Oracle Commerce Guided Search

Tools and Frameworks Migration Guide

Version 11.2 • October 2015

ORACLE®

COMMERCE

Contents

Copyright and disclaimer

for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

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

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 Commerce Guided SearchTools and Frameworks to the most recent version, and how to migrate an 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 Commerce Guided Search 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 Notes

Refer to the release notes for information about known issues for this release. You can download the *Oracle Commerce Guided Search Release Notes* from the Oracle Technology Network.

Oracle Commerce Guided Search Getting Started Guide

The *Oracle Commerce Guided Search Getting Started Guide* gives an overview of basic 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 Commerce Guided Search Tools and Frameworks 11.1 to Tools and Frameworks 11.2.

Unsupported upgrade paths

If you are migrating to Oracle Commerce Guided Search Tools and Frameworks 11.2 from a version other than those listed above, you must first migrate to the previous release (11.1).

Upgrading on a single host

This guide describes how upgrade an Oracle Commerce Guided Search application on a single machine.

Chapter 2

Migrating Tools and Frameworks from 11.1 to 11.2

The following section lists the steps for migrating from Tools and Frameworks 11.1 to Tools and Frameworks 11.2.

About migrating from 11.1 to 11.2

To migrate your Tools and Frameworks installation from 11.1 to 11.2, 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 backups.

Migrating from Tools and Frameworks 11.1 to 11.2 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. Use the export scripts included with Tools and Frameworks 11.1 to export user information from version 11.1.
   c. Back up Workbench by running the `export_site` script for each application you are migrating.
   d. Back up any Forge data ingest configuration by creating a copy of the `config\pipeline` directory in your application.
   e. If you use CAS, back up `config\CAS` and `config\MDEX` directories.
   f. Optionally, back up each deployed application directory, for example, `C:\Endeca\apps\Discover`.
   g. Stop and uninstall your Tools and Frameworks 11.1 Workbench.

2. Upgrade the MDEX Engine to 6.5.2.
3. Upgrade Platform Services to 11.2
4. Upgrade Tools and Frameworks to version 11.2.
5. Restore Workbench configuration from the backups created in Step 1a.
6. Restore each of your applications:

   a. Deploy the 11.2 version of your application.
   b. If you are using Experience Manager, and you want to use the workspace feature introduced in 11.2, export the deployment settings to a temporary directory.
   c. Restore Forge configuration from backups.
   d. Restore CAS configuration from backups.

> ✎ **Note:** You should see a warning in the console that the Dgraph is down and search term expansions have to be manually updated by running the update_search_term_expansions script.

    e. Initialize your application.

    f. Repeat the substeps above for each application you are migrating.

    g. Restore Workbench user information using the `import_users` script.

    h. Restore Workbench content from the modified export file created in Step 6h by running the `import_site` script.

    i. Run a baseline update to load data into the Dgraph and start it.

    j. If you are upgrading to 11.2 for the first time, run the `update_search_term_expansions` script to add the searchTermExpansions property.

    k. Verify that Workbench configuration and application content are present in Workbench. Optionally, promote content to your live servers.

    l. Repeat the substeps from 6g above for each application you are migrating.

7. If you are using Experience Manager, and you want to use the workspace feature introduced in 11.2, import the deployment settings you exported in Step 6b.

8. Add the `userState` property to the following bean definitions in the Assembler context file for your application, `assembler-context.xml`:

    a. CartridgeHandler_RedirectAwareContentInclude

    b. CartridgeHandler_ContentInclude

    c. NavigationCartridgeHandler

9. In the Assembler context file for your application, `assembler-context.xml`, delete the `ecrStoreFac¬tory` bean. In 11.2, all store factory configurations must use FileStoreFactory. Update all references to the `ecrStoreFactory` bean to `storeFactory`, or the ID used in the `FileStoreFactory` bean.

