Oracle Commerce Guided Search

Tools and Frameworks Migration Guide

Version 11.1 • July 2014

ORACLE®

COMMERCE

Contents

Copyright and disclaimer

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

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

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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

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

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

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

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

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

Preface

Oracle Commerce Guided Search is the most effective way for your customers to dynamically explore your storefront and find relevant and desired items quickly. An industry-leading faceted search and Guided Navigation solution, Guided Search enables businesses to influence customers in each step of their search experience. At the core of Guided Search is the MDEX Engine™, a hybrid search-analytical database specifically designed for high-performance exploration and discovery. The Oracle Commerce Content Acquisition System provides a set of extensible mechanisms to bring both structured data and unstructured content into the MDEX Engine from a variety of source systems. The Oracle Commerce Assembler dynamically assembles content from any resource and seamlessly combines it into results that can be rendered for display.

Oracle Commerce Experience Manager enables non-technical users to create, manage, and deliver targeted, relevant content to customers. With Experience Manager, you can combine unlimited variations of virtual product and customer data into personalized assortments of relevant products, promotions, and other content and display it to buyers in response to any search or facet refinement. Out-of-the-box templates and experience cartridges are provided for the most common use cases; technical teams can also use a software developer's kit to create custom cartridges.

About this guide

This guide describes how to upgrade earlier versions of Oracle Endeca Commerce Tools and Frameworks to the most recent version, and how to migrate an Endeca application and users to the most recent version of Tools and Frameworks.

Complete the steps described for your migration arc, as well as any steps in the Required Changes section that apply to features you use. Changes described in Behavioral Changes either do not require action on your part, or are optional.

Who should use this guide

This guide is intended for application developers and administrators who are using Oracle Endeca Commerce Tools and Frameworks and are responsible for migration tasks.

Conventions used in this guide

This guide uses the following typographical conventions:

Code examples, inline references to code elements, file names, and user input are set in `monospace` font. In the case of long lines of code, or when inline monospace text occurs at the end of a line, the following symbol is used to show that the content continues on to the next line: ¬

When copying and pasting such examples, ensure that any occurrences of the symbol and the corresponding line break are deleted and any remaining space is closed up.

Contacting Oracle Support

Oracle Support provides registered users with answers to implementation questions, product and solution help, and important news and updates about Guided Search software.

You can contact Oracle Support through the My Oracle Support site at *https://support.oracle.com*.

Chapter 1

Introduction

This section contains basic information about the migration process.

Recommended reading

In addition to reading this document, Oracle recommends that you read the following documents for important information about the release.

Release Announcement

The Release Announcement provides an explanation of the new features that were added in the latest version. The Release Announcement is available as part of the Tools and Frameworks documentation set on the Oracle Technology Network.

Release Notes

The Release Notes (README.txt) provide information about bug fixes and known issues for this release. The Release Notes are installed into `ToolsAndFrameworks/<version>`.

Note: Although the release notes are available with the installation packages, the latest versions (and possible revisions) of release notes for each package are also available on the Oracle Technology Network.

Oracle Commerce Guided Search Getting Started Guide

The *Oracle Commerce Guided Search Getting Started Guide* gives an overview of Endeca components and provides setup and operations instructions for a single machine environment. You can download the *Oracle Commerce Guided Search Getting Started Guide* from the Oracle Technology Network.

Oracle Commerce Guided Search Migration Guides

In addition to the *Oracle Commerce Tools and Frameworks Migration Guide*, you may also need to upgrade other components of Oracle Commerce Guided Search. The migration paths for each component are documented in the following guides:

- *Oracle Commerce MDEX Engine Migration Guide*
- *Oracle Commerce Guided Search Platform Services Migration Guide*
- *Oracle Commerce Content Acquisition System Migration Guide*

Each guide is available on the Oracle Technology Network.

Package compatibilities

See the *Oracle Commerce Supported Environments Matrix* document in the My Oracle Support knowledge base at `https://support.oracle.com/` for information on supported operating systems and Web browsers.

Upgrade paths

There is one supported upgrade path:

- From Oracle Endeca Tools and Frameworks 11.0 to Oracle Endeca Tools and Frameworks 11.1.

Unsupported upgrade paths

If you are migrating to Oracle Endeca Tools and Frameworks 11.1 from a version other than those listed above, you must first migrate to the previous release (11.0).

Upgrading on a single host

For the sake of simplicity, this guide describes how upgrade the software and migrate an Endeca application on a single machine. This makes installation, configuration, and communication among components simpler as you migrate application configuration.

Chapter 2

Migrating Tools and Frameworks from 11.0 to 11.1

The following section lists the steps for migrating from Tools and Frameworks 11.0 to Tools and Frameworks 11.1.

About migrating from 11.0 to 11.1

To migrate your Tools and Frameworks installation from 11.0 to 11.1, you must first create backups of your configuration files and application content, then upgrade to the latest Tools and Frameworks package. Afterwards, you can restore Workbench configuration from your backups, and update application configuration from a combination of backups and migration scripts.

Migrating from Tools and Frameworks 11.0 to 11.1 consists of the steps below. They are described in detail in the following sections:

1. Create backups:
   a. Back up Workbench configuration from the `%ENDECA_TOOLS_CONF%\conf` directory.
   b. Back up application state from the Workbench `%ENDECA_TOOLS_CONF%\state` directory.
   c. Use the export scripts included with Tools and Frameworks 11.0 to export user information from version 11.0.
   d. Back up Workbench by running the `export_site` script for each application you are migrating.
   e. Back up any Forge data ingest configuration by creating a copy of the `config\pipeline` directory in your application.
   f. If you use CAS, back up `config\CAS` and `config\MDEX` directories.
   g. Optionally, back up each deployed application directory, for example, `C:\Endeca\apps\Discover`.
   h. Stop and uninstall your Tools and Frameworks 11.0 Workbench.
2. Upgrade the MDEX Engine to 6.5.1.
3. Upgrade Platform Services to 11.1
4. Upgrade Tools and Frameworks to version 11.1:
   a. Install Tools and Frameworks 11.1 as documented in the *Tools and Frameworks Installation Guide*.
5. Restore Workbench configuration:
   a. Restore Workbench configuration from the backups created in Step 1a.
   b. Restore the application state directories in Workbench from the backups created in Step 1b.
6. Restore each of your applications:

a. Deploy the 11.1 version of your application.
b. Restore Forge configuration from backups.
c. Restore CAS configuration from backups.
d. Initialize your application.
e. Repeat the substeps above for each application you are migrating.
f. Restore Workbench user information using the `import_users` script.
g. Configure the the `migrate-workbench` script. Optionally, you may also configure logging for the migration script.
h. Migrate application configuration by running the `migrate-workbench` script.

