ORACLE®

ATG WEB COMMERCE

Merchandising

Version 10.0.2

Administration Guide

Oracle ATG
One Main Street
Cambridge, MA 02142
USA

ATG Merchandising Administration Guide

Document Version

MERCH10.0.2 MERCHADMINv1 4/15/2011

Copyright

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/us/corporate/accessibility/index.html.

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/support/contact.html or visit http://www.oracle.com/accessibility/support.html if you are hearing impaired.

Contents

1 Introduction

ATG Merchandising provides sophisticated tools to facilitate the creation and management of catalog content, promotions, pricing, and search results for one or more online commerce sites. When ATG Merchandising is purchased and licensed, it is installed with the ATG platform.

About this Document

This document summarizes ATG Merchandising's main concepts, and tells how to configure the application so that its features can be used effectively to meet the needs of the merchandising initiative.

Audience

This guide is intended for the following audiences:

- ATG Merchandising administrators

  Administrators install, configure, and maintain the ATG Merchandising environment. They also configure it so that merchandisers can create projects and Merchandising options to support business requirements and goals.

- ATG Merchandising business users (merchandisers)

  Merchandisers create and maintain product catalogs, price lists, and promotions that are visible on one or more commerce Web sites.

- ATG Merchandising page developers/programmers

  Page developers integrate site page components with ATG Merchandising content that can target the specific needs of site visitors.

Document Conventions

This document uses the following conventions:

Convention	Description
`<ATG10dir>`	The directory where ATG is installed. The default is `C:\ATG\ATG10.0.1`.

`Code samples` and `file names`	Indicates a line of code, a file name or path, or the name of an ATG component or property.
`Variables`	Indicates a variable in a code sample or file name that should be replaced with an appropriate value.

Related Documentation

This document explains ATG Merchandising concepts and components, and provides instructions for preparing and customizing the ATG Merchandising environment for the internal ATG Merchandising user.

ATG Merchandising interacts with other ATG applications, including ATG Commerce and ATG Content Administration. If you use Search Merchandising, interaction also occurs with ATG Search. You perform most ATG Merchandising tasks through the ATG Business Control Center (BCC) user interface. You perform the remaining tasks through the ATG Control Center (ACC) user interface.

The following table lists manuals that provide instructions for installing and working with ATG Merchandising and the other ATG components. A document may be listed more than once, when it contains significant information about more than one product.

Product or Utility	**Documentation**
ATG platform	*ATG Installation and Configuration Guide*
ATG Commerce	*ATG Commerce Programming Guide* *ATG Commerce Guide to Setting Up a Store*
ATG Content Administration	*ATG Content Administration Programming Guide* *ATG Content Administration Guide for Business Users* *ATG Business Control Center Administration and Development Guide*
ATG Control Center (ACC)	*ATG Personalization Guide for Business Users* *ATG Commerce Guide to Setting Up a Store*
ATG Business Control Center (BCC)	*ATG Business Control Center User's Guide* *ATG Business Control Center Administration and Development Guide*
ATG Merchandising	*ATG Merchandising Guide for Business Users* *ATG Merchandising Administration Guide*

Product or Utility	Documentation
ATG Search	*ATG Search Installation and Configuration Guide* *ATG Search Administration Guide* *ATG Search Query Guide* *ATG Merchandising Guide for Business Users*
Site Administration utility	*ATG Multisite Administration Guide*

2 Overview

ATG Merchandising provides a merchandising 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, ATG Merchandising employs some features of ATG 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 ATG Merchandising to support more than one commerce site. For an overview of multisite concepts and possibilities, see the *ATG Multisite Administration Guide*.

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

ATG Merchandising Tools

The ATG Business Control Center (BCC) provides the user interface for most ATG Merchandising and ATG 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 ATG 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.
- Workflows for ATG Merchandising projects. See Working with Projects.
- The topology for deployment, including staging and production (live) sites. See *Setting Up Deployment* in the *ATG Content Administration Programming Guide*.

- Configurations that influence the search results when ATG Merchandising interacts with ATG Search. See information about Search Merchandising in the *ATG Merchandising Guide for Business Users*.
- View mapping and task configuration file settings to customize the ATG Merchandising UI. See the chapter *Customizing the UI for Specific Tasks and Activities* in the *ATG Business Control Center Administration and Development Guide*.
- Site pages, to implement displays of content provided by ATG Merchandising assets, such as targeters, scenarios, and faceted search results. See the *ATG Page Developer's Guide*.

Tools for Merchandisers

Business users (merchandisers) use ATG 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. ATG Merchandising tools enable merchandisers to do the following:

- Create projects to track the creation and management of commerce assets. See the *ATG Merchandising Guide for Business Users*.
- Create catalogs and populate them with categories, products, and SKUs. See the *ATG Merchandising Guide for Business Users*.
- Manage the catalog items, images, and copy that will be deployed to a live Web site. See the *ATG Merchandising Guide for Business Users*.
- Create and manage personalization assets: user segments, content groups, and targeters. These assets can be site-aware, specifying an execution site and a content search site. See the *ATG Business Control Center User's Guide* and the *ATG Personalization Guide for Business Users*.
- Manage price lists: manage price inheritance; set standard, sale, and volume pricing; assign custom prices to individuals or groups. See the *ATG Merchandising Guide for Business Users*.
- Create promotions and scenarios. See *Creating and Maintaining Promotions* in the *ATG Commerce Guide to Setting Up a Store* and *Creating Scenarios* in the *ATG Personalization Guide for Business Users*.
- Manage Search Merchandising features, including facets, for your catalogs. See the *ATG Merchandising Guide for Business Users*.

Assets and Asset Management

ATG Content Administration supports asset management for ATG Merchandising and other ATG applications. It is a Web content management system that enables you to do the following:

- Create and edit content
- Maintain versioned content
- Manage the project workflow

- Publish content to one or more commerce sites

The content is created and managed in units called assets.

Assets

Assets are the fundamental building blocks of Content Administration. Assets include items such as products, catalogs, folders, scenarios, targeters, promotions, price lists, text files, HTML files, and binary files.

A merchandiser creates catalog assets and populates them with categories, products, and SKUs. The products and SKUs can have images and text copy that a merchandiser creates, refines, and ultimately deploys to a live Web site. To find, display, and promote specified catalog assets to specified users, a merchandiser creates personalization assets called promotions, targeters, and scenarios.

Folders

A folder acts as a container to organize other assets. Not every asset type is contained in a folder; for example, SKUs are organized under products, not folders.

Media Assets

The types of media assets are:

- Media-internal-binary: Binary files, such as images
- Media-internal-text: Blocks of text
- Media-external: Links to an image or a text file

A media asset is usually associated with a catalog asset. For example, an image is often associated with a SKU.

Catalogs

A catalog contains a collection of other commerce assets: categories, products, and SKUs. *Categories* organize products into a hierarchy. A category may contain products, other categories, and child catalogs. For example, a category called Women's Sweaters might contain two categories: Short-Sleeved and Long-Sleeved. A *product* exists under a category and is represented by one or more stock keeping units (SKUs). Each *SKU* represents a unique attribute or combination of attributes, such as material, size, or color.

A single site can use more than one catalog. For example, each catalog could contain a unique group of products and categories.

In a multisite environment, a catalog can be associated with one or more registered sites.

Catalog Links

A link is created automatically when you add an asset to a catalog. The link is an association between the catalog asset and the asset that contains it, called the parent. An asset may have more than one parent;

for example, a Mini-Blender product could be the child of two categories: Small Kitchen Tools, and Gifts for the Chef. When a single asset is linked to multiple parents, a change to the asset, such as price, is reflected in all locations.

Projects

A merchandiser creates and manages assets within a project. The project is associated with a workflow, which lists a sequence of tasks, such as reviewing work and deploying assets. The workflow enforces the execution sequence of the tasks and establishes the eligibility of internal users to perform them. The project tracks the creation, modification, and deletion of each asset. When the project is complete (deployed and accepted), its assets are checked in to the Content Administration versioned repository. Every new version of an asset is saved as a discrete version.

ATG Repositories

The ATG repository technology provides unique access to ATG databases. Merchandisers and others do not address the databases directly; instead, they specify data items in the repositories. For an explanation of repositories, see the *ATG Repository Guide* and the chapters *Understanding the Content Development Environment* and *Creating a Versioned Module* in the *ATG Content Administration Programming Guide*.

The Content Administration (Versioned) Database

ATG Content Administration maintains versions of ATG Merchandising assets in the ATG Content Administration database, also called the versioned database. This enables ATG Content Administration to maintain asset history, to coordinate multi-user access to the assets, and to ensure that the latest changes are deployed to the target sites.

The Production Database

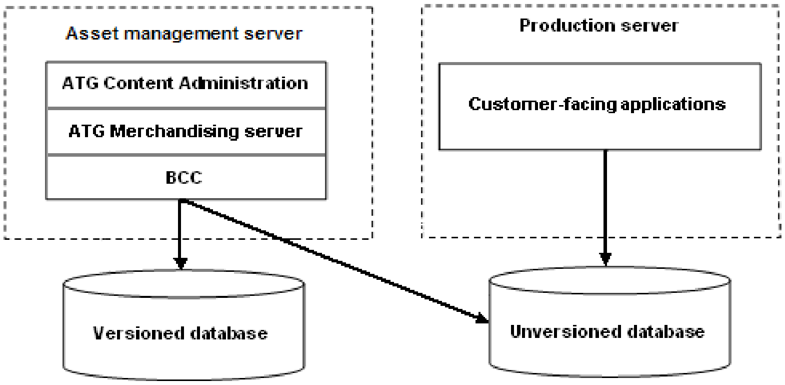
The production database supports the live commerce site. The production database contains only the most recently deployed version of each asset.

ATG Merchandising Architecture

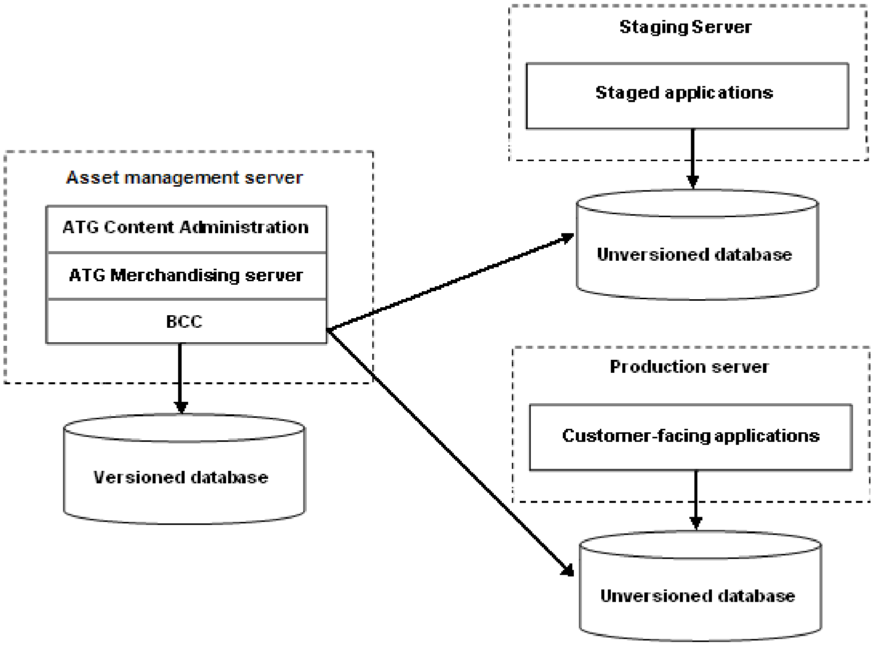
ATG Merchandising resides on the asset management server, along with ATG Content Administration. When the environment includes ATG Search, ATG Search Administration is on this same server.

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 ATG Merchandising deployment architecture for a production deployment target.



The following diagram shows the deployment architecture for staging and production targets.



3 Configuring ATG Merchandising

The installation of the ATG platform includes ATG 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 ATG Merchandising

ATG Merchandising requires the ATG Business Control Center and ATG Content Administration and can be installed with the ATG platform. For information, refer to the *ATG Installation and Configuration Guide*.