10. In the Assembler context file for your application, add the `appName` property to the Store Factory bean and the `mdexResource` bean:

```
<property name="appName" value="${workbench.app.name}" />
```

11. Delete the `store.factory` property from the `WEB-INF/assembler.properties` file.

```
store.factory=ecrStoreFactory
```

Upgrading Projects to Developer Studio 11.2

To upgrade an existing project created with Developer Studio 11.1 or earlier to Developer Studio 11.2, follow these steps:

1. Open the project file of the existing project.
2. When prompted to upgrade the project to the new version (11.2), 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 11.2 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=" 1120">
```

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

Workbench uses several configuration files located in `%ENDECA_TOOLS_CONF%\conf` (on Windows) or `$EN¬DECA_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

Exporting Workbench users from Tools and Frameworks 11.1

You can use the `export_users` script in Tools and Frameworks 11.1 to export user information from your 11.1 Workbench installation.

To export Workbench users from Tools and Frameworks 11.1:

1. Configure the `export_users` script:

a) Navigate to the `ToolsAndFrameworks\11.1.0\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 = 11.1.0
source.workbench.host=myHost.myDomain.com
source.workbench.port=8006
destination.directory=C:/Endeca/ToolsAndFrameworks/11.2.0
```

d) Save and close the file.

2. Navigate to the `ToolsAndFrameworks\11.1.0\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 11.2 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\11.1.0\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`.

Note: 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-2015-10-30.xml
   ```

   On UNIX:

   ```
   ./export_site.sh ../migration/Discover/ECR-backups/discover-2015-10-30.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\index_config`, `<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 11.2, you should back up your application directory, for example: `C:\Endeca\apps\Discover`. You must do this for every application that you have deployed.

To back up the application directory:

1. Locate the application in your application directory.
   For example, `C:\Endeca\Apps\Discover`.
2. Copy the application folder and its contents to a temporary directory that is outside the 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 application directory.

Upgrading the MDEX Engine to 6.5.2

After backing up Workbench and application configuration and content, upgrade the Oracle Commerce MDEX Engine to 6.5.2.

See the *Oracle Commerce MDEX Engine Migration Guide 6.5.2* and perform the tasks in "Upgrading th MDEX Engine."

Upgrade Platform Services to 11.2

After upgrading your MDEX Engine, uninstall your pre-11.2 version of Platform Services and install Oracle Commerce Guided Search Platform Services 11.2.

See the *Oracle Commerce Guided Search Platform Services Migration Guide 11.2* and perform the tasks in "Upgrading Platform Services to Version 11.2."

Installing Tools and Frameworks 11.2

After upgrading to Platform Services 11.2, uninstall your Tools and Frameworks 11.1 installation and install Tools and Frameworks 11.2.

To install Tools and Frameworks 11.2:

1. Stop your 11.1 Tools and Frameworks installation:

Option	Description
On Windows:	Stop the Tools Service or run the `ToolsAndFrameworks\11.1.0\server\bin\stop.bat` file.

Option	Description
On UNIX:	Go to `ToolsAndFrameworks/11.1.0/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 11.1. See the *Tools and Frameworks Installation Guide* and perform the tasks in "Uninstalling Oracle Commerce Guided Search Tools and Frameworks".
3. Install Tools and Frameworks 11.2. See the *Tools and Frameworks Installation Guide* and perform the tasks in "Installing Oracle Commerce Guided Search Tools and Frameworks on Windows" or "Installing Oracle Commerce Guided Search Tools and Frameworks on UNIX".

Restoring Workbench configuration

After updating to Tools and Frameworks 11.2, you can restore your Workbench configuration from backups.

If you plan to use the 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 Tools Service (on Windows) or the `workbench` service (on UNIX) is configured and running.

After updating to Tools and Frameworks 11.2, 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 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 Tools Service.

Migrating applications from 11.1 to 11.2

To migrate an application from Tools and Frameworks 11.1 to 11.2 you must restore the application configuration from backups.

Deploying the 11.2 application

Use the Deployment Template to deploy an 11.2 version of your application.

To deploy an 11.2 version of your 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\11.2.0\deployment_template\bin>deploy
   --app C:\Endeca\ToolsAndFrameworks\11.2.0\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 11.1 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 11.2 application

To initialize your 11.2 application, follow these steps:

1. Log in using the username and password provided by your administrator. You will be prompted to change your password. You must change your password before you can proceed with these instructions.
2. Add the new password to the Oracle Credentials Store (OCS), following these steps:
   a) Navigate to `%ENDECA_TOOLS_ROOT%/credential_store/bin`
   b) Run the `manage_credentials[.bat|sh]` script with following arguments and provide the new password that you specified:

   ```
   manage_credentials[.bat|.sh] add --key ifcr --user admin
   ```

3. Open the command prompt.

4. Navigate to the `control` directory of the deployed application.

   For example: `C:\Endeca\Apps\Discover\control`.

