ORACLE®

COMMERCE

11.3.2

Merchandising Administration Guide

Merchandising Administration Guide

Product version: 11.3.2
Release date: 12-11-19
Document identifier: MerchandisingAdminGuide1912090102

Table of Contents

1 Introduction to Merchandising

Oracle Commerce Merchandising provides sophisticated tools to facilitate the creation and management of catalog content, media content, promotions, and pricing for one or more online commerce sites. When Merchandising is purchased and licensed, it is installed with the Oracle Commerce Core Platform.

About this Manual

This document summarizes Merchandising's main concepts, and describes how to configure the application so that its features can be used effectively to meet the needs of the merchandising initiative. This manual is intended for administrators and developers.

2 Overview

Merchandising provides an environment that can be configured to meet the needs of the merchandiser and the current initiative. For example, administrators and programmers define project workflows, and they can control internal user access to assets and activities, based on user roles. Merchandisers use projects to manage assets. They use the asset interface to control product listing and pricing, and to create a variety of promotions.

To develop and deliver content and promotions to a commerce site, Merchandising employs some features of Core Commerce and ATG Content Administration. ATG Content Administration tools simplify and track the creation and development of merchandising assets. Versions are maintained in a versioned database. Conflicts that are caused by concurrent users are flagged, and a choice of resolutions is offered. The deployed versions of assets (the versions that are published to the live site) are contained in a separate, unversioned database.

You can use Merchandising to support more than one commerce site. For an overview of multisite concepts and possibilities, see the *Multisite Administration Guide*

This chapter introduces the basic concepts, tools, structures, and terminology of Merchandising.

Merchandising Tools

The Oracle Commerce Business Control Center provides the user interface for most Merchandising and Merchandising administration activities. The ATG Control Center (ACC) provides the interface for the remaining tasks.

Tools for Administrators and Programmers

Merchandisers have specialized requirements for the development of each commerce site. Before merchandisers begin to create assets, administrators and programmers must set up and configure the Merchandising environment to accommodate these requirements. Among the setups are:

- Internal user accounts and roles, which define internal user access to parts of the application. See Setting Up Roles and User Accounts (page 12).

- Workflows for Merchandising projects. See Working with Projects (page 32).

- The topology for deployment, including staging and production (live) sites. See *Setting Up Deployment* in the *Content Administration Programming Guide*

- View mapping and task configuration file settings to customize the Merchandising UI. See the chapter *Customizing the UI for Specific Tasks and Activities* in the *Business Control Center Administration and Development Guide*

- Site pages, to implement displays of content provided by Merchandising assets, such as targeters and scenarios. See the *Page Developer's Guide*.

Tools for Merchandisers

Business users (merchandisers) use Merchandising to establish and manage one or more commerce sites. Merchandisers build catalogs of saleable items, and create a variety of targeted and personalized promotions. Merchandising tools enable merchandisers to do the following tasks. Unless otherwise specified, these tasks are described in the *Merchandising Guide for Business Users*

- Create projects to track the creation and management of commerce assets.

- Create catalogs and populate them with categories, products, and SKUs.

- Manage the catalog items, images, and copy that will be deployed to a live web site.

- Create and manage personalization assets: user segments, content groups, and targeters. In a multisite environment, personalization assets can be site-aware. See the *Business Control Center User's Guide* and the *Personalization Guide for Business Users*

- Manage price lists: manage price inheritance; set standard, sale, and volume pricing; assign custom prices to individuals or groups.

- Create promotions and scenarios. See *Creating and Maintaining Promotions* in the *Guide to Setting Up a Store* and *Creating Scenarios* in the *Personalization Guide for Business Users*

Merchandising Architecture

Merchandising resides on the asset management server, along with ATG Content Administration.

By default, the ATG Content Administration environment includes a production deployment target; it can be configured to include a staging target.

The following diagram shows the Merchandising deployment architecture for a production deployment target.
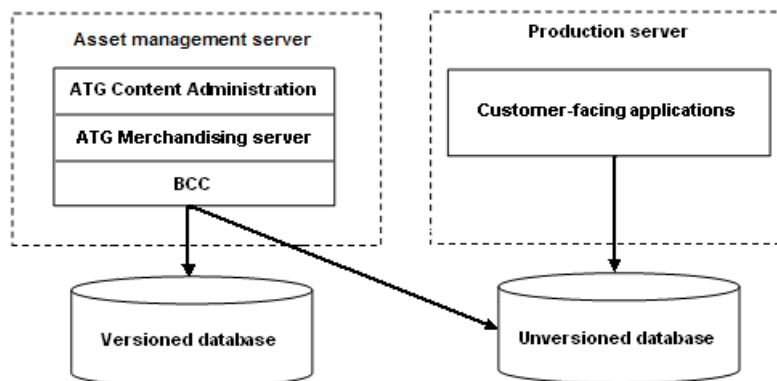
The following diagram shows the deployment architecture for staging and production targets.
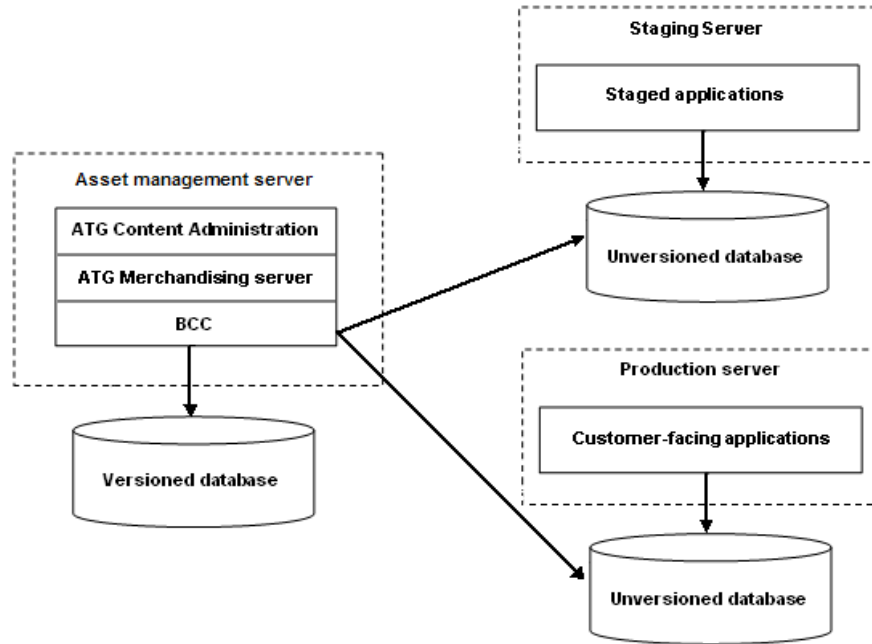
3   Configuring Merchandising

The installation of the Oracle Commerce Core Platform includes Merchandising. This chapter tells you what you need to do prior to the installation, and what must be done afterward to prepare the environment for commerce site development.

Installing Merchandising

Merchandising requires the Business Control Center and ATG Content Administration and can be installed with the Oracle Commerce Core Platform. For information, refer to the *Platform Installation and Configuration Guide*, which contains installation instructions and important configuration information. The remainder of this chapter assumes that the Core Platform has been installed.

After Installation: First Steps

After Merchandising has been installed successfully, you must complete the following tasks to make the application usable for commerce site development:

- Finish setting up the environment so that the Merchandising software can run. This includes assembling and deploying the application, publishing, and production servers. See Assembling a Merchandising Application (page 11).

- Install the ACC, if it is required.

- Start the Business Control Center.

- Configure a working Merchandising development environment.

You can use the Configuration and Installation Manager (CIM) to guide you through the following tasks:

- Specify the application server (that is already installed).

- Configure the database connections and create the database tables for ATG Content Administration.

    Important: On a production database, you must use the CIM option to drop the existing tables before you recreate them.

- Create Merchandising tables and import required content.

- Configure, assemble, and deploy instances of the production and asset manager servers. (On CIM, the asset manager server is called the publishing server.)

Using CIM ensures that you complete the required steps, in the correct order. CIM often provides the preferred selections as default values.

If you do not use CIM, you must complete the same tasks using other means. The basic reference for these tasks is the *Platform Installation and Configuration Guide*. The following documents may also be helpful:

- For instructions to configure the database connections and import required data to the selected data sources, see Configuring Database Tables (page 8) in this manual. For more database configuration information, see the *Content Administration Programming Guide* and the *Repository Guide*

- For instructions to assemble and deploy the EAR files for the production and publishing servers, see:

  - Assembling a Merchandising Application (page 11) in this manual

  - The *Content Administration Programming Guide*

  - The chapter *Developing and Assembling Nucleus-based Applications* in the *Platform Programming Guide*

Configuring Database Tables

If you use CIM to create the database tables and import the required data, you do not need to perform the procedures in this section.

To set up the ATG Content Administration database, you must create the required database tables and import the required data.

Important: To avoid errors later on, drop the existing tables before you create the new ones if you plan to use the existing schemas.

For details about the tables that are created in this section, see *Configuring and Populating a Production Database* in the *Core Commerce Programming Guide*.

Configuring Database Tables on an Evaluation Database

Oracle Commerce Platform supports the MySQL database for use in evaluation and development installations. You can use a MySQL database if you want to configure an Oracle Commerce Platform server quickly and you are not concerned about large-scale performance and reliability. Database configuration for production use requires more sophisticated preparation. Configuring databases for production use is covered in *Configuring Databases and Database Access* section in the *Platform Installation and Configuration Guide*.

To use a MySQL evaluation database:

1. Install MySQL. See *MySQL Databases* in the *Platform Installation and Configuration Guide*.

2. Start the database.

3. Follow the instructions below to create and populate versioned database tables as required by Merchandising and the Oracle Commerce Core Commerce.

4. If you intend to use other Oracle Commerce Platform applications, see the documentation for each application to learn the database table requirements.

5. Import data required by Merchandising. See Importing Required Data (page 10).

Configuring Database Tables on a Production-Ready Database

Before you begin to build the web application for your production web site, you must install and configure a production-ready database.

1. Set up an asset management server.

   Follow the steps in the *Setting Up an Asset Management Server* chapter of the *Content Administration Programming Guide*

2. Start the database.

   Follow instructions in the documentation from the database vendor.

3. To create the tables for Content Administration, run this script:

   ```
   <ATG11dir>/BIZUI/sql/install/database-vendor/bizui_all_ddl.sql
   ```

   For more information, see the *Content Administration Programming Guide*.

4. Follow the instructions below to create and populate versioned database tables as required by Merchandising and, optionally, Oracle Commerce Core Commerce.

5. Initialize and populate the Content Administration database by running the scripts that are described in the section *Initialize the Database* in the chapter *Setting Up an Asset Manager Server* in the *Content Administration Programming Guide*.

