Oracle® Fusion Middleware

Upgrading with the Upgrade Assistant

12*c* (12.2.1.1)

E71492-01

June 2016

This tool-specific reference guide provides command-line instructions and screen references for the Oracle Fusion Middleware Upgrade Assistant.

ORACLE®

Oracle Fusion Middleware Upgrading with the Upgrade Assistant, 12c (12.2.1.1)

E71492-01

Contents

Preface

This preface contains the following sections:

Audience

Conventions

Audience

This manual is intended for Oracle Fusion Middleware system administrators who are responsible for upgrading Oracle Fusion Middleware. It is assumed that the readers of this manual have knowledge of the following:

- Oracle Fusion Middleware 12*c* system administration and configuration information for the existing deployment

- The configuration and expected behavior of the system or systems being upgraded

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.
italic	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 Using the Upgrade Assistant

The Oracle Fusion Middleware Upgrade Assistant automates many upgrade tasks. This topic describes how to use the Upgrade Assistant.

You perform the upgrade tasks using the Upgrade Assistant. You can also use the Upgrade Assistant in –readiness mode to perform a readiness check prior to an upgrade.

> **Note:** This guide is meant to be used as a reference tool for your upgrade. Always consult your comonent-specific upgrade documentation for detailed information about the supported upgrade starting points, as well as the prerequisites and upgrade paths for specific installation types.

This section describes the following:

About the Upgrade Assistant

Using the Upgrade Assistant in the Overall 12c Upgrade Process
    The Upgrade Assistant is used in conjunction with other Fusion Middleware tools and processes to complete an upgrade of supported Oracle Fusion Middleware 11g and 12*c* components to the latest release.

Identifying Schemas that Can be Upgraded with the Upgrade Assistant
    Review the list of available schemas before you begin the upgrade by querying the schema version registry.

Component Schemas that Can Be Upgraded to this Release

Identifying Configurations that can be Upgraded with the Upgrade Assistant
    You can use the Upgrade Assistant to upgrade supported component configurations to 12*c* (12.2.1.1).

Before You Begin Using the Upgrade Assistant

Starting the Upgrade Assistant

Upgrading Schemas with the Upgrade Assistant
    Use the Upgrade Assistant to upgrade supported schemas to this release of Oracle Fusion Middleware.

Upgrading Oracle WebLogic Component Configurations

Performing Post Upgrade Procedures

1.1 About the Upgrade Assistant

The Oracle Fusion Middleware Upgrade Assistant is used to upgrade supported 11g and 12c component schemas, component configurations, and standalone system component configurations to Fusion Middleware 12*c* (12.2.1.1). As of release 12c (12.2.1.1), the Upgrade Assistant can also be used to run a pre-upgrade readiness check.

The Upgrade Assistant is available at the following location of your Fusion Middleware 12*c* (12.2.1.1) installation:

- On the Unix operating system, the Upgrade Assistant is located at

  `oracle_common/upgrade/bin/ua`

- On the Windows operating system, the Upgrade Assistant is located at

  `oracle_common\upgrade\bin\ua`

When you run the Upgrade Assistant, it performs the following tasks:

- When run in `−readiness` mode, the Upgrade Assistant runs a pre-upgrade check on the schemas and component configurations associated with a domain. You must perform this check while the domain is online.

- For schema upgrades, the Administration server is offline and the Upgrade Assistant examines the current Oracle home directory and identifies the list of possible component schemas to be upgraded. It also determines the order of schema upgrade based on certain dependency information present.

- For component configuration upgrades, the Administration server is offline and the list of component configurations to upgrade is determined by reading the domain configuration in an offline manner.

> **Note:** Before you start the Upgrade Assistant, Oracle recommends that you create a non-SYSDBA user. See Creating a Non-SYSDBA User

1.2 Using the Upgrade Assistant in the Overall 12c Upgrade Process

The Upgrade Assistant is used in conjunction with other Fusion Middleware tools and processes to complete an upgrade of supported Oracle Fusion Middleware 11g and 12*c* components to the latest release.

The 11*g* to 12*c* upgrade process is very different from previous releases. To understand how the Upgrade Assistant and the other upgrade tools are used in the upgrade process, see Understanding the 12c Upgrade Process

In a typical upgrade, the Upgrade Assistant may be used several times, as described below:

About Using the Upgrade Assistant to Perform a Readiness Check Before an Upgrade

About Upgrading Schemas using the Upgrade Assistant
The Upgrade Assistant provides two options for upgrading schemas: **Individually Selected Schemas** and **All Schemas Used By a Domain**.

About Using the Upgrade Assistant to Upgrade Component Configurations

1.2.1 About Using the Upgrade Assistant to Perform a Readiness Check Before an Upgrade

The Upgrade Assistant can be run in `-readiness` mode to identify potential upgrade issues before you perform an actual upgrade. The readiness check can be performed while the system is online. Readiness check is a read-only operation that scans your existing domain or database schemas and produces a text file with the results of the scan. Readiness check does not require schemas to be a part of any particular domain. If your pre-upgrade environment has issues, you can correct those issues and then rerun the readiness check before you upgrade..

Alternatively, you can run the Readiness Check in `-response` mode to perform a silent readiness check using a response file. For more information on using a response file with the Upgrade Assistant, see Starting the Upgrade Assistant in Response File Mode.

Understanding the difference between the Examine Phase and a Readiness Check

The Examine phase of the upgrade process is a separate operation from running a readiness check. Typically the readiness checks are more thorough than the checks performed during the Examine phase. The readiness check is a read-only process performed before you start the actual upgrade, while the Examine phase occurs just before you click **Upgrade** and identifies potential issues with the schemas and configuration data.

> **Note:** Run the readiness check before you start with an actual upgrade. If you have performed any of the upgrade steps, you should not run readiness check as the results can be invalid.

CAUTION: If you perform an upgrade and encounter issues that prevent a successful upgrade, you may have to revert to your backed-up, pre-upgrade environment and start the upgrade again. Running the readiness check can help identify some of these issues and prevent extended system downtime.

1.2.2 About Upgrading Schemas using the Upgrade Assistant

The Upgrade Assistant provides two options for upgrading schemas: **Individually Selected Schemas** and **All Schemas Used By a Domain**.

Individually Selected Schemas

This option enables you to choose which component schemas to upgrade. Select this option when you want to select individual schemas for upgrade and you do not want to upgrade all of the schemas used by the domain..

For example, if you want to make a trial run of Upgrade Assistant by creating schemas with RCU that are outside the domain, and then use Upgrade Assistant to upgrade them.

All Schemas Used By a Domain

This option allows the Upgrade Assistant to detect all of the available schemas within the specified domain and to include them in the upgrade. This is also known as a domain assisted schema upgrade.

1.2.3 About Using the Upgrade Assistant to Upgrade Component Configurations

After the domain has been reconfigured, you will use the Upgrade Assistant again to upgrade any component configurations.

1.3 Identifying Schemas that Can be Upgraded with the Upgrade Assistant

Review the list of available schemas before you begin the upgrade by querying the schema version registry.

This optional step can be used if you want to manually query the `schema_version_registry` table before you start the upgrade process. It is important to note that the Upgrade Assistant identifies all schemas that are available for an upgrade and allows you to select the individual schemas you want to upgrade or allow Upgrade Assistant to upgrade all of the schemas in the domain. In addition, when you run the Upgrade Assistant in `-readiness` mode, you will receive a report with all of the schemas and their current pre-upgrade versions.

If you are using an Oracle database, connect to the database as a user having Oracle DBA privileges, and run the following from SQL*Plus to get the current version numbers:

```
SET LINE 120
COLUMN MRC_NAME FORMAT A14
COLUMN COMP_ID FORMAT A20
COLUMN VERSION FORMAT A12
COLUMN STATUS FORMAT A9
COLUMN UPGRADED FORMAT A8
SELECT MRC_NAME, COMP_ID, OWNER, VERSION, STATUS, UPGRADED FROM
SCHEMA_VERSION_REGISTRY ORDER BY MRC_NAME, COMP_ID ;
```

The following report is generated when saved to a SQL script, for example `version.sql`.

If the number in the "VERSION" is at 11.1.1.7.0 or higher, and the STATUS column is 'VALID', then the schema is supported for upgrade.

If an upgrade is not needed for a schema, the `schema_version_registry` table retains the schemas at their pre-upgrade version after the upgrade.

> **Tip:**
>
> Compare the information you collect from the schema version registry and the corresponding schemas to determine whether there are schemas in your domain that are not available for an upgrade yet.

Notes about the schemas that need to be upgraded

- For **most** components, the only schema version starting points that are valid for upgrading are 11*g* Release 1 (1.1.1.7.0, 11.1.1.8.0, or 11.1.1.9.0) or 12*c* (12.1.2, 12.1.3,

or 12.2.1.0). If your schemas are not at a supported version, then you must upgrade them before using the 12*c* (12.2.1.1) upgrade procedures.

Some components, such as Oracle Enterprise Data Quality and Oracle Golden Gate Veridata, support an upgrade from versions other than the standard Oracle Fusion Middleware supported versions.

Refer to your component-specific installation and upgrade documentation for additional information about the schemas that are required for your upgrade.

- If you used an OID-based policy store in 11*g*, make sure that you create a new 12*c* (12.2.1.1) OPSS schema before you perform the upgrade. After the upgrade, the OPSS schema will still remain LDAP-based store.

- You can only upgrade schemas for products that are available for upgrade in the Oracle Fusion Middleware 12*c* (12.2.1.1) release. **Do not attempt to upgrade a domain that includes components that are not yet available for upgrade to 12*c* (12.2.1.1)**.

1.4 Component Schemas that Can Be Upgraded to this Release

When schemas are created in your database, the Repository Creation Utility (RCU) creates and maintains a table called `schema_version_registry`. Refer to this table before you perform an upgrade as some schemas may not require an upgrade.

NOTE: The Schema(s) column shows the default schema name format of prefix and schema name separated by an underscore (_) character. The default prefix is `DEV`, but you may have created new prefixes for your schemas while using the RCU.

Table 1-1   Schemas that can be upgraded to Oracle Fusion Middleware 12c (12.2.1.1)

Component Name	Schema(s)	Schema Version Before Upgrade	Schema Version After Upgrade	Dependencies and Additional Information
Audit Services[1]	*prefix*_IAU	11.1.1.7 11.1.1.9 12.1.2.0 12.1.3.0 12.2.1.0	12.2.1.0	No dependencies. 12.2.1.0 is the latest version for this schema.
Metadata Services	*prefix*_MDS	11.1.1.7 11.1.1.9 12.1.2.0 12.1.3.0 12.2.1.0	12.2.1.1	No dependencies.
Platform Security Services[2]	*prefix*_OPSS	11.1.1.7 11.1.1.9 12.1.2.0 12.1.3.0 12.2.1.0	12.2.1.0	The *prefix*_IAU schema must be upgraded first. 12.2.1.0 is the latest version for this schema.

Table 1-1   (Cont.) Schemas that can be upgraded to Oracle Fusion Middleware 12c (12.2.1.1)

Component Name	Schema(s)	Schema Version Before Upgrade	Schema Version After Upgrade	Dependencies and Additional Information
Common Infrastructure Services (LocalSvcTbl)	*prefix*_STB	12.1.2.0 12.1.3.0 12.2.1.0	12.2.1.1	The *prefix*_STB schema must be upgraded when upgrading other components, such as SOAINFRA.
User Messaging Service	*prefix*_ORASDPM	11.1.1.7 11.1.1.9 12.1.2.0 12.1.3.0 12.2.1.0	12.2.1.1	When upgrading a system that is a new installation of 12.2.1 the schema name will be called prefix_UMS.
WebLogic Diagnostic Framework	*prefix*_WLDF	11.1.1.7 12.1.2.0 12.2.1.0	12.2.1.1	
Data Integrator (Master and Work Repository)	*prefix*_ODI_REPO	11.1.1.7 11.1.1.9 12.1.2.0 12.1.3.0 12.2.1.0	12.2.1.1	None.
SOA Infrastructure (including Oracle Business Activity Monitoring, Business Process Management, and Oracle Service Bus	*prefix*_SOAINFRA	11.1.1.7 11.1.1.9 12.1.3 12.2.1.0	12.2.1.1	The *prefix*_STB schema must be upgraded first.
Oracle Enterprise Scheduler	*prefix*_ESS	11.1.1.7 11.1.1.9 12.1.3 12.2.1.0	12.2.1.1	
Oracle Managed File Transfer	*prefix*_MFT	12.1.3.0 12.2.1.0	12.2.1.1	The *prefix*_MDS, *prefix*_IAU, and *prefix*_UMS must also be upgraded.
Oracle WebCenter Content Server	*prefix*_OCS	11.1.1.7 11.1.1.8[3] 11.1.1.9 12.2.1.0	12.2.1.1	The *prefix*_**MDS** schema must be upgraded first.

Table 1-1   (Cont.) Schemas that can be upgraded to Oracle Fusion Middleware 12c (12.2.1.1)

Component Name	Schema(s)	Schema Version Before Upgrade	Schema Version After Upgrade	Dependencies and Additional Information
Oracle WebCenter Enterprise Capture	`prefix_CAPTURE`	11.1.1.8 12.2.1.0	12.2.1.1	The *prefix*_**MDS** and *prefix*_**OPSS** schemas must be upgraded first.

Oracle WebCenter Portal	`prefix_PORTLET`
	`prefix_ACTIVITIES`
	`prefix_DISCUSSIONS`
	`prefix_DISCUSSIONS_CRAWLER`
	`prefix_WEBCENTER`
		11.1.1.7[4]
		11.1.1.8
		12.2.1.0
			12.2.1.1	The *prefix*_**MDS** schema must be upgraded first..

Oracle WebCenter Sites	`prefix_SITES`	12.2.1.0	12.2.1.1	
Enterprise Data Quality	`prefix_EDQCONF` `prefix_EDQRES`	11.1.1.7.3 12.1.3.0 12.2.1.0	12.2.1.0	12.2.1.0 is the latest version for this schema.
Golden Gate Monitor	`prefix_OGGMON`	11.1.1.7 12.1.3.0 12.2.1.0	12.2.1.0	12.2.1.0 is the latest version for this schema.
GoldenGate Studio	`prefix_OGGSTUDIO`	12.2.1.0	12.2.1.1	
Veridata	`prefix_VERIDATA`	11.2.1.0 12.1.3.0 12.2.1.0	12.2.1.0	12.2.1.0 is the latest version for this schema.

[1]   When upgrading _IAU from 11*g* starting points, the Upgrade Assistant will update two auxilary schemas, IAU_APPEND and IAU_VIEWER and add them to the schema_version_registry for 12.2.1.1. These schemas were not part of the 11.1.1.7 schema_version_registry table.

[2]   As of release 11.1.1.7, OPSS audit data is stored in IAU common tables instead of the JPS table. When upgrading _OPSS schemas to 12.2..11 from 11.1.1.6, any existing data in the JPS table can remain unchanged for reporting or auditing purposes, and there is no need to upgrade JPS table with extra columns.

[3]   First release to support Edition-Based Redefinition (EBR) enabled schemas

[4]   First release to support Edition-Based Redefinition (EBR) enabled schemas

1.5 Identifying Configurations that can be Upgraded with the Upgrade Assistant

You can use the Upgrade Assistant to upgrade supported component configurations to 12*c* (12.2.1.1).

When you use the Upgrade Assistant to upgrade your configurations, it automatically determines the list of components that are configured and can be upgraded to the

latest version. After you run the Upgrade Assistant for schema upgrades, you will use the Reconfiguration Wizard to reconfigure your domain. The Upgrade Assistant is used again, after the reconfiguration, to upgrade any remaining configuration properties.

You can use the Upgrade Assistant to upgrade the following component configurations to 12*c* (12.2.1.1):

CAUTION: Do not attempt an upgrade if you want to continue using existing Fusion Middleware 11*g* components that have been deprecated or not yet available for use in the 12.2.1.1 release.

- Oracle Data Integrator (ODI)

- Oracle Enterprise Data Quality (EDQ)

- Oracle Forms

- Oracle Reports

- Oracle Golden Gate Monitor (OGGMON)

- Oracle Golden Gate Veridata

- Oracle HTTP Server (OHS)

- JRF Infrastructure (WLS)

- Oracle MapViewer

- Oracle Managed File Transfer (MFT)

- Oracle Traffic Director (OTD)

- Oracle User Messaging Service (UMS)

- Oracle WebCenter Content (WCC)

- Oracle Web Services Manager

1.6 Before You Begin Using the Upgrade Assistant

This section describes the procedures you must perform before you run the Upgrade Assistant.

> **Note:** This is not a complete list of tasks you may be required to complete before starting your upgrade process. The Oracle Fusion Middleware Pre-Upgrade Checklist provides a list of all of the tasks that may be required for a successful upgrade. This checklist is located in the component-specific upgrade guides and should be reviewed before starting an upgrade.

Checking for Invalid Database Objects Before an Upgrade

Creating a Complete Backup

Before you install the new 12*c* (12.2.1.1) distributions and begin upgrading your existing Oracle Fusion Middleware deployment, be sure

you have backed up all system-critical files; including all of the databases that host your Oracle Fusion Middleware schemas.

Stopping Servers and Processes Before an Upgrade

Creating a Non-SYSDBA User to Run Upgrade Assistant
Oracle recommends that you create a non-SYSDBA user called FMW to run the Upgrade Assistant. The FMW user has the privileges required to modify schemas, but does not have full administrator privileges.

1.6.1 Checking for Invalid Database Objects Before an Upgrade

If you are using an Oracle database, you can recompile database objects before running the Upgrade Assistant by connecting to the database as SYS and running the following from SQL*Plus:

```
SQL>@/rdbms/admin/utlrp.sql
```

This will compile the database objects.

Then use the following query to ensure there are no invalid database objects:

```
SELECT owner, object_name FROM all_objects WHERE status='INVALID';
```

None of the database objects should be INVALID before you upgrade.

If there are any invalid objects, run the `utlrp.sql` command again. If the problem persists, you should file a service request.

1.6.2 Creating a Complete Backup

Before you install the new 12*c* (12.2.1.1) distributions and begin upgrading your existing Oracle Fusion Middleware deployment, be sure you have backed up all system-critical files; including all of the databases that host your Oracle Fusion Middleware schemas.

Performing a complete database backup prior to performing a schema upgrade is a prerequisite for running Upgrade Assistant. In the Upgrade Assistant prerequisites GUI screen, you will be required to acknowledge that backups have been performed, before proceeding with the actual upgrade.

For more information, see Backing Up Your Oracle Fusion Middleware Environment and Upgrading and Preparing Your Oracle Databases for 12*c* .

> **Note:** Your system backup must include the `SYSTEM.SCHEMA_VERSION_REGISTRY$` table.

Backing Up the Schema Version Registry Table
Your system backup must include the `SYSTEM.SCHEMA_VERSION_REGISTRY$` table.

1.6.2.1 Backing Up the Schema Version Registry Table

Your system backup must include the `SYSTEM.SCHEMA_VERSION_REGISTRY$` table.

Each Fusion Middleware schema has a row in the `SYSTEM.SCHEMA_VERSION_REGISTRY$` table. If you run the Upgrade Assistant to update an existing schema and it does not succeed, you must restore the original

schema before you can try again. Make sure you back up your existing database schemas before you run the Upgrade Assistant.

> **Note:** Performing these backups prior to performing a schema upgrade is a prerequisite for running Upgrade Assistant. In the Upgrade Assistant prerequisites GUI screen, you will be required to acknowledge that backups have been performed, before proceeding with the actual upgrade.

1.6.3 Stopping Servers and Processes Before an Upgrade

Before running Upgrade Assistant, shut down all Oracle Fusion Middleware Managed Servers, Administration Servers, and system components (such as OHS) that may be using the schemas or configuration data you want to upgrade.

In addition, the databases that contain Fusion Middleware database schemas must be quiesced before running Upgrade Assistant. You can either shutdown and restart the database server, or ensure that there are no other database users operating on the same database instance that will be upgraded by Upgrade Assistant. Failure to do so may result in an incomplete or failed upgrade.

If you are running Node Manager, you should also stop Node Manager. You can do this by closing the console window in which Node Manager is running, or by using the `stopNodeManager` WLST command.

Instructions for stopping an Oracle Fusion Middleware environment are provided in "Stopping an Oracle Fusion Middleware Environment".

1.6.4 Creating a Non-SYSDBA User to Run Upgrade Assistant

Oracle recommends that you create a non-SYSDBA user called FMW to run the Upgrade Assistant. The FMW user has the privileges required to modify schemas, but does not have full administrator privileges.

SYSDBA is an administrative privilege, required to perform high-level administrative operations such as creating, starting up, shutting down, backing up, or recovering the database. The SYSDBA system privilege is for a fully empowered database administrator. When you connect with the SYSDBA privilege, you connect with a default schema and not with the schema that is generally associated with your user name. For SYSDBA, this schema is SYS. Access to a default schema can be a very powerful privilege. For example, when you connect as user SYS, you have unlimited privileges on data dictionary tables. Therefore, Oracle recommends that you create a Non-SYSDBA user to upgrade the schemas. The privileges listed in this topic must be granted before starting the Upgrade Assistant.

> **Note:** The v$xatrans$ table does not exist by default. You must run the `XAVIEW.SQL` script to create this table before creating the user. Moreover, grant on v$xatrans$ table is required only for Oracle Identity Manager. If you do not require Oracle Identity Manager for configuration or if you do not have the v$xatrans$ table, then remove the following line from the script:
>
> ```
> grant select on v$xatrans$ to FMW with grant option;
> ```

In the example below, `welcome1` is the password. Make sure that you specify your actual password when granting privileges.

```
create user FMW identified by welcome1;
grant dba to FMW;
grant execute on DBMS_LOB to FMW with grant option;
grant execute on DBMS_OUTPUT to FMW with grant option;
grant execute on DBMS_STATS to FMW with grant option;
grant execute on sys.dbms_aqadm to FMW with grant option;
grant execute on sys.dbms_aqin to FMW with grant option;
grant execute on sys.dbms_aqjms to FMW with grant option;
grant execute on sys.dbms_aq to FMW with grant option;
grant execute on utl_file to FMW with grant option;
grant execute on dbms_lock to FMW with grant option;
grant select on sys.V_$INSTANCE to FMW with grant option;
grant select on sys.GV_$INSTANCE to FMW with grant option;
grant select on sys.V_$SESSION to FMW with grant option;
grant select on sys.GV_$SESSION to FMW with grant option;
grant select on dba_scheduler_jobs to FMW with grant option;
grant select on dba_scheduler_job_run_details to FMW with grant option;
grant select on dba_scheduler_running_jobs to FMW with grant option;
grant select on dba_aq_agents to FMW with grant option;
grant execute on sys.DBMS_SHARED_POOL to FMW with grant option;
grant select on dba_2pc_pending to FMW with grant option;
grant select on dba_pending_transactions to FMW with grant option;
grant execute on DBMS_FLASHBACK to FMW with grant option;
grant execute on dbms_crypto to FMW with grant option;
grant execute on DBMS_REPUTIL to FMW with grant option;
grant execute on dbms_job to FMW with grant option;
grant select on pending_trans$ to FMW with grant option;
grant select on dba_scheduler_job_classes to fmw with grant option;
grant select on SYS.DBA_DATA_FILES to FMW with grant option;
grant select on SYS.V_$ASM_DISKGROUP to FMW with grant option;
grant select on v$xatrans$ to FMW with grant option;
grant execute on sys.dbms_system to FMW with grant option;
grant execute on DBMS_SCHEDULER to FMW with grant option;
grant select on dba_data_files to FMW with grant option;
grant execute on UTL_RAW to FMW with grant option;
grant execute on DBMS_XMLDOM to FMW with grant option;
grant execute on DBMS_APPLICATION_INFO to FMW with grant option;
grant execute on DBMS_UTILITY to FMW with grant option;
grant execute on DBMS_SESSION to FMW with grant option;
grant execute on DBMS_METADATA to FMW with grant option;
grant execute on DBMS_XMLGEN to FMW with grant option;
grant execute on DBMS_DATAPUMP to FMW with grant option;
grant execute on DBMS_MVIEW to FMW with grant option;
grant select on ALL_ENCRYPTED_COLUMNS to FMW with grant option;
grant select on dba_queue_subscribers to FMW with grant option;
grant execute on SYS.DBMS_ASSERT to FMW with grant option;
```

Note:

Oracle Database 11.2.0.3 Database Users ONLY: You must apply **Oracle Patch 13036331** before you begin the upgrade. Go to My Oracle Support to download the patch.