5. Run the `initialize_services` script to provision your application within the application Controller and create 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.

Exporting application deployment settings

If you are using Experience Manager, and you want to use the workspace feature introduced in 11.2, export your application deployment settings to a temporary directory.

To export application deployment settings:

1. Open the command prompt.
2. Navigate to `Endeca\apps\<application_name>\control`.
3. You can use the `runcommand` script in the `\control` directory of your application to run the command that exports the deployment settings.
   For example, the following command exports deployment settings to a temporary directory named deploymentSettings:

```
runcommand IFCR exportContent configuration/tools/deploymentSettings
C:\Temp\exportDirectory\tools\deploymentSettings
```

Repeat this procedure for all of your applications.

Exporting application report settings

If you are using Experience Manager, export your application report settings to a temporary directory.

To export application deployment settings:

1. Open the command prompt.
2. Navigate to `Endeca\apps\<application_name>\control`.
3. You can use the `runcommand` script in the `\control` directory of your application to run the command that exports the report settings.
   For example, the following command exports report settings to a temporary directory named reportSettings:

```
runcommand IFCR exportContent configuration/tools/reportSettings C:\Temp\export¬
Directory\tools\reportSettings
```

Repeat this procedure for all of your applications.

Importing 11.1 Workbench users to Tools and Frameworks 11.2

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

To import 11.1 Workbench users to Tools and Frameworks 11.2:

1. Configure the `import_users` script:

   a) Navigate to the `ToolsAndFrameworks\11.2.0\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 11.2 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\11.2.0\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 11.1.0.
   - `--config` — Required. 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\<version>\admin\logs\import_validation_failed.log`. The main log file is output to `ToolsAndFrameworks\<version>\admin\logs\import_users.log`.

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 export file produced by the `export_site` script.

   For example:

   ```
   C:\Endeca\apps\Discover\control>import_site.bat
   C:\migration\Discover\ECR-backups\discover-2015-10-30.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.

Importing application deployment settings

If you are using the Experience Manager, and you want to use the workspace feature introduced in 11.2, import the 11.2 application deployment settings.

To import the 11.2 application deployment settings:

1. Open the command prompt.
2. Navigate to `Endeca\apps\<application_name>\control`.
3. Use the `runcommand` script to run the command that imports the deployment settings.
   For example, the following command imports deployment settings from a directory named deploymentSettings. This is the temporary directory where you saved the 11.2 deployment settings:

   ```
   runcommand IFCR importContent configuration/tools/deploymentSettings
   C:\Temp\exportDirectory\tools\deploymentSettings
   ```

Repeat this procedure for all of your applications.

Importing application report settings

If you are using the Experience Manager, import the 11.2 application report settings.

To import the 11.2 application report settings:

1. Open the command prompt.
2. Navigate to `Endeca\apps\<application_name>\control`.
3. Use the `runcommand` script to run the command that imports the report settings.

For example, the following command imports report settings from a directory named reportSettings. This is the temporary directory where you saved the 11.2 report settings:

```
runcommand IFCR importContent configuration/tools/reportSettings C:\Temp\export¬
Directory\tools\reportSettings
```

Repeat this procedure for all of your applications.

Updating assembler-context.xml

In order to use new features introduced in 11.2, you must update several bean definitions in the `assembler-context.xml` file.

To update `assembler-context.xml`:

1. Navigate to the Assembler context file for your application, `WEB-INF/assembler-context.xml`.
2. Open `assembler-context.xml`.
3. Add the `userState` property to three bean definitions.
   a) Find the `CartridgeHandler_ContentInclude` bean.
   b) Add the `userState` property.