Before you install the ATG platform and ATG Merchandising:

- Decide whether ATG Search is required, and make sure it is installed if it is needed.
- Be sure the JDK, application server, and database are installed.

Then, proceed with the installation. In addition to installation instructions, the *ATG Installation and Configuration Guide* contains important configuration information.

The remainder of this chapter assumes that the ATG platform has been installed.

After Installation: First Steps

After ATG 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 ATG Merchandising software can run. This includes assembling and deploying the application, publishing, and production servers. See Assembling an ATG Merchandising Application.
- Install the ACC, if it is required.
- Start the BCC.
- Configure a working ATG Merchandising development environment.

You can use the ATG Configuration and Installation Manager (CIM) to guide you through the following tasks. CIM interacts with a SOLID evaluation database or a production-ready database.

- 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.

- As required, create tables that support ATG Search integration and ATG Search features.
- Create ATG Merchandising tables and import required content.
- Configure, assemble, and deploy instances of the ATG 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. The *ATG Commerce Reference Store Installation and Configuration Guide* contains step-by-step instructions for starting and using CIM for these tasks.

If you do not use CIM, you must complete the same tasks using other means. The basic reference for these tasks is the *ATG 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 in this manual. For more database configuration information, see the *ATG Content Administration Programming Guide* and the *ATG Repository Guide*.
- For instructions to assemble and deploy the EAR files for the production and publishing servers, see:
  - Assembling an ATG Merchandising Application in this manual
  - The *ATG Content Administration Programming Guide*
  - The chapter *Developing and Assembling Nucleus-based Applications* in the *ATG Programming Guide*

Configuring Database Tables

Note: 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.

This section contains the following topics:

Creating Versioned ATG Business Commerce Tables

Creating Versioned ATG Search Tables and Versioned Catalog Tables with ATG Search Support

Importing Required Data

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

Configuring Database Tables on an Evaluation Database

The SOLID evaluation database is provided with the installation, as a convenience for evaluation users. You must create database tables as part of the evaluation setup.

To use the SOLID evaluation database on Microsoft Windows:

1. Start the database by opening the Start menu and choosing Programs > ATG 10.0.1 > ATG Content Administration > Start SOLID Server. See the section *Starting the SOLID SQL Database* in the *ATG Installation and Configuration Guide*.
2. Follow the instructions below, to create and populate versioned database tables as required by ATG Merchandising and, optionally, ATG Consumer Commerce, ATG Business Commerce, and ATG Search. Note that the instructions include separate steps for the SOLID database and for production databases.
3. If you intend to use other ATG applications beside ATG Search, see the documentation for each application to learn the database table requirements.
4. Import data required by ATG Merchandising. See Importing Required Data.

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 the ATG asset management server.

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

2. Start the database.

   Follow instructions in the documentation from the database vendor.

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

   ```
   <ATG10dir>/BIZUI/sql/install/database-vendor/bizui_all_ddl.sql
   ```

   For more information, see the section *Create ATG Content Administration Tables* in the *ATG Content Administration Programming Guide*.

4. If you have ATG Search, create the tables that support ATG Search integration. Run this script:

```
<ATG10dir>/DAF/Search/Index/sql/db_components/
database-vendor/search_ddl.sql
```

For more information, see the section *Creating the Search Database Tables* in the chapter *Installing ATG Search* in the *ATG Search Installation and Configuration Guide*.

5. Follow the instructions below to create and populate versioned database tables as required by ATG Merchandising and, optionally, ATG Consumer Commerce, ATG Business Commerce, and ATG Search. Note that the instructions include separate steps for the SOLID database and for production databases.

6. Initialize and populate the ATG 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 *ATG Content Administration Programming Guide*.

7. Import data required by ATG Content Administration and ATG Merchandising. See Importing Required Data.

8. If you use additional ATG applications, see the documentation for each ATG application to learn the database tables requirements for the application.

Creating Versioned ATG Commerce Tables and Versioned Catalog Tables

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

To create versioned ATG Commerce tables and versioned catalog tables on a production-ready database, run the following scripts:

```
<ATG10dir>/DCS/Versioned/sql/install/database-vendor/dcs_versioned_ddl.sql

<ATG10dir>/DAF/Search/Index/sql/db_components/database-vendor/
search_ddl.sql

<ATG10dir>/DAF/Search/Versioned/sql/db_components/database-vendor/
versioned_search_site_ddl.sql

<ATG10dir>/DAF/Search/Routing/sql/db_components/database-vendor/
routing_ddl.sql
```

To create the versioned ATG Commerce tables on a SOLID evaluation database, open a command prompt and enter the following commands, which include the default arguments for a SOLID database:

```
<ATG10dir>\DAS\solid\i486-unknown-win32\solsql.exe "tcp 1313"
publishing publishing <ATG10dir>\DCS\Versioned\sql\install\solid\
dcs_versioned_ddl.sql

<ATG10dir>DAS\solid\i486-unknown-win32\solsql.exe "tcp 1313"
publishing publishing <ATG10dir>\DAF\Search\Index\sql\db_components\solid\
search_ddl.sql
```

```
<ATG10dir>DAS\solid\i486-unknown-win32\solsql.exe "tcp 1313"
publishing publishing <ATG10dir>\DAF\Search\Versioned\sql\
db_components\solid\versioned_search_site_ddl.sql

<ATG10dir>DAS\solid\i486-unknown-win32\solsql.exe "tcp 1313"
publishing publishing <ATG10dir>\DAF\Search\Routing\
sql\db_components\solid\routing_ddl.sql
```

where: `<ATG10dir>` is the path to the ATG 10 directory, `1313` is the port for the SOLID database, and `publishing` is the username and password for the ATG Merchandising table space on SOLID.

Creating Versioned ATG Business Commerce Tables

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

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

```
<ATG10dir>/B2BCommerce/Versioned/sql/install/database-vendor/
b2bcommerce_versioned_ddl.sql
```

To create the versioned ATG Business Commerce tables on a SOLID evaluation database, open a command prompt and enter the following command, which includes the default arguments for a SOLID database:

```
<ATG10dir>\DAS\solid\i486-unknown-win32\solsql.exe "tcp 1313"
publishing publishing <ATG10dir>\B2BCommerce\Versioned\sql\
install\solid\b2bcommerce_versioned_ddl.sql
```

where: `<ATG10dir>` is the path to the ATG 10 directory, 1313 is the port for the SOLID database, and `publishing` is the username and password used for the ATG Merchandising table space on SOLID.

Creating Versioned ATG Search Tables and Versioned Catalog Tables with ATG Search Support

Before you can use ATG Merchandising with the ATG Search add-on features (Search Merchandising and Faceted Search), you must create versioned database tables that support these features. For additional information, see the section *Creating the Search Database Tables* in the *Installing ATG Search* chapter of the *ATG Search Installation and Configuration Guide*.

To create the versioned ATG Search tables on a production-ready database :run the following scripts:

```
<ATG10dir>/DAF/Search/versioned/sql/db_components/database-vendor/
versioned_refinement_ddl.sql

<ATG10dir>/DCS/Search/Versioned/sql/install/database-vendor/
dcs_search_versioned_ddl.sql
```

```
<ATG10dir>/DAF/Search/common/sql/db_components/database-vendor/
searchtesting_ddl.sql
```

To create the versioned Search tables on a SOLID evaluation database, open a command prompt and enter the following commands, which include the default arguments for a SOLID database:

```
<ATG10dir>\DAS\solid\i486-unknown-win32\solsql.exe "tcp 1313"
publishing publishing <ATG10dir>\DAF\Search\Versioned\sql\
db_components\solid\versioned_refinement_ddl.sql

<ATG10dir>\DAS\solid\i486-unknown-win32\solsql.exe "tcp 1313"
publishing publishing <ATG10dir>\DCS\Search\Versioned\sql\
install\solid\dcs_search_versioned_ddl.sql

<ATG10dir>\DAS\solid\i486-unknown-win32\solsql.exe "tcp 1313"
publishing publishing <ATG10dir>\DAF\Search\Common\sql\
db_components\solid\searchtesting_ddl.sql
```

where: `<ATG10dir>` is the path to the ATG 10 directory, `1313` is the port for the SOLID database, and `publishing` is the username and password used for the ATG Merchandising table space on SOLID.

Importing Required Data

This step is not required for SOLID database users.

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 *ATG Content Administration Programming Guide*.

The default `DataSource` connection pool, `JTDataSource`, uses the `FakeXADataSource` component, which is configured by default for the SOLID database. When you use a database other than SOLID, you must configure the desired connection pool properties, but note that this data source should be used only to run ATG data import scripts. See the section *Configuring ATG Data Sources for Data Import* in the *ATG Installation and Configuration Guide*.

You must import ATG Merchandising data before you can use ATG 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 `<ATG10dir>/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:

```
<ATG10dir>/Publishing/base/install/importPublishing.bat
<ATG10dir>/BIZUI/install/importBizui.bat
<ATG10dir>/AssetUI/install/importAssetUI.bat
<ATG10dir>/DPS-UI/install/importDPSUI.bat
<ATG10dir>/BCC/install/importBCC.bat
```

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

   ```
   <ATG10dir>/DCS-UI/install/importMerchandising.bat|sh
   ```

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

   ```
   <ATG10dir>/DCS-UI/Versioned/install/importDCSUIVersioned.bat|sh
   ```

2. If you have ATG Merchandising and ATG Search, run the following script to import the view mappings that are used to display search configuration and search configuration folder information:

   ```
   <ATG10dir>/AssetUI/Search/install/importAssetUISearch.bat|sh
   ```

3. Run the following script to import the view mapping content that is required to use ATG Business Commerce or ATG Consumer Commerce in conjunction with ATG Search:

   ```
   <ATG10dir>/DCS-UI/Search/install/importDCSUISearchCustomCatalogs.bat|sh
   ```

4. Run the following script:

   ```
   <ATG10dir>/DCS-UI/Search/install/importSearchTesting.bat|sh
   ```

Assembling an ATG Merchandising Application

Before you can access the ATG Merchandising environment, the ATG 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 ATG Merchandising. When the application assembly is finished, you deploy and start the application.

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

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

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

To build an application to run with ATG Merchandising, include the following modules. See the ATG Merchandising Modules table for information about each module.

- DCS-UI.Versioned
- DCS-UI.Search and DCS.Search.Versioned, to support ATG Search
- Either B2CCommerce.Versioned, for ATG Consumer Commerce, or B2BCommerce.Versioned, for ATG Business Commerce
- 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

If your environment includes a staging target, use the -layer stagingandprod flag. For information about deployment targets, see *Target Site Basics* in the *ATG 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 *ATG Business Control Center Administration and Development Guide*.