   The script converts the application content. The new content is stored as a modified version of the Workbench export file.

i. Restore Workbench content from the modified export file created in Step 6f by running the `import_site` script.
j. Move packaged services to a new services page under /pages for Experience Manager
k. Run a baseline update to load data into the Dgraph and start it.
l. Verify that Workbench configuration and application content are present in Workbench. Optionally, promote content to your live servers.
m. Repeat the substeps from 6g above for each application you are migrating.

Upgrading Projects to Developer Studio 11.1

To upgrade an existing project created with Developer Studio 11.0 or earlier to Developer Studio 11.1, follow these steps:

1. Open the project file of the existing project.
2. When prompted to upgrade the project to the new version (11.1), click Yes.
3. Save the upgraded project to a different location from the location of the existing project file.

Be sure to use the upgraded project, rather than the old project, for all subsequent activities.

Note: If you are upgrading to 11.1 from a version earlier than 6.1, manually editing the project version number in the Developer Studio configuration files is not recommended, because upgrading in this way can omit changes that have been introduced by intervening releases.

However, you can upgrade from version 6.1, without omitting changes, by editing the version number. In this case, simply edit the following line in the Developer Studio configuration file, as follows:

```
< <STUDIO_PROJECT APP_VERSION="6.1" NAME="My Project" VERSION=" 600"> --- >

  <STUDIO_PROJECT APP_VERSION="6.1" NAME="My Project" VERSION=" 1110">
```

Creating backups

Before upgrading your Tools and Frameworks installation, you must back up your Workbench and application configuration and content.

Backing up the Workbench configuration files

Oracle Endeca Workbench uses several configuration files located in `%ENDECA_TOOLS_CONF%\conf` (on Windows) or `$ENDECA_TOOLS_CONF/conf` (on UNIX) to customize the behavior of various aspects of Workbench.

These files store Workbench configuration, user authentication configuration, and definitions of the menus and extensions in Workbench. If you have manually modified any of the following files from their default state, you should copy them to a backup location:

File name	Description
`Login.conf`	Configuration for user authentication using LDAP
`webstudio.properties`	Miscellaneous configuration parameters for Workbench
`webstudio.log4j.properties`	Configuration for the Workbench system log and audit log
`ws-extensions.xml`	Definitions of Workbench extensions
`ws-mainMenu.xml`	Definitions of the Workbench navigation menu and launch page

Backing up the application state

The `webstudiostore` and `emanager` directories contain information about your application state, including user and permission settings, preview application settings, content XML, and resource metadata.

To back up the application state directories:

1. Stop the Endeca Tools Service.
2. Copy the `webstudiostore` directory and its subdirectories from `%ENDECA_TOOLS_CONF%\state\` (on Windows) or `$ENDECA_TOOLS_CONF/state/` (on UNIX) to another location.

   This directory contains information such as users and permissions, as well as preview application settings.
3. Copy the `emanager` directory and its subdirectories from `%ENDECA_TOOLS_CONF%\state\` (on Windows) or `$ENDECA_TOOLS_CONF/state/` (on UNIX) to another location.

   This directory contains resource metadata, state information, and content XML.
4. Start the Endeca Tools Service.

Exporting Workbench users from Tools and Frameworks 11.0

You can use the `export_users` script in Tools and Frameworks 11.0 to export user information from your 11.0 Workbench installation.

To export Workbench users from Tools and Frameworks 11.0:

1. Configure the `export_users` script:
   a) Navigate to the `ToolsAndFrameworks\11.0.0\admin\conf` directory.

b) Open the `export_users.properties` file in a text editor.

c) Set the following properties:

Property	Description
`source.version`	The version of Tools and Frameworks you are migrating from.
`source.workbench.host`	The host machine of the Workbench instance you are exporting from.
`source.workbench.port`	The port of the Workbench instance you are exporting from.
`destination.directory`	The directory of the Workbench instance you are migrating to.

For example:

```
source.version = 11.0.0
source.workbench.host=myHost.myDomain.com
source.workbench.port=8006
destination.directory=C:/Endeca/ToolsAndFrameworks/11.1.0
```

d) Save and close the file.

2. Navigate to the `ToolsAndFrameworks\11.0.0\admin\bin` directory.
3. Run the `export_users` script with the following parameters:
   - `--config` — Required. The path to the `export_users.properties` file.
   - `--output` — Optional. The path of the output file. If you do not specify this option, the output file is `.\user_<timestamp>.json`.

   For example:

   ```
   > export_users.bat --config ..\conf\export_users.properties
   ```

   This step exports users that can be imported into the 11.1 Workbench.

4. Confirm that the JSON file exists and contains the expected users.

   Note the following information as you review the results:

   - All administrator users and groups have an *admin* property set to `true` and do not have permission attributes.
   - LDAP users and groups have a *principalSource* value of `LDAP`. All others have the default value of `Workbench`.

   Sample user :

   ```
   {
       "id": "mmartin",
       "firstName": "melanie",
       "lastName": "martin",
       "email": "mmartin@example.com",
       "principalSource": "WORKBENCH",
       "admin": false,
       "permissions": [
           {
               "application": "Discover",
                   "tools": [
                       "extension1"
                   ],
           }
       ]
   }
   ```

Sample group:

```
{
    "id": "global",
    "groupName": "global merchandising",
    "email": "global@example.com",
    "principalSource": "LDAP",
    "permissions": [
        {
            "application": "Discover",
            "tools": [],
        }
    ]
}
```

You can check the script log at `Endeca\ToolsAndFrameworks\11.0.0\admin\logs\export_users.log`.

Backing up application content from Workbench

You back up application content in Workbench using the `export_site` script provided with the Deployment Template.

The script exports application content in a format that can be re-imported to Workbench. The script connects to the Workbench instance for the current application based on the configuration in `AppConfig.xml`.

Important: To guarantee consistent data, ensure that no baseline or partial updates are running during the backup process.

To back up application content in Workbench:

1. Navigate to the `control` directory of your deployed application, for example, `C:\Endeca\apps\Discover\control`.
2. Run the `export_site` script, passing in an optional name for the export file, as in the following examples:

   ```
   export_site.bat C:\migration\Discover\ECR-backups\discover-21-04-2014.xml
   ```

   On UNIX:

   ```
   ./export_site.sh ../migration/Discover/ECR-backups/discover-21-04-2014.xml
   ```

   If no file name is provided, it defaults to a file named according to the pattern `<appname>-yyyy-mm-dd_time.xml` in the working directory.

