intended Audience

This guide is intended for users who are installing Oracle Fusion Middleware products and are comfortable running some system administration operations, such as creating users and groups, adding users to groups, and installing operating system patches on the computer where Oracle Fusion Middleware products will be installed. Users on UNIX systems need `root` access to run some scripts.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Documents

For additional information, see the following resources:

- Preparing for an Oracle Fusion Middleware Installation in *Planning an Installation of Oracle Fusion Middleware*

This document contains important information about planning and preparing for Oracle Fusion Middleware product installations.

- Installing the Infrastructure Software in *Installing and Configuring the Oracle Fusion Middleware Infrastructure*

The Repository Creation Utility is included with the Oracle Fusion Middleware Infrastructure distribution.

Conventions

The following text conventions are used in this document:

Convention	Meaning
Boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>Italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
Monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

About the Repository Creation Utility

Many of the Oracle Fusion Middleware components require the existence of schemas in a database before configuration. These schemas are created and loaded in your database by using the Repository Creation Utility (RCU).

The following topics introduce you to the concepts and features that you should be familiar with before you use RCU:

- [Verifying RCU System and Database Requirements](#)
Use the certification and system requirements documents to find important information about supported platforms for RCU, certified databases, and database configuration information.
- [Preparing for Schema Creation](#)
Before you create your schemas, ensure that you understand the concepts regarding schema creation and organization.
- [Using RCU with Java Access Bridge \(Windows Only\)](#)
Java Access Bridge enables assistive technologies, such as JAWS screen reader, to read Java applications that run on the Windows platform.

Verifying RCU System and Database Requirements

Use the certification and system requirements documents to find important information about supported platforms for RCU, certified databases, and database configuration information.

Read this information carefully before you obtain and run RCU.

- [RCU Supported Platforms](#)
Use the system requirements document to review the supported platforms on which you can run RCU.
- [Finding a Certified Database](#)
If you do not have a database where you can install schemas, you must install and configure a certified database.
- [Database Requirements for RCU](#)
Ensure to review the general and component-specific database requirements that must be met before you run RCU.
- [Additional Requirements for IBM DB2 Databases](#)
If you are creating component schemas on an IBM DB2 database, there are additional requirements to consider.

RCU Supported Platforms

Use the system requirements document to review the supported platforms on which you can run RCU.

To know the platforms on which you can run RCU, see *Oracle Fusion Middleware System Requirements and Specifications* document.

Finding a Certified Database

If you do not have a database where you can install schemas, you must install and configure a certified database.

For a list of certified databases that can be used with RCU, see *System Requirements and Supported Platforms for Oracle Fusion Middleware 12c* certification document for your release, located on the *Oracle Fusion Middleware Supported System Configurations* page.

Database Requirements for RCU

Ensure to review the general and component-specific database requirements that must be met before you run RCU.

Before you begin using RCU, review the Repository Creation Utility Requirements section in the *Oracle Fusion Middleware System Requirements and Specifications* document.

Not all schemas are supported on all databases. Make sure you have read the information in this section carefully so that you configure a certified database that supports the schemas that you need for your Fusion Middleware components.

Additional Requirements for IBM DB2 Databases

If you are creating component schemas on an IBM DB2 database, there are additional requirements to consider.

In addition to the typical space and configuration database requirements, IBM DB2 databases also have the following special requirements:

- On IBM DB2 databases running on Linux operating systems, there is a limitation with regards to the length of the schema names.
- Create one database operating system user for each schema that is created in an IBM DB2 database.

For example, if you plan to create a schema named `DEV_STB` using RCU, then the operating system user must be named `dev_stb` (all lowercase letters).

You can create the operating system user by running the following command as `root` (this example creates the operating system user `dev_stb` and assigns the provided password to the user):

```
/usr/sbin/useradd dev_stb -p password -d /scratch/dev_stb
```

You can then set the password of the user (for example, `dev_stb`) by running the following commands as `root`:

```
passwd -u dev_stb
```

```
passwd dev_stb
```

Refer to your system documentation for more information.

- Create operating system level users in IBM DB2 for required schemas before you run the RCU.

For example:

Audit Services – `<prefix>_iau`, `<prefix>_iau_a` (Audit Services Append), and `<prefix>_iau_v` (Audit Services Viewer).

Service Table — `<prefix>_stb`

WebLogic Services - `<prefix>_wls`

- On IBM DB2/AIX, before you run the RCU or Upgrade Assistant Tool, you must ensure that the auxiliary schemas are available for the following components:

WebLogic Services – `<prefix>_wlsr`

Preparing for Schema Creation

Before you create your schemas, ensure that you understand the concepts regarding schema creation and organization.

- [About System Load and Product Load](#)
Schema creation in RCU is performed in multiple phases; each phase requires a different level of access to your database.
- [Granting Permissions to Users for Querying System Load Objects](#)
If you want to give a user with limited privileges the ability to query the system load objects, you must grant the user certain permissions before they can query the system load.
- [About Custom Prefixes](#)
Use custom prefixes to organize and group schemas in your database.
- [About the Service Table Schema](#)
The Service Table schema is a special schema that is automatically created once per prefix. The service table stores basic schema configuration information (for example, schema prefixes and passwords) that can then be accessed and used by other Oracle Fusion Middleware components during domain creation.
- [Planning Your Schema Creation](#)
Before you run RCU, make sure that you understand how schemas can be grouped together and distributed depending on your specific environment.
- [Integrating Components Using Declarative XML](#)
RCU provides extensibility with XML DTDs. Using these DTDs, component owners can integrate their components and prerequisites with RCU by providing a configuration file that adheres to the provided DTD.

About System Load and Product Load

Schema creation in RCU is performed in multiple phases; each phase requires a different level of access to your database.

- **System Load Phase**
During the system load phase, RCU creates the necessary tablespaces and schemas. If it is not already present, RCU also creates the `schema_version_registry`. One entry for each component is created in

`schema_version_registry`, and the entry will have the appropriate access and status set to **LOADED** in the `schema_version_registry` table.

User with SYS or SYSDBA privileges must perform these actions; authentication credentials must be provided on the [Database Connection Details](#) screen when running RCU.

If you do not have the necessary privileges, you can select the **Prepare Scripts for System Load** on the [Create Repository](#) screen. This option generates a SQL script that contains the same SQL statements and blocks that would have been called if RCU were to execute the actions for the selected components. After the script is generated, a user with the necessary SYS or SYSDBA privileges can execute the script to complete the system load phase.

After the system load phase is complete, any user with the privileges described in [Granting Permissions to Users for Querying System Load Objects](#) can then run RCU again to complete the schema creation by performing the product load phase.

 **Note:**

If you need to generate the script for system load, you can only create schemas on Oracle and Oracle EBR databases; the system load script is not supported on any other database.

If you are performing the system load with full SYS or SYSDBA privileges, then you can create the schemas on any certified database.

- **Product Load Phase**

During the product load phase, RCU creates procedures, functions, tables, indexes, and other objects within schemas and runs any action that does not require DBA access. Any non-DBA user or the REGISTRYOWNER user can be used for this step.

Before performing the product load phase, the user must be granted the following:

```
grant REGISTRYACCESS to user;  
grant STBROLE to user;
```

After the product load phase is complete, status is changed to **VALID**, if there is no post-product load for that component.

- **Optional Post-Product Load Phase**

This step is only needed for components that generate the `script_postDataLoad.sql` script that requires DBA privileges during the system load phase. After the product load phase is complete, someone with SYS or SYSDBA privileges should execute the `script_postDataLoad.sql` script.

This step is required for the following components:

- Audit Services (IAU)
- Oracle Enterprise Scheduler (ESS)

 **Note:**

The Repository Creation Utility fails to load schemas for SOAINFRA when you run schemas on an Oracle hardened database. See [SOAINFRA Schema Creation Fails on a Hardened Database](#).

Granting Permissions to Users for Querying System Load Objects

If you want to give a user with limited privileges the ability to query the system load objects, you must grant the user certain permissions before they can query the system load.

 **Note:**

This user will be used for connecting to the system for queries, but the generated scripts from the system load phase must be run by someone with DBA privileges.

```
grant select_catalog_role to user;  
grant select any dictionary to user;  
grant create session to user;  
grant select on schema_version_registry to user;
```

 **Note:**

You may encounter a *Table or view does not exist* error message when you execute the last command; this can be ignored.

After you perform the system load, grant the following permission to the same user in order for them to perform data load:

```
grant REGISTRYACCESS to user;  
grant STBROLE to user;
```

About Custom Prefixes

Use custom prefixes to organize and group schemas in your database.

 **Note:**

For important information regarding custom prefixes in IBM DB2 databases, see [Additional Requirements for IBM DB2 Databases](#).

The prefix is prepended to and separated from the schema name with an underscore (`_`) character, as shown below:

prefix_schemaname

Prefixes:

- Can only contain alpha-numeric characters; no spaces or other special characters.
- Must begin with a letter.
- Must not be longer than 12 characters.

 **Note:**

The total length of the prefixed schema name (your custom prefix plus the component schema name) must not be longer than 30 characters.

The default prefix used by RCU is `DEV`; if `DEV` has already been used, then RCU defaults to `DEV1`, then `DEV2`, and so on. Prefixes are used to create and organize logical groups of schemas. For example, you may want to create a test version of the Metadata Services (schema name `MDS`) called `TEST_MDS`; then, when you are ready for your production version, you can create a second version of the schema called `PROD_MDS`. Both `TEST_MDS` and `PROD_MDS` may reside on the same or separate databases.

You are only allowed to use a prefix once per schema within a single database. For example, if you have a version of the Metadata Services schema called `DEV_MDS`, then you cannot use the `DEV` prefix again to create another version of the Metadata Services schema (for example, `DEV_MDS2`).

If you want to create another version of the schema using the same prefix, you must first drop the existing schema and then create the schema again.

The mapping between the prefixes and schemas is maintained in `schema_version_registry`.

About the Service Table Schema

The Service Table schema is a special schema that is automatically created once per prefix. The service table stores basic schema configuration information (for example, schema prefixes and passwords) that can then be accessed and used by other Oracle Fusion Middleware components during domain creation.

For example, the configuration wizard has screens which you can configure to use the data that is stored in the service table when RCU was run. After you provide the service table schema credentials, the data from the service table is used to populate the fields on the screen, thus saving you the need to manually populate that data yourself.

Once created, service tables are used to wire Oracle Fusion Middleware components together. For more information about wiring various components together, see *Wiring Components to Work Together* in *Administering Oracle Fusion Middleware*.

Planning Your Schema Creation

Before you run RCU, make sure that you understand how schemas can be grouped together and distributed depending on your specific environment.

For more information, the following examples are provided:

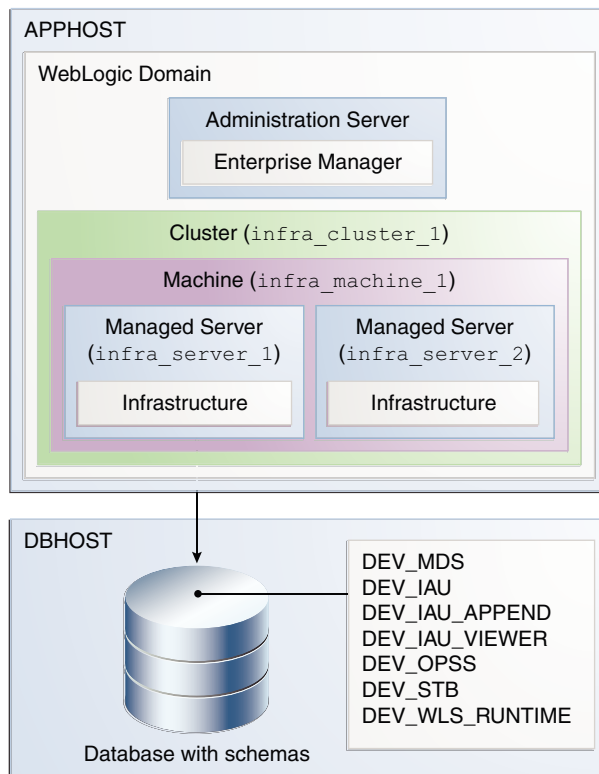
- [Organizing Schemas on a Single Database for a Single Domain](#)
This example shows a set of schemas in a single database being used by a single WebLogic domain.
- [Organizing Schemas on Multiple Databases for a Single Domain](#)
This example shows a single set of schemas that are distributed on multiple databases being used by a single WebLogic domain.
- [Organizing Schemas on a Single Database for Multiple Domains](#)
This example shows how schemas on a single database should be grouped for multiple domains.
- [Organizing Schemas on Multiple Databases for Multiple Domains](#)
This example shows how to organize schemas on multiple databases for use with multiple WebLogic domains.

Organizing Schemas on a Single Database for a Single Domain

This example shows a set of schemas in a single database being used by a single WebLogic domain.

This is the simple scenario in which all schemas using the `DEV` prefix are grouped together and used by a single WebLogic domain.

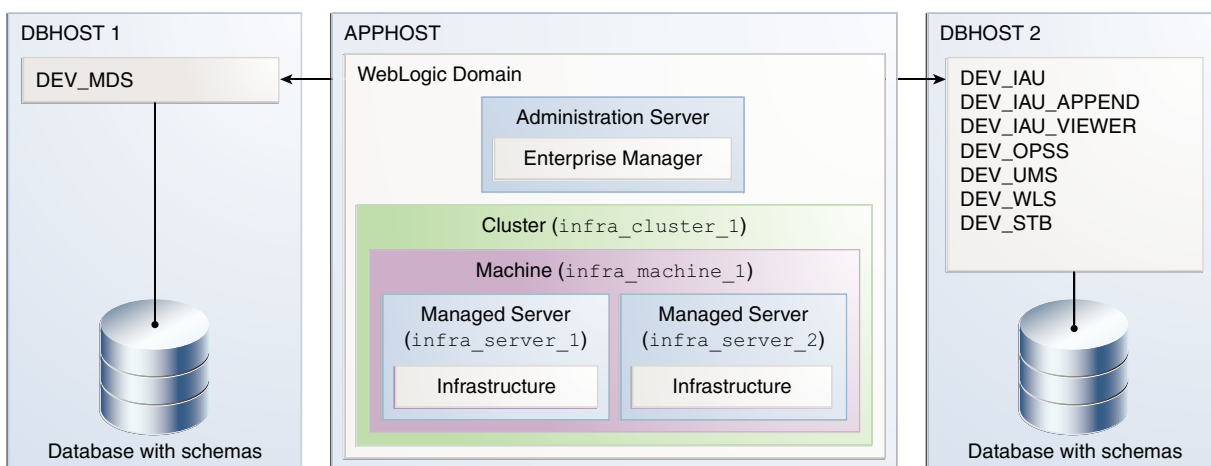
Figure 1-1 Schemas on a Single Database for a Single Domain



Organizing Schemas on Multiple Databases for a Single Domain

This example shows a single set of schemas that are distributed on multiple databases being used by a single WebLogic domain.

Figure 1-2 Schemas on Multiple Databases for a Single Domain

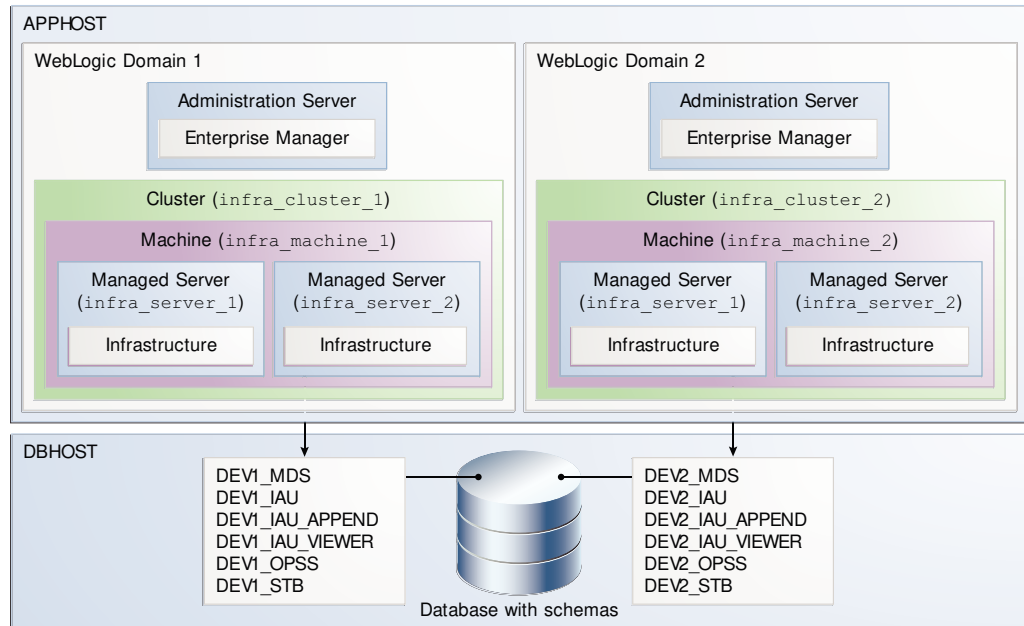


You can use the same schema prefix (in this case, `DEV`) can be used to group these related schemas together, even across multiple databases.

Organizing Schemas on a Single Database for Multiple Domains

This example shows how schemas on a single database should be grouped for multiple domains.

Figure 1-3 Schemas on a Single Database for Multiple Domains



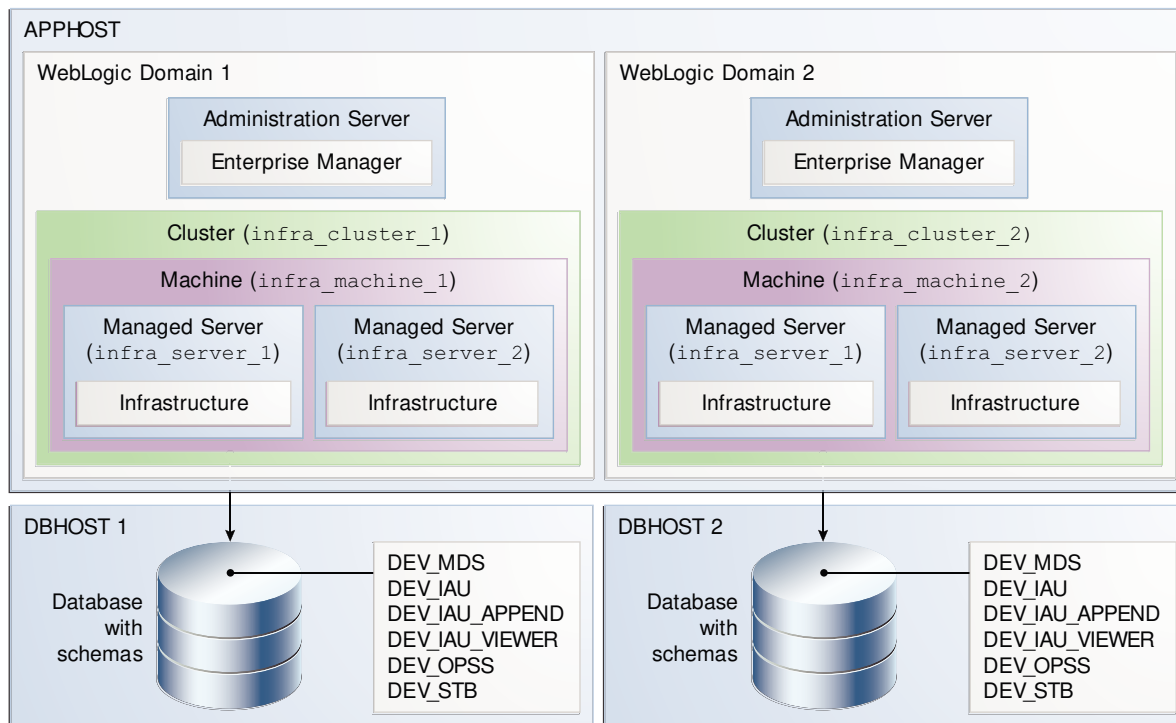
In this example, the prefixes are grouped together by using `DEV1` for one set of schemas (used by WebLogic Domain 1), and `DEV2` for the second set (used by WebLogic Domain 2).

It is not possible to share a single set of schemas across multiple domains; each domain must have its own set of schemas.

Organizing Schemas on Multiple Databases for Multiple Domains

This example shows how to organize schemas on multiple databases for use with multiple WebLogic domains.

Figure 1-4 Schemas on Multiple Databases for Multiple Domains



In this scenario, it is possible to have separate domains on the same host that use schemas with the same name and prefix (*DEV*), since the schemas are located on different databases.

Integrating Components Using Declarative XML

RCU provides extensibility with XML DTDs. Using these DTDs, component owners can integrate their components and prerequisites with RCU by providing a configuration file that adheres to the provided DTD.

For more information about integrating various components, see [Extending Repository Creation Utility to Configure Custom Application Repositories](#).

Using RCU with Java Access Bridge (Windows Only)

Java Access Bridge enables assistive technologies, such as JAWS screen reader, to read Java applications that run on the Windows platform.

Assistive technologies can read Java-based interfaces, such as Oracle Universal Installer and Oracle Enterprise Manager.

For more information, see the following topics:

- [Install Java Access Bridge](#)
Follow these steps to download and install Java Access Bridge.

- [Configure RCU to Use Java Access Bridge](#)
To configure RCU to use Java Access Bridge after you complete the installation, set the system variable `ORACLE_OEM_CLASSPATH` to point to the installed Java Access Bridge files.

Install Java Access Bridge

Follow these steps to download and install Java Access Bridge.

1. Download Java Access Bridge from the following URL:
<http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136191.html>
2. For complete information and procedure about Java Access Bridge installation, see [Java Access Bridge Installation procedure](#).

Configure RCU to Use Java Access Bridge

To configure RCU to use Java Access Bridge after you complete the installation, set the system variable `ORACLE_OEM_CLASSPATH` to point to the installed Java Access Bridge files.

1. Display **System** in the Control Panel.
2. Select the **Advanced** tab.
3. Click **New** button in the System Variable list. The New System Variable dialog appears.
4. In the Variable Name field, enter `ORACLE_OEM_CLASSPATH`.
5. In the Variable Value field, enter the full path for `theaccess-bridge.jar` and `jaccess-1_4.jar`.

Use a semicolon to separate the two paths. Do not use quotes or character spaces.

6. Click **OK**.

2

Obtaining and Running Repository Creation Utility

Use these instructions to obtain and run the Repository Creation Utility (RCU) to create and drop schemas.

- [Obtaining RCU](#)
In 12c, RCU is available with the Oracle Fusion Middleware Infrastructure distribution.
- [Starting RCU](#)
After you install Oracle Fusion Middleware Infrastructure and create your Oracle home, you can start RCU from the `ORACLE_HOME/oracle_common/bin` directory.
- [Creating Schemas](#)
Follow these instructions to create schemas in the database and verify that they are installed properly.
- [Dropping Schemas](#)
Follow these instructions to drop schemas from the database.

Obtaining RCU

In 12c, RCU is available with the Oracle Fusion Middleware Infrastructure distribution.

For information about how to install and obtain RCU, see *Installing the Infrastructure Software* in *Installing and Configuring the Oracle Fusion Middleware Infrastructure*.

Starting RCU

After you install Oracle Fusion Middleware Infrastructure and create your Oracle home, you can start RCU from the `ORACLE_HOME/oracle_common/bin` directory.

Note:

- If you run the RCU by using a non-English database, you will need to set the following language environment variables: `LANG`, `LC_ALL`, and `NLS_LANG`. Use the environment commands that are appropriate for your environment.

For example, for UNIX operating systems running `csh` run the following:

```
setenv LANG en_US.UTF8
setenv LC_ALL $LANG
setenv NLS_LANG american_america
```

- To start RCU, you do not need to set the `JAVA_HOME` environment variable. But, in case you want to use a specific JDK, ensure that the `JAVA_HOME` environment variable is set to the location of a certified JDK on your system.

For example, on UNIX operating systems, if your JDK is located in `/home/Oracle/Java/`:

```
setenv JAVA_HOME /home/Oracle/Java/jdk1.8.0_131
```

Be sure to replace the JDK location in this example with the actual JDK location on your system.

On Linux operating systems:

```
cd ORACLE_HOME/oracle_common/bin
./rcu
```

On Windows operating systems:

```
cd ORACLE_HOME\oracle_common\bin
rcu.bat
```

RCU provides a command-line interface in situations where Xserver is not available or you have access to telnet terminals without display capabilities. The command-line interface also allows you to embed RCU from command-line scripts or with some Oracle Fusion Middleware components (for example, Enterprise Manager).

For more information about using the Command Line Interface (CLI), see [Running Repository Creation Utility from the Command Line](#).

Creating Schemas

Follow these instructions to create schemas in the database and verify that they are installed properly.

- [Creating Schemas as a User with Full SYS or SYSDBA Privileges](#)
If you are a user with full SYS or SYSDBA privileges and are able to provide valid authentication credentials for database access, follow these instructions to create schemas.
- [Creating Schemas as a User With Limited Database Privileges](#)
If you are a user without SYS or SYSDBA privileges, follow these instructions to create schemas.
- [Verifying Schema Version Numbers](#)
When the schemas are created in your database, RCU creates and maintains a table called `schema_version_registry`. This table contains schema information such as version number, component name and ID, date of creation and modification, and custom prefix.

Creating Schemas as a User with Full SYS or SYSDBA Privileges

If you are a user with full SYS or SYSDBA privileges and are able to provide valid authentication credentials for database access, follow these instructions to create schemas.

Click the screen name to see detailed information for that screen. Unless otherwise noted, click **Next** to continue to the next screen.

Table 2-1 Schema Creation Steps for Full-Privilege Users

Screen	Description
Welcome	This screen introduces you to RCU.
Create Repository	Select Create Repository , then select System Load and Product Load . If you do not have full SYS or SYSDBA privileges on the database, select Prepare Scripts for System Load and follow the schema creation instructions in Creating Schemas as a User With Limited Database Privileges .
Database Connection Details	Specify your database connection credentials. Remember that if you are creating schemas on an IBM DB2 database, you must have already created one operating system user for each schema that you want to create. See Additional Requirements for IBM DB2 Databases for more information. Click Next when you have specified your credentials. A separate dialog window appears while RCU checks connectivity and the database prerequisites. When the database checking has passed without errors, click OK to dismiss the dialog window and go to the next screen.
Select Components (for Create Operation)	Select the components for which you want to create schemas, and specify a prefix to group them together. You must remember the prefix and schema names for the components you are installing; you need this information during the configuration phase of your product installation. Oracle recommends that you write down these values.
Schema Passwords	Specify the passwords for your schema owners. You must remember the passwords you enter on this screen; you need this information during the configuration phase of your product installation. Oracle recommends that you write down these values.