If you do not apply this patch, then you will have to grant additional privileges for some schemas.

1.7 Starting the Upgrade Assistant

This topic describes how to run the Oracle Fusion Middleware Upgrade Assistant. You can run the Upgrade Assistant either in the Graphical User Interface (GUI) mode or in the response file mode.

You can use the optional command-line interface arguments, documented in this topic, to upgrade your Oracle Fusion Middleware components to the latest version. In addition, a response file can be generated to automate some of the upgrade tasks.

This topic contains the following sections:

Starting the Upgrade Assistant in Graphical User Interface (GUI) Mode
The Upgrade Assistant is used to upgrade schemas, component configurations and standalone system components.

Starting the Upgrade Assistant in Response File Mode
Silent or "hands free" upgrades can be performed using a response file. The response file can only be created after you have provided the information in the Upgrade Assistant screens.

1.7.1 Starting the Upgrade Assistant in Graphical User Interface (GUI) Mode

The Upgrade Assistant is used to upgrade schemas, component configurations and standalone system components.

Oracle recommends that you successfully complete the upgrade of schemas and component configurations for a single domain before beginning the upgrade of another domain.

Note: The Upgrade Assistant should be run by a non-SYSDBA user whenever possible. The steps to create a user who has the privileges required to upgrade the schemas are described in Creating a Non-SYSDBA User.

To start the Upgrade Assistant:

1. On UNIX operating systems:: change directory to *ORACLE_HOME*/ `oracle_common/upgrade/bin`.

   On Windows operating systems: change directory to *ORACLE_HOME* `\oracle_common\upgrade\bin`.

2. Enter the following command to start the Upgrade Assistant:

   On UNIX operating systems:

   `./ua`

   On Windows operating systems:

   `ua.bat`

   You can also launch the Upgrade Assistant with logging parameters as shown in the UNIX example below:

   ./ua [-logLevel <*log_level*>] [-logDir <*log_directory*>]

   Logging level. Select one of the following:

- TRACE

- NOTIFICATION

- WARNING

- ERROR

- INCIDENT_ERROR

The default logging level is NOTIFICATION.

> **Note:** When troubleshooting, consider setting the `-logLevel` to `TRACE` so that more information will be logged. If additional information is not needed, change the logLevel as the Upgrade Assistant's log files can become very large when `-logLevel TRACE` is used.

Starting the Upgrade Assistant with Additional Parameters (Optional)

1.7.1.1 Starting the Upgrade Assistant with Additional Parameters (Optional)

Table 1-2 lists the command-line parameters you can use while running the Upgrade Assistant in the GUI mode. The following example shows the usage of these parameters on your respective operating system:

On UNIX operating systems:

Change directory to *ORACLE_HOME*/oracle_common/upgrade/bin

Enter the command: `./ua -help`

On Windows operating systems:

Change directory to *ORACLE_HOME*\oracle_common\upgrade\bin

Enter the command: `ua.bat -help`

> **Note:** If you get an Xlib error when starting the Oracle Upgrade Assistant such as "Failed to connect to server", "Connection refused by server", or "Can't open display", then you must set the DISPLAY environment variable and restart the Upgrade Assistant as described in Setting the DISPLAY Environment Variable.

Table 1-2   Upgrade Assistant GUI Command Line Parameters

Table 1-2   (Cont.) Upgrade Assistant GUI Command Line Parameters

Parameter	Required or Optional Parameter?	Description
-logLevel	Optional.	Logging level. Select one of the following: • TRACE • NOTIFICATION • WARNING • ERROR • INCIDENT_ERROR The default logging level is NOTIFICATION.

Note: When troubleshooting, consider setting the -logLevel to TRACE so that more information will be logged. If additional information is not needed, change the logLevel as the Upgrade Assistant's log files can become very large when -logLevel TRACE is used.

Note: TRACE message

Table 1-2   (Cont.) Upgrade Assistant GUI Command Line Parameters

Parameter	Required or Optional Parameter?	Description

-logDir	Optional.	Change the default location of upgrade log files and temporary files. You must specify an existing, writable directory where Upgrade Assistant will create log files and temporary files.
		
		On UNIX operating systems, the default locations are:
		
		`ORACLE_HOME/oracle_common/ upgrade/logs`
		`ORACLE_HOME/oracle_common/ upgrade/temp`
		
		On Windows operating systems, the default locations are:
		
		`ORACLE_HOME\oracle_common \upgrade\logs`
		`ORACLE_HOME\oracle_common \upgrade\temp`

	Note: If you have installed the Upgrade Assistant in a read-only directory, you must specify this parameter to ensure the log directory is writable. If the log directory is read-only, the Upgrade Assistant will fail.	
-threads	Optional	Identify the number of threads available for concurrent schema upgrades or readiness checks of the schemas. The value must be a positive integer in the range 1 to 8. Default is 4 threads.

Table 1-2   (Cont.) Upgrade Assistant GUI Command Line Parameters

Parameter	Required or Optional Parameter?	Description
-readiness	Required for Readiness Check	Performs the upgrade readiness check without performing any actual examines or upgrades. Schemas and configuration are checked. **Note:** -readiness option may not appear on the command line with -examine option. For more information, see Performing the Readiness Check
-help	Optional.	Prints all the command-line options to the console.

1.7.2 Starting the Upgrade Assistant in Response File Mode

Silent or "hands free" upgrades can be performed using a response file. The response file can only be created after you have provided the information in the Upgrade Assistant screens.

The following topics describe how you can upgrade the supported Oracle Fusion Middleware components using a response file. The response file collects all the information that you have entered through the Upgrade Assistant's graphical user interface screens, and performs exactly the same function that the Upgrade Assistant wizard performs.

Note: You must run the Upgrade Assistant in GUI mode first to generate the response file that will be used to complete your silent upgrade.

Creating an Upgrade Response File

Using the Response File to Upgrade Fusion Middleware

1.7.2.1 Creating an Upgrade Response File

The **Save Response File** option on the Upgrade Summary screen creates a file that uses the information you have already provided in the Upgrade Assistant screens. The response file enables you to use the saved information instead of manually entering data through the Upgrade Assistant wizard screens.

Once you select the **Save Response File** option, you will be prompted for a name and location where you want to create this response file. After it is created, you can use it exactly as-is to replicate the upgrade options on other systems, or modify it as needed.

For more information, see Using the Response File to Upgrade Fusion Middleware.

1.7.2.2 Using the Response File to Upgrade Fusion Middleware

To perform upgrades using a response file from the command-line interface (CLI), use the following command:

On Unix operating systems:

Change directory to *ORACLE_HOME*/oracle_common/upgrade/bin

Execute the following:

```
./ua -response <response_file> [-examine] [-logLevel
<log_level>] [-logDir <log_directory>] [-threads <number>]
```

On Windows operating systems:

Change directory to *ORACLE_HOME*\oracle_common\upgrade\bin

Execute the following:

```
ua.bat -response <response_file> [-examine] [-logLevel
<log_level>] [-logDir <log_directory>] [-threads <number>]
```

Table 1-3   Upgrade Assistant Response File Mode Command Line Parameters

Parameter	Required or Optional Parameter?	Description
-readiness	Required for Readiness Check	Performs the upgrade readiness check without performing any actual examines or upgrades. Schemas and configuration are checked.

Note:
Do not specify this parameter if you have specified the -examine parameter.

For more information, see Performing the Readiness Check

Table 1-3   (Cont.) Upgrade Assistant Response File Mode Command Line Parameters

Parameter	Required or Optional Parameter?	Description
-threads	Optional	Identify the number of threads available for concurrent schema upgrades or readiness checks of the schemas. The value must be a positive integer in the range 1 to 8. Default is 4 threads.
-response	Required.	File containing inputs required to perform an upgrade. This file can be generated from inputs entered when the Upgrade Assistant is run in graphical mode.
-examine	Optional.	If this option is present, Upgrade Assistant performs the examine phase but DOES NOT perform any actual upgrades. **Note:** Do not specify this parameter if you have specified the -readiness parameter.

Table 1-3   (Cont.) Upgrade Assistant Response File Mode Command Line Parameters

Parameter	Required or Optional Parameter?	Description
`-logLevel`	Optional.	Logging level. Select one of the following: • TRACE • NOTIFICATION • WARNING • ERROR • INCIDENT_ERROR The default logging level is NOTIFICATION. **Note:** Consider setting the `-logLevel` to `TRACE` so that more information will be logged. This will be useful when troubleshooting a failed upgrade. The Upgrade Assistant's log files can become very large if `-logLevel` `TRACE` is used.

Table 1-3 (Cont.) Upgrade Assistant Response File Mode Command Line Parameters

Parameter	Required or Optional Parameter?	Description
-logDir	Optional.	Change the default location of upgrade log files and temporary files. You must specify an existing, writable directory where Upgrade Assistant will create log files and temporary files. On UNIX operating systems, the default locations are: *ORACLE_HOME*/ oracle_common/upgrade/ logs *ORACLE_HOME*/ oracle_common/upgrade/ temp On Windows operating systems, the default locations are: *ORACLE_HOME* \oracle_common\upgrade \logs *ORACLE_HOME* \oracle_common\upgrade \temp
-help	Optional.	View all of the command line options.

1.8 Upgrading Schemas with the Upgrade Assistant

Use the Upgrade Assistant to upgrade supported schemas to this release of Oracle Fusion Middleware.

The screens that are displayed when you run the Upgrade Assistant to upgrade schemas vary depending on the options you select and the contents of your pre-upgrade environment.

Table 1-4 Upgrading Schemas: Navigating the Upgrade Assistant Screens

Screen Title	When does the screen appear?	Description
Welcome	Always.	This screen provides an overview of the Upgrade Assistant and some information about important pre-upgrade tasks.

Table 1-4   (Cont.) Upgrading Schemas: Navigating the Upgrade Assistant Screens

Screen Title	When does the screen appear?	Description
Schemas	Always.	Select the schema upgrade operation that you want to perform on this screen. The options on the screens change depending on what you select from the following: • **Individually Selected Schemas** • **All Schemas Used by a Domain**
Available Components	When you select **Individually Selected Schemas**.	This screen provides a list of installed Oracle Fusion Middleware components that have schemas that can be upgraded. When you select a component, the schemas and any dependencies are automatically selected.
All Schemas Component List	When you select **All Schemas Used by a Domain**.	This screen is read-only, and it displays all the components and schemas found in the specific domain directory that are included in the upgrade.
Prerequisites	Always.	This screen requires you to acknowledge that all prerequisites have been met before you continue with the upgrade. Check the boxes before you continue.
Schema Credentials Screen	Always.	Enter the information required to connect to the selected schema and the database that hosts the schema on this screen. The screen name changes based on the type of schema selected ("MDS Schema", for example). Since the component ID or schema name is changed for UCSUMS schema as of release 12.1.2, the Upgrade Assistant does not automatically recognize the possible schemas and display them in a drop-down list. You must manually enter the name in a text field. The name can be either *prefix*_ORASDPM or *prefix*_UMS, depending on the starting point for the upgrade.

Table 1-4   (Cont.) Upgrading Schemas: Navigating the Upgrade Assistant Screens

Screen Title	When does the screen appear?	Description
Examine	Always.	This screen displays the status of the Upgrade Assistant as it examines each component, verifying that the component is ready for upgrade.
		Note: Issues detected during the Examination phase can be fixed without restoring from backup.
Upgrade Summary	Always.	Review a summary of the options you have selected and to start the upgrade process on this screen.
Upgrade Progress	Always.	This screen displays the status of the upgrade process.
Upgrade Success	When the upgrade is successful.	The upgrade is successful. The Post-Upgrade Actions window describes the manual tasks you must perform to make the component function in the new installation.

Table 1-4   (Cont.) Upgrading Schemas: Navigating the Upgrade Assistant Screens

Screen Title	When does the screen appear?	Description
Upgrade Failure	When the upgrade fails.	The upgrade failed for the specified component(s). You must restart the Upgrade Assistant. The Upgrade Assistant logs are available at *ORACLE_HOME*/`oracle_common/upgrade/logs`. **Note:** If the upgrade fails you must restore your pre-upgrade environment from backup, fix the issues and then restart the Upgrade Assistant. You cannot fix the issues and restart the Upgrade Assistant because the files are modified during this operation.

1.9 Upgrading Oracle WebLogic Component Configurations

If you are running the Upgrade Assistant from an Oracle home that contains managed WebLogic domain components, then the **All Configurations Used by a Domain** upgrade option is available.

After upgrading the component schemas, you must run the Reconfiguration Wizard to reconfigure the domain. For more information, see "*Reconfiguring an Oracle WebLogic Domain with the Reconfiguration Wizard*". After the reconfiguring the domain, use the Upgrade Assistant again to upgrade the component configurations as described in this section.

The Upgrade Assistant will display the following screens when upgrading a WebLogic component configurations, such as Oracle Web Services Manager (OWSM).

Table 1-5   Upgrade Assistant Screens: Upgrading Oracle WebLogic Component Configurations

Screen	When Screen Appears	Description
Welcome	Always.	This screen provides an overview of the Upgrade Assistant and some information about important pre-upgrade tasks.
All Configurations Used by a Domain	When **All Configurations Used by a Domain** is selected as the upgrade type, the screen name is WebLogic Components.	Select the **All Configurations Used by a Domain** option to upgrade component configurations for a managed WebLogic Server domain. Configuration upgrades are offline in 12.2.1. You must enter the domain directory for the domain that you are upgrading now.
WebLogic Server Component List	Only when the **All Configurations Used by a Domain** option is selected.	This screen provides a list of components that will be included in the WebLogic domain's component configuration upgrade. The name of the domain is provided along with the list of components located within the domain.
Prerequisites	Always.	This screen requires you to acknowledge that all prerequisites have been met before you continue with the upgrade. You must check the boxes before you can continue.

Table 1-5 (Cont.) Upgrade Assistant Screens: Upgrading Oracle WebLogic Component Configurations

Screen	When Screen Appears	Description
User Messaging Service Configuration	Only when there are remote managed servers hosting UMS 11g configuration files, that is, when the required configuration files are not locally accessible at the Administration Server.	This screen specifies the login credentials of the remote managed servers hosting your UMS 11g configuration files. The Upgrade Assistant automatically copies remote configuration files if all necessary prerequisites are met and the required login information is provided.
Examine	Always.	This screen displays the status of the Upgrade Assistant as it examines each component, verifying that the component is ready for upgrade.
Upgrade Summary	Always.	Use this screen to review a summary of the options you have selected and to start the upgrade process.
Upgrade Progress	Always.	This screen shows the status of the upgrade process.
Upgrade Success Or Upgrade Failure	Always.	The upgrade was successful. The Post-Upgrade Actions window describes the manual tasks you must perform to make the component function in the new installation. Or: The upgrade failed for the specified component(s) and Upgrade Assistant will have to be restarted.

1.10 Performing Post Upgrade Procedures

The following topics provide information about basic tasks performed after the upgrade is complete. Some of the tasks may not apply to your environment, as you may not be upgrading the products listed.

Always refer to your component-specific upgrade documentation for more information on post-upgrade procedures.

NOTE: You should be able to successfully complete the tasks described in this section after an upgrade. If you are unable to complete one or more of these tasks in your newly upgraded environment, see Troubleshooting Your Upgrade .

Performing Basic Post-Upgrade Administrative Tasks

Verifying a Successful Schema Upgrade

Checking for Invalid Database Objects

1.10.1 Performing Basic Post-Upgrade Administrative Tasks

This section describes some common administration tasks you will likely want to perform on your newly upgraded domain.

NOTE: The administrative tasks in the table below are optional. Only perform those tasks that apply to your upgraded environment.

Table 1-6   Basic Post-Upgrade Administration Tasks

Task	Description	More Information
Starting and stopping products and servers	Learn how to start and stop Oracle Fusion Middleware, including the Administration Server, Managed Servers, and components. Performing these tasks will validate that the upgrade was successful.	"Starting and Stopping Oracle Fusion Middleware"
Starting and stopping upgraded applications.	Learn how to start your upgraded applications in the new 12.2.1 environment to verify they are working as expected.	"Starting and Stopping Applications"
Configuring Secure Sockets Layer (SSL)	Learn how to set up secure communications among between Oracle Fusion Middleware components using SSL.	"Configuring SSL in Oracle Fusion Middleware"
Deploying Applications	Learn how to deploy your applications to Oracle Fusion Middleware.	"Deploying Applications"
Monitoring Oracle Fusion Middleware	Learn how to keep track of the status of Oracle Fusion Middleware components.	"Monitoring Oracle Fusion Middleware"
Adding a Web Tier front-end to your WebLogic domain	Oracle Web Tier hosts Web pages (static and dynamic), provides security and high performance along with built-in clustering, load balancing, and failover features. In particular, the Web Tier contains Oracle HTTP Server.	*Installing and Configuring Oracle HTTP Server*.

Table 1-6   (Cont.) Basic Post-Upgrade Administration Tasks

Task	Description	More Information
Tuning and configuring Coherence for your topology.	The standard installation topology includes a Coherence cluster that contains storage-enabled Managed Coherence Servers. This configuration is a good starting point for using Coherence, but depending upon your specific requirements, consider tuning and reconfiguring Coherence to improve performance in a production environment.	For information about Coherence clusters, see "Configuring and Managing Coherence Clusters" For information about tuning Coherence, see *Administering Oracle Coherence*. For information about storing HTTP session data in Coherence, see "Using Coherence*Web with WebLogic Server". For more information about creating and deploying Coherence applications, see *Developing Oracle Coherence Applications for Oracle WebLogic Server*.

1.10.2 Verifying a Successful Schema Upgrade

You can use the following SQL command to verify that the schema version in `schema_version_registry` has been properly upgraded:

```
SET LINE 120
COLUMN MRC_NAME FORMAT A14
COLUMN COMP_ID FORMAT A20
COLUMN VERSION FORMAT A12
COLUMN STATUS FORMAT A9
COLUMN UPGRADED FORMAT A8
SELECT MRC_NAME, COMP_ID, OWNER,
VERSION, STATUS, UPGRADED FROM
SCHEMA_VERSION_REGISTRY ORDER BY MRC_NAME, COMP_ID ;
```

All the numbers in the "VERSION" column should be the versions specified during your pre-upgrade check . For more information, see Identifying Schemas that Can be Upgraded with the Upgrade Assistant.

In the query results, the STATUS field will be either "UPGRADING" or "UPGRADED" during the schema patching operation, and will become "VALID" when the operation is finished.

If the status appears as "INVALID" then the schema upgrade failed. You should examine the logs files to determine the reason for the failure.

1.10.3 Checking for Invalid Database Objects

If you are using an Oracle database, you should recompile database objects after running the Upgrade Assistant by connecting to the database as SYS and running the following from SQL*Plus:

```
SQL>@/rdbms/admin/utlrp.sql
```

This will compile the database objects that were upgraded by Upgrade Assistant.

Then issue the following query to ensure there are no longer any invalid database objects:

```
SELECT owner, object_name FROM all_objects WHERE status='INVALID';
```

None of the database objects for the upgraded schema should be invalid at this point. If there are any, run the `utlrp.sql` command again and check again. If the problem persists, you should file a service request.

2

About Running a Pre-Upgrade Readiness Check with the Upgrade Assistant

You can run the Upgrade Assistant in -readiness mode to detect issues before you perform the actual upgrade. This can be done using the GUI or with silent upgrades using the response files.

The Upgrade Assistant readiness check performs a **read-only**, pre-upgrade review of your existing Oracle Fusion Middleware schemas and Oracle WebLogic configurations.

The readiness check generates a formatted, time-stamped readiness report so you can address potential issues before you attempt the actual upgrade. If no issues are detected, you can begin the upgrade process. Oracle recommends that you read this report thoroughly before performing an upgrade.

For more information, see the Sample Readiness Report..

> **Note:** Alternatively, you can run the readiness check in -response mode to perform a silent readiness check using a response file. For more information on using a response file with the Upgrade Assistant, see Starting the Upgrade Assistant in Response File Mode.

You can run the readiness check while your existing Oracle Fusion Middleware domain is online (while other users are actively using it), or offline.

Readiness checks can be run any number of times before any actual upgrades are attempted. However, do not run the readiness check after an upgrade has been performed, as the report will not provide valid results.

> **Note:**
>
> Oracle recommends that you run the readiness checks during off-peak hours to prevent possible performance degradation.

Starting the Upgrade Assistant in Readiness Mode

Performing the Readiness Check

Understanding the Readiness Report

2.1 Starting the Upgrade Assistant in Readiness Mode

To perform a readiness check on your pre-upgrade environment, you will launch the Upgrade Assistant in `-readiness` mode as shown below:

1. Change directory to *ORACLE_HOME*/oracle_common/upgrade/bin on Unix operating systems or *ORACLE_HOME*\oracle_common\upgrade\bin on Windows operating systems.

2. Enter the following command to start the Upgrade Assistant.

    On UNIX operating systems:

    ```
    ./ua -readiness
    ```

    On Windows operating systems:

    ```
    ua.bat -readiness
    ```

    Provide the required information in each of the Upgrade Assistant screens. The screens you see will vary depending on the upgrade options you select. The sections below describe the upgrade options and the information you will need to provide.

2.2 Performing the Readiness Check

When the Upgrade Assistant is started in -readiness mode, the following screens appear.

Alternatively, you can run the readiness check using a response file. For more information on using a response file with the Upgrade Assistant, see Starting the Upgrade Assistant in Response File Mode.

Note that these screens are a subset of the screens you will see.

Table 2-1   Upgrade Assistant Screens: Readiness Check

Screen	When Screen Appears	Description
Welcome	Always.	This screen provides an overview of the readiness check.
Readiness Check Type: • Individually Selected Schemas • Domain Based	Always.	Readiness checks are only performed on schemas or component configurations that are at a supported upgrade starting point. There are two options to choose from. These options are described below: • Readiness Check Type: Individually Selected Schemas Use the **Individually Selected Schemas** option to be able to select the schemas you want to review prior to upgrade. • Readiness Check Type: Domain Based Use the **Domain Based** option to let the Upgrade Assistant perform a readiness check per domain.

Table 2-1   (Cont.) Upgrade Assistant Screens: Readiness Check

Screen	When Screen Appears	Description
Available Components	When **Individually Selected Schemas** option is selected.	This screen lists the available components for which the schemas will be selected. If you select something here, readiness check will be performed on that component's schema.
Schema Credentials	Always.	Use this screen to enter information required to connect to the selected schema and the database that hosts the schema. If the schema that is to be upgraded was created by RCU in a prior Fusion Middleware release then you will see a drop-down menu listing the possible schema names.
Readiness Summary	Always.	This screen provides a high-level overview of the readiness checks to be performed based on your selections. Click **Save Response File** if you plan to run the Upgrade Assistant again in —response (or silent) mode.
Readiness Check	Always.	This screen displays the current status of the readiness check. Depending on what you have selected to check, the process can take several minutes. For a detailed report, click **View Readiness Report**. This button appears only after all the readiness checks are complete. **Caution:** To prevent performance degradation, consider running the readiness check during off-peak hours.