Repeat this procedure for every application that you are migrating.

Backing up Forge pipeline configuration

Back up the Forge pipeline configuration for each application that you have deployed. The Forge configuration is stored in the `<app dir>\config\pipeline` directory of your deployed application on disk.

To back up Forge pipeline configuration, copy the `<app dir>\config\pipeline` directory and its contents to another location. This directory contains data ingest pipeline configuration for search interfaces, navigable dimensions, and other configuration authored with Developer Studio.

Backing up CAS configuration

If you are using Oracle Commerce Content Acquisition System (CAS), you must back up the `<app dir>\config\cas` and `<app dir>\config\mdex` directories. You must do this for each application that you have deployed.

These directories contain MDEX sort configuration, search character configuration, and other information.

Backing up the application directory

Before installing Tools and Frameworks 11.1, you should back up your application directory, for example: `C:\Endeca\apps\Discover`. You must do this for every application that you have deployed.

While these files are not removed when uninstalling a previous version of Tools and Frameworks and upgrading to version 11.1, you should have a backup available since they are used as migration source files for the migration scripts.

To back up the application directory:

1. Locate the application in your Endeca application directory.
   For example, `C:\Endeca\Apps\Discover`.
2. Copy the application folder and its contents to a temporary directory that is outside the Endeca installation directory.
   For example, you might copy it to `C:\migration\Discover`.
3. Open a command prompt window and remove the application from the EAC Central Server by running the `eaccmd` utility and the `remove-app` command.
   For example, `eaccmd remove-app --force --app Discover`
4. Delete the application directory from your Endeca application directory.

Upgrading the MDEX Engine to 6.5.1

After backing up Workbench and application configuration and content, upgrade the Oracle Commerce MDEX Engine to 6.5.1.

See the *Oracle Commerce MDEX Engine Migration Guide 6.5.1* and perform the tasks in "Upgrading th MDEX Engine."

Upgrade Platform Services to 11.1

After upgrading your MDEX Engine, uninstall your pre-11.1 version of Platform Services and install Oracle Commerce Guided Search Platform Services 11.1.

See the *Oracle Commerce Guided Search Platform Services Migration Guide 11.1* and perform the tasks in "Upgrading Endeca Platform Services to Version 11.1."

Installing Tools and Frameworks 11.1

After upgrading to Platform Services 11.1, uninstall your Tools and Frameworks 11.0 installation and install Tools and Frameworks 11.1.

To install Tools and Frameworks 11.1:

1. Stop your 11.0 Tools and Frameworks installation:

Option	Description
On Windows:	Stop the Endeca Tools Service or run the `ToolsAndFrameworks\11.0.0\server\bin\stop.bat` file.
On UNIX:	Go to `ToolsAndFrameworks/11.0.0/server/bin` and open `workbench-init-d.sh` in an editor. Update the `ENDECA_TOOLS_ROOT` to your server and also `ENDECA_USER` to the system user and save the file. Run the `workbench-init-d.sh` shell script with the `/sbin/service workbench stop` command.

2. Uninstall Tools and Frameworks 11.0. See the *Tools and Frameworks Installation Guide* and perform the tasks perform the tasks in "Uninstalling Oracle Endeca Tools and Frameworks".
3. Install Tools and Frameworks 11.1. See the *Tools and Frameworks Installation Guide* and perform the tasks in "Installing Oracle Endeca Tools and Frameworks on Windows" or "Installing Oracle Endeca Tools and Frameworks on UNIX".

Restoring Workbench configuration

After updating to Tools and Frameworks 11.1, you can restore your Workbench configuration from backups. You must also restore the application state directories.

If you plan to use the Endeca Tools Service on Windows or the `workbench` service on UNIX to manage the Workbench application container, follow the steps in the *Tools and Frameworks Installation Guide* to install it before proceeding further.

Restoring a backup of the Workbench configuration files

You can restore your Workbench configuration directory, `%ENDECA_TOOLS_CONF%\conf`, by merging in changes from your backup files into the new installation.

The steps below assume the Endeca Tools Service (on Windows) or the `workbench` service (on UNIX) is configured and running.

After updating to Tools and Frameworks 11.1, you can restore your Workbench configuration from backups. This includes Workbench extensions, menu configuration, and authentication settings.

Note: User profiles must be re-created manually.

To restore a backup of the Workbench configuration files:

1. Stop the Endeca Tools Service.
2. Open your configuration backup files.

   These are the files created from *Backing up the Workbench configuration files* on page 13.
3. Manually merge any configuration changes into the files located at `%ENDECA_TOOLS_CONF%\conf` (on Windows) or `$ENDECA_TOOLS_CONF/conf` (on UNIX).
4. Save and close the files.
5. Start the Endeca Tools Service.

Restoring a backup of the application state

Restore backups of the application state directories, `webstudiostore` and `emanager`, by copying them into the new installation of Workbench.

To restore backups of the application state directories:

1. Stop the Endeca Tools Service.
2. Delete the `webstudiostore` directory from `%ENDECA_TOOLS_CONF%\state\` (on Windows) or `$ENDECA_TOOLS_CONF/state/` (on UNIX).
3. Copy the backup of the `webstudiostore` directory, including all its subdirectories, to `%ENDECA_TOOLS_CONF%\state` (on Windows) or `$ENDECA_TOOLS_CONF/state/` (on UNIX).
4. Delete the `emanager` directory from `%ENDECA_TOOLS_CONF%\state\` (on Windows) or `$ENDECA_TOOLS_CONF/state/` (on UNIX).
5. Copy the backup of the `emanager` directory and its subdirectories to `%ENDECA_TOOLS_CONF%\state\` (on Windows) or `$ENDECA_TOOLS_CONF/state/` (on UNIX).
6. Start the Endeca Tools Service.

Migrating Endeca applications from 11.0 to 11.1

To migrate an Endeca application from Tools and Frameworks 11.0 to 11.1 you must restore the application configuration from backups and run migration scripts for 11.1.

Deploying the 11.1 application

Use the Deployment Template to deploy an 11.1 version of your application.

To deploy an empty destination application:

1. Open the command prompt.
2. Navigate to `ToolsAndFrameworks\<version>\deployment_template\bin`.
3. From the `bin` directory, run the `deploy` script with the `--app` flag and an argument that specifies the path to the `deploy.xml` deployment descriptor file.
   For example:

   ```
   C:\Endeca\ToolsAndFrameworks\11.1.0\deployment_template\bin>deploy
   --app C:\Endeca\ToolsAndFrameworks\11.1.0\reference\discover-data\deploy.xml
   ```

4. Unless your environment requires you to use different ports, accept the default values during the deployment process. You must specify the same application name that you used in your 11.0 application.