Table 2-1 (Cont.) Schema Creation Steps for Full-Privilege Users

Screen	Description
Custom Variables	<p>Use the Custom Variables screen to specify additional product configuration information.</p> <p>The Custom Variables screen appears only if you selected one or more of the following components on the Select Components screen. You can select a component on the Select Components screen only if the Oracle home in which RCU was started contains that component.</p> <ul style="list-style-type: none"> • If you selected Oracle Data Integrator, the Custom Variables for Oracle Data Integrator are visible. • If you selected SOA Suite, the Custom Variables for Oracle SOA Suite are visible. • If you selected WebCenter Portal - Analytics, the Custom Variable for Oracle WebCenter Portal Analytics is visible. • If you selected Oracle GoldenGate - Repository, the Custom Variables for Oracle GoldenGate Studio are visible. • If you selected Oracle Data Integrator, SOA Suite, and WebCenter Portal - Analytics, the custom variables for Oracle Data Integrator, Oracle SOA Suite, and Oracle WebCenter Portal Analytics are visible on this screen.
Map Tablespaces	<p>Use this screen to configure the desired tablespace mapping for the schemas that you want to create.</p> <p>When you click Next, a separate dialog window appears asking you to confirm that you want to create these tablespaces. Click OK to proceed and dismiss the dialog window.</p> <p>A second dialog window appears showing the progress of tablespace creation. After the tablespaces are created, click OK to dismiss this window and go to the next screen.</p>
Summary (for Create Operation)	Verify the information on this screen, then click Create to begin schema creation.
Completion Summary (for Create Operation)	Review the information on this screen to verify that the operation was completed successfully. Click Close to complete the schema creation and dismiss RCU.

Creating Schemas as a User With Limited Database Privileges

If you are a user without SYS or SYSDBA privileges, follow these instructions to create schemas.

Note:

To create schemas in PDB using a user other than SYSDBA, RCU users need to create a common user and grant DBA to that common user, before running RCU. Local user with DBA privilege will not work correctly.

Table 2-2 Schema Creation Steps for Limited-Privilege Users

Screen	Description
Welcome	This screen introduces you to RCU.

Table 2-2 (Cont.) Schema Creation Steps for Limited-Privilege Users


Screen	Description
Create Repository	Select Create Repository , then select Prepare Scripts for System Load . See About System Load and Product Load for more information about these operations.
Database Connection Details	Specify your database connection credentials. Click Next when you have specified your credentials. A separate dialog window appears while RCU checks connectivity and the database prerequisites. When the database checking has passed without errors, click OK to dismiss the dialog window and go to the next screen.
	<div style="border: 1px solid #0070c0; padding: 10px; background-color: #e6f2ff;"> <p> Note:</p> <p>Performing system load and product load separately is only supported on Oracle and Oracle EBR databases.</p> </div>
Select Components (for Create Operation)	Select the components for which you want to create schemas, and specify a prefix to group them together. You must remember the prefix and schema names for the components you are installing; you need this information during the configuration phase of your product installation. Oracle recommends that you write down these values.
Custom Variables	Use the Custom Variables screen to specify additional product configuration information. The Custom Variables screen appears only if you selected one or more of the following components on the Select Components screen. You can select a component on the Select Components screen only if the Oracle home in which RCU was started contains that component. <ul style="list-style-type: none"> • If you selected SOA Suite, the Custom Variables for Oracle SOA Suite are visible. • If you selected WebCenter Portal - Analytics, the Custom Variable for Oracle WebCenter Portal Analytics is visible. • If you selected SOA Suite and WebCenter Portal - Analytics, the custom variables for Oracle SOA Suite and Oracle WebCenter Portal Analytics are visible on this screen.
Map Tablespaces	Use this screen to configure the desired tablespace mapping for the schemas that you want to create.
Summary (for Create Operation)	Verify the information on this screen, then click Generate to begin script generation. By default, the scripts are generated in the <code>RCUdate_timestamp_random_number/logs</code> directory inside the temporary directory on your system. For example, on UNIX operating systems, the scripts are generated in the <code>/tmp/RCUdate_timestamp_random_number/logs</code> directory, by default. If you want to specify a different location, click Browse and select a location on your system. The names of the scripts generated are <code>script_systemLoad.sql</code> and <code>script_postDataLoad.sql</code> .
Completion Summary (for Create Operation)	Review the information on this screen to verify that the operation was completed successfully. Click Close to complete the script generation and dismiss RCU.

Table 2-2 (Cont.) Schema Creation Steps for Limited-Privilege Users

Screen	Description
N/A	<p>After the scripts are created, user with SYS or SYSDBA privileges must execute the <code>script_systemLoad.sql</code> script:</p> <ol style="list-style-type: none"> 1. Sign in to SQL*Plus. 2. Enter the following command to execute the <code>script_systemLoad.sql</code> script (replace the path to the script if you chose to save it in a custom location): <code>@/tmp/RCUdate_timestamp_random_number/logs/script_systemLoad.sql</code> 3. While the script is running, the user is prompted to set the schema password for each schema being created. Specify the passwords as prompted. You are returned to your system prompt when the script has been executed.
N/A	After the script is executed, start RCU again and perform the product load phase to complete schema creation.
Welcome	This screen introduces you to RCU.
Create Repository	<p>Select Create Repository, then select Perform Product Load. See About System Load and Product Load for more information about these operations.</p>
Database Connection Details	<p>Specify your database connection credentials. Click Next when you have specified your credentials. A separate dialog window appears while RCU checks connectivity and the database prerequisites. When the database checking has passed without errors, click OK to dismiss the dialog window and go to the next screen.</p>
Select Components (for Create Operation)	<p>From the drop-down list, select the prefix you specified earlier in this procedure. Then, select the components for which you want to perform the product load.</p>
Schema Passwords	<p>Specify the passwords for your schema owners. Ensure to specify the same passwords that were entered while running the <code>script_systemLoad.sql</code> script through SQL*Plus. You must remember the passwords you enter on this screen; you need this information during the configuration phase of your product installation. Oracle recommends that you write down these values.</p>
Custom Variables	<p>Use the Custom Variables screen to specify additional product configuration information. The Custom Variables screen appears only if you selected one or more of the following components on the Select Components screen. You can select a component on the Select Components screen only if the Oracle home in which RCU was started contains that component.</p> <ul style="list-style-type: none"> • If you selected Oracle Data Integrator, the Custom Variables for Oracle Data Integrator are visible. • If you selected SOA Suite, the Custom Variables for Oracle SOA Suite are visible. • If you selected WebCenter Portal - Analytics, the Custom Variable for Oracle WebCenter Portal Analytics is visible. • If you selected Oracle GoldenGate - Repository, the Custom Variables for Oracle GoldenGate Studio are visible. • If you selected Oracle Data Integrator, SOA Suite, and WebCenter Portal - Analytics, the custom variables for Oracle Data Integrator, Oracle SOA Suite, and Oracle WebCenter Portal Analytics are visible on this screen.

Table 2-2 (Cont.) Schema Creation Steps for Limited-Privilege Users

Screen	Description
Summary (for Create Operation)	Verify the information on this screen, then click Data Load to begin schema creation.
Completion Summary (for Create Operation)	Review the information on this screen to verify that the operation was completed successfully. Click Close to complete the schema creation and dismiss RCU.
N/A	<p>This optional step is needed for components that generate and must execute product load scripts requiring DBA privileges. After the product load phase is complete, someone with SYS or SYSDBA privileges must execute the <code>script_postDataLoad.sql</code> script.</p> <p>This step is needed for the Audit Services (IAU) and Oracle Enterprise Scheduler (ESS) components.</p> <ol style="list-style-type: none"> 1. Sign in to SQL*Plus. 2. Enter the following command to execute the <code>script_postDataLoad.sql</code> script (replace the path to the script if you chose to save it in a custom location): <pre>@/tmp/RCUdate_timestamp_random_number/logs/script_postDataLoad.sql</pre> <p>You are returned to your system prompt when the script has been executed.</p>

Verifying Schema Version Numbers

When the schemas are created in your database, RCU creates and maintains a table called `schema_version_registry`. This table contains schema information such as version number, component name and ID, date of creation and modification, and custom prefix.

To verify that the schemas are installed properly, run the following query after you sign in to SQL*Plus:

```
select comp_name, version from schema_version_registry;
```

The `comp_name` argument retrieves the name of the component, and `version` retrieves the version number. For example:

```
SQL> select comp_name, version from schema_version_registry;
```

```

COMP_NAME
-----
VERSION
-----
Audit Service
12.2.1.2.0

Audit Service Append
12.2.1.2.0

Audit Service Viewer
12.2.1.2.0

COMP_NAME
```

```

-----
VERSION
-----
Metadata Services
12.2.1.3.0

Oracle Platform Security Services
12.2.1.0.0

Service Table
12.2.1.3.0

COMP_NAME
-----
VERSION
-----
User Messaging Service
12.2.1.0.0

WebLogic Services
12.2.1.0.0

8 rows selected.

```

Dropping Schemas

Follow these instructions to drop schemas from the database.

To see detailed information for a particular screen, click on the screen name. Unless otherwise noted, click **Next** to continue to the next screen.

Table 2-3 RCU Screens and Description for Dropping Schemas

Screen	Instructions and Action Required
Welcome	This screen introduces you to RCU.
Create Repository	Select Drop Repository .
Database Connection Details	Specify your database connection credentials. Click Next when you have specified your credentials. A separate dialog window appears while RCU checks connectivity and the database prerequisites. When the database checking has passed without errors, click OK to dismiss the dialog window and go to the next screen.
Select Components (for Drop Operation)	Select the prefix and the schemas that you want to drop, then click Next . A separate dialog window appears asking you to verify that you want to drop the selected schemas. Click OK to dismiss this window. A second dialog window appears while RCU checks the prerequisites for the schemas that you are dropping. After prerequisite checking is complete, click OK to dismiss this window and go to the next screen.

Table 2-3 (Cont.) RCU Screens and Description for Dropping Schemas

Screen	Instructions and Action Required
Custom Variables	<p>The Custom Variables screen appears only if you selected one or more of the following components on the Select Components screen. You can select a component on the Select Components screen only if the Oracle home in which RCU was started contains that component.</p> <ul style="list-style-type: none"> If you selected Oracle Data Integrator, the Custom Variables for Oracle Data Integrator are visible. If you selected SOA Suite, the Custom Variables for Oracle SOA Suite are visible. If you selected WebCenter Portal - Analytics, the Custom Variable for Oracle WebCenter Portal Analytics is visible. If you selected Oracle Data Integrator, SOA Suite, and WebCenter Portal - Analytics, the custom variables for Oracle Data Integrator, Oracle SOA Suite, and Oracle WebCenter Portal Analytics are visible on this screen.
Summary (for Drop Operation)	Review the information on this screen, then click Drop to drop the schemas.
Completion Summary (for Drop Operation)	Note the location of the log files, then click Close to dismiss the screen.

- [About Dropping Shared Tablespaces](#)
Tablespaces that are shared among multiple schemas are not dropped.
- [Dropping Schemas and Deleting Datafiles \(Windows Only\)](#)
If you used RCU to drop a schema from a Windows-based database, and you want to recreate the dropped schema, you will have to manually delete the datafiles that were not automatically removed when the schema was dropped.

About Dropping Shared Tablespaces

Tablespaces that are shared among multiple schemas are not dropped.

For example, if you created both the Audit Services (for example, `DEV_IAU`) and Metadata Services (for example, `DEV_MDS`) schemas, both schemas would use the temporary tablespace `DEV_IAS_TEMP` (see [Default Tablespace Mappings](#)).

If you then drop the `DEV_IAU` schema, the `DEV_IAS_TEMP` tablespace would not be dropped since it is also used by the `DEV_MDS` schema.

Dropping Schemas and Deleting Datafiles (Windows Only)

If you used RCU to drop a schema from a Windows-based database, and you want to recreate the dropped schema, you will have to manually delete the datafiles that were not automatically removed when the schema was dropped.

Change to the `DB_HOME\oradata` directory and manually delete any remaining datafiles before you recreate the schema.

Oracle recommends that you check this directory for any remaining datafiles before you attempt to recreate any dropped schema.

3

Running Repository Creation Utility from the Command Line

The command-line interface (CLI) is necessary for integration with both the Oracle Fusion Middleware installer and Enterprise Manager during application deployment. Additionally, you can use the CLI in cases where Xserver is not configured or if you are using a telnet terminal that does not have proper display capabilities. For more information on how to run RCU from the command line, see the following topics:

- [Command-Line Syntax and Parameters](#)
To run RCU from the command line, you need to specify a mode, an operation, and any necessary parameters. Make sure that you understand the syntax and the various command-line parameters you can use for your operation.
- [Using the -silent Command](#)
Use the `-silent` command if you want to run RCU with minimal interaction once you have entered the command.
- [Using the -interactive Command](#)
Use the `-interactive` command to run the RCU graphical interface. This is the default value if neither `-silent` nor `-interactive` is specified.
- [Using Response Files](#)
Before you run RCU, you can provide information that is required to perform a specific operation in a response file. For example, you can provide your database connection details, a list of components to be added or dropped, and the schema prefix to RCU through a response file.
- [Creating a Repository from the Command Line](#)
If you are a user with full SYS or SYSDBA privileges, use the `-createRepository` operation to create a repository.
- [Generating a System Load Script From the Command Line](#)
If you are a user without SYS or SYSDBA privileges, use the `-generateScript` operation to generate a script. A user with DBA privileges can run the script later to complete the system load phase.
- [Loading Data Into the Repository From the Command Line](#)
After the system load scripts are run by a user with DBA privileges, any user can then use the `-dataLoad` operation to load data into a repository.
- [Dropping a Repository from the Command Line](#)
Use the `-dropRepository` operation to drop a repository.
- [Generating a List of Components in an Oracle Home from the Command Line](#)
Use the `-listComponents` command to generate a list of valid components that are available in a given Oracle home. This command displays the name, component ID, and the supported database types for each component that is available in the Oracle home in which RCU was started.

- [Generating a List of Schemas in a Database from the Command Line](#)
Use the `-listSchemas` command to generate a list of valid schemas in a given database.
- [RCU Environment Variables](#)
The RCU environment variables are the variables picked up by RCU from the environment. If an environment variable is not set, then RCU uses the default value.

Command-Line Syntax and Parameters

To run RCU from the command line, you need to specify a mode, an operation, and any necessary parameters. Make sure that you understand the syntax and the various command-line parameters you can use for your operation.

The syntax for the RCU command-line interface is:

```
rcu mode operation {parameters}
```

There are two modes (`-silent` and `-interactive`) and eight operations (`-generateResponseFile`, `-responseFile`, `-createRepository`, `-generateScript`, `-dataLoad`, `-dropRepository`, `-listComponents`, and `-listSchemas`). So, the command can be written out as follows:

```
rcu [-silent | -interactive] [-generateResponseFile | -responseFile | -
createRepository | -generateScript | -dataLoad |
-dropRepository | -listComponents | -listSchemas]
{parameters}
```

The following topics describe each of the modes, operations, and parameters:

- [RCU Modes Available from the Command Line](#)
There are two RCU modes available from the command line, `-silent` mode and `-interactive` mode.
- [RCU Operations Available from the Command Line](#)
There are eight RCU operations that are available from the command-line.
- [RCU Command Line Parameters](#)
You can pass the RCU command-line parameters to RCU as input by directly entering the parameters on the command line or by using a response file.

RCU Modes Available from the Command Line

There are two RCU modes available from the command line, `-silent` mode and `-interactive` mode.

Table 3-1 RCU Modes Available from the Command Line

Command	Description
<code>-silent</code>	Run RCU with minimal or no interaction from the command line. For more information about silent mode, see Using the -silent Command .

Table 3-1 (Cont.) RCU Modes Available from the Command Line

Command	Description
-interactive	Run the RCU graphical interface. This is the default if neither <code>-silent</code> nor <code>-interactive</code> is specified. This command (whether specified or not) allows you to prepopulate certain screens with the information that is specified in the response file or from the command line. You can prepopulate the Create Repository and Database Connection Details screens. For more information about graphical mode, see Using the -interactive Command .

RCU Operations Available from the Command Line

There are eight RCU operations that are available from the command-line.

Table 3-2 RCU Operations Available from the Command Line

Command	Description
-generateResponseFile	Run RCU with <code>-silent</code> to generate a RCU response file that contains all valid RCU command line parameters. See Creating a RCU Response File from the Command Line
-responseFile	Run RCU with either <code>-silent</code> or <code>-interactive</code> to perform a RCU operation using a response file. See Performing RCU Operations by Using a Response File
-createRepository	Run RCU with either <code>-silent</code> or <code>-interactive</code> to create a repository. See Creating a Repository from the Command Line .
-generateScript	Run RCU with either <code>-silent</code> or <code>-interactive</code> to generate a script for system load. See Generating a System Load Script From the Command Line .
-dataLoad	Run RCU with either <code>-silent</code> or <code>-interactive</code> to load data into the repository. See Loading Data Into the Repository From the Command Line .
-dropRepository	Run RCU with either <code>-silent</code> or <code>-interactive</code> to drop a repository. See Dropping a Repository from the Command Line .
-listComponents	Run RCU with <code>-silent</code> to generate a list of components for a given Oracle home. See Generating a List of Components in an Oracle Home from the Command Line .
-listSchemas	Run RCU with <code>-silent</code> to list the schemas in a given database. See Generating a List of Schemas in a Database from the Command Line .

RCU Command Line Parameters

You can pass the RCU command-line parameters to RCU as input by directly entering the parameters on the command line or by using a response file.

Table 3-3 RCU Command Line Parameters and Descriptions

Parameter	Required?	Valid for Which Operation?	Description
-compInfoXMLLocation	No	-createRepository -generateScript -dataLoad -dropRepository	The full path to the location of the ComponentInfo.xml file. The default location is <i>ORACLE_HOME/oracle_common/rcu/config</i> (on UNIX operating systems) or <i>ORACLE_HOME\oracle_common\rcu\config</i> (on Windows operating systems).
-storageXMLLocation	No	-createRepository -generateScript -dataLoad -dropRepository	The full path to the location of the StorageInfo.xml file. The default location is <i>ORACLE_HOME/oracle_common/rcu/config</i> (on UNIX operating systems) or <i>ORACLE_HOME\oracle_common\rcu\config</i> (on Windows operating systems).
-databaseType	No	-createRepository -generateScript -dataLoad -dropRepository -listSchemas	The type of database to which you are connecting. If you perform the system load and product load concurrently as a user with full DBA privileges, the valid options are ORACLE, SQLSERVER, IBMBD2, MYSQL, or EBR. If you do not have permissions to perform the system load and product load concurrently and need to generate a script, you can only use the ORACLE and EBR database types. For more information about system load and product load, see About System Load and Product Load .
-connectString	Yes	-createRepository -generateScript -dataLoad -dropRepository -listSchemas	The credentials to connect to your database. For Oracle or EBR-enabled databases, use the following format: <i>host:port:sid</i> OR: <i>host:port/service</i> For all other database types, use: <i>server_name/host:port:database_name</i>
-edition	No	-createRepository -generateScript -dataLoad -dropRepository -listSchemas	The edition name. This parameter is only valid if you specify <code>databaseType=EBR</code> .

Table 3-3 (Cont.) RCU Command Line Parameters and Descriptions

Parameter	Required?	Valid for Which Operation?	Description
-dbUser	Yes	-createRepository -generateScript -dataLoad -dropRepository -generateResponseFile -listSchemas	The database user name. For example, the SYS user on Oracle databases). This parameter is required if the -createWallet parameter is provided during the -generateResponseFile operation. RCU uses the database user name as the key or alias to store the database user password when generating a wallet. If the -createWallet parameter is not provided, the database user name is optional.
-dbRole	No	-createRepository -generateScript -dataLoad -dropRepository -listSchemas	The database user role. For example, SYSDBA for the SYS user on Oracle databases). This parameter is needed only if -dbUser is specified and the role is SYSDBA.
-unicodeSupport	No	-createRepository -dropRepository	Specify either Yes or No for unicode support. Default parameter is Yes. This parameter is only valid if you specify databaseType=SQLSERVER.
-skipCleanupOnFailure	No	-createRepository -generateScript -dataLoad	If the schema creation fails, specify whether you want to skip the schema cleanup. The valid values are Yes or No. The default value is No.
-skipTablespaceDropOnFailure	No	-createRepository -generateScript -dataLoad	If schema creation fails, whether or not you want to skip the schema cleanup. The valid values are Yes or No. The default value is No. Specify Yes to retain tablespaces for failed components during schema cleanup. Specify No to drop tablespaces for failed components during schema cleanup.
-scriptLocation	No	-generateScript	Specify the location to save the generated script for system load repository.
-useSamePasswordForAllSchemaUsers	No	-createRepository -dataLoad -generateResponseFile	Whether you want to use the same password for all schemas. The valid values are true or false. The default value is false.
-selectDependentsForComponents	No	-createRepository -generateScript -dataLoad -dropRepository -generateResponseFile	Whether you want to have RCU automatically select dependent schemas for your components. The valid values are true or false. The default value is false. STB schema is created independent of this switch, whether the value is true or false

Table 3-3 (Cont.) RCU Command Line Parameters and Descriptions

Parameter	Required?	Valid for Which Operation?	Description
-honorOMF	No	-createRepository -generateScript	If the value is set to true, RCU creates datafiles using the Oracle-Managed Files (OMF) naming format. The valid values are true or false. The default value is false.
-encryptTablespace	No	-createRepository -generateScript	Specify whether you want to encrypt all new tablespaces that are created by RCU. This option is valid only if you have TDE (Transparent Data Encryption) enabled in the database (Oracle or Oracle EBR) when you start RCU. TDE tablespace encryption allows you to encrypt sensitive data stored in tablespaces. The valid values are true or false. The default value is false. If the value is set to true, <i>all new</i> tablespaces created by RCU are encrypted only if TDE is enabled in the database. Tablespaces that were created before RCU was launched cannot be encrypted by using this option.
-skipTablespaceDrop	No	-dropRepository	This parameter allows you to skip dropping tablespaces during the -dropRepository operation. The valid values are Yes or No. Specify Yes to skip tablespace drop. The default value is No.
-variables	No	-createRepository -generateScript -dataLoad -dropRepository	Comma-separated variables in the format <i>variablename=value</i> . See RCU Environment Variables for a list of RCU environment variables.
-schemaPrefix	No	-createRepository -generateScript -dataLoad -dropRepository -generateResponseFile	Prefix for your schemas. For more information about schema prefixes, see Planning Your Schema Creation and Select Components (for Create Operation) . This parameter is required for the -createRepository, -generateScript, -dataLoad, and -dropRepository operations. Also, this parameter is required if the -createWallet parameter is provided during the -generateResponseFile operation. RCU uses the prefixed schema name as the key or alias to store schema passwords when generating a wallet. If the -createWallet parameter is not provided, the schema prefix is optional.

Table 3-3 (Cont.) RCU Command Line Parameters and Descriptions

Parameter	Required?	Valid for Which Operation?	Description
-component	Yes	-createRepository -generateScript -dataLoad -dropRepository	<p>ID of the component that you want to add or drop. To specify a single component, use the format:</p> <pre>-component component_ID</pre> <p>To specify multiple components, use the format:</p> <pre>-component component_ID -component component_ID</pre> <p>For a list of valid component (schema) IDs, see About Repository Creation Utility Schemas, IDs, and Tablespaces. Multiple components are sometimes necessary because of dependencies; for example, you can not create the Oracle Platform Security Services (OPSS) schema if the Audit Services (IAU) schema does not exist. In this case, you must specify both schemas:</p> <pre>-component OPSS -component IAU</pre>
-tablespace	No	-createRepository -generateScript	Tablespace name of the component. This tablespace must exist in the database if you are using this option.
-tempTablespace	No	-createRepository -generateScript	TEMP tablespace name of the component. This tablespace must exist in the database if you are using this option.
-createWallet	No	-generateResponseFile	<p>Provide this parameter if you want to create a wallet file to securely store your RCU passwords, such as the database user password, schema passwords, and custom variable passwords.</p> <p>RCU accepts password credentials stored in an Oracle Wallet. That is, you can use the generated wallet later in silent situations to provide your passwords to RCU without having to directly enter any passwords on the command line.</p> <p>The wallet is generated in the same directory as the response file.</p>

Table 3-3 (Cont.) RCU Command Line Parameters and Descriptions

Parameter	Required?	Valid for Which Operation?	Description
-walletDir	No	-createRepository -generateScript -dataLoad -dropRepository -listSchemas	<p>The absolute path of the wallet directory where your RCU passwords that are used for creating or dropping schemas are stored. If you choose to use a wallet to pass passwords to RCU, RCU retrieves the required passwords from the specified directory. This option is valid for silent mode only.</p> <p>If you use this option, the wallet must already exist and contain the required passwords. If the wallet does not contain the required passwords, RCU prompts you for the missing passwords on the command line.</p> <p>This directory should contain a valid <code>cwallet.sso</code> file. RCU supports auto-login wallets only (password is not needed to open wallet).</p>
-responseFileDir	No	-generateResponseFile	<p>The absolute path of the response file directory. If specified, RCU generates a response file named <code>rcuResponseFile.properties</code> in the given directory.</p> <p>If not specified, RCU generates a response file in the <code>\$HOME/.rcu</code> directory by default.</p>
-componentList	No	-generateResponseFile	<p>Comma-separated list of component IDs for the components in the Oracle home that you want to add to the response file. To generate a list of valid component IDs for a specific Oracle home, see Generating a List of Components in an Oracle Home from the Command Line.</p> <p>If not specified, all valid components in the Oracle home are added to the response file by default.</p>
-validate	No	-createRepository -generateScript -dataLoad -dropRepository	<p>Use this option to execute validation checks in silent mode and verify that all prerequisites for the specified operation have been met.</p> <p>With this option, RCU only checks for the prerequisites that are required to perform the operation and then exits. For more information about validate parameter, see Using the -validate Parameter to Verify Prerequisites.</p>
-schemaPrefixes	No	-listSchemas	Comma-separated list of schema prefixes. If provided, RCU only lists the schemas with the given prefixes.

Using the `-silent` Command

Use the `-silent` command if you want to run RCU with minimal interaction once you have entered the command.

You must specify all mandatory command-line parameters in the command. For example:

```
rcu -silent -createRepository -connectString database_connect_string -dbUser database_user -component component
```

In this scenario, RCU prompts you for the database and component schema passwords from the command line. If you specify multiple components, you are prompted for the passwords in the order in which the components are specified.

Instead of directly entering all the RCU parameters on the command line, you can provide the location and name of a file that contains certain input values. This file (called a response file) allows you to pass values for all valid command-line parameters to RCU through a text file. When you run RCU with a response file, all values are read from the file to create or drop your schemas. For more information about response file, see [Using Response Files](#).

If you want to avoid all interaction from the command line, you can create a text file contains all the necessary passwords (one password per line) and then use the `-f` option to pass this password file to RCU. For example, if you create a file called `passwordfile.txt`, you can use the command below:

```
rcu -silent -createRepository -connectString database_connect_string -dbUser database_user -component component1_name -component component2_name -f < passwordfile.txt
```

The `passwordfile.txt` file would contain, in order:

```
database_password  
component1_schema_password  
component2_schema_password
```

When schemas are created for multiple components, `-tablespace` and `-tempTablespace` has to be declared for each component individually.

For example:

```
./rcu -silent -createRepository -connectString <connectString> -dbUser SYS -dbRole SYSDBA -schemaPrefix DEV01 -selectDependentsForComponents true \  
-component MDS -tablespace FMW_INFRA -tempTablespace TEMP \  
-component IAU -tablespace FMW_INFRA -tempTablespace TEMP \  
-component IAU_APPEND -tablespace FMW_INFRA -tempTablespace TEMP \  
-component IAU_VIEWER -tablespace FMW_INFRA -tempTablespace TEMP \  
-component STB -tablespace FMW_INFRA -tempTablespace TEMP -f <
```

It is important to make sure that the passwords in the file are specified in the same order as the components on the command line.

Once the installation is complete, the password file must be removed. The passwords are maintained in cleartext format, and therefore present a security risk if the password file is left in place after installation.