Table 2-1   (Cont.) Upgrade Assistant Screens: Readiness Check

Screen	When Screen Appears	Description
Readiness Success	If the readiness check completes successfully.	You can now review the complete report. If the readiness check encounters an issue or error, review the log file to identify the issues, correct the issues, and then restart the readiness check.

2.3 Understanding the Readiness Report

Now that you have completed the readiness checks for your domain, review the report to determine what actions - if any - need to be taken before the completion of a successful upgrade.

Each Readiness Report contains the following information:

Report Information	Description	Required Action
Overall Readiness Status: SUCCESS or FAILURE	The top of the report indicates whether the Upgrade readiness check passed or completed with one or more errors.	If the report completed with one or more errors, search for FAIL and correct the failing issues before attempting to upgrade. You can re-run the readiness check as many times as necessary before an upgrade.
Timestamp	This is the date and time that the report was generated.	No action required.
Log file location	This is the directory location of the generated log file.	No action required.
Readiness Report location	This is the directory location of the generated readiness report.	No action required.
Names of components that were checked	The names and versions of the components included in the check and status.	If your domain includes components that cannot be upgraded to this release, such as SOA Core Extension, then do not attempt an upgrade.
Names of schemas that were checked	The names and current versions of the schemas included in the check and status.	Review the version numbers of your schemas. If your domain includes schemas that cannot be upgraded to this release, then do not attempt an upgrade.
Status: FAIL	The individual readiness check test detected an issue.	Do not upgrade until all FAILED issues have been resolved.

Report Information	Description	Required Action
Status: PASS	The readiness check test detected no issues.	If your readiness check report shows only the PASS status, then you can upgrade your environment. Note, however, that the Readiness Check cannot detect issues with externals such as hardware or connectivity during an upgrade. You should always monitor the progress of your upgrade.

Here is a sample Readiness Report file. Your report may or may not include all of these checks.

```
Upgrade readiness check completed with one or more errors.

This readiness check report was created on Tue May 30 11:15:52 EDT 2016
Log file is located at: /scratch/yourname/oracle/work/middleware_latest/
oracle_common/upgrade/logs/ua2016-05-30-11-14-06AM.log
Readiness Check Report File: /scratch/yourname/oracle/work/middleware_latest/
oracle_common/upgrade/logs/readiness2016-05-30-11-15-52AM.txt

Starting readiness check of components.

Oracle Metadata Services
    Starting readiness check of Oracle Metadata Services.
      Schema User Name: DEV11_MDS
      Database Type: Oracle Database
      Database Connect String: machinename@yourcompany.com
      VERSION Schema DEV11_MDS is currently at version 12.1.1.1.0.  Readiness checks
will now be performed.
    Starting schema test:  TEST_REQUIRED_TABLES  Test that the schema contains all
the required tables
    Completed schema test: TEST_REQUIRED_TABLES --> Test that the schema contains all
the required tables +++ PASS
    Starting schema test:  TEST_REQUIRED_PROCEDURES  Test that the schema contains
all the required stored procedures
      EXCEPTION     Schema is missing a required procedure: GETREPOSITORYFEATURES
    Completed schema test: TEST_REQUIRED_PROCEDURES --> Test that the schema contains
all the required stored procedures +++ FAIL
    Starting schema test:  TEST_REQUIRED_VIEWS  Test that the schema contains all the
required database views
    Completed schema test: TEST_REQUIRED_VIEWS --> Test that the schema contains all
the required database views +++ PASS
    Starting index test for table MDS_ATTRIBUTES:  TEST_REQUIRED_INDEXES --> Test
that the table contains all the required indexes
    Completed index test for table MDS_ATTRIBUTES: TEST_REQUIRED_INDEXES --> Test
that the table contains all the required indexes +++ PASS
    Starting index test for table MDS_COMPONENTS:  TEST_REQUIRED_INDEXES --> Test
that the table contains all the required indexes
    Completed index test for table MDS_COMPONENTS: TEST_REQUIRED_INDEXES --> Test
that the table contains all the required indexes +++ PASS
    Starting index test for table MDS_DEPENDENCIES:  TEST_REQUIRED_INDEXES --> Test
that the table contains all the required indexes
    Completed index test for table MDS_DEPENDENCIES: TEST_REQUIRED_INDEXES --> Test
that the table contains all the required indexes +++ PASS
    Starting index test for table MDS_DEPL_LINEAGES:  TEST_REQUIRED_INDEXES --> Test
that the table contains all the required indexes
    Completed index test for table MDS_DEPL_LINEAGES: TEST_REQUIRED_INDEXES --> Test
that the table contains all the required indexes +++ PASS
    Starting index test for table MDS_LABELS:  TEST_REQUIRED_INDEXES --> Test that
```

```
the table contains all the required indexes
    Completed index test for table MDS_LABELS: TEST_REQUIRED_INDEXES --> Test that
the table contains all the required indexes +++ PASS
    Starting index test for table MDS_LARGE_ATTRIBUTES:  TEST_REQUIRED_INDEXES -->
Test that the table contains all the required indexes
    Completed index test for table MDS_LARGE_ATTRIBUTES: TEST_REQUIRED_INDEXES -->
Test that the table contains all the required indexes +++ PASS
    Starting index test for table MDS_METADATA_DOCS:  TEST_REQUIRED_INDEXES --> Test
that the table contains all the required indexes
    Completed index test for table MDS_METADATA_DOCS: TEST_REQUIRED_INDEXES --> Test
that the table contains all the required indexes +++ PASS
    Starting index test for table MDS_NAMESPACES:  TEST_REQUIRED_INDEXES --> Test
that the table contains all the required indexes
    Completed index test for table MDS_NAMESPACES: TEST_REQUIRED_INDEXES --> Test
that the table contains all the required indexes +++ PASS
    Starting index test for table MDS_PARTITIONS:  TEST_REQUIRED_INDEXES --> Test
that the table contains all the required indexes
    Completed index test for table MDS_PARTITIONS: TEST_REQUIRED_INDEXES --> Test
that the table contains all the required indexes +++ PASS
    Starting index test for table MDS_PATHS:  TEST_REQUIRED_INDEXES --> Test that the
table contains all the required indexes
    Completed index test for table MDS_PATHS: TEST_REQUIRED_INDEXES --> Test that the
table contains all the required indexes +++ PASS
    Starting index test for table MDS_PURGE_PATHS:  TEST_REQUIRED_INDEXES --> Test
that the table contains all the required indexes
    Completed index test for table MDS_PURGE_PATHS: TEST_REQUIRED_INDEXES --> Test
that the table contains all the required indexes +++ PASS
    Starting index test for table MDS_SANDBOXES:  TEST_REQUIRED_INDEXES --> Test that
the table contains all the required indexes
    Completed index test for table MDS_SANDBOXES: TEST_REQUIRED_INDEXES --> Test that
the table contains all the required indexes +++ PASS
    Starting index test for table MDS_STREAMED_DOCS:  TEST_REQUIRED_INDEXES --> Test
that the table contains all the required indexes
    Completed index test for table MDS_STREAMED_DOCS: TEST_REQUIRED_INDEXES --> Test
that the table contains all the required indexes +++ PASS
    Starting index test for table MDS_TRANSACTIONS:  TEST_REQUIRED_INDEXES --> Test
that the table contains all the required indexes
    Completed index test for table MDS_TRANSACTIONS: TEST_REQUIRED_INDEXES --> Test
that the table contains all the required indexes +++ PASS
    Starting index test for table MDS_TXN_LOCKS:  TEST_REQUIRED_INDEXES --> Test that
the table contains all the required indexes
    Completed index test for table MDS_TXN_LOCKS: TEST_REQUIRED_INDEXES --> Test that
the table contains all the required indexes +++ PASS
   Starting schema test:  TEST_REQUIRED_TRIGGERS  Test that the schema has all the
required triggers
   Completed schema test: TEST_REQUIRED_TRIGGERS --> Test that the schema has all
the required triggers +++ PASS
   Starting schema test:  TEST_MISSING_COLUMNS  Test that tables and views are not
missing any required columns
   Completed schema test: TEST_MISSING_COLUMNS --> Test that tables and views are
not missing any required columns +++ PASS
   Starting schema test:  TEST_UNEXPECTED_TABLES  Test that the schema does not
contain any unexpected tables
   Completed schema test: TEST_UNEXPECTED_TABLES --> Test that the schema does not
contain any unexpected tables +++ PASS
   Starting schema test:  TEST_UNEXPECTED_PROCEDURES  Test that the schema does not
contain any unexpected stored procedures
   Completed schema test: TEST_UNEXPECTED_PROCEDURES --> Test that the schema does
not contain any unexpected stored procedures +++ PASS
   Starting schema test:  TEST_UNEXPECTED_VIEWS  Test that the schema does not
contain any unexpected views
```

```
    Completed schema test: TEST_UNEXPECTED_VIEWS --> Test that the schema does not
contain any unexpected views +++ PASS
    Starting index test for table MDS_ATTRIBUTES:  TEST_UNEXPECTED_INDEXES --> Test
that the table does not contain any unexpected indexes
    Completed index test for table MDS_ATTRIBUTES: TEST_UNEXPECTED_INDEXES --> Test
that the table does not contain any unexpected indexes +++ PASS
    Starting index test for table MDS_COMPONENTS:  TEST_UNEXPECTED_INDEXES --> Test
that the table does not contain any unexpected indexes
    Completed index test for table MDS_COMPONENTS: TEST_UNEXPECTED_INDEXES --> Test
that the table does not contain any unexpected indexes +++ PASS
    Starting index test for table MDS_DEPENDENCIES:  TEST_UNEXPECTED_INDEXES --> Test
that the table does not contain any unexpected indexes
    Completed index test for table MDS_DEPENDENCIES: TEST_UNEXPECTED_INDEXES --> Test
that the table does not contain any unexpected indexes +++ PASS
    Starting index test for table MDS_DEPL_LINEAGES:  TEST_UNEXPECTED_INDEXES -->
Test that the table does not contain any unexpected indexes
    Completed index test for table MDS_DEPL_LINEAGES: TEST_UNEXPECTED_INDEXES -->
Test that the table does not contain any unexpected indexes +++ PASS
    Starting index test for table MDS_LABELS:  TEST_UNEXPECTED_INDEXES --> Test that
the table does not contain any unexpected indexes
    Completed index test for table MDS_LABELS: TEST_UNEXPECTED_INDEXES --> Test that
the table does not contain any unexpected indexes +++ PASS
    Starting index test for table MDS_LARGE_ATTRIBUTES:  TEST_UNEXPECTED_INDEXES -->
Test that the table does not contain any unexpected indexes
    Completed index test for table MDS_LARGE_ATTRIBUTES: TEST_UNEXPECTED_INDEXES -->
Test that the table does not contain any unexpected indexes +++ PASS
    Starting index test for table MDS_METADATA_DOCS:  TEST_UNEXPECTED_INDEXES -->
Test that the table does not contain any unexpected indexes
    Completed index test for table MDS_METADATA_DOCS: TEST_UNEXPECTED_INDEXES -->
Test that the table does not contain any unexpected indexes +++ PASS
    Starting index test for table MDS_NAMESPACES:  TEST_UNEXPECTED_INDEXES --> Test
that the table does not contain any unexpected indexes
    Completed index test for table MDS_NAMESPACES: TEST_UNEXPECTED_INDEXES --> Test
that the table does not contain any unexpected indexes +++ PASS
    Starting index test for table MDS_PARTITIONS:  TEST_UNEXPECTED_INDEXES --> Test
that the table does not contain any unexpected indexes
    Completed index test for table MDS_PARTITIONS: TEST_UNEXPECTED_INDEXES --> Test
that the table does not contain any unexpected indexes +++ PASS
    Starting index test for table MDS_PATHS:  TEST_UNEXPECTED_INDEXES --> Test that
the table does not contain any unexpected indexes
    Completed index test for table MDS_PATHS: TEST_UNEXPECTED_INDEXES --> Test that
the table does not contain any unexpected indexes +++ PASS
    Starting index test for table MDS_PURGE_PATHS:  TEST_UNEXPECTED_INDEXES --> Test
that the table does not contain any unexpected indexes
    Completed index test for table MDS_PURGE_PATHS: TEST_UNEXPECTED_INDEXES --> Test
that the table does not contain any unexpected indexes +++ PASS
    Starting index test for table MDS_SANDBOXES:  TEST_UNEXPECTED_INDEXES --> Test
that the table does not contain any unexpected indexes
    Completed index test for table MDS_SANDBOXES: TEST_UNEXPECTED_INDEXES --> Test
that the table does not contain any unexpected indexes +++ PASS
    Starting index test for table MDS_STREAMED_DOCS:  TEST_UNEXPECTED_INDEXES -->
Test that the table does not contain any unexpected indexes
    Completed index test for table MDS_STREAMED_DOCS: TEST_UNEXPECTED_INDEXES -->
Test that the table does not contain any unexpected indexes +++ PASS
    Starting index test for table MDS_TRANSACTIONS:  TEST_UNEXPECTED_INDEXES --> Test
that the table does not contain any unexpected indexes
    Completed index test for table MDS_TRANSACTIONS: TEST_UNEXPECTED_INDEXES --> Test
that the table does not contain any unexpected indexes +++ PASS
    Starting index test for table MDS_TXN_LOCKS:  TEST_UNEXPECTED_INDEXES --> Test
that the table does not contain any unexpected indexes
    Completed index test for table MDS_TXN_LOCKS: TEST_UNEXPECTED_INDEXES --> Test
```

```
that the table does not contain any unexpected indexes +++ PASS
   Starting schema test:  TEST_UNEXPECTED_TRIGGERS  Test that the schema does not
contain any unexpected triggers
   Completed schema test: TEST_UNEXPECTED_TRIGGERS --> Test that the schema does not
contain any unexpected triggers +++ PASS
   Starting schema test:  TEST_UNEXPECTED_COLUMNS  Test that tables and views do not
contain any unexpected columns
   Completed schema test: TEST_UNEXPECTED_COLUMNS --> Test that tables and views do
not contain any unexpected columns +++ PASS
   Starting datatype test for table MDS_ATTRIBUTES:  TEST_COLUMN_DATATYPES_V2 -->
Test that all table columns have the proper datatypes
   Completed datatype test for table MDS_ATTRIBUTES: TEST_COLUMN_DATATYPES_V2 -->
Test that all table columns have the proper datatypes +++ PASS
   Starting datatype test for table MDS_COMPONENTS:  TEST_COLUMN_DATATYPES_V2 -->
Test that all table columns have the proper datatypes
   Completed datatype test for table MDS_COMPONENTS: TEST_COLUMN_DATATYPES_V2 -->
Test that all table columns have the proper datatypes +++ PASS
   Starting datatype test for table MDS_DEPENDENCIES:  TEST_COLUMN_DATATYPES_V2 -->
Test that all table columns have the proper datatypes
   Completed datatype test for table MDS_DEPENDENCIES: TEST_COLUMN_DATATYPES_V2 -->
Test that all table columns have the proper datatypes +++ PASS
   Starting datatype test for table MDS_DEPL_LINEAGES:  TEST_COLUMN_DATATYPES_V2 -->
Test that all table columns have the proper datatypes
   Completed datatype test for table MDS_DEPL_LINEAGES: TEST_COLUMN_DATATYPES_V2 -->
Test that all table columns have the proper datatypes +++ PASS
   Starting datatype test for table MDS_LABELS:  TEST_COLUMN_DATATYPES_V2 --> Test
that all table columns have the proper datatypes
   Completed datatype test for table MDS_LABELS: TEST_COLUMN_DATATYPES_V2 --> Test
that all table columns have the proper datatypes +++ PASS
   Starting datatype test for table MDS_LARGE_ATTRIBUTES:  TEST_COLUMN_DATATYPES_V2
--> Test that all table columns have the proper datatypes
   Completed datatype test for table MDS_LARGE_ATTRIBUTES: TEST_COLUMN_DATATYPES_V2
--> Test that all table columns have the proper datatypes +++ PASS
   Starting datatype test for table MDS_METADATA_DOCS:  TEST_COLUMN_DATATYPES_V2 -->
Test that all table columns have the proper datatypes
   Completed datatype test for table MDS_METADATA_DOCS: TEST_COLUMN_DATATYPES_V2 -->
Test that all table columns have the proper datatypes +++ PASS
   Starting datatype test for table MDS_NAMESPACES:  TEST_COLUMN_DATATYPES_V2 -->
Test that all table columns have the proper datatypes
   Completed datatype test for table MDS_NAMESPACES: TEST_COLUMN_DATATYPES_V2 -->
Test that all table columns have the proper datatypes +++ PASS
   Starting datatype test for table MDS_PARTITIONS:  TEST_COLUMN_DATATYPES_V2 -->
Test that all table columns have the proper datatypes
   Completed datatype test for table MDS_PARTITIONS: TEST_COLUMN_DATATYPES_V2 -->
Test that all table columns have the proper datatypes +++ PASS
   Starting datatype test for table MDS_PATHS:  TEST_COLUMN_DATATYPES_V2 --> Test
that all table columns have the proper datatypes
   Completed datatype test for table MDS_PATHS: TEST_COLUMN_DATATYPES_V2 --> Test
that all table columns have the proper datatypes +++ PASS
   Starting datatype test for table MDS_PURGE_PATHS:  TEST_COLUMN_DATATYPES_V2 -->
Test that all table columns have the proper datatypes
   Completed datatype test for table MDS_PURGE_PATHS: TEST_COLUMN_DATATYPES_V2 -->
Test that all table columns have the proper datatypes +++ PASS
   Starting datatype test for table MDS_SANDBOXES:  TEST_COLUMN_DATATYPES_V2 -->
Test that all table columns have the proper datatypes
   Completed datatype test for table MDS_SANDBOXES: TEST_COLUMN_DATATYPES_V2 -->
Test that all table columns have the proper datatypes +++ PASS
   Starting datatype test for table MDS_STREAMED_DOCS:  TEST_COLUMN_DATATYPES_V2 -->
Test that all table columns have the proper datatypes
   Completed datatype test for table MDS_STREAMED_DOCS: TEST_COLUMN_DATATYPES_V2 -->
Test that all table columns have the proper datatypes +++ PASS
```

```
   Starting datatype test for table MDS_TRANSACTIONS:  TEST_COLUMN_DATATYPES_V2 -->
Test that all table columns have the proper datatypes
   Completed datatype test for table MDS_TRANSACTIONS: TEST_COLUMN_DATATYPES_V2 -->
Test that all table columns have the proper datatypes +++ PASS
   Starting datatype test for table MDS_TXN_LOCKS:  TEST_COLUMN_DATATYPES_V2 -->
Test that all table columns have the proper datatypes
   Completed datatype test for table MDS_TXN_LOCKS: TEST_COLUMN_DATATYPES_V2 -->
Test that all table columns have the proper datatypes +++ PASS
   Starting permissions test for table MDS_ATTRIBUTES:  TEST_TABLE_GRANTS --> Test
that tables have the proper GRANT permissions
   Completed permissions test for table MDS_ATTRIBUTES: TEST_TABLE_GRANTS --> Test
that tables have the proper GRANT permissions +++ PASS
   Starting permissions test for table MDS_COMPONENTS:  TEST_TABLE_GRANTS --> Test
that tables have the proper GRANT permissions
   Completed permissions test for table MDS_COMPONENTS: TEST_TABLE_GRANTS --> Test
that tables have the proper GRANT permissions +++ PASS
   Starting permissions test for table MDS_DEPENDENCIES:  TEST_TABLE_GRANTS --> Test
that tables have the proper GRANT permissions
   Completed permissions test for table MDS_DEPENDENCIES: TEST_TABLE_GRANTS --> Test
that tables have the proper GRANT permissions +++ PASS
   Starting permissions test for table MDS_DEPL_LINEAGES:  TEST_TABLE_GRANTS -->
Test that tables have the proper GRANT permissions
   Completed permissions test for table MDS_DEPL_LINEAGES: TEST_TABLE_GRANTS -->
Test that tables have the proper GRANT permissions +++ PASS
   Starting permissions test for table MDS_LABELS:  TEST_TABLE_GRANTS --> Test that
tables have the proper GRANT permissions
   Completed permissions test for table MDS_LABELS: TEST_TABLE_GRANTS --> Test that
tables have the proper GRANT permissions +++ PASS
   Starting permissions test for table MDS_LARGE_ATTRIBUTES:  TEST_TABLE_GRANTS -->
Test that tables have the proper GRANT permissions
   Completed permissions test for table MDS_LARGE_ATTRIBUTES: TEST_TABLE_GRANTS -->
Test that tables have the proper GRANT permissions +++ PASS
   Starting permissions test for table MDS_METADATA_DOCS:  TEST_TABLE_GRANTS -->
Test that tables have the proper GRANT permissions
   Completed permissions test for table MDS_METADATA_DOCS: TEST_TABLE_GRANTS -->
Test that tables have the proper GRANT permissions +++ PASS
   Starting permissions test for table MDS_NAMESPACES:  TEST_TABLE_GRANTS --> Test
that tables have the proper GRANT permissions
   Completed permissions test for table MDS_NAMESPACES: TEST_TABLE_GRANTS --> Test
that tables have the proper GRANT permissions +++ PASS
   Starting permissions test for table MDS_PARTITIONS:  TEST_TABLE_GRANTS --> Test
that tables have the proper GRANT permissions
   Completed permissions test for table MDS_PARTITIONS: TEST_TABLE_GRANTS --> Test
that tables have the proper GRANT permissions +++ PASS
   Starting permissions test for table MDS_PATHS:  TEST_TABLE_GRANTS --> Test that
tables have the proper GRANT permissions
   Completed permissions test for table MDS_PATHS: TEST_TABLE_GRANTS --> Test that
tables have the proper GRANT permissions +++ PASS
   Starting permissions test for table MDS_PURGE_PATHS:  TEST_TABLE_GRANTS --> Test
that tables have the proper GRANT permissions
   Completed permissions test for table MDS_PURGE_PATHS: TEST_TABLE_GRANTS --> Test
that tables have the proper GRANT permissions +++ PASS
   Starting permissions test for table MDS_SANDBOXES:  TEST_TABLE_GRANTS --> Test
that tables have the proper GRANT permissions
   Completed permissions test for table MDS_SANDBOXES: TEST_TABLE_GRANTS --> Test
that tables have the proper GRANT permissions +++ PASS
   Starting permissions test for table MDS_STREAMED_DOCS:  TEST_TABLE_GRANTS -->
Test that tables have the proper GRANT permissions
   Completed permissions test for table MDS_STREAMED_DOCS: TEST_TABLE_GRANTS -->
Test that tables have the proper GRANT permissions +++ PASS
   Starting permissions test for table MDS_TRANSACTIONS:  TEST_TABLE_GRANTS --> Test
```

```
   that tables have the proper GRANT permissions
      Completed permissions test for table MDS_TRANSACTIONS: TEST_TABLE_GRANTS --> Test
   that tables have the proper GRANT permissions +++ PASS
      Starting permissions test for table MDS_TXN_LOCKS:  TEST_TABLE_GRANTS --> Test
   that tables have the proper GRANT permissions
      Completed permissions test for table MDS_TXN_LOCKS: TEST_TABLE_GRANTS --> Test
   that tables have the proper GRANT permissions +++ PASS
      Starting permissions test:  TEST_DBA_TABLE_GRANTS  Test that DBA user has
   privilege to view all user tables
      Completed permissions test: TEST_DBA_TABLE_GRANTS --> Test that DBA user has
   privilege to view all user tables +++ PASS
      Starting schema test:  TEST_ENOUGH_TABLESPACE  Test that the schema tablespaces
   automatically extend if full
      Completed schema test: TEST_ENOUGH_TABLESPACE --> Test that the schema
   tablespaces automatically extend if full +++ PASS
      Starting schema test:  TEST_USER_TABLESPACE_QUOTA  Test that tablespace quota for
   this user is sufficient to perform the upgrade
      Completed schema test: TEST_USER_TABLESPACE_QUOTA --> Test that tablespace quota
   for this user is sufficient to perform the upgrade +++ PASS
      Starting schema test:  TEST_ONLINE_TABLESPACE  Test that schema tablespaces are
   online
      Completed schema test: TEST_ONLINE_TABLESPACE --> Test that schema tablespaces
   are online +++ PASS
      Starting schema test:  TEST_DATABASE_VERSION  Test that the database server
   version number is supported for upgrade
        INFO    Database product version: Oracle Database 11g Enterprise Edition Release
   11.2.0.3.0 - 64bit Production
   With the Partitioning, OLAP, Data Mining and Real Application Testing options
      Completed schema test: TEST_DATABASE_VERSION --> Test that the database server
   version number is supported for upgrade +++ PASS
      Finished readiness check of Oracle Metadata Services with status: FAILURE.

   Common Infrastructure Services
      Starting readiness check of Common Infrastructure Services.
        Schema User Name: DEV1212_STB
        Database Type: Oracle Database
        Database Connect String: machinename@yourcompany.com
        VERSION Schema STB is currently at version 12.1.2.0.0.  Readiness checks will
   now be performed.
      Starting schema test:  TEST_REQUIRED_TABLES  Test that the schema contains all
   the required tables
      Completed schema test: TEST_REQUIRED_TABLES --> Test that the schema contains all
   the required tables +++ PASS
      Completed schema test: TEST_REQUIRED_TABLES --> Test that the schema contains all
   the required tables +++ PASS
      Starting schema test:  TEST_UNEXPECTED_TABLES  Test that the schema does not
   contain any unexpected tables
      Completed schema test: TEST_UNEXPECTED_TABLES --> Test that the schema does not
   contain any unexpected tables +++ PASS
      Starting schema test:  TEST_REQUIRED_VIEWS  Test that the schema contains all the
   required database views
      Completed schema test: TEST_REQUIRED_VIEWS --> Test that the schema contains all
   the required database views +++ PASS
      Starting schema test:  TEST_MISSING_COLUMNS  Test that tables and views are not
   missing any required columns
      Completed schema test: TEST_MISSING_COLUMNS --> Test that tables and views are
   not missing any required columns +++ PASS
      Starting schema test:  TEST_UNEXPECTED_COLUMNS  Test that tables and views do not
   contain any unexpected columns
      Completed schema test: TEST_UNEXPECTED_COLUMNS --> Test that tables and views do
   not contain any unexpected columns +++ PASS
```

```
    Starting schema test:  TEST_UNEXPECTED_PROCEDURES  Test that the schema does not
contain any unexpected stored procedures
    Completed schema test: TEST_UNEXPECTED_PROCEDURES --> Test that the schema does
not contain any unexpected stored procedures +++ PASS
    Starting permissions test:  TEST_DBA_TABLE_GRANTS  Test that DBA user has
privilege to view all user tables
    Starting permissions test:  TEST_DBA_TABLE_GRANTS  Test that DBA user has
privilege to view all user tables
    Completed permissions test: TEST_DBA_TABLE_GRANTS --> Test that DBA user has
privilege to view all user tables +++ PASS
    Completed permissions test: TEST_DBA_TABLE_GRANTS --> Test that DBA user has
privilege to view all user tables +++ PASS
    Starting permissions test for table COMPONENT_SCHEMA_INFO:  TEST_TABLE_GRANTS -->
Test that tables have the proper GRANT permissions
    Completed permissions test for table TEST_TABLE_GRANTS: Test that tables have the
proper GRANT permissions --> PASS +++ {3}
    Starting permissions test for table SERVICETABLE:  TEST_TABLE_GRANTS --> Test
that tables have the proper GRANT permissions
    Completed permissions test for table TEST_TABLE_GRANTS: Test that tables have the
proper GRANT permissions --> PASS +++ {3}
    Starting datatype test for table COMPONENT_SCHEMA_INFO:  TEST_COLUMN_DATATYPES_V2
--> Test that all table columns have the proper datatypes
    Completed datatype test for table COMPONENT_SCHEMA_INFO: TEST_COLUMN_DATATYPES_V2
--> Test that all table columns have the proper datatypes +++ PASS
    Starting datatype test for table SERVICETABLE:  TEST_COLUMN_DATATYPES_V2 --> Test
that all table columns have the proper datatypes
    Completed datatype test for table SERVICETABLE: TEST_COLUMN_DATATYPES_V2 --> Test
that all table columns have the proper datatypes +++ PASS
    Starting index test for table COMPONENT_SCHEMA_INFO:  TEST_REQUIRED_INDEXES -->
Test that the table contains all the required indexes
    Completed index test for table COMPONENT_SCHEMA_INFO: TEST_REQUIRED_INDEXES -->
Test that the table contains all the required indexes +++ PASS
    Starting index test for table COMPONENT_SCHEMA_INFO:  TEST_UNEXPECTED_INDEXES -->
Test that the table does not contain any unexpected indexes
    Completed index test for table COMPONENT_SCHEMA_INFO: TEST_UNEXPECTED_INDEXES -->
Test that the table does not contain any unexpected indexes +++ PASS
    Starting index test for table SERVICETABLE:  TEST_REQUIRED_INDEXES --> Test that
the table contains all the required indexes
    Completed index test for table SERVICETABLE: TEST_REQUIRED_INDEXES --> Test that
the table contains all the required indexes +++ PASS
    Starting index test for table SERVICETABLE:  TEST_UNEXPECTED_INDEXES --> Test
that the table does not contain any unexpected indexes
    Completed index test for table SERVICETABLE: TEST_UNEXPECTED_INDEXES --> Test
that the table does not contain any unexpected indexes +++ PASS
    Starting schema test:  TEST_UNEXPECTED_TRIGGERS  Test that the schema does not
contain any unexpected triggers
    Completed schema test: TEST_UNEXPECTED_TRIGGERS --> Test that the schema does not
contain any unexpected triggers +++ PASS
    Starting schema test:  TEST_ENOUGH_TABLESPACE  Test that the schema tablespaces
automatically extend if full
    Starting schema test:  TEST_ENOUGH_TABLESPACE  Test that the schema tablespaces
automatically extend if full
    Completed schema test: TEST_ENOUGH_TABLESPACE --> Test that the schema
tablespaces automatically extend if full +++ PASS
    Completed schema test: TEST_ENOUGH_TABLESPACE --> Test that the schema
tablespaces automatically extend if full +++ PASS
    Starting schema test:  TEST_ONLINE_TABLESPACE  Test that schema tablespaces are
online
    Starting schema test:  TEST_ONLINE_TABLESPACE  Test that schema tablespaces are
online
    Completed schema test: TEST_ONLINE_TABLESPACE --> Test that schema tablespaces
```

```
are online +++ PASS
    Completed schema test: TEST_ONLINE_TABLESPACE --> Test that schema tablespaces
are online +++ PASS
    Starting schema test:  TEST_USER_TABLESPACE_QUOTA  Test that tablespace quota for
this user is sufficient to perform the upgrade
    Starting schema test:  TEST_USER_TABLESPACE_QUOTA  Test that tablespace quota for
this user is sufficient to perform the upgrade
    Completed schema test: TEST_USER_TABLESPACE_QUOTA --> Test that tablespace quota
for this user is sufficient to perform the upgrade +++ PASS
    Completed schema test: TEST_USER_TABLESPACE_QUOTA --> Test that tablespace quota
for this user is sufficient to perform the upgrade +++ PASS
    Starting schema test:  TEST_DATABASE_VERSION  Test that the database server
version number is supported for upgrade
    Starting schema test:  TEST_DATABASE_VERSION  Test that the database server
version number is supported for upgrade
     INFO   Database product version: Oracle Database 11g Enterprise Edition Release
11.2.0.3.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
    Completed schema test: TEST_DATABASE_VERSION --> Test that the database server
version number is supported for upgrade +++ PASS
    Completed schema test: TEST_DATABASE_VERSION --> Test that the database server
version number is supported for upgrade +++ PASS
    Finished readiness check of Common Infrastructure Services with status: SUCCESS.

Finished readiness check of components.
```