   For details on running the `deploy` script, see the *Oracle Commerce Administrator's Guide*.

Repeat this procedure for all of your applications.

Restoring Forge pipeline configuration

You can restore Forge pipeline configuration by copying the backup of the `config\pipeline` directory over the `<app dir>\config\pipeline` directory for all of your migrated applications.

Restoring the CAS configuration.

If you are using CAS, you can restore the CAS configuration for each application. Copy the backups of the `<app dir>\config\cas` and `<app dir>\config\mdex` directories over the `<app dir>\config\pipeline` `<app dir>\config\cas` and `<app dir>\config\mdex` directories of your migrated applications.

Initializing the 11.1 application

Run the `initialize_services` script to provision your application in the Endeca Application Controller.

To provision your application:

1. Open the command prompt.
2. Navigate to the `control` directory of the deployed application.

   For example: `C:\Endeca\Apps\Discover\control`.
3. Run the `initialize_services` script.

   This provisions the application within the Endeca Application Controller and creates the necessary structure in Workbench.

Repeat this procedure for all of your applications. Deploying and provisioning the application enables you to upload the modified export file to the Workbench.

Migrating Workbench users to Oracle Commerce Single Sign-On

If you are migrating Workbench LDAP users to Oracle Commerce Single Sign On (SSO), you must change your exported user JSON files to specify Commerce as the data source and create corresponding LDAP organizations in the Oracle Commerce Platform.

Commerce SSO integrates authentication for Workbench and the Oracle Commerce Business Control Center, allowing a user to switch between tools without encountering additional login screens. If you plan on migrating to Commerce SSO, you must store users and groups in the Oracle Commerce Platform internal profile repository.

1. Navigate to the output JSON file that was created when you exported user information from your 11.0 Workbench installation when you performed the tasks in *Exporting Workbench users from Tools and Frameworks 11.0* on page 13.

   For example,

   ```
   ToolsAndFrameworks\11.0.0\admin\user_20140514.json
   ```

2. Open the JSON file in an editor and replace all the instances where *LDAP* appears as the `principalSource` value with `COMMERCE` for all groups and users.

   For example, this is an updated sample group:

   ```
   {
       "id": "global",
       "groupName": "global merchandising",
       "email": "global@example.com",
       "principalSource": "COMMERCE",
       "permissions": [
           {
               "application": "Discover",
               "tools": [],
   ```

```
        }
    ]
```

This is an updated sample user:

```
{
    "id": "mmartin",
    "firstName": "melanie",
    "lastName": "martin",
    "email": "mmartin@example.com",
    "principalSource": "COMMERCE",
    "admin": false,
    "permissions": [
        {
            "application": "Discover",
                "tools": [
                    "extension1"
                ],
        }
     ]
}
```

3. Make a note of every group ID in the `id` attribute.
4. Save and close the JSON file.
5. In the Oracle Commerce Business Control Center create corresponding LDAP organizations that match the 11.0 Workbench LDAP groups.
    a) Log in to the Business Control Center.
    b) Click **Administration** > **Access Control**.
    c) Click the arrow next to **Organizations**.
    d) Click the plus sign (+) and then **Organization**.
    e) In the New Organization section, under **Source**, click **LDAP**.
    f) In the **Name** field, enter the ID of your first Workbench LDAP group exactly as it appears in your exported user JSON file.
    g) Validate the organization.
    h) Click **Create**.

    See the *Oracle Commerce Business Control Center Administration and Development Guide* for more information on configuring LDAP organizations.

Next, you can import your users into Tools and Frameworks 11.1.

Importing 11.0 Workbench users to Tools and Frameworks 11.1

You can use the `import_users` script to import user information from a JSON format to your Workbench installation.

To import 11.0 Workbench users to Tools and Frameworks 11.1:

1. Configure the `import_users` script:
    a) Navigate to the `ToolsAndFrameworks\11.1.0\admin\conf` directory.
    b) Open the `import_users.properties` file in a text editor.
    c) Set the `dest.workbench.host` and `dest.workbench.port` properties to the host and port of your Tools and Frameworks 11.1 Workbench.

        By default, these values are set to `localhost` and `8006`, respectively.

For example:

```
dest.workbench.host=myhost.mycompany.com
dest.workbench.port=8006
```

d) Set the `abort.on.duplicate.users` Boolean property:

- `true` — If duplicate users are detected, the script stops and creates a `duplicate_users.log` log file that lists all duplicate user names.
- `false` — If duplicate users are detected, the script attempts to merge them into a single entry.

For example:

```
abort.on.duplicate.users=false
```

e) Comment out the `tools.mapping.file` property:

```
#-------- location of tools mapping (required if source.version 2.1.x & 3.1.0)
#tools.mapping.file=<path-to-tools-mapping-json>
```

f) Optionally, set the `convert.ldap` property to `true` if you wish to convert all users to LDAP users during import.
For example:

```
convert.ldap=true
```

g) Save and close the file.

2. Navigate to the `ToolsAndFrameworks\11.1.0\admin\bin` directory.
3. Run the `import_users` script with the following parameters:

   - `--input` — Required. The path to the user data file that you got from running the `export_users` script in 11.0.0.
   - `--config` — Requred. The path to the `import_users.properties` file.
   - `--default-user-password` — The password value for any users that do not have a password set in the exported users file.
   - `--single-app` — Optional. A single application for which to import users.

   For example:

   ```
   > import_users.bat --input user_20140514.json --config ..\conf\import_users.prop¬
   erties
   --default-user-password CHANGEME --single-app Discover
   ```

   If a name matches an existing name but with characters in a different case (JOHN_Doe and John_Doe) and the `abort.on.duplicate.users` property is set to `true`, the script stops. Invalid entries are logged to `ToolsAndFrameworks\<vesrion>\admin\logs\import_validation_failed.log`. The main log file is output to `ToolsAndFrameworks\<version>\admin\logs\import_users.log`.

Configuring the migrate-workbench script

Before running the `migrate-workbench` script, you must modify the `config.properties` file with source and destination directories for each application that you are migrating.

Oracle Commerce Experience Manager 11.1 lets you build an application that can run multiple web sites using a single index. Even if you do not deploy multiple sites, your 11.1 application must contain at least one site. This site contains all the pages for your application - for both Oracle Commerce Guided Search and Oracle Commerce Experience Manager. The `migrate-workbench` script creates a site for your application and moves all your application pages into the site.