Instead of using a plain text password file, you can also create an Oracle Wallet file that contains all the necessary passwords and then use the `-walletDir` parameter to pass this file to RCU. In this case, RCU does not prompt you for any passwords that are stored in the wallet. For more information about using wallet, see [Using a Wallet to Store Passwords](#).

For more information on running RCU in `-silent` mode, see the following topics:

- [Using the `-validate` Parameter to Verify Prerequisites](#)
Before you run an operation from the command line, you can use the `-validate` option in silent mode to verify that all RCU prerequisites have been met.
- [Using a Wallet to Store Passwords](#)
If you are running RCU from the command line, password credentials (for example, schema passwords) can be stored in an Oracle Wallet file. This wallet file can be provided to RCU from the command line.

Using the `-validate` Parameter to Verify Prerequisites

Before you run an operation from the command line, you can use the `-validate` option in silent mode to verify that all RCU prerequisites have been met.

When you run RCU with the `-validate` option, RCU performs validation checks to validate all prerequisites for the specified operation, and then exits. For example, this command validates the values for any command-line parameters and verifies that the minimum requirements are met in the database prior to schema creation.

You can add the `-validate` option to any valid operation to run the validation checks, as shown in the following example:

```
./rcu -silent -createRepository -useSamePasswordForAllSchemaUsers true -  
databaseType ORACLE  
-connectString examplehost.exampledomain.com:1521:exampleSID -dbUser sys -  
dbRole sysdba  
-selectDependentsForComponents true -schemaPrefix DEV -component MDS -component  
OPSS  
-component STB -component BIPLATFORM -validate
```

If a prerequisite check fails, RCU returns a non-zero exit code and prints the issue to the console. You can refer to the command output and log file to manually fix the issue, and then rerun the `-validate` command to ensure that all checks are successful.

If all prerequisite checks pass, the exit status is zero (0).

If a validation check fails, the following table provides more information about the exit codes returned by RCU.

Validation Check	Description	Exit Code Number
Database Connection Check	<p>Validates the provided connection credentials for the database in which you create or drop schemas.</p> <p>This check ensures that a database connection can be established by using the values provided for the database connect string, user name, and password.</p> <p>If the <code>-encryptTablespace true</code> option is provided, RCU verifies that TDE (Transparent Data Encryption) is enabled in the database.</p>	<ul style="list-style-type: none"> • Database connection failed: 100 • Unsupported database type: 101 • Empty field: 102 • Invalid port: 103
Global Prerequisites Check	<p>Verifies the global prerequisites listed in the global <code>ComponentInfo.xml</code> file (in <code>ORACLE_HOME/oracle_common/rcu/config</code>).</p> <p>For example, this check verifies the database version and the values for the database initialization parameters.</p>	<ul style="list-style-type: none"> • Global prerequisite failed: 200 • Metadata initialization failed: 201 • TDE not enabled: 202
Schema Prefix Check	<p>Validates the <code>-schemaPrefix</code> parameter.</p> <ul style="list-style-type: none"> • For the create operation, RCU ensures that the prefix follows the expected pattern and does not exceed the allowed number of characters. • For the drop operation, RCU ensures that the schema prefix already exists. 	<ul style="list-style-type: none"> • Schema prefix validation failed: 300
Component Name Check	<p>Validates the list of component schema names provided.</p> <p>For the create operation, RCU ensures that the components do not exist for the given prefix and any dependent components are selected.</p>	<ul style="list-style-type: none"> • Component validation failed: 400 • Component dependency validation failed: 401
Schema Password Check	<p>Verifies that a given schema password follows the supported password pattern and does not exceed 30 characters.</p>	<ul style="list-style-type: none"> • Schema password validation failed: 700
Component Prerequisites Check	<p>Validates the component-specific prerequisites of each component.</p> <p>The prerequisites are listed in the component repository configuration file (<code>component.xml</code>) in <code>ORACLE_HOME/product_directory/common/sql/component</code> for each component.</p>	<ul style="list-style-type: none"> • Component prerequisite failed: 500

Validation Check	Description	Exit Code Number
Tablespace Validation Check	Validates the tablespace names and verifies that the tablespace prerequisites are met.	<ul style="list-style-type: none"> • Tablespace validation failed: 600 • Mapping tablespace to component validation failed: 601
Custom Variables Check	Validates component custom variables and their values.	<ul style="list-style-type: none"> • Custom variable validation failed: 800
Script Location Validation Check	Validates the value provided for the <code>-scriptLocation</code> argument and checks whether the value is null or empty.	<ul style="list-style-type: none"> • Value for <code>-scriptLocation</code> argument is invalid or not provided: 900

Using a Wallet to Store Passwords

If you are running RCU from the command line, password credentials (for example, schema passwords) can be stored in an Oracle Wallet file. This wallet file can be provided to RCU from the command line.

When you run RCU from the command line, RCU prompts you to enter values for any required passwords by default. To avoid entering passwords on the command line, you can create a wallet file to securely store any necessary password credentials. You can then pass the wallet to RCU by providing the wallet location with the `walletDir` parameter.

Creating a Wallet Using RCU

A wallet for providing RCU passwords can be generated by RCU as part of the response file creation:

- To create a wallet using the RCU graphical interface, see [Creating a RCU Response File from the Graphical Interface](#).
- To create a wallet using the RCU command-line interface, see [Creating a RCU Response File from the Command Line](#).

Once the wallet is created, this file can be reused to store and provide passwords for multiple sessions.

RCU only supports auto-login wallets (no password is needed to open a wallet). Password-protected wallets are not supported.

When RCU generates the wallet, RCU uses the prefixed schema name as the key or alias for storing schema passwords, which allows you to reuse the same wallet for multiple sessions. For the database user password, the alias is the database user name (for example, SYS). If the same password is used for all the schemas, then RCU uses the alias `prefix_COMMON_SCHEMA_PASSWORD` to save the password in the wallet.

Example 3-1 Using the `-walletDir` Parameter

After the wallet is created and contains the necessary passwords, use the `-walletDir` parameter to pass this file to RCU so that you do not have to directly enter your passwords on the command line. Alternatively, the wallet location can be provided to RCU through a response file.

For example:


```
./rcu -silent -createRepository -connectString  
examplehost.exampledomain.com:1521:example_sid -dbUser sys -dbRole sysdba  
-schemaPrefix DEV -component MDS -component STB -walletDir /home/exampleuser/  
wallet
```

When you pass the wallet to RCU, RCU expects all the necessary passwords required to perform an operation to be in the wallet, which include your:

- Database user password
- Schema passwords
- Custom Variable passwords (for Oracle Data Integrator)

In this example, the wallet should contain passwords for the database user (sys), the Metadata Services (MDS) schema, and the Service Table (STB) schema. If a wallet does not contain all the necessary passwords, you are prompted to enter the missing passwords on the command line.

Using the `-interactive` Command

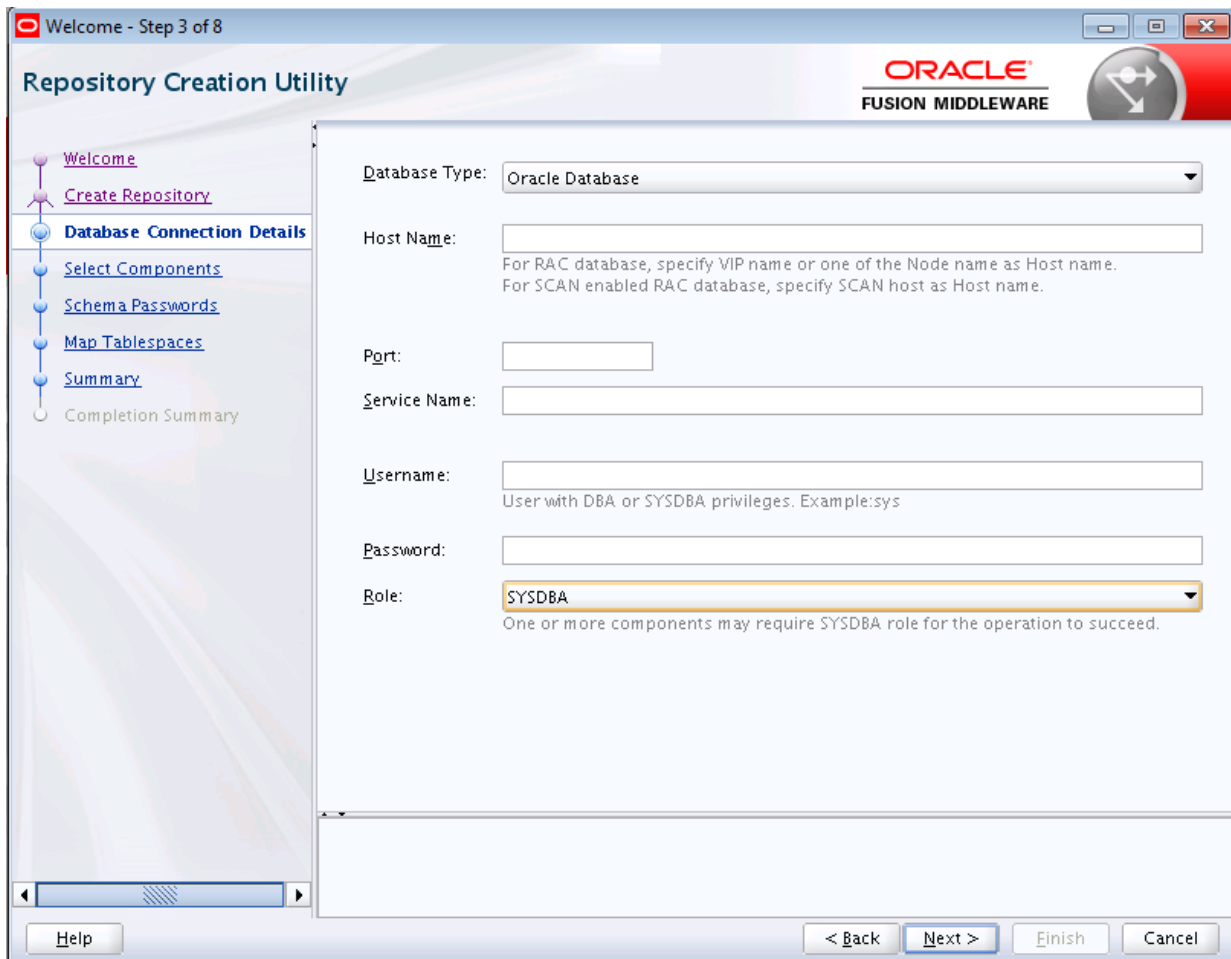
Use the `-interactive` command to run the RCU graphical interface. This is the default value if neither `-silent` nor `-interactive` is specified.

You can specify information from the command line that would be populated in the graphical interface to expedite your RCU operation. For example, if you run RCU with the following command:

```
./rcu
```

The [Database Connection Details](#) page contains blank fields, as shown below:

Figure 3-1 Database connection Details — blank screen



But if you run RCU with a few parameters from the command line:

```
./rcu -interactive -createRepository -connectString  
examplehost.exampledomain.com:1521:example_sid -dbUser SYS -component MDS
```

The [Database Connection Details](#) page contains the information supplied from the command line:

Figure 3-2 Database Connection Details

Welcome - Step 3 of 8

Repository Creation Utility

ORACLE
FUSION MIDDLEWARE

Welcome

Create Repository

Database Connection Details

Select Components

Schema Passwords

Map Tablespaces

Summary

Completion Summary

Database Type: Oracle Database

Host Name: examplehost.exampledomain.com
For RAC database, specify VIP name or one of the Node name as Host name.
For SCAN enabled RAC database, specify SCAN host as Host name.

Port: 1521

Service Name: example_sid

Username: SYS
User with DBA or SYSDBA privileges. Example:sys

Password:

Role: SYSDBA
One or more components may require SYSDBA role for the operation to succeed.

Help < Back Next > Finish Cancel

When you run the RCU graphical interface, you can provide the location and name of a file (called a response file) that contains certain input values (for example, your database connection information). When you use the graphical interface, you are prompted for these values. The values you enter in a response file can be used to populate certain RCU screens. For more information about using response file, see [Using Response Files](#).

Using Response Files

Before you run RCU, you can provide information that is required to perform a specific operation in a response file. For example, you can provide your database connection details, a list of components to be added or dropped, and the schema prefix to RCU through a response file.

You can use a response file to automate a RCU session. When you run RCU by using a response file, RCU reads the information provided in the response file to complete the operation.

For more information, see the following topics:

- [About Response Files](#)
A response file is a properties file in which all RCU command-line parameters can be provided as input to RCU to perform an operation.
- [Creating a RCU Response File from the Graphical Interface](#)
Use the **Save Response File** option on the Summary screen to save the information you provide in the RCU screens to a response file.
- [Creating a RCU Response File from the Command Line](#)
Use the `-generateResponseFile` command to generate a response file.
- [Performing RCU Operations by Using a Response File](#)
Use the `-responseFile` command to perform an operation with a response file. You use a response file to provide input to RCU to automate an operation from the command line or from the graphical interface.

About Response Files

A response file is a properties file in which all RCU command-line parameters can be provided as input to RCU to perform an operation.

Rather than directly entering the [RCU Command Line Parameters](#) on the command line, you can provide values for these parameters to RCU through a text file. You can generate this file from the RCU graphical interface by using the values you provide in the RCU screens or from the command line. After you create a response file, the parameters can be modified in a text editor or be used exactly as-is to perform an operation at a later time.

If you choose to run RCU from the command line with a response file, the values you specify in a response file are the values for the RCU parameters that you would normally provide on the command line during a typical silent RCU session. If you choose to run the RCU graphical interface with a response file, the values you specify in a response file are used to prepopulate the fields on certain screens.

Creating a RCU Response File from the Graphical Interface

Use the **Save Response File** option on the Summary screen to save the information you provide in the RCU screens to a response file.

After you click the **Save Response File** option, you are prompted to specify a location to create this file. By default, RCU saves the response file in the `$HOME/.rcu` directory.

Because the response file needs to contain password information, RCU provides the **Save Passwords in Wallet** option to securely store any passwords that you provide on the RCU screens in a wallet. You can use the wallet later in silent situations to provide the necessary passwords to RCU so that you do not have to directly enter your passwords on the command line. When you run RCU from the command line, the wallet location can be passed to RCU on the command line or through a response file. If a wallet directory is not passed to RCU, you are prompted for the required passwords.

Creating a RCU Response File from the Command Line

Use the `-generateResponseFile` command to generate a response file.

The full syntax for the RCU command-line interface to generate a RCU response file is shown below:

```
rcu -silent -generateResponseFile
  [-responseFileDir absolute_path_of_response_file_directory]
  [-schemaPrefix schema_prefix]
  [-componentList list_of_component_IDs]
  [-createWallet]
  [-useSamePasswordForAllSchemaUsers [true|false]]
  [-selectDependentsForComponents [true|false]]
  [-dbUser database_username]
```

If the `-createWallet` parameter is provided, then values for `-dbUser` and `-schemaPrefix` are required. Otherwise, the `-dbUser` and `-schemaPrefix` parameters are optional. RCU uses the prefixed schema name and the database user name as the key or alias to store schema passwords and the database user password in the wallet, respectively.

If you provide the `-createWallet` and the `-useSamePasswordForAllSchemaUsers true` parameters on the command line, then all schema passwords are set to the same value in the wallet.

If the `-createWallet` parameter is provided, RCU will prompt you to enter a value for each password that is stored in the wallet.

Below is a sample command to generate a response file on a UNIX operating system:

```
./rcu -silent -generateResponseFile -componentList OPSS,IAU -schemaPrefix DEV
-createWallet
-responseFileDir /home/exampleuser/rcu -useSamePasswordForAllSchemaUsers true
-selectDependentsForComponents true -dbUser sys
```

In this example, a response file and wallet is created in the `/home/exampleuser/rcu` directory, and the information provided on the command line (such as the list of components) is added to the response file.

Performing RCU Operations by Using a Response File

Use the `-responseFile` command to perform an operation with a response file. You use a response file to provide input to RCU to automate an operation from the command line or from the graphical interface.

If you choose to run the graphical interface with a response file, RCU prepopulates certain RCU screens (such as the [Database Connection Details](#) screen) based on the input values that you provide in a response file. If you choose to run RCU from the command line, you can use a response file to provide values for all valid RCU parameters rather than directly entering these parameters on the command line.

The full syntax for the RCU command-line interface to perform a RCU operation with a response file is shown below:

```
rcu [-silent|-interactive] -responseFile absolute_path_of_response_file
```

Following is a sample command to automate a silent RCU session by using a response file on a UNIX operating system:

```
./rcu -silent -responseFile /home/exampleuser/rcu/rcuResponseFile.properties
```

 **Note:**

- If you provide a value for a parameter on the command line and in the response file, RCU uses the value that is provided on the command line.
- Before you run RCU, ensure that you have provided the required information in the response file. If some of the mandatory properties in the response file are missing values when you run RCU, RCU will notify you and exit.
- When you run RCU, the `dbRole` parameter is not normally required to connect to certain databases (for example, SQL Server). However, when a RCU response file is generated, the `dbRole` parameter is marked as required in the response file by default. As a result, if you use a response file, the `dbRole` field must not be left blank, even if the database role is not required to connect to a database like SQL Server.

Creating a Repository from the Command Line

If you are a user with full SYS or SYSDBA privileges, use the `-createRepository` operation to create a repository.

The full syntax for the RCU command-line interface to create a repository is shown below:

```
rcu [-silent | -interactive] -createRepository
  [-compInfoXMLLocation ComponentInfo.xml_file_location]
  [-storageXMLLocation Storage.xml_file_location]
  [-databaseType [ORACLE|EBR]]
  -connectString database_connect_string
  [-edition edition_name]
  -dbUser database_username
  [-dbRole database_user_role]
  [-unicodeSupport [Yes|No]]
  [-skipCleanupOnFailure [true|false]]
  [-skipTablespaceDropOnFailure [Yes|No]]
  [-useSamePasswordForAllSchemaUsers [true|false]]
  [-selectDependentsForComponents [true|false]]
  [-honorOMF [true|false]]
  [-encryptTablespace [true|false]]
  [-variables variablename=value]
  [-schemaPrefix schema_prefix]
  -component component_ID
  [-tablespace component_tablespace_name]
  [-tempTablespace component_temp_tablespace_name]
  [-walletDir absolute_path_of_wallet_directory]
  [-validate]
```

When you specify the `-component`, you must use the correct component IDs, which are listed in [About Repository Creation Utility Schemas, IDs, and Tablespaces](#).

Before you create any schemas, you must be aware of and specify all dependencies for the component that you are loading. For example, the `SOAINFRA` schema depends on the `MDS` and `ORASDPM` schemas; if you try to load the `SOAINFRA` schema without

specifying both the MDS and ORASDPM schemas, or if the MDS and ORASDPM schemas do not already exist in the database, RCU stops before any loading takes place.

Following is a sample command to create the OPSS schema on a UNIX operating system:

```
./rcu -silent -createRepository -databaseType ORACLE -connectString
examplehost.exampledomain.com:1521:exampleSID -dbUser sys
-dbRole sysdba -schemaPrefix TEST -component OPSS -component MDS
```

Generating a System Load Script From the Command Line

If you are a user without SYS or SYSDBA privileges, use the `-generateScript` operation to generate a script. A user with DBA privileges can run the script later to complete the system load phase.

The full syntax for the RCU command-line interface to generate a system load script is shown below:

```
rcu [-silent | -interactive] -generateScript
  [-compInfoXMLLocation ComponentInfo.xml_file_location]
  [-storageXMLLocation Storage.xml_file_location]
  [-databaseType [ORACLE|EBR]]
  -connectString database_connect_string
  [-edition edition_name]
  -dbUser database_username
  [-dbRole database_user_role]
  [-skipCleanupOnFailure [Yes|No]]
  [-skipTablespaceDropOnFailure [Yes|No]]
  [-scriptLocation script_location]
  [-selectDependentsForComponents [true|false]]
  [-honorOMF [true|false]]
  [-encryptTablespace [true|false]]
  [-variables variablename=value]
  [-schemaPrefix schema_prefix]
  -component component_ID
  [-tablespace component_tablespace_name]
  [-tempTablespace component_temp_tablespace_name]
  [-walletDir absolute_path_of_wallet_directory]
  [-validate]
```

When specifying the `-component`, you must use the correct component IDs, which are listed in [About Repository Creation Utility Schemas, IDs, and Tablespaces](#).

Before you create a script, you must be aware of and specify the all component dependencies. For example, the SOAINFRA schema depends on the MDS and ORASDPM schemas; if you specify the SOAINFRA schema without specifying both the MDS and ORASDPM schemas, or if the MDS and ORASDPM schemas do not already exist in the database, RCU stops before completing the operation.

Below is a sample command to create a system load script on a UNIX operating system:

```
./rcu -silent -generateScript -databaseType ORACLE -connectString
examplehost.exampledomain.com:1521:exampleSID -dbUser sys
-dbRole sysdba -scriptLocation /tmp/RCUdate_timestamp_random_number/logs/ -
schemaPrefix TEST
-component OPSS -component MDS
```

Loading Data Into the Repository From the Command Line

After the system load scripts are run by a user with DBA privileges, any user can then use the `-dataLoad` operation to load data into a repository.

The full syntax for the RCU command-line interface to load data into a repository is shown below:

```
rcu [-silent | -interactive] -dataLoad
  [-compInfoXMLLocation ComponentInfo.xml_file_location]
  [-storageXMLLocation Storage.xml_file_location]
  [-databaseType [ORACLE|EBR]]
  -connectString database_connect_string
  [-edition edition_name]
  -dbUser database_username
  [-dbRole database_user_role]
  [-skipCleanupOnFailure [Yes|No]]
  [-skipTablespaceDropOnFailure [Yes|No]]
  [-useSamePasswordForAllSchemaUsers [true|false]]
  [-selectDependentsForComponents [true|false]]
  [-variables variablename=value]
  [-schemaPrefix schema_prefix]
  -component component_ID
  [-walletDir absolute_path_of_wallet_directory]
  [-validate]
```

Below is a sample command to perform a data load on a UNIX operating system:

```
./rcu -silent -dataLoad -databaseType ORACLE -connectString
examplehost.exampledomain.com:1521:exampleSID -dbUser sys -dbRole
sysdba -schemaPrefix TEST -component OPSS -component MDS
```

Dropping a Repository from the Command Line

Use the `-dropRepository` operation to drop a repository.

The full syntax for the RCU command-line interface to drop a repository is shown below:

```
rcu [-silent | -interactive] -dropRepository
  [-compInfoXMLLocation ComponentInfo.xml_file_location]
  [-storageXMLLocation Storage.xml_file_location]
  [-databaseType [ORACLE|EBR]]
  -connectString database_connect_string
  [-edition edition_name]
  -dbUser database_username
  [-dbRole database_user_role]
  [-unicodeSupport [Yes|No]]
  [-selectDependentsForComponents [true|false]]
  [-skipTablespaceDrop [Yes|No]]
  [-variables variablename=value]
  [-schemaPrefix schema_prefix]
  -component component_ID
  [-walletDir absolute_path_of_wallet_directory]
  [-validate]
```

You must also be aware of schema dependencies when dropping schemas (see [About Repository Creation Utility Schemas, IDs, and Tablespaces](#)). For example, several

schemas require the MDS schema to be present; if you choose to drop the MDS schema, then all the schemas that require the MDS schema stops working.

Below is a sample command to drop the OPSS schema on a UNIX operating system:

```
./rcu -silent -dropRepository -databaseType ORACLE -connectString
examplehost.exampledomain.com:1521:exampleSID -dbUser sys -dbRole
sysdba -schemaPrefix TEST -component OPSS
```

Generating a List of Components in an Oracle Home from the Command Line

Use the `-listComponents` command to generate a list of valid components that are available in a given Oracle home. This command displays the name, component ID, and the supported database types for each component that is available in the Oracle home in which RCU was started.

The full syntax for the RCU command-line interface to generate a list of components is shown below:

```
rcu -silent -listComponents
```

The Component ID and database type displayed for each component in the command output are the values that you must specify with the `-component` and `-databaseType` parameters, respectively, when you create schemas from the command line.

Generating a List of Schemas in a Database from the Command Line

Use the `-listSchemas` command to generate a list of valid schemas in a given database.

The full syntax for the RCU command-line interface to list the schemas in a database is shown below:

```
rcu -silent -listSchemas
[-databaseType [ORACLE|EBR]]
-connectString database_connect_string
[-edition edition_name]
-dbUser database_username
[-dbRole database_user_role]
[-schemaPrefixes comma_separated_list_of_prefixes]
[-walletDir absolute_path_of_wallet_directory]
```

Below is a sample command to list the schemas in an Oracle database on a UNIX operating system:

```
./rcu -silent -listSchemas -databaseType ORACLE -connectString
examplehost.exampledomain.com:1521:exampleSID
-dbUser sys -dbRole sysdba
```

You can also specify a comma-separated list of prefixes using the `-schemaPrefixes` argument. If provided, RCU only displays the schemas with those prefixes.

For example:

```
./rcu -silent -listSchemas -databaseType ORACLE -connectString
examplehost.exampledomain.com:1521:exampleSID -dbUser sys
-dbRole sysdba -schemaPrefixes DEV,DEV1
```

RCU Environment Variables

The RCU environment variables are the variables picked up by RCU from the environment. If an environment variable is not set, then RCU uses the default value.

Table 3-4 RCU Environment Variables

Variable	Default Value	Description
RCU_LOG_LOCATION	/tmp (UNIX operating systems) C:\Users\user_name\AppData\Local\Temp (Windows operating systems)	Location of the RCU log file.
RCU_TIMESTAMP_LOG_DIR	true	Determines whether or not a directory with the format <i>logdir.yyyy-dd_hh-mm</i> is created for the RCU log file. Set this variable to true or false.
RCU_LOG_NAME	rcu.log	Name of the RCU log file.
RCU_LOG_LEVEL	ERROR	Determines the RCU log level. Set this variable to one of SEVERE, ERROR, NOTIFICATION, or TRACE.
RCU_JAVA_OPTIONS	N/A	Set this variable to any Java Virtual Machine (JVM) argument. For example: setenv RCU_JAVA_OPTIONS -Djava.io.tmpdir=/scratch/mytmpDir

A

Repository Creation Utility Screens

Use this information to learn more about the various features and elements of the Repository Creation Utility screens. Each screen will prompt you for information required to create or drop your schemas.

- [Welcome](#)
The Welcome screen is the first screen that appears when you start RCU. Note the navigation pane on the left that summarizes the tasks that RCU helps you complete.
- [Create Repository](#)
Use the Create Repository screen to select the action that you want to perform.
- [Database Connection Details](#)
Use the Database Connection Details screen to specify the connection credentials to the database in which you will be create or drop your schemas.
- [Select Components \(for Create Operation\)](#)
Use the Select Components screen to select the components for which you want to create schemas, and specify a prefix to group them together.
- [Select Components \(for Drop Operation\)](#)
Use the Select Components screen to select the prefix and the schemas that you want to drop.
- [Schema Passwords](#)
Use the Schema Passwords screen to specify the passwords for your schema owners; there are three ways to specify schema passwords on this screen.
- [Custom Variables](#)
If you are installing a product that requires custom variables, you will see the Custom Variables screen. Use this screen to specify additional product configuration information.
- [Map Tablespaces](#)
Use the Map Tablespaces screen to configure the desired tablespace mapping for the schemas that you want to create.
- [Summary \(for Create Operation\)](#)
On the Summary (for Create operation) screen, verify your information, then click **Create** to begin schema creation.
- [Summary \(for Drop Operation\)](#)
On the Summary (for Drop operation) screen, verify your information, then click **Drop** to drop the schemas.
- [Completion Summary \(for Create Operation\)](#)
The Completion Summary (for Create Operation) screen contains information about the log files that were created from this RCU operation. You can click on the name of a particular log file to view the contents of that file.