The modules that you use for the asset management server, where ATG 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 ATG 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 (ATG Consumer Commerce or ATG Business Commerce), as well as other ATG Commerce features. For example, the following command includes ATG Merchandising (DCS-UI.Versioned), BCC components (PubPortlet), and ATG Consumer Commerce (B2CCommerce.Versioned):

```
<ATG10dir>/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.

ATG Merchandising Modules

The following table lists the modules that you are most likely to use with ATG Merchandising. More modules are listed in the *ATG Modules* appendix in the *ATG Programming Guide*.

Module	Description
DCS.DynamicCustomCatalogs.Versioned	Included with DCS-UI.Versioned. Supports multisite functionality.

Module	Description
DCS-UI	Included with DCS-UI.Versioned. Contains sub-modules that contain HTML and Flex UIs for ATG Commerce applications that run on ATG Content Administration. MerchandisingProductCatalog and anything that is not specific to the implementation of ATG Merchandising has been moved to this module from DCS-UI.management (now DCS-UI.Versioned).
DCS-UI.Versioned	(Formerly DCS-UI.management.) Contains an HTML UI in which ATG Merchandising activities are performed. This module accesses ATG Content Administration (Publishing.base, BIZUI), and the UI elements that are used to build the UIs for a Content Administration environment (AssetUI), ATG Commerce, and ATG Merchandising (DCS). It also makes ATG Merchandising and ATG Content Administration available to a single instance. Including this module also includes DCS.DynamicCustomCatalogs.Versioned, which supports multisite functionality.
DCS-UI.Search	Contains an HTML UI that is used by Search Merchandising and Faceted Search, which are ATG Search add-on features of ATG Merchandising. This module accesses the ATG Commerce parts of the ATG Search integration framework (DCS.Search), ATG Merchandising (DCS-UI.Versioned), and additions to the UI framework in support of these ATG Search add-on features (AssetUI.Search). Files from the DCS-UI.Search module that were in the //applications/DCS-UI depot have been moved to //product.
DCS.Search.Versioned	Contains code that supports versioned ATG Commerce extensions to the ATG Search integration framework. If you use ATG Search, ATG Content Administration and ATG Commerce, you must specify this module.
DCS.Search.CustomCatalogs.Versioned	Contains code that supports versioned custom catalog extensions to ATG Search. If you use ATG Search, ATG Content Administration and ATG Business Commerce with ATG Consumer Commerce, you must specify this module.

Module	Description
`B2BCommerce.Versioned`	Contains ATG Content Administration extensions to the ATG Business product. Including this module also includes `B2BCommerce`.
`B2CCommerce.Versioned`	Contains ATG Content Administration extensions to the ATG Consumer Commerce product. Including this module also includes `B2CCommerce`.
`PubPortlet`	Contains BCC components. Required by ATG Merchandising.
`SiteAdmin.Versioned`	Contains the Site Administration components for a system that does not support ATG Search. Required only if users on the server need the Site Administration utility.

Starting the ATG Business Control Center

The BCC is the internal user interface to the ATG Merchandising environment. You use the BCC to create, preview, approve, deploy, and revise site content, and to launch ATG Merchandising and other ATG applications.

Before you log in to the BCC, you must:

- Start the appropriate database.
- Obtain the appropriate user profile and role requirements. The user profile specifies the user name and password that you must use to log in.

To open the BCC login page:

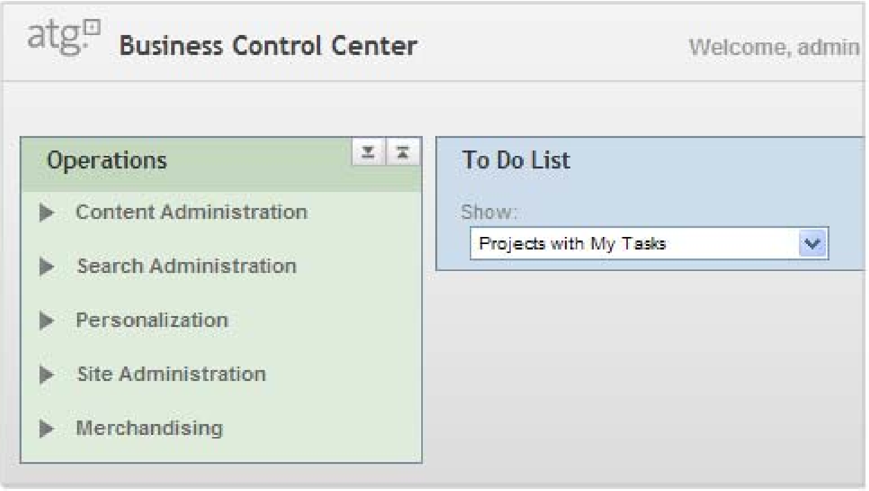
1. Open a browser window and enter the following URL address:

   `http://`*`hostname`*`:`*`port`*`/atg/bcc`

   where: *`hostname`* is the name of the server where ATG Merchandising is running and *`port`* is the port that your application server specifies for ATG applications. **Note:** If ATG Merchandising is on a local machine, you may substitute `localhost` for *`hostname`*.
2. When the BCC opens, enter your username and password.

   These fields are case-sensitive. For evaluation purposes, the values are `admin` and `admin`.

After a correct login, the BCC home page appears:



See the *Introduction* chapter of the *ATG Business Control Center User's Guide* for more information about the following subjects:

- User profile and role requirements for login
- Instructions to start the ATG products that you want to run with ATG Merchandising
- How to start the BCC

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

ATG Business Control Center User's Guide

ATG Business Control Center Administration and Development Guide

ATG Content Administration Guide for Business Users

Using the ATG Control Center

The 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 *ATG Installation and Configuration Guide* and the *ATG Personalization Guide for Business Users* for instructions about ACC installation, startup, and use.

Configuring for ATG Merchandising Development

The administrator configures the ATG Merchandising environment to accommodate the development needs of the merchandising initiative. Each step in this procedure is explained later in this chapter or in subsequent chapters of this book.

1. Create accounts for the internal users of ATG Merchandising. See Setting Up Roles and User Accounts.
2. Configure the Catalog Maintenance Service and the Catalog Completion Service for each target server. See Configuring a Target Cluster.
3. Set up the repositories to be secured or unsecured to meet the requirements of the implementation. See Configuring Secured and Unsecured Repositories.
4. (Optional) Turn off Live Scrolling. See Live Scrolling.
5. Configure projects, with workflows, to be available to ATG Merchandisers. See Working with Projects.
6. Set up deployment resources. See Configuring Deployment Resources.
7. Understand pricing models. Create price lists to minimize price configurations and maintenance. See the chapter Configuring Pricing in ATG Merchandising.
8. If you want to use the ATG Search Merchandising features, read the following chapters in this manual:
   - Administering Faceted Search
   - Configuring Search Merchandising
   - Configuring Search Testing

Setting Up Roles and User Accounts

Without modification after installation, user access to the ATG Merchandising application is limited to the following:

- The ATG Content Administration publishing user (username and password: `publishing`); an evaluation account that enables access to the BCC.
- The administrative user (username and password: `admin`); available for evaluation purposes; includes setting profiles for other users.

  Note: Administrative tasks that are related to ATG Search, such as creating search configuration folders and specifying the properties that affect how search results appear to customers, are applicable and visible only to users who have ATG Search.
- The merchandising user (username and password: `merchandising`); enables access to all non-administrative parts of the BCC.
- Roles that have been given access to ATG Content Administration.

Roles for Merchandisers

The following roles and internal users (members) are among those that are preconfigured in the ATG Merchandising installation:

Global Role	Members
EPub-User, for users who create and manage assets	merchandising
EPub-Super-Admin, for users who review and approve content, and who can deploy content to a production target.	admin, publishing

You give merchandising access to a user by assigning two roles:

- An existing role that has access privileges to ATG Content Administration, such as `EPub-User` or `EPub-Manager` from the `EPublishing` folder
- The role `100001-member` from the `bizui` folder

You can also create new roles. The new role must have access to the BCC and the ACC.

Access to the BCC

For a list, with descriptions, of requirements for login to the BCC, see *Login Requirements* in the *ATG Business Control Center Administration and Development Guide*.

For a list, with descriptions, of preconfigured roles that CA provides for use in the BCC, see *ATG Content Administration Users* in the *ATG Content Administration Programming Guide*.

For more information about creating organizations and roles in the BCC, see the chapter *Creating Organizations and Roles* in the *ATG Business Control Center Administration and Development Guide*.

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

Access to the ACC

ACC users need to be set up as Control Center Users (People and Organizations > Control Center Users) and assigned to specific ACC groups (People and Organizations > Control Center Groups). To manage ATG Content Administration items, the groups need access to the following areas in the ACC:

- Scenarios task areas
- Targeting task areas
- Publishing > Workflows task area
- Publishing > Publishing task area

By default, the administrators-group that is provided with the ATG platform has access to the areas listed above. See *PublishingRepository Security* in the *ATG Content Administration Programming Guide* for information about the access rights of the administrators-group to the PublishingRepository.

For more information on setting up ACC users and groups, refer to the *ATG Personalization Programming Guide*.

Price List Access

Price Lists have their own interface for secure access configuration. See Configuring Price List Access.

Creating New Roles

To create a new role:

1. Log in as admin.
2. Open the BCC.
3. In the Operations list, expand Personalization.
4. Select Internal Users.
5. From the Show menu, select Organizations and Roles.
6. Click the Create icon (**+**) and select Role.
7. In the General tab of the Editing pane, enter a name for the role.
8. At the top right of the Editing pane, click the Select Parent button.
9. Use the asset picker to choose a parent folder for the role.

To assign the new role to a user:

1. From the Show menu, select Users.
2. In the Browse pane, select the user to add as a member.
3. In the Editing pane, click the Orgs and Roles tab.
4. In the Roles section, click the Add Existing button.
5. Use the asset picker to find and select the new role.
6. Click OK.

To view the new assignment of the user in the role configuration:

1. Find and select the role in the Browse pane.
2. In the Editing pane, click the Members tab.
3. The name of the assigned member appears in the Direct Assignees and All Members lists.

Creating Internal User Accounts

The same mechanism that establishes and administers site visitor profiles can be used to organize and administer *internal* users and organizations. You may create additional roles and internal users and roles that are related to merchandising. ATG Merchandising administrators can assign or restrict the rights of internal users and organizations regarding their access to assets and the administrative functions.

Access control refinements depend on the role, user organization, and access rights. Refer to the *ATG Business Control Center Administration and Development Guide* for extensive information about setting up organizations and roles, and instructions to configure BCC home page security. See also the section *Using Roles for Access Control* in the *ATG Personalization Programming Guide*.

To configure a new internal user account:

1. From the BCC, open a Merchandising project.
2. In the Operations list, expand Personalization.
3. Select Internal Users.
4. Click the Create icon (+) and select Internal User.
5. Use the tabs in the Editing pane to configure the login name, password, roles, organizations, and other aspects of the user account.
6. Click Save.

Configuring a Target Cluster

When you use ATG Merchandising with ATG Content Administration, you must make the following configurations for each agent server in a cluster that will receive deployed ATG Merchandising content:

- Specify the `DCS.PublishingAgent` module during application assembly and startup, to ensure that the Catalog Maintenance Service runs as expected.
- 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 stagingandprod` 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.

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 *ATG 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 *ATG 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.

ATG Merchandising uses three versioned ATG Commerce repositories.

- Merchandising Product Catalog. By default, this repository is secured when the ATG Business Commerce module is running, and is not secured when the ATG Consumer Commerce module is running.
- Price Lists. By default, this repository is secured.
- Claimable Repository. By default, this repository is not secured.

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

	Merchandising Product Catalog	**Price List**	**Claimable Repository**
Unsecured Repository	`<ATG10dir>/DCS/config/config.jar/atg/commerce/catalog/ProductCatalog`	`<ATG10dir>/DCS/config/config.jar/atg/commerce/pricing/priceLists/PriceLists`	`<ATG10dir>/DCS/config/config.jar/atg/commerce/claimable/ClaimableRepository`
Secured Repository	`<ATG10dir>/B2BCommerce/config/config.jar/atg/commerce/catalog/SecureProductCatalog.properties`	`<ATG10dir>/DCS/config/config.jar/atg/commerce/pricing/priceLists/SecurePriceLists.properties`	none
Secured Repository XML Repository Definition	`<ATG10dir>/DCS/Versioned/config/config.jar/atg/commerce/catalog/custom/secureCustomCatalog.xml`	`<ATG10dir>/DCS/Versioned/config/config.jar/atg/commerce/pricing/priceLists/securePriceLists.xml`	none

Notes:

- Users who have ATG Merchandising and ATG Search have an additional repository called `RefinementRepository`. There is no need to secure this repository, whose assets are facets.
- The XML files are in the `Versioned` module because they must replace their counterparts in the unversioned module.

Securing a Repository

You can secure standard repositories, as well as custom ones. If you modified your installation to use unsecured versions of the ATG 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. For a versioned repository, this is of class: `atg.adapter.secure.GenericSecuredMutableVersionContentRepository`\`

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

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

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 `<ATG10dir>/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.

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.
3. Update all views that display assets from the repository that you are securing. See Modifying the Views for Secured or Unsecured Repositories.
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 *ATG Business Control Center Security* chapter of the *ATG 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.

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 `<ATG10dir>/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 ATG repository views, see the *ATG Repository Guide*.

When you change the Product Catalog repository between secured and unsecured, the repository property and the views are updated, automatically; ATG 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, ATG 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 *ATG Repository Guide*.

For more information about secured repositories, see the chapter *Secured Repositories* in the *ATG 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 ATG Business Control Center applications, ATG Merchandising uses Access Control Lists (ACLs) to manage asset security and determine the operations that a given user can perform on an asset. You set access rights for ATG 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 *ATG 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 ATG Merchandising and suggests best practices for securing assets. For information about setting access rights, see *ATG Business Control Center Security* in the *ATG Business Control Center Administration and Development Guide*.

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 at asset's access rights, ATG Merchandising checks only the subtype's ACL, not the parent item type's ACL. To ensure ATG 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 ATG 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 `<ATG10dir>\Publishing\base\config\atg\epub\file`. For information on how to modify a repository definition file, refer to the *ATG 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.
Delete	Allows a user to remove this item from the repository. In ATG 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, an ATG 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 merchandiser does not have Read access to Products, only Catalogs and Categories appear in the hierarchy; the merchandiser cannot navigate to Products.

This behavior alsp applies to the Browse panel in th easset 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 alsp 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 ATG 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 ATG Merchandising, see *Creating and Managing* assets in the *ATG 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 to which the asset is 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 ATG Merchandising, see *Creating and Managing* assets in the *ATG 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 ATG Merchandising, see *Creating and Managing* assets in the *ATG 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 beingduplicated.
- Write access to the destination parent asset and its item descriptor.

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

Deleting Assets

In order to delete an existing asset in ATG 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 Unlink From All.

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

Working with Assets in the Collection Editor

For Collection properties such lists, maps, sets, and arrays, the user must have Read access to the item type in order to see the property in the ATG Merchandising Details pane. For example, a user who does not have Read access to media items will not see a product's auxiliary media property when viewing that product in ATG Merchandising.

If a user has Read access to both an asset and its collection property, the collection editor in the ATG 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 ten SKUs in its FixedChildSkus collection. A merchandiser does not need Read access to all ten 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.

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

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
```