To configure the `migrate-workbench` script:

1. Navigate to `ToolsAndFrameworks\<version>\migration\workbench\11.0.0-11.1.0\conf`.
2. Open `config.properties` in a text editor.
3. Set `endeca.source.exportedSiteFile` to the exported site file previously created from running the `export_site` script on your 11.0 application.

   Note: When you enter values for settings in the `config.properties` file, you must use forward slashes in the directory paths - even for Windows.

   For example, `C:/migration/Discover/ECR-backups/discover-21-04-2014.xml`.
4. Set `endeca.target.exportedSiteFile` the output file where you wish to store the migrated site content.
   For example, `C:/migration/Discover/11.1-export.xml`.
5. You can optionally configure the following properties for the single site created by the migration script.

Property	Description
`endeca.target.defaultSite.id`	The ID of the site that the Assembler uses when a request cannot be matched to any other site. The default value is the EAC application name. Do not enter a forward slash '/' as the first character in the value. The script inserts that character automatically.
`endeca.target.defaultSite.displayName`	The name that displays in Experience Manager. If the display name is omitted, the site ID is used.
`endeca.target.defaultSite.description`	The description that displays in Experience Manager.
`endeca.target.defaultSite.urlPattern`	The pattern used to match the request URL to this site. A pattern can be a domain, like example.com, or a URI, like /summersale.

6. Save and close the file.

After modifying the `config.properties` file, you can optionally configure logging for the script.

Configuring logging for the migrate-workbench script

You can configure the log level and output directory for the `migrate-workbench` logs by modifying the `logging.properties` file.

The `migrate-workbench` script logs its migration operations. By default, logs are stored in a `migration.log` file within the directory from which you execute the migration script.

To configure logging for the `migrate-workbench` script:

1. Navigate to `ToolsAndFrameworks\<version>\migration\workbench\11.0.0-11.1.0\conf`.
2. Open `logging.properties` in a text editor.
3. Set `java.util.logging.FileHandler.pattern` to the desired logging output directory.
4. Set `java.util.logging.FileHandler.level` to the desired log level.

   The script provides information at the following logging levels:

   - `SEVERE` — Logs errors that terminate the migration process.
   - `WARNING` — Logs information about problems during migration that are not severe enough to stop the process, but may require action from the user to repair.
   - `INFO` — Logs the status of the migration process.

- FINER — Logs additional information about the migration process, such as configuration information and template metadata.
- FINEST — Logs verbose information about the migration process, such as source and target content XML and template files.

For additional information, see the Oracle documentation for the java.util.logging.Level class.

5. Save and close the file.

Running the migrate-workbench script

Once the migrate-workbench script is configured, you can run it to convert application content to the 11.1 format.

To migrate Workbench configuration:

1. Open a command prompt window and navigate to Endeca\ToolsAndFrameworks\<version>\migration\workbench\11.0.0-11.1.0\bin .
2. Run the migrate-workbench batch or shell script.

   The script logs a message to the console indicating successful migration and the location of the output files.
3. Navigate to the output paths and confirm an export XML file is present.

Repeat this procedure for every application that you are migrating.

Uploading migrated application content and templates to Workbench

Running the import_site script uploads your migrated application content to Workbench.

Important: To guarantee consistent data, ensure that no baseline or partial updates are running during this process.

To upload migrated application content in Workbench:

1. Navigate to the control directory of your deployed application.

   For example, C:\Endeca\apps\Discover\control.
2. Run the import_site script, passing in the file name of the modified export file produced by the migrate-workbench script.

   For example:

   ```
   C:\Endeca\apps\Discover\control>import_site.bat
   C:\migration\Discover\11.1.0-export.xml
   ```

   The following prompt appears:

   ```
   "Application '<app name>' already exists in IFCR.
   Delete existing content and continue? [Y/N]:"
   ```
3. Enter Y to proceed.
4. Confirm that the import operation completes successfully.

Repeat this procedure for every application that you are migrating. After updating your Workbench content, you must move packaged services for Oracle Commerce Experience Manager deployments.

Moving packaged services for Experience Manager deployments

If you use Oracle Commerce Experience Manager you must move the packaged services to /pages.

- guidedsearch
- recorddetails,
- dimensionsearch

Skip these steps if you use Rule Manager.

1. Navigate to the control directory of your deployed application.

   For example, C:\Endeca\apps\Discover\control.

2. Run the export application script to export your migrated application.

   For example:

   ```
   c:\Endeca\apps\Discover\control>runcommand.bat IFCR exportApplication c:\Ende¬
   ca\apps\Discover11.1-export true
   ```

3. Create a services folder under /pages/<Site Id> and add the following attributes to a _.json file in the /pages/<Site Id>/services directory.

   ```
   {
       "contentType": "Page",
       "ecr:type": "page"
   }
   ```

   <Site Id> is the site generated by the migrate_workbench script.

4. Move all the packaged services files from /services/<App Name> folder to /pages/<Site Id>/services.
5. Run the import application script to import the updated application.

   For example:

   ```
   c:\endeca\apps\Discover\control>runcommand.bat IFCR importApplication c:\ende¬
   ca\apps\Discover11.1-export
   ```

Although the packaged services for Experience Manager deployments have moved from the /services page root to a /services page in /pages, the URL for accessing packaged services as a JSON response has not changed. For example: http://localhost:8006/assembler/json/services/guidedsearch.

The packaged services are visible in the left navigation in the Experience Manager interface. Oracle recommends that you use the User Management tool in Workbench to make these pages *read-only* by not granting any users write permission to these pages. Since guided search pages do not have a content template, an "Invalid content item XML could not be loaded" message appears when you try to view them in Experience Manager.

You cannot preview these pages unless you add a content template

Repeat this procedure for every application that you are migrating. Afterwards, you must run a baseline update to index data and start the Dgraph for your newly-deployed application.

Running a baseline update

After you have deployed your application and uploaded the migrated content and configuration, you must initialize and start the Dgraph to make MDEX Engine data available.

To run a baseline update:

1. Navigate to the `control` directory of your deployed application.

   For example, `C:\Endeca\apps\Discover\control`.
2. Run the `load_baseline_test_data` script.
3. Run the `baseline_update` script.

Repeat this procedure for every application that you are migrating.

Verifying a migrated application

Once you have finished migrating Workbench and application configuration, log in to Workbench to confirm that everything is present and functioning as expected.