- [Completion Summary \(for Drop Operation\)](#)
The Completion Summary (for Drop operation) screen contains information about the log files that are created from this RCU operation. Note the log file names for each component that are visible in the *Logfile* column.

Welcome

The Welcome screen is the first screen that appears when you start RCU. Note the navigation pane on the left that summarizes the tasks that RCU helps you complete.

Each item in the navigation pane represents a specific screen that prompts you for information that is required to create or drop your schemas.

Create Repository

Use the Create Repository screen to select the action that you want to perform.

- [Create Repository](#)
Select **Create Repository** to create component schemas in the database.
- [Drop Repository](#)
Select **Drop Repository** to remove component schemas from the database.

Create Repository

Select **Create Repository** to create component schemas in the database.

The following table describes the options available:

Option	Description
System Load and Product Load	Select this option to perform both System Load and Product Load operations. Users must have DBA or SYSDBA permissions to select this option. This option can be performed on any certified database.
Prepare Scripts for System Load	Select this option to perform actions that require DBA or SYSDBA permissions: <ul style="list-style-type: none"> • Create tablespaces and schemas. • Create the <code>schema_version_registry</code> (if not already present). • Create entries in <code>schema_version_registry</code> for each selected component, set the proper access permissions, and set the status of the component to <i>LOADED</i> in the <code>schema_version_registry</code> table. Any SYSDBA user or non-DBA user with the privileges described in Granting Permissions to Users for Querying System Load Objects can select this option. However, DBA privileges are required to execute the SQL script to complete the system load phase. This option is only supported on Oracle and Oracle EBR databases.

Option	Description
Perform Product Load	<p>Select this option to load and create procedures, functions, tables, indexes and other objects within schemas and run any action that does not require DBA access.</p> <p>Any non-DBA user or the REGISTRYOWNER user can select this option.</p> <p>Before you select this option, the user must be granted the following permissions to perform data load:</p> <pre>grant REGISTRYACCESS to user; grant STBROLE to user;</pre>

Drop Repository

Select **Drop Repository** to remove component schemas from the database.

Database Connection Details

Use the Database Connection Details screen to specify the connection credentials to the database in which you will be create or drop your schemas.

Click **Next** when you have finished typing the connection credentials for your database. The following screen appears, indicating the progress of the installer establishing the connection with the specified database:

If an error occurs while the connection is being established, the error messages appear in the Messages field on the Database Connection Details screen.

Specific database requirements for the various schemas can be found in the *Oracle Fusion Middleware System Requirements and Specifications* document.

For certified database versions, see the *System Requirements and Supported Platforms for Oracle Fusion Middleware 12c* document, which is available on the Oracle Fusion Middleware Supported System Configurations page.

Select one of the following depending on your database:

- [Connection Credentials for Oracle Databases and Oracle Databases with Edition-Based Redefinition](#)
To connect to an Oracle database or Oracle database with edition-based redefinition, you need to specify connection information about the database, such as Host Name, Port, Service Name, Username, Password, and Role.
- [Connection Credentials for Microsoft SQL Server Databases](#)
To connect to a Microsoft SQL Server database, you need to specify connection information about the database, such as Unicode Support, Server Name, Port, Database Name, Username, and Password.
- [Connection Credentials for IBM DB2 Databases](#)
To connect to an IBM DB2 database, you need to specify connection information about the database, such as Server Name, Port, Database Name, Username, and Password.

Connection Credentials for Oracle Databases and Oracle Databases with Edition-Based Redefinition

To connect to an Oracle database or Oracle database with edition-based redefinition, you need to specify connection information about the database, such as Host Name, Port, Service Name, Username, Password, and Role.

Use the **Database Connection Details** screen to specify the connection credentials to the database in which you want to create or drop schemas. **Database Connection Details** screen provides two types of Connection String formats as follows:

- Connection Parameters
- Connection String

For various database types and both **Connection String Format**, specify the following connection credentials:

Element	Description	Database Types
Host Name	<p>Enter the name of the server where your database is running. Use the following format:</p> <p><code>examplehost.exampledomain.com</code></p> <p>For Oracle RAC databases, specify the VIP name or one of the node names in this field.</p>	<ul style="list-style-type: none"> • Oracle Database • Oracle Database enabled for edition-based redefinition • MySQL Database
Port	<p>Enter the port number for your database. The default port number for Oracle databases is 1521.</p>	<ul style="list-style-type: none"> • Oracle Database • Oracle Database enabled for edition-based redefinition • Microsoft SQL Server • IBM DB2 • MySQL Database
Service Name	<p>Specify the service name for the database. Typically, the service name is the same as the global database name. If you are unsure what the service name for your database is, you can obtain it from the <code>SERVICE_NAMES</code> parameter in the initialization parameter file of the database. If the initialization parameter file does not contain the <code>SERVICE_NAMES</code> parameter, then the service name is the same as the global database name, which is specified in the <code>DB_NAME</code> and <code>DB_DOMAIN</code> parameters.</p> <p>For Oracle RAC databases, specify the service name of one of the nodes in this field. For example:</p> <p><code>examplehost.exampledomain.com</code></p>	<ul style="list-style-type: none"> • Oracle Database • Oracle Database enabled for edition-based redefinition
Username	<p>Enter the user name for your database.</p>	<ul style="list-style-type: none"> • Oracle Database • Oracle Database enabled for edition-based redefinition • Microsoft SQL Server • IBM DB2 • MySQL Database

Element	Description	Database Types
Password	Enter the password for your database user.	<ul style="list-style-type: none"> Oracle Database Oracle Database enabled for edition-based redefinition Microsoft SQL Server IBM DB2 MySQL Database
Role	Select the database user's role from the drop-down list: <ul style="list-style-type: none"> Normal SYSDBA 	<ul style="list-style-type: none"> Oracle Database Oracle Database enabled for edition-based redefinition
Unicode Support	Select the Unicode Support from the drop-down list: <ul style="list-style-type: none"> Yes No 	Microsoft SQL Server
Server Name	For Microsoft SQL server, type one of the following: <ul style="list-style-type: none"> Hostname IP address Complete servername in host\server format. For IBM DB2, type the IP address of the server.	<ul style="list-style-type: none"> Microsoft SQL Server IBM DB2
Database Name	Type the database name. It is case-sensitive.	<ul style="list-style-type: none"> Microsoft SQL Server IBM DB2 MySQL Database
Connect String	For Oracle Databases and Oracle Database enabled for edition-based redefinition types, type Connect String in one of the following formats: <ul style="list-style-type: none"> <host>:<port>/<service> <host>:<port>:<SID> DESCRIPTION=(ADDRESS=(host=host_name)(protocol=protocol_name)(port=port_name))(CONNECT_DATA=(SERVICE_NAME=service_name)) For Microsoft SQL Server, IBM DB2, and MySQL Database, type Connect String in <Server name>/<host>:<port>:<databaseName> format.	<ul style="list-style-type: none"> Oracle Database Oracle Database enabled for edition-based redefinition Microsoft SQL Server IBM DB2 MySQL Database

Click **Next** once you have entered the connection credentials for your database. A screen indicating the progress of the installer establishing the connection with the specified database appears. If an error occurs while the connection is being established, the error message appears in the Messages field on the Database Connection Details screen.

Specific database requirements for the various schemas can be found in the *Oracle Fusion Middleware System Requirements and Specifications* document.

View the *System Requirements and Supported Platforms for Oracle Fusion Middleware 12c* document for other certified databases that can be used with RCU. This document is available on the *Oracle Fusion Middleware Supported System Configurations* page.

Connection Credentials for Microsoft SQL Server Databases

To connect to a Microsoft SQL Server database, you need to specify connection information about the database, such as Unicode Support, Server Name, Port, Database Name, Username, and Password.

Field	Description
Unicode Support	Select Yes or No from the drop-down list.
Server Name	Enter the host name, IP address, or complete server name in host\server format of the server where your database is running.
Port	Enter the port number for your database.
Database Name	Specify the name of your database.
Username	Specify the name of a user with administrator privileges.
Password	Enter the password for your database user.

Connection Credentials for IBM DB2 Databases

To connect to an IBM DB2 database, you need to specify connection information about the database, such as Server Name, Port, Database Name, Username, and Password.

Field	Description
Server Name	Enter the host name, IP address, or complete server name in host/server format of the server where your database is running.
Port	Enter the port number for your database.
Database Name	Specify the name of your database.
Username	Specify the name of a user with DB Owner privileges. The default user name for IBM DB2 databases is db2admin.
Password	Enter the password for your database user.

Select Components (for Create Operation)

Use the Select Components screen to select the components for which you want to create schemas, and specify a prefix to group them together.

When you selected **Create Repository** on the [Create Repository](#) screen, the following **Select Component** screens appears:

The following table describes the elements on this screen.

Element	Description
Select existing prefix	Use this option to create schemas by using an existing prefix, which you can select from the drop-down list. If you select this option, you can only create schemas that do not already have this prefix in the database. For example, if DEV_MDS already exists, you are unable to select the MDS schema on the page to create it again).
Create new prefix	Use this option to create a new custom prefix that can be used to group your schema together. The prefix name must be a minimum of one character in length and cannot exceed 12 alphanumeric characters (0-9, a-z, or A-Z) in length (not including the underscore character). Prefixes should not start with a number. No whitespace or special characters are allowed. You can edit the schema name or prefix by clicking the item that you want to change in the <i>Schema Owner</i> column and editing the table cell directly. For more information about custom prefixes, see About Custom Prefixes .
Component	Use the <i>Component</i> column in the table to select the component schema that you want to create. When you select a component, any other components that may be required by the component you select are also selected. For example, if you select Oracle Platform and Security Services , then the Audit Services schema is also automatically selected, if it has not already been selected. The Audit Services schema is required by the Oracle Platform and Security Services schema. If a component has a plus sign (+) next to its name, then there are sub components available. Click on the plus sign (+) to expand the category to view all sub components. If you want to select a component with all its subcomponents, click on the top-most box with the plus sign (+). If you are creating component schemas on an IBM DB2 database, see important information in Additional Requirements for IBM DB2 Databases .
Schema Owner	This column shows the name of the schema owner for each component (custom prefix and schema name). You can edit the schema owner value by clicking the item that you want to change in this column and editing the table cell directly.

 **Note:**

You must remember the Schema Owner name for the components that you are installing; you need this information during the configuration phase of your Oracle Fusion Middleware product installation. Oracle recommends that you write this value down.

Click **Next** when you are finished specifying your prefix and schema names, and when you are finished selecting components. The following screen appears, indicating the progress of component prerequisite checking before the schemas are created.

If an error occurs during the prerequisite checking, the error messages appear in the Messages field on the Select Components screen.


Click **OK** to dismiss this screen.

Select Components (for Drop Operation)

Use the Select Components screen to select the prefix and the schemas that you want to drop.

Below is the Select Components screen if you selected **Drop Repository** on the [Create Repository](#) screen.

The following table describes the elements on this screen:

Element	Description
Select schemas with prefix of	Use the drop-down list in this field to select an existing custom prefix.
Component	Select the component schemas you want to remove.
	<div style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> Note:</p> <p>Unlike create operations, schema dependencies are not handled automatically by RCU for drop operations. When you remove a schema, you must also remove the schemas which depend on the schema you are removing. For example, the Audit Services schema is required by the Oracle Platform Security Services schema; if you remove the Audit Services schema, the Oracle Platform Security Services schema will no longer works.</p> </div>
Schema Owner	This column shows the schema owner (custom prefix and schema name). For drop operations, this column cannot be edited.

Click **Next** when you are finished. The following screen appears:

If an error occurs during the prerequisite checking, the error message(s) appear in the Messages field on the Select Components screen.

Click **OK** to dismiss this screen.

Schema Passwords

Use the Schema Passwords screen to specify the passwords for your schema owners; there are three ways to specify schema passwords on this screen.

The following table describes the three ways that you can specify schema passwords:

Option	Description
Use same password for all schemas	Select this option if you want to use a single password for all schemas and their auxiliary schemas. In the Password field, enter your password. Enter your password again in the Confirm Password field.

Option	Description
Use main schema passwords for auxiliary schemas	Select this option if you want to specify different passwords for the main schemas, but still have the same password used for their respective auxiliary schemas. If you select this option, only the main schemas are visible in the table. For each schema, you must enter each schema's password in the Schema Password column in the table, and enter the same password in the Confirm Password column.
Specify different passwords for all schemas	Select this option if you want to specify unique passwords for the main schemas and auxiliary schemas. If you select this option, all main schemas and auxiliary schemas are visible in the table. For each schema and auxiliary schema, you must enter the password in the Schema Password column in the table, and enter the same password in the Confirm Password column.

 **Note:**

You must remember the passwords you enter on this screen; you need this information during the configuration phase of your product installation. Oracle recommends that you write down these values.

Custom Variables

If you are installing a product that requires custom variables, you will see the Custom Variables screen. Use this screen to specify additional product configuration information.

The Custom Variables screen appears only if you selected one or more of the following components on the Select Components screen. You can select a component on the Select Components screen only if the Oracle home in which RCU was started contains that component.

- If you selected **Oracle Data Integrator**, the custom variables for Oracle Data Integrator are visible.
- If you selected **SOA Suite**, the custom variables for Oracle SOA Suite are visible.
- If you selected **WebCenter Portal - Analytics**, the custom variable for Oracle WebCenter Portal Analytics is visible.
- If you selected **Oracle GoldenGate - Repository**, the custom variables for Oracle GoldenGate Studio are visible.
- If you selected **Oracle Data Integrator**, **SOA Suite**, and **WebCenter Portal - Analytics**, the custom variables for Oracle Data Integrator, Oracle SOA Suite, and Oracle WebCenter Portal Analytics are visible on this screen.

For more information, see the following topics:

- [Custom Variables for Oracle Data Integrator](#)
For Oracle Data Integrator, use the Custom Variables screen to specify a password for the supervisor user, the type of Work Repository to create, a password and unique name for the Work Repository, and the length of the key to use for the encryption algorithm.

- [Custom Variables for Oracle SOA Suite](#)
For Oracle SOA Suite, use the Custom Variables screen to specify the database profile that you want to use and whether or not you want to enable Healthcare Integration.
- [Custom Variable for Oracle WebCenter Portal Analytics](#)
For Oracle WebCenter Portal Analytics, use the Custom Variables screen to specify whether or not you want to install Analytics with database partitioning enabled.
- [Custom Variables for Oracle GoldenGate Studio](#)
For Oracle GoldenGate Studio, use the Custom Variables screen to specify a password for the supervisor user and the length of the key to use for the encryption algorithm.

Custom Variables for Oracle Data Integrator

For Oracle Data Integrator, use the Custom Variables screen to specify a password for the supervisor user, the type of Work Repository to create, a password and unique name for the Work Repository, and the length of the key to use for the encryption algorithm.

The custom variables for Oracle Data Integrator are described in the following table:

Variable	Description
Supervisor Password	Password of the supervisor user. You must enter a password for the Supervisor Password and Confirm Supervisor Password fields. These fields are required before you can continue. The password should be between 6 and 12 characters.
Work Repository Type	Specify how the Work Repository will be used: <ul style="list-style-type: none"> • Use Development (D) for creating a development repository. This type of repository allows management of design-time objects such as data models and projects (including interfaces, procedures, etc.) A development repository also includes the run-time objects (scenarios and sessions). This type of repository is suitable for development environments. • Use Execution (E) for creating an execution repository: This type of repository only includes run-time objects (scenarios, schedules and sessions). It allows launching and monitoring of data integration jobs in Operator Navigator. Such a repository cannot contain any design-time artifacts. Designer Navigator cannot be used with it. An execution repository is suitable for production environments. This field is optional. The default value is D.
Work Repository Name	A unique name for the Work Repository (for example: DEVWORKREP1). This field is optional.
Work Repository Password	Provide a password for the Work Repository. If you provide a password, you must confirm the password on the following line. The Work Repository Password and Confirm Work Repository Password fields are optional.
Encryption Algorithm	Select the encryption algorithm, either AES-128 or AES-256. This field is optional. If this field is left blank, the default value is AES-128.

Custom Variables for Oracle SOA Suite

For Oracle SOA Suite, use the Custom Variables screen to specify the database profile that you want to use and whether or not you want to enable Healthcare Integration.

The custom variables for Oracle SOA Suite are described in the following table:

Variable	Description
Database Profile	Specify the database profile that you want to use for Oracle SOA Suite. This determines the size of the SOA database. Use the drop-down list to specify the size. The available options are Small, Medium, or Large. The default value is Small.
Healthcare Integration	Specify whether you want to enable Healthcare Integration for Oracle SOA Suite. The default value is No.

Custom Variable for Oracle WebCenter Portal Analytics

For Oracle WebCenter Portal Analytics, use the Custom Variables screen to specify whether or not you want to install Analytics with database partitioning enabled.

Specify `Y` if you want to install Analytics with database partitioning enabled, or `N` if you do not want to enable database partitioning. The default value is `N`.

Custom Variables for Oracle GoldenGate Studio

For Oracle GoldenGate Studio, use the Custom Variables screen to specify a password for the supervisor user and the length of the key to use for the encryption algorithm.

The custom variables for Oracle GoldenGate Studio are described in the following table:

Variable	Description
Supervisor Password	Password of the supervisor user. You must enter a password in the Supervisor Password and Confirm Supervisor Password fields. These fields are required before you can continue. The password should be between 6 and 12 characters. Make a note of the password that you set on this screen; you need it later on when you provide your Oracle GoldenGate Studio connection information to connect to the repository after you start Oracle GoldenGate Studio.
Encryption Algorithm	Select the encryption algorithm, either AES-128 or AES-256. This field is optional. If this field is left blank, the default value is AES-128.

Map Tablespaces

Use the Map Tablespaces screen to configure the desired tablespace mapping for the schemas that you want to create.

This screen only appears if you selected the **Create Repository** option on the [Create Repository](#) screen.

Click **Next** when you are finished with your tablespace information. The following screen appears, asking you to confirm the creation of tablespaces for any new schemas.



Note:

- RCU only creates tablespaces for those components associated with RCU.
- The **Encrypt Tablespace** check box appears on this screen only if Transparent Data Encryption (TDE) is enabled in the database (Oracle or Oracle EBR) when RCU is started. For more information about encrypting Tablespaces, see [Encrypting Tablespaces](#).

Click **OK** to continue. The following screen appears, indicating the progress of the tablespace creation.

Click **Stop** to cancel tablespace creation. When the tablespaces are created, click **OK** to dismiss this window.

See the following topics for more about the features and options on the Map Tablespaces screen:

- [Default Tablespace Mappings](#)
In the *Default Tablespace* and *Temp Tablespace* columns, you can click on the tablespace cell to select from a list of available additional tablespace names.
- [Changing Default and Temporary Tablespaces](#)
To change the default tablespace for a component, select the tablespace name in the Default Tablespace column, then select the tablespace name that you want to use from the drop-down list.
- [Viewing and Changing Additional Tablespaces](#)
To view and change additional tablespaces associated with the selected components, click the **Additional Tablespaces** button. Only those components with additional tablespaces as defined in the configuration files appear on this screen.
- [Managing Tablespaces and Datafiles](#)
To manage your tablespaces and datafiles, click the **Manage Tablespaces** button.
- [Encrypting Tablespaces](#)
Select the **Encrypt Tablespace** check box on the Map Tablespaces screen to encrypt all the new tablespaces that are created by RCU.

Default Tablespace Mappings

In the *Default Tablespace* and *Temp Tablespace* columns, you can click on the tablespace cell to select from a list of available additional tablespace names.

The default tablespace mapping for each component are shown in [About Repository Creation Utility Schemas, IDs, and Tablespaces](#).

Changing Default and Temporary Tablespaces

To change the default tablespace for a component, select the tablespace name in the Default Tablespace column, then select the tablespace name that you want to use from the drop-down list.

You can have your components use as many or as few tablespaces as desired to suit your configuration.

To change the temporary tablespace for a component, select the tablespace name in the Temp Tablespace column, then select the tablespace name that you want to use from the drop-down list.

Viewing and Changing Additional Tablespaces

To view and change additional tablespaces associated with the selected components, click the **Additional Tablespaces** button. Only those components with additional tablespaces as defined in the configuration files appear on this screen.

Some components have additional tablespaces associated with their schemas. If this is the case, the **Additional Tablespaces** button appears on this screen. If none of the selected components have additional tablespaces, then this button will not appear.

To change the tablespace you want to use for a component, click in the *Tablespace Name* column and select the tablespace that you want to use from the drop-down list.

Click **OK** when you are finished.

Managing Tablespaces and Datafiles

To manage your tablespaces and datafiles, click the **Manage Tablespaces** button.

The following topics are covered in this section:

- [Adding, Modifying, and Removing Tablespaces](#)
Only tablespaces that are created by RCU can be modified or removed. Tablespaces that existed before RCU was launched are visible on this screen but are grayed out and cannot be modified or removed.
- [Adding, Modifying, and Removing Datafiles](#)
In the Datafiles section, specify the datafiles that make up the selected tablespace.

Adding, Modifying, and Removing Tablespaces

Only tablespaces that are created by RCU can be modified or removed. Tablespaces that existed before RCU was launched are visible on this screen but are grayed out and cannot be modified or removed.

Only tablespaces that are used by a component are created. You can specify a new tablespace here, but unless it is actually used by a component it is not created.

To modify a tablespace, select the tablespace name on the left-hand portion of the screen, and edit the fields as described in the following table:

Field	Description
Name	Edit the tablespace name in this field to change the name of your tablespace.
Type	Specify whether you want this tablespace to be a temporary tablespace or permanent tablespace.
Block Size (KB)	Specify the block size (in Kilobytes) to be used for data retrieval.
Storage Type	Select Use Bigfile Tablespace if you want to create a bigfile tablespace; this is typically used if you have single large files instead of multiple small files. Select Use Automatic Segment Space Management if you want to use bitmaps to manage the free space within segments.

To add a tablespace, click **Add** and specify the same details as above (for modifying a tablespace) for your new tablespace.

To remove a tablespace, select the tablespace name from the navigation tree, then click **Remove**. This tablespace is not created.

Adding, Modifying, and Removing Datafiles

In the Datafiles section, specify the datafiles that make up the selected tablespace.

To add a datafile, click the icon with the plus sign (+):



The Add Datafile screen appears:

Provide the information described in the following table:

Field	Description
File Name	Specify the name of the datafile.

 **Note:**

Datafile names with a dash or hyphen (-) character are not permitted.

Field	Description
File Directory	Specify the location where this datafile will reside.
Size	Specify the initial size of the datafile. Use the drop-down list to specify the size in kilobytes (KB), megabytes (MB), or gigabytes (GB).
Automatically extend datafile when full (AUTOEXTEND)	<p>Select Automatically extend datafile when full (AUTOEXTEND) to automatically extend the size of your datafile when it becomes full. In the <i>Increment</i> field, specify the size by which your datafile should be increased each time it becomes full. Use the drop-down list to specify the size in kilobytes (KB), megabytes (MB), or gigabytes (GB).</p> <p>If you want to limit maximum size of the datafile, specify this value in the <i>Maximum Size</i> field.</p>

Similarly, to modify or edit a datafile, select the icon next to the datafile name that you want to edit, then click the icon with the pencil:



To delete a datafile, select the icon next to the datafile name that you want to delete, then click the icon with the "X":



Encrypting Tablespaces

Select the **Encrypt Tablespace** check box on the Map Tablespaces screen to encrypt all the new tablespaces that are created by RCU.

When you start RCU, the **Encrypt Tablespace** check box appears on the [Map Tablespaces](#) screen only if you have TDE (Transparent Data Encryption) enabled in the database (Oracle or Oracle EBR) when you start RCU. If TDE is not enabled, this check box does not appear. If TDE is enabled in the database and you provide the `-encryptTablespace true` option, the check box is selected by default.

TDE tablespace encryption allows you to encrypt sensitive data stored in tablespaces. For more information about Transparent Data Encryption, see Introduction to Transparent Data Encryption in *Oracle Database Advanced Security Guide*.

Note:

When you select this option, *all new* tablespaces created by RCU are encrypted.

You cannot select specific tablespaces on this screen to be encrypted. Also, tablespaces that were created before RCU was launched cannot be encrypted by using this option.

Summary (for Create Operation)

On the Summary (for Create operation) screen, verify your information, then click **Create** to begin schema creation.

Summary screen that appears when you select **Create Repository** on the [Create Repository](#) screen.

Click **Save Response File** to save the RCU session to a text file (called a response file). The response file collects all the information that you have provided in the RCU screens.

The values in a response file can be used exactly as-is (or modified as needed) to run RCU at a later time. If you choose to run the graphical interface, the data provided in a response file can be used to prepopulate the RCU screens. If you choose to run RCU from the command line, you can use a response file to provide values for all valid command-line parameters to RCU rather than directly specifying these parameters on the command line. For more information, see [Using Response Files](#).

When you click **Save Response File**, you see the following screen:

Figure A-1 Save Repository File Dialogue Box



The following table describes the fields on this screen:

Element	Description
Directory to save Response file	Specify the full path of a directory. This directory is where the response file is saved. By default, RCU saves the response file in the \$HOME/.rcu directory. Click Browse to search for a specific directory.
Save Passwords in Wallet	Select this option to securely store the passwords that you entered on the RCU screens in an Oracle Wallet file. For example, the schema passwords entered on the Schema Passwords screen are saved to this file. By default, this option is not selected. RCU does not save your passwords unless you select this option. You can use the wallet later in silent situations to pass the necessary passwords to RCU without having to directly enter your passwords on the command line. If a wallet is not passed to RCU in silent mode, you are prompted to enter the required passwords.

Click **OK** to generate the wallet (if selected) and response file in the given location and return to the **Summary screen**.

Review the information on the Summary screen, and click **Create** to begin schema creation. When you click **Create**, the operations summarized on this page are performed .

While the schemas are being created, you will see the progress screen:

Click **Stop** to stop creating the schemas.

Summary (for Drop Operation)

On the Summary (for Drop operation) screen, verify your information, then click **Drop** to drop the schemas.

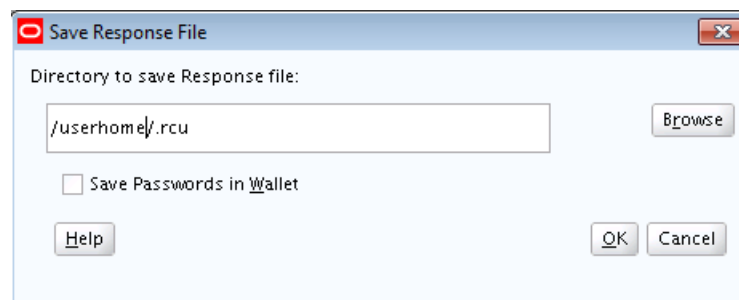
If you select **Drop Repository** on the [Create Repository](#) screen, the **Summary screen** appears:

Click **Save Response File** to save the RCU session to a text file (called a response file). The response file collects all the information that you have provided in the RCU screens.

The values in a response file can be used exactly as-is (or modified as needed) to run RCU at a later time. If you choose to run the graphical interface, the data provided in a response file can be used to prepopulate the RCU screens. If you choose to run RCU from the command line, you can use a response file to provide values for all valid command-line parameters to RCU rather than directly specifying these parameters on the command line. For more information, see [Using Response Files](#).