6. Import data required by Content Administration and Merchandising. See Importing Required Data (page 10).

7. If you use additional Oracle Core Platform applications, see the documentation for each application to learn the database-tables requirements for the application.

Creating Versioned Core Commerce Tables and Versioned Catalog Tables

The versioned versions of the Core Commerce tables have additional columns to store data for versioned assets. Merchandising requires the versioned tables.

To create versioned Core Commerce tables and versioned catalog tables on a production-ready or evaluation database, run the following script:

```
<ATG11dir>/DCS/Versioned/sql/install/database-vendor/dcs_versioned_ddl.sql
```

For information about starting and working with a MySQL evaluation database, see *MySQL Databases* in the *Platform Installation and Configuration Guide*.

Creating Versioned Business Commerce Tables

To use Merchandising with Business Commerce, you must create versioned database tables that support the combination of these products.

To create the versioned Business Commerce tables on a production-ready database, run the following script:

```
<ATG11dir>/B2BCommerce/Versioned/sql/install/database-vendor/
b2bcommerce_versioned_ddl.sql
```

For example, to install the tables on a MySQL evaluation database, use the following script:

```
<ATG11dir>/B2BCommerce/Versioned/sql/install/mysql/
b2bcommerce_versioned_ddl.sql
```

For information about starting and working with a MySQL evaluation database, *see MySQL Databases* in the *Platform Installation and Configuration Guide*.

Importing Required Data

If it has not yet been done, import the data that is required by ATG Content Administration. See the section *Initialize the Database* in the chapter *Setting Up an Asset Manager* in the *Content Administration Programming Guide*

The default `DataSource` connection pool, `JTDataSource`, uses the `FakeXADataSource` component. You must configure the desired connection pool properties, but note that this data source should be used only to run data import scripts. See the section *Configuring Data Sources for Data Import* in the *Platform Installation and Configuration Guide*

You must import Merchandising data before you can use Merchandising to perform any merchandising tasks, such as creating commerce assets. This data is part of the installation. It supports default user accounts, the UI, and other items, as indicated in the procedure.

To import the data:

1. Start the database.

2. Set the `DYNAMO_HOME` environment variable to `<ATG11dir>/home`, if you have not already done so.

3. Set the `JAVA_HOME` environment variable to your Java installation directory, if you have not already done so.

4. Run the following scripts:

    ```
    <ATG11dir>/Publishing/base/install/importPublishing.bat
    <ATG11dir>/BIZUI/install/importBizui.bat
    <ATG11dir>/AssetUI/install/importAssetUI.bat
    <ATG11dir>/DPS-UI/install/importDPSUI.bat
    <ATG11dir>/BCC/install/importBCC.bat
    ```

5. Run the following script to import the user role and view mapping content that is used by Merchandising:

    ```
    <ATG11dir>/DCS-UI/install/importMerchandising.bat|sh
    ```

1. Run the following script to import the information required to make segments available to Merchandising:

```
<ATG11dir>/DCS-UI/Versioned/install/importDCSUIVersioned.bat|sh
```

Assembling a Merchandising Application

Before you can access the Merchandising environment, the Merchandising application must be assembled as a J2EE application and deployed to an application server. The J2EE application must contain an instance of the modules that are used by Merchandising. When the application assembly is finished, you deploy and start the application.

The asset manager (the publishing server, on CIM) and production applications must also be assembled and deployed in this way.

Note: CIM automatically assembles the Merchandising application. If you use CIM to configure your installation, you do not need to perform the steps in this section.

The Oracle Commerce Core Platform provides an assembly tool for creating J2EE applications. See the chapter *Developing and Assembling Nucleus-Based Applications* in the *Platform Programming Guide*.

To build an application to run with Merchandising, include the following modules.

- `DCS-UI.Versioned`

- Either `B2CCommerce.Versioned` or `B2BCommerce.Versioned`

- The `PubPortlet` module

- The `SiteAdmin.Versioned` module, to support the Site Administration utility (required only if users on the server need the Site Administration utility)

- Other modules that contain your custom code

For information about the web content management modules, see *Web Content Management* in the *Content Administration Programming Guide*.

If your environment includes a staging target, use the `-layer staging` flag. For information about deployment targets, see *Target Site Basics* in the *Content Administration Programming Guide*.

If your environment includes a preview server, use the `-layer Preview` switch. For information about setting up preview, see the chapter *Setting Up Preview* in the *Business Control Center Administration and Development Guide*

The modules that you use for the asset management server, where Merchandising runs, are generally the versioned equivalents of those that are used in production. If the production server uses the `B2BCommerce` module, the asset management server uses the `B2BCommerce.Versioned` module, indicating Business Commerce with support for versioning (ATG Content Administration).

When you build the application, you specify the type of commerce product that your site implements (Consumer Commerce or Business Commerce), as well as other Commerce features. For example, the following command includes Merchandising (`DCS-UI.Versioned`), BUSINESS CONTROL CENTER components (`PubPortlet`), and Consumer Commerce (`B2CCommerce.Versioned`):

```
<ATG11dir>/home/bin/runAssembler MyApp.ear -m DCS-UI.Versioned
PubPortlet B2CCommerce.Versioned
```

For additional information about how to deploy your J2EE application, see the documentation from the application server vendor.

Starting the Business Control Center

The Business Control Center is the internal user interface to the Merchandising environment. You use the Business Control Center to create, preview, approve, deploy, and revise site content, and to launch Merchandising and other Oracle Commerce applications. See the *Introduction* chapter of the *Business Control Center User's Guide* for information about the following subjects:

- User profile and role requirements for login

- Instructions to start the products that you want to run with Merchandising

- How to start the Business Control Center

The *Merchandising Guide for Business Users* describes how merchandisers use the Business Control Center. The following documents also contain substantive information about setting up and using the Business Control Center:

For requirements for logging into the Business Control Center, see *Login Requirements* in the *Business Control Center Administration and Development Guide*

Business Control Center User's Guide

Business Control Center Administration and Development Guide

Content Administration Guide for Business Users

Using the ATG Control Center

The ATG Control Center (ACC) interface enables administrators to make configurations such as target groups of site visitors, and scenario promotions. You also use the ACC to create business reports. Refer to the *Platform Installation and Configuration Guide* and the *Personalization Guide for Business Users* for instructions about ACC installation, startup, and use.

Setting Up Roles and User Accounts

The following default user profiles are included with the Merchandising installation:

- admin – provides access to most areas of the Business Control Center. The EPub-Super-Admin global role is assigned to this user.

- Merchandising User – provides access to the Business Control Center for users who perform typical merchandising tasks with catalog and web content management assets. The Merchandising, EPub-User, and Business Control Center User roles are assigned to this user.

There are no default passwords for these accounts. Your organization sets the passwords through CIM or the ACC. For more information, see the *Platform Installation and Configuration Guide*.

Roles for Merchandisers

The following table describes the default roles available with the Merchandising installation. These roles represent typical Business Control Center users and simplify the process of creating internal users.

Role	Access Rights
Content Production	Catalog Media Read-Only Access Content Full Access Preview Users Full Access Price Lists Read-Only Access SEO Tags Read-Only Access Site Administration Read-Only Access Site Catalogs Read-Only Access Store Locations Read-Only Access
Internal User Administration	Internal Users Full Access
Merchandising	Site Catalogs Full Access Catalog Media Full Access Content Full Access Price Lists Full Access Preview Users Full Access Site Administration Read-Only Access SEO Tags Full Access Store Locations Full Access
Promotions User	Promotions Standard User Access
Promotions Administrator	Promotions Full Access
Personalization	Targeting & Segmentation Full Access Site Catalogs Read-Only Access Catalog Media Read-Only Access Content Read-Only Access Site Administration Read-Only Access External Users Full Access Preview Users Full Access
Site Administration	Site Administration Full Access Site Catalogs Full Access Content Full Access Price Lists Full Access Preview Users Full Access

For more information about creating and assigning users and roles in the Business Control Center, see *Creating Organizations and Roles* in the *Business Control Center Administration and Development Guide*

To learn about other default roles available in the Business Control Center, see *ATG Content Administration Users* in the *Content Administration Programming Guide*.

For information about using roles for access control, see *Using Roles for Access Control* in the *Platform Programming Guide* and *Configuring BCC Home Page Security* in the *Business Control Center Administration and Development Guide*.

Access to the ACC

Administrators use the ACC to manage versioned scenario and personalization assets, workflows, and occasionally other `PublishingRepository` items, such as projects. For information about managing user access and security for the ACC, see the *Content Administration Programming Guide*. For information on setting up ACC users and groups, see the *Personalization Programming Guide*

Access to Price Lists

Price lists have their own interface for secure access configuration. See .

Access Rights

The following table displays the access rights for the default roles available with the Merchandising installation. Each access right provides one of the following levels of access:

- Full Access: Provides Read, List, Write, Create, Delete, View Owner, Set Owner, View Access Rights, and Set Access Rights access.

- Read-Only Access: Provides List and Read access.

Access Right	Asset Access	Task Access
Content Read-Only Access	Read-only access to the following items in the `Content Management` Repository: article, mediaContent, folder, tag	Can access the Commerce Merchandising menu item and the Commerce Assets link (and its associated Browse link) on the Business Control Center home page.
Content Full Access	Full access to the following items in the `Content Management` Repository: article, mediaContent, folder, tag	Can access the Merchandising menu item and the Commerce Assets link (and its associated Browse link) on the Business Control Center home page.
Site Catalogs Read-Only Access	Read-only access to the following items in the `Product Catalog` Repository: Catalog, Category, Category Info, Product, Product Info, SKU, Configurable SKU, Foreign Catalog, Manufacturer, Generic Folder, Catalog Folder, and SKU Link.	Can access the Merchandising menu item and the Commerce Assets link (and its associated Browse link) on the Business Control Center home page.