3

Troubleshooting Your Upgrade

This topic describes how to resolve common problems and issues that may occur while you are using the Upgrade Assistant to upgrade your Oracle Fusion Middleware deployment to this release.

This topic contains the following topics:

Troubleshooting an Upgrade

Reviewing Log Files

Investigating Examination Failures

Investigating Upgrade Failures

Resolving Common Upgrade Assistant Errors

Attempting to Upgrade an Unsupported Domain

Restarting the Upgrade Assistant After a Failure

3.1 Troubleshooting an Upgrade

If errors occur while you are running the Upgrade Assistant, use the following steps to troubleshoot the problem:

If you get any errors during Examination phase, and no components or schemas have been upgraded yet, run the following command to perform a thorough readiness check. The types of checks performed by the readiness check are more thorough than Examination phase of upgrade. Examination phase is read-only and you can rerun it.

On Unix operating machine: `./ua -readiness`

On Windows operating machine: `ua.bat -readiness`

Errors that occur during the **Upgrade phase**, however, require you to restore your environment from your backup copies, correct the errors and then restart the upgrade process from the beginning.

Your corrective course of action will depend on the phase in which the error occurred.

1. Locate and open the Upgrade Assistant log file with a text editor:

   For the location of the log file, see Reviewing Log Files.

2. To verify the correct version of Upgrade Assistant jar file run the following command:

   `unzip -p ua.jar META-INF/MANIFEST.MF`

   You can run this command only after you change directory to the `jlib` directory:.

   `cd ORACLE_HOME/oracle_common/upgrade/jlib`

> **Note:** The output of the `unzip -p ua.jar META-INF/MANIFEST.MF` command identifies the development label that was used to build the Upgrade Assistant, and that information identifies the date and version of the Upgrade Assistant software that was run. If you raise a Service Request you need to provide this information to Oracle.

3. Locate any error messages that are identified by number; for example, `UPGAST-00091`.

4. Look up the error in the Oracle Fusion Middleware Error Messages guide..

   The description of the error in the should include a description of the cause of the error, as well as the action you should take to resolve the error.

5. Based on whether or not you can locate an error message and the error message description, do the following:

   • If, by reviewing the log files and error message descriptions, you are able to identify a solution to the upgrade failure, you can implement your solution and then re-start the Upgrade Assistant and perform the upgrade again.

     When you re-run the Upgrade Assistant, any components that were upgraded successfully during the previous run will not be affected. However, the Upgrade Assistant will attempt to upgrade any components that were not upgraded successfully during a previous run of the utility.

   • Contact Oracle Support for any errors that are not documented or that cannot be resolved by following documented actions. Note that some errors that occur will require the repository to be restored from backup, the problem to be resolved, and another upgrade to be run. Note that Oracle Support requires both the `UA.log` file and if present, the `UA.out` file, to be provided for troubleshooting purposes. Provide complete logs and not just excerpts of those files.

3.2 Reviewing Log Files

Should any failures occur when running Upgrade Assistant, log files will be needed to help diagnose and correct the problem; do not delete them. When running the Upgrade Assistant, you can alter the contents of your log files by specifying a different `-logLevel` from the command line. The default value is `-logLevel NOTIFICATION`. You can alter the location of your log files using the `-logDir` parameter. You can obtain a more detailed logging information by running the Upgrade Assistant with `-logLevel TRACE` parameter. However, in some situations, this can cause the log file to become very large.

> **Note:** TRACE messages are not included in the Upgrade Assistant Log File Viewer. To view TRACE messages you must use another tool of your choice

> **Tip:** To expedite the review process, search for the word "ERROR".

For more information on understanding error messages in your log files, see Resolving Common Upgrade Assistant Errors.

Log files are stored in the following default directory:

On UNIX operating systems:

ORACLE_HOME/oracle_common/upgrade/logs/ua<timestamp>.log

On Windows operating systems:

ORACLE_HOME\oracle_common\upgrade\logs\ua*<timestamp>*.log

Some components will create a second log file called ua*<timestamp>*.out, also in the same location.

Where, *timestamp* states the actual date and time when the Upgrade Assistant was run.

For database schema upgrades of certain components, there can also be an output (.out) file that will contain the screen output of commands that were run in a shell process or as PL/SQL scripts. You can locate these output files in the same default directory.

In the event that there are questions or issues about an upgrade failure that cannot be resolved with the information in this guide, it will be important to retain the log files. If you must raise a service request, you must upload the entire Upgrade Assistant .log file and the .out files (if available) along with the service request.

3.3 Investigating Examination Failures

To determine the cause of an examination failure:

1. Note the name of the failed component in the Upgrade Assistant dialog or command-line output.

2. Open the following Upgrade Assistant log file.

   For the location of the log file, see Reviewing Log Files.

3. In the log file, search for the message Starting to examine *component_name*.

To complete the upgrade, resolve the issues and then launch the Upgrade Assistant again, or, if possible, click **Back** to return to a previous screen and make the necessary changes.

> **Note:** The readiness check feature performs checks to a far greater level of detail than the Examination phase. If examine fails you should run the Upgrade Assistant with the -readiness parameter and make sure the report does not show any test failures.

Issues detected during the Examination phase can be resolved without restoring from backup. However, if you attempt to resolve an examination error in a way that changes the state of the system, you should take an another-point-in-time backup and retain all such backups. Also, you need to restore the entire system to that point-in-time before anything was changed or any upgrade operations were attempted. Issues detected during the Upgrade phase, however, cannot be resolved. The pre-upgrade environment must be restored from backup to a point-in-time before any of the upgrade operations was attempted. If a partially upgrade system encounters upgrade errors later, the entire system must still be restored.

3.4 Investigating Upgrade Failures

To determine the cause of an upgrade failure:

1. Note the name of the failed component in the Upgrade Assistant dialog or command-line output.

2. Open the Upgrade log file:

    For the location of the log file, see Reviewing Log Files.

3. Search for the message `Starting to upgrade component_name`.

To complete the upgrade, restore the entire environment using your pre-upgrade backup, to a point in time before any upgrade operations were attempted, resolve the issues and then launch the Upgrade Assistant again. You will have to start the upgrade process from the beginning to ensure a successful (complete) upgrade.

> **Note:** You should be backing up all databases with RMAN and be able to do a point-in-time recovery from those backups. If the Fusion Middleware repository for your domains spans multiple Oracle Database server, you must restore from each of those backups.

3.5 Resolving Common Upgrade Assistant Errors

If errors occur while you are running the Upgrade Assistant, you must correct the conditions that caused them before you try the upgrade again. The following sections provide some common errors that can occur.

This section provides descriptions of the most common upgrade errors. For a complete list of Fusion Middleware errors, see the *Error Messages* guide.

Ensuring there is sufficient disk space

Resolving Database Connection Problems When Upgrading Schemas

Setting the DISPLAY Environment Variable

3.5.1 Ensuring there is sufficient disk space

If an upgrade fails due to the database server running out of disk space, you must restore the database server environment from backups, add sufficient disk space or remove unwanted files (such as temp or trace files) from the database server, and then retry the upgrade.

NOTE: Once a database schema upgrade has failed due to this class of error, you cannot simply add more disk space and retry the upgrade. The schemas have been left in an inconsistent state and may have been marked "INVALID". You cannot recover from this error without restoring the original database state from backups.

The following examples show some insufficient disk space errors you may encounter:

ORA-01658: unable to create INITIAL extent for segment in tablespace

Cause: The existing schema tablespace does not have sufficient space to complete the upgrade.

Action: Make sure that the tablespace has sufficient room (space) for a successful upgrade. Oracle recommends that you add more data files to the existing database tablespaces, otherwise the upgrade will fail.

ORA-01114: IO error writing block to file *<block number>*

Cause: The device on which the file resides is probably offline. If the file is a temporary file, then it is also possible that the device has run out of space. This could happen because disk space of temporary files is not necessarily allocated at file creation time.

Action: Restore access to the device or remove unnecessary files to free up space.

ORA-09945: Unable to initialize the audit trail file

Cause: The system is unable to write header information to the file being used as the audit trail. The audit_trail_dest or audit trail destination is full for generation of audit file.

Action: Free up space and retry the operation.

3.5.2 Resolving Database Connection Problems When Upgrading Schemas

If you have trouble connecting to a database when using the Upgrade Assistant to upgrade a component schema, try connecting to the database using another tool, such as SQL*Plus. This will help you troubleshoot the problem by verifying that the database is up and running and available on the network.

3.5.3 Setting the DISPLAY Environment Variable

When running Upgrade Assistant in GUI mode, you must set the DISPLAY variable properly or you may receive errors such as:

Xlib: connection to ":1.0" refused by server

Xlib: No protocol specified

Cause: These errors indicate that the `DISPLAY` variable is not set up properly to allow a GUI to be displayed to the screen.

Action: Set the `DISPLAY` environment variable to the system name or IP address of your local workstation, and re-run Upgrade Assistant.

If you continue to receive these errors after setting the `DISPLAY` variable, try launching another GUI tool, such as vncconfig. If you see the same errors, your `DISPLAY` environment variable may not be set correctly.

3.6 Attempting to Upgrade an Unsupported Domain

If you receive an error stating that the specified domain cannot be upgraded, you must first upgrade the domain to a supported version. **Do not attempt to upgrade schemas or domain configurations in an unsupported domain.**

3.7 Restarting the Upgrade Assistant After a Failure

If the Upgrade Assistant fails during the upgrade phase or only partially upgrades your components, try to resolve the and then follow these steps:

1. Restore your backed-up 11*g* or 12*c* environment.

2. Start the Upgrade Assistant in GUI or command-line mode.

If you continue to experience upgrade failures, consider setting the `-logLevel` to `TRACE` so that more information will be logged. This will be useful when troubleshooting a failed upgrade, but be sure to reset the `-logLevel` to `NOTIFICATION` after the issue has been resolved to avoid performance issues.

A

Understanding the Upgrade Assistant Screens

The screens for the Oracle Fusion Middleware Upgrade Assistant vary depending upon the type of Oracle Fusion Middleware software you are upgrading.

The screens listed in this topic are not necessarily in sequential order. Depending on the options you select or the schemas and configurations in your domain, you may not see all of the screens. See About Using the Upgrade Assistant for general information on the screens that are used for each upgrade type.

The Upgrade Assistant can be run in two modes: upgrade and readiness (pre-upgrade.) The screens for each mode are described in the following sections:

Upgrade Assistant Screens

Readiness Check Screens
This section describes the screens that are presented when running the Upgrade Assistant in -readiness mode.

A.1 Upgrade Assistant Screens

This section describes all of the Upgrade Assistant screens.

> **Note:**
>
> The screens you will see while using the Oracle Fusion Middleware Upgrade Assistant vary depending upon the type of Oracle Fusion Middleware software you are upgrading. Not all screens will be shown to you.

Welcome

Schemas

All Schemas Used by Domain

All Configurations Used by a Domain

Standalone Components

Available Components

All Schemas Component List

WebLogic Server Component List

Prerequisites

Edition-Based Redefinition (EBR) Database Upgrade

A.1.1 Welcome

The Oracle Fusion Middleware Upgrade Assistant is used to upgrade component schemas, component configurations, and standalone system component configurations from Fusion Middleware 11*g* and 12*c* releases to the latest Fusion Middleware 12*c* release.
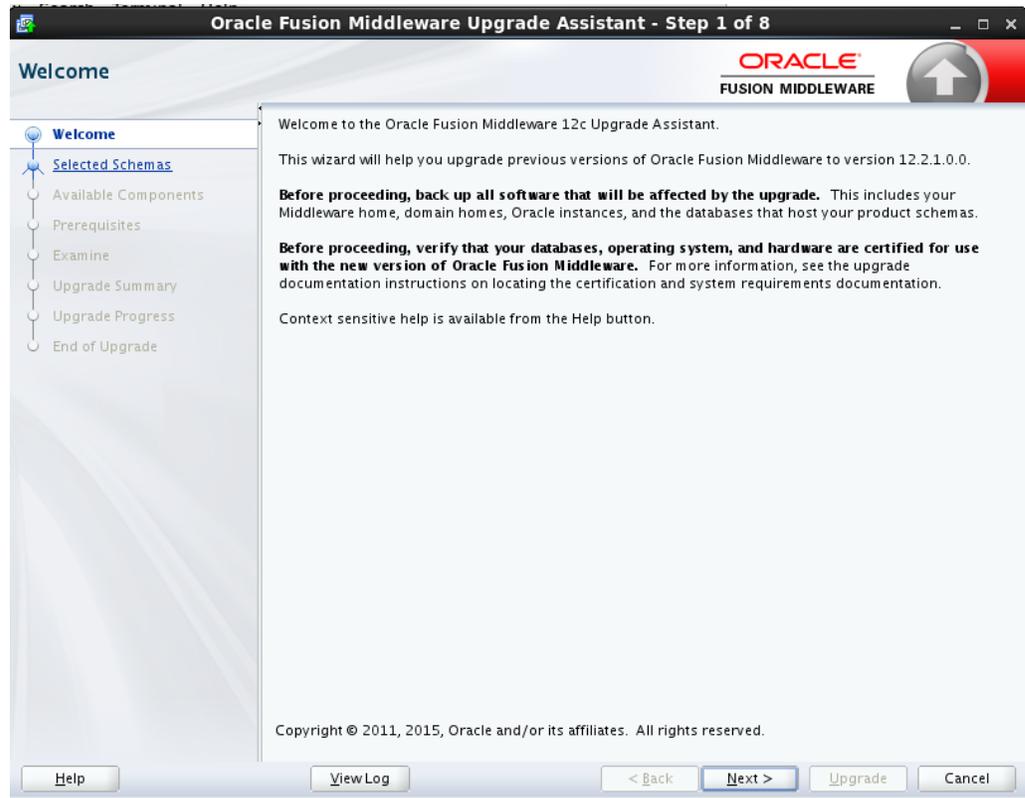
A.1.2 Schemas

Select **Individually Selected Schemas** to upgrade selected schemas for your installed components. The Upgrade Assistant will identify components that are candidates for a schema upgrade and then you can select which schemas to include in the upgrade

CAUTION: Upgrade only those schemas that will be used to support your 12.2.1.0.0 components. Do not upgrade schemas that are currently being used to support 11g or 12*c* components that are not included in the Oracle Fusion Middleware 12.2.1 release.