When you click **Save Response File**, you see the following screen:

Figure A-2 Save Repository File Dialogue Box



The following table describes the fields on this screen:

Element	Description
Directory to save Response file	Specify the full path of a directory. This directory is where the response file is saved. By default, RCU saves the response file in the \$HOME/.rcu directory. Click Browse to search for a specific directory.

Element	Description
Save Passwords in Wallet	<p>Select this option to securely store the passwords that you entered on the RCU screens in an Oracle Wallet file. For example, the schema passwords entered on the Schema Passwords screen are saved to this file.</p> <p>By default, this option is not selected. RCU does not save your passwords unless you select this option.</p> <p>You can use the wallet later in silent situations to pass the necessary passwords to RCU without having to directly enter your passwords on the command line. If a wallet is not passed to RCU in silent mode, you are prompted to enter the required passwords.</p>

Click **OK** to generate the wallet (if selected) and response file in the given location and return to the **Summary screen**.

Review the information on the Summary screen, and click **Drop** to begin the operations summarized on this page.

While the schema(s) are being dropped, you will see the progress screen:

Click **Stop** if you want to cancel the operation.

Completion Summary (for Create Operation)

The Completion Summary (for Create Operation) screen contains information about the log files that were created from this RCU operation. You can click on the name of a particular log file to view the contents of that file.

If you selected **Create Repository** on the [Create Repository](#) screen, **Completion Summary** screen appears:

If there are any problems encountered during schema creation, you can troubleshoot the issue by using the log files. For more information about using log files, see [RCU Log Files](#).

If errors are encountered during a create operation, or if a create operation fails for any component, the **Cleanup for failed components** and **Drop tablespaces for failed components** check boxes appear on this page and are selected by default:

If both options are selected, RCU performs cleanup operations and drops tablespaces for the components that failed during the create operation.

If you select **Cleanup for failed components** but choose not to select **Drop tablespaces for failed components**, the tablespaces for the failed components are retained during cleanup.

If you choose not to select the **Cleanup for failed components** check box, the cleanup of failed components and tablespaces are skipped. You can clean up the failed components at a later time by performing a drop operation for the failed components.

Review the information on this screen, then click **Close** to dismiss this screen.

Completion Summary (for Drop Operation)

The Completion Summary (for Drop operation) screen contains information about the log files that are created from this RCU operation. Note the log file names for each component that are visible in the *Logfile* column.

If you select **Drop Repository** on the [Create Repository](#) screen, **Completion Summary** screen appears:

By default, the main RCU log (*rcu.log*) and component log files are written to the *RCUdate_timestamp_random_number/logs* directory inside the temporary directory on your system.

For example:

- On UNIX operating systems:

```
/tmp/RCUdate_timestamp_random_number/logs
```

- On Windows operating systems:

```
C:\Users\user_name\AppData\Local\Temp\RCUdate_timestamp_random_number\  
logs
```

Click **rcu.log** to view the contents of the main RCU log in a separate window.

If there are any problems encountered during schema creation, you can troubleshoot the issue using the log files. For more information about using log files, see [Troubleshooting Repository Creation Utility](#).

Review the information on this screen, then click **Close** to dismiss this screen.

B

About Repository Creation Utility Schemas, IDs, and Tablespaces

Before you run RCU, make sure you understand the available schemas that can be created by using RCU, and also their component IDs and dependencies. Following sections explain about the RCU schemas and its dependencies:

- [Repository Creation Utility Schemas, IDs, and Tablespaces](#)
Each schema that can be created by using RCU has a schema owner, component ID, and default tablespace mappings. You should also be aware of schema dependencies when you create or drop schemas.

Repository Creation Utility Schemas, IDs, and Tablespaces

Each schema that can be created by using RCU has a schema owner, component ID, and default tablespace mappings. You should also be aware of schema dependencies when you create or drop schemas.

[Table B-1](#) lists the schemas along with their component IDs, tablespace mappings, and dependencies.

The *Schema Owner* is the name of the schema that you see in RCU, and is also the name that you must provide in the Oracle Fusion Middleware Configuration Wizard on the JDBC Component Schema screen, prepended by the custom prefix.

The *Component ID* is the value that you must specify with the `-component` parameter when you create or drop schemas from the command line.

Note:

Not all schemas are supported on all database types. For more information about supported schemas, see *Repository Creation Utility Requirements* in the *Oracle Fusion Middleware System Requirements and Specifications* document.

Table B-1 Schema Component IDs, Tablespace Mappings, and Dependencies

Component	Schema Owner	Component ID	Default Tablespace	Temp Tablespace	Dependencies
AS Common Schemas					
Metadata Services	MDS	MDS	MDS	IAS_TEMP	None

Table B-1 (Cont.) Schema Component IDs, Tablespace Mappings, and Dependencies

Component	Schema Owner	Component ID	Default Tablespace	Temp Tablespace	Dependencies
Audit Services	IAU	IAU	IAU	IAS_TEMP	Audit Services Append (IAU_APPEND) Audit Services Viewer (IAU_VIEWER)
Audit Services Append	IAU_APPEND	IAU_APPEND	IAU	IAS_TEMP	None
Audit Services Viewer	IAU_VIEWER	IAU_VIEWER	IAU	IAS_TEMP	None
Oracle Platform Security Services	OPSS	OPSS	IAS_OPSS	IAS_TEMP	Audit Services (IAU) Audit Services Append (IAU_APPEND) Audit Services Viewer (IAU_VIEWER)
User Messaging Service	UMS	UCSUMS	IAS_UMS	IAS_TEMP	Metadata Services (MDS)
WebLogic Services	WLS	WLS	WLS	IAS_TEMP	None
Oracle Enterprise Scheduler	ESS	ESS	ESS	IAS_TEMP	Metadata Services (MDS) Audit Services (IAU) Audit Services Append (IAU_APPEND) Audit Services Viewer (IAU_VIEWER) Oracle Platform Security Services (OPSS)
Common Infrastructure Services (formerly Service Table) ¹	STB	STB	STB	IAS_TEMP	None
Oracle Data Integrator					
Master and Work Repository	ODI_REPO	ODI	ODI_USER	ODI_TEMP	
SOA Suite					

Table B-1 (Cont.) Schema Component IDs, Tablespace Mappings, and Dependencies

Component	Schema Owner	Component ID	Default Tablespace	Temp Tablespace	Dependencies
SOA Infrastructure	SOAINFRA	SOAINFRA	SOAINFRA	IAS_TEMP	Metadata Services (MDS) Audit Services (IAU) Audit Services Append (IAU_APPEND) Audit Services Viewer (IAU_VIEWER) Oracle Platform Security Services (OPSS) User Messaging Service (UCSUMS) WebLogic Services (WLS)
Managed File Transfer					
Managed File Transfer	MFT	MFT	MFT	IAS_TEMP	Metadata Services (MDS) Audit Services (IAU) Audit Services Append (IAU_APPEND) Audit Services Viewer (IAU_VIEWER) Oracle Platform Security Services (OPSS) User Messaging Service (UCSUMS) Oracle Enterprise Scheduler (ESS)
Oracle Business Intelligence					

Table B-1 (Cont.) Schema Component IDs, Tablespace Mappings, and Dependencies

Component	Schema Owner	Component ID	Default Tablespace	Temp Tablespace	Dependencies
Business Intelligence Platform	BIPLATFORM	BIPLATFORM	BIPLATFORM	IAS_TEMP	Metadata Services (MDS) Oracle Platform Security Services (OPSS) Audit Services (IAU) Audit Services Append (IAU_APPEND) Audit Services Viewer (IAU_VIEWER) WebLogic Services (WLS)
WebCenter Portal					
Portal and Services	WEBCENTER	WEBCENTER	IAS_WEBCENTER	IAS_TEMP	Metadata Services (MDS) Oracle Platform Security Services (OPSS) Audit Services (IAU) Audit Services Append (IAU_APPEND) Audit Services Viewer (IAU_VIEWER)
Portlet Producers	PORTLET	PORTLET	IAS_PORTLET	IAS_TEMP	Oracle Platform Security Services (OPSS) Audit Services (IAU) Audit Services Append (IAU_APPEND) Audit Services Viewer (IAU_VIEWER)

Table B-1 (Cont.) Schema Component IDs, Tablespace Mappings, and Dependencies

Component	Schema Owner	Component ID	Default Tablespace	Temp Tablespace	Dependencies
Analytics	ACTIVITIES	ACTIVITIES	IAS_ACTIVI TY	IAS_TEMP	Oracle Platform Security Services (OPSS) Audit Services (IAU) Audit Services Append (IAU_APPEND) Audit Services Viewer (IAU_VIEWER)
Discussions	DISCUSSIONS	DISCUSSIONS	IAS_DISCUS S	IAS_TEMP	Oracle Platform Security Services (OPSS) Audit Services (IAU) Audit Services Append (IAU_APPEND) Audit Services Viewer (IAU_VIEWER)
WebCenter Content					
Oracle WebCenter Content Server — Search Only	OCSSEARCH	CONTENTSEARC H	OCSSEARCH	OCSSEARCH_ TEMP	None
Oracle WebCenter Content Server — Complete	OCS	CONTENT	OCS	OCS_TEMP	Oracle Platform Security Services (OPSS) Audit Services (IAU) Audit Services Append (IAU_APPEND) Audit Services Viewer (IAU_VIEWER)

Table B-1 (Cont.) Schema Component IDs, Tablespace Mappings, and Dependencies

Component	Schema Owner	Component ID	Default Tablespace	Temp Tablespace	Dependencies
Oracle WebCenter Enterprise Capture	CAPTURE	CAPTURE	CAPTURE	CAPTURE_TEMP	Metadata Services (MDS) Oracle Platform Security Services (OPSS) Audit Services (IAU) Audit Services Append (IAU_APPEND) Audit Services Viewer (IAU_VIEWER)
WebCenter Sites					
WebCenter Sites	WCSITES	WCSITES	TS_WCSITES	TS_TMP_WCSITES	Oracle Platform Security Services (OPSS) Audit Services (IAU) Audit Services Append (IAU_APPEND) Audit Services Viewer (IAU_VIEWER)
WebCenter Sites — Visitor Services	WCSITES_VS	WCSITESVS	TS_WCSITES_VS	TS_TMP_WCSITES_VS	Oracle Platform Security Services (OPSS) Audit Services (IAU) Audit Services Append (IAU_APPEND) Audit Services Viewer (IAU_VIEWER)
Oracle GoldenGate					
Repository	OGGSTUDIO_REPO	OGGSTUDIO	OGGSTUDIO_USER	OGGSTUDIO_TEMP	None

Table B-1 (Cont.) Schema Component IDs, Tablespace Mappings, and Dependencies

Component	Schema Owner	Component ID	Default Tablespace	Temp Tablespace	Dependencies
Monitor Server	OGGMON	OGGMON	OGGMON	IAS_TEMP	Oracle Platform Security Services (OPSS) Audit Services (IAU) Audit Services Append (IAU_APPEND) Audit Services Viewer (IAU_VIEWER)
Veridata Repository	VERIDATA	VERIDATA	VERIDATA_USER	VERIDATA_TEMP	Oracle Platform Security Services (OPSS) Audit Services (IAU) Audit Services Append (IAU_APPEND) Audit Services Viewer (IAU_VIEWER)
Oracle Enterprise Data Quality (EDQ)					
EDQ Config Repository	EDQCONFIG	EDQ_CONF	EDQ_CONF	EDQ_CONFTEMP	Oracle Platform Security Services (OPSS) Audit Services (IAU) Audit Services Append (IAU_APPEND) Audit Services Viewer (IAU_VIEWER)

Table B-1 (Cont.) Schema Component IDs, Tablespace Mappings, and Dependencies

Component	Schema Owner	Component ID	Default Tablespace	Temp Tablespace	Dependencies
EDQ Results Repository	EDQRESULTS	EDQ_RES	EDQ_RES	EDQ_RESTEM P	Oracle Platform Security Services (OPSS) Audit Services (IAU) Audit Services Append (IAU_APPEND) Audit Services Viewer (IAU_VIEWER) EDQ Config Repository (EDQ_CONF)
EDQ Staging Repository	EDQSTAGING	EDQ_STAGING	EDQ_STAGING	EDQ_STAGING GTEMP	Oracle Platform Security Services (OPSS) Audit Services (IAU) Audit Services Append (IAU_APPEND) Audit Services Viewer (IAU_VIEWER) EDQ Config Repository (EDQ_CONF) EDQ Results Repository (EDQ_RES)

¹ For more information about the Service Table schema, see [About the Service Table Schema](#).

C

Extending Repository Creation Utility to Configure Custom Application Repositories

You can use configuration XML files to generate custom schema creation and deletion scripts.

RCU provides an XML-based framework for component owners to plug in your schema creation and deletion scripts into RCU. The following topics provide some details of the configuration XML files and script-writing guidelines that are used to integrate your components with RCU:

- [RCU Integration Options](#)
RCU provides several options for integrating component scripts.
- [RCU Configuration Files](#)
RCU provides several configuration file types for component integration.
- [RCU Script Writing Guidelines](#)
Use these guidelines when writing RCU scripts.

RCU Integration Options

RCU provides several options for integrating component scripts.

RCU JDBC Engine Compliant SQL*Plus Scripts is the recommended option for integrating component scripts. SQL*Plus and External Processes are only intended for integrating Legacy or Classic components such as Oracle Portal 10g or Identity Management. Components that have a dependency on SQL*Plus scripts cannot be loaded with RCU when running from the installed Oracle home. They can only be used when running RCU from CD.

- [RCU JDBC Engine Compliant SQL*Plus Scripts](#)
The RCU JDBC Engine emulates a set of SQL*Plus features over JDBC. This set is broad enough to cover the requirements of schema creation.
- [Pure JDBC Scripts](#)
This option is recommended for non-Oracle databases (for Oracle databases, RCU JDBC Engine Compliant SQL*Plus scripts should be used).
- [SQL*Plus Scripts](#)
This option is mainly for the consumption of legacy components that need to be loaded from RCU.
- [External Processes](#)
This option is provided only for those components that have their own configuration tool for schema creation, like OPCA (Oracle Portal 10g).
- [Java Code Using JavaAction](#)
This option is provided to components that have Java code, which can accept a JDBC connection and execute SQL statements.

RCU JDBC Engine Compliant SQL*Plus Scripts

The RCU JDBC Engine emulates a set of SQL*Plus features over JDBC. This set is broad enough to cover the requirements of schema creation.

Your component teams can integrate existing SQL*Plus scripts with a few minor changes.

The RCU JDBC Engine parses the SQL*Plus script to get individual statements and then runs each statement over JDBC. Command-line arguments to scripts and substitution using DEFINE variables are supported. Script can be nested, for example, one script can call other scripts. Component teams can specify a list of expected errors and fatal errors to RCU through configuration files and RCU would interpret these when running the scripts.

These scripts are easy to maintain and use as they can be run in SQL*Plus in a development environment. However, Oracle recommends that you also use the *RCU JDBC Engine* tool in your development environment to ensure that these scripts run properly when integrated with RCU.

Pure JDBC Scripts

This option is recommended for non-Oracle databases (for Oracle databases, RCU JDBC Engine Compliant SQL*Plus scripts should be used).

Contents of the script file should be a valid PL/SQL block, which can be called with `Connection.prepareCall()` or `Connection.createStatement()`. Standard JDBC Bind variables with '?' convention are supported.

Some disadvantages of this option are:

- No nested scripts, which can mean a larger number of scripts.
- May require significant rework for component teams to rewrite the scripts in this format.
- Difficult to maintain as every DDL statement has to be wrapped with in *EXECUTE IMMEDIATE*.
- Cannot be run by using SQL*Plus in a development environment.
- Less useful error support since the whole block would fail in case of any errors.

Below is an example:

```
<Action TYPE="JDBC" PERCENT_PROGRESS="20">
  <ValidIf DBTYPE="ORACLE" />
  <Command TYPE="INLINE">DROP USER %SCHEMA_USER% CASCADE</Command>
</Action>
```

And a second example:

```
<Action TYPE="Java" PERCENT_PROGRESS="100">
  <Command TYPE="METHOD">
    oracle.ias.version.SchemaVersionUtil:utilSetComponentValid
  </Command>

  <Parameters>
    <Parameter TYPE="String">MDS</Parameter>
```

```

    </Parameters>
  </Action>

```

SQL*Plus Scripts

This option is mainly for the consumption of legacy components that need to be loaded from RCU.

Example:

```

<Action TYPE="SQLPlus" PERCENT_PROGRESS="100">
  <Command TYPE="SCRIPT">%SCRIPT_HOME%/oid/scripts/seedldap.sql</Command>
  <IgnorableErrors>
    <Error Type="ORA-01918">user name does not exist</Error>
  </IgnorableErrors>
</Action>

```

And a second example:

```

<Action TYPE="JDBCsqlScript" PERCENT_PROGRESS="20">
  <ValidIf DBTYPE="ORACLE" />
  <Command TYPE="SCRIPT">%SCRIPT_HOME%/mds/sql/mds_user.sql</Command>
  <Parameters>
    <Parameter TYPE="CmdLine">%SCHEMA_USER%</Parameter>
    <Parameter TYPE="CmdLine">%SCHEMA_PASSWORD%</Parameter>
    <Parameter TYPE="CmdLine">%DEFAULT_TABLESPACE%</Parameter>
    <Parameter TYPE="CmdLine">%TEMPORARY_TABLESPACE%</Parameter>
  </Parameters>
</Action>

```

External Processes

This option is provided only for those components that have their own configuration tool for schema creation, like OPCA (Oracle Portal 10g).

This is not a recommended option for any new component, as this option cannot make use of the RCU error handling framework.

Example:

```

<Action TYPE="HostCmd">
  <Command TYPE="SCRIPT">%RCU_HOME%/rcu/integration/cdb/config/bin/configure</Command>
  <Parameters>
    <Parameter TYPE="ProcessInput">%JDBC_CONNECT_STRING%</Parameter>
    <Parameter TYPE="ProcessInput">%DBADMIN_USER%</Parameter>
    <Parameter TYPE="ProcessInput">%DBADMIN_PASSWORD%</Parameter>
    <Parameter TYPE="ProcessInput">%PREFIX_NAME%</Parameter>
    <Parameter TYPE="ProcessInput">%SCHEMA_USER%</Parameter>
    <Parameter TYPE="ProcessInput">%SCHEMA_PASSWORD%</Parameter>
    <Parameter TYPE="ProcessInput">%DEFAULT_TABLESPACE%</Parameter>
    <Parameter TYPE="ProcessInput">%TEMPORARY_TABLESPACE%</Parameter>
  </Parameters>
</Action>

```


Java Code Using JavaAction

This option is provided to components that have Java code, which can accept a JDBC connection and execute SQL statements.

This is generally used when huge amounts of data has to be seeded or Large Objects (LOBs) need to be created.

Example:

```
<Action TYPE="Java">
  <Command TYPE="METHOD">
    oracle.ias.version.SchemaVersionUtil:utilCreateRegistryEntry
  </Command>

  <Parameters>
    <Parameter TYPE="Connection"></Parameter>
    <Parameter TYPE="String">%SCHEMA_USER%</Parameter>
  </Parameters>
</Action>
```

A second example:

```
<Action TYPE="Java">
  <Command TYPE="METHOD">oracle.webdb.config.PortalConfigAssistant:main</Command>
  <Parameters>
    <Parameter TYPE="StringArray">-mode PORTAL -s %SCHEMA_USER% -p
%DBADMIN_PASSWORD% -c %DB_HOSTNAME%:%DB_PORTNUMBER%:%DB_SERVICE% -silent -
verbose -owa -u %DEFAULT_TABLESPACE% -t %TEMPORARY_TABLESPACE% -d
%SCHEMA_USER%_DOC -l %SCHEMA_USER%_LOG -in %SCHEMA_USER%_IDX -demo -report -voh
%RCU_HOME% -log %RCU_LOG_LOCATION% -oh %SQLPLUS_HOME% -mrc %PREFIX_NAME% -rcu </
Parameter>
  </Parameters>
</Action>
```

RCU Configuration Files

RCU provides several configuration file types for component integration.

The following are the configuration files defined by RCU:

- [XML DTDs Defined by RCU](#)
These are the XML DTDs defined by RCU; they are located in the *ORACLE_HOME/oracle_common/rcu/config* directory.
- [Component Repository Configuration File](#)
A Component Repository Configuration File (*component.xml*) lists the prerequisites and the list of scripts or actions that need to be performed to load or drop a schema.
- [Component List Configuration File](#)
The Component List configuration file (*ComponentInfo.xml*) lists all the components, their respective configuration files and their default user and tablespace mappings.

- [Soft-Prerequisite Support](#)
If a particular <DBPrerequisiteSet> or <DBPrerequisite> is not mandatory, then you can use the soft-prerequisite option by setting the SOFT attribute to TRUE in the ComponentInfo.xml file
- [Default Tablespaces Configuration File](#)
The default tablespaces configuration file (Storage.xml) lists the components, for which, tablespaces are created out-of-the-box. This file is located in the `ORACLE_HOME/oracle_common/rcu/config` directory on UNIX operating systems or in the `ORACLE_HOME\oracle_common\rcu\config` directory on Windows operating systems.

XML DTDs Defined by RCU

These are the XML DTDs defined by RCU; they are located in the `ORACLE_HOME/oracle_common/rcu/config` directory.

Following are the configuration files:

- [Component Descriptor Configuration File](#)
The Component Descriptor configuration file is called ComponentInfo.dtd and is located in the `ORACLE_HOME/oracle_common/rcu/config` directory on UNIX operating systems or in the `ORACLE_HOME\oracle_common\rcu\config` directory on Windows operating systems.
- [Repository Configuration File](#)
The Repository configuration file is called RepositoryConfig.dtd and is located in the `ORACLE_HOME/oracle_common/rcu/config` directory on UNIX operating systems directory or in the `ORACLE_HOME\oracle_common\rcu\config` directory on Windows operating systems.
- [Master List of Supported Components](#)
RCU maintains a master list of supported components, which contains entries for each supported component. Every time a new component is added, the master list of supported components is updated with the reference of the XML integration file, which is provided by component owner.
- [Storage Attributes Configuration File](#)
RCU maintains the list of tablespaces or datafiles and their attributes that needs to be created. This way the tablespaces and datafiles attributes can be modified externally.

Component Descriptor Configuration File

The Component Descriptor configuration file is called ComponentInfo.dtd and is located in the `ORACLE_HOME/oracle_common/rcu/config` directory on UNIX operating systems or in the `ORACLE_HOME\oracle_common\rcu\config` directory on Windows operating systems.

Each component owner would provides a configuration file that adheres to the following DTD, which lists the prerequisites and actions:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!ENTITY % commonDTD SYSTEM "RCUCommon.dtd">
%commonDTD;
<!ELEMENT ComponentInfo (Display?, PrefixSettings?, Component*,
ComponentConfigFiles?, PrerequisiteDescriptor*,
ExecutionDescriptor*, FatalErrors?, IgnorableErrors?)>
```

```

<!ATTLIST ComponentInfo
    VERSION CDATA #IMPLIED
    TYPE CDATA #IMPLIED
    RESOURCE_BUNDLE_PACKAGE CDATA #IMPLIED>
<!ELEMENT PrefixSettings (DetectQuery*)>
<!ATTLIST PrefixSettings
    USE_SCHEMA_PREFIX (TRUE|FALSE) "TRUE"
    USE_TABLESPACE_PREFIX (TRUE|FALSE) "TRUE">
<!ELEMENT ComponentConfigFiles (ConfigFile*)>
<!ATTLIST ConfigFile
    ID CDATA #REQUIRED>
<!ELEMENT ConfigFile (#PCDATA)>
<!ELEMENT Component (ValidIfSet?, ValidIf?, Display,Consolidate*,SchemaBase*,
RepositoryConfigFile?, DetectQuery*, SchemaVersion?,
SchemaUser?, AdditionalSchemaUser*, Dependents?, DatabaseName?, Tablespaces?,
CustomVariables?)>
<!ATTLIST Component
    ID CDATA #REQUIRED
    PROGRESS_UNITS CDATA #IMPLIED
    IS_GROUPING_COMPONENT (TRUE|FALSE) "FALSE"
    DEFAULT_SELECTED (TRUE|FALSE) "FALSE"
    CHILD_OF CDATA #IMPLIED
    MANDATORY_SELECTED (TRUE|FALSE) "FALSE">
<!ELEMENT Display (#PCDATA)>
<!ATTLIST Display
    NLS_ID CDATA #IMPLIED>
<!ELEMENT Consolidate EMPTY>
<!ATTLIST Consolidate
    VALID (TRUE|FALSE) "FALSE"
    DYNAMIC_OBJECTS (TRUE|FALSE) "FALSE">
<!ELEMENT SchemaBase (#PCDATA) >
<!ATTLIST SchemaBase
    PARENT (TRUE|FALSE) "FALSE"
    TBS_OVERRIDE (TRUE|FALSE) "FALSE"
    TBS_OVERRIDE_USE_CUSTOM (TRUE|FALSE) "FALSE"
    COMPONENT_DROP (TRUE|FALSE) "FALSE">
<!ELEMENT RepositoryConfigFile (#PCDATA)>
<!ELEMENT DetectQuery (#PCDATA)>
<!ATTLIST DetectQuery
    OPERATION (CREATE|DROP|DATALOAD) 'CREATE'
    TYPE (ORACLE|SQLSERVER|IBMDB2|EBR|MYSQL|JAVADB) 'ORACLE'>
<!ELEMENT SchemaVersion (#PCDATA)>
<!ELEMENT SchemaUser (#PCDATA)>
<!ATTLIST SchemaUser
    USER_EDITABLE (TRUE|FALSE) "TRUE"
    PREFIXABLE (TRUE|FALSE) "TRUE"
    IS_CREATED (TRUE|FALSE) "TRUE"
    MAX_LENGTH CDATA "30">
<!ELEMENT AdditionalSchemaUser (#PCDATA)>
<!ATTLIST AdditionalSchemaUser
    STARTS_WITH_SCHEMA_USER (TRUE|FALSE) "TRUE"
    COMPONENT_ID CDATA #IMPLIED>
<!ELEMENT Dependents (Dependent*)>
<!ELEMENT Dependent (#PCDATA)>
<!ATTLIST Dependent
    COMPONENT_ID CDATA #REQUIRED
    ALT_COMPONENT_ID CDATA #IMPLIED>
<!ELEMENT DatabaseName (#PCDATA)>
<!ELEMENT Tablespaces (Tablespace*)>
<!ATTLIST Tablespace TYPE (DEFAULT_TABLESPACE|TEMPORARY_TABLESPACE|
ADDITIONAL_TABLESPACE1|ADDITIONAL_TABLESPACE2|ADDITIONAL_TABLESPACE3 |

```

```

ADDITIONAL_TABLESPACE4|ADDITIONAL_TABLESPACE5|ADDITIONAL_TABLESPACE6|
ADDITIONAL_TABLESPACE7|ADDITIONAL_TABLESPACE8|ADDITIONAL_TABLESPACE9|
ADDITIONAL_TABLESPACE10|ADDITIONAL_TABLESPACE11|ADDITIONAL_TABLESPACE12|
ADDITIONAL_TABLESPACE13|ADDITIONAL_TABLESPACE14|ADDITIONAL_TABLESPACE15|
ADDITIONAL_TABLESPACE16|ADDITIONAL_TABLESPACE17|ADDITIONAL_TABLESPACE18|
ADDITIONAL_TABLESPACE19|ADDITIONAL_TABLESPACE20|ADDITIONAL_TABLESPACE21|
ADDITIONAL_TABLESPACE22|ADDITIONAL_TABLESPACE23|ADDITIONAL_TABLESPACE24|
ADDITIONAL_TABLESPACE25|ADDITIONAL_TABLESPACE26|ADDITIONAL_TABLESPACE27|
ADDITIONAL_TABLESPACE28|ADDITIONAL_TABLESPACE29|ADDITIONAL_TABLESPACE30|
ADDITIONAL_TABLESPACE31) "DEFAULT_TABLESPACE">
<!ELEMENT Tablespace (Prompt, TablespaceName)>
<!ELEMENT Prompt (#PCDATA)>
<!ATTLIST Prompt NLS_ID CDATA #IMPLIED>
<!ELEMENT TablespaceName (#PCDATA)>
<!ELEMENT CustomVariables (Variable*, ValidationDescriptor*)>
<!ATTLIST Variable
    TYPE (STRING|NUMBER|PASSWORD|EXISTING_PASSWORD|FILE|LIST) "STRING"
    MAX_LENGTH CDATA "30"
    MIN_LENGTH CDATA "0"
    MAX_VALUE CDATA #IMPLIED
    MIN_VALUE CDATA #IMPLIED
    OPERATION (CREATE|DROP|SYSTEM_LOAD|DATA_LOAD|BOTH) "BOTH"
    DEFAULT_VALUE CDATA ""
    PROMPT_ON_DEFAULT_VALUE (YES|NO) "YES"
    VALUES CDATA "">
<!ELEMENT Variable (Name,Display)>
<!ELEMENT ValidationDescriptor (Action*)>
<!ATTLIST ValidationDescriptor
    OPERATION (CREATE|DROP|SYSTEM_LOAD|DATA_LOAD|BOTH) "BOTH">
<!ELEMENT Name (#PCDATA)>

```

Repository Configuration File

The Repository configuration file is called `RepositoryConfig.dtd` and is located in the `ORACLE_HOME/oracle_common/rcu/config` directory on UNIX operating systems directory or in the `ORACLE_HOME\oracle_common\rcu\config` directory on Windows operating systems.