Access Right	Asset Access	Task Access
Site Catalogs Full Access	Full access to the following items in the `Product Catalog` Repository: Catalog, Category, Category Info, Product, Product Info, SKU, Configurable SKU, Foreign Catalog, Manufacturer, Generic Folder, Catalog Folder, and SKU Link.	Can access the Merchandising menu item and the Commerce Assets link (and its associated Browse link) on the Business Control Center home page.
Catalog Media Read-Only Access	Read-only access to the following items in the `Product Catalog` Repository: Media Folder, Media, Media External, Media Internal Binary, and Media Internal Text.	Can access the Merchandising menu item and Commerce Assets link (and its associated Browse link) on the Business Control Center home page.
Catalog Media Full Access	Full access to the following items in the `Product Catalog` Repository: Media Folder, Media, Media External, Media Internal Binary, and Media Internal Text.	Can access the Merchandising menu item and the Commerce Assets link (and its associated Browse link) on the Business Control Center home page.
Store Content Read-Only Access	Read-only access to the following items in the `StoreRepository` Repository: Base resource text, Long resource text, Short resource text, Resources list, and Promotional Content.	Can access the Merchandising menu item and Commerce Assets link (and its associated Browse link) on the Business Control Center home page.
Store Content Full Access	Full access to the following items in the `StoreRepository` Repository: Base resource text, Long resource text, Short resource text, Resources list, and Promotional Content.	Can access the Merchandising menu item and the Commerce Assets link (and its associated Browse link) on the Business Control Center home page.
Price Lists Read-Only Access	Read-only access to the following items in the `Product Catalog` Repository: Price List, Price, Complex Price, Price Level, Generic Folder, and Price List Folder.	Can access the Price Lists menu item and the Commerce Assets link (and its associated Browse link) on the Business Control Center home page.
Price Lists Full Access	Full access to the following items in the `Product Catalog` Repository: Price List, Price, Complex Price, Price Level, Generic Folder, and Price List Folder.	Can access the Price Lists menu item and the Commerce Assets link (and its associated Browse link) on the Business Control Center home page.
Targeting and Segmentation Full Access	N/A	Can access the Targeting and Segmentation menu item and Targeting and Segmentation link (and its associated Browse link) on the Business Control Center home page.

Access Right	Asset Access	Task Access
`Store Locations Read-Only Access`	Read-only access to the following items in the `LocationRepository` Repository: Location and Store.	Can access the Merchandising menu item and the Commerce Assets link (and its associated Browse link) on the Business Control Center home page.
`Store Locations Full`	Full access to the following items in the `LocationRepository` Repository: Location and Store.	Can access the Merchandising menu item and the Commerce Assets link (and its associated Browse link) on the Business Control Center home page.
`SEO Tags Read-Only Access`	Read-only access to the following items in the `SEORepository` Repository: SEO Tag and SEO Translation.	Can access the SEO Tags menu item and the Commerce Assets link (and its associated Browse link) on the Business Control Center home page.
`SEO Tags Full Access`	Full access to the following items in the `SEORepository` Repository: SEO Tag and SEO Translation.	Can access the SEO Tags menu item and the Commerce Assets link (and its associated Browse link) on the Business Control Center home page.
`Site Administration Read-Only Access`	Read-only access to the following items in the `SiteRepository` Repository: Site, Site Group, and Site Template.	Can access the Site Administration menu item and the Site Assets and Site Categories links on the Business Control Center home page.
`Site Administration Full Access`	Full access to the following items in the `SiteRepository` Repository: Site, Site Group, and Site Template.	Can access the Site Administration menu item and the Site Assets and Site Categories links on the Business Control Center home page.
`External Users Full Access`	N/A	Can access the External Users menu item and the External Users link on the Business Control Center home page.
`Preview Users Full Access`	N/A	Can access the Preview Users menu item and the Preview Users link on the Business Control Center home page.
`Internal Users Full Access`	N/A	Can access the Access Control link on the Business Control Center home page.

Access Right	Asset Access	Task Access
`Promotions Standard User Access`	Full access to the following items in the `claimableRepository` Repository: Promotions, Coupons, Coupon Batches, Promotion Folders, and Coupon Folders. Read-Only access to the following items in the `claimableRepository` Repository: Stacking Rules	Can access the Merchandising menu item and the Commerce Assets link (and its associated Browse link) on the Business Control Center home page. Can access a promotion's Advanced tab in read-only mode.
`Promotions Read-Only Access` **Note:** This access right is not applied to any of the roles that are installed by default.	Read-only access to the following items in the `claimableRepository` Repository: Promotions, Coupons, Coupon Batches, Promotion Folders, Coupon Folders, and Stacking Rules.	Can access the Merchandising menu item and the Commerce Assets link (and its associated Browse link) on the Business Control Center home page. Can access a promotion's Advanced tab in read-only mode.
`Promotions Full Access`	Full access to the following items in the `claimableRepository` Repository: Promotions, Coupons, Coupon Batches, Promotion Folders, Coupon Folders, and Stacking Rules.	Can access the Merchandising menu item and the Commerce Assets link (and its associated Browse link) on the Business Control Center home page. Can access a promotion's Advanced tab and make changes to its filters.

Controlling Access to Assets in a Multisite Environment

In a multisite environment, you can control a Merchandising user's access to catalog assets, promotions, and content assets by site, displaying or providing access only to assets that belong to specified registered sites. This enables businesses that manage multiple sites to assign specific users or organizations to each site.

Most assets managed in Merchandising have a `SiteIDs` property, which specifies at list of sites an asset is associated with. You use the Access Control application in the Business Control Center to assign a user's site access settings by specifying sites to which the user has full, read-only, or no access. If an asset is associated with at least one site to which the user has read-only access, she will have read-only access to that asset in Merchandising. If an asset is associated with at least one site to which the user has full access, she will have full access to that asset in Merchandising.

Note that if you make changes to site access while there are users working on the sites, the access rights will not be applied until they have logged out and back into the application. For more information about assigning site access to internal users, see *Managing User Profiles* in the *Business Control Center Administration and Development Guide*. For more information about assets and site membership, see the *Multisite Administration Guide* and the *Core Commerce Programming Guide*.

Important: This is not a true security feature, but a usability enhancement that helps keep a merchandiser's view of the UI more streamlined by displaying only assets associated with sites that they regularly works with. This feature does not use secured repositories. Users who cannot see an asset in Merchandising because of site

access settings but who have access to other Oracle Commerce Platform interfaces, like Content Administration or the ACC, may be able to access and edit the asset there.

Content Assets

In a multisite environment, business users can specify that an article or media content asset applies only to specific sites. By default, an article or media-content asset applies to all sites, that is, its `SiteIDs` property is null.

Note: Folders and tags do not have `SiteIDs` properties and so their display in Merchandising is not affected by a user's site access settings.

Promotions

In a multisite environment, business users can specify that a promotion applies only to items purchased from specific sites. By default, a promotion applies to all sites. For more information about associating promotions with sites, see *Creating Promotions* in the *Merchandising Guide for Business Users*.

Configuring a Target Cluster

When you use Merchandising with ATG Content Administration, you must configure the Catalog Maintenance Service, and if you have custom catalogs, the Catalog Completion Service, for each agent server in a cluster that will receive deployed Merchandising content, perform the following tasks:

- Specify the `DCS.PublishingAgent` module during application assembly and startup to enable the commerce agent deployment operations. This includes the Catalog Maintenance Service, which should only run on a single, non-page-serving (auxiliary) instance in production. For servers that do not run the Catalog Maintenance Service, set the `active` property `false` in the `/atg/epub/CatalogMaintenanceHelper` component.

- Set the `enabled` property of the `/atg/commerce/catalog/custom/CatalogCompletionService` component to `true`, to ensure that asset properties are recalculated automatically after changes to the property values on which they depend.

- For Staging and Production targets: During `.ear` file creation on the ATG Content Administration side, include the `-layer staging` flag. When you use this flag, you can use the standard staging and production server repository configurations on the ATG Content Administration side, without modifying the workflow.

Note: The agent that runs the Catalog Maintenance Service must be set as an essential agent. If the agent is not made essential, it will not receive the message that triggers the Catalog Maintenance Server for the second deployment. This means that the data source will not receive the Catalog Maintenance Server update on the next switch and products may not be shown because of missing data.

In addition, you must configure one agent server in the target site cluster to run the Catalog Maintenance Service (CMS). If more than one agent server is configured to do this, only one will run the CMS after a successful deployment. See the chapter *Using the Catalog Maintenance System* in the *Core Commerce Programming Guide*

Additional configuration is required for one-off targets that accept trial deployments, which are outside the constraints of workflows. For more information about target servers, see the *Content Administration Programming Guide*.

Configuring Secured and Unsecured Repositories

An administrator secures a repository in order to control user access to assets, asset properties, and asset types. For example, you can make all products visible to a user, but prevent the same user from editing product properties.

Merchandising uses the following versioned Commerce repositories.

- Merchandising Product Catalog. By default, this repository is secured.

- Content Management. By default, this repository is secured.

- Price Lists. By default, this repository is secured.

- Claimable. By default, this repository is not secured.

The following table lists the names and paths of the secured and unsecured versions of the versioned Commerce repositories that are provided with Merchandising. The paths of the Secured Repository and XML Repository definitions are also shown.

	Unsecured Repository	Secured Repository	Secured Repository XML Repository Definition
Merchandising Product Catalog	`/atg/commerce/ catalog/ ProductCatalog`	`/atg/commerce/ catalog/ SecureProductCatalog.properties`	`/atg/commerce/ catalog/custom/ secureCustomCatalog.xml`
Price List	`atg/commerce/ pricing/priceLists/ PriceLists`	`/atg/commerce/ pricing/priceLists/ SecurePriceLists.properties`	`/atg/commerce/ pricing/priceLists/ securePriceLists.xml`
Content Management	`/atg/content/ ContentManagementRepository`	`/atg/content/ SecureContentManagementRepository`	`/atg/content/ secureContentManagementRepository.`
Claimable Repository	`/atg/commerce/ claimable/Claimable Repository`	None	None

Securing a Repository

You can secure standard repositories, as well as custom ones. If you modified your installation to use unsecured versions of the Merchandising Product Catalog and Price List repositories, you can re-implement the secured versions. The following instructions tell how to secure different types of repositories.

To secure all repositories *excluding* Merchandising Product Catalog and Price List:

1. Create and configure a Secured Repository Adapter component of class:
   `atg.adapter.secure.GenericSecuredMutableVersionContentRepository`

   The corresponding ATG Content Administration component is called `SecuredPublishingRepository`; this is described in the *Content Administration Programming Guide*.

   Note: Each repository requires a unique component instance.

2. If you have not done so already, make a copy of the `/atg/registry/ContentRepositories` file in your `localconfig` directory. In this file, replace the entry for the unsecured repository property with the path and name for the Secured Repository Adapter that you created in step 1.

3. Specify preliminary access controls to assets in a repository definition file.

   Create and configure a secured repository definition file, as described in the *Secured Repositories* chapter of the *Repository Guide*

4. Update all views that display assets from the repository you are securing. See Modifying the Views for Secured or Unsecured Repositories (page 21).

5. Include the secured repository in the `VersionManagerService` as a `versionedRepository` listing.

To secure the Merchandising Product Catalog repository:

1. Change the `componentPath` property of the `MerchandisingProductCatalog` component to the secured repository.

   Note: `MerchandisingProductCatalog` is an alias for `ProductCatalog`. It is found at `<ATG11dir>/DCS-UI/config/atg/commerce/catalog/MerchandisingProductCatalog.properties`.