```
<!--
    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    ~ BEAN: CartridgeHandler_ContentInclude
    ~ Used by the assembler service when keyword redirects are not enabled
-->
<bean id="CartridgeHandler_ContentInclude"
  class="com.endeca.infront.content.ContentIncludeHandler"
  scope="prototype">
    <property name="contentSource" ref="contentSource" />
    <property name="siteState" ref="siteState"/>
    <property name="userState" ref="${user.state.ref}"/>
</bean>
```

   c) Repeat step 3 for the `CartridgeHandler_RedirectAwareContentInclude` and `Navigation¬ CartridgeHandler` bean definitions.
4. Update the Store Factory bean.

   In 11.2, all store factory configurations (for both staging and production environments) must use FileStoreFactory.
   a) Delete the `ecrStoreFactory` bean.
   b) Find the `FileStoreFactory` bean.
   c) Optionally update the ID to `storeFactory`. This is recommended but not required.
   d) Ensure the class name is `com.endeca.infront.content.source.FileStoreFactory`.

```
<bean id="storeFactory" class="com.endeca.infront.content.source.FileStoreFac¬
tory"
            init-method="init" destroy-method="destroy">
    <property name="configurationPath" value="${repository.configura¬
tion.path}"/>
    <property name="isAuthoring" value="${preview.enabled}"/>
    <property name="appName" value="${workbench.app.name}" />
    <property name="host" value="${workbench.host}" />
    <property name="clientPort" value="${workbench.publishing.clientPort}"/>

    <property name="serverPort" value="${workbench.publishing.serverPort}"/>
</bean>
```

5. Update all references to the `ecrStoreFactory` bean to `storeFactory`, or the ID used in the `File¬`
   `StoreFactory` bean.

   a) Find the `adminService` bean.

   b) Update references to the `ecrStoreFactory` bean.

   ```
   <!--
       ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
       ~ Administration Service
       -->
   <bean id="adminService"
     class="com.endeca.infront.assembler.servlet.admin.AdministrationService">
         <property name="storeFactory" ref="storeFactory"/>
   </bean>
   ```

   c) Repeat step 5 for the `siteManager` and `contentSource` bean definitions.

6. Add the `appName` property to the Store Factory bean and the `mdexResource` bean:

   ```
   <property name="appName" value="${workbench.app.name}" />
   ```

Repeat this procedure for all of your applications.

Updating assembler.properties

In 11.2, all store factory configurations use FileStoreFactory. You must delete the `store.factory` property
from the `assembler.properties` file.

To delete the `store.factory` property:

1. Navigate to the `assembler.properties` file for your application, `WEB-INF/assembler.properties`.

2. Open `assembler.properties`.

3. Delete the `store.factory` property.

   ```
   store.factory=ecrStoreFactory
   ```

Repeat this procedure for all of your applications.

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.

Updating rules

If you are upgrading to 11.2 for the first time, after you have run a baseline update, you must update rules to add the searchTermExpansions property.

The searchTermExpansions property, introduced in 11.2, is used by the Assembler to determine which content-items and keyword redirects should trigger for a given search term. searchTermExpansions has the format: `"tokenNumber": {"token": ["representative stems"]}`. This property is for internal use only. Oracle recommends that this property not be edited manually.

To update rules:

From the `control` directory of your deployed application, run the `update_search_term_expansions` script.

Alternatively, you can invoke the public API directly as follows:

```
runcommand.bat IFCR updateSearchTermExpansions
```

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 11.2.0 host is different from the 11.1.0 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.

Required Changes

The following section lists required changes in Oracle Commerce Guided Search Tools and Frameworks 11.2. You must make the changes specified in this section if the changes apply to your application.

Updating preview settings

If you set device settings in the **Preview Settings** interface in Workbench in Tool and Frameworks 11.1, you need to manually update these settings in 11.2.

Oracle Commerce added height and width properties to preview settings for devices in 11.2. You must enter values for these settings for all devices after you migrate to 11.2.

All devices can be rotated while previewing a page, so you no longer need to enter a device profile for both landscape and portrait views of the same device. You can enter the height and width of the page orientation that is previewed more. If you had pairs of devices for portrait and landscape views in 11.1, you can remove one profile for each pair after you migrate. You must also update the names of devices.

Follow these steps:

1. In Workbench, navigate to the **Application Settings > Preview Settings** tool.
2. For each device or pair of devices follow these steps:
   a) In the **Manage Preview Devices** section, delete one of the preview setting profiles for each device pair.
   b) For the remaining profile for the device, enter values for the height and width of the page orientation that is previewed more often.
   c) Optionally, enter a new name for the device.
   d) Optionally, enter a zoom factor.

      The zoom factor can simulate displays on devices other than the current monitor. For example, the display on a retina display monitor can be simulated by setting the Zoom factor to 30 -- that is, 30%.

3. Click **Save**.

Example

The following example shows the device preview settings before and after they have been manually updated.

- The second entry for a pair of settings has been deleted.
- The device name has been updated by removing `(Landscape)`.
- Height and width have been entered.

Table 1: Before

Name	User Agent	Height	Width
Handheld (Landscape)	Mozilla/5.0 (iPhone; U; CPU like Mac OS X; en) AppleWebKit/420+ (KHTML, like Gecko) Version/3.0 Mobile/1A537a Safari/419.3		
Handheld (Portrait)	Mozilla/5.0 (iPhone; U; CPU like Mac OS X; en) AppleWebKit/420+ (KHTML, like Gecko) Version/3.0 Mobile/1A537a Safari/419.3		

Table 2: After

Name	User Agent	Height	Width
Handheld	Mozilla/5.0 (iPhone; U; CPU like Mac OS X; en) AppleWebKit/420+ (KHTML, like Gecko) Version/3.0 Mobile/1A537a Safari/419.3	**680**	**400**

Logging changes for Assembler applications

In Tools and Frameworks 11.2, Assembler logs let you correlate application Assembler requests with MDEX query entries in the Dgraph request log. This helps you to identify and troubleshoot problems.

Prior to 11.2, there was limited information logged in the Dgraph request log and in the JSON response page. In 11.2, additional logging data has been added to the JSON response page and Dgraph request log.

In a Spring implementation prior to 11.2, logging information was formerly configured in the `ContentItemAug¬mentAdapter` and the `LogServerAdapter` of the `assembler-context.xml` file as follows:

```
<bean class="com.endeca.infront.assembler.event.request.ContentItemAugmentAdapter">

    <constructor-arg ref="springUtility"/>
</bean>
<!-- Remove the following lines to disable logging to an Oracle Endeca Log Server
 -->
<bean class="com.endeca.infront.navigation.event.LogServerAdapter">
    <constructor-arg ref="springUtility"/>
    <constructor-arg value="${logserver.host}"/>
    <constructor-arg value="${logserver.port}"/>
    <constructor-arg value="${logserver.sslEnabled}"/>
</bean>
```

In 11.2, the `RequestEvent` generation has been moved to its own class. The `ContentItemAugmentAdapter` and `LogServerAdapter` services require `RequestEventInitializer` to log request-related information. The `RequestEventInitializer` should be configured before these services. The `MdexQueryInfoIni¬tializer` is also configured so that query information such as the request id, and the session id is added to the Dgraph request log.

```
<bean class="com.endeca.infront.assembler.event.request.RequestEventInitializer">

    <property name="sessionIdProvider" ref="springUtility"/>
    <property name="requestIdProvider" ref="springUtility"/>
</bean>
```

```
<bean class="com.endeca.infront.navigation.event.MdexQueryInfoInitializer">
</bean>
<bean class="com.endeca.infront.assembler.event.request.ContentItemAugmentAdapter">
</bean>
<!-- Remove the following lines to disable logging to an Oracle Endeca Log Server
 -->
<bean class="com.endeca.infront.navigation.event.LogServerAdapter">
    <property name="logServerHost" value="${logserver.host}"/>
    <property name="logServerPort" value="${logserver.port}"/>
    <property name="isSslEnabled" value="${logserver.sslEnabled}"/>
</bean>
```

Updating keyword redirect groups

If your deployment of Tools and Frameworks meets all of the following criteria you must update your keyword redirect groups after you migrate to 11.2.

- You have applied *Patch 19336952: Allow site specific keyword redirects* to your Tools and Frameworks 11.1 deployment
- You have created keyword redirect groups for sites in your application.

  The keyword redirect group JSON files are located in the import\pages folder at: *<app dir>*\config\import\pages*<site_id>*\<page>\redirects\

To update the keyword redirect groups you must complete the following steps:

- Move your redirect group folder from its location in the `import\pages` folder to the `import\redirects` folder.
- Update your search results page template with a `redirectGroup` property.
- All pages and rules based on a template that has been updated with the new `redirectGroup` property must also have the `redirectGroup` property. Update the `content xml` files of search results pages with a string value containing the redirects folder name. All other content based on the updated template can have a `redirectGroup` property with no value.

Follow these instructions:

1. Export the application for which you want to update keyword redirect groups.
   a) Navigate to the `<app dir>\control\` directory on Windows (`<app dir>/control/` directory on Unix).
   b) From the command line, export the application by entering the following command:
      `runcommand.<bat|sh> IFCR exportApplication <destination> true.`

2. In the destination directory, navigate to the page that contains the relevant redirects folder for a site: `<app dir>\config\import\pages\<site_id>\<page>\`

   For example:

   `Discover\config\import\pages\DiscoverCameras\browse\`

3. Cut the redirects folder.

   > 📝 **Note:** Cutting it, rather than copying it, ensures that you actually remove the folder from its old location.

4. Navigate to the redirects folder for the application (not a site): `<app dir>\config\import\redirects`

For example:

```
Discover\config\import\redirects
```

5. Paste the redirects folder that you just cut into this folder.

   This results in a redirects folder nested inside a redirects folder. For example:

```
Discover\config\import\redirects\redirects
```

6. Rename the nested redirects folder with a name that is appropriate for your site's keyword redirect group. For example:

```
Discover\config\import\redirects\DiscoverCameras
```

   Make a note of the folder name because you need this information when you edit the `content.xml` of the search results page.

7. Oracle recommends editing the `redirects\_.JSON` file to remove the line that contains the `enableThe¬saurus` property. This property is not currently used.

   For example, remove the bold text in the following example:

```
{
    "ecr:type": "redirect-group",
    "displayName": "Discover Cameras",
    "enableStemming": "TRUE",
    "enableThesaurus": "FALSE",
    "redirects": [
        {
            "matchmode": "MATCHEXACT",
            "url": "/browse/cameras/_/N-1z141ya",
            "searchTerms": "camera"
        },
        {
            "matchmode": "MATCHEXACT",
            "url": "/contact-us",
            "searchTerms": "locations"
        }
    ]
}
```

8. Navigate to the `template.xml` file for the page template on which your search results page is based: `\import\templates\<page>\template.xml`.

9. Use a text editor to add a `redirectGroup` property to the `template.xml` file.

```
<Property name="redirectGroup"><String/></Property>
```

10. Navigate to the `context.xml` file for the relevant search results page for the site.

    For example, the `context.xml` file for Discover Cameras whose search results can be rendered by a browse page would be stored at: `pages\DiscoverCameras\browse\`

11. Use a text editor to update the `context.xml` file with the site-based keyword redirect group name appropriate for your site. The group name must match the folder name where the redirects JSON is stored.

    For example, for Discover Cameras, the redirects group string has the following name:

```
</Property>
  <Property name="redirectGroup">
    <String>/redirects/DiscoverCameras</String>
  </Property>
```

12. Import the content with the new keyword group updates.

a) Navigate to the *<app dir>*`\control\` directory on Windows (*<app dir>*`/control/` directory on Unix).

b) From the command line, import the updated content by entering the following commands:

- `runcommand.`*<bat/sh>* `IFCR importContent pages <path to source>`
- `runcommand.`*<bat/sh>* `IFCR importContent redirects <path to source>`
- `runcommand.`*<bat/sh>* `IFCR importContent templates <path to source>`

For example:

```
runcommand.bat IFCR importContent pages c:\myexports\Discover\config\im¬
port\pages
```

```
runcommand.bat IFCR importContent redirects c:\myexports\Discover\config\im¬
port\redirects
```

and

```
runcommand.bat IFCR importContent templates c:\myexports\Discover\config\im¬
port\templates
```

If you do not update either the template or `content.xml` for the search results page, then business users editing content in the Experience Manager might see the following warning:

Note: The template for this rule or one of its children has been updated. Some content could be discarded when you save changes. If you do not want to continue, cancel and contact your technical team.

Discover reference application update

In Tools and Frameworks 11.2, the `BasicActionPathProvider` class has changed. A constructor has been added specifically to handle changes in the authoring environment. It is required that this constructor is used in authoring.

An additional parameter, `UserState` is now required. If you use or extend this class prior to 11.2, you will need to update your implementation to reflect these changes. This also means you will need to update your Spring configuration.