```

<?xml version="1.0" encoding="UTF-8" ?>
<!ENTITY % commonDTD SYSTEM "RCUCommon.dtd">
%commonDTD;
<!ELEMENT RepositoryConfig (PrerequisiteDescriptor*, ExecutionDescriptor,
DeleteDescriptor?)>
<!ATTLIST RepositoryConfig
    COMP_ID CDATA #REQUIRED>
<!ELEMENT DeleteDescriptor (Action*)>

```

Master List of Supported Components

RCU maintains a master list of supported components, which contains entries for each supported component. Every time a new component is added, the master list of supported components is updated with the reference of the XML integration file, which is provided by component owner.

This configuration file is called `RCUCommon.dtd` and is located in the `ORACLE_HOME/oracle_common/rcu/config` directory on UNIX operating systems or in the `ORACLE_HOME\oracle_common\rcu\config` directory on Windows operating systems:

```

<?xml version="1.0" encoding="UTF-8" ?>
<!ELEMENT PrerequisiteDescriptor (DBPrerequisiteSet*, DBPrerequisite*)>
<!ATTLIST PrerequisiteDescriptor
    TYPE (CREATE|DROP|REGISTER|DEREGISTER|SYSTEMLOAD|DATALOAD) 'CREATE'>
<!ELEMENT DBPrerequisiteSet (ValidIfSet?, ValidIf?, PrereqSetErrorMsg?,
DBPrerequisite*)>
<!ATTLIST DBPrerequisiteSet
    OPERATOR (OR|AND) "OR"
    SOFT (TRUE|FALSE) "FALSE">
<!ELEMENT DBPrerequisite (ValidIfSet?, ValidIf?, PrereqIdentifier, PrereqValue,
PrereqErrorMsg?)>
<!ATTLIST DBPrerequisite
PREREQ_TYPE (InitParameter|DBOption|Java|DBComponent|DBVersion|DBObject|
CustomSQL|TablespaceFreeMB) "CustomSQL"
    DATA_TYPE (STRING|NUMBER|VERSION) "STRING"
    COMPARE_OPERATOR (EQ|GT|LT|NE|GE|LE|COMPARE_VERSION) "EQ"
    SOFT (TRUE|FALSE) "FALSE">

<!ELEMENT PrereqIdentifier (#PCDATA)>
<!ELEMENT PrereqValue (#PCDATA)>
<!ELEMENT PrereqSetErrorMsg (#PCDATA)>
<!ATTLIST PrereqSetErrorMsg
    NLS_ID CDATA #IMPLIED>
<!ELEMENT PrereqErrorMsg (#PCDATA)>
<!ATTLIST PrereqErrorMsg
    NLS_ID CDATA #IMPLIED>
<!ATTLIST PrereqValue
    UNIT (KB|MB|NoUnit) 'NoUnit'>
<!ELEMENT ExecutionDescriptor (Action*)>
<!ATTLIST ExecutionDescriptor
    TYPE (Load|PreLoad|PostLoad|DataLoad|PostDataLoad) "Load">
<!ELEMENT Action (ValidIfSet?, ValidIf?, Command, Parameters?, FatalErrors?,
IgnorableErrors?)>
<!ATTLIST Action
    TYPE (JDBCScript|JDBC|SQLPlus|HostCmd|Java) "JDBCScript"
    DB_VERSION CDATA #IMPLIED
    PERCENT_PROGRESS CDATA #IMPLIED
    CONNECT_AS_OWNER (TRUE|FALSE) "FALSE"
    CONNECT_AS_REGISTRYOWNER (TRUE|FALSE) "FALSE"
    RESET_SESSION (TRUE|FALSE) "FALSE">
<!ELEMENT Command (#PCDATA)>
<!ATTLIST Command
    TYPE (SCRIPT|INLINE|METHOD) "SCRIPT">
<!ELEMENT Parameters (Parameter*)>
<!ELEMENT Parameter (#PCDATA)>
<!ATTLIST Parameter
    TYPE (BindVar|CmdLine|ProcessInput|EnvVar|Connection|int|String|
StringArray|boolean) "CmdLine">
<!ELEMENT FatalErrors (Error*)>
<!ELEMENT IgnorableErrors (Error*)>
<!ELEMENT Error (#PCDATA)>
<!ATTLIST Error
    Type CDATA #REQUIRED>
<!ELEMENT ValidIfSet (ValidIf*)>
<!ATTLIST ValidIfSet
    DBTYPE CDATA #IMPLIED
    DBVERSION CDATA #IMPLIED
    OSNAME CDATA #IMPLIED
    OPERATION_TYPE CDATA "ALL"
    OPERATOR (OR|AND) "OR">
<!ELEMENT ValidIf (CustomQueryFilter?)>

```

```

<!ATTLIST ValidIf
  DBTYPE CDATA #IMPLIED
  DBVERSION CDATA #IMPLIED
  OSNAME CDATA #IMPLIED
  OPERATION_TYPE CDATA "ALL">
<!ELEMENT CustomQueryFilter (#PCDATA)>
<!ATTLIST CustomQueryFilter
  DATA_TYPE (STRING|NUMBER) "STRING"
  COMPARE_OPERATOR (EQ|GT|LT|NE|GE|LE|COMPARE_VERSION) "EQ"
  VALUE CDATA #REQUIRED >

```

Storage Attributes Configuration File

RCU maintains the list of tablespaces or datafiles and their attributes that needs to be created. This way the tablespaces and datafiles attributes can be modified externally.

The Storage Attributes configuration file is called `Storage.dtd` and is located in the `ORACLE_HOME/oracle_common/rcu/config` directory on UNIX operating systems) or in the `ORACLE_HOME\oracle_common\rcu\config` directory on Windows operating systems.

```

<?xml version="1.0" encoding="UTF-8" ?>
<!ELEMENT StorageAttributes ( ComponentStorageFiles?, TablespaceAttributes*)>
<!ELEMENT ComponentStorageFiles (StorageFile*)>
<!ATTLIST StorageFile
  ID CDATA #REQUIRED>
<!ELEMENT StorageFile (#PCDATA)>
<!ELEMENT TablespaceAttributes (ValidIfSet?,
ValidIf?,Type?,DefaultTemp?,BlockSize?,ExtentSize?,PageSize?,AutoResize?,Increase
Size?,MaxSize?,Bigfile?,AutoSegmentSpaceManagement?, DatafilesList)>
<!ATTLIST TablespaceAttributes
  NAME CDATA #REQUIRED
  OMF (TRUE|FALSE) "FALSE">
<!ELEMENT ValidIfSet (ValidIf*)>
<!ATTLIST ValidIfSet
  DBTYPE CDATA #IMPLIED
  DBVERSION CDATA #IMPLIED
  OSNAME CDATA #IMPLIED
  OPERATOR (OR|AND) "OR">
<!ELEMENT ValidIf (CustomQueryFilter?)>
<!ATTLIST ValidIf
  DBTYPE CDATA #IMPLIED
  DBVERSION CDATA #IMPLIED
  OSNAME CDATA #IMPLIED >
<!ELEMENT Type (#PCDATA)>
<!ELEMENT DefaultTemp (#PCDATA)>
<!ELEMENT BlockSize (#PCDATA)>
<!ELEMENT ExtentSize (#PCDATA)>
<!ELEMENT PageSize (#PCDATA)>
<!ATTLIST PageSize
  UNIT (KB|NoUnit) 'KB'>
<!ELEMENT AutoResize (#PCDATA)>
<!ELEMENT IncreaseSize (#PCDATA)>
<!ATTLIST IncreaseSize
  UNIT (KB|MB|GB) 'MB'>
<!ELEMENT MaxSize (#PCDATA)>
<!ATTLIST MaxSize
  UNIT (KB|MB|GB) 'MB'>
<!ELEMENT Bigfile (#PCDATA)>
<!ELEMENT AutoSegmentSpaceManagement (#PCDATA)>

```

```

<!ELEMENT DatafilesList (DatafileAttributes+)>
<!ELEMENT DatafileAttributes (Size, Reuse?, AutoExtend?, Increment?, Maxsize?)>
<!ATTLIST DatafileAttributes
  ID CDATA #REQUIRED>
<!ELEMENT Size (#PCDATA)>
<!ATTLIST Size
  UNIT (KB|MB|GB) 'MB'>
<!ELEMENT Reuse (#PCDATA)>
<!ELEMENT AutoExtend (#PCDATA)>
<!ELEMENT Increment (#PCDATA)>
<!ATTLIST Increment
  UNIT (KB|MB|GB) 'KB'>
<!ELEMENT Maxsize (#PCDATA)>
<!ATTLIST Maxsize
  UNIT (KB|MB|GB) 'MB'>

```

Component Repository Configuration File

A Component Repository Configuration File (*component.xml*) lists the prerequisites and the list of scripts or actions that need to be performed to load or drop a schema.

This file is provided and maintained by component owners. This configuration file is referenced from the [Component List Configuration File](#) (*ComponentInfo.xml*).

Each *component.xml* file can be found in the *ORACLE_HOME/oracle_common/common/sql/component/component.xml* directory on UNIX operating systems) or in the *ORACLE_HOME\oracle_common\common\sql\component\component.xml* directory on Windows operating systems.

Component owners can use a set of predefined RCU parameters, which are substituted at runtime by RCU based on user input. Here is the list of predefined parameters:

Table C-1 Predefined RCU Parameters

RCU Parameter	Description
%ORACLE_HOME%	Location of the Oracle home directory.
%SCRIPT_HOME%	Location where scripts are located. It may be same as <i>ORACLE_HOME</i> .
%SCHEMA_USER%	Database schema name (owner) entered by the user in RCU.
%SCHEMA_PASSWORD%	Database schema password entered by the user in RCU.
%ADDITIONAL_SCHEMA_USER%	Additional schema users as defined in the <i>ComponentInfo.xml</i> file
%ADDITIONAL_SCHEMA_PASSWORD<n>%	Password for the additional schema users.
%DEFAULT_TABLESPACE%	Default tablespace assigned to the component by the user.
%TEMPORARY_TABLESPACE%	Temporary tablespace assigned to the component by the user.
%ADDITIONAL_TABLESPACE<n>%	Additional tablespace assigned to the component by the user. Up to three additional tablespaces are supported.
%DEFAULT_PERMANENT_TABLESPACE%	Default permanent tablespace in the database (for example, <i>USERS</i> or <i>SYSTEM</i>) is none is set.
%DEFAULT_TEMP_TABLESPACE%	Default temporary tablespace in the database (for example, <i>TEMP</i> in Oracle shipped databases or <i>SYSTEM</i>) if none is set.
%DATAFILE_LOCATION%	Default location where the tablespace/datafile will be created.

Table C-1 (Cont.) Predefined RCU Parameters

RCU Parameter	Description
%JDBC_CONNECT_STRING%	JDBC connect string.
%PREFIX_NAME%	User-specified prefix for schema and tablespace names.
%CONNECTION%	Already-connected <code>java.sql.Connection</code> object to be passed into <code>JavaAction</code> .
%DBADMIN_USER%	Database admin user that is provided on the Database Connection Details .
%DBADMIN_PASSWORD%	Database admin user password that is provided on the Database Connection Details .
%DBADMIN_ROLE%	Database admin user role that is provided on the Database Connection Details .
%DB_HOSTNAME%	Database hostname that is provided on the Database Connection Details .
%DB_SERVICE%	Database service name.
%DB_PORTNUMBER%	Database port number that is provided on the Database Connection Details .
%RCU_HOME%	Directory where RCU is installed.
%SQLPLUS_HOME%	<code>ORACLE_HOME</code> where SQL*Plus is located.
%RCU_LOG_LOCATION%	Location of the directory where RCU log files are created.
%DATABASE_NAME%	Database name (for SQLServer database).

Below is a sample Component Repository Configuration file for OPSS (`opss.xml`), which lists the series of prerequisites and actions:

```
<?xml version="1.0" encoding="UTF-8" ?>

<!DOCTYPE RepositoryConfig SYSTEM "RepositoryConfig.dtd">
<RepositoryConfig COMP_ID="OPSS">

  <!-- Prerequisites for OPSS -->
  <PrerequisiteDescriptor>
    <!-- ORACLE Prerequisites -->
    <DBPrerequisite PREREQ_TYPE="TablespaceFreeMB" DATA_TYPE="NUMBER"
      COMPARE_OPERATOR="GT">
      <ValidIf DBTYPE="ORACLE" />
      <PrereqIdentifier>%DEFAULT_TABLESPACE%</PrereqIdentifier>
      <PrereqValue>50</PrereqValue>
    </DBPrerequisite>
    <DBPrerequisite PREREQ_TYPE="TablespaceFreeMB" DATA_TYPE="NUMBER"
      COMPARE_OPERATOR="GT">
      <ValidIf DBTYPE="ORACLE" />
      <PrereqIdentifier>%TEMPORARY_TABLESPACE%</PrereqIdentifier>
      <PrereqValue>50</PrereqValue>
    </DBPrerequisite>

    <!-- ORACLE EBR Prerequisites -->
    <DBPrerequisite PREREQ_TYPE="TablespaceFreeMB" DATA_TYPE="NUMBER"
      COMPARE_OPERATOR="GT">
      <ValidIf DBTYPE="EBR" />
      <PrereqIdentifier>%DEFAULT_TABLESPACE%</PrereqIdentifier>
```



```

        <PrereqValue>50</PrereqValue>
    </DBPrerequisite>
    <DBPrerequisite PREREQ_TYPE="TablespaceFreeMB" DATA_TYPE="NUMBER"
        COMPARE_OPERATOR="GT">
        <ValidIf DBTYPE="EBR" />
        <PrereqIdentifier>%TEMPORARY_TABLESPACE%</PrereqIdentifier>
        <PrereqValue>50</PrereqValue>
    </DBPrerequisite>

    <!-- SQLServer Prerequisites -->

    <!-- DB2 Prerequisites -->

</PrerequisiteDescriptor>

<PrerequisiteDescriptor TYPE="DROP">
    <!-- ORACLE Prerequisites -->
    <DBPrerequisite PREREQ_TYPE="CustomSQL" DATA_TYPE="NUMBER"
        COMPARE_OPERATOR="EQ">
        <ValidIf DBTYPE="ORACLE" />
        <PrereqIdentifier>
            select count(*) from v$session where
username='%SCHEMA_USER%'
        </PrereqIdentifier>
        <PrereqValue>0</PrereqValue>
        <PrereqErrorMsg>
            The schema owner %SCHEMA_USER% is connected to the
            database. Please disconnect and try again.
        </PrereqErrorMsg>
    </DBPrerequisite>

    <!-- ORACLE EBR Prerequisites -->
    <DBPrerequisite PREREQ_TYPE="CustomSQL" DATA_TYPE="NUMBER"
COMPARE_OPERATOR="EQ">
        <ValidIf DBTYPE="EBR" />
        <PrereqIdentifier>select count(*) from v$session where
username='%SCHEMA_USER%'</PrereqIdentifier>
        <PrereqValue>0</PrereqValue>
        <PrereqErrorMsg NLS_ID="WC_USER_CONNECTED">The schema owner
%SCHEMA_USER% is connected to the database. Please disconnect and try again.</
PrereqErrorMsg>
    </DBPrerequisite>

    <!-- SQLServer Prerequisites -->

    <!-- DB2 Prerequisites -->

</PrerequisiteDescriptor>

    <!-- Creating the OPSS Schema and Setting it to valid in the Registry -->
    <ExecutionDescriptor>
        <Action TYPE="Java" PERCENT_PROGRESS="20">
            <Command
TYPE="METHOD">oracle.ias.version.SchemaVersionUtil:utilCreateRegistryEntry</
Command>
            <Parameters>
                <Parameter TYPE="Connection"></Parameter>
                <Parameter TYPE="String">OPSS</Parameter>
                <Parameter TYPE="String">Oracle Platform Security Services</

```

```

Parameter>
    <Parameter TYPE="String">%PREFIX_NAME%/Parameter>
    <Parameter TYPE="String">OPSS/Parameter>
    <Parameter TYPE="String">OPSS/Parameter>
    <Parameter TYPE="String">%SCHEMA_USER%/Parameter>
    <Parameter TYPE="String">12.1.2.0.0/Parameter>
    <Parameter TYPE="String">LOADING/Parameter>
  </Parameters>
</Action>
<Action TYPE="JDBCsqlScript" PERCENT_PROGRESS="40">
  <ValidIf DBTYPE="ORACLE" />
  <Command TYPE="SCRIPT">%SCRIPT_HOME%/opss/scripts/opss_user.sql</
Command>
  <Parameters>
    <Parameter TYPE="CmdLine">%SCHEMA_USER%/Parameter>
    <Parameter TYPE="CmdLine">%SCHEMA_PASSWORD%/Parameter>
    <Parameter TYPE="CmdLine">%DEFAULT_TABLESPACE%/Parameter>
    <Parameter TYPE="CmdLine">%TEMPORARY_TABLESPACE%/Parameter>
  </Parameters>
  <IgnorableErrors>
    <Error Type="ORA-01918">user name does not exist</Error>
  </IgnorableErrors>
</Action>
<Action TYPE="JDBCsqlScript" PERCENT_PROGRESS="20">
  <ValidIf DBTYPE="EBR" />
  <Command TYPE="SCRIPT">%SCRIPT_HOME%/opss/scripts/oracleEbr/
opss_user.sql</Command>
  <Parameters>
    <Parameter TYPE="CmdLine">%SCHEMA_USER%/Parameter>
    <Parameter TYPE="CmdLine">%SCHEMA_PASSWORD%/Parameter>
    <Parameter TYPE="CmdLine">%DEFAULT_TABLESPACE%/Parameter>
    <Parameter TYPE="CmdLine">%TEMPORARY_TABLESPACE%/Parameter>
    <Parameter TYPE="CmdLine">%EDITION_NAME%/Parameter>
  </Parameters>
  <IgnorableErrors>
    <Error Type="ORA-01918">user name does not exist</Error>
  </IgnorableErrors>
</Action>

  <Action TYPE="JDBCsqlScript" PERCENT_PROGRESS="40">
    <ValidIf DBTYPE="SQLSERVER" />
    <Command TYPE="SCRIPT">%SCRIPT_HOME%/opss/scripts/sqlserver/
opss_user.sql</Command>
    <Parameters>
      <Parameter TYPE="CmdLine">%SCHEMA_USER%/Parameter>
      <Parameter TYPE="CmdLine">%SCHEMA_PASSWORD%/Parameter>
      <Parameter TYPE="CmdLine">%DATABASE_NAME%/Parameter>
    </Parameters>
  </Action>
  <Action TYPE="JDBCsqlScript" PERCENT_PROGRESS="40">
    <ValidIf DBTYPE="IBMDB2" />
    <Command TYPE="SCRIPT">%SCRIPT_HOME%/opss/scripts/db2/opss_user.sql</
Command>
    <Parameters>
      <Parameter TYPE="CmdLine">%SCHEMA_USER%/Parameter>
      <Parameter TYPE="CmdLine">%DEFAULT_TABLESPACE%/Parameter>
      <Parameter TYPE="CmdLine">%SCHEMA_USER%/Parameter>
    </Parameters>
  </Action>

  <Action TYPE="JDBCsqlScript" PERCENT_PROGRESS="60">

```

```

    <ValidIf DBTYPE="ORACLE" />
    <Command TYPE="SCRIPT">%SCRIPT_HOME%/opss/scripts/opss_tables.sql</
Command>
    <Parameters>
        <Parameter TYPE="CmdLine">%SCHEMA_USER%</Parameter>
    </Parameters>
    <IgnorableErrors>
        <Error Type="ORA-00942">table or view does not exist</Error>
    </IgnorableErrors>
    </Action>
    <Action TYPE="JDBCScript" PERCENT_PROGRESS="60">
    <ValidIf DBTYPE="EBR" />
    <Command TYPE="SCRIPT">%SCRIPT_HOME%/opss/scripts/oracleEbr/
opss_tables.sql</Command>
    <Parameters>
        <Parameter TYPE="CmdLine">%SCHEMA_USER%</Parameter>
        <Parameter TYPE="CmdLine">%EDITION_NAME%</Parameter>
    </Parameters>
    <IgnorableErrors>
        <Error Type="ORA-00942">table or view does not exist</Error>
    </IgnorableErrors>
    </Action>

    <Action TYPE="JDBCScript" PERCENT_PROGRESS="60">
    <ValidIf DBTYPE="SQLSERVER" />
    <Command TYPE="SCRIPT">%SCRIPT_HOME%/opss/scripts/sqlserver/
opss_tables.sql</Command>
    <Parameters>
        <Parameter TYPE="CmdLine">%DATABASE_NAME%</Parameter>
    </Parameters>
    </Action>
    <Action TYPE="JDBCScript" PERCENT_PROGRESS="60">
    <ValidIf DBTYPE="IBMDB2" />
    <Command TYPE="SCRIPT">%SCRIPT_HOME%/opss/scripts/db2/opss_tables.sql</
Command>
    <Parameters>
        <Parameter TYPE="CmdLine">%SCHEMA_USER%</Parameter>
        <Parameter TYPE="CmdLine">%DEFAULT_TABLESPACE%</Parameter>
        <Parameter TYPE="CmdLine">%TEMPORARY_TABLESPACE%</Parameter>
    </Parameters>
    </Action>

    <Action TYPE="JDBCScript" PERCENT_PROGRESS="70">
    <ValidIf DBTYPE="ORACLE" />
    <Command TYPE="SCRIPT">%SCRIPT_HOME%/opss/scripts/opss_version.sql</
Command>
    <Parameters>
        <Parameter TYPE="CmdLine">%SCHEMA_USER%</Parameter>
    </Parameters>
    </Action>
    <Action TYPE="JDBCScript" PERCENT_PROGRESS="70">
    <ValidIf DBTYPE="EBR" />
    <Command TYPE="SCRIPT">%SCRIPT_HOME%/opss/scripts/oracleEbr/
opss_version.sql</Command>
    <Parameters>
        <Parameter TYPE="CmdLine">%SCHEMA_USER%</Parameter>
        <Parameter TYPE="CmdLine">%EDITION_NAME%</Parameter>
    </Parameters>
    </Action>

    <Action TYPE="JDBCScript" PERCENT_PROGRESS="70">

```

```
<ValidIf DBTYPE="SQLSERVER" />
<Command TYPE="SCRIPT">%SCRIPT_HOME%/opss/scripts/sqlserver/
opss_version.sql</Command>
<Parameters>
  <Parameter TYPE="CmdLine">%DATABASE_NAME%</Parameter>
</Parameters>
</Action>
<Action TYPE="JDBCScript" PERCENT_PROGRESS="70">
  <ValidIf DBTYPE="IBMDB2" />
  <Command TYPE="SCRIPT">%SCRIPT_HOME%/opss/scripts/db2/opss_version.sql</
Command>
  <Parameters>
    <Parameter TYPE="CmdLine">%SCHEMA_USER%</Parameter>
  </Parameters>
</Action>

<Action TYPE="JDBCScript" PERCENT_PROGRESS="80">
  <ValidIf DBTYPE="ORACLE" />
  <Command TYPE="SCRIPT">%SCRIPT_HOME%/opss/scripts/opss_gencatalog.sql</
Command>
  <Parameters>
    <Parameter TYPE="CmdLine">%SCHEMA_USER%</Parameter>
  </Parameters>
</Action>

<Action TYPE="JDBCScript" PERCENT_PROGRESS="80">
  <ValidIf DBTYPE="SQLSERVER" />
  <Command TYPE="SCRIPT">%SCRIPT_HOME%/opss/scripts/sqlserver/
opss_gencatalog.sql</Command>
  <Parameters>
    <Parameter TYPE="CmdLine">%DATABASE_NAME%</Parameter>
  </Parameters>
</Action>

<Action TYPE="JDBCScript" PERCENT_PROGRESS="80">
  <ValidIf DBTYPE="IBMDB2" />
  <Command TYPE="SCRIPT">%SCRIPT_HOME%/opss/scripts/db2/
opss_gencatalog.sql</Command>
  <Parameters>
    <Parameter TYPE="CmdLine">%SCHEMA_USER%</Parameter>
  </Parameters>
</Action>

<Action TYPE="JDBCScript" PERCENT_PROGRESS="80">
  <ValidIf DBTYPE="EBR" />
  <Command TYPE="SCRIPT">%SCRIPT_HOME%/opss/scripts/oracleEbr/
opss_gencatalog.sql</Command>
  <Parameters>
    <Parameter TYPE="CmdLine">%SCHEMA_USER%</Parameter>
    <Parameter TYPE="CmdLine">%EDITION_NAME%</Parameter>
  </Parameters>
</Action>
<Action TYPE="JDBCScript" CONNECT_AS_OWNER="TRUE"
PERCENT_PROGRESS="100">
  <ValidIf DBTYPE="ORACLE" />
  <Command TYPE="SCRIPT">%SCRIPT_HOME%/opss/scripts/upgd_mgmt_schema.sql</
Command>
  </Action>

<Action TYPE="Java">
  <Command
```

```
TYPE="METHOD">oracle.ias.version.SchemaVersionUtil:utilSetComponentValid</
Command>
    <Parameters>
        <Parameter TYPE="String">OPSS</Parameter>
    </Parameters>
</Action>
</ExecutionDescriptor>