   The `ContentRepositories.initialRepositories` property should always be set to `MerchandisingProductCatalog`, whose `componentPath` property is then mapped to the exact repository that is used.

2. Specify preliminary access controls to assets in a repository definition file.

   Modify the existing definition files listed in the table shown in Configuring Secured and Unsecured Repositories (page 19).

3. Include the secured repository in the `VersionManagerService` as a `versionedRepository` listing.

To secure the Price List repository:

1. Replace the entry for the unsecured repository in the `/atg/registry/ContentRepositories.initialRepositories` property with the path and name for the secured version of the repository.

2. Specify preliminary access controls to assets in a repository definition file.

   Modify the existing definition files listed in the table shown in Configuring Secured and Unsecured Repositories (page 19).

3. Update all views that display assets from the repository that you are securing. See Modifying the Views for Secured or Unsecured Repositories (page 21).

4. Include the secured repository in the `VersionManagerService` as a `versionedRepository` listing.

After you have set up and secured the repositories, you can change access rights for assets by following the instructions in the *Configuring Access Rights for Assets* section in the *Business Control Center Security* chapter of the *Business Control Center Administration and Development Guide*.

Unsecuring a Repository

The following instructions tell how to unsecure different types of repositories.

To unsecure all repositories *excluding* Merchandising Product Catalog

1. In `/atg/registry/ContentRepositories.initialRepositories`, replace the entry for the secured repository property with the path and name for the unsecured version of the repository.

2. Remove the Secured Repository in the `VersionManagerService` as a `versionedRepository` listing.

3. Update all the views that display assets from the repository. See Modifying the Views for Secured or Unsecured Repositories (page 21).

To unsecure the Merchandising Product Catalog Repository

1. Change the `componentPath` property of the `MerchandisingProductCatalog` component to the unsecured repository.

   Note: `MerchandisingProductCatalog` is an alias for `ProductCatalog`. It is found at `<ATG11dir>/DCS-UI/config/atg/commerce/catalog/MerchandisingProductCatalog.properties`. The `ContentRepositories.initialRepositories` property should always be set to `MerchandisingProductCatalog`, whose `componentPath` property is then mapped to the exact repository that's used.

2. Remove the Secured Repository in the `VersionManagerService` as a `versionedRepository` listing.

Modifying the Views for Secured or Unsecured Repositories

A view displays a subset of the data in the repository. The contents of each Repository View are determined by the object classes of the view's item descriptor, and the view's search roots. When a query is performed on the view, items are returned that reside in one of the specified search roots and satisfy the view's item descriptor. The search root may include the entire directory tree. For more information about repository views, see the *Repository Guide*

When you change the Product Catalog repository between secured and unsecured, the repository property and the views are updated, automatically; Merchandising uses the `MerchandisingProductCatalog` component, which points to the correct repository. When you change the Price List or Claimable repository between secured and unsecured, you must update all the views that display assets from the repository.

Note: Although it is possible for one repository that is used by your application to be secured and for another to be unsecured, *all the views for one repository must refer to the same type*: secured or unsecured.

The following table shows the views that must be updated.

Repository	Views
Merchandising Product Catalog	Catalog, Unlinked Catalog Items, Media, and Promotions. Views are updated automatically when the repository is changed between secured and unsecured.
Price List	Price Lists, and Price Lists and Folders. Views require manual update when the repository is changed between secured and unsecured.
Claimable	Coupons. View requires manual update when the repository is changed between secured and unsecured.

To update a view:

1. Locate the component that represents the view. Standard options include:

   • Promotions view:
     `/atg/commerce/web/assetmanager/PromotionsBrowseTab`
     `Configuration`

   • Media view:
     `/atg/commerce/web/browse/MediaTreeDefinition`

   • Price Lists view:
     `/atg/commerce/web/browse/PriceListsTreeDefinition`

   • Price Lists and Folders view:
     atg/commerce/web/browse/PriceListFoldersTreeDefinition

   • Coupons view:
     atg/commerce/web/assetmanager/CouponBrowseTabConfiguration

   Note: No changes are required for a Catalog or Unlinked Catalog Items view.

2. Change the value of the `repositories` property of this component to the repository you want to use.

Specifying Repositories that Allow Import from Spreadsheets

Normally, Merchandising allows users to import data from a spreadsheet to any valid repository, without additional configuration. However, importing data to an unsecured repository is not desirable. The administrator can control this situation by registering a list of secured and/or unsecured repositories that allow data import. A user can import data only to the repositories on this list. Specifying a target that is not on the list will not work.

To create a registry of the repositories that allow data import from a spreadsheet:

1. Locate the following component property:
   `/atg`/remote/assetmanager/transfer/service/
   ValidationEngine.importRepositoryPaths

2. Set the `importRepositoryPaths` property to the repositories that you want to allow as targets for asset imports. Use commas to separate multiple repository paths. For example:

   `/atg/commerce/catalog/SecureProductCatalog,/atg/commerce/claimable/`
   `SecureClaimableRepository`

   Important: If the value of the `importRepositoryPaths` property is null, users can import data to any valid repository.

For more information about importing versioned data, see the section *Importing Versioned Repository Data* in the *Repository Loader* chapter the *Repository Guide*.

Specifying Properties When Exporting SKU Prices

The `PricePreviewEngine` component contains the `skuPricePropertyNames` and the `productPricePropertyNames` properties that allow you to identify which properties are available when exporting prices.

For information on exporting prices, refer to the Exporting Prices chapter of the *Merchandising Guide for Business Users*.

For more information about secured repositories, see the chapter *Secured Repositories* in the *Repository Guide*.

Configuring Access Rights for Assets

In many cases, it may not be appropriate or desirable for all users in your organization to be able to view, create, edit, or delete all types of assets. Like all Business Control Center applications, Merchandising uses Access Control Lists (ACLs) to manage asset security and determine the operations that each user can perform on an asset. You set access rights for Merchandising assets by modifying the ACL in the definition file for the secured repository used to store the assets. ACLs are described in detail in the *Secured Repositories* chapter of the *Repository Guide*. In general terms, the mechanism works as follows: in the XML definition file of the secured repository that is used to manage the assets, there is an optional ACL setting for each item descriptor that represents a type of asset. This setting is the default security policy for all assets (repository items) of that type. The `<descriptor-acl>` tag contains a string specifying the user directory principals that have access to that item type and the level of access each principal has.

This section describes the access rights users must have to view and work with assets in Merchandising and suggests best practices for securing assets. For information about setting access rights, see *Business Control Center Security* in the *Business Control Center Administration and Development Guide*

A number of roles and access rights are available by default to simplify the task of creating Merchandising users. For more information, see Setting Up Roles and User Accounts (page 12).

Note: Item descriptor subtypes inherit the security settings defined for their parent item type by default, but you can define different security settings for each subtype. When determining an asset's access rights, Merchandising checks only the subtype's ACL, not the parent item type's ACL. To ensure Merchandising uses the access right you intended, specifically set access rights for each subtype in your secured repository. For example, a catalog might define a SKU item type with `furniture-SKU` and `clothing-SKU` subtypes. You should specify access rights for the `SKU` item descriptor and the `furniture-SKU` and `clothing-SKU` subtypes.

You can specify the following access rights for Merchandising assets and properties:

Access right	Description
Create	Controls whether a user can create new instances of an asset type.
	By default, the file used for content targeters, user segments, and content groups is `publishingFileSecurity.xml`, which is located in `<ATG11dir>\Publishing \base\config\atg\epub\file`. For information on how to modify a repository definition file, refer to the *Repository Guide*.
List	Controls whether this item appears as the result of a query, for example in the Browse or Find panel. In the Default Security Policy, List access is implied if the principal has Read access.
Read	Allows a user to view (but not edit or delete) the properties of this item.
Write	Allows a user to edit an item.

Access right	Description
Delete	Allows a user to remove this item from the repository. In Merchandising terms, this means that the user can add the asset to a project for the purposes of deleting it from the system. **Note:** Deleting an item also requires Destroy access to that Item.
Destroy	Allows a user to remove this item from the repository and destroy its contents. **Note:** Most secured repositories also require Delete access to the item's `RepositoryItemDescriptor`.

Viewing Assets in the Navigation Pane

In order to see assets in the Find and Browse panels, a Merchandising user must have both List and Read access to the asset and Read access to the asset type.

Viewing Assets in the Browse Panel

To see an asset in the Browse panel a user must have List and Read access to the asset and Read access to the asset type. If a user does not have Read access to an asset type, that type will not show up as a header in the Browse panel hierarchy. For example, if a Merchandising user does not have Read access to Products, only Catalogs and Categories appear in the hierarchy; the user cannot navigate to Products.

This behavior also applies to the Browse panel in the asset selector dialog that appears when you click the Select button in a Collection editor.

Viewing Assets in the Find Panel

To find a particular type of asset, a user must have Read access to an asset type. Otherwise, the asset type does not appear in the Look In dropdown on the Find panel.

To see an asset in the results of a query in the Find panel, a user must have both List and Read access to the asset and Read access to the asset type.

This behavior also applies to the Find panel in the asset selector dialog that appears when you click the Select button in a Collection editor.

Creating Assets

In order to create a new asset in Merchandising, a user must have both Write and Create access to an asset type to see and select it from the Create menu.

For more information about creating and editing assets in Merchandising, see *Creating and Managing* assets in the *Merchandising Guide for Business Users*.

Linking and Unlinking Assets

In order to link or unlink an asset, a user must have Write access to all of the following:

• The asset being linked or unlinked.

- The parent of the asset being linked or unlinked.

- The item descriptors for both the parent and child assets.

These rules also apply to assets linked by drag and drop.

For more information about linking and unlinking assets in Merchandising, see *Creating and Managing* assets in the *Merchandising Guide for Business Users*

Moving Assets

In order to move an asset, a user must have Write access to all of the following:

- The asset being moved.

- The asset's old (pre-move) and new (post-move) parents.

- The item descriptors for both the parent and child assets.

These rules also apply to assets moved by drag-and-drop.

For more information about moving assets in Merchandising, see *Creating and Managing* assets in the *Merchandising Guide for Business Users*.

Duplicating Assets

In order to duplicate an asset, a user must have the following access:

- Write and Create access to the item descriptor of the asset being duplicated

- Write access to the destination parent asset and its item descriptor

For more information about duplicating assets in Merchandising, see *Creating and Managing* assets in the *Merchandising Guide for Business Users*.

Deleting Assets

In order to delete an existing asset in Merchandising, a user must have Delete access to the asset's item descriptor and Destroy access to the asset itself. If the asset has parents, for example, a product in a category, the user must also have access to the Unlink From All menu item in Merchandising.