To verify that your application migrated correctly:

1. Navigate to the URL for your application and confirm that it is running.

   For example, the authoring version of the reference application is typically available from `http://localhost:8006/discover-authoring`.
2. Log in to Workbench and verify that you can see thesaurus entries.
3. If the 11.1.0 host is different from the 11.0.0 host that you migrated from, go to **Application Settings - Preview Settings** and update the Preview URL and Link Service URL information.
4. Open Experience Manager and confirm the following:
   - Pages display within a site in the Site Pages section.
   - Application content is present (content items).
   - The preview application displays as expected.
5. If your application has been successfully migrated, you may optionally promote the content and configuration to your live servers.

   In the Discover Electronics reference application, this is accomplished by running the `promote_content` script in the `control` directory.

Repeat this procedure for every application that you are migrating. If you encounter any issues with migration, see *Troubleshooting migration errors* on page 25.

Troubleshooting migration errors

The `migrate-workbench` script outputs error messages on failure. Refer to the table below if you are encountering issues running the script.

Error	Action
Could not find `config.properties`	Make sure that a `config.properties` file exists in the `migration\workbench\11.0.0-11.1.0\conf` directory of your Tools and Frameworks installation.

Error	Action
Could not find `logging.properties`	Make sure that a `logging.properties` file exists in the `migration\workbench\11.0.0-11.1.0\conf` directory of your Tools and Frameworks installation.
Source export file path cannot be empty.	Make sure that you have entered a value for the `endeca.source.exportedSiteFile` property in `config.properties`.
Source export file must exist.	Make sure that the `endeca.source.exportedSiteFile` property is set to the exported 11.0 XML file.
Target export file path cannot be empty.	Make sure that you have entered a value for the `endeca.target.exportedSiteFile` property in `config.properties`.
Target export file already exists.	Make sure that the `endeca.target.exportedSiteFile` property is set to a file path that does not already exist.
Source and target export file paths cannot be same.	Make sure that `endeca.source.exportedSiteFile` and `endeca.target.exportedSiteFile` are not set to the same path.
Error while parsing `appConfig` file.	Make sure that the `appConfig.xml` file in `{appDir}/config/` script is correct XML
Cannot find `appConfig` file.	Make sure that the `appConfig.xml` file is in `{appDir}/config/script` directory.
Cannot read appConfig file.	Make sure that the migrator has permission to read the appConfig file.

Chapter 3

Required Changes

The following section lists required changes as of version 11.1. You must make the changes specified in this section if the changes apply to your application.

Deprecated features

The features listed below are deprecated in the Tools and Frameworks 11.1 release. They are still fully supported.

Assembler repackaging

As part of the 11.1 release, the Assembler has been repackaged into two JAR files, `endeca_assembler_core-<version>.jar` and `endeca_assembler_navigation-<version>.jar`.

Updating Configuration for AbstractAssemblerServlet

For any application that runs Assembler as a service, you must update the configuration for SpringAssemblerServlet (which implements AbstractAssemblerServlet).

To do this, you must modify both the application's `web.xml` configuration file and its Spring configuration.

Updating web.xml

If your application uses SpringAssemblerServlet, you must update the application's `web.xml` configuration file to include the new `contentItemId` property,and to remove the `enableKeywordRedirects` property; for example:

```
<servlet>
  <servlet-name>JsonAssemblerServiceServlet</servlet-name>
  <servlet-class>com.endeca.infront.assembler.servlet.spring.SpringAssem¬
blerServlet</servlet-class>
   ....
  <init-param>
    <param-name>contentItemId</param-name>
    <param-value>rootContentItem</param-value>
  </init-param>
```

```
</servlet>
   ....
<servlet>
  <servlet-name>XmlAssemblerServiceServlet</servlet-name>
  <servlet-class>com.endeca.infront.assembler.servlet.spring.SpringAssem¬
blerServlet</servlet-class>
   ....
  <init-param>
    <param-name>contentItemId</param-name>
    <param-value>rootContentItem</param-value>
  </init-param>
</servlet>
```

Configuring the Root Content Item

If your application uses SpringAssemblerServlet, you must update the application's Spring configuration to include configuration for the root content item with which to begin assembly, such as RedirectAwareContentInclude; for example:

```
<!-- BEAN:RootContentItem  Specify the root content item to be processed for the
 Assembler service -->
<bean id="rootContentItem"
      class="com.endeca.infront.cartridge.RedirectAwareContentInclude"
      scope="request">
  <constructor-arg ref="pathInfo"/>
</bean>

<bean  id="pathinfo"
       scope="request"
       factory-bean="httpServletRequest"
       factory-method="getPathInfo" />
```

Preview application changes

In previous releases of Tools and Frameworks, business users were only able to audit in the preview application. Now business users can both audit and edit in the preview application. You must make the follow changes to support editing in the preview application.

These required changes are for applications like the Discover Electronics reference application that use JSP files for cartridge renderers. If you have used a different technology stack to implement your Assembler application, you must update your own preview functionality to support editing. See *Preview application changes for non-Java applications* on page 30.

New previewAnchor tag

In previous releases of Tools and Frameworks, slots were wrapped with an `<endeca:includeSlot />` tag that contained preview information. In 11.1, you must use `<endeca:previewAnchor />` instead. In this release, you must not only wrap slots, but also all other content items. Oracle recommends that rather than modifying every renderer, that you have a centralized place where content items are dispatched. In the Discover Electronics reference application, this is done in the `include.tag`. For example, here is `include.tag` in a previous release:

```
<%-- save the parent's component currently in request scope into page scope --%>
<c:set var="parentComponent" scope="page" value="${requestScope['component']}"/>
<%-- set the content item the child will use as this one (the one passed in the
tag) --%>
<c:set var="component" scope="request" value="${component}"/>
```

```
<c:catch var="importException">
    <c:import url="${resourcePath}" charEncoding="UTF-8"/>
</c:catch>
```

Here is include.tag in 11.1 with the new previewAnchor tag:

```
<%-- save the parent's component currently in request scope into page scope --%>
<c:set var="parentComponent" scope="page" value="${requestScope['component']}"/>
<%-- set the content item the child will use as this one (the one passed in the
tag) --%>
<c:set var="component" scope="request" value="${component}"/>
<c:catch var="importException">
    <endeca:previewAnchor contentItem="${component}">
        <c:import url="${resourcePath}" charEncoding="UTF-8"/>
    </endeca:previewAnchor>
</c:catch>
```