```
<bean id="actionPathProvider" scope="request" class="com.endeca.infront.refapp.nav¬
igation.BasicActionPathProvider">
   <constructor-arg index="0" ref="contentSource"/>
   <constructor-arg index="1" ref="httpServletRequest"/>
   <!-- navigationActionUriMap -->
   <constructor-arg index="2">
     <map>
        <entry key="^/pages/[^/]*/mobile/detail$" value="/mobile/browse" />
        <entry key="^/pages/[^/]*/services/recorddetails/.*$" value="/ser¬
vices/guidedsearch" />
        <entry key="^/pages/[^/]*/detail$" value="/browse" />
        <entry key="^/services/.*$" value="/services/guidedsearch" />
     </map>
   </constructor-arg>
   <!-- recordActionUriMap -->
   <constructor-arg index="3">
      <map>
        <entry key="^/pages/[^/]*/mobile/.*$" value="/mobile/detail" />
        <entry key="^/pages/[^/]*/services/.*$" value="/services/recorddetails"
 />
```

```
            <entry key="^/pages/[^/]*/.*$" value="/detail" />
            <entry key="^/services/.*$" value="/recorddetails" />
         </map>
      </constructor-arg>
      <constructor-arg index="4" ref="siteState"/>
      <constructor-arg index="5" ref="${user.state.ref}" />
</bean>
```

Related Links

Updating preview settings on page 25
> If you set device settings in the **Preview Settings** interface in Workbench in Tool and Frameworks 11.1, you need to manually update these settings in 11.2.

Logging changes for Assembler applications on page 26
> In Tools and Frameworks 11.2, Assembler logs let you correlate application Assembler requests with MDEX query entries in the Dgraph request log. This helps you to identify and troubleshoot problems.

Updating keyword redirect groups on page 27
> If your deployment of Tools and Frameworks meets all of the following criteria you must update your keyword redirect groups after you migrate to 11.2.

Chapter 4

Behavioral Changes

This section describes changes in the 11.2 release that do not require action on your part, but do have an effect on how your application behaves after you upgrade.

Oracle Commerce Workbench

The following changes have been made to Workbench.

Add Location Dialog

In previous releases of Workbench, the Autosuggest feature of the Add Location dialog returned no more than 20 refinements. This limitation has been removed.

Add Location does not return dimension names among search refinements

In 11.2, dimension names are no longer included among the refinements returned by Add Location searches. For example, the dimension name `product.category` is no longer included among the refinements returned by a search for 'category'.

In previous releases, dimension names were included among the refinements in an Add Location search.

Passwords

Passwords for an Admin user can be used only once. Admin users will be prompted to change their passwords immediately after they enter them for the first time. They cannot use Workbench until they have changed their passwords.

The admin password is no longer accepted in the Installer screen.

Similarly, the passwords for Workbench users created by an administrator can be used only once. Workbench users will be prompted to change them immediately after they log in with their original passwords for the first time.

Migrating Report Settings to a Newer Version of Workbench

You can move report settings to a newer version of Workbench either by exporting and importing the report settings alone, or by exporting and importing the report settings as part of an overall export and import of site configuration.

Migration of Report Settings Alone

You can migrate the report settings alone by running the `exportContent` and `importContent` scripts, using the `runCommand` utility at `%ENDECA_APP%\control\`, as follows:

1. Export your report settings. To do this, execute the following command:

```
runcommand.[sh/bat] IFCR exportContent %PATH_TO_REPORT_SETTINGS_NODE% %TARGET_DI¬
RECTORY% true
```

2. Upgrade your Workbench to a newer version. For information about how to do this, refer to the `Oracle Commerce Guided Search Tools and Frameworks Migration Guide`.

3. Import your report settings. To do this, execute the following command:

```
runcommand.[sh/bat] IFCR importContent %PATH_TO_REPORT_SETTINGS_NODE%
%SOURCE_DIRECTORY%
```

Migration of Report Settings as Part of a Full Site Migration

You can also migrate report settings to a new version of Workbench as part of a full-site migration. To perform a full-site migration, run the `export_site` script.

For information about the format in which report settings are exported and imported, refer to the *Oracle Commerce Guided Search Administrator's Guide*.

Related Links

Oracle Commerce Workbench on page 31
     The following changes have been made to Workbench.

Documentation changes on page 34
     The following documents have been changed or removed in this release.

SSL Protocol Changes

In release 11.2 of Tools and Frameworks, the cryptographic protocols TLSv1.1 and TLSv1.2 are enabled by default.

These protocols provide protection against serious security threats that have emerged recently. The protocols SSL 3.0 and TLS 1.0 do not provide similar protection and are disabled by default. Note that if you enable SSL 3.0 and TLS 1.0 -- for compatibility or any other reason -- you thereby make your application vulnerable to the serious threats against which TLSv1.1 and TLSv1.2 provide protection.

Be sure to verify that any non-SSL connector in `ENDECA_TOOLS_CONF/conf/server.xml` is commented out, and uncomment the following SSL connector:

```
<Connector port="8446" SSLEnabled="true"
  protocol="org.apache.coyote.http11.Http11Protocol" maxPostSize="0"
  maxThreads="150" scheme="https" secure="true" clientAuth="false"
  sslEnabledProtocols="TLSv1.1,TLSv1.2"
  keystoreFile="C:/Endeca/ToolsAndFrameworks/11.1.0/deployment_template~
    /ssl_certs_utility/bin/ssl/hostname.ks"
  keystorePass="eacpass"
  truststoreFile="C:/Endeca/ToolsAndFrameworks/11.1.0/deployment_template~
    /ssl_certs_utility/bin/ssl/TS-hostname.ks"
  truststorePass="eacpass" URIEncoding="UTF-8" />
```

Steps to enable the SSL 3.0 and TLS 1.0 protocols

> **Note:** If you enable SSL 3.0 and TLS 1.0 -- for compatibility or any other reason -- you thereby make your application vulnerable to the serious threats against which TLSv1.1 and TLSv1.2 provide protection.

To enable the SSL 3.0 protocol for Tools and Frameworks, follow these steps:

1. Open `server.xml` at `%ENDECA_TOOLS_ROOT%\server\workspace\conf`.

2. Change `sslEnabledProtocols` to `sslEnabledProtocols="SSLv3.0"` in the SSL connector.

```
<Connector port="8443" SSLEnabled="true"
   protocol="org.apache.coyote.http11.Http11Protocol"
   maxPostSize="0"
   maxThreads="150" scheme="https" secure="true"
   clientAuth="true" sslEnabledProtocols="SSLv3"
   keystoreFile="cert.ks" keystorePass="eacpass"
   truststoreFile="ca.ks" truststorePass="eacpass"
   URIEncoding="UTF-8"
```

3. Open `java.security` file in `%ENDECA_TOOLS_ROOT%/server/j2sdk/jre/lib/security`.

4. Uncomment the `jdk.tls.disabledAlgorithms` property and disable all other protocols except `SSLv3` `"jdk.tls.disabledAlgorithms=TLSv1, TLSv1.1, TLSv1.2"`.

5. Restart the Tools and Frameworks server.

To enable the TLS 1.0 protocol, follow these steps:

1. Open `server.xml` at `%ENDECA_TOOLS_ROOT%\server\workspace\conf`.

2. Change `sslEnabledProtocols` to `sslEnabledProtocols="TLSv1"` in the SSL connector.

```
<Connector port="8443" SSLEnabled="true"
   protocol="org.apache.coyote.http11.Http11Protocol"
   maxPostSize="0"
   maxThreads="150" scheme="https" secure="true"
   clientAuth="true" sslEnabledProtocols="TLSv1"
   keystoreFile="cert.ks" keystorePass="eacpass"
   truststoreFile="ca.ks" truststorePass="eacpass"
   URIEncoding="UTF-8"
```

3. Open `java.security` file in `%ENDECA_TOOLS_ROOT%/server/j2sdk/jre/lib/security`.

4. Uncomment the `jdk.tls.disabledAlgorithms` property and disable all other protocols except `TLSv1`:

```
jdk.tls.disabledAlgorithms=SSLv3, TLSv1.1, TLSv1.2
```

5. Restart the Tools and Frameworks server.

Related Links

The following changes have been made to Workbench.

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

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