<!-- Deleting the OPSS Schema and removing it from the Registry -->
<DeleteDescriptor>
    <Action TYPE="JDBC" PERCENT_PROGRESS="50">
        <ValidIf DBTYPE="ORACLE" />
        <Command TYPE="INLINE">DROP USER %SCHEMA_USER% CASCADE</Command>
    </Action>
    <Action TYPE="JDBC" PERCENT_PROGRESS="50">
        <ValidIf DBTYPE="EBR" />
        <Command TYPE="INLINE">DROP USER %SCHEMA_USER% CASCADE</Command>
    </Action>

    <Action TYPE="JDBCsqlScript" PERCENT_PROGRESS="20">
        <ValidIf DBTYPE="SQLSERVER" />
        <Command TYPE="SCRIPT">%SCRIPT_HOME%/opss/scripts/sqlserver/
opss_drop_table.sql</Command>
        <Parameters>
            <Parameter TYPE="CmdLine">%DATABASE_NAME%</Parameter>
        </Parameters>
    </Action>
    <Action TYPE="JDBCsqlScript" PERCENT_PROGRESS="50">
        <ValidIf DBTYPE="SQLSERVER" />
        <Command TYPE="SCRIPT">%SCRIPT_HOME%/opss/scripts/sqlserver/
opss_drop_user.sql</Command>
        <Parameters>
            <Parameter TYPE="CmdLine">%DATABASE_NAME%</Parameter>
            <Parameter TYPE="CmdLine">%SCHEMA_USER%</Parameter>
        </Parameters>
    </Action>

    <Action TYPE="Java" PERCENT_PROGRESS="50">
        <ValidIf DBTYPE="IBMDB2" />
        <Command
TYPE="METHOD">oracle.sysman.assistants.common.dbutil.jdbc.DB2DropUtil:dropCompRep
os</Command>
        <Parameters>
            <Parameter TYPE="Connection"></Parameter>
            <Parameter TYPE="String">%SCHEMA_USER%</Parameter>
        </Parameters>
    </Action>

    <Action TYPE="Java" PERCENT_PROGRESS="100">
        <Command
TYPE="METHOD">oracle.ias.version.SchemaVersionUtil:utilDropRegistryEntry</
Command>
        <Parameters>
            <Parameter TYPE="Connection"></Parameter>
            <Parameter TYPE="String">OPSS</Parameter>
            <Parameter TYPE="String">%PREFIX_NAME%</Parameter>
            <Parameter TYPE="String">OPSS</Parameter>
        </Parameters>
    </Action>
</DeleteDescriptor>
</RepositoryConfig>
```

Component List Configuration File

The Component List configuration file (`ComponentInfo.xml`) lists all the components, their respective configuration files and their default user and tablespace mappings.

This file also lists the high-level prerequisite checks and high level actions (like creating `schema_version_registry` table) to be done globally for all the components. Also, a list of global Ignorable or Fatal errors can be specified.

Note:

RCU uses dynamic discovery to detect the available components. As a result, all of the components might not be listed in the Component List configuration file.

This file can be found in the `ORACLE_HOME/oracle_common/rcu/config` directory on UNIX operating systems or in the `ORACLE_HOME\oracle_common\rcu\config` directory on Windows operating systems.

Below is a sample `ComponentInfo.xml` file:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- DOCTYPE ComponentInfo SYSTEM "dtds/ComponentInfo.dtd" -->
<!DOCTYPE ComponentInfo SYSTEM "ComponentInfo.dtd" [ ]>
<ComponentInfo VERSION="11.0.0.0" TYPE="AS_REPOSITORY"
RESOURCE_BUNDLE_PACKAGE="oracle.sysman.rcu.as.ASBundle">
  <Display NLS_ID="ASREP_ID">Oracle AS Repository Components</Display>
  <PrefixSettings USE_SCHEMA_PREFIX="TRUE" USE_TABLESPACE_PREFIX="TRUE">
    <DetectQuery>
      Select distinct mrc_name from schema_version_registry
    </DetectQuery>
    <DetectQuery TYPE="IBMDB2">
      Select distinct mrc_name from NULLID.schema_version_registry
    </DetectQuery>
    <DetectQuery TYPE="JAVADB">
      Select distinct mrc_name from ORACLEFMW.schema_version_registry
    </DetectQuery>
  </PrefixSettings>

  <!-- AS Common GROUP START -->
  <Component ID="AS_COMMON" IS_GROUPING_COMPONENT="TRUE">
    <Display NLS_ID="AS_COMMON_ID">AS Common Schemas</Display>
  </Component>

  <Component ID="SOA" IS_GROUPING_COMPONENT="TRUE">
    <Display NLS_ID="SOA">SOA Suite</Display>
  </Component>

  <Component ID="ODI_REPOSITORIES" IS_GROUPING_COMPONENT="TRUE">
    <Display NLS_ID="ODI_REPOSITORIES">Oracle Data Integrator</Display>
  </Component>
```

```

<ComponentConfigFiles>
  <ConfigFile ID="MDS">%RCU_HOME%/../oracle_common/common/sql/mds/
mds_ComponentInfo.xml</ConfigFile>
  <ConfigFile ID="IAU">%RCU_HOME%/../oracle_common/common/sql/iau/
iau_ComponentInfo.xml</ConfigFile>
  <ConfigFile ID="OPSS">%RCU_HOME%/../oracle_common/common/sql/opss/
opss_ComponentInfo.xml</ConfigFile>
  <ConfigFile ID="UCSMESSAGING">%RCU_HOME%/../oracle_common/common/sql/
ucs.messaging/ucs.messaging_ComponentInfo.xml</ConfigFile>
  <ConfigFile ID="WLS">%RCU_HOME%/../oracle_common/common/sql/wlsservices/
wlsservices_ComponentInfo.xml</ConfigFile>
  <ConfigFile ID="ESS">%RCU_HOME%/../oracle_common/common/sql/ess/
ess_ComponentInfo.xml</ConfigFile>
  <ConfigFile ID="SOAINFRA">%RCU_HOME%/../soa/common/sql/soainfra/
soainfra_ComponentInfo.xml</ConfigFile>
  <ConfigFile ID="ODI">%RCU_HOME%/../odi/common/sql/odi/odi_ComponentInfo.xml</
ConfigFile>
<ConfigFile ID="OER">%RCU_HOME%/../oer/common/sql/oer/oer_ComponentInfo.xml</
ConfigFile>
<ConfigFile ID="OWLCS">%RCU_HOME%/../oracle_common/common/sql/ucs.callcontrol/
ucs.callcontrol_ComponentInfo.xml</ConfigFile>
<ConfigFile ID="SVCTBL">%RCU_HOME%/../oracle_common/common/sql/svctbl/
svctbl_ComponentInfo.xml</ConfigFile>
<ConfigFile ID="MFT">%RCU_HOME%/../mft/common/sql/mft/mft_ComponentInfo.xml</
ConfigFile>
</ComponentConfigFiles>

<PrerequisiteDescriptor>
  <DBPrerequisiteSet OPERATOR="OR">
    <ValidIf DBTYPE="ORACLE" />
    <DBPrerequisite PREREQ_TYPE="InitParameter" DATA_TYPE="NUMBER"
COMPARE_OPERATOR="GE">
      <PrereqIdentifier>SHARED_POOL_SIZE</PrereqIdentifier>
      <PrereqValue UNIT="KB">147456</PrereqValue>
    </DBPrerequisite>
    <DBPrerequisite PREREQ_TYPE="InitParameter" DATA_TYPE="NUMBER"
COMPARE_OPERATOR="GE">
      <PrereqIdentifier>SGA_MAX_SIZE</PrereqIdentifier>
      <PrereqValue UNIT="KB">147456</PrereqValue>
    </DBPrerequisite>
  </DBPrerequisiteSet>

  <DBPrerequisiteSet OPERATOR="AND">
    <ValidIf DBTYPE="ORACLE" />
    <DBPrerequisite PREREQ_TYPE="CustomSQL" DATA_TYPE="NUMBER"
COMPARE_OPERATOR="EQ" SOFT="TRUE">
      <ValidIf DBTYPE="ORACLE">
        <CustomQueryFilter DATA_TYPE="NUMBER" COMPARE_OPERATOR="EQ" VALUE="1">
          select 1 from dual where exists (select column_name from dba_tab_columns
where table_name(+) like 'V_$INSTANCE' and column_name(+) = 'EDITION') union
select 0 from dual where not exists (select column_name from dba_tab_columns
where table_name(+) like 'V_$INSTANCE' and column_name(+) = 'EDITION')
        </CustomQueryFilter>
      </ValidIf>
      <PrereqIdentifier>select count (*) from V_$INSTANCE where EDITION =
'XE'</PrereqIdentifier>
      <PrereqValue>0</PrereqValue>
      <PrereqErrorMsg>
        The database you are connecting is not a
supported version. Refer to the certification matrix for supported DB versions.
      </PrereqErrorMsg>
    </DBPrerequisite>
  </DBPrerequisiteSet>

```

```

</DBPrerequisite>

<DBPrerequisite PREREQ_TYPE="CustomSQL" DATA_TYPE="NUMBER"
COMPARE_OPERATOR="EQ">
  <ValidIf DBTYPE="ORACLE">
    <CustomQueryFilter DATA_TYPE="NUMBER" COMPARE_OPERATOR="EQ" VALUE="1">
      select 1 from dual where exists (select column_name from dba_tab_columns
where table_name(+) like 'V_$INSTANCE' and column_name(+) = 'EDITION') union
select 0 from dual where not exists (select column_name from dba_tab_columns
where table_name(+) like 'V_$INSTANCE' and column_name(+) = 'EDITION')
    </CustomQueryFilter>
  </ValidIf>
  <PrereqIdentifier>select count(*) from product_component_version
where product like 'Oracle%Database%' AND version >= '12.0.0.0' AND version
&lt; '12.1.0.1.0'</PrereqIdentifier>
  <PrereqValue>0</PrereqValue>
  <PrereqErrorMsg>
    The database you are connecting is not a
supported version. Enter Database with version equal to or higher than
12.1.0.1.0 in 12c. Refer to the certification matrix for supported DB versions.
  </PrereqErrorMsg>
</DBPrerequisite>

<DBPrerequisite PREREQ_TYPE="CustomSQL" DATA_TYPE="NUMBER"
COMPARE_OPERATOR="EQ" SOFT="TRUE">
  <ValidIf DBTYPE="ORACLE">
    <CustomQueryFilter DATA_TYPE="NUMBER" COMPARE_OPERATOR="EQ" VALUE="1">
      select 1 from dual where exists (select column_name from dba_tab_columns
where table_name(+) like 'V_$INSTANCE' and column_name(+) = 'EDITION') union
select 0 from dual where not exists (select column_name from dba_tab_columns
where table_name(+) like 'V_$INSTANCE' and column_name(+) = 'EDITION')
    </CustomQueryFilter>
  </ValidIf>
  <PrereqIdentifier>select count(*) from product_component_version where
product like 'Oracle%Database%' AND version > '12.1.0.1.9'</PrereqIdentifier>
  <PrereqValue>0</PrereqValue>
  <PrereqErrorMsg>
    The database you are connecting to, is a more recent one than the
supported version. Refer to the certification matrix for supported DB versions.
  </PrereqErrorMsg>
</DBPrerequisite>

</DBPrerequisiteSet>

<DBPrerequisite PREREQ_TYPE="InitParameter" DATA_TYPE="NUMBER"
COMPARE_OPERATOR="GE">
  <ValidIf DBTYPE="ORACLE" />
  <PrereqIdentifier>DB_BLOCK_SIZE</PrereqIdentifier>
  <PrereqValue UNIT="KB">8</PrereqValue>
</DBPrerequisite>
<!--DBPrerequisite PREREQ_TYPE="CustomSQL" DATA_TYPE="STRING"
COMPARE_OPERATOR="NE">
  <ValidIf DBTYPE="ORACLE" >
    <CustomQueryFilter DATA_TYPE="NUMBER" COMPARE_OPERATOR="EQ" VALUE="0">
      select 1 from dual where exists (select column_name from
dba_tab_columns where table_name(+) like 'V_$INSTANCE' and column_name(+) =
'EDITION') union select 0 from dual where not exists (select column_name from
dba_tab_columns where table_name(+) like 'V_$INSTANCE' and column_name(+) =
'EDITION')
    </CustomQueryFilter>
  </ValidIf>
</DBPrerequisite>

```



```

        </CustomQueryFilter>
    </ValidIf>
    <PrereqIdentifier>version</PrereqIdentifier>
    <PrereqValue>11.1.0.6.0</PrereqValue>
    <PrereqErrorMsg>
        The database you are connecting is 11.1.0.6.0
        version. 11.1.0.6.0 is not a supported version. The database version should be
        11.1.0.7.0 or greater.
    </PrereqErrorMsg>
    </DBPrerequisite-->
    <DBPrerequisite PREREQ_TYPE="DBVersion" DATA_TYPE="STRING"
    COMPARE_OPERATOR="GE">
    <ValidIf DBTYPE="ORACLE" >
    <CustomQueryFilter DATA_TYPE="NUMBER" COMPARE_OPERATOR="EQ" VALUE="0">
        select 1 from dual where exists (select column_name from
        dba_tab_columns where table_name(+) like 'V_$INSTANCE' and column_name(+) =
        'EDITION') union select 0 from dual where not exists (select column_name from
        dba_tab_columns where table_name(+) like 'V_$INSTANCE' and column_name(+) =
        'EDITION')
    </CustomQueryFilter>
    </ValidIf>
    <PrereqIdentifier>version</PrereqIdentifier>
    <PrereqValue>10.2.0.4.0</PrereqValue>
    <PrereqErrorMsg>
        The database you are connecting is not a supported version. Enter Database
        with version equal to or higher than 10.2.0.4.0 in 10g or version equal to or
        higher than 11.1.0.7.0 in 11g. Refer to the certification matrix for supported
        DB versions
    </PrereqErrorMsg>
    </DBPrerequisite>

    <DBPrerequisite PREREQ_TYPE="CustomSQL" DATA_TYPE="NUMBER" COMPARE_OPERATOR="EQ">
    <ValidIf DBTYPE="ORACLE" >
    <CustomQueryFilter DATA_TYPE="NUMBER" COMPARE_OPERATOR="EQ" VALUE="0">
        select 1 from dual where exists (select column_name from
        dba_tab_columns where table_name(+) like 'V_$INSTANCE' and column_name(+) =
        'EDITION') union select 0 from dual where not exists (select column_name from
        dba_tab_columns where table_name(+) like 'V_$INSTANCE' and column_name(+) =
        'EDITION')
    </CustomQueryFilter>
    </ValidIf>
    <PrereqIdentifier>select count(*) from product_component_version where
    product like 'Oracle%Database%' AND version BETWEEN '11' AND '11.1.0.6.0' </
    PrereqIdentifier>
    <PrereqValue>0</PrereqValue>
    <PrereqErrorMsg>
        The database you are connecting is not a
        supported version. Enter Database with version equal to or higher than
        10.2.0.4.0 in 10g or version equal to or higher than 11.1.0.7.0 in 11g. Refer to
        the certification matrix for supported DB versions
    </PrereqErrorMsg>
    </DBPrerequisite>

    <DBPrerequisite PREREQ_TYPE="CustomSQL" DATA_TYPE="NUMBER" COMPARE_OPERATOR="EQ">
    <ValidIf DBTYPE="IBMDB2"/>
    <PrereqIdentifier>select count(*) from syscat.tablespace where
    tbspace = 'SYSCATSPACE' and pagesize >= 32768
    </PrereqIdentifier>
    <PrereqValue>1</PrereqValue>

```

```
<PrereqErrorMsg>
    Component      : RCU
    Error          : Database prerequisite check failed.
    Cause         : Database: '%DATABASE_NAME%' needs to be
configured with
                    default pagesize 32768 or 32K.
                    Action      : Modify the default of the current
database or create a new
                    database with the required default pagesize.
</PrereqErrorMsg>
</DBPrerequisite>

<DBPrerequisite PREREQ_TYPE="CustomSQL" DATA_TYPE="STRING" COMPARE_OPERATOR="EQ">
    <ValidIf DBTYPE="SQLSERVER" />
    <PrereqIdentifier>SELECT count(*) where
CAST(SERVERPROPERTY('productversion') as VARCHAR(20)) like '1.%' or
CAST(SERVERPROPERTY('productversion') as VARCHAR(20)) like '4.%' or
CAST(SERVERPROPERTY('productversion') as VARCHAR(20)) like '6.%' or
CAST(SERVERPROPERTY('productversion') as VARCHAR(20)) like '7.%' or
CAST(SERVERPROPERTY('productversion') as VARCHAR(20)) like '8.%'</
PrereqIdentifier>
    <PrereqValue>0</PrereqValue>
    <PrereqErrorMsg>
        The database you are connecting is not a supported version.
Enter Database with version equal to or higher than 2005. Refer to the
certification matrix for supported DB versions.
    </PrereqErrorMsg>
</DBPrerequisite>

<DBPrerequisite PREREQ_TYPE="CustomSQL" DATA_TYPE="STRING"
COMPARE_OPERATOR="EQ">
    <ValidIf DBTYPE="IBMDB2" />
    <PrereqIdentifier>select count(*) FROM TABLE
(sysproc.env_get_inst_info()) where INT(substr(service_level,
POSSTR(service_level, 'v')+1, LOCATE('.', service_level,POSSTR(service_level,
'v') +1) - POSSTR(service_level, 'v') -1 )) = 9 AND INT(substr(service_level,
POSSTR(service_level, '.')+1, LOCATE('.',service_level,POSSTR(service_level,
'.' ) +1) - POSSTR(service_level, '.') -1 )) &lt; 7 OR INT(substr(service_level,
POSSTR(service_level, 'v')+1, LOCATE('.', service_level,POSSTR(service_level,
'v') +1) - POSSTR(service_level, 'v') -1 )) &lt; 9</PrereqIdentifier>
    <PrereqValue>0</PrereqValue>
    <PrereqErrorMsg>
        The database you are connecting is not a supported
version. Enter Database with version equal to or higher than 9.7. Refer to the
certification matrix for supported DB versions.
    </PrereqErrorMsg>
</DBPrerequisite>

<DBPrerequisite PREREQ_TYPE="CustomSQL" DATA_TYPE="STRING"
COMPARE_OPERATOR="EQ">
    <ValidIf DBTYPE="ORACLE" />
    <PrereqIdentifier>select GRANTED_ROLE from DBA_ROLE_PRIVS
where((GRANTED_ROLE='DBA' and GRANTEE=(select user from dual) and
lower(SYS_CONTEXT ('USERENV', 'SESSION_USER'))='sys') OR(GRANTED_ROLE='DBA' and
GRANTEE=(select user from dual)))</PrereqIdentifier>
    <PrereqValue>DBA</PrereqValue>
    <PrereqErrorMsg>
        User should have sysdba or dba privilages.
    </PrereqErrorMsg>
</DBPrerequisite>
```

```
<DBPrerequisite PREREQ_TYPE="CustomSQL" DATA_TYPE="STRING" COMPARE_OPERATOR="EQ">
  <ValidIf DBTYPE="SQLSERVER" />
  <PrereqIdentifier>select Is_Member('db_owner')</
PrereqIdentifier>
  <PrereqValue>1</PrereqValue>
  <PrereqErrorMsg>
    User should have sysdba or dba privileges.
  </PrereqErrorMsg>
</DBPrerequisite>

  <DBPrerequisite PREREQ_TYPE="CustomSQL" DATA_TYPE="STRING"
COMPARE_OPERATOR="EQ" SOFT="TRUE">
  <ValidIf DBTYPE="ORACLE" />
  <PrereqIdentifier>select value from nls_database_parameters
where parameter = 'NLS_CHARACTERSET'</PrereqIdentifier>
  <PrereqValue>AL32UTF8</PrereqValue>
  <PrereqErrorMsg>
    The database you are connecting is with non-
AL32UTF8 character set. Oracle strongly recommends using AL32UTF8 as the
database character set.
  </PrereqErrorMsg>
</DBPrerequisite>

  <DBPrerequisite PREREQ_TYPE="CustomSQL" DATA_TYPE="NUMBER"
COMPARE_OPERATOR="EQ" SOFT="TRUE">
  <ValidIf DBTYPE="ORACLE" >
  <CustomQueryFilter DATA_TYPE="NUMBER" COMPARE_OPERATOR="EQ" VALUE="0">
    select 1 from dual where exists (select column_name from
dba_tab_columns where table_name(+) like 'V_$INSTANCE' and column_name(+) =
'EDITION') union select 0 from dual where not exists (select column_name from
dba_tab_columns where table_name(+) like 'V_$INSTANCE' and column_name(+) =
'EDITION')
  </CustomQueryFilter>
  </ValidIf>
  <PrereqIdentifier>select count(*) from product_component_version where
product like 'Oracle%Database%' AND version BETWEEN '10.2.0.6.0' AND
'10.9.9.9.9' </PrereqIdentifier>
  <PrereqValue>0</PrereqValue>
  <PrereqErrorMsg>
    The database you are connecting to, is a more recent than the
supported version. Refer to the certification matrix for supported DB versions.
  </PrereqErrorMsg>
</DBPrerequisite>

  <DBPrerequisite PREREQ_TYPE="CustomSQL" DATA_TYPE="NUMBER"
COMPARE_OPERATOR="EQ" SOFT="TRUE">
  <ValidIf DBTYPE="ORACLE" >
  <CustomQueryFilter DATA_TYPE="NUMBER" COMPARE_OPERATOR="EQ" VALUE="0">
    select 1 from dual where exists (select column_name from
dba_tab_columns where table_name(+) like 'V_$INSTANCE' and column_name(+) =
'EDITION') union select 0 from dual where not exists (select column_name from
dba_tab_columns where table_name(+) like 'V_$INSTANCE' and column_name(+) =
'EDITION')
  </CustomQueryFilter>
  </ValidIf>
  <PrereqIdentifier>select count(*) from product_component_version where
product like 'Oracle%Database%' AND version > '11.1.0.7.0' AND version <=
'11.1.9.9.9'</PrereqIdentifier>
  <PrereqValue>0</PrereqValue>
```

```

        <PrereqErrorMsg>
            The database you are connecting to, is a more recent one than the
            supported version. Refer to the certification matrix for supported DB versions.
        </PrereqErrorMsg>
    </DBPrerequisite>

    <DBPrerequisite PREREQ_TYPE="CustomSQL" DATA_TYPE="NUMBER"
COMPARE_OPERATOR="EQ" SOFT="TRUE">
    <ValidIf DBTYPE="ORACLE" >
    <CustomQueryFilter DATA_TYPE="NUMBER" COMPARE_OPERATOR="EQ" VALUE="0">
        select 1 from dual where exists (select column_name from
dba_tab_columns where table_name(+) like 'V_$INSTANCE' and column_name(+) =
'EDITION') union select 0 from dual where not exists (select column_name from
dba_tab_columns where table_name(+) like 'V_$INSTANCE' and column_name(+) =
'EDITION')
    </CustomQueryFilter>
    </ValidIf>
    <PrereqIdentifier>select count(*) from product_component_version where
product like 'Oracle%Database%' AND version between '11.2.0.4.0' AND
'11.2.9.9.9' </PrereqIdentifier>
    <PrereqValue>0</PrereqValue>
    <PrereqErrorMsg>
        The database you are connecting to, is a more recent one than the
        supported version. Refer to the certification matrix for supported DB versions.
    </PrereqErrorMsg>
    </DBPrerequisite>

<DBPrerequisite PREREQ_TYPE="CustomSQL" DATA_TYPE="NUMBER" COMPARE_OPERATOR="EQ">
    <ValidIf DBTYPE="ORACLE" >
    <CustomQueryFilter DATA_TYPE="NUMBER" COMPARE_OPERATOR="EQ" VALUE="0">
        select 1 from dual where exists (select column_name from
dba_tab_columns where table_name(+) like 'V_$INSTANCE' and column_name(+) =
'EDITION') union select 0 from dual where not exists (select column_name from
dba_tab_columns where table_name(+) like 'V_$INSTANCE' and column_name(+) =
'EDITION')
    </CustomQueryFilter>
    </ValidIf>
    <PrereqIdentifier>select count(*) from
product_component_version where product like 'Oracle%Database%' AND version
between '11.2.0.0.0' AND '11.2.0.2.9' </PrereqIdentifier>
    <PrereqValue>0</PrereqValue>
    <PrereqErrorMsg>
        The database you are connecting is not a
        supported version. Enter Database with version equal to or higher than
        11.2.0.3.0 in 11g. Refer to the certification matrix for supported DB versions.
    </PrereqErrorMsg>
    </DBPrerequisite>
<!--
<DBPrerequisite PREREQ_TYPE="CustomSQL" DATA_TYPE="NUMBER" COMPARE_OPERATOR="EQ"
SOFT="TRUE">
    <ValidIf DBTYPE="ORACLE" >
    <CustomQueryFilter DATA_TYPE="NUMBER" COMPARE_OPERATOR="EQ" VALUE="0">
        select 1 from dual where exists (select column_name from
dba_tab_columns where table_name(+) like 'V_$INSTANCE' and column_name(+) =
'EDITION') union select 0 from dual where not exists (select column_name from
dba_tab_columns where table_name(+) like 'V_$INSTANCE' and column_name(+) =
'EDITION')
    </CustomQueryFilter>
    </ValidIf>
    <PrereqIdentifier>select count(*) from
product_component_version where product like 'Oracle%Database%' AND version >=