Figure A-1 Individually Selected Schemas
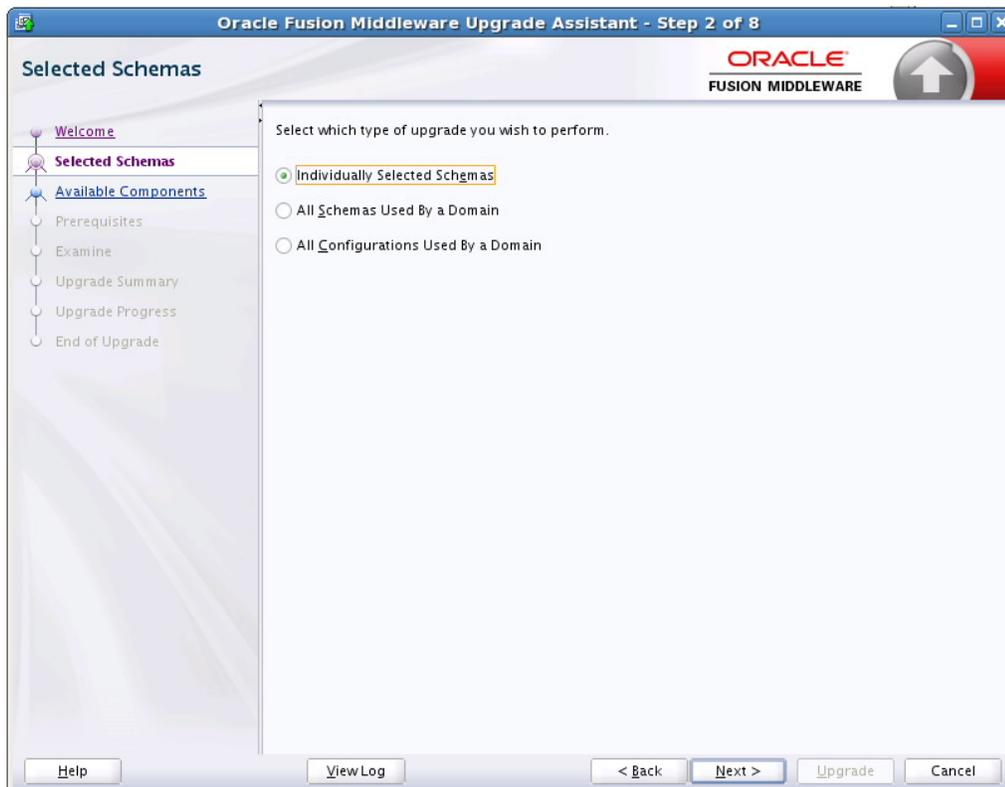
A.1.3 All Schemas Used by Domain

As of release 12.2.1, the Oracle Fusion Middleware Upgrade Assistant (UA) provides an option for upgrading all schemas used by a specified domain (sometimes referred to as Domain Assisted Schema Upgrade or DASU). When you select **All Schemas Used By a Domain**, the Upgrade Assistant discovers and selects all components that have schemas available to upgrade. In addition, where possible, the Upgrade Assistant pre-populates the connection information on schema input screens.

Also, you must browse and provide the 11*g* domain in the **Domain Directory** field.
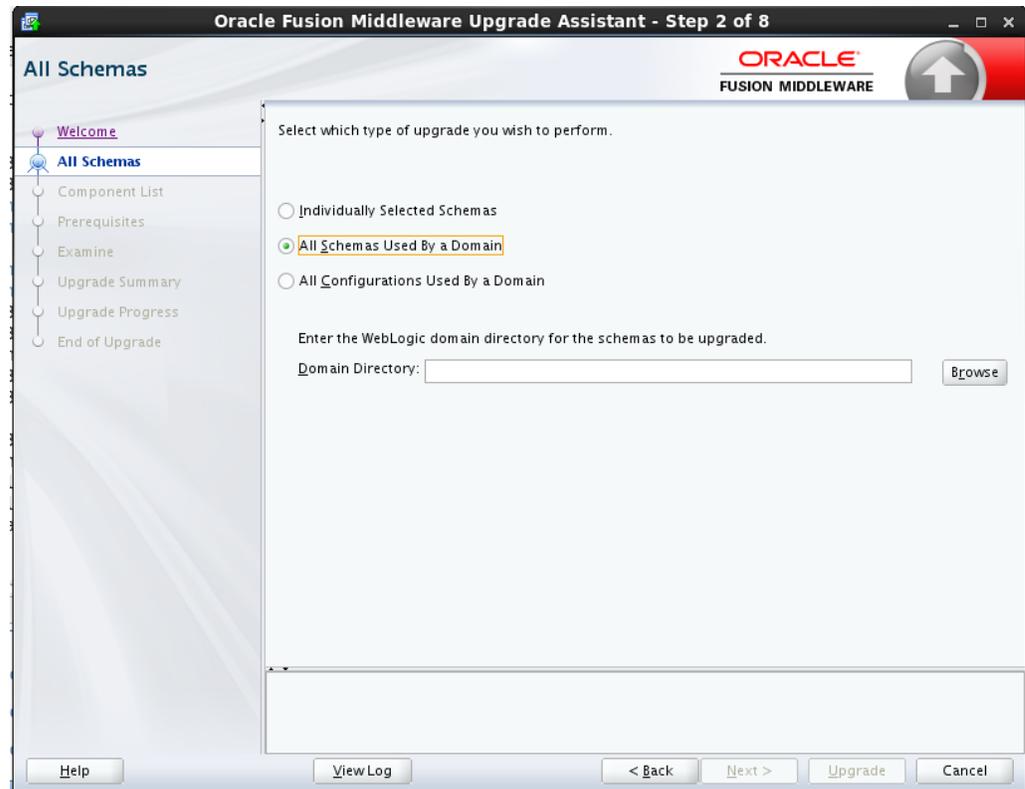
A.1.4 All Configurations Used by a Domain

Select the **All Configurations Used by a Domain** option to upgrade component configurations for a managed WebLogic Server domain. Click **Browse** and use the navigation tree to select a valid domain directory. A domain directory contains a `config` directory, which contains a `config.xml` file.
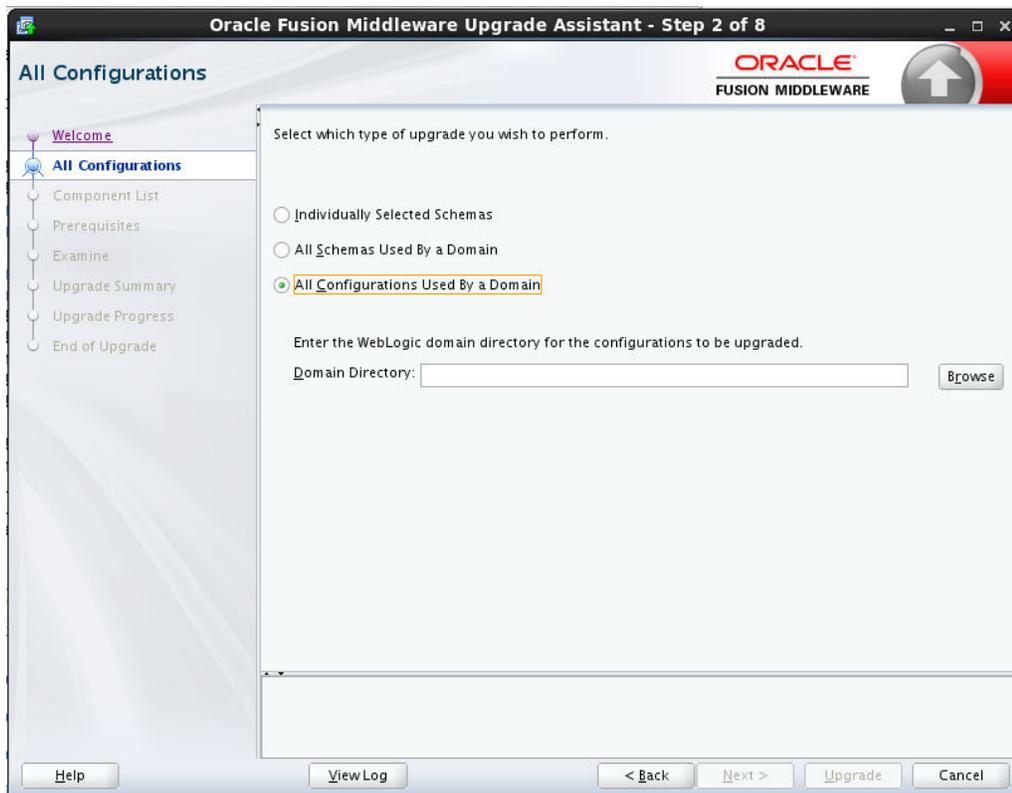
A.1.5 Standalone Components

Select the **Standalone System Component Configurations** option when you will be upgrading a standalone system component, such as Oracle HTTP Server (OHS).
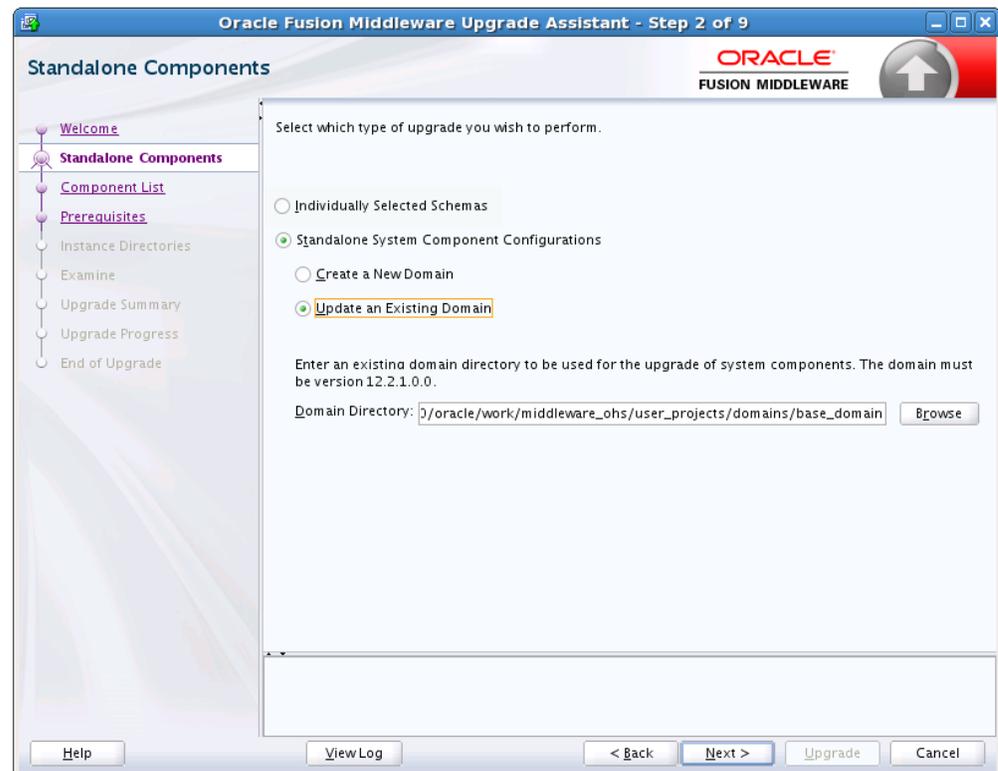
You will be prompted to select one of the following:

Option	Description
Create a New Domain	Standalone system components will have a separate standalone domain in 12*c*. A standalone domain is a container for system components, such as Oracle HTTP Server. It has a directory structure similar to an Oracle WebLogic domain, but it does not contain an Administration Server or Managed Servers. It can contain one or more instances of system components of the same type, such as Oracle HTTP Server, or a mix of types.
	Management tools, such as the Configuration Wizard, pack and unpack, WLST, and Node Manager can operate on standalone domains.
Update an Existing Domain	Once a standalone domain has been created for a system component, you can select this option to extend that domain for another standalone system component. This option is also used when upgrading from 12.1.2 or 12.1.3.
	You must provide the location of the existing 12c domain.

A.1.6 Available Components

If you selected the **Individually Selected Schemas** option in the previous screen to select individual schemas to be upgraded - instead of upgrading all schemas used by the domain - this screen displays the components with schemas that can be upgraded. If you select a component that requires another schema, the Upgrade Assistant will automatically select those schemas for you.

Figure A-2   List of Components with Schemas Available for Upgrade
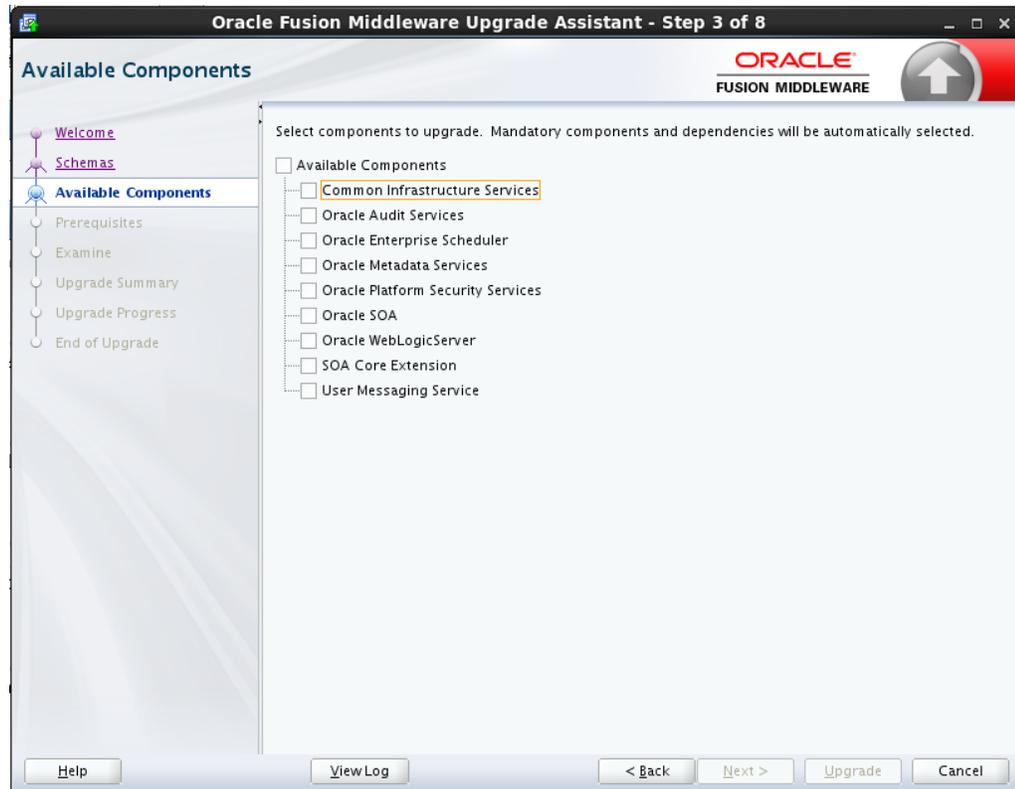
A.1.7 All Schemas Component List

If you selected **All Schemas Used by the Domain**, then this screen provides a list of schemas that will be included in the WebLogic domain upgrade. The names of the components are provided along with the schemas located within the domain.

Review the list carefully to verify that the correct schemas will be upgraded. If you do not see the components or schemas you want to upgrade, you may have selected the wrong domain. Use the **Back** button to specify a different domain.

If there are components or schemas listed that you do not want included, navigate back to the All Schemas screen and select **Individually Selected Schemas** instead of **All Schemas Used by the Domain**. The **Individually Selected Schemas** option allows you to select only those schemas you want included in the upgrade.

Figure A-3   Component List for All Schema Used by the Domain Upgrade
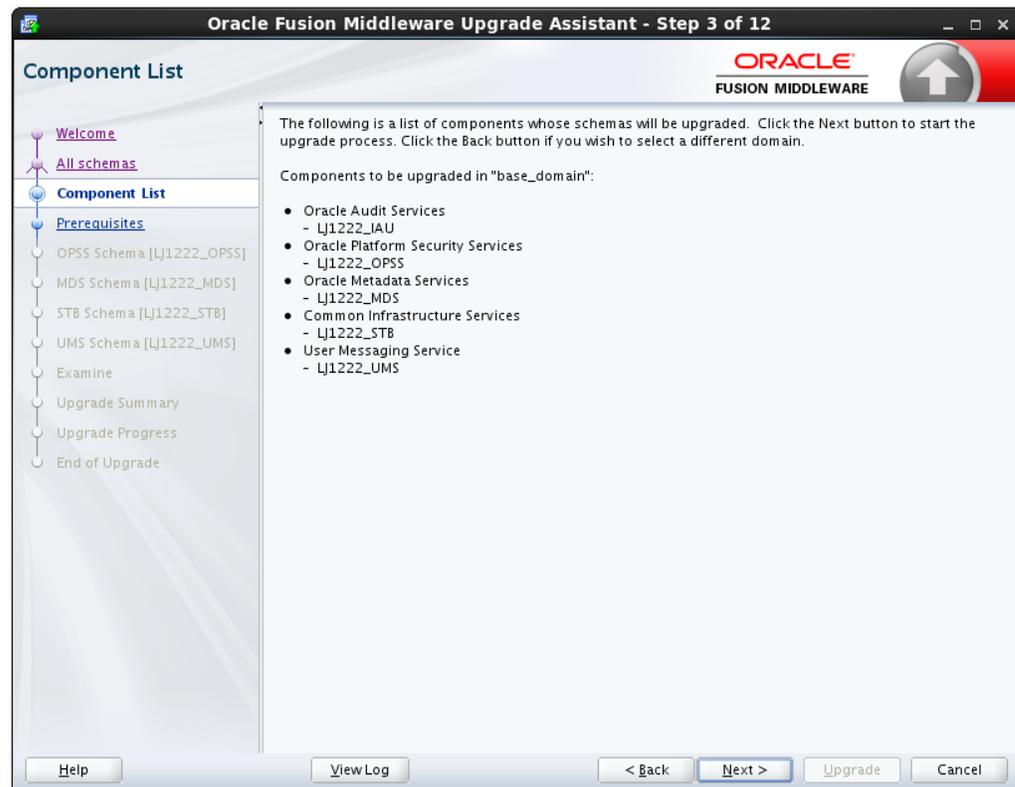
A.1.8 WebLogic Server Component List

When **All Configurations Used by a Domain** is selected for upgrade, the domain's components are listed on this read-only screen. Review the list of components before you proceed.

Figure A-4   Component List for WebLogic Component Configurations

A.1.9 Prerequisites

This screen requires you to acknowledge that all prerequisites have been met before you continue with the upgrade. You must check the boxes before you can continue.

> **Warning:**
>
> The Upgrade Assistant will not verify that the prerequisites have been met.

Figure A-5   Prerequisites
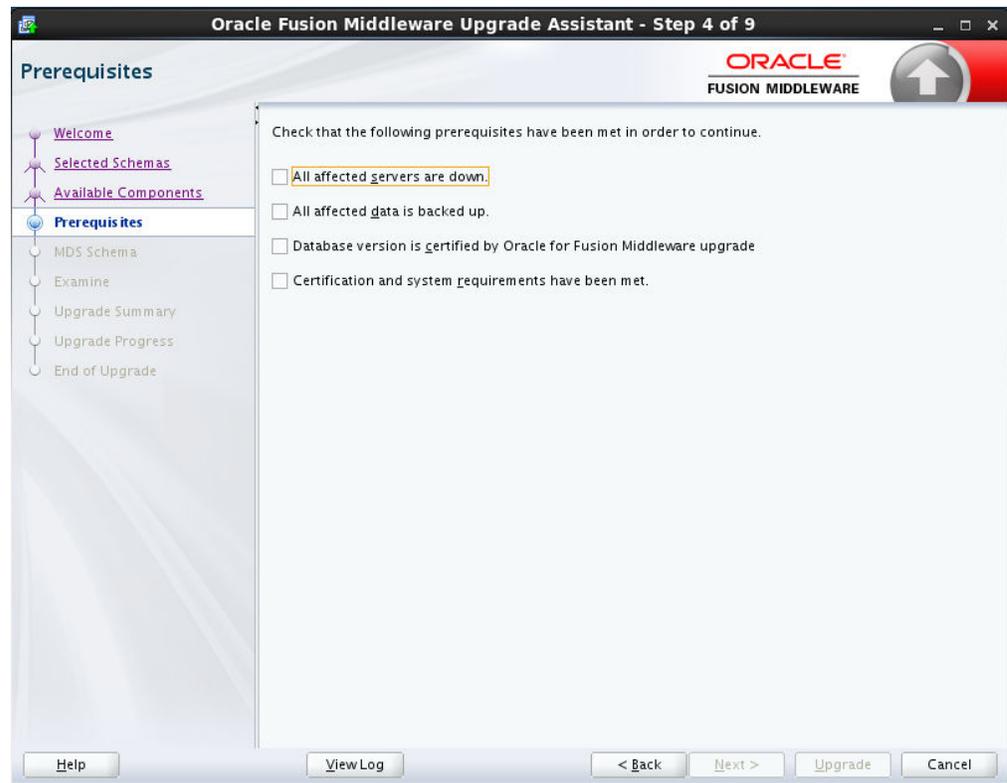
A.1.10 Edition-Based Redefinition (EBR) Database Upgrade

Use this screen to select the child edition from edition drop down list for edition-based redefinition databases. You must create the child edition before starting the upgrade.

For more information on creating an edition on the server for edition-based redefinition, see "Creating an Edition on the Server for Edition-Based Redefinition" in *Planning an Upgrade of Oracle Fusion Middleware*.

Figure A-6   EBR Database Upgrade

A.1.11 Schema Credentials Screen

Use this screen to enter information required to connect to the selected schema and the database that hosts the schema. If the schema that is to be upgraded was created by RCU in a prior Fusion Middleware release then you will see a drop-down menu listing the possible schema names as shown below. Click **Connect** to connect to the database then select the schema to be upgraded.

NOTE: Most schemas will have this information pre-populated. If, however, the Upgrade Assistant is unable to detect the connection details, then they must be entered manually as shown below.

Figure A-7   Select Schemas - Dropdown Menu

Figure A-8   Manually Entered Schema Data

The following table describes the elements that appear on this screen.

Element	Description
Database Type	Select the database type from the drop-down menu. The types of databases available in the menu varies, depending on the schema you are about to upgrade.
	The database type chosen for upgrade must be identical to the database type that was selected when RCU originally created the schema.
	If you select Oracle Edition-Based Redefinition (EBR) as the database type, the schema that you are upgrading also must have been created by RCU using the EBR database type. In particular, Upgrade Assistant never converts schemas from one database type to another.
Database Connect String	Enter the location of the database.
	For example, if you are selecting an Oracle database, the following URL format could be used:
	`host:port/db_service_name`
	If you are using a Microsoft SQL Server or IBM DB2 database, then select the database type from the drop-down menu, and review the text below the field, which provides the syntax required for each database type.
	NOTE: The Upgrade Assistant accepts other valid forms of connection strings. For example, the Oracle Database TNS style connection string may also be used.
DBA User Name	Enter the database user name used to connect to the database.
	NOTE: The DBA user must have sufficient privileges to run the Upgrade Assistant, but the user does NOT have to have SYS/SYSDBA privileges. A non-sysdba user can now be used.
	On certain database platforms user names are case sensitive, and the DBA user name might consist of lower case letters. The Upgrade Assistant connects to the name the user enters and does not convert the user name to upper case.
DBA Password	Enter the password associated with the specified DBA database user.
Schema User Name	Select the schema user name from the drop-down list or enter the user name of the schema, for example, **DEV_MDS**.
	Note that all Oracle Fusion Middleware schema names consist solely of upper case characters on all database platforms. Also, all schema names are stored as upper case in the `schema_version_registry` table. If you type lower case letters in the Schema User Name field, the Upgrade Assistant converts the name to upper case.
	For WebLogic Server domain, UMS, and Veridata schemas you need to manually enter the 11*g* schema user name and password.
Schema Password	Enter the password associated with the specified schema user name.

Element	Description
Edition Name	When Oracle Database enabled for edition-based redefinition is selected as the database type, you must specify the existing edition name.
	NOTE: Before upgrading an EBR-enabled schema from Fusion Middleware 11g release or from a previous 12*c* release, you must first connect to the database server and create an edition on the database server for 12c (12.2.1). The new edition for 12.2.1 must be a child of your 11.1.1.6.0, 11.1.1.7.0, 12.1.2, or 12.1.3 edition.
	For more information on creating an edition on the server for edition-based redefinition, see "Creating an Edition on the Server for Edition-Based Redefinition" in *Planning an Upgrade of Oracle Fusion Middleware*.

A.1.12 Instance Directories

When upgrading system components, such as OHS, you must provide the directory locations of the 11g instances that will be used as a starting point for creating new 12c component instances.

Use the **Add** button to include more than one instance, if needed.