Working with Projects

ATG Merchandising uses ATG Content Administration to manage and deploy versioned commerce assets. In order to use the ATG Content Administration mechanisms, ATG 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

ATG Merchandising includes the following project type selections on the BCC Operations list:

- Manage Commerce Assets

  A merchandiser uses this project type to create, add, modify, organize, and delete commerce assets.
- Administer Commerce Search (ATG Search is required)

  An administrator uses this project type to configure Search Merchandising, including creating the search configuration tree and adding properties for property prioritization.

When you are logged in to the BCC as a merchandiser, your ATG 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
- Search 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 *ATG 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 *ATG Merchandising Guide for Business Users* for information about creating merchandising assets.

Workflows

An ATG 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 *ATG Content Administration Programming Guide*. The chapter *Setting Up Deployment* in the same manual contains instructions to define the deployment topology.

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 *ATG Content Administration Programming Guide*.

Customizing Workflows

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

```
<ATG10dir>/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 e-mail 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.
- *Dafter*

Important: For information about the workflows that are included with ATG Content Administration, see the *Project Workflows* chapter of the *ATG 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 *ATG Personalization Guide for Business Users*
- *Creating and Configuring Workflows* in the *ATG 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 *ATG Business Control Center Administration and Development Guide*.

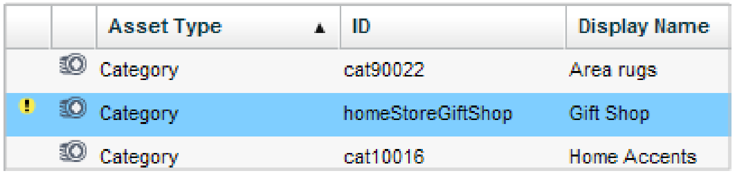
Customizing Asset Version Conflict Resolution for Collection Properties

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.

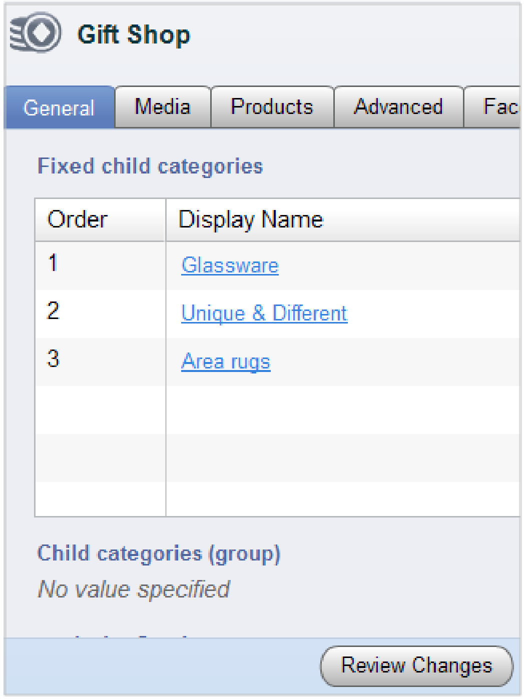
ATG Merchandising notifies you about an asset version conflict when you try to act on a project task. When the conflict occurs between assets that are collection properties (lists, maps, sets, or arrays), ATG Merchandising suggests a resolution that combines the changes.

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



		Asset Type ▲	ID	Display Name
		Category	cat90022	Area rugs
!		Category	homeStoreGiftShop	Gift Shop
		Category	cat10016	Home Accents

The areas of conflict are displayed.



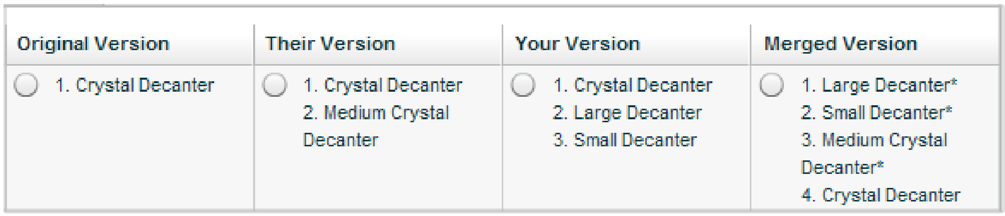
Fixed child categories

Order	Display Name
1	Glassware
2	Unique & Different
3	Area rugs

Child categories (group)

No value specified

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



Original Version	Their Version	Your Version	Merged Version
○ 1. Crystal Decanter	○ 1. Crystal Decanter 2. Medium Crystal Decanter	○ 1. Crystal Decanter 2. Large Decanter 3. Small Decanter	○ 1. Large Decanter* 2. Small Decanter* 3. Medium Crystal Decanter* 4. Crystal Decanter

On a page that displays conflicts for a collection, ATG Merchandising displays only the first 2300 items in a collection, by default. Therefore, if either version of the collection has more than 2300 items, ATG 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:

- Setting the Maximum Number of Collection Items to Combine
- Setting the Display Maximum for Collection Properties during Conflict Resolution

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

Setting the Maximum Number of Collection Items to Combine

You can override the 2300-item limit for which ATG 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.

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, `<ATG10dir>/home/localconfig`.

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

- To *change* the limit for which ATG 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 ATG 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 *ATG Programming Guide*.

Setting the Display Maximum for Collection Properties during Conflict Resolution

When a collection properties version conflict exists, ATG Merchandising can produce a table that includes a suggested merged version, as shown in Customizing Asset Version Conflict Resolution for Collection Properties. By default, ATG 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.
- 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 ATG 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 ATG Merchandising server to cause the changes to take effect.

For more information about editing view mappings, see the *Customizing Asset Display* chapter of the *ATG 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 ATG 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: <ATG10dir>/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 *ATG Programming Guide*.

Configuring Deployment Resources

Deployment is the process of copying assets from the ATG 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 ATG 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 *ATG 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 *ATG 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 *ATG Content Administration Programming Guide*.
4. Add any versioned repositories that are not already there, including custom repositories and the `RefinementRepository` that is used by Search Merchandising and Faceted Search. Follow the instructions in the *Setting Up Deployment* chapter of the *ATG 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 ATG Merchandising

Because ATG Merchandising is installed with the ATG platform, you cannot uninstall or remove it separately.

To stop the ATG Merchandising application, stop the application instance on the application server. The instructions for stopping are specific to the server. The chapter *Running Nucleus-Based Applications* in the *ATG Installation and Configuration Guide* contains server-specific instructions.

4 Configuring Price Lists in ATG Merchandising

ATG Commerce supports two types of pricing for Web sites:

- 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. For example, in 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.

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

You manage price lists in the ATG Merchandising Price Lists view.

Before You Begin

Before you can work with price lists in ATG Merchandising, you must configure pricing resources to implement, display, and calculate order prices. The *ATG Commerce Programming Guide* includes several chapters that describe how to configure pricing in ATG Commerce. 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 how you can extend them.

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

Configuring Price List Access

By default, ATG Merchandising assigns user roles different levels of access to price lists. For example, by default, users with the EPub-Admin role can create price list folders but users with the EPub-User role cannot. You can change access rights for individual price lists through the Security tab in the ATG 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 *ATG Business Control Center Security* in the *ATG Business Control Center Administration and Development Guide* to learn about configuring asset security.

To set access rights for a price list:

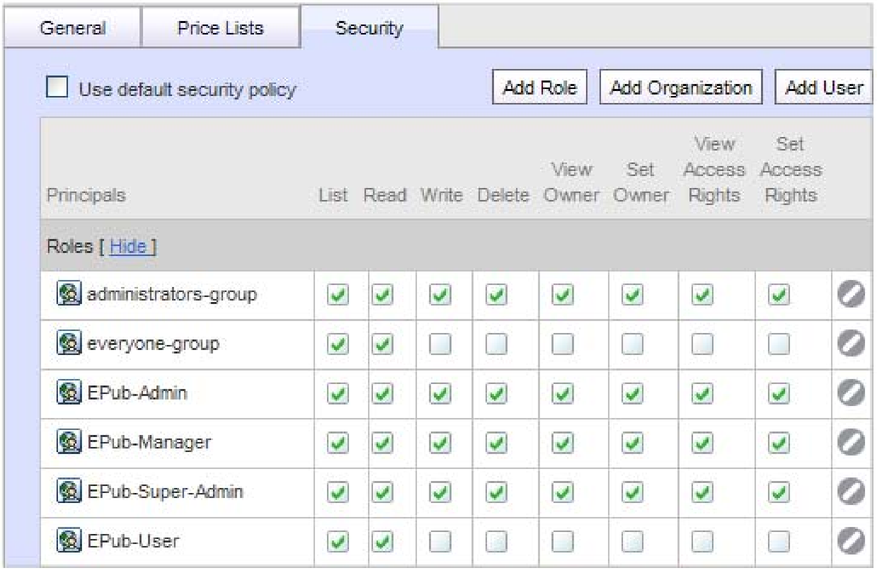
1. In a Managing Commerce Assets project, navigate to Price Lists.
2. Click the price list whose access rights you want to change.
3. Click the Security tab to display the default access rights for this price list.

   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 button enabled:



☐ Use default security policy

Principals	List	Read	Write	Delete	View Owner	Set Owner	View Access Rights	Set Access Rights	
Roles [Hide]									
administrators-group	☑	☑	☑	☑	☑	☑	☑	☑	⊘
everyone-group	☑	☑	☐	☐	☐	☐	☐	☐	⊘
EPub-Admin	☑	☑	☑	☑	☑	☑	☑	☑	⊘
EPub-Manager	☑	☑	☑	☑	☑	☑	☑	☑	⊘
EPub-Super-Admin	☑	☑	☑	☑	☑	☑	☑	☑	⊘
EPub-User	☑	☑	☐	☐	☐	☐	☐	☐	⊘

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.

Price Lists in a Multisite Environment

Typically, most multisite environments use price lists, not SKU-based pricing, 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 on different sites for the same SKU.

5 Administering Faceted Search

In an environment that includes both ATG Merchandising and ATG Search, the Faceted Search feature enables ATG Commerce sites to provide a navigational structure that is not strictly based on the catalog hierarchy. Facets are like virtual categories that are populated by the results of search queries. Facets are implemented as search refinements, which are used to narrow down search results by searching within those results for only the items that fulfill a certain constraint. For example, a search might return men's shirts, and then the customer might select a facet value that narrows the results to men's shirts that are made of cotton.

This chapter describes configurations that enable merchandisers to create and edit facets in ATG Merchandising:

Configuring Facet Inheritance for Categories

Categories in a catalog can inherit global, catalog, or category facets that belong to parent assets. Merchandisers determine which facets a category inherits by setting the category's inheritance options in ATG Merchandising. By default, merchandisers can set specific inheritance options for a category even if it has no local facets of its own. You can change this behavior so that any catalog or category inheritance settings are ignored unless the category contains at least one category facet. Global facets are not affected by this setting.

To configure the facet inheritance settings for categories without their own local facets:

1. In your local configuration, create a properties file for the following component:

   ```
   /atg/commerce/search/refinement/
   RefinementConfigurationCollectorFactory
   ```

2. Set the `propagateFacetInheritanceFlagsValues` property to `true` or `false`:

   `false` (default): The local inheritance settings of a category are always used, even if it has no local facets.

   `true`: A category that has no category facets defined has the same facet inheritance settings as its parent. Any local inheritance settings are ignored.

If a category has at least one local facet defined, the `propagateFacetInheritanceFlagsValues` property has no affect on the facet inheritance behavior.

Facet Inheritance in a Multisite Environment

Facet inheritance settings control how a catalog or category inherits facets from a parent:

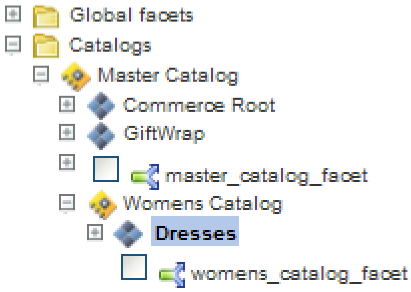
- Global facets can be inherited by all catalogs or sets of linked catalogs.
- Catalog facets can be inherited by a catalog's categories and subcatalogs.
- Category facets can be inherited by a category's subcategories.

Catalog facet inheritance behaves differently in a multisite environment than it does in a non-multisite environment:

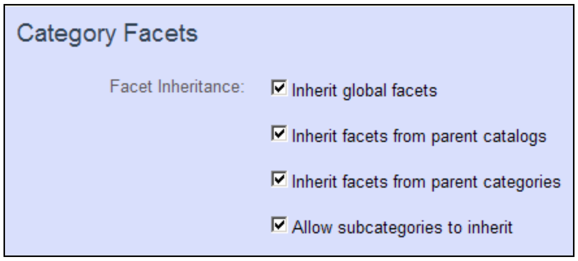
- In a non-multisite environment, a subcatalog cannot inherit catalog facets from a parent catalog if the merchandising user's profile contains only the subcatalog. The user profile must also contain the parent catalog in order for facets from the parent catalog to be available for that user.
- In a multisite environment, a subcatalog can inherit catalog facets from parent catalogs, regardless of whether the parents are in the merchandising user's profile.