For more information about deleting assets in Merchandising, see *Creating and Managing* assets in the *Merchandising Guide for Business Users*.

Working with Assets in Collection Editors

In order to view Collection properties, a user must have Read access to an asset's item type. For example, a user who does not have Read access to media items will not see a product's auxiliary media collection property when viewing that product in Merchandising.

If a user has Read access to both an asset and its collection property, the collection editor in the Merchandising Details pane displays all the asset references it contains, regardless of the user's access rights to any assets referenced in the collection. For example, suppose a product has 50 SKUs in its `FixedChildSkus` collection.

A merchandiser does not need Read access to all 50 SKUs in order to see them listed in the Fixed Child SKUs collection editor in the product's Details pane.

Users who can access the collection editor for a property can reorder items in and remove items from the collection, regardless of access rights on those items.

For access rights required to add assets to collections, see Creating Assets (page 24).

For more information about working with collections, see *Working with Collection Editors* in the *Merchandising Guide for Business Users*.

Working with Parent Categories

Users without Write access to the Category item descriptor can see but not use these property editors.

• The Parent Categories editor that appears on a product's Categorization tab

• The Fixed Child Products editor that appears on a category's Products tab

For more information about working with parent categories, see *Working with Products* and *Working with Categories* in the *Merchandising Guide for Business Users*.

Working with Assets in the Light Table

In order to display the light table a user must have Read access to both Category and Product asset item types. Otherwise, the light table option is disabled in the menu that appears when the user right-clicks an asset in site view.

In order to reorder collections in the light table a user must have Read and Write access to both Category and Product asset item types. Otherwise, the user can only view the products in the light table and customize the light table's size and number of columns.

For more information about working with the light table, see *Organizing Collections with the Light Table* in the *Merchandising Guide for Business Users*.

Working with the Promotions Visibility Window

In order to display the promotions visibility window a user must have Read access to Promotion, Stacking Rule, Coupon, and Upsell item descriptors. Otherwise, the promotions button is disabled in the preview toolbar in layout view.

To see an asset in the promotions visibility window for example, a promotion in the Applied list or a target item on the Evaluation tab, a user must have both List and Read access to the asset itself and Read access to the asset type.

For more information about working with the promotions visibility window, see *Previewing and Testing Promotions* in the *Merchandising Guide for Business Users*.

Implementing a Custom Icon

To change the Catalog Media browse node icon, do the following:

- Provide the icon attribute value to the image for `id="media"` in `ContentBrowseHierarchy.xml`, and remove the existing `icon-resource` attribute.

  The following is sample code for providing the custom icon for browse mode:

```
<!-- The media node -->
 <browse-item id="media"
              label-resource="node.media.label"
              is-root="true"
              icon="images/stat_error_16.png">
   <list-definition id="mediaRootFolders"
                    retriever="query"
                    child-type="/atg/commerce/catalog/
MerchandisingProductCatalog:folder">
     <retriever-parameter name="query"
                          value="parentFolder IS NULL" />
     <retriever-parameter name="refresh-property"
                          value="parentFolder" />
   </list-definition>
 </browse-item> <!-- end media -->
```

To implement custom icons for the Catalog Asset type, do the following:

1. Find the icon attributes from `customCatalog.xml` for `item-descriptor` with `name="catalog"`.

   The attributes are `flexIconResource` and `largeFlexIconResource`. Their corresponding values are `catalogIconSwf` and `catalogLargeIconSwf`. Find these keys in the `resourceBundle` file present for the same item descriptor (locale specific, for example, `CustomCatalogTemplateResources_en.properties`).

2. Use the following path for the Tomcat server: `ATG-DIR\DCS\buildtree\atg\commerce`

3. Change this file in the `exe` installation and restart the server.

   The following is sample code for overriding the catalog icon:

```
catalogIconSwf=images/gray_x_icon.png
catalogLargeIconSwf=images/gray_x_icon.png
```

The following is sample code when using Font Awesome icons instead of an image:

```
catalogIconSwf=fa fa-calculator
catalogLargeIconSwf=fa fa-calculator
```

The following are customization examples:

`MerchConfig.003`, for example, to specify a localized display string for the previous `sku-color` example the following steps are necessary:

1. For the module containing the repository extension, ensure that the manifest file `META-INF/MANIFEST.MF` includes an `"ATG-Class-Path"` entry that appends the module's own classes and resources to the server class path.

2. Create a file named `Resources.properties`, with the following content:
   `categoryPresentation=Presentation sku.color=Color`

3. Place the file into a directory named `my/module`, in the module's class path.

4. Modify the SKU item descriptor as follows:

```
<item-descriptor name="sku"> <table name="my_sku_extensions" type="auxiliary" id-
column-name="sku_id"> <property name="color" data-type="string" column-name="color"
 category-resource="categoryPresentation" display-name-resource="sku.color"> <attribute
 name="resourceBundle" value="my.module.Resources"/> </property> </table> </item-
descriptor>
```

Defining Custom Asset Icons

When an asset is displayed in the UI, its name is often accompanied by an icon that represents its type. You can specify icons for a repository item type in the GSA template for the repository that contains the item descriptor.

An item descriptor definition can include the following attributes for specifying icons to be displayed in the Flex-based Asset Manager:

*   `flexIconResource`: The icon that appears in lists inside navigation panels.

*   `largeFlexIconResource`: The icon that appears in the header of the asset editor.

Both of these attributes specify keys into a resource bundle that is specified using the `"resourceBundle"` attribute. Each resource string represents the URL of an icon image file.

`MerchConfig.019`, for example, to define a custom icon for the `"facility"` item type from the previous examples:

1. Add attributes such as the following to the `"facility"` item descriptor in the `ProductCatalog` GSA template:

```
<attribute name="resourceBundle" value="my.module.Resources"/>
 <attribute name="flexIconResource" value="facility.icon"/> <attribute
 name="largeFlexIconResource" value="facility.largeIcon"/>
```

1. Add lines such as the following to `my/module/Resources.properties`, in your module's class path:

```
facility.icon=/MerchConfig/images/facility.png facility.largeIcon=/MerchConfig/images/
facility.png
```

1. Add the `facility.png` file to the images directory of the `/MerchConfig` web app.

Specifying an Icon for an Organizational Node

For an organizational node in the Browse panel, use the `"icon"` attribute for the associated `browse-item` element in the `ContentBrowseHierarchy.xml` file. For example:

```
<browse-item id="facilities" label="Repair Facilities" is-root="true" icon="/
MerchConfig/images/facilities.png">…
```

Configuring Partial URLs

You can define partial URLs for any property editor that accepts URLs, such as the URL property editor for a media content asset's URL property. Strings that you define do not modify the stored value of the URL in the repository; the string is prepended only when the URL is loaded.

You define a partial URL by creating a root string for the `url` property of a specified asset type. Merchandising prepends the root string to any URL accessed by that property before it is loaded. If a root string is defined for a specified property of an asset, it is always prepended. Merchandising does not validate the root string or the formatted URL.

The `urlFormatStrings` property of `/atg/remote/assetmanager/common/service/PartialURLRegistry` maps a specific repository/item type/property name combination to a root string.

The key portion of each key/value pair in the map must be an encoded string in the following format:

`<repositoryName>:<itemType>.<propertyName>`

The value portion of each key/value pair in the map must be the root string to prepend to the URL.

To create root strings for URLs, add the appropriate key/value pairs to the `urlFormatStrings` property in the `/atg/remote/assetmanager/common/service/PartialURLRegistry.properties` file. The following example defines the root strings for both media-content and media-external URLs as `/crsdocroot`:

```
# Define root strings for mediaContent and media-external items.
urlFormatStrings=ProductCatalog:media-external.url=/crsdocroot,\
                 ContentManagementRepository:mediaContent.url=/crsdocroot
```

Configuring Partial URLs for Subtypes

A repository item type can have subtypes that include additional properties that are not part of the base item type. By default, if any `itemType` has subtypes that are not already explicitly defined in the `urlFormatStrings` map, a duplicate entry for each subtype is automatically added to the map. This allows subtypes to inherit a root string from their parent, if a root string does not exist for that type.

To prevent a subtype from inheriting from its parent, you must define an explicit mapping for the subtype and set its value to an empty root string. The following example creates an explicit mapping so `media-external` items do not inherit root strings from the parent `media`:

```
# One of the subtypes of media is media-external. Create an explicit mapping
# so media-external items do not inherit media's root string.

urlFormatStrings=ProductCatalog:media.url=/crsdocroot,\
                 ProductCatalog:media-external.url=,\
                 ContentManagementRepository:mediaContent.url=/crsdocroot
```

Live Scrolling

When you scroll through a list with live scrolling in effect, the display moves through the list as you scroll. The feature depends on frequent updates to the list view. To scroll through large lists more rapidly, you can turn off live scrolling. When live scrolling is off, the display is not updated until you release the mouse button.

By default, live scrolling is turned on for the following displays:

• The data grids in the collection editor and project dialogs

• The child list in the Browse panel

• The results list in the Find panel

To turn off live scrolling in all three displays, set the `liveScrolling` property to `false` in this file:

```
/atg/remote/controlcenter/service/ControlCenterService.properties
```

Configuring Default Asset Searches

Searches for assets can be performed by providing a combination of asset ID and name, or just by asset ID. By default, searches are configured to use both an asset's ID and name. In environments that contain a large number of assets, search performance can be improved by searching only for asset IDs. To set the default search type, modify the `FindFlexExpressionModel` component. This component uses two properties that determine how an asset search works:

• `defaultSearchUsesIdOnly`: When set to `true`, searches will only be performed using asset IDs. By default, this property is set to `false`.

• `exactSearchWithIdsOnlySearch`: This property, which is used only when `defaultSearchesUsesIdOnly` is set to `true`, identifies if asset IDs are returned only when there is an exact match. By default this property is set to `false`, which indicates that the results will provide both asset IDs with an exact match and those that contain some portion of the ID. Setting the property to `true` will provide only those asset IDs that exactly match the ID.

The following example sets ID searches as the default search type, and only returns IDs that match the search criteria exactly:

```
defaultSearchUsesIdOnly=true
exactSearchWithIdsOnlySearch=false
```

Working with Tender-Based Promotions

Tender-based promotions allow a merchandiser to offer promotions based on a specific tender or payment type. For example, promotions such as "Pay for your order with cash and get 10% off" or "Use your store credit card and get 30% off your order" identify a specific tender or payment type.

You can configure the payment types that are presented to the merchandising user when they are creating a promotion. You can also configure if a payment can be partial when configuring promotions based on payment type.

For detailed information on configuring tender-based promotions, refer to the *Core Commerce Programming Guide*.