Updated pageBody tag

You must modify the input of the endeca:pageBody tag. In the 11.1 release, endeca:pageBody requires the root and current content item attributes. Here is an example of endeca:pageBody in a previous release where only the root content item attribute was required.:

```
<script type="text/javascript" src="<c:url value='/js/global.js'/>"></script>
</head>
<body>
    <endeca:pageBody rootContentItem="${rootComponent}">
        <div class="PageContent">
            <%--include user panel --%>
            <%@ include file="/WEB-INF/views/userPanel.jsp" %>
```

Here is endeca:pageBody in 11.1 with the additional current content item attribute.

```
<script type="text/javascript" src="<c:url value='/js/global.js'/>"></script>
</head>
<body>
    <endeca:pageBody rootContentItem="${rootComponent}" contentItem="${component}">

        <div class="PageContent">
            <%--include user panel --%>
            <%@ include file="/WEB-INF/views/userPanel.jsp" %>
```

Assembler API changes

There is one new class in 11.1 that was described in the previous sections: PreviewAnchor.java. This class wraps content items in a <div> tag that contains preview information.

The following classes have been deprecated:

- IncludeSlot.java

  Replaced by PreviewAnchor.java but it can still be used to deliver the new output for slots.

- PageBody.java

  Replaced by pageBody.tag.

Assembler Factory changes

The previewModuleUrl property in the Assembler context file, `WEB-INF\assembler-context.xml` no longer points to `http://${workbench.host}:${workbench.port}/preview`. In 11.1, it takes `http://${workbench.host}:${workbench.port}/ifcr`:

```
<!--
    ########################################################################
    # ASSEMBLER FACTORY
    #
    # Required.
    #
-->
<bean id="assemblerFactory"
class="com.endeca.infront.assembler.spring.SpringAssemblerFactory"
scope="singleton">
    <constructor-arg>
        <bean class="com.endeca.infront.assembler.AssemblerSettings">
            <property name="previewEnabled" value="${preview.enabled}" />
            <property name="previewModuleUrl" value="http://${work¬
bench.host}:${workbench.port}/ifcr" />
        </bean>
    </constructor-arg>
    <constructor-arg>
        <list>
            <bean class="com.endeca.infront.logger.SLF4JAssemblerEventLogger" />

        </list>
    </constructor-arg>
</bean>
```

Preview application changes for non-Java applications

You must make the following changes to support editing in preview for applications that do not use JSP filles for cartridge renderers.

In 11.1, the Assembler cartridges output now includes the repository path and inner path. For slots and in audit mode, `ruleFireData` is also included. The following properties of a content item have been moved in 11.1:

- `contentPaths` is now included in the `endeca:auditInfo` object and has been renamed `ecr:contentSlotContentPaths`.
- `templateTypes` is now included in the `endeca:auditInfo` object and it has been renamed `ecr:contentSlotTemplateTypes`.
- `templateIds` is now included in the `endeca:auditInfo` object and it has been renamed `ecr:contentSlotTemplateIds`.

The Experience Manager preview framework automatically loads any tag (for example, `h3`, `div`, `ul`) that includes the `data-oc-audit-info` attribute. The expected contents of `data-oc-audit-info` are shown in the following example. All of the expected properties are included inside the `endeca:auditInfo` object in the Assembler response. Clients can serialize the contents and add them to the container tag of the content item.

```
<div data-oc-audit-info="
    {
        ecr:resourcePath: "content/Web/General/Pages/Home",
        ecr:resourceInnerPath: "secondaryContent[1]",
        ecr:contentSlotRuleFireData: [
            {
                endeca.internal.created.by: "admin",
                rule.msg: "Fired",
                endeca.internal.templateType: "SecondaryContent",
                endeca.internal.templateId: "GuidedNavigation",
                endeca.internal.last.modified.by: "admin",
                rule.msgcode: "13",
                rule.state: "fired",
                endeca.internal.resource.path: "/content/Shared/Guided Naviga¬
tion/Default Guided Navigation",
                endeca.internal.workflow.state: "ACTIVE",
                type: "DGraph.MerchDebug",
                Title: "Default Guided Navigation",
                endeca.internal.priority: "30",
                rule.id: "12",
                endeca.internal.last.modified: "2014-01-16T12:01",
                endeca.internal.date.created: "2014-01-16T12:01",
                Zone: "xmgr_zone"
            }
        ],
        ecr:name: "Guided Navigation Slot",
        ecr:contentSlotContentPaths: [ "/content/Shared/Guided Navigation" ],
        ecr:contentSlotTemplateTypes: [ "SecondaryContent" ],
        ecr:contentSlotTemplateIds: [ ],
    }">
</div>
```

Add the following information to the head tags in your non-Java application:

```
<head>
    <!-- beginning imports from PageHead tag -->
    <script type="text/javascript" src="http://@@WORKBENCH_HOST@@:@@WORK¬
BENCH_PORT@@/ifcr/tools/xmgr/app/preview/js/jquery-1.6.2.min.js"></script>
    <script type="text/javascript">
        if (typeof(Endeca) == "undefined") Endeca = {};
        Endeca.jQuery = jQuery.noConflict(true);
    </script>
    <script type="text/javascript" src="http://@@WORKBENCH_HOST@@:@@WORK¬
BENCH_PORT@@/ifcr/tools/xmgr/app/preview/js/util/util.js"></script>
    <script type="text/javascript" src="http://@@WORKBENCH_HOST@@:@@WORK¬
BENCH_PORT@@/ifcr/tools/xmgr/app/preview/js/util/json.js"></script>
    <script type="text/javascript" src="http://@@WORKBENCH_HOST@@:@@WORK¬
BENCH_PORT@@/ifcr/tools/xmgr/app/preview/js/urlgen/urlgen.js"></script>
    <script type="text/javascript" src="http://@@WORKBENCH_HOST@@:@@WORK¬
BENCH_PORT@@/ifcr/tools/xmgr/app/preview/js/iframe-site/iframe-site.js"></script>

    <script type="text/javascript" src="http://@@WORKBENCH_HOST@@:@@WORK¬
BENCH_PORT@@/ifcr/tools/xmgr/app/preview/js/framework/framework.js"></script>
    <script type="text/javascript" src="http://@@WORKBENCH_HOST@@:@@WORK¬
BENCH_PORT@@/ifcr/tools/xmgr/app/preview/js/plugins/endeca/endeca.js"></script>
    <link rel="stylesheet" href="http://@@WORKBENCH_HOST@@:@@WORK¬
BENCH_PORT@@/ifcr/tools/xmgr/app/preview/css/audit-site.css">
    <!-- end of imports from PageHead tag -->
    <!-- Setting the page content URI for the audit rules -->
    <script type="text/javascript">
        if(Endeca.Site) {
            Endeca.Site.Iframe.setContentUri(@@CONTENT_PATH@@, @@SITE_NAME@@);
            Endeca.Framework.configure('ecr', 'contentUri', @@CONTENT_PATH@@);
        }
    </script>
    <!-- End of setting the content URI -->
</head>
```

Where:

- `@@CONTENT_PATH@@` is the location of the content, for example `/pages/DiscoverElectronics/browse`.
- `@@SITE_NAME@@` is the name of the site, for example `DiscoverElectronics`.
- `@@WORKBENCH_HOST@@`is the hostname where workbench resides.
- `@@WORKBENCH_POST@@` is the port where workbench resides.