```

```
'12.2.0.0.0' </PrereqIdentifier>
    <PrereqValue>0</PrereqValue>
    <PrereqErrorMsg>
        The database you are connecting to, is a more recent one than the
supported version. Refer to the certification matrix for supported DB versions.
    </PrereqErrorMsg>
</DBPrerequisite>

<DBPrerequisite PREREQ_TYPE="CustomSQL" DATA_TYPE="NUMBER" COMPARE_OPERATOR="EQ">
    <ValidIf DBTYPE="ORACLE" >
        <CustomQueryFilter DATA_TYPE="NUMBER" COMPARE_OPERATOR="EQ" VALUE="0">
            select 1 from dual where exists (select column_name from
dba_tab_columns where table_name(+) like 'V_$INSTANCE' and column_name(+) =
'EDITION') union select 0 from dual where not exists (select column_name from
dba_tab_columns where table_name(+) like 'V_$INSTANCE' and column_name(+) =
'EDITION')
        </CustomQueryFilter>
    </ValidIf>
    <PrereqIdentifier>select count(*) from product_component_version where
product like 'Oracle%Database%' AND version like '12.0.0.0.0' </
PrereqIdentifier>
    <PrereqValue>0</PrereqValue>
    <PrereqErrorMsg>
        The database you are connecting is not a
supported version. Enter Database with version equal to or higher than
12.1.0.0.0 in 12c. Refer to the certification matrix for supported DB versions.
    </PrereqErrorMsg>
</DBPrerequisite>
-->

<DBPrerequisite PREREQ_TYPE="CustomSQL" DATA_TYPE="STRING"
COMPARE_OPERATOR="EQ" SOFT="TRUE">
    <ValidIf DBTYPE="SQLSERVER" />
    <PrereqIdentifier>SELECT count(*) where
CAST(SERVERPROPERTY('productversion') as VARCHAR(20)) not like '1.%' and
CAST(SERVERPROPERTY('productversion') as VARCHAR(20)) not like '4.%' and
CAST(SERVERPROPERTY('productversion') as VARCHAR(20)) not like '6.%' and
CAST(SERVERPROPERTY('productversion') as VARCHAR(20)) not like '7.%' and
CAST(SERVERPROPERTY('productversion') as VARCHAR(20)) not like '8.%' and
CAST(SERVERPROPERTY('productversion') as VARCHAR(20)) not like '9.%' and
CAST(SERVERPROPERTY('productversion') as VARCHAR(20)) not like '10.%' </
PrereqIdentifier>
    <PrereqValue>0</PrereqValue>
    <PrereqErrorMsg>
        The database you are connecting to, is a more recent than
the supported version. Refer to the certification matrix for supported DB
versions.
    </PrereqErrorMsg>
</DBPrerequisite>

<DBPrerequisite PREREQ_TYPE="CustomSQL" DATA_TYPE="STRING"
COMPARE_OPERATOR="EQ" SOFT="TRUE">
    <ValidIf DBTYPE="IBMDB2" />
    <PrereqIdentifier>select count(*) FROM TABLE (sysproc.env_get_inst_info())
where INT(substr(service_level, POSSTR(service_level, 'v')+1, LOCATE('.',
service_level, POSSTR(service_level, 'v') +1) - POSSTR(service_level, 'v')
-1 ) ) = 9 and INT(substr(service_level, POSSTR(service_level, '.')+1,
LOCATE('.', service_level, POSSTR(service_level, '.') +1) -
POSSTR(service_level, '.') -1 ) ) > 7 or INT(substr(service_level,
POSSTR(service_level, 'v')+1, LOCATE('.', service_level, POSSTR(service_level,
'v') +1) - POSSTR(service_level, 'v') -1 ) ) > 9</PrereqIdentifier>
```

```
<PrereqValue>0</PrereqValue>
  <PrereqErrorMsg>
    The database you are connecting to, is a more recent
    than the supported version. Refer to the certification matrix for supported DB
    versions.
  </PrereqErrorMsg>
</DBPrerequisite>

<DBPrerequisite PREREQ_TYPE="Java" DATA_TYPE="STRING" COMPARE_OPERATOR="NE"
SOFT="TRUE">
  <ValidIf DBTYPE="JAVADB" />
  <PrereqIdentifier>%RCU_HOME%/../oracle_common/rcu/config/
JavaDB.jar:javadbproject.JavaDB:checkJavaDBVersion</PrereqIdentifier>
  <PrereqValue>SOFTPASS</PrereqValue>
  <PrereqErrorMsg>
    The database you are connecting to, is a more recent than the
    supported version. Refer to the certification matrix for supported DB versions.
  </PrereqErrorMsg>
</DBPrerequisite>

<DBPrerequisite PREREQ_TYPE="Java" DATA_TYPE="STRING" COMPARE_OPERATOR="NE">
  <ValidIf DBTYPE="JAVADB" />
  <PrereqIdentifier>%RCU_HOME%/../oracle_common/rcu/config/
JavaDB.jar:javadbproject.JavaDB:checkJavaDBVersion</PrereqIdentifier>
  <PrereqValue>FAIL</PrereqValue>
  <PrereqErrorMsg>
    The database you are connecting is not a supported version. Enter
    Database with version equal to or higher than 10.5.3.0. Refer to the
    certification matrix for supported DB versions.
  </PrereqErrorMsg>
</DBPrerequisite>

  <DBPrerequisite PREREQ_TYPE="CustomSQL" DATA_TYPE="NUMBER"
COMPARE_OPERATOR="EQ">
    <ValidIf DBTYPE="MYSQL" />
    <PrereqIdentifier>select version() &lt; '5.5.14'</
PrereqIdentifier>
    <PrereqValue>0</PrereqValue>
    <PrereqErrorMsg>
      The database you are connecting is not a supported
      version. Enter Database with version equal to or higher than 5.5.14. Refer to
      the certification matrix for supported DB versions.
    </PrereqErrorMsg>
  </DBPrerequisite>

  <DBPrerequisite PREREQ_TYPE="CustomSQL" DATA_TYPE="NUMBER"
COMPARE_OPERATOR="EQ" SOFT="TRUE">
    <ValidIf DBTYPE="MYSQL" />
    <PrereqIdentifier>select version() &gt; '5.6' or version() =
'5.6'</PrereqIdentifier>
    <PrereqValue>0</PrereqValue>
    <PrereqErrorMsg>
      The database you are connecting to, is a more recent
      than the supported version. Refer to the certification matrix for supported DB
      versions.
    </PrereqErrorMsg>
  </DBPrerequisite>
```

```

<DBPrerequisite PREREQ_TYPE="CustomSQL" DATA_TYPE="STRING" COMPARE_OPERATOR="EQ">
  <ValidIf DBTYPE="MYSQL" />
  <PrereqIdentifier>select VARIABLE_VALUE from
INFORMATION_SCHEMA.GLOBAL_VARIABLES where VARIABLE_NAME =
'INNODB_FILE_PER_TABLE'</PrereqIdentifier>
  <PrereqValue>ON</PrereqValue>
  <PrereqErrorMsg>
    DB Init Param Prerequisite failure for
INNODB_FILE_PER_TABLE. Its value should be 'ON'.
  </PrereqErrorMsg>
</DBPrerequisite>

<DBPrerequisite PREREQ_TYPE="CustomSQL" DATA_TYPE="STRING" COMPARE_OPERATOR="EQ">
  <ValidIf DBTYPE="MYSQL" />
  <PrereqIdentifier>select VARIABLE_VALUE from
INFORMATION_SCHEMA.GLOBAL_VARIABLES where VARIABLE_NAME = 'INNODB_FILE_FORMAT'</
PrereqIdentifier>
  <PrereqValue>Barracuda</PrereqValue>
  <PrereqErrorMsg>
    DB Init Param Prerequisite failure for INNODB_FILE_FORMAT.
Its value should be 'Barracuda'.
  </PrereqErrorMsg>
</DBPrerequisite>

<DBPrerequisite PREREQ_TYPE="CustomSQL" DATA_TYPE="STRING" COMPARE_OPERATOR="EQ">
  <ValidIf DBTYPE="MYSQL" />
  <PrereqIdentifier>select VARIABLE_VALUE from
INFORMATION_SCHEMA.GLOBAL_VARIABLES where VARIABLE_NAME = 'INNODB_LARGE_PREFIX'</
PrereqIdentifier>
  <PrereqValue>ON</PrereqValue>
  <PrereqErrorMsg>
    DB Init Param Prerequisite failure for
INNODB_LARGE_PREFIX. Its value should be 'ON'.
  </PrereqErrorMsg>
</DBPrerequisite>

<DBPrerequisite PREREQ_TYPE="CustomSQL" DATA_TYPE="STRING" COMPARE_OPERATOR="EQ">
  <ValidIf DBTYPE="MYSQL" />
  <PrereqIdentifier>select VARIABLE_VALUE from
INFORMATION_SCHEMA.GLOBAL_VARIABLES where VARIABLE_NAME
='log_bin_trust_function_creators'</PrereqIdentifier>
  <PrereqValue>ON</PrereqValue>
  <PrereqErrorMsg>
    DB Init Param Prerequisite failure for
log_bin_trust_function_creators. Its value should be 'ON'.
  </PrereqErrorMsg>
</DBPrerequisite>

</PrerequisiteDescriptor>

  <ExecutionDescriptor TYPE="PreLoad">
    <Action TYPE="Java" PERCENT_PROGRESS="60">
      <ValidIf DBTYPE="ORACLE,EBR">
        <CustomQueryFilter DATA_TYPE="NUMBER" COMPARE_OPERATOR="EQ"
VALUE="0">
          select count(*) from dba_views where VIEW_NAME=
'SCHEMA_VERSION_REGISTRY'
        </CustomQueryFilter>
      </ValidIf>
    <Command
TYPE="METHOD">oracle.ias.version.SchemaVersionUtil:utilCreateRegistryTable</

```

```

Command>
    <Parameters>
        <Parameter TYPE="Connection"></Parameter>
    </Parameters>
</Action>
    <Action TYPE="Java" PERCENT_PROGRESS="60">
    <ValidIf DBTYPE="SQLSERVER">
        <CustomQueryFilter DATA_TYPE="NUMBER" COMPARE_OPERATOR="EQ"
VALUE="0">
            select count(*) from INFORMATION_SCHEMA.TABLES where
TABLE_NAME='SCHEMA_VERSION_REGISTRY'
        </CustomQueryFilter>
    </ValidIf>
    <Command
TYPE="METHOD">oracle.ias.version.SchemaVersionUtil:utilCreateRegistryTable</
Command>
    <Parameters>
        <Parameter TYPE="Connection"></Parameter>
    </Parameters>
</Action>
    <Action TYPE="Java" PERCENT_PROGRESS="60">
        <ValidIf DBTYPE="IBMDB2">
            <CustomQueryFilter DATA_TYPE="NUMBER" COMPARE_OPERATOR="EQ"
VALUE="0">
                select count(*) from syscat.tables where
TABNAME='SCHEMA_VERSION_REGISTRY'
            </CustomQueryFilter>
        </ValidIf>
        <Command
TYPE="METHOD">oracle.ias.version.SchemaVersionUtil:utilCreateRegistryTable</
Command>
    <Parameters>
        <Parameter TYPE="Connection"></Parameter>
    </Parameters>
</Action>
    <Action TYPE="Java" PERCENT_PROGRESS="60">
    <ValidIf DBTYPE="JAVADB">
        <CustomQueryFilter DATA_TYPE="NUMBER"
COMPARE_OPERATOR="EQ" VALUE="0">
            SELECT COUNT(*) FROM SYS.SYSTABLES INNER JOIN SYS.SYSSCHEMAS ON
SYS.SYSTABLES.SCHEMAID = SYS.SYSSCHEMAS.SCHEMAID WHERE SCHEMANAME='ORACLEFMW'
AND TABLENAME='SCHEMA_VERSION_REGISTRY_T'
        </CustomQueryFilter>
    </ValidIf>
    <Command
TYPE="METHOD">oracle.ias.version.SchemaVersionUtil:utilCreateRegistryTable</
Command>
    <Parameters>
        <Parameter TYPE="Connection"></Parameter>
    </Parameters>
</Action>

    <Action TYPE="Java" PERCENT_PROGRESS="60">
    <ValidIf DBTYPE="MYSQL">
        <CustomQueryFilter DATA_TYPE="NUMBER" COMPARE_OPERATOR="EQ"
VALUE="0">
            select count(*) from INFORMATION_SCHEMA.TABLES where
TABLE_NAME='SCHEMA_VERSION_REGISTRY'
        </CustomQueryFilter>
    </ValidIf>
    <Command

```



```
TYPE="METHOD">oracle.ias.version.SchemaVersionUtil:utilCreateRegistryTable</
Command>
  <Parameters>
    <Parameter TYPE="Connection"></Parameter>
  </Parameters>
</Action>

</ExecutionDescriptor>

<ExecutionDescriptor TYPE="PostLoad">
  <Action TYPE="Java" PERCENT_PROGRESS="60">
    <ValidIf DBTYPE="EBR"/>
    <Command
TYPE="METHOD">oracle.ias.version.SchemaVersionUtil:utilSetEditionName</Command>
      <Parameters>
        <Parameter TYPE="String">%EDITION_NAME%</Parameter>
      </Parameters>
    </Action>

    <Action TYPE="Java">
      <ValidIf DBTYPE="ORACLE,EBR"/>
      <Command
TYPE="METHOD">com.oracle.cie.rcu.ComponentSchemaUtil:loadCompSchemaInfo</Command>
        <Parameters>
          <Parameter TYPE="Connection"></Parameter>
          <Parameter TYPE="String">%SERVICE_TABLE_STRING%</Parameter>
          <Parameter TYPE="String">%PREFIX_NAME%</Parameter>
          <Parameter TYPE="String">%DB_HOSTNAME%</Parameter>
          <Parameter TYPE="String">%DB_SERVICE%</Parameter>
          <Parameter TYPE="String">%DB_PORTNUMBER%</Parameter>
        </Parameters>
      </Action>
      <Action TYPE="Java">
        <ValidIf DBTYPE="SQLSERVER,IBMDB2,MYSQL,JAVADB"/>
        <Command
TYPE="METHOD">com.oracle.cie.rcu.ComponentSchemaUtil:loadNonOracleCompSchemaInfo<
/Command>
      <Parameters>
        <Parameter TYPE="Connection"></Parameter>
        <Parameter TYPE="String">%SERVICE_TABLE_STRING%</Parameter>
        <Parameter TYPE="String">%PREFIX_NAME%</Parameter>
        <Parameter TYPE="String">%DB_HOSTNAME%</Parameter>
        <Parameter TYPE="String">%DB_PORTNUMBER%</Parameter>
        <Parameter TYPE="String">%DATABASE_NAME%</Parameter>
      </Parameters>
    </Action>
  </ExecutionDescriptor>

<FatalErrors>
  <Error Type="ORA-17439">Invalid SQL type</Error>
  <Error Type="ORA-01435">user does not exist</Error>
  <Error Type="ORA-01435">user does not exist</Error>
  <Error Type="ORA-00955">name is already used by an existing object</Error>
  <Error Type="ORA-01031">name is already used by an existing object</
Error>
</FatalErrors>

<IgnorableErrors>
  <Error Type="ORA-02289">sequence does not exist</Error>
  <Error Type="ORA-00904">invalid identifier</Error>
```

```

      <Error Type="ORA-01920">user name conflicts with another user or role name</
Error>
      <Error Type="ORA-01418">specified index does not exist</Error>
      <Error Type="ORA-00942">table or view does not exist</Error>
      <Error Type="ORA-06512">Not found</Error>
      <Error Type="ORA-01403">no data found</Error>
        <Error Type="ORA-04043">does not exist</Error>
        <Error Type="ORA-04080">Trigger does not exist</Error>
        <Error Type="ORA-00959">Tablespace does not exist</Error>
        <Error Type="ORA-24035">AQ agent not subscriber</Error>
        <Error Type="ORA-24185">Transformation not found</Error>
        <Error Type="ORA-24042">Does not exist</Error>
        <Error Type="ORA-24088">Does not exist</Error>
    </IgnorableErrors>
  </ComponentInfo>

```

Soft-Prerequisite Support

If a particular `<DBPrerequisiteSet>` or `<DBPrerequisite>` is not mandatory, then you can use the soft-prerequisite option by setting the `SOFT` attribute to `TRUE` in the `ComponentInfo.xml` file

When a soft-prerequisite is not met, a pop-up dialog window with an error or warning message appears; the user has the option to ignore the message or abort the operation. You can define a soft-prerequisite at the `<DBPrerequisiteSet>` level, the `<DBPrerequisite>` level, or both; if both are defined, then `<DBPrerequisiteSet>` takes higher precedence.

Below is an example of setting a soft-prerequisite at the `<DBPrerequisite>` level:

```

<DBPrerequisiteSet>
...
  <DBPrerequisite PREREQ_TYPE="CustomSQL" DATA_TYPE="STRING" COMPARE_OPERATOR="EQ"
SOFT="TRUE">
    <PrereqIdentifier>select value from nls_database_parameters where parameter =
'NLS_CHARACTERSET'</PrereqIdentifier>
    <PrereqValue>AL32UTF8</PrereqValue>
    <PrereqErrorMsg>
      The database you are connecting is with non-AL32UTF8 character set.
      Oracle strongly recommends using AL32UTF8 as the database character
      set.
    </PrereqErrorMsg>
  </DBPrerequisite>
...
</DBPrerequisiteSet>

```

Default Tablespaces Configuration File

The default tablespaces configuration file (`Storage.xml`) lists the components, for which, tablespaces are created out-of-the-box. This file is located in the `ORACLE_HOME/oracle_common/rcu/config` directory on UNIX operating systems or in the `ORACLE_HOME\oracle_common\rcu\config` directory on Windows operating systems.

The actual tablespace configuration file for each component is located in the `ORACLE_HOME/oracle_common/rcu/integrationcomponent/component_Storage.xml` directory on UNIX operating systems or in the `ORACLE_HOME\oracle_common\rcu\integrationcomponent\component_Storage.xml`

directory on Windows operating systems. Each component has its own tablespaces configuration file.

Below is a sample `opss_Storage.xml` file:

```
<?xml version="1.0" encoding="UTF-8" ?>

<!-- OPSS START -->
<TablespaceAttributes NAME="IAS_OPSS">
  <Type>Permanent</Type>
  <DefaultTemp>False</DefaultTemp>
  <Bigfile>False</Bigfile>
  <DatafilesList>
    <DatafileAttributes ID="%DATAFILE_LOCATION%/ias_opss.dbf">
      <Size UNIT="MB">60</Size>
      <Reuse>True</Reuse>
      <AutoExtend>True</AutoExtend>
    </DatafileAttributes>
  </DatafilesList>
</TablespaceAttributes>
<!-- OPSS END -->
```

RCU Script Writing Guidelines

Use these guidelines when writing RCU scripts.

Schema user names and passwords should not be hard coded. They should be coded as substitutable variables.

- If schema user needs to be created, it should be created first by using the parameters passed by RCU.
- Tablespace and temporary tablespace references should not be hard coded; they should be coded as variables.
- Do not use `CONNECT`; instead, use `ALTER SESSION SET CURRENT_SCHEMA = <SCHEMA_OWNER>` after you create the schema user.
- The set of ignorable and fatal `ORA` errors (if any) should be listed in the RCU XML component configuration file.
- Avoid any *shutdown* or *startup* that would bounce the database instance.
- `SCHEMA_VERSION_REGISTRY` should be updated before and after loading schema. This can be done by using `JavaAction` as shown in [Java Code Using JavaAction](#) or within the component scripts by using the `SCHEMA_VERSION PL/SQL` package.
- Block comments that contain line comments (`/* -- comment */`) are not supported.

The following are the guidelines for RCU for various scripts:

- [Guidelines for RCU JDBC Engine Compliant SQL*Plus Scripts](#)
Use these guidelines for writing the RCU JDBC Engine SQL*Plus scripts.
- [Guidelines for Pure JDBC Scripts](#)
Use these guidelines for writing Pure JDBC scripts for RCU.
- [Guidelines for SQL*Plus Scripts](#)
Use these guidelines for writing SQL*Plus scripts for RCU.

- [Guidelines for SQL Server-Based Scripts](#)
Use these guidelines for writing SQL Server-based scripts for RCU.

Guidelines for RCU JDBC Engine Compliant SQL*Plus Scripts

Use these guidelines for writing the RCU JDBC Engine SQL*Plus scripts.

- All statements must be terminated with appropriate terminating chars. *CREATE PACKAGE*, *TYPE* needs to be terminated with a ; (semicolon) with / (forward slash) on the next line. All other statements (Create TABLE, VIEW, and so on.) need to be terminated by either ; (semicolon) or / (forward slash).
- *EXECUTE* calls should be replaced with *BEGIN/END* blocks.
- *DEFINE* statements should be in one line, no comments in the same line and no ; (semicolon) at the end.
- *SET*, *SHOW*, *SPOOL*, *WHENEVER*, *BREAK*, *EXIT* statements are ignored.
- *HOST* command is not supported yet.
- *VARIABLE* and *COL(UMN)* are not supported.

Dynamically calling another SQL Script within a PL/SQL block using the following technique is not supported:

```
VARIABLE initfile VARCHAR2(32)
COLUMN :initfile NEW_VALUE init_file NOPRINT;
BEGIN
  IF (some condition) THEN
    :initfile := 'initcdc.sql';
  ELSE
    :initfile := 'nothing.sql';
  END IF;
END;
/
SELECT :initfile FROM DUAL;
@@&init_file
```

The work around is to have a separate Action with a *ValidIf* tag to specify the condition.

Guidelines for Pure JDBC Scripts

Use these guidelines for writing Pure JDBC scripts for RCU.

- Should not contain any SQL*Plus directives (like SET, WHENEVER, and so on.).
- All DEFINES should be changed to PL/SQL variable declarations.
- All SQL statements should be wrapped in *EXECUTE IMMEDIATE*.
- PL/SQL style comments are allowed, but SQL*Plus style (REM) comments are not allowed.
- DROP statements that precedes CREATE statements do not work. DROP should only be done after checking for the existence of the object. Ideally, all DROP statements should put into different PL/SQL script and RCU can call this script before calling a CREATE script, if that is desired.

- Contents of the script file should be a valid PL/SQL block, which can be called within `Connection.prepareCall()`.

Guidelines for SQL*Plus Scripts

Use these guidelines for writing SQL*Plus scripts for RCU.

- Should not have any *exit* statements or *WHENEVER ERROR EXIT* directives. This would cause RCU SQL*Plus session to exit unexpectedly and may impact other component scripts to be executed later.
- Scripts should not have any spool commands. RCU would generate a spool log for each component.

Guidelines for SQL Server-Based Scripts

Use these guidelines for writing SQL Server-based scripts for RCU.

- Support is a subset of what is supported in t-sql scripts that can be executed by `sqlcmd`.
- *ValidIf* tags should be added around all database-specific Actions and Prerequisites. For example:

```
<DBPrerequisite PREREQ_TYPE="TablespaceFreeMB" DATA_TYPE="NUMBER"
COMPARE_OPERATOR="GT">
  <ValidIf DBTYPE="ORACLE" />
  <PrereqIdentifier>%DEFAULT_TABLESPACE%</PrereqIdentifier>
  <PrereqValue>50</PrereqValue>
</DBPrerequisite>
```

- RCU supports recursive variable definitions such as:
- ```
setvar var1 value1
setvar var2 $(var1)
```
- There should be a *go* statement to end blocks of statements. All statements that precedes the *go* statement are executed as a single statement over JDBC.
  - The JDBC connection is created in the auto-commit *on* mode.
  - Currently, the begin transaction and commit transaction statements are not supported.
  - Variables passed to scripts through the XML file are passed as follows:

```
Script.sql -v v1=value1 v2=value2
```

This is only for scripts called by using the XML files. If a script calls another script, you can use any other variable name.

# D

## Troubleshooting Repository Creation Utility

Review this information to help you understand and troubleshoot any errors or issues that you encounter while you run the RCU.

- [General Troubleshooting Tips](#)  
Use these tips to help troubleshoot any errors that you encounter during installation.
- [RCU Log Files](#)  
When you run RCU to create or drop your schemas, the main RCU log file is written to the `/tmp/RCUdate_timestamp_random_number/logs` directory on UNIX operating systems or `C:\Users\user_name\AppData\Local\Temp\RCUdate_timestamp_random_number\logs` directory on Windows operating systems, by default.
- [RCU Fails to Connect with ORA-604 ORA-1882 Error Codes](#)  
Error codes ORA-604 and ORA-1882.
- [SOAINFRA Schema Creation Fails on a Hardened Database](#)  
Use this workaround if the RCU schemas for SOAINFRA does not load successfully on an Oracle hardened database.
- [Need More Help?](#)  
If this guide does not solve the problem that you encountered, try looking for a solution on My Oracle Support (formerly Oracle MetaLink).

### General Troubleshooting Tips

Use these tips to help troubleshoot any errors that you encounter during installation.

- See Install and Configure in the *Release Notes for Oracle Fusion Middleware Infrastructure* for the latest updates and issues related to Oracle Fusion Middleware product installation and configuration.
- Verify that your computer meets the requirements as specified in the *Oracle Fusion Middleware System Requirements and Specifications* document.  
Select the document that is applicable for your release.
- Verify that your environment meets the certification requirements for your release and platform, as specified on the Oracle Fusion Middleware Supported System Configurations page.
- Make sure that your database is up and running.
- If you entered incorrect information on one of the screens, use the navigation pane on the left most side of the graphical interface to return to that screen.
- If an error occurred while running RCU:
  1. Note the error and review the installation log files (see [RCU Log Files](#)).

2. Correct the issue that caused the error. Depending on the type of error, you may either continue with your RCU operation, or you are forced to restart RCU.
3. Continue or restart RCU to complete your desired operation.

## RCU Log Files

When you run RCU to create or drop your schemas, the main RCU log file is written to the `/tmp/RCUdate_timestamp_random_number/logs` directory on UNIX operating systems or `C:\Users\user_name\AppData\Local\Temp\RCUdate_timestamp_random_number\logs` directory on Windows operating systems, by default.

For example, on a UNIX operating system:

```
/tmp/RCU2014-01-02_03-00_412547075/logs/rcu.log
```

### Accessing RCU Log Files on Windows

On Windows operating systems, the `AppData` folder might be hidden. To be able to navigate to the RCU log files on Windows, ensure that the `AppData` folder is visible:

1. From the **Start** menu, select **Control Panel**, then **Appearance and Personalization**, and then **Folder Options**.  
The **Folder Options** window appears.
2. Select the **View** tab.
3. Under Advanced settings, ensure that **Show hidden files, folders, and drives** is selected and click **OK**.

### RCU Component Log Files

In addition to the general log file, each component writes a log file of its own. All component log files are also written to the same directory as the `rcu.log` file.

[Table D-1](#) lists the component log file names in alphabetical order by log file name.

**Table D-1 RCU Component Log File Names**

| Component                                     | Log File Name     |
|-----------------------------------------------|-------------------|
| Analytics                                     | activities.log    |
| Business Intelligence Platform                | biplatform.log    |
| Oracle WebCenter Enterprise Capture           | capture.log       |
| Oracle WebCenter Content Server — Complete    | content.log       |
| Oracle WebCenter Content Server — Search Only | contentsearch.log |
| Discussions                                   | discussions.log   |
| EDQ Config Repository                         | edq_conf.log      |
| EDQ Results Repository                        | edq_res.log       |
| Oracle Enterprise Scheduler                   | ess.log           |
| Audit Services                                | iau.log           |

**Table D-1 (Cont.) RCU Component Log File Names**

| Component                           | Log File Name  |
|-------------------------------------|----------------|
| Audit Services Append               | iau_append.log |
| Audit Services Viewer               | iau_viewer.log |
| Metadata Services                   | mds.log        |
| Managed File Transfer               | mft.log        |
| Master and Work Repository          | odi.log        |
| Monitor Server                      | oggmon.log     |
| Oracle GoldenGate Studio Repository | oggstudio.log  |
| Oracle Platform Security Services   | opss.log       |
| Portlet Producers                   | portlet.log    |
| SOA Infrastructure                  | soainfra.log   |
| Common Infrastructure Services      | stb.log        |
| User Messaging Service              | ucsums.log     |
| Veridata Repository                 | veridata.log   |
| WebCenter Sites                     | wcsites.log    |
| WebCenter Sites — Visitor Services  | wcsitesvs.log  |
| Portal and Services                 | webcenter.log  |
| WebLogic Services                   | wls.log        |

## RCU Fails to Connect with ORA-604 ORA-1882 Error Codes

Error codes ORA-604 and ORA-1882.

RCU fails to connect database with the following errors:

- ORA-604: error occurred at recursive SQL level 1
- ORA-1882: timezone region not found

### Setting RCU\_JAVA\_OPTIONS before Invoking RCU

Set the below property and invoke RCU:

```
RCU_JAVA_OPTIONS -Doracle.jdbc.timezoneAsRegion=false
```

## SOAINFRA Schema Creation Fails on a Hardened Database

Use this workaround if the RCU schemas for SOAINFRA does not load successfully on an Oracle hardened database.



In case of an Oracle hardened database, when the public grants are revoked for the PL/SQL packages, the schema creation for SOAINFRA fails with the following error:

```
Received Non-Ignorable Error: ORA-04063: package body "XDB.DBMS_CSX_INT"
has errors
```

Grant the following permission to the user and run the Repository Creation Utility (RCU) again to create the schemas for SOAINFRA.

```
GRANT EXECUTE ON DBMS_LOB TO XDB;
```

## Need More Help?

If this guide does not solve the problem that you encountered, try looking for a solution on My Oracle Support (formerly Oracle MetaLink).

<https://support.oracle.com/>

If you are unable to find a solution for your problem, open a service request.