Displaying Tender-Based Promotions Payment Types

Payment types can be conditions for promotion. For example, "Use your American Express card and get 10% off." When a merchandising user creates a promotion in the BCC, he or she can select as many payment types as necessary from the Tender Type menu.

Payment types that can be displayed when creating a promotion, such as `creditCard`, can also specify a variation, such as `creditCard:Visa`.

You can configure the payment types and the variations that are listed to the merchandising user using the `DCS/config/atg/commerce/payment/PaymentTypeStringEnumProvider`. The `PaymentTypesStringEnumProvider` uses the `getResources` and `getCodes` methods to return a list of strings of all supported payment types. The result is a generic type for each payment group, as well as a payment type for each variation of the group. By default, the system displays only generic types for selection when there are no variations available, but this can be configured differently.

Use the `PaymenTypePresentationSetting` property to configure the payment types that are displayed to the user:

- `genericAndVariations`: Indicates that no payment types will be removed. Both 'Any Credit Card' and 'Visa' will be displayed.

- `genericOnlyIfNoVariations`: Default. The generic types will be included if no variation exists. For example, 'Any Credit Card' will be displayed only if there are no variations, such as 'Visa.'

- `genericOnly`: This removes everything that is a variation. For example, the system will display 'Any Credit Card', but not 'Visa.'

Important: If you add new payment types to your system, tender-based promotions must be updated by the merchandising user to include these new payment types.

For detailed information on tender-based promotions, refer to the *Core Commerce Programming Guide*.

Displaying Uncategorized Groups

Properties that are not contained explicitly within a group are displayed in the default group. If a default group contains one or more required properties, it is displayed at the top of the tab. For example, if an uncategorized group within catalog items contains required properties, it will be displayed on the top of the Advanced Tab. If an uncategorized group within articles contains required properties, it will be displayed on the top of the General Tab.

If a default group does not contain required properties, it is displayed at the bottom of the tab. For example, uncategorized groups within catalog items that do not contain required properties are displayed on the bottom

or the Advanced Tab, and uncategorized groups within articles that do not contain required properties are displayed on the bottom of the General Tab.

You can modify this so that all default groups are displayed at the bottom of the tab. Use the `/atg/remove/assetmanager/editor/service/TabPropertyFilter.properties` file to set the `requiredUncategorizedGroupFirst` property to `false`. Uncategorized groups that contain either required and non-required properties will be displayed on the bottom of its respective tab.

Working with Projects

Merchandising uses ATG Content Administration to manage and deploy versioned commerce assets. In order to use the ATG Content Administration mechanisms, Merchandising requires users to create and manage assets within a project.

The project enables a merchandiser to control the following aspects of asset management:

- Creating, editing, and maintaining versioned commerce assets

- Tracking the project's assets through the review and approval processes

- Deploying the project's assets to a live web site

Merchandising includes the following project type selections on the Business Control Center Operations list:

- Manage Commerce Assets

    A merchandiser uses this project type to create, add, modify, organize, and delete commerce assets.

When you are logged in to the Business Control Center as a merchandiser, your Merchandising project can include and manage assets in the following categories. These appear as selections in the drop-down menu above the Browse pane:

- Merchandising

- SEO Tags

- Price Lists

- Site Administration (if your site is configured for the Site Administration utility)

- Targeting and Segmentation

See the chapter *Creating and Managing Projects* in the *Content Administration Guide for Business Users*, for basic information about projects, workflows, and versioned assets in ATG Content Administration. See *Creating and Organizing Assets* in the *Merchandising Guide for Business Users* for information about creating merchandising assets.

Workflows

A Merchandising project associates a workflow with a set of assets. The workflow specifies the number and sequence of tasks that determine the project's life cycle.

Tasks such as the following are part of a typical workflow:

- Create an asset, revise asset content, or delete an asset

- Review new or changed content

- Approve the project for deployment

- Accept the deployment

The workflow also specifies whether to deploy only to a production target, or to a staging target as well. See *Processes and Projects* in the *Content Administration Programming Guide*. The chapter *Setting Up Deployment* in the same manual contains instructions to define the deployment topology.

Emergency Workflows

When a user selects an Emergency Project from the Merchandising menu, they initiate an emergency workflow. Emergency workflows are significantly different than standard projects as they allow users to make changes to a site that become available within minutes. In order to use an emergency workflow, the user must have the correct security settings.

Note: Emergency Projects are designed to be emergency deployments and should not be used as an alternative to standard deployment processes.

Emergency workflows are available for both Production-Only and Production/Staging environments. The Emergency Project, however, contains only two workflow tasks:

- Author

- Deploy to Production

Workflow Security

The administrator can set access rights on a workflow and on its individual task elements. For information about this configuration, see the *Managing User Access and Security* and *Project Workflows* chapters in the *Content Administration Programming Guide*.

Customizing Workflows

By default, Merchandising projects use the common workflow that is found in:

```
<ATG11dir>/Publishing/base/config/atg/registry/data/epubworkflows/Common/
commonWorkflow.wdl
```

The business requirements at your site may require a custom workflow, with additional steps such as an email notification or an extra authorization step.

If you need to create a custom workflow, consider these cautions:

- It is preferable to copy an existing workflow and modify the copy than to create a completely new workflow.

- Adhere as closely as possible to the original workflow, especially with respect to the elements that handle deployment and check-in.

- Do not change the check-in model on a workflow. Asset check-in prior to project deployment ("early check-in") is no longer supported. Assets are checked in immediately after a project is deployed and the deployment is verified/accepted ("late check-in").

Important: All the workflows in use must use the same deployment model:

- Production-only

- Staging and production

For information about the workflows that are included with ATG Content Administration, see the *Project Workflows* chapter of the *Content Administration Programming Guide*. In the same manual, the section *Workflow Action Elements* contains a list of workflow action elements.

The ACC provides the user interface for viewing, editing, and creating workflows. The application developer must create the custom workflow elements for your organization. For more information, see the following:

- *Using Workflows* in the *Personalization Guide for Business Users*

- *Creating and Configuring Workflows* in the *Personalization Programming Guide*

For information about modifying the user interface to accommodate workflow display, see *Customizing the UI for Specific Tasks and Activities* in the *Business Control Center Administration and Development Guide*.

Customizing Asset Version Conflict Resolution for Collection Properties

An asset version conflict can result when the same asset is edited in more than one project.

When you add or change an asset, a new working version of the asset is created. When an asset is changed in two projects, the result is two working versions of the asset—one in each project. The version that is deployed and checked in first becomes the next head version. Differences between the new head version and the version in the undeployed project must be resolved before the second project can complete its current task.

Merchandising notifies you about an asset version conflict when you try to act on a project task. When the conflict occurs between assets that are collection properties (lists, maps, sets, or arrays), Merchandising suggests a resolution that combines the changes.

You can view the proposed resolution by double-clicking a flagged asset in the project's changed asset list.

The areas of conflict are displayed.

Clicking the Review Changes button produces a table that shows the differences between the asset versions.

On a page that displays conflicts for a collection, Merchandising displays only the first 2300 items in a collection, by default. Therefore, if either version of the collection has more than 2300 items, Merchandising does not suggest a combined value because it cannot be fully displayed.

To accommodate large collections, some options of the suggested values feature are configurable. See:

An additional setting accommodates the handling of removed items in a suggested combination. See Setting the Precedence of Removed Items in Suggested Collection Values (page 36).

Setting the Maximum Number of Collection Items to Combine

You can override the 2300-item limit for which Merchandising will suggest a combination for resolving a collection properties version conflict. You can change the limit or completely remove the limit. In the latter case, the collection versions will be combined, regardless of the number of items in the collection. Note that for the combination limit change to be visible to the user, you must also extend the display limit past 2300 items. See Setting the Display Maximum for Collection Properties during Conflict Resolution (page 35).

To change or remove the item limit for combining collection items, you override the `atg/diff/merge/MergeHandlerFactory` component in your local configuration directory; for example, `<ATG11dir>/home/localconfig`.

Locate the `maxSizeLimit` property and set the value as follows:

• To *change* the limit for which Merchandising will suggest a combination, set the value to the maximum number of items to allow. When either collection exceeds this number, the collection values are not combined and a suggestion is not displayed.

• To *remove* the limit for which Merchandising will suggest a combination, set the value to **-1**. All items in the collection are combined to create a suggested value, regardless of whether the full collection is displayed.

For instructions to set component properties, see the section *Managing Properties Files* in the *Platform Programming Guide*.

Setting the Display Maximum for Collection Properties during Conflict Resolution

When a collection properties version conflict exists, Merchandising can produce a table that includes a suggested merged version, as shown in Customizing Asset Version Conflict Resolution for Collection Properties (page 34). By default, Merchandising displays only the first 2300 items in a collection. You can set the display limit to another value by changing the `maxNumDisplayedValues` attribute on the `propertyView` for collections that have conflicts.

Note: In some browsers, increasing the limit above 2300 may cause issues with the collections display.

To set the maximum number of collection items that can be displayed during conflict resolution:

1. Create a new `propertyViewAttributeDefinition` to hold the new value. The `propertyView` item descriptor is defined in `ViewMapping.xml`, as shown in this code excerpt :

```
<!-- attributes property -->
<table name="vmap_pv2pvad_rel" type="multi" id-column-name="view_id"
multi-column-name="name">
<property name="attributes"
component-item-type="propertyViewAttributeDefinition"
column-name="attr_id" data-type="map" cascade="update,delete"/>
</table>
</item-descriptor>
```

You will need to provide a meaningful name that is easy to locate, a unique ID, and a description; then set the attribute value to the new display limit.

2. Locate the following `propertyView` items and add the new `propertyViewAttributeDefinition` to their attributes. For each `propertyView`, add a new entry in the attributes list named `maxNumDisplayedValues` and map it to the ID of the `propertyViewAttributeDefinition` that you created in step 1.

Collection type	propertyView
Arrays	Name: `AssetManager array conflict viewer` ID: `aMgrPvCArray`
Lists	Name: `AssetManager list conflict viewer` ID: `aMgrPvCList`
Sets	Name: `AssetManager set conflict viewer` ID: `aMgrPvCSet`

3. Log in to Merchandising and verify that the Review Changes page displays the expected number of items when you resolve conflicts between very large collections.

Tip: If you use the ACC to edit this attribute, you may have to restart the Merchandising server to cause the changes to take effect.

For more information about editing view mappings, see the *Customizing Asset Display* chapter of the *Content Administration Programming Guide*.

Setting the Precedence of Removed Items in Suggested Collection Values

When one project edits an item and another project removes it, the default behavior is that Merchandising does not include the item in the suggested combination. You can change the default behavior so that the changed item is kept in the suggested combination. For example, if the value of a map item is changed in one project and the map item is deleted in another project, the changed map item is kept in the combined changes.