NOTE: You cannot use the Upgrade Assistant to upgrade Oracle 10g instances to Oracle 12c. You must first upgrade Oracle 10g instances to 11g. For more information on migrating 10g to 11g, see the 11g upgrade documentation for your components.

Figure A-9   Instance Directories
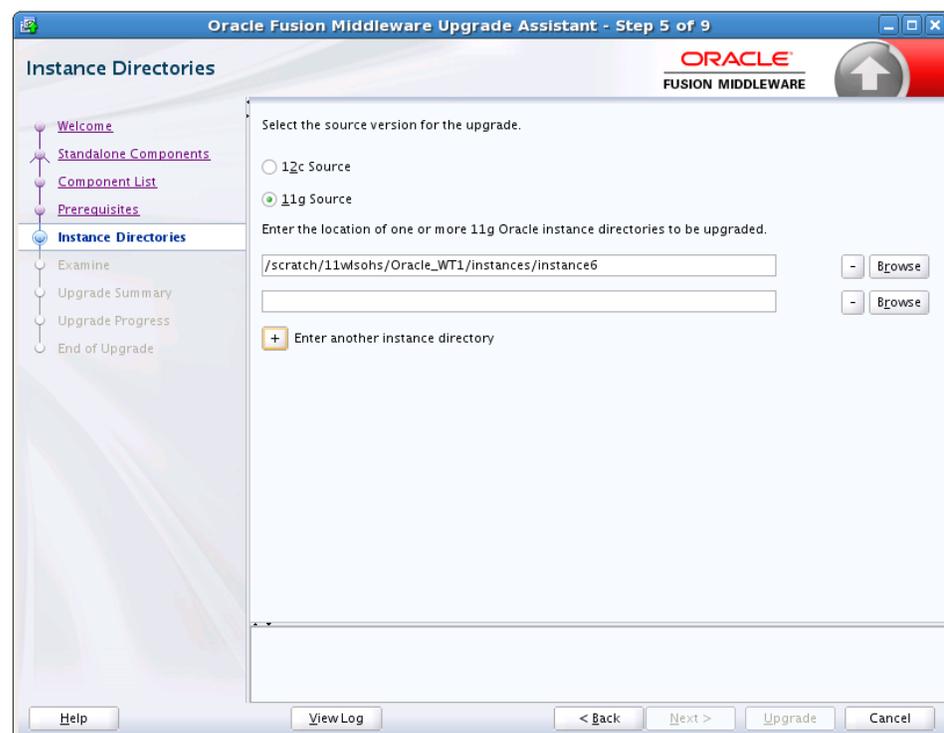
A.1.13 Node Manager

Use this screen to specify the credentials of the Node Manager that will be used to create a domain during the upgrade of standalone system components.

Note that the fields displayed in the screenshot may not appear during your upgrade. The conditions that trigger the fields to display are described in the table below.

The user name and password are only used to authenticate connections between Node Manager and clients. They are independent from the server Administrator ID and password.

Figure A–10   Node Manager

Element	Description
User Name	The user name used to access Node Manager.
Password	The password used to access Node Manager. You will need to re-enter the password for confirmation.
Listen Address	Enter the DNS name or IP address upon which Node Manager listens in the Listen Address field.
Listen Port	The listening port number of Node Manager.

A.1.14 Oracle Traffic Director

Provide the Oracle Traffic Director Information as presented on the screen (fields may vary):

- Administration Server Host

  The name of the host where the Traffic Director Administration Server is running. Search for a host by clicking the Search icon located at the end of the field.

  Note: On selecting the host, the Agent URL field will be automatically populated.

- Administration Server SSL Port

  SSL Port of the Administration Server.

- User Name

  Name of the administrator allowed to access the Traffic Director Administration Server.

- Password

  Password of the administrator allowed to access the Traffic Director Administration Server.

- SNMP Port

  The port on which the SNMP agent is listening. All the SNMP agents on all Traffic Director instance hosts should be running on the same port.

- Oracle Home

  Directory where the Traffic Director binaries have been installed.

A.1.15 ODI Supervisor

When you created the Master and Work repositories for ODI, the Repository Creation Utility prompted you to supply a password for the default SUPERVISOR account. On the ODI Supervisor screen, enter the following:

Element	Description
ODI Supervisor User Name	Supervisor account name for the ODI repository to be upgraded. The Supervisor user should be SUPERVISOR (all CAPS).
ODI Supervisor Password	Password that you created for the ODI Supervisor account.

Figure A-11   Oracle Data Integrator (ODI) Supervisor
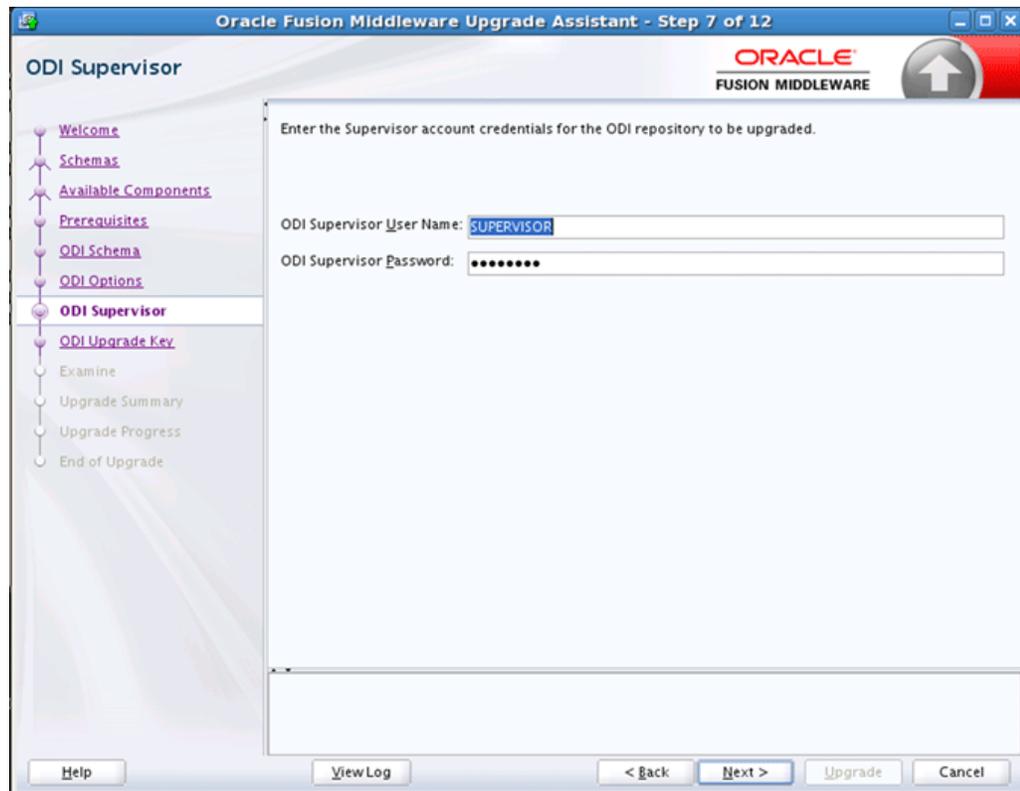
A.1.16 ODI Upgrade Key

For 11g to 12c upgrades only. This screen generates a unique identifier or *upgrade key* to convert 11g IDs for repository objects into unique GUIDs. You can use the auto-generated upgrade key or you can specify your own key in the Upgrade Key field.

Consider the following two scenarios when selecting an upgrade key:

- You know that an ID used in the 11g repository is the same as a project ID located in an XML file exported from the same repository. Use the Upgrade Key field to enter the project ID that was used in the 11g repository.

  In this scenario, the upgrade key used to upgrade the repository should be the same as the upgrade key used to import the XML file into the upgraded 12c repository. This ensures that the project object in the import file will be properly matched with the project object in the repository (when using one of SYNONYM import modes).

- You have 11g XML export file provided from a source containing objects created in another repository and you do not know which IDs were used. Use the auto-generated upgrade key or specify your own unique ID to avoid duplicate IDs.

  In this scenario, there is a chance that the file may contain a project that has the same internal ID. To prevent erroneous object matching, which may corrupt the metadata, a different, custom upgrade key should be used when importing that file into the repository.

NOTE: When multiple copies of the same object exist (in a repository or exported in XML files), the same GUID should be produced for all copies of the object. For this

reason, the same upgrade key must be used for all upgrade operations involving the copies of that particular object.

Figure A-12   Oracle Data Integrator (ODI) Upgrade Key

A.1.17 Oracle Golden Gate Monitor OGGMON Schema Prefix

Use this screen to specify the OGGMON schema prefix used when creating the Monitor schema with the Repository Creation Utility (RCU).

RCU requires that you supply a schema owner prefix for each schema you create. Provide the exact schema prefix used for the schema to be upgraded. The default prefix is DEV; as in DEV_OGGMON.

Figure A-13   Oracle Golden Gate Monitor Schema Prefix
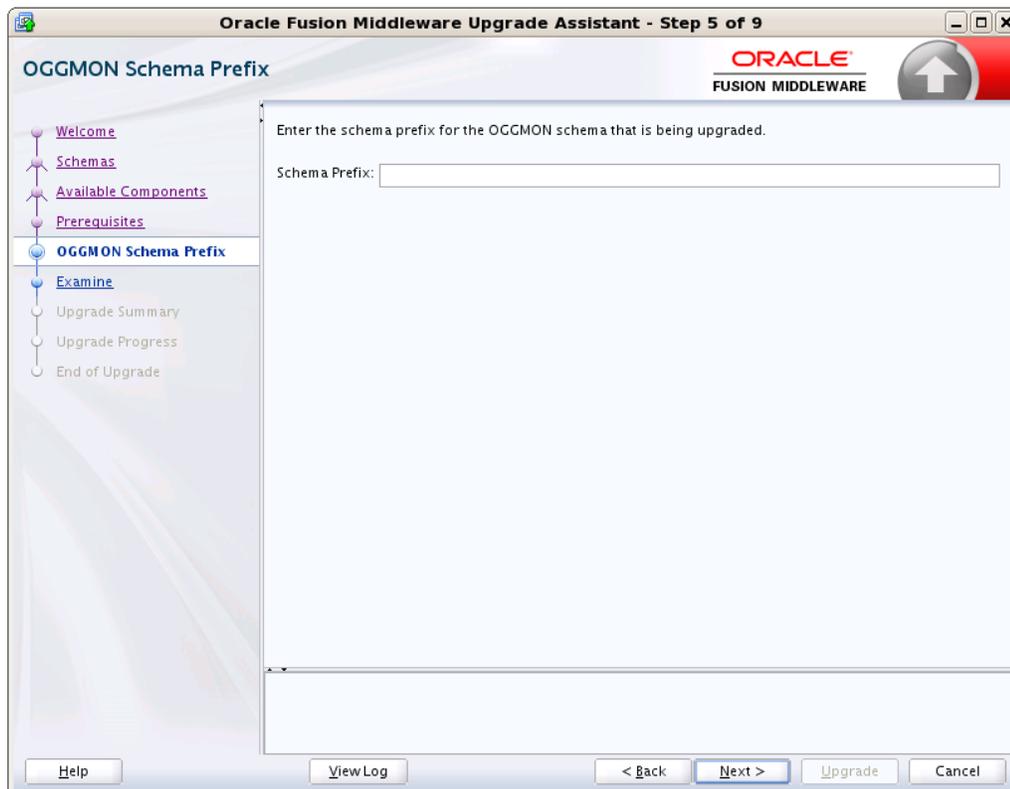
A.1.18 Veridata Directory

Use this screen to enter the location of the exisiting Veridata 11g home directory to be upgraded.

Click **Browse** and use the navigation tree to select the Veridata domain directory.

Figure A-14   Veridata DIrectory
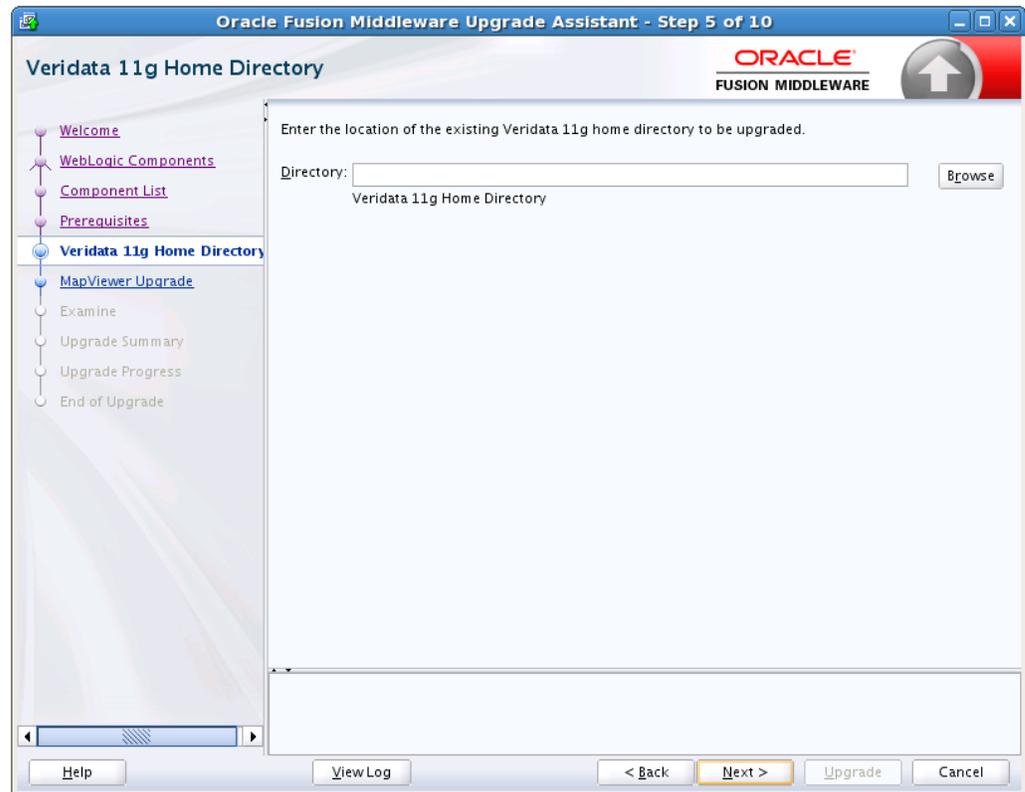
A.1.19 Veridata Schema Prefix

Use this screen to specify the VERIDATA schema prefix used when creating the Veridata schema with the Repository Creation Utility (RCU).

RCU requires that you supply a schema owner prefix for each schema you create. Provide the exact schema prefix used for the schema to be upgraded. The default prefix is DEV; as in DEV_VERIDATA.

Figure A-15   Veridata Schema Prefix
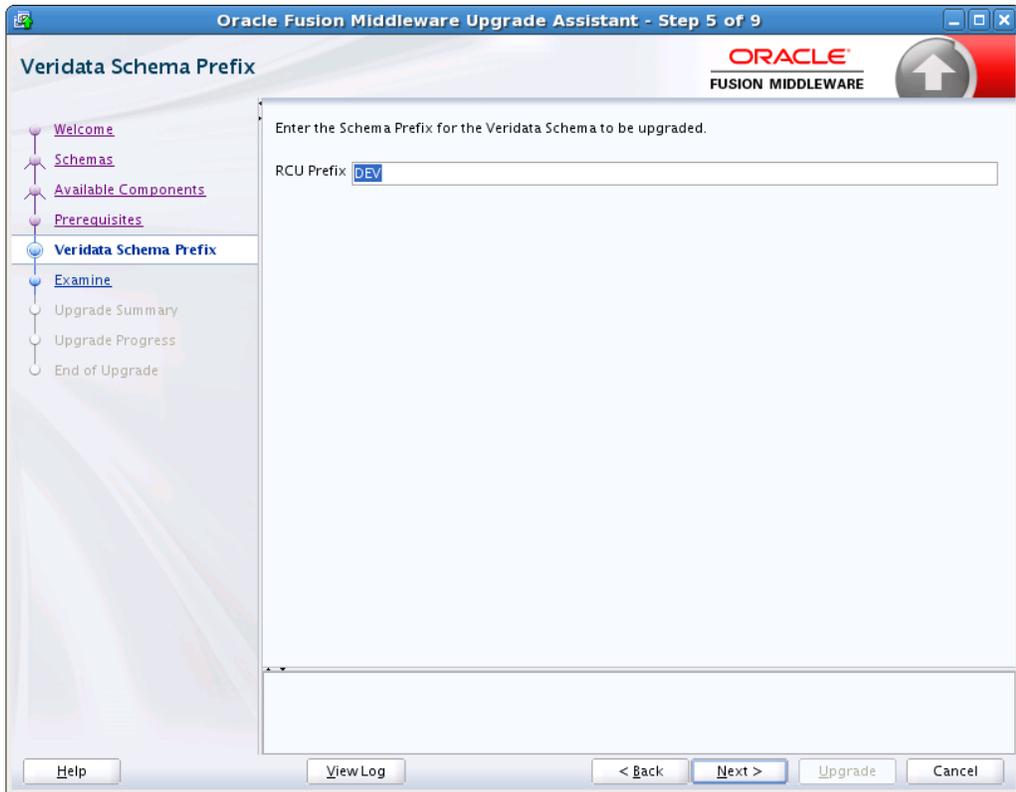
A.1.20 User Messaging Service Configuration

Figure A-16   User Messaging Service Configuration
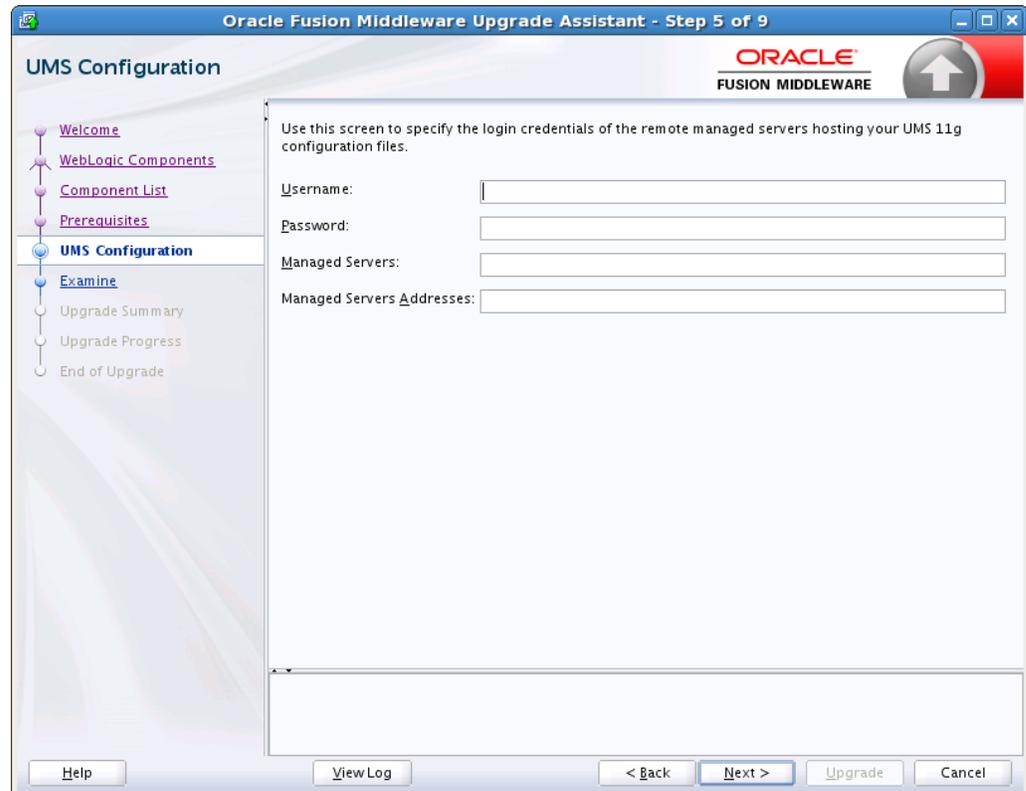
Use this screen to specify the login credentials of the remote managed servers hosting your UMS 11g configuration files. The Upgrade Assistant automatically copies remote configuration files if all necessary prerequisites are met and the required login information is provided as described in the table below.

If the UMS configuration files are not locally accessible on the machine where the upgrade is being executed, then you must manually enter the login credentials for each managed server (**ums_server1**,**ums_server2** for example).

In some cases, the configuration files must be copied to the machine where the upgrade is being executed (in most cases to the AdminServer machine). The Upgrade Assistant will attempt to copy the files, but if it cannot locate them, then you will have to manually copy them to the Admin Server.

For more information, see "Copying UMS Configuration Files" in *Upgrading to the Oracle Fusion Middleware Infrastructure*.

You will only need to copy the files manually if you receive a message stating that the Upgrade Assistant is not able to copy the configuration files. Once you have copied the files, you can restart the Upgrade Assistant and proceed with the upgrade.

Element	Description
Username	Provide the **Operating System user** who installed the product. This user will be used to fetch the remote configuration files. **NOTE**: This user must have permission to connect via `ssh` to the nodes where the remote managed servers are installed. The Username field is shown if: • no configuration files are found for the managed server on the local machine hosting the admin server or on a shared disk • there are more managed servers in the 12c domain where UMS is targeted.
Password	Provide the password associated with this user.
Managed Servers	If the Upgrade Assistant was unable to automatically detect the managed servers, then you must provide a comma separated list containing the names of the remote managed servers that contain the configuration files. For example: **ums_server1**,**ums_server2**
Managed Servers Addresses	Provide a comma separated list containing the complete hostnames or IP addresses for the nodes where the remote managed servers are running. The order of this list has to correspond with the list of managed server names provided above. For example: `fusionHost1.example.com,fusionHost2.example.com` where: `fusionHost1.example.com` hosts **ums_server1** and `fusionHost2.example.com` hosts **ums_server2**

A.1.21 MapViewer

On the MapViewer Upgrade page, for **File** specify the mapViewerConfig.xml file of the old MapViewer deployment, .and click Next.

This file is needed for migrating its system configuration settings. From this file's location, the old MapViewer's deployment directories are also derived.