  Note: This behavior does not apply to Category facets. A subcatalog does not inherit category facets from categories in a parent catalog if the parent is not in the user's profile.

For example, the following illustration shows a catalog hierarchy that contains a root catalog called Master Catalog and its subcatalog, Womens Clothing. The Master Catalog contains a catalog facet, `master_catalog_facet`. The Womens Clothing catalog contains a catalog facet, `womens_catalog_facet`.



The Womens Catalog is associated with a merchandiser's profile, but the Master Catalog is not. The merchandiser navigates to the Dresses category and selects Inherit facets from parent categories.



Category Facets

Facet Inheritance:
☑ Inherit global facets
☑ Inherit facets from parent catalogs
☑ Inherit facets from parent categories
☑ Allow subcategories to inherit

In a multisite environment, the `womens_catalog_facet` appears in the list of inherited facets. In a non-multisite environment, it does not.

For information about setting facet inheritance, see the *ATG Merchandising Business User Guide*. For information about setting up internal user profiles, see the *ATG Business Control Center Administration and Development Guide*.

Facet Labels

You have the following options for using the facet's Name property to set the facet label on your Web sites:

- Use a resource bundle to map the facet's Name property to a label in a properties file. This method supports localization.

  For instructions, see *Changing UI Labels* in *ATG Business Control Center Administration and Development Guide*.

- Use the facet name as the label that will appear on the Web site.

  If you do not specify a resource bundle, the Name property of the facet is used as the label on your production Web site, unless coded differently in the page. For instructions on how to edit the Name property, see Defining Facets in the ATG Merchandising Business User Guide.

6 Configuring Search Merchandising

The Search Merchandising feature enables ATG Commerce sites to customize search results based on site initiatives and customer purchasing patterns. A merchandiser uses ATG Merchandising to create *search configurations*, which are sets of rules that determine the order of search results and which items are excluded from the results.

Search configurations are stored in the `RefinementRepository` (along with the refinement configurations used for Faceted Search), and are included by ATG Search when the product catalog is indexed. When a site visitor issues a search query, ATG platform components determine which search configuration to use, and include this information with the query. When ATG Search returns the results from the query, it applies the rules in the search configuration to those results.

This chapter describes how to configure and administer search configurations. It includes the following sections:

Designing a Search Configuration Strategy

Creating a Search Merchandising strategy involves creating and configuring several assets. There are few restrictions placed on the order in which you perform tasks, but you may find it easier to approach them in this order.

First, an administrator performs the following tasks:

1. Determines the structure for the search configuration tree, which involves deciding how you want to organize your search configurations and search configuration folders, the dimensions and dimension values held by each, and the base search configurations you want to create. The section Structuring the Search Configuration Tree, contains the information you need to make those decisions.
2. Creates the search configuration folders, sets up the dimension language, segment, and site values, and selects the properties that can affect result ordering using property prioritization. These tasks are described in the Administering Search Configurations section.

Once these tasks are complete, a merchandiser does the following tasks. These tasks are described in *Defining Search Configurations* in the *ATG Merchandising Guide for Business Users*.

1. Creates base search configurations.
2. Specifies the rules and settings used by the base search configurations.
3. Creates search configurations.
4. Specifies the rules and settings used by the search configurations.
5. (Optional) Runs search tests that list the results that would appear on the Web site if that search configuration were deployed.

Structuring the Search Configuration Tree

When a visitor enters a search term on a Web site that uses Search Merchandising, the software determines which search configuration to apply, and includes this information in the query it sends to ATG Search. The logic used to select the search configuration is based on the tree structure you create in ATG Merchandising. This structure can take into account three dimensions:

- Language: Lets you target search results by language. This is useful if your site is in multiple languages or has visitors from different countries
- User segment: Lets you target search results to a group of people in your target audience. You create segments in ATG Targeting and Segmentation by defining rules that include or exclude users according to their profile properties.
- Site: In a multisite environment, lets you target specific search results to visitors searching from specific registered sites.

The dimension tree structure is stored as repository items in the `RefinementRepository`. When the product catalog is deployed to the target site, this repository is deployed as well.

The dimension services that create the tree structure in ATG Merchandising, `/atg/search/config/LanguageDimensionService`, `/atg/commerce/search/config/SiteDimensionService`, and `/atg/commerce/search/config/SegmentDimensionService`, are also present on the ATG Search client environment. When a site visitor submits a query entered in a search form, the form handler invokes the `/atg/commerce/search/config/SearchConfigNameService` component, which uses these services to traverse the dimension tree based on the visitor's language (locale), the user segments he or she is a member of, and the current site. It proceeds through the tree until it finds the first search configuration that matches these values. It then adds a reference to this search configuration to the query. The search configuration rules are applied to the results returned. If there is no matching search configuration, then no search configuration rules are applied.

Tree Organization

A tree is made up of search configurations organized in search configuration folders. When you create search configurations, it's a good idea to first create base search configurations that act as templates by providing settings that can be used or overridden in the search configurations they are associated with. A search configuration can associate any base search configuration with itself as long as it doesn't support a conflicting language. For more information about languages, see Configuring Language Support.

Depending on how you configure your tree, you can create a single search configuration for all Web site visitors, or create different search configurations for different groups of visitors. Visitors can be grouped by one or more of the following dimensions:

- Language: useful if your site is in multiple languages or has visitors from multiple countries
- User segment: useful for targeting specific search results to specific types of visitors
- Site: In a multisite environment, the site dimension lets you target specific search results to visitors of specific sites.

Before merchandisers can create search configurations and base search configurations, an administrator must create the search configuration folder structure that defines the tree and configure the languages, user segments, and sites that will be available as dimension values.

Types of Search Configuration Trees

You can create the following types of search configuration trees:

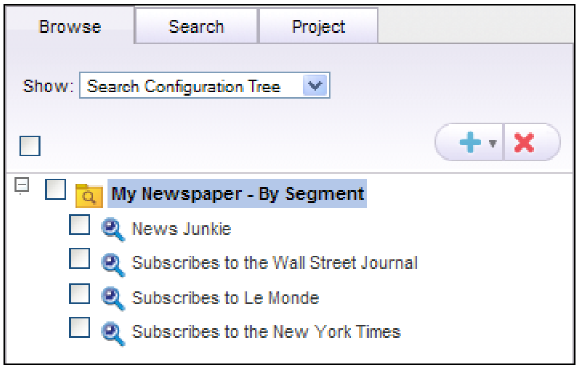
- One Search Configuration for All Site Visitors
- One Search Configuration Folder Containing One or More Search Configurations
- Multiple Search Configuration Folders and Search Configurations

One Search Configuration for All Site Visitors

You can create a tree that contains a single search configuration at the top level. This search configuration doesn't use a dimension because its settings apply to all site visitors. Use this approach if you want to control the display of search results in the same way for all visitors.

One Search Configuration Folder Containing One or More Search Configurations

You can create a tree that contains a single search configuration folder at the top level. One or more search configurations are created inside this folder. If you want to limit the visitors to whom a search configuration applies by one type of dimension, you design a tree using this strategy.



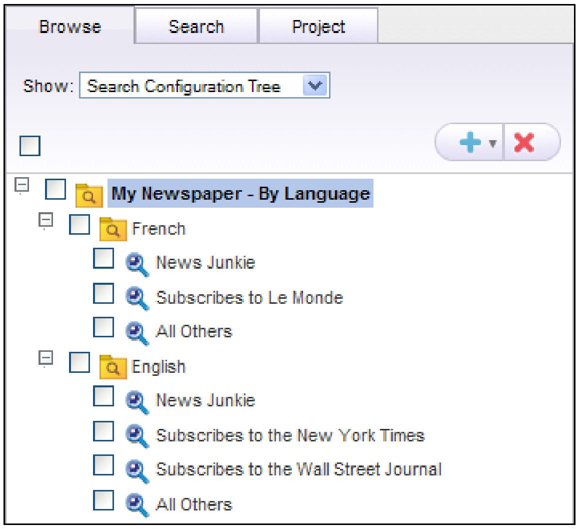
Show: Search Configuration Tree

- My Newspaper - By Segment
  - News Junkie
  - Subscribes to the Wall Street Journal
  - Subscribes to Le Monde
  - Subscribes to the New York Times

In this example, My Newspaper - By Segment is a search configuration folder, using segment as its dimension type. Each configuration it contains specifies a different segment. In order to see the results produced from these search configurations, a visitor must belong to the specified segment and use the default language.

Multiple Search Configuration Folders and Search Configurations

One search configuration folder opens to one or two levels of search configurations and search configuration folders:



Show: Search Configuration Tree

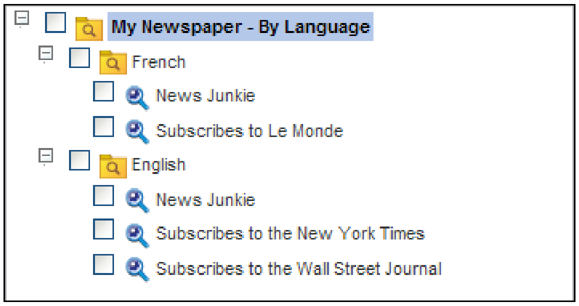
- My Newspaper - By Language
  - French
    - News Junkie
    - Subscribes to Le Monde
    - All Others
  - English
    - News Junkie
    - Subscribes to the New York Times
    - Subscribes to the Wall Street Journal
    - All Others

In this example, My Newspaper – By Language is a folder that uses language as its dimension type. Both of the folders it contains (English, French) specify a particular language as their dimension value. They must also supply a dimension type, which determines the type of dimension values their search configurations must provide, not the type of dimension value they provide. For example, the English search configuration folder has a dimension value of English and a dimension type of segment.

Using a multi-tiered hierarchy permits a greater degree of specificity. One search configuration is used for French people who read a variety of newspapers (New Junkie), for example, and another for similar

English speaking visitors. Because this example uses two types of dimension folders, the tree can incorporate, at most, three levels: one top-level search configuration folder using one dimension type, a nested search configuration folder using the other dimension type, and nested search configurations.

When you plan a multi-tiered tree, you need to decide which dimension type you want to begin with. Whether you begin with language, segment, or site has no impact on how search configurations are applied. There's also no implementation difference if one dimension is applied to search configurations and the other applied to search configuration folders, so you should organize your dimensions in the way that creates less setup work for you. For example, consider this tree structure:

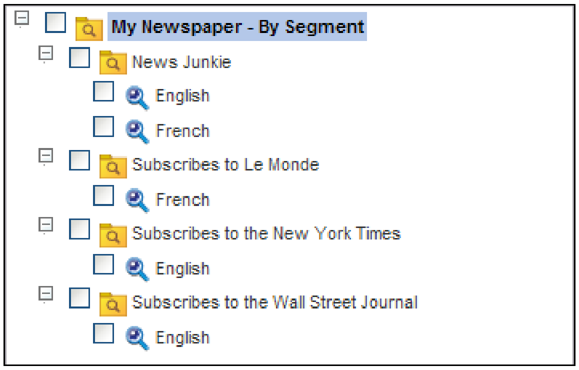