To change the precedence of removed items in suggested collection values:

1. Find the `removesWin` property in the `atg/diff/merge/MergeHandlerFactory` component in the local configuration directory; for example: `<ATG11dir>/home/localconfig`

2. Do one of the following:

   • To include deleted items (changes to an item override removal of an item), set the `removesWin` property value to `false`.

   • To restore the default precedence (removal of an item overrides changes to an item), set the `removesWin` property value to `true`.

For instructions to set component properties, see the section *Managing Properties Files* in the *Platform Programming Guide*.

Configuring Deployment Resources

Deployment is the process of copying assets from the Merchandising development environment to a staging or production environment.

Each repository that contains deployable assets exists in three forms:

- A source repository. This repository runs in the development environment. It supports versioned copies of assets in progress. It points to the ATG Content Administration database.

- A destination repository. This repository runs in the development environment. It is an unversioned repository that supports the assets you want to deploy. A destination repository is required for every target repository that receives deployed data, and it points to the same staging or production database as the target repository.

- A target (staging or production) repository. This repository runs in the staging or production environment, to support deployed assets. If both environments exist, there is a target repository in each one. The repository is unversioned. It points to the database in the staging or production environment to which you have deployed assets.

Source and destination repositories are already set up for assets that use the standard, unmodified Merchandising repositories. Creating the target repositories is part of the process of defining the target server, which is step 2 of the following procedure.

To configure deployment resources:

1. Read the *Deployment Concepts* chapter of the *Content Administration Programming Guide* to become familiar with the deployment process.

2. Set up the target server and database, as described in the *Setting Up Deployment* chapter of the *Content Administration Programming Guide*.

   Note: If you made changes to the standard repositories, such as adding properties to an existing item descriptor, you must make the same changes to the corresponding target repositories. The standard repositories and their locations are the same on the target server as they are on the asset management server.

3. If you created custom repositories, configure them for deployment, as described in the *Setting Up Deployment* chapter of the *Content Administration Programming Guide*.

4. Add any versioned repositories that are not already there, including custom repositories. Follow the instructions in the *Setting Up Deployment* chapter of the *Content Administration Programming Guide*.

5. If you are including a completely new file, reassemble, redeploy, and restart the application. If you used `AssetResolver` to add a custom repository, you do not need to build a new `.ear` file; you need only to restart the application.

Stopping Merchandising

Because Merchandising is installed with the Oracle Commerce Platform, you cannot uninstall or remove it separately. To stop Merchandising, stop the application instance on the application server. See *Running Nucleus-Based Applications* in the *Platform Installation and Configuration Guide* for details about stopping individual applications.

4 Configuring Visual Merchandising

This chapter describes how to enable and configure the Visual Merchandising, which lets Merchandising users select assets directly from a preview session to edit and organize. This chapter describes how to configure Visual Merchandising in your environment. For information about using Visual Merchandising, see the *Merchandising Guide for Business Users*.

This chapter contains the following topics:

Configuring Preview

Visual Merchandising requires a deployed preview application. This section describes how to optimize the configuration of your preview application for Visual Merchandising. For details about creating a preview application, see *Setting Up Preview* in the *Business Control Center Administration and Development Guide*.

Creating Default Preview URLs

Visual Merchandising requires a default preview URL that includes no `$[template.url]` properties. Specify a default preview URL with the properties of the `/atg/dynamo/service/preview/PreviewURLManager` component.

The `defaultNoAssetPreviewURL` property specifies the preview URL that is used when a merchandiser clicks the Layout button in Merchandising when no asset is currently loaded in the Details pane. For example, if a merchandiser starts a Merchandising project and immediately clicks the Layout button, the `defaultNoAssetPreviewURL` is displayed. This URL should not contain any asset-specific parameters. It is a good idea to specify a URL that points to the home page of the preview site.

If the `defaultNoAssetPreviewURL` property is not set, the default preview URL is specified by the `defaultPreviewURL` property. If neither property is set, the default preview URL is the first URL in the `previewURLs` list that does not include a `$[template.url]` property.

Specifying Default Preview Users and Sites

Visual Merchandising requires a default preview user and, in a multisite environment, a default preview site. Specify default preview users and sites with the properties of the `atg/dynamo/service/preview/PreviewManager` component.

The `defaultPreviewUserId` and `defaultPreviewSiteId` properties specify the preview user and site that are used when a merchandiser clicks the Layout button in Merchandising when no asset is currently loaded in the Details pane.

If `defaultPreviewUserId` is not set, the default preview user is an anonymous user. If `defaultPreviewSiteId` is not set, the default preview site is the first site in the ordered list of sites.

Specifying a Preview Domain

If your environment includes remote preview (that is, the preview application is deployed on servers that do not run the Business Control Center), you must specify a preview domain or host name.

If your environment uses a domain name-based URL strategy, both the remote preview application and the publishing server must run in the same domain. When merchandisers launch a preview session, Merchandising attempts to obtain the domain from the server name in the preview URL. To ensure the correct domain is used, specify a preview domain with the `previewDomain` property of the `atg/dynamo/service/preview/PreviewManager` component. For example:

```
# PreviewManager.properties
# Explicitly sets the document.domain property in both the BCC page and the
#preview page to "oracle.com"
previewDomain=oracle.com
```

Configuring Browser Security Settings for Merchandising Users

If your environment includes remote preview, configure all browsers from which users access Merchandising to automatically accept, or prompt for acceptance of, third-party cookies. Alternately, specify the preview domain as a trusted domain from which cookies are always accepted. This type of configuration is recommended for all supported browsers but is required for Microsoft Internet Explorer 9.

To configure Microsoft Internet Explorer 9 to accept cookies from the preview server, complete the following steps. Note that this procedure configures an individual installation of Internet Explorer 9. You can also configure multiple installations with a Group Policy.

1. In Internet Explorer, click the Tools button, then click Internet options.

2. Click the Privacy tab, then move the slider to a position between the top and bottom, neither blocking all nor allowing all cookies.

3. Click Sites.

4. In the Address of website box, type the name of the preview server, including the domain name, then click Allow.

5. Click OK.

Visual Merchandising allows Merchandising users to select assets to organize and edit directly from a preview session. Visual Merchandising then automatically updates the preview session with the merchandiser's saved changes.

Visual Merchandising works through sensors, which detect specific elements encountered in a JSP when rendering an HTML page in a preview session launched in Merchandising. These sensors fire events that are detected by event listeners, which then insert CSS classes, called *item identifiers*, to class tag attributes when the HTML page is rendered. Item identifiers contain the repository item type and repository ID of the most recently accessed category, product, or SKU. When Merchandising users right-click a marked asset in a preview session, these classes are used to display the Visual Merchandising pop-up-menu.

You can configure Visual Merchandising to automatically insert item identifiers, but we recommend that you explicitly tag JSP pages for use with Visual Merchandising so you can more precisely control where item identifiers are inserted. You tag JSP pages for Visual Merchandising by inserting `<preview:repositoryItem>` tags in JSP pages.

The `<preview:repositoryItem>` tags work through calls to the `setExplicitIdentifier` method of the `/atg/dynamo/service/preview/PreviewTaggingUtils` component. `PreviewTaggingUtils` invokes the `setExplicitIdentifier` method of the `/atg/dynamo/service/preview/fluoroscope/sensor/PreviewTaggingSensor` component that fires events of type `PreviewTaggingSensor.ExplicitIdentifierPreviewTaggingSensorEvent`. This event updates the class attribute of a tagged page element (category or product) with an item identifier that starts with `atgRepo-atgrep:/` and include an asset URI composed of the repository name, item descriptor name, and repository ID. For example, the following class string identifies a specific product: `atgRepo-atgrep:/ProductCatalog/product/xprod2015`. Merchandising uses the item identifier to display the Visual Merchandising menu when a Merchandising user right-clicks the asset during a preview session.

For more information, see

Tagging Pages for Visual Merchandising

Configuring Visual Merchandising to automatically insert item identifiers in a page does not guarantee that HTML pages will get marked up the way you intend. Depending on how a page is coded, Visual Merchandising may not be able to determine where an asset begins and ends. For example, it might mark a product link as a category link if a category property was the last thing a Merchandising user accessed during the preview session. Then, when the user right-clicks the product link and selects Edit from the Visual Merchandising pop-up menu, the last-accessed category appears for editing instead of the product. This happens because the `RepositoryPropertySensor` fired an event when the category property was accessed.

Explicitly tagging JSP pages allows you to precisely control where item identifiers are inserted in a page, ensuring that Merchandising users can access the assets they intend to in Layout mode. This section describes how to use the custom `<preview:repositoryItem tag>` to explicitly tag pages.

Enabling Explicit Tagging

To enable explicit tagging with the `<preview:repositoryItem>` tag, set the value of the `previewTagsProcessingEnabled` property of the `/atg/dynamo/service/preview/PreviewTaggingUtils` component to `true`.

On the server where preview is running, configure the `atg.service.fluoroscope.SensorManagerService` component:

1. Enable the `PreviewTaggingSensor` sensor.

2. Enable the `CachingListener`, `HtmlMarkingSensorListener` and `PositionMarkingSensorListener` listeners.

3. Make these settings global to all sessions.

Preview Tag Library

The Preview tag library contains a single tag, `<preview:repositoryItem>`. The `<preview:repositoryItem>` tag inserts item identifiers in JSP pages. The tag library source and definition files are included in your installation at the following location:

```
<ATG11dir>/WebUI/Preview/taglibs/webui-preview/lib
```

Copy the library to your site's WAR file. In order to call the `<preview:repositoryItem>` tag, the JSP must include this taglib directive:

```
<%@ taglib prefix="preview" uri="http://www.atg.com/taglibs/webui-preview_rt" %>
```

preview:repositoryItem

The `<repositoryItem>` tag sets an explicit item identifier to be used in the body of a page. The identifier is calculated based on repository item data. The tag is defined as follows:

```
<preview:repositoryItem
  [item]
  [repositoryName]
  [itemType]
  [itemID]>
  ...
</preview:repositoryItem>
```

Attribute	Description
item	Page parameter. (See the `categoryPanel.jsp` example that follows.)
repositoryName	Name of the repository that contains the asset
itemType	Asset type
itemID	Asset ID

In the following example, the `<preview:repositoryItem>` tag calculates a class for the specified category repository item.

```
<dsp:getvalueof var="currentCategoryId"  bean="/atg/endeca/assembler/cartridge/
StoreCartridgeTools.currentCategoryId"/>

<preview:repositoryItem itemId="${currentCategoryId}"
          itemType="category"
          repositoryName="ProductCatalog">
```

```
    <h2 class="title">
     <c:out value="${contentItem.headerTitle}" />
    </h2>
</preview:repositoryItem>
```

Configuring the Promotions Visibility Window

The Promotions Visibility window lets merchandisers see how their promotions behave during a simulated shopping experience in layout view. For more information, see *Previewing and Testing Promotions* in the *Merchandising Guide for Business Users*.