Figure A-17   MapViewer Configuration File Directory
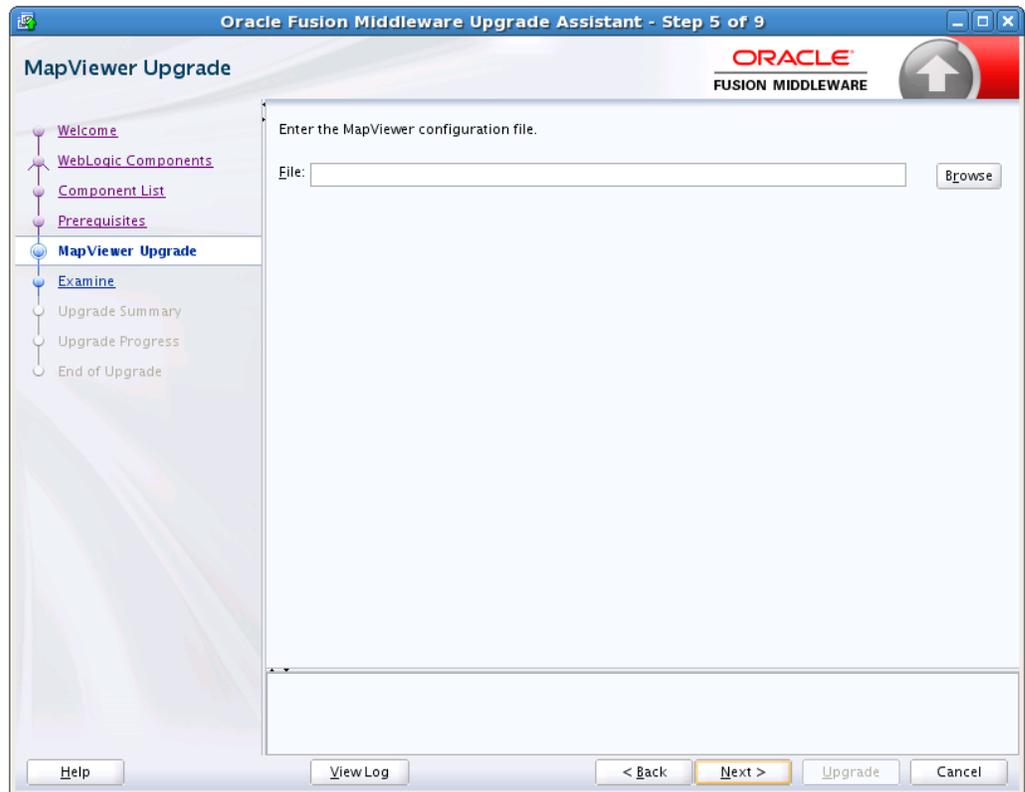
A.1.22 Examine

Figure A-18   Examine
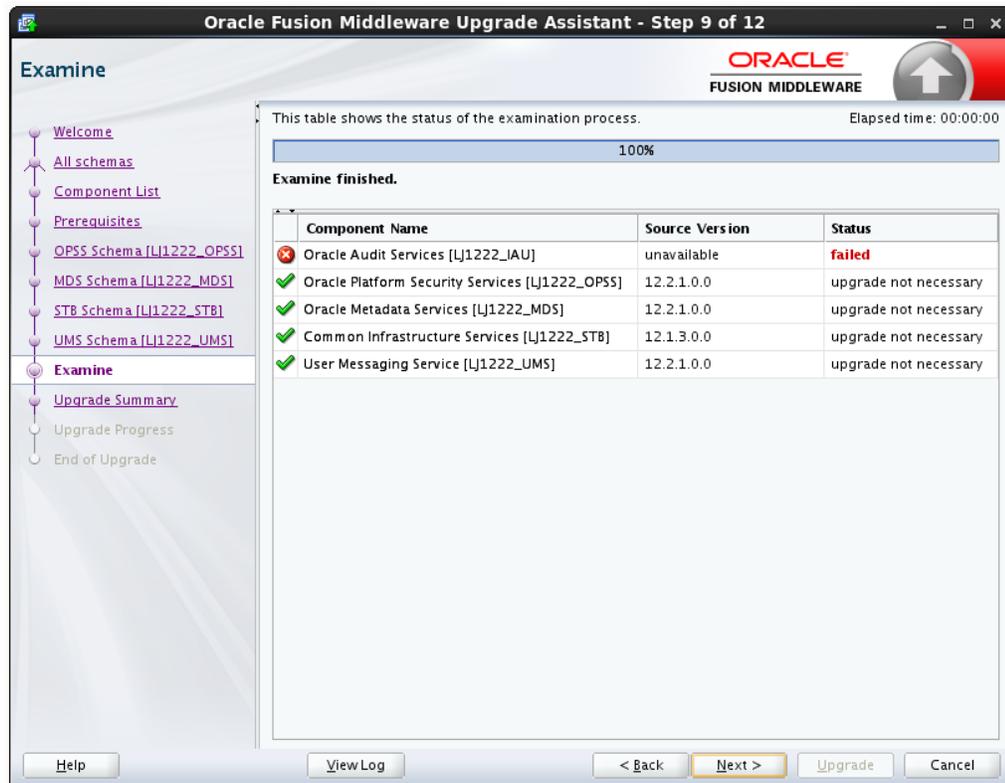
The Upgrade Assistant examines each component to be sure it meets a minimum set of criteria before you begin the upgrade process.

This screen displays the status of the Upgrade Assistant as it examines each component, verifying that the component is ready for upgrade.

The Upgrade Assistant examines each component to be sure it meets a minimum set of criteria before you begin the upgrade process.

Upgrade Assistant displays the schema **Source Version** of the schema on this screen if the information is listed in the schema version registry table. If the schema was not created using RCU, or the source version cannot be found, the source version will display `unavailable`.

Note: Issues detected during the Examination phase may be resolved and the Upgrade Assistant can be started again. Once the Upgrade phase has started, however, you will need to restore your pre-upgrade environment from backup before starting the Upgrade Assistant again.

The description of the **Status** indicators for the components is listed in the following table:

Status	Description
in progress...	The Upgrade Assistant is examining the upgrade items for the component.
pending...	The component will be examined when the Upgrade Assistant finishes the preceding component.
failed	Upgrade items were missing or did not meet upgrade criteria. The Upgrade Assistant cannot upgrade the component until the issues have been resolved. Click **View Log** to troubleshoot the errors and then restart the Upgrade Assistant.
succeeded	Upgrade items were found and are valid for upgrade.

Canceling the examination process has no effect on the schemas or configuration data; the only consequence is that the information the Upgrade Assistant has collected must be collected again in a future upgrade session.

A.1.23 Examine Failure

Figure A-19 Examine Failure
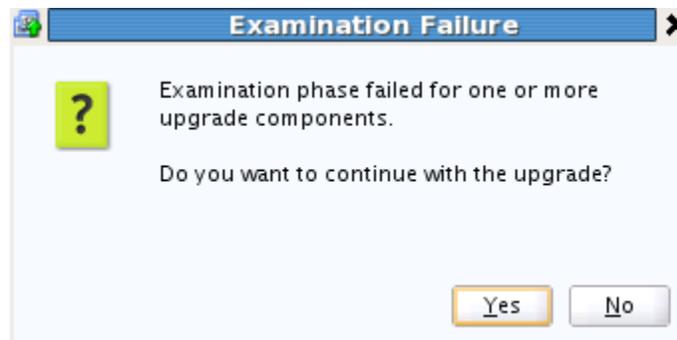
This dialog box appears when one of more of your components failed the examination phase and you selected to continue with the upgrade. If there was an examination failure, you should consider canceling the upgrade (click **No**) and review the log files.Since the upgrade has not yet started, you can resolve the issues detected during the examination phase and restart the Upgrade Assistant without restoring from backup.

UMS Upgrades Only:

During the configuration upgrade you can get this error during the **examination phase**. For User Messaging Service, the way to recover is to copy the UMS config files manually and restart the Upgrade Assistant.

If you can get an error during the **upgrade phase** , the way to recover is to restore backups and copy the config files manually and restart the Upgrade Assistant.

A.1.24 Upgrade Summary

Figure A-20   Upgrade Summary
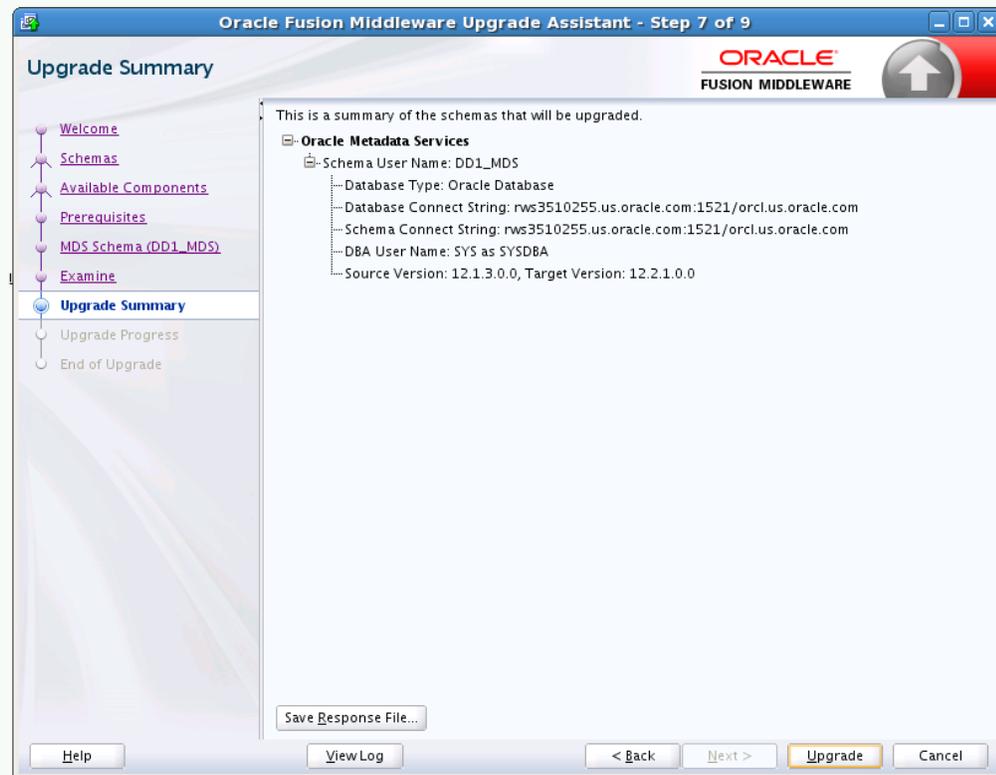
Reviewing the Upgrade Summary

Expand and collapse the tree to show or hide details about the data provided in the wizard screens, such as schema details, Oracle WebLogic Server connection details, and Oracle WebLogic domain directory information.

The Summary screen also displays the **Source Version** of the schema being upgraded and the resulting **Target Version** post upgrade. Make sure that both versions are correct before proceeding with the upgrade.

Starting the Upgrade Process

Click **Upgrade** to start the upgrade process.

If you are upgrading a schema, verify that you have a backup of the database that hosts the schema.

Save Response File

The **Save Response File** option creates a file that can be used as input to the Upgrade Assistant. The response file collects all the information that you have entered through the Upgrade Assistant's graphical user interface screens, and enables you to perform a silent upgrade at a later time. The silent upgrade performs exactly the same function that the Upgrade Assistant wizard performs, but you do not have to manually enter the data again.

A.1.25 Upgrade Progress

Figure A-21   Upgrade Progress
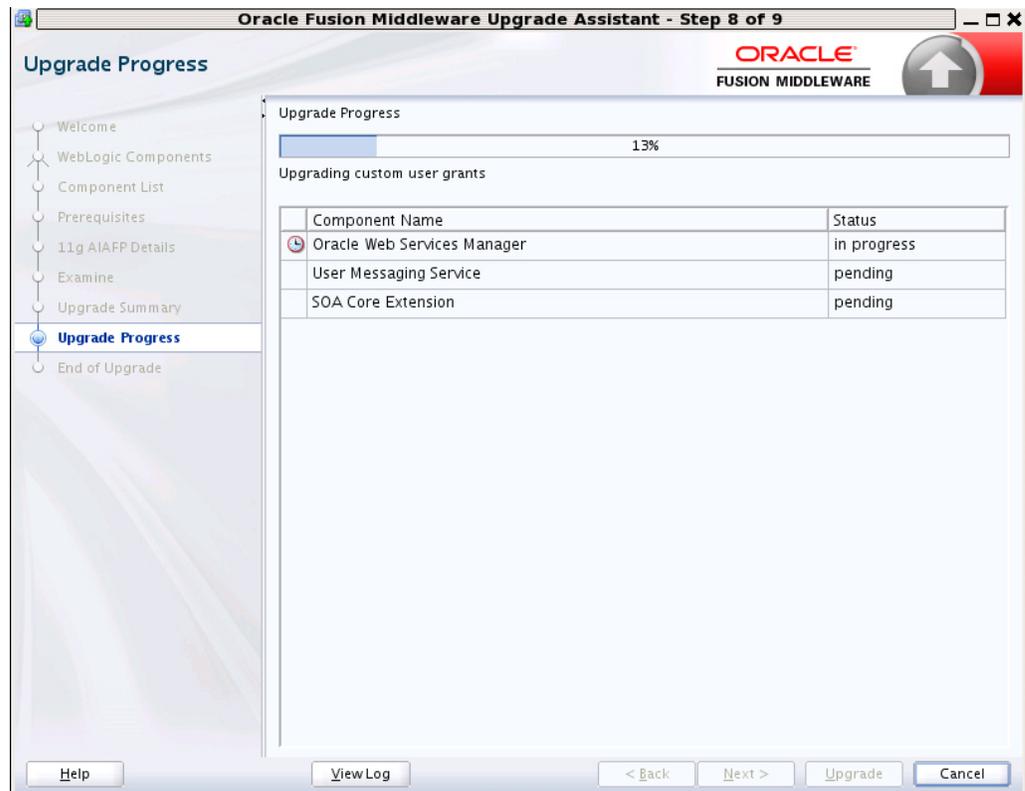
This screen shows the status of the current upgrade process and the projected **Target Version** of the component after a successful upgrade.

Note that the progress bar is NOT a measure of time remaining for the upgrade. The progress bar is a moving graphical display of completed upgrade steps for each component being upgraded. In some cases, the progress bar does not move at a steady pace. It might move very quickly over a certain portion of the progress bar, and move very slowly, or even appear to freeze, for another component that is performing a long-running database operation. That does not mean that the upgrade progress is stalled, it simply indicates that a long-running operation is being performed. Different upgrade operations, especially during a schema upgrade, will operate at different paces.

Caution: Allow the Upgrade Assistant enough time to perform the upgrade. Do not cancel the upgrade operation unless absolutely necessary. Doing so may result in an unstable environment.

The status of each component upgrade is indicated by one of the following messages that can appear next to the component name. The following table describes each status message.

Status	Description
in progress...	The Upgrade Assistant is upgrading the component's upgrade items.

Status	Description
pending...	The component will be upgraded when the Upgrade Assistant finishes the preceding component.
upgrade not necessary	The component was upgraded previously by the Upgrade Assistant or the component was just installed and is already at the latest version. No action will be taken on this component.
skipped	The component is dependent on another component which has a status of "failed". When the status is "skipped" no upgrade is attempted for that component.
failed	Upgrade items were missing or did not meet upgrade criteria. The component cannot be upgraded. Click View Log to troubleshoot the errors.
succeeded	Upgrade items were upgraded successfully.

If any components are not upgraded successfully, refer to the Upgrade Assistant log files for more information.

A.1.26 Upgrade Success

Figure A-22   Upgrade Success
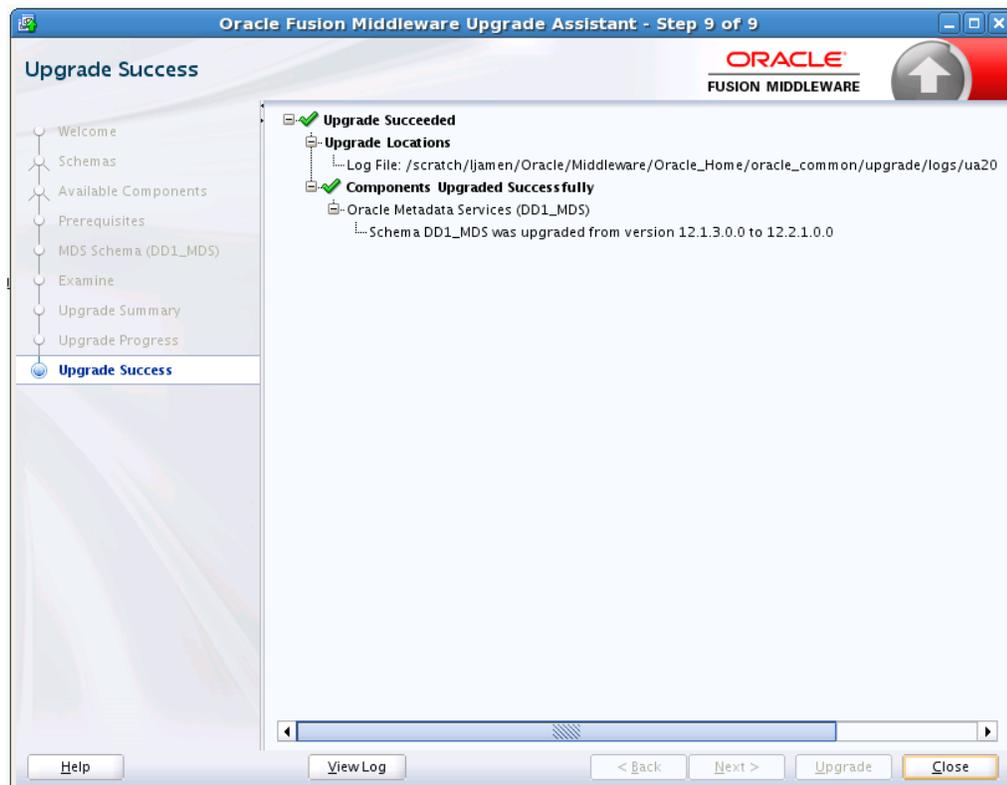
The upgrade was successful. The Post-Upgrade Actions window describes the manual tasks you must perform to make the component function in the new installation. This is an optional window that will only show up if a component has post-upgrade steps.

In addition, be sure to do the following:

- View the `postupgrade.txt` file in the Oracle home:

  On Unix systems:

  `ORACLE_HOME/oracle_common/upgrade/logs`

  On Windows systems:

  `ORACLE_HOME\oracle_common\upgrade\logs`

- Review the upgrade topics specific to your Oracle Fusion Middleware environment for any additional post-upgrade tasks.

A.1.27 Upgrade Failure

Figure A-23   Upgrade Failure
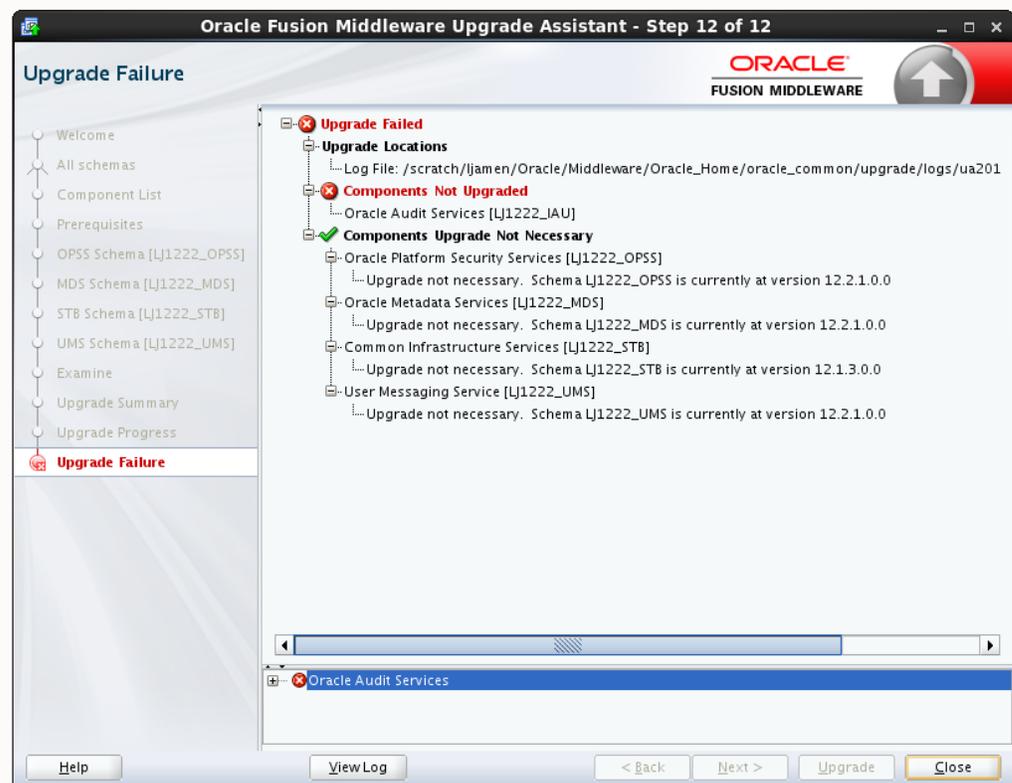
The upgrade of one or more components has failed. The component cannot be upgraded at this time. Click **View Log** to troubleshoot the errors.

You will have to fix the issues in the pre-upgrade environment before starting the Upgrade Assistant again. Restore your pre-upgrade environment from backup (making sure to keep the original backup files in a separate location), fix the issues, and restart the Upgrade Assistant.

A.1.28 Cancel Upgrade

Figure A-24   Cancel Upgrade

You get the above Confirm Cancel screen when you click **Cancel** while the upgrade plugin is actively running (that is, you are on the Upgrade page and the progress bar is less than 100%).

Figure A-25   Cancel Upgrade
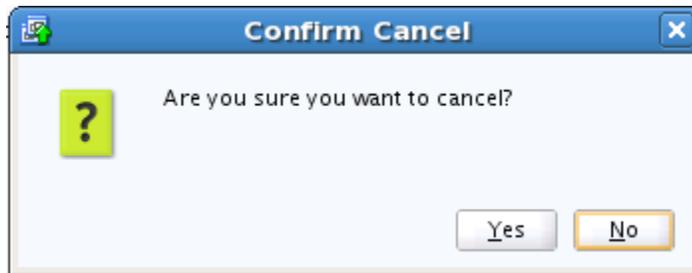
You get the above Confirm Cancel screen on clicking **Cancel** when no upgrade plugins are actively running.

Important Note: If you cancel a schema upgrade, you must restore a backup of the database that hosts the schema and its environment (the pre-upgrade directory structure).

A.1.29 Log Viewer

Click **View Log** from any of the screens to see the latest logged information.

The log file is managed by the command line options you set. See Starting the Upgrade Assistant with Additional Parameters (Optional) for more information.

Figure A-26   Sample Log File
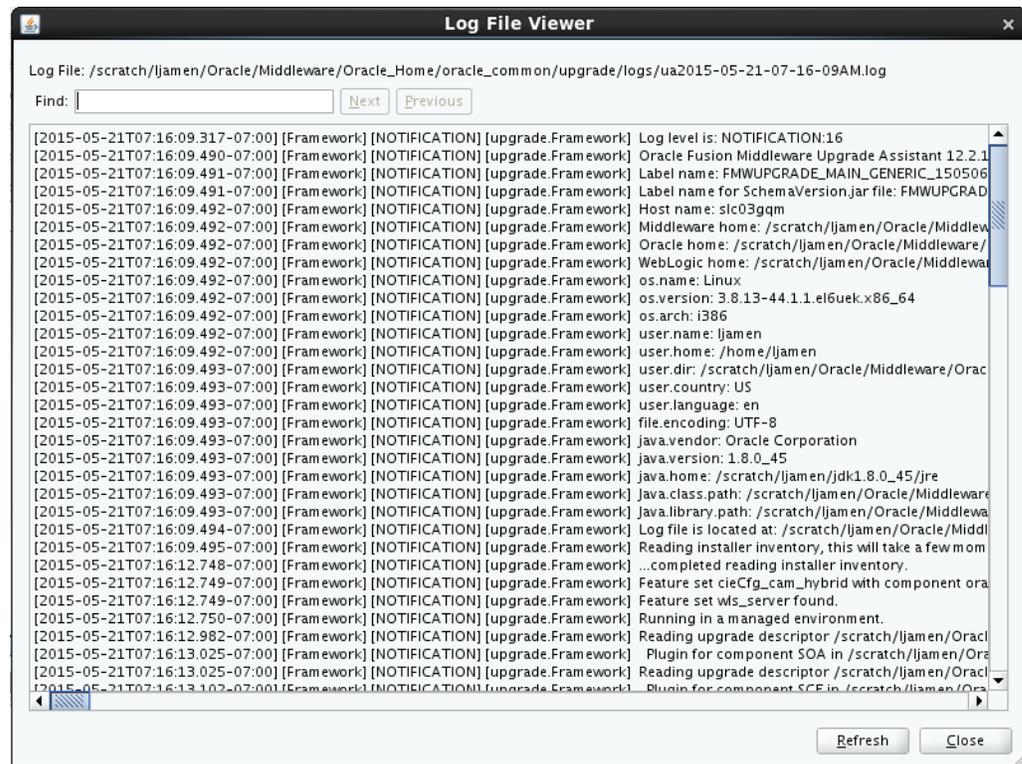
A.2 Readiness Check Screens

This section describes the screens that are presented when running the Upgrade Assistant in -readiness mode.