```
⊟ □ My Newspaper - By Language
  ⊟ □ French
      □ News Junkie
      □ Subscribes to Le Monde
  ⊟ □ English
      □ News Junkie
      □ Subscribes to the New York Times
      □ Subscribes to the Wall Street Journal
```

This hierarchy requires you to create the fewest assets for two reasons:

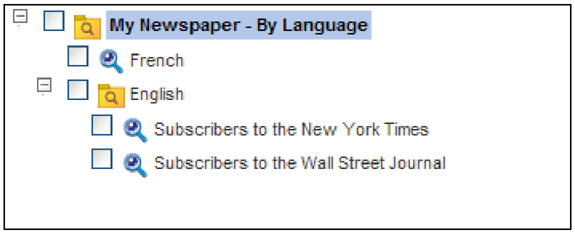
- There are two languages represented and four segments. Starting with the type of dimension that has fewer values ensures that you have fewer to create.
- Of the four segments, one groups visitors of both languages. The other three apply to one language grouping each.

If you were to organize your tree using segments first, as shown below, it would include ten assets instead of eight:



```
⊟ □ My Newspaper - By Segment
  ⊟ □ News Junkie
      □ English
      □ French
  ⊟ □ Subscribes to Le Monde
      □ French
  ⊟ □ Subscribes to the New York Times
      □ English
  ⊟ □ Subscribes to the Wall Street Journal
      □ English
```

Consider the hierarchy you could use if the News Junkie segment didn't exist. All segments would apply to either English or French. Because all visitors who are part of Subscribes to Le Monde are also French speakers, you could create a search configuration to represent French, thereby removing one level in the hierarchy.


⊟ □ My Newspaper - By Language
□ French
⊟ □ English
□ Subscribers to the New York Times
□ Subscribers to the Wall Street Journal

Ordering Assets in the Tree

Assets at the same level in the tree are applied in the order they appear in the tree. For site visitors who fall into two segments, the first search configuration that appears in the tree is used.

For example, consider an English visitor who is part of the Subscribes to the New York Times segment. ATG Search locates the correct search configuration by beginning at the top of the tree with the French configuration folder. Because French is not the visitor's language, ATG Search skips immediately to English for which the visitor is a match. ATG Search then checks the search configurations in that branch, stopping at Subscribes to the New York Times because the visitor satisfies all of its requirements.

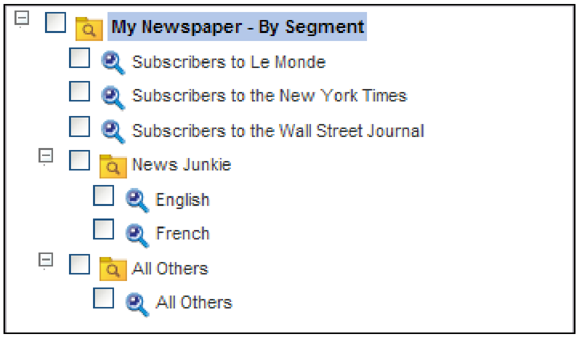
Likewise, if a site organizes search configurations by language, and one dimension value is a superset of another (en and en_US, for example), placing the search configuration that uses en higher in the tree prevents the en_US search configuration from being used because all site visitors who use en_US also use en.

Had the visitor used any language other than the two identified, the All Others search configuration folder branch would have been traversed. All Others is a dimension value available to both search configurations and search configuration folders to represent visitors for whom none of the other search configurations apply. For example, a visitor who has never identified newspaper preferences on a user profile page or has no recorded purchasing history uses the search configuration with the All Others dimension value.

The ordering principle applies to the All Other dimension as well, so putting a search configuration with All Others at the top of the tree would cause it to be used for every site visitor, thereby negating the other search configurations. For that reason, the search configuration or search configuration folder with the All Others value always appears at the bottom of the tree within a given level.

When Search Configurations and Search Configuration Folders are on the Same Level

The following hierarchy has search configurations and search configuration folders on the same level in the tree:


My Newspaper - By Segment
Subscribers to Le Monde
Subscribers to the New York Times
Subscribers to the Wall Street Journal
News Junkie
English
French
All Others
All Others

In this example, Subscribers to Le Monde, Subscribes to the New York Times, and Subscribes to the Wall Street Journal are search configurations because the same settings should be applied to all visitors in these groups regardless of language. Search configurations that are on the same level as search configuration folders always appear at the top of the list, which causes them to be considered first. A visitor who is part of both the Subscribes to the New York Times and News Junkie (English) segments sees search results that coincide with the Subscribes to the New York Times search configuration because it comes first in the list. The only exception is for a search configuration that has a dimension value of All Others, which will always be considered last.

Notice that there is no search configuration on the same level as a folder with dimension value of All Others. ATG Merchandising prevents you from making this selection because such a search configuration would divert all visitors away from any search configuration or search configuration folder beneath it on the tree.

In short, the top of a given level begins with a group of search configurations followed by search configuration folders and if one has a dimension value of All Others, it appears last in the list.

Implicit Language Inheritance

When you specify language as a dimension type for a search configuration folder, the search configurations and sub-folders it contains must specify a language that ensures that only the site visitors for whom that language is appropriate access that resource.

Every search configuration uses language to restrict the site visitors to whom it applies. This is true even for those search configurations that do not have a language dimension configured.

The following process describes how ATG finds the language for each search configuration:

1. First, ATG checks the search configuration for a language specified as a dimension value for the search configuration itself or a parent search configuration folder.
2. If no language dimension is specified, ATG uses the language assigned to its base search configuration, if one exists. In order to associate a base search configuration with a search configuration, the configurations need not have identical languages (a base search configuration might accept any language, for example, or a search configuration might not have an explicit language), but they must not be conflicting.
3. If there is no associated base search configuration or if the base search configuration's language is set to Any (not language specific), ATG uses the default language specified

by your site administrator. If no default language is specified, the search engine's default language is used.

One consequence to having an implicit language for each search configuration is that users may define rules without realizing how the implicit language influences their outcome. For example, a positioning rule that moves a set of products to the top of the results when a user searches on "Size Medium" is applied only when the language is English, because this term won't locate any results using the French dictionary. If language isn't a dimension in the search configuration, users may not think to create rules with words for each language their site supports.

Creating Search Configuration Folders

A search configuration tree can have only one asset at the root. Unless your environment uses a single search configuration, that root asset must be a search configuration folder that holds other search configuration folders and/or search configurations.

To create a top-level search configuration folder:

1. In a Search Merchandising project, display the Search Configuration Tree and click the Project tab.
2. Click the Create New button and then click Search Configuration Folder in the .
3. In the Name text box, enter a descriptive a name for the folder. Choosing a name that represents the dimension associated with the folder helps distinguish it from others.
4. If this is not the top-level search configuration folder (which would also be the first search configuration folder you create), you can see a drop-down list named for the type of dimension held by this folder: Segment, Language, or Site. Select the appropriate dimension. The option All Others indicates that the search configurations in the search configuration folder are used only when a site visitor's language isn't represented by an existing search configuration or search configuration folder.

   Note: If you don't see an option in the list, it may be in use by another resource; a dimension value can be used just once in a given level. Additional languages and segments may be added by your site administrator.

   From the Contents Vary By drop-down list, select a dimension type. All items contained by this search configuration folder will need to provide unique values for the dimension type you select. Only one dimension type appears in the list if the folder you are creating is not the root folder.
5. Click the Create button.

   The search configuration folder appears in the Navigation pane tree.

Deleting Search Configuration Folders

When you delete a search configuration folder, ATG Merchandising also deletes all the other folders and search configurations it contains.

To delete a folder:

1. On the Browse tab, select the folder to delete.

2. Click the Delete button.
3. Click OK to confirm that you want to delete the folder.

Administering Search Configurations

An administrator must perform the following tasks before merchandisers can work with Search Merchandising.

1. Specify the properties available for property prioritization.
2. Configure language support.
3. Configure the segment dimension.
4. Configure the site dimension (multisite environment only).
5. Manage search configuration folders.

Most administration tasks described in this section must be preformed in ATG Merchandising by an administrator whose user account includes the `EPub-Admin` role or `EPub-Super-Admin` role. Configuring language support involves working with component property values, so it is the only task that must be performed in the ATG Control Center or a properties file.

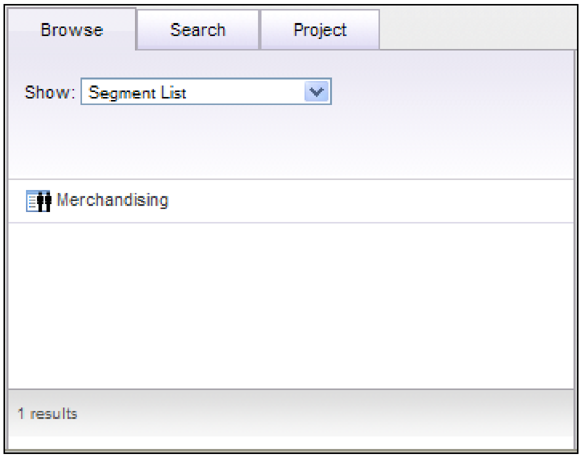
Search Merchandising Administrative Views

When you administer Search Merchandising using an Administer Commerce Search project, you can work with the Segment List view and the Property Prioritization view, which are not available in Manage Commerce Assets projects. Additionally, the Search Configurations Tree view and Base Search Configurations view provide more functionality to administrators in Administer Commerce Search projects than they do in Manage Commerce Assets projects.

Segment List View

The Segment List view displays the segment lists that you can configure for ATG Merchandising. You use the Merchandising segment list to designate the segments available for use as dimension values in search configurations and search configuration folders.

The following figure shows how the Merchandising segment list appears in Segment List view:



Merchandising

1 results

From this view, you can perform the following tasks:

- Select segments created in the Personalization module for use in ATG Merchandising.
- Rearrange the order in which segments will appear in dialog boxes to merchandisers.
- Remove segments from the list of segments available to ATG Merchandising.

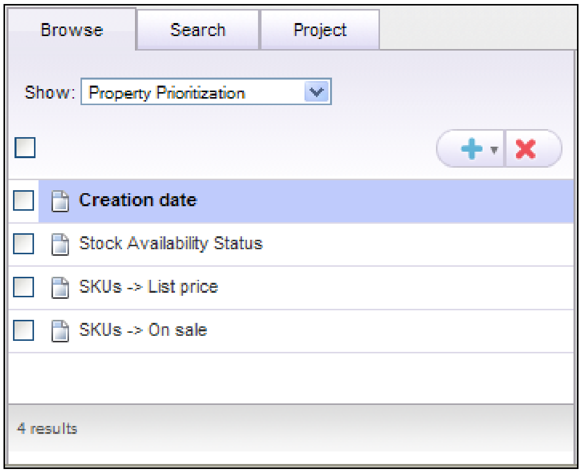
For instructions, see Configuring the Segment Dimension.

Note: You can also edit segment lists from the Personalization area of ATG Business Control Center. See *Configuring Segment Lists* in the *ATG Business Control Center Administration and Development Guide*.

Property Prioritization View

In Property Prioritization view, you work with the properties that merchandisers rank in search configurations and base search configurations.

The following figure shows an example of Property Prioritization view:



□ Creation date

□ Stock Availability Status

□ SKUs -> List price

□ SKUs -> On sale

4 results

From this view, you can perform the following tasks:

- Select product or SKU properties that can be used for ranking properties in result prioritization rules.
- Remove a property from property prioritization so that it no longer appears when creating result prioritization rules.

For instructions, see Managing the Properties Available for Property Prioritization. For more information about result prioritization rules, see *Defining Search Configurations* in the *ATG Merchandising Guide for Business Users*.

Search Tree Configuration View

In an Administer Commerce Search project, you can perform the following additional administrative tasks from the Search Tree Configuration view:

- Create and delete search configuration folders.
- Reorganize the search configuration tree.

For instructions, see Managing Search Configuration Folders and Moving Search Configurations and Search Configuration Folders.

Base Search Configurations View

In an Administer Commerce Search project, you can perform the following additional administrative tasks from the Base Search Configurations view:

- Create base search configurations.
- Define the settings in a base search configuration.
- Delete base search configurations

For instructions, see *Creating Base Search Configurations* in the *ATG Merchandising Guide for Business Users*.

Managing the Properties Available for Property Prioritization

The search result order is affected by rankings provided both to properties as well as to property values using property prioritization in the Result Prioritization tab. For example, a shopper who searches on a clothing brand will see a $300 jacket before a $10 t-shirt when the search configuration organizes search results by price (the property) and places higher priced items (property value) at the top of the list.

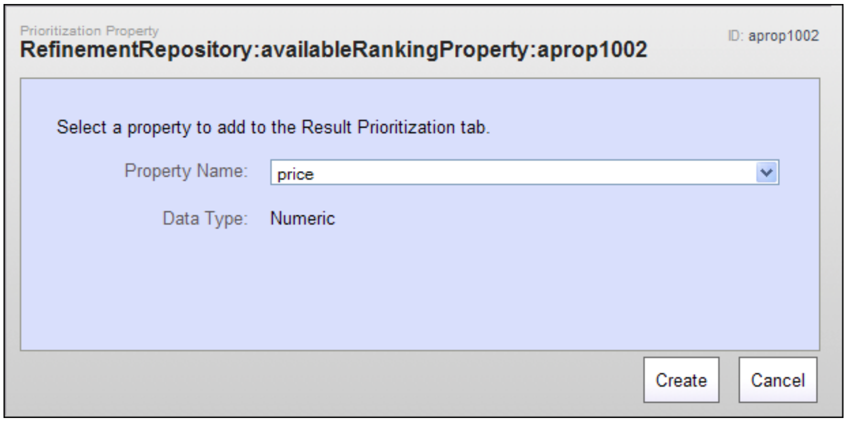
Administrators select the properties available for property prioritization from the list of properties indexed by the search engine. If products are indexed, product and SKU properties will be among those you must select. When SKUs are indexed, you can choose SKU properties only.

Selecting Properties

When you choose properties, keep in mind that both the properties and property values can be ranked. The properties you select are visible on all current and future search configurations.

1. While working in an Administer Commerce Search project, navigate to the Project tab or the Property Prioritization view of the Browse tab.
2. Click Create New > Property Prioritization.
3. From the Property Name list, select the property that you want to use in search configurations to prioritize search results.

   The Data Type text box displays the data type of the property.



Prioritization Property

RefinementRepository:availableRankingProperty:aprop1002

ID: aprop1002

Select a property to add to the Result Prioritization tab.

Property Name: price

Data Type: Numeric

4. Click Create.

   The property displays in the Navigation pane.

Removing Properties

When you remove a property, it is no longer visible in the property prioritization portion of the Result Prioritization tab, meaning it is removed from existing search configurations and unavailable to new search configurations. You can remove properties that are in use (the property has a weighting and/or its values have rankings in a search configuration). For information about removing properties, see *Defining Search Configurations* in the *ATG Merchandising Guide for Business Users*.

Configuring Language Support

When a merchandiser defines a search configuration in ATG Merchandising, he or she selects a language from a preconfigured list. Though the term “language” is used throughout ATG Merchandising, the values selected actually represent Java locales. So, for example, the list of available languages might include both British English (representing the locale `en_GB`) and US English (representing the locale `en_US`).

Each ATG Commerce site implements its own mechanism for finding a user’s locale. One site might select a locale from the HTTP header of the request. Another site might supply a Profile page in which the user specifies a locale that ATG saves as a user profile property. A third site might provide a Language Preference drop-down list on a home page, and when a user makes a selection, it’s appended to the request as a query parameter that is processed by a custom pipeline servlet. Most multi-language sites will be designed to incorporate a combination of these options, giving a priority to each.

When you organize configurations by language, a search configuration folder uses language as its dimension type. Folders and search configurations it contains must specify a particular language so that the search configuration applied depends on the language specified by a visitor’s locale.

Note: In order to have different search configurations apply to different languages, you must first create a search configuration folder and then create configurations in it. Otherwise, configurations will get applied to all users, and the language only decides what language processing to apply to all users.

If a search configuration inherits settings from a base configuration, the assets do not have to use the same language as the base configuration, but they must not have conflicting languages. There may be a one-to-one relationship between the languages supported in each, but that’s not frequently true. It is best to use general languages, or the Any (not language specific) option, on your base search configurations.

For example, suppose the tree begins with a Language search configuration folder that contains many search configurations, among them one for French speaking Canadians and another for French speaking Belgians. A base search configuration with a French language (`fr`) can be used by both search configurations (`fr_CA`, `fr_BE`).

To define the languages, you update component properties either by accessing the component in the ATG Control Center or by creating a new properties file for the component in your local configuration directory:

1. Access the resource you want to use to update the `/atg/search/config/LanguageDimensionService` component properties.
2. To specify languages for base configurations, set the `locales` property to the list of locales you want to support. For example:

   ```
   locales=en,fr
   ```

3. To specify languages for search configuration folders and search configurations, set the `languagesMap` property to a mapping of resource bundle key to locale. For example:

   ```
   languageMap+=\
   americanEnglish=en_US,\
   canadianEnglish=en_CA,\
   ```