The session-scoped `/atg/remote/assetmanager/preview/service/PreviewPricingService` component specifies whether the Promotions Visibility window displays promotions that were not applied to the current cart in a multisite environment because they are not valid on the site that the preview user is currently browsing:

- If `true`, the left pane of the Promotions Visibility window includes the Ineligible promotions list.

- If `false`, the left pane of the Promotions Visibility window does not include the Ineligible promotions list, only the Applied and Not Applied lists.

Configuring the Asset Tray

The asset tray that appears at the top of the Merchandising window bookmarks assets that merchandisers can organize and edit, for example, with Multi Edit operations. For more information, see *Using the Asset Tray* in the *Merchandising Guide for Business Users*.

The `/atg/remote/assetmanager/workbench/service/WorkbenchService` component defines configuration settings that customize the behavior of the asset tray:

Property	Description
addNewTilesToEnd	Specifies whether new tiles or stacks are added to the beginning (left-hand side) or end (right-hand side) of the asset tray.
	This property controls cases where tiles added to the asset tray by any means other than drag and drop, for example, with the Add to Asset Tray command. When merchandisers drag assets to the asset tray, tiles are created at the spot where they are dropped.
	If you set this property to true, new tiles are added to the right-hand side of the asset tray.
	If you set this property to false (default), new tiles are added to the left-hand side of the asset tray.

Property	Description
allowLinkOperations	Specifies whether merchandisers can perform link operations in the asset tray. For more information, see Linking Assets in the Asset Tray in the *Merchandising Guide for Business Users*. By default, this property is set to `true`.
autoAddNewAssets	Specifies whether Merchandising automatically adds a tile to the asset tray any time a merchandiser creates a new asset. Tiles are added only for assets created in Merchandising, not for imported assets. By default, this property is set to `true`.
confirmBeforeRemovingReport	Specifies whether Merchandising displays a confirmation dialog when merchandisers add assets to the stack after a Multi Edit operation has completed. Adding assets to the stack automatically removes the report for that Multi Edit operation. By default, this property is set to `true`.
groupByDefault	Specifies how tiles are grouped when merchandisers add assets to the asset tray. If you set this this property to `true` (default), Merchandising automatically groups assets added to the asset tray based on asset type. For example, products are added to one stack while SKUs are added to another. If you set this property to `false`, Merchandising creates a separate tile for each asset added to the asset tray.

5   Configuring Price Lists in Merchandising

Core Commerce supports two types of pricing for web sites, SKU pricing and price lists.

SKU pricing is a basic pricing model that lets you specify the same prices for a SKU to all users. Price lists provide a more sophisticated model that allows you to target a set of prices to a specific group of customers.

Typically, most multisite environments use price lists to price their items. With SKU-based pricing, you may only specify one price per item. Price lists are more flexible and make it possible to have different prices for the same SKU.

For example, if you are using a multisite environment, you could implement a price list called German Prices to show one set of prices in Euros to shoppers who visit your German web site, and another price list called US Prices to show another set of prices in dollars to shoppers who visit your US web site. You could also use price lists to create time-based prices, for example, modifying the price of an item for a specified amount of time.

Typically, prices are defined in a base price list and inherited by other price lists. This allows you to create a hierarchy of price lists and change prices only when you want to override inherited values or specify volume discounts.

Configuring Pricing Resources

Before you can work with price lists in Merchandising, you must configure pricing resources to implement, display, and calculate order prices. The *Platform Programming Guide* includes several chapters that describe how to configure pricing. Start with *Commerce Pricing Services Overview*, which provides an introduction to concepts and terms. The chapters that follow it provide details on default pricing services and describe how to extend them.

This chapter provides an overview of price lists and describes how to configure price list access so merchandisers can perform price list tasks in Merchandising. See *Creating and Managing Price Lists* in the *Merchandising Guide for Business Users* to learn how to create, edit, and delete price lists in Merchandising.

Configuring Price List Access

By default, Merchandising assigns user roles different levels of access to price lists. For example, by default, users with the Merchandising role have full access to create, edit, and delete price lists. You can change access rights for individual price lists through the Security tab in the Merchandising Price Lists or Price Lists view. These rights replace the default security policy for the price list. Before you change price list access, read the chapter *Business Control Center Security* in the *Business Control Center Administration and Development Guide* to learn about configuring asset security.

To set access rights for a price list:

1. In a Commerce Assets project, navigate to Price Lists.

2. Click the price list whose access rights you want to change.

3. Click the Security tab to display the default access rights for this price list.

    Note: Your own user profile must have Write, View Access Rights, and Set Access Rights privileges for price lists. Otherwise, the Security tab does not appear in the Price Lists view.

4. Clear the Use Default Security Policy checkbox. As described earlier, the Default Security Policy is the set of rights specified in the repository definition file for all items of this type. When you choose to define individual access rights for this price list, you are replacing the default rights.

    When you clear the checkbox, three buttons appear that you can use to specify access rights for this price list:

    *   Add Role: Allows you to define access rights to this price list for one or more roles. The roles can be global or organizational. The rights will apply to any user who has the specified role.

    *   Add Organization: Allows you to define access rights to this price list for one or more organizations. The rights will apply to any user or role that is assigned to the specified organization.

    *   Add User: Allows you to define access rights to this price list for one or more individual user profiles.

    The image below shows the Security tab with these buttons enabled:
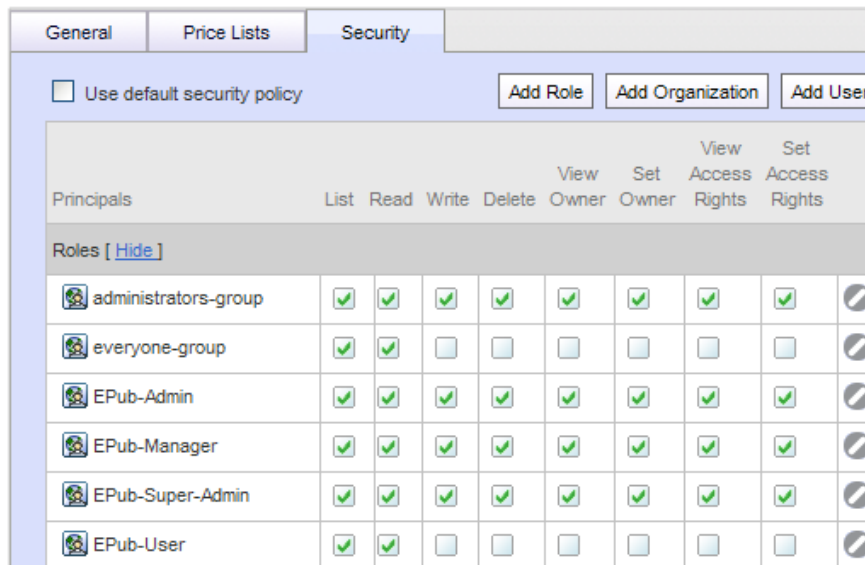
5. Click the appropriate button and use the asset picker to select the roles, organizations, or users for which you want to define access rights. Note that you can select a combination of all three if required.

   Important: You must specify access rights for all principals that need access to this price list.

6. For each access right, click the corresponding check box and select Allow, Deny, or Unspecified, as applicable. Unspecified means that the default security policy setting is used for this principal.

   To remove a role, organization, or user from this tab, click the Remove icon for that item.

7. When you have finished setting access rights for this folder, click Save. If an error appears informing you that you cannot remove List and Read access for yourself, make sure that you have specified access rights for your own profile or a role that is assigned to you.

Configuring Pricing Display in Merchandising

You can configure Merchandising to control how pricing UIs appear to business users. You can configure the following pricing strategies:

- Product- level price list pricing.

- SKU- level price list pricing.

- Products-and-Sub-SKUs level price list pricing. (A SKU can belong to multiple parent products and can have a different price for each parent product.)

- SKU pricing. (This is the default pricing strategy.)

The `/atg/remote/content/assetmanager/ContentConfiguration` component defines configuration settings that customize the display of pricing UIs in Merchandising:

Property	Description
`showProductPriceLists`	If you set this this property to `true`, the Price Lists tab appears in the Details pane for each Product in your catalog. By default, this property is set to `false`.
`showSKUPriceLists`	If you set this this property to `true`, the Price Lists tab appears in the Details pane for each SKU in your catalog. By default, this property is set to `false`.
`showProductsSubSKUsPriceLists`	If you set this property property to `true`, Merchandising displays information that support a Products-and-Sub-SKUs pricing strategy on the Price Lists tab in the Details pane for each SKU in your catalog. By default, this property is set to `false`.

Property	Description
showSKUPricing	If you set this this property to `true` (default), the SKU Pricing tab appears in the Details pane for each SKU in your catalog.
	If you set this property to `false`, the SKU Pricing tab does not appear.

Index

UIs, 3
uninstalling, 37
merchandising tasks, 3
Merchandising tools, 3
MySQL database, 8
configuring tables for, 8

P

partial URLs, 29
preview
Internet Explorer, 40
Visual Merchandising, 39
preview server, 11
Preview tag library, 42
previewing
promotions, 43
price lists
about, 45
projects
overview, 32
used to manage assets, 32
promotions
multisite, 18
sites tab, 18
tender-based, 30
testing, 43
promotions visibility
access rights, 26

R

removed items
precedence in suggested value, 36
repositories
allowing import to, 22
restricting import to, 22
securing, 19, 19
unsecuring, 19, 20
used in deployment, 36
views, 21

S

search
configuring default asset, 30
secured repositories, 19
security
asset, 23
setting up
an evaluation copy of Merchandising, 8
source repository, 36
staging and production targets, 4, 11
starting
Merchandising, 12
stopping

Merchandising, 37
suggested collection value, 36

T

tag library (see Preview tag library)
target repository, 36
target server cluster
configuring, 18
tasks
merchandising, 3
Tender-based promotions, 30
testing
promotions, 43
tools
for administrators and programmers, 3
for merchandisers, 4
Merchandising, 3

U

Uncategorized groups, 31
uninstalling Merchandising, 37
unsecured repositories, 19

V

versioned
databases, 8
Visual Merchandising
configuring, 39
configuring preview, 39
explicit tagging, 41
visual merchandising
promotions, 43

W

workflow
configuring, 32
customizing, 33
deployment model, 33
role in project, 32
security, 33