The Upgrade Assistant can be run in `-readiness` mode before you perform the actual upgrade to detect any potential problems with the pre-upgrade environment.

Welcome

Readiness Check Type: Individually Selected Schemas

Readiness Check Type: Domain Based
> The **Domain Based** option is used to check all of the upgrade-eligible schemas and/or component configurations used by the domain. The Upgrade Assistant detects all of the schemas for you. You can check schemas and component configurations at the same time. Or, if you prefer, you can select one or the other. In either case, you must specify the **Domain Directory** that is to be reviewed.

Available Components
> This screen appears if you select **Individually Selected Schemas** in the Schemas screen.

Schema Credentials

Readiness Summary

Readiness Check

Log Viewer

Readiness Success

Sample Readiness Report

A.2.1 Welcome

Figure A-27   Readiness Welcome
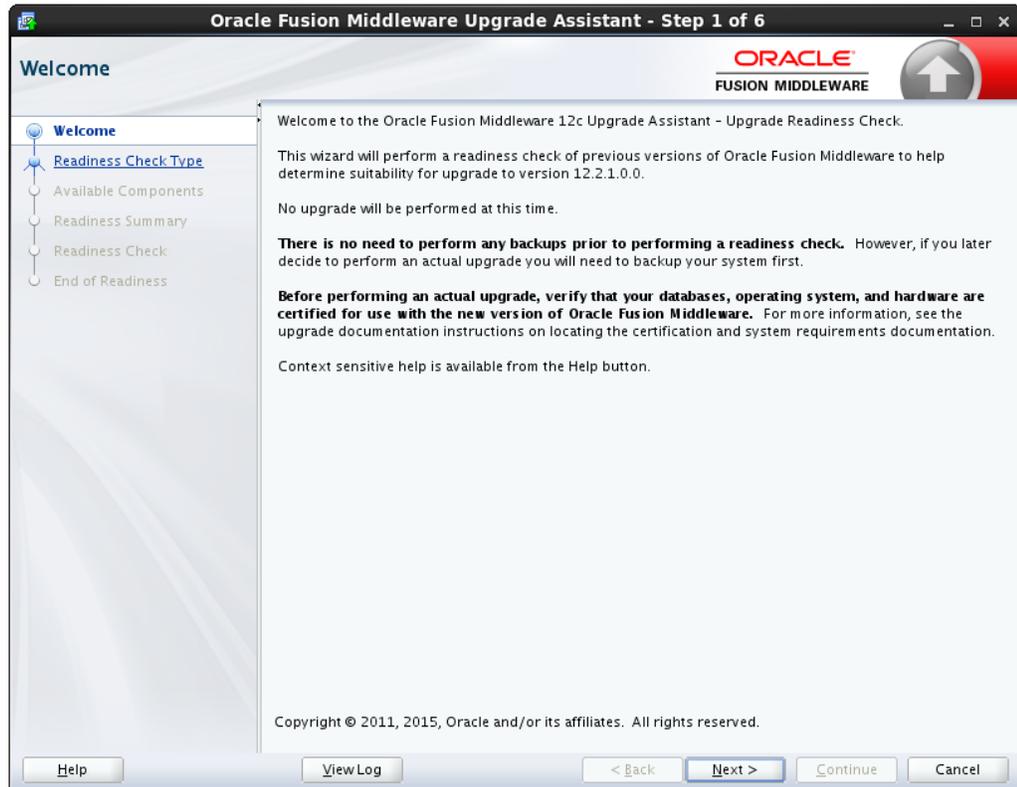
The Upgrade Assistant Readiness Check performs a read-only, pre-upgrade review of your existing Oracle Fusion Middleware schemas and Oracle WebLogic component configurations.

The Readiness Check provides a formatted, time-stamped Readiness Report so you can address any issues before you attempt the actual upgrade. If no issues are detected, you can begin the upgrade process.

The Upgrade Assistant Readiness Check can be run with your existing 11g or 12c domain online or offline.

Note: While readiness check ships with 12.2.1, it only checks supported pre-upgrade environments.

The Readiness Check can be run any number of times before the actual upgrade is performed. However, do not run after the Readiness Check after an upgrade has been performed, as the report will not provide valid results.

Oracle recommends that you read this report thoroughly before performing an upgrade.

A.2.2 Readiness Check Type: Individually Selected Schemas

Figure A-28   Individually Selected Schemas
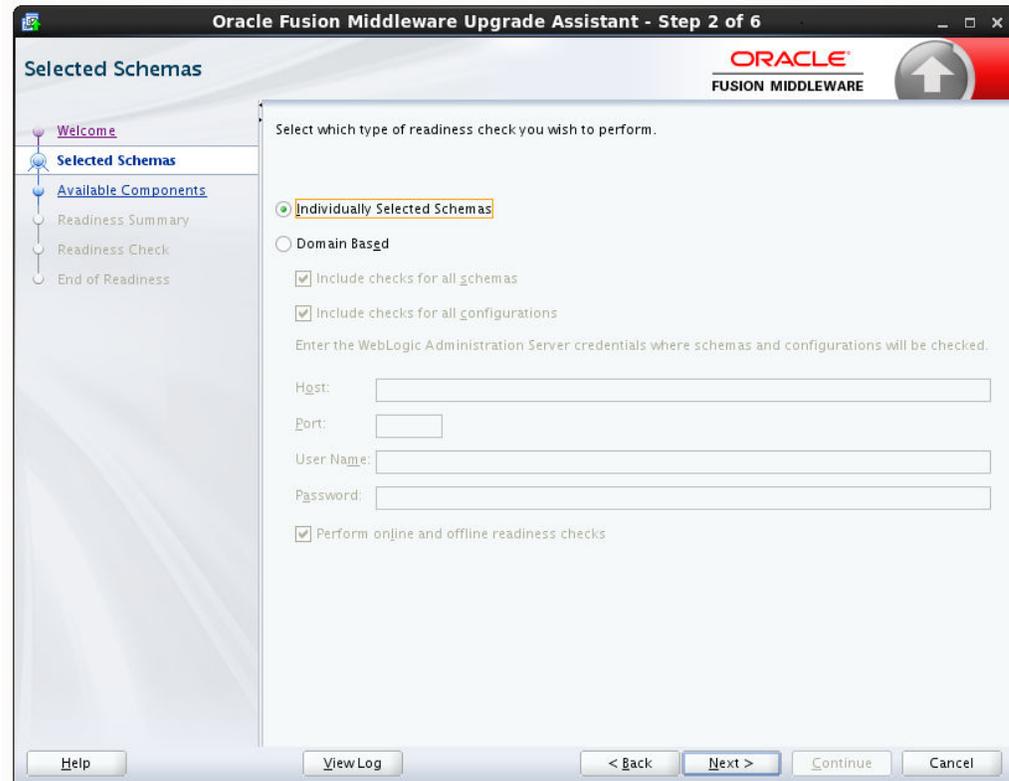
You have two options when running the readiness check:

- **Individually Selected Schemas**

- **Domain Based**

Select the **Individually Selected Schemas** option to limit the check to specific schemas. Click **Next** and you will be required to supply the schema credentials.

Readiness checks are performed on the schemas that you connect to. The readiness check report tells you whether a schems is supported for an upgrade, or where an upgrade is needed.

A.2.3 Readiness Check Type: Domain Based

The **Domain Based** option is used to check all of the upgrade-eligible schemas and/or component configurations used by the domain. The Upgrade Assistant detects all of the schemas for you. You can check schemas and component configurations at the same time. Or, if you prefer, you can select one or the other. In either case, you must specify the **Domain Directory** that is to be reviewed.

You have several options when checking the WebLogic Server domain.

You can select one - or more - of the following options each time you run the Domain Based Readiness Check:

- **Include checks for all schemas**

Select this option to enable the Upgrade Assistant to discover and review all components that have a schema available to upgrade.

- **Include checks for all configurations**

  Select this option to review component configurations for a managed WebLogic Server domain.

  You can perform domain configuration check even when the domain is online or offline.

- **Perform online and offline readiness checks**.

  Select this option to perform additional online readiness checks . This option will require your domain to be online. You must provide the domain's host name, port, user name, and password that you plan to check.

  If you do not select this option your domain can be offline. You must provide the domain location that you plan to check.

Figure A-29   WebLogic Server Readiness Check Options

A.2.4 Available Components

This screen appears if you select **Individually Selected Schemas** in the Schemas screen.

Figure A-30   Available Components
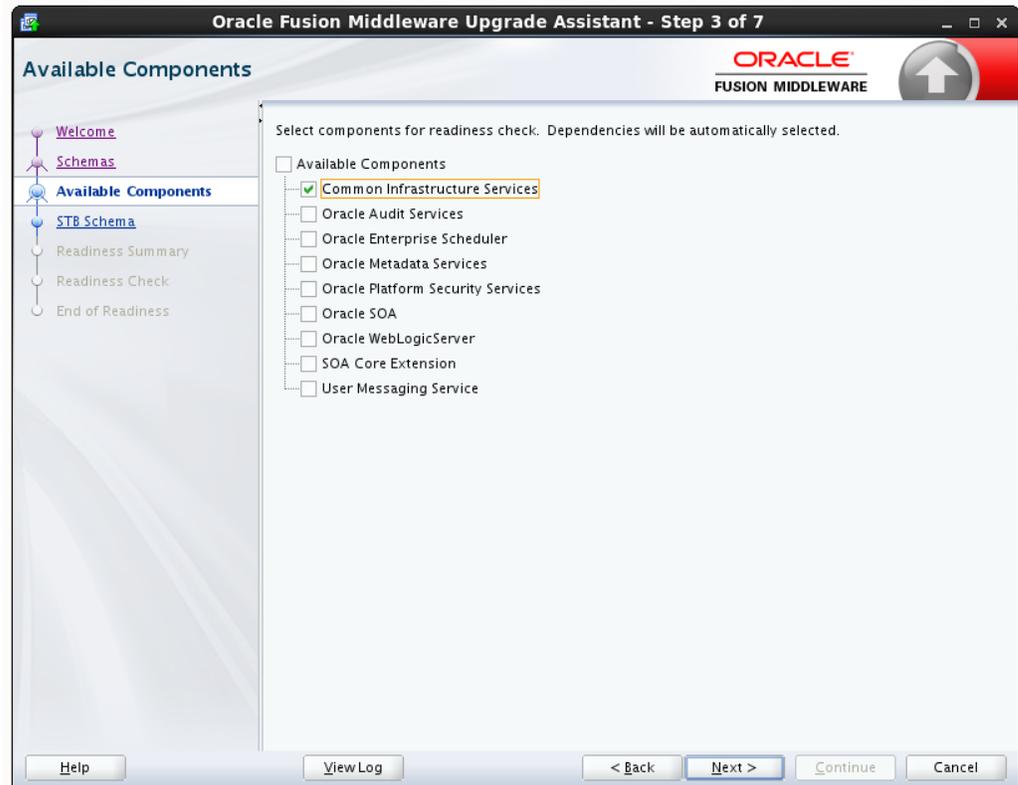
If you chose **Individually Selected Schemas** this screen lists the available components for which the schemas will be selected. If you select something here, readiness check will be performed on that component's schema. You must select one or more components from the list to perform readiness check on them.

A.2.5 Schema Credentials

Use this screen to enter information required to connect to the selected schema and the database that hosts the schema. If the schema that is to be reviewed was created by RCU in a prior Fusion Middleware release then you will see a drop-down menu listing the possible schema names as shown below.

Click **Connect** to connect to the database then select the schema to be reviewed. NOTE: Most schemas will have this information pre-populated. If, however, the Upgrade Assistant is unable to detect the connection details, then they must be entered manually as shown below.

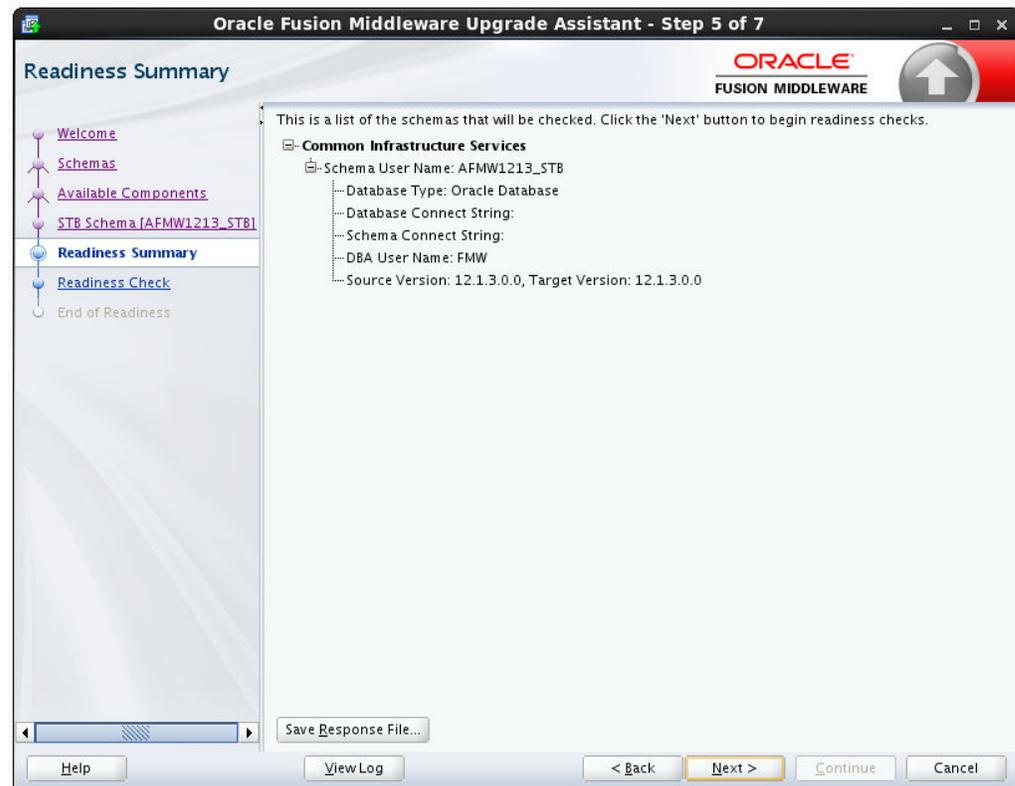If multiple components are selected, then the Schema Credential screens appear in dependency order.

Figure A-31 Schema Credentials

A.2.6 Readiness Summary

This screen provides a high-level overview of the readiness checks performed based on your selections.

For a detailed report, click **View Log**.

Figure A-32   Readiness Summary

A.2.7 Readiness Check

This screen shows the overall progress and completion details of the running readiness check. If you are checking multiple components, then each gets component will have its own progress bar and will be checked in parallel. Once completed, click **View Readiness Report** to see the full text report.

CAUTION: If you are running the readiness check on your online production environment, Oracle recommends that you perform the check during off-peak hours to prevent performance degradation.
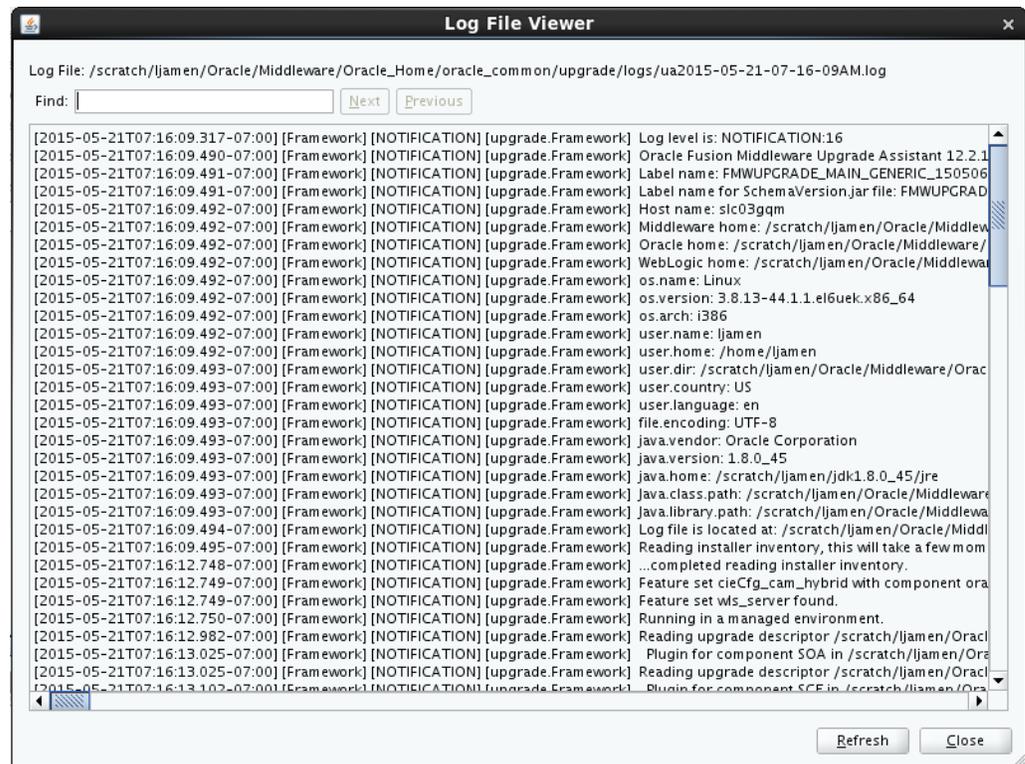
Figure A-33   Readiness Check Complete
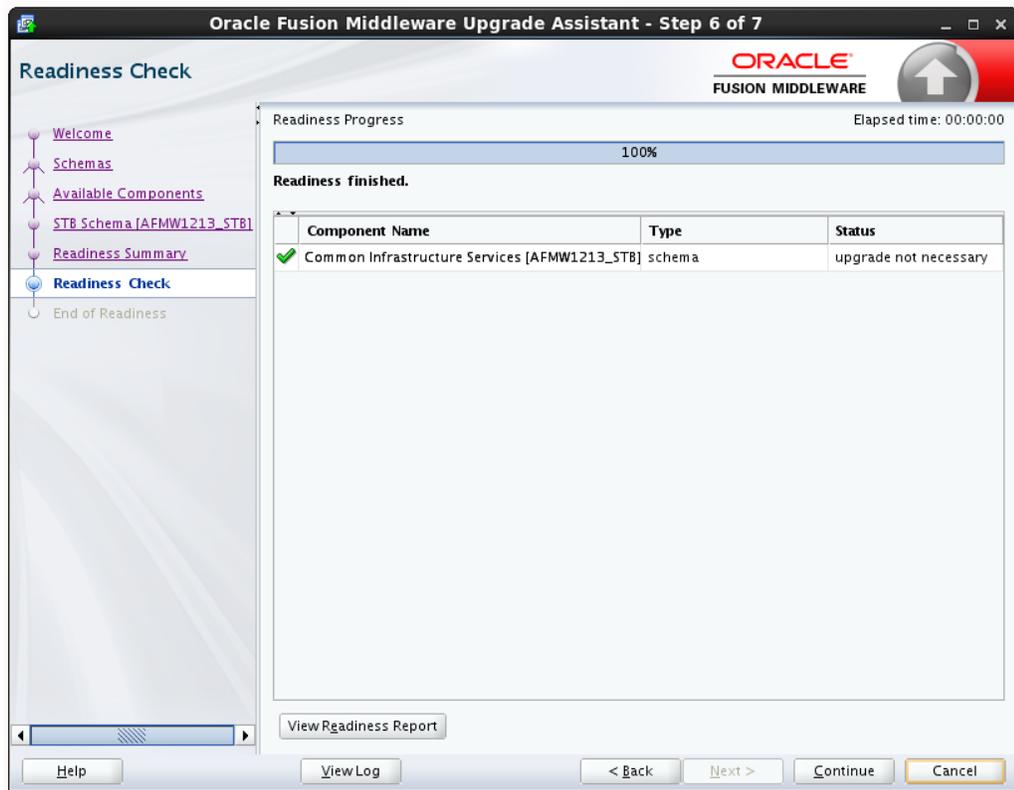
A.2.8 Log Viewer

Click **View Log** from any of the screens to see the latest logged information.

The log file is managed by the command line options you set. See Starting the Upgrade Assistant with Additional Parameters (Optional) for more information.
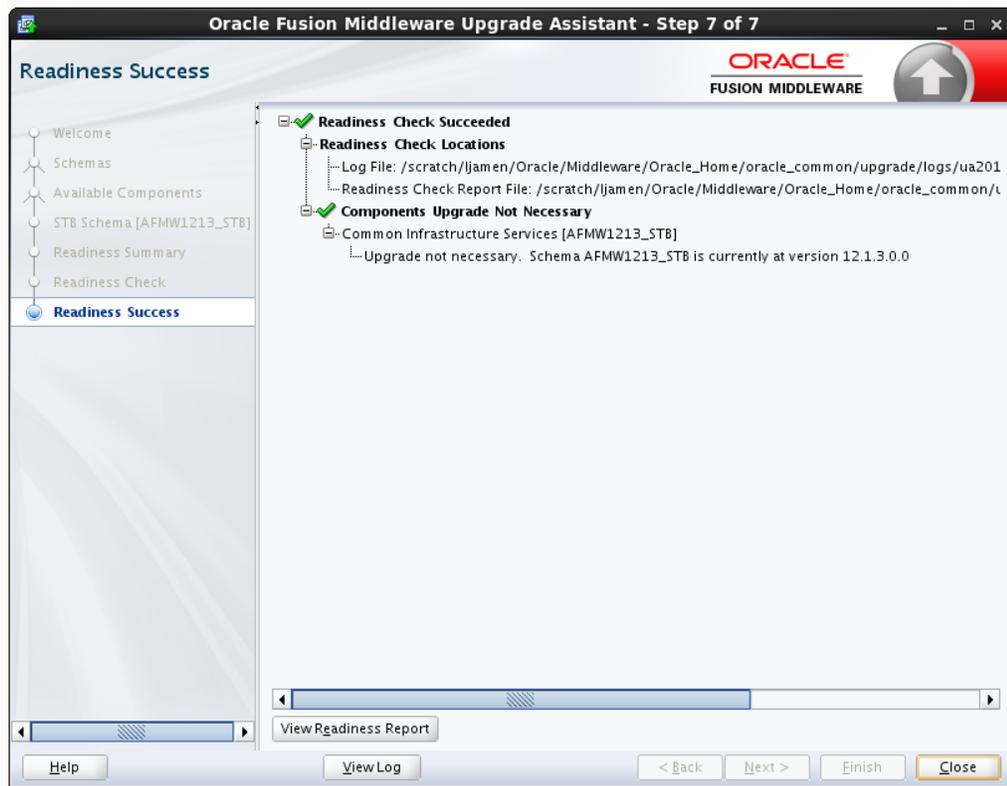
Figure A-34   Sample Log File

A.2.9 Readiness Success

Readiness success indicates that the readiness review was successfully completed.

Even with a successful completion of the review, you should still click **View Readiness Report** and review the Readiness Report before you perform the actual upgrade.

Figure A-35   End of Readiness: Readiness Success

A.2.10 Sample Readiness Report

A formatted Readiness Report is prepared for you after running the check. Make sure that you review the report and correct any issues before you start the actual upgrade. Use the Find option to search for a particular word within the report (such as a schema name or command, for example.)

The report also indicates where the completed Readiness Check Report file is located.

Figure A-36   Readiness Report Viewer

Index

E

errors

errors *(continued)*
 resolving when using Upgrade Assistant, *3-3*
examination failures, *3-3*