If you have implemented preview in a non-Java application in a previous release, you must remove the following Javascript from the content slots renderer

```
<div class="endeca-slot" auditId="3">
  <script type="text/javascript">
    if (Endeca.Site) {
      Endeca.Site.Audit.contents["3"] = {/* Audit Info */ }
    }
  </script>
  <!-- renderer output for the slot's content goes here -->
</div>
```

Managing Oracle Commerce Platform user segments

If you are migrating an application that uses Oracle Commerce Platform user segments in Tools and Frameworks Workbench, you must either disable security or add a new credential for user segment security to work in 11.1. Of these two options, Oracle recommends adding a new credential.

To disable security, which Oracle does not recommend doing, set the `enabled` property to *false* on the `RequestCredentialAccessController` component. See "About the RequestCredentialAccessController Component" in the *Oracle Commerce Platform-Guided Search Integration Guide* for more information.

To add a new credential, you must add credentials to both the Oracle Commerce Platform and Workbench. See "Managing Credentials" in the *Oracle Commerce Platform-Guided Search Integration Guide* for more information.

Import locations for media, editor configuration, and cartridge templates

Media, editors configuration, and cartridge templates import content into the ECR from the following folders:

Content	Imported from
editor configuration	`<app>/config/ifcr/configuration/tools/xmgr`
media	`<app>/config/ifcr/media`
cartridge templates	`<app>/config/import/templates`

You can export content to locations other than the ECR using `legacyExportContent` (for editor configuration and media), or `exportContent` (for cartridge templates). For more information about these commands, refer to the *Oracle Commerce Administrator's Guide*.

Chapter 4

Behavioral Changes

This section describes changes in the 11.1 release that do not require action on your part, but do have an effect on how your Endeca application behaves after you upgrade.

Changes to Behavior of Export Scripts

This release introduces the following changes to the behavior of get_templates, get_media, and get_editors_config scripts:

- The `get_templates`, `get_media`, and `get_editors_config` scripts overwrite the existing content of the directories to which they export configuration data.
- Media, and configurations of previews, services, and editors must be placed under the `/config/ifcr` directory of the deployed application. These types of content are imported by the legacyUpdateContent command.
- Content, pages, phrases, redirects, templates, thesaurus, userSegments must be placed under the `/config/import` directory of the deployed app. These types of content are imported by the importContent command.

For information about legacyUpdateContent and importContent, refer to the *Oracle Commerce Guided Search Administrator's Guide*

Changes to Contents of Assembler Jars

New packages have been added and and previously existing packages have been deprecated to accommodate the division of Assembler functionality into two jars, a Core jar and a Navigation jar. No new classes have been added, however, and the functionality of existing classes remains unchanged.

Package Changes in the Core Jar

A number of previously existing packages have been moved to the Core jar, where they are now deprecated. The deprecated packages are:

```
com.endeca.infront.cartridge.ContentInclude
com.endeca.infront.cartridge.ContentIncludeHandler
com.endeca.infront.cartridge.ContentSlotConfig
com.endeca.infront.cartridge.ContentSlotHandler
com.endeca.infront.cartridge.ContentSlotListConfig
com.endeca.infront.cartridge.ContentSlotListHandler
com.endeca.infront.cartridge.ConfigInitializer
com.endeca.infront.cartridge.RequestParamMarshaller
com.endeca.infront.navigation.UserState
```

The following new packages have been added to the Core jar:

```
com.endeca.infront.content.cartridge.ContentInclude
com.endeca.infront.content.cartridge.ContentIncludeHandler
com.endeca.infront.content.cartridge.ContentSlotConfig
com.endeca.infront.content.cartridge.ContentSlotHandler
com.endeca.infront.content.cartridge.ContentSlotListConfig
com.endeca.infront.content.cartridge.ContentSlotListHandler
com.endeca.infront.content.cartridge.ConfigInitializer
com.endeca.infront.content.cartridge.RequestParamMarshaller
com.endeca.infront.content.UserState
```

Package Changes in the Navigation Jar

A number of previously existing packages have been moved to the Navigation jar, where they are now deprecated. The deprecated packages are:

```
com.endeca.infront.content.MdexContentRequestBroker
com.endeca.infront.assembler.servlet.AbstractPreviewLinkServlet
com.endeca.infront.assembler.servlet.spring.SpringPreviewLinkServlet
```

The following new packages have been added to Navigation jar:

```
com.endeca.infront.navigation.MdexContentRequestBroker
com.endeca.infront.navigation.servlet.AbstractPreviewLinkServlet
com.endeca.infront.navigation.servlet.spring.SpringPreviewLinkServlet
```

Updating Applications that Extend or Reference Deprecated Packages

If your applications extend any of the classes in the deprecated packages, it is recommended that you recompile the applications to reference the classes in their new java packages.

If your applications reference any of the classes in the deprecated packages, update these references to point to the java packages to which the deprecated classes have been moved.

Oracle Commerce Workbench

The following changes have been made to Workbench

Pages

In previous releases of the Experience Manager, the navigation tree on the left had a **Pages** section that displayed the pages in your application. In 11.1, Oracle has introduced the capability of adding multiple sites to your application, so this section has been renamed **Site Pages**. You can still find pages by expanding a site node.

Preview

In previous releases of the Experience Manager, when you clicked **Preview**, the preview application opened in a new window. If you wanted to edit an item, you would have to return to Experience Manager and edit the item there. In 11.1, the preview application no longer opens in a new window; it opens in the current window.

You now have the capability of editing content in the preview application. You can still return to Experience Manager and edit content clicking **Data View** in the preview application.

Documentation changes

The following documents have been changed or removed in this release.

The content in the *Deployment Template Usage Guide* has been moved into the *Oracle Commerce Guided Search Administrator's Guide*.

Index