```
britishEnglish=en_GB,\
canadianFrench=fr_ca,\
belgianfrench=fr_be,\
```

4. To set the default language for search configurations, specify a language or locale in the `defaultLocale` property.

   The default is used as the language for the All Others language dimension value when it does not inherit a language from a base configuration) or when the tree does not use language as a dimension.

   Note: If you do not specify a default language for search configurations, they automatically use the search engine default language.

Customizing Names for Languages

ATG Merchandising displays the names of the languages commonly associated to the locales you specify. For example, `en_US` and `en_BG` both display as English by default. You can provide other names for your locales by using a resource bundle.

Resource bundles hold labels for views, buttons, tabs, and other UI elements, so you can localize your JSP based ATG applications, like Search Merchandising. Most labels that you see in Search Merchandising are defined in the following resource bundle:

`<ATG10dir>/DCS-UI10.0.1/DCS-UI/management/lib/classes.jar/atg/commerce/web/WebAppResources.properties`.

For example, the names of the views visible in the Browse tab are specified there.

To provide other names for locales:

1. Create a resource bundle in `/atg/core/i18n/MultiBundleFormatter` of your local configuration directory, if you haven't already created a custom resource bundle there.
2. For the base configuration languages, use this format:

   ```
   atg.search.dimension.language.locale.<locale>=<display_name>
   ```

   For example:

   ```
   atg.search.dimension.language.locale.en=English (US, CA & GB)
   atg.search.dimension.language.locale.fr=French (CA & BE)
   ```

3. For the search configuration and search configuration folder languages, use this format:

   ```
   atg.search.dimension.LanguageDimensionService.<locale>=<display_name>
   ```

   For example:

   ```
   atg.search.dimension.language.americanEnglish=English (US)
   atg.search.dimension.language.canadianEnglish=English (CA)
   atg.search.dimension.language.britishEnglish=English (GB)
   ```

```
atg.search.dimension.language.canadianFrench=French (CA)
atg.search.dimension.language.belgianFrench=French (BE)
```

The resource bundle keys use a prefix (`atg.search.dimension`, by default) defined in the `dimensionValueKeyPrefix` property of the `LanguageDimensionService` component. If want to use a different prefix or no prefix in the resource bundle keys, change the prefix in both places.

Configuring the Segment Dimension

To use segments as a dimension, you need to create segments in the ATG Business Control Center as described in the *Creating User Segments* chapter of the *ATG Business Control Center Administration and Development Guide*. Then you need to make those segments available as options, by adding them to the Merchandising list, as follows:

1. While working in an Administer Commerce Search project, navigate to Segment List view of the Browse tab.
2. Click the Merchandising segment.
3. In the Details pane, update the list of segments:
   - Add a segment by clicking Add. When the Select an Item dialog box opens, navigate to the segment, click the check box beside it, and click OK. If you don't see the set of segments you expected, your site administrator should make sure that the correct modules are running.
   - Rearrange the position of a segment in the list by changing the number beside it. To preview the new order, click Reorder. The order you provide here doesn't affect how segments appear in other Merchandising lists.
   - Remove a segment from the list by clicking the Delete button beside the segment.
4. Click Save.

About Removing Segments

If you remove a segment from the Merchandising segment list in Search Merchandising, you make it unavailable for selection as the dimension for search configurations or search configuration folders. If you remove a segment that is already assigned to search configurations or search configuration folders, the segment remains in use only for those assets, but cannot be selected for new ones.

If, however, a segment is deleted from the ATG Business Control Center, that segment becomes unusable by all applications. Any search configuration or search configuration folder with a deleted segment can be deployed but will never be applied because, although the segment may appear in ATG Merchandising, it won't be visible to ATG Search. Because there won't be any visitor attributes associated with the resource that uses the deleted segment, that resource will never be applied.

Configuring the Current Site Value in Result Prioritization Ranking

In a multisite environment, the Site property is available for property prioritization. When a merchandiser selects the Site property for ranking, ATG Merchandising displays the property values of all registered sites available in the site repository, along with the values `[Current Site]` and `All Others`.

The Current Site label is defined in the following resource bundle:

```
<ATG10dir>/DCS-UI10.0.1/DCS-UI/management/lib/classes.jar/atg/commerce/search/
SitePropertyProviderResources.properties
```

7 Configuring Search Testing

This chapter describes how to configure the components required for creating and running Search tests. This chapter contains the following information:

Search Testing Components

This section describes the components you configure for Search Testing. See the *ATG Search Query Guide* for more information about these components.

A Search test replicates a user-entered Web site query

For details about how ATG Search works, see Chapter 2 in the *ATG Search Administration Guide*.

1. The `SearchTestingFormHandler` receives the search input information and passes it to the `QueryFormHandler`, which, in turn, passes information to the respective request processor components:

   - Facet Search Tools uses facet information to determine the `refineConfig` for the search query as well as the facet selections that will constrain it, updating `QueryRequest` accordingly.

   - The Search Configuration Request Processor passes search configuration information to `QueryRequest`.

   All other inputs are passed directly to `QueryRequest`.

2. The `QueryFormHandler` sends the `QueryRequest` via Routing components to ATG Search.

   - If your environment defines and indexes multiple catalogs, all indexed catalogs are searched.

   - In a multisite environment, if your environment defines and indexes more than one site, all indexed sites are searched.

3. ATG Search generates a result set in the form of an XML document that it sends back through the Routing components, which translate the XML document into a Search Response object.

4. The `QueryFormHandler` receives the Search Response object and passes it to the following components for processing:

   - `FacetSearchTools` determines, from the Search Response object, the list of facet values and selections to display.
   - `SearchConfigurationRedirectProcessor` uses the Search Response object to determine whether the results activate redirection rules and if so, effectively disables those rules, so that the users are not redirected but instead view test results. In this circumstance, test results will indicate that redirection rules were activated.

5. `FacetSearchTools` and `SearchConfigurationRequestProcessor` save their results to the Search Response object via the `QueryFormHandler`.

6. The Search Testing UI receives the Search Response object and displays test results according to the settings the merchandiser specified when creating the test.

7. Saving a search test causes the `SearchTestingFormHandler` to create a `searchTest` item in the Search Testing repository.

SearchTestingFormHandler

Location: `/atg/search/web/assetmanager`

Class: `atg.commerce.search.web.assetmanager.SearchTestingFormHandler`

Collects input values from the Search Testing UI page. When the search results are returned from ATG Search, this form handler prepares them for display in the Search Testing user interface. See the *ATG Search Query Guide* for more information about form handlers.

QueryFormHandler for Search Testing

Location: `/atg/commerce/search/testing`

Class: `atg.search.formhandlers.QueryFormHandler`

The main form handler for submitting queries to ATG Search. Later, this form handler passes the search results it receives from ATG Search to processor components that convert them into a response that is applicable for the Search Testing UI. See the *ATG Search Query Guide* for more information about form handlers.

QueryRequest component

Location: `/atg/commerce/search/catalog`

Class: `atg.search.routing.command.search.QueryRequest`

Receives general test inputs and references to facets and search configurations (or the XML for the repository items) from request processors and the Search Testing Form Handler.

Search Configuration Request Processor

Location: `/atg/commerce/search/testing`

Class: `atg.search.query.config.SearchConfigurationRequestProcessor`

Passes the search configuration associated with a given request to the query request.

Search Testing Service

Location: `/atg/search/testing`

Class: `atg.repository.search.testing.SearchTestingService`

Provides options to the Testing Environments, Search Text Language, and Sorting By input options in the Search Testing UI.

Production Environment and Staging Environment

Location: `/atg/search/testing`

Class: `atg.repository.search.testing.SearchTestingEnvironment`

Represents a search index available for testing and provides information about that search index, such as name and language, to the Search Testing Service, which makes it visible in the Search Testing UI.

Updating Components with Production Values

Search Testing replicates search conditions on your production site, so you must configure Search Testing components with the same property values as their production counterparts. Usually, you can copy property values from the production component to the corresponding Search Testing component. If your production site is coded to overwrite some properties in components with dynamic values provided by site JSP pages, when you are copying property values, be sure to check your JSPs for tags that update values. If you find any, you'll need to update the Search Testing component with the static value that is most often passed to the JSP.

Although you need to make updates to some component properties, a number of properties are provided for you and in fact, if you change them, Search Testing will overwrite your changes. Some components, such as form handlers, automatically derive property values from the production instance of that component, so although it's vital that the values of these two components are identical, they will be by default.

Note: You may add additional instances of Facet Search Tools and Search Configuration Request Processor, but you should not overwrite the instances defined in Search Testing.

If you have created or updated any of the following components on your production site, create or update the corresponding Search Testing components. Most resources will have the same name and location in both instances, though any differences are noted in the table below:

Resource	Property
Component: MultisiteConstraint Class: `search.routing.command.search.MultisiteSetConstraint`	all properties
Component: `DocumentSetConstraint` Class: `atg.search.routing.command.search.DocumentSetConstraint`	all properties
Component: `ParserOptions` Class: `atg.search.routing.command.search.ParserOptions`	all properties
Component: `/atg/commerce/search/catalog/QueryRequest` `Class: atg.search.routing.command.search.QueryRequest`	`documentSetConstraints`, `parserOptions`, and `relQuestSettings` (subproperty `activeSolutionZones` in particular)
Search Testing Instance: `/atg/commerce/search/testing/SearchTestingQueryFormHandler` Production Instance: a component of class `atg.search.routing.command.search.queryFormHandler`, which is most likely `/atg/commerce/search/catalog/QueryFormHandler`	`searchRequestProcessors`

Configuring Input Options

This section describes how to work with settings that control search test inputs.

Setting the categoryAndSearchTextMode Property

By default, merchandisers can specify both search text and a category for the query in a Search test. You can configure Search Testing so that users can specify either search text or category inputs, but not both.

Find the `SearchTestingFormHandler` in the `/atg/search/web/assetmanager` directory and edit the `categoryAndSearchTextMode` property:

`Cooperative`.(default): Both search text and a category can be specified.

`Exclusive`: Search text or category but not both can be specified.

Select cooperative for Web sites with these types of search capabilities:

- Users navigate through selections implemented by facets.
- Users can navigate to a particular grouping and then search within it for a particular item. The grouping could be, for example, a selection that's dynamically generated by facets or a category that's part of the static catalog structure.
- Users can navigate to a particular grouping, and they can search across the catalog for a particular item, for example, a selection that's dynamically generated by facets or a category that's part of the static catalog structure.

Select exclusive for Web sites with these types of search capabilities:

- Users navigate through a static category structure only (no facets), and they can search across the site, but aren't able to constrain a search with a user-selected category. (Search Text only)
- Users navigate the site using only selections that are dynamically generated by facets. (Category only)

Setting up Sorting Options

Merchandisers can specify the order in which the products or SKUs appear in the results list. The only option provided out of the box, Score, uses relevance to place products or SKUs that have the highest relevancy score at the top of the list and consequently, those that have the lowest relevancy score at the bottom. You can, however, sort results by any meta-property that's defined for the type of asset, product, or SKU, that's been indexed. It may be beneficial to provide the same sorting mechanisms that your Web site provides, so merchandisers can replicate the experience site visitors have.

For example, to provide inventory status, price, and response language as sort options for a configuration that displays only SKUs, update Search Testing Service located in the `/atg/search/testing` directory by setting the `sortByPropertyNames` property to:

```
stockAvailabilityStatus,childSKUs.price,language
```

For each property that you add, the Search Testing UI provides two entries in the Sort By control: one for descending order and one for ascending order. Only one sort option can be applied at a time. Regardless of the number of sorting options, the Score is always the default selection.

Permitting Multi-Selectable Facet Selections

For users who have defined facets, after running a search test, a list of facets displays, accompanied by facet selections on the bottom of the inputs portion of the Search Testing screen. To see only the products and SKUs in a given facet selection, you run a secondary test with that selection checked. The default setting lets you specify one facet selection per test; however, you can configure ATG Merchandising to permit multiple selections.

If users create tests to troubleshoot their Web site’s behavior, whether you want to use single or multiple selection depends largely on how the site implements facets. The single-select option replicates sites that use faceting:

- As a navigational tool that permits users to click one facet before displaying the products and SKUs it contains. For example, clicking the Women faceting selection displays all clothes for females.
- As a searching tool, so that when users search for a given term, the results are facet selections. Clicking a facet selection displays the products and SKUs it contains. For example, searching for “shoes” returns two selections: Men and Women. Clicking Men displays individual shoes suited for men.

The multi-select option represents sites with inherited and/or nested facets that employ faceting:

- As a navigation tool by permitting users to click successive selections to filter out certain results. For example, clicking the Women selection causes the selections Clothes, Shoes and Accessories to display. In turn, clicking Shoes returns another group of selections, some representing prices (up to $100, $100-200, $200- 300) others representing sizes (5, 5.5, 6. 6.5), and a third for sale items. These selections hold products.
- As a searching tool where users can use faceting selections as search criteria. For example, a user might define criteria to find all shoes from $100-200 in size 7 that include the search text “high heels.”

When ATG Merchandising is configured to use multi-select, you are able to specify multiple selections for a given facet (up to $100 and $100-200) as well as specify selections derived from different facets ($100-200 and size 7).

To change your setting from single-select to multi-select, find the `SearchTestingQueryFormHandler` in the `/atg/commerce/search/testing` directory and change the `facetInputMode` property from `single` to `multiple`. If the property is empty, `multiple` is used.

Specifying a Default Price List

If your Web site uses price lists and you want Search test results to include pricing information, you must specify a default price list for Search Testing. A default price list is necessary if you configured the Sort by option to order test results by price or customized the Search Testing page to include price as a column in the results table.

Typically ATG uses the active profile to determine the price list to apply. In the absence of an external profile, the default price list is applied for all tests, which may cause the pricing to be inaccurate for some users if your site uses multiple price lists.

Note: If you use SKU Pricing instead of price lists, set up sorting as you would with any other property.

To specify a default price list:

1. Find the ID for the default price list.

   A price list ID begins with `plist` and it is located at the top of the Details pane when a price list is selected. For information about locating price lists, see Configuring Pricing in ATG Merchandising.

2. Create a properties file for the Price List Property Mapping component with the `defaultPriceListId` set to a specific price list ID and save it to `/atg/commerce/search`:

   ```
   $class=atg.commerce.search.PriceListPropertyMapping
   $scope=global

   defaultPriceListId=plist90002
   ```

   If you prefer to have a global default price list available to all modules, create a component of class `atg.commerce.pricing.pricelists.PriceListManager` and set the `defaultPriceListId` property to the price list ID that will act as the default.

Specifying Catalog Constraints

The `/atg/commerce/search/testing/FacetSearchTools` component manages catalog constraints for all search tests.

By default, when a merchandising user runs a search test, `FacetSearchTools` checks the user's profile to see if he or she is assigned a catalog. If so, `FacetSearchTools` constrains the test to include only items in this catalog.

This behavior may not be desirable in all situations. In a multisite environment that has a separate catalog for each site, this logic may interfere with cross-site searches, because the search will be constrained to a single catalog (and thus a single site). To prevent applying a constraint based on the catalog assigned to the user, set the `queryByCatalog` property of the `FacetSearchTools` component to `false`.

If `queryByCatalog` is `false`, you can still have `FacetSearchTools` constrain queries to specific catalogs by setting its `catalogIds` property to an array of the catalog IDs of the catalogs to include in the search. For example:

```
catalogIds=masterCatalog,homeStoreCatalog
```

Note that if `catalogIds` is not null, the value of `queryByCatalog` is ignored. In this case, even if `queryByCatalog` is `true`, the catalog assigned to the user is ignored and the catalogs in the `catalogIds` property are used for the catalog constraint.

Tip: The catalog ID appears at the top of the Details pane when you select a catalog in ATG Merchandising.

Determine Locale When Using Preview Profiles in Search Testing

When you use a preview profile to test the search configuration tree, Search Testing takes the segment and language dimension values from the preview profile. However, some sites use profiles only to determine the user's segment and not the user's locale. If you want to prevent Search Testing from using the `Locale` property of the preview profile, or you want to use a different profile property for determining the language dimension, you can change the value of the `testProfileLocaleProperty` property of the `LanguageDimensionService` component.

Note: If you choose not to use the preview profile locale (you clear the `testProfileLocaleProperty` property) but your search configuration tree includes language as a dimension, then the All Others language dimension is always used when testing with a preview profile.

To use only the segment dimension of a preview profile when testing the search configuration tree:

1. In your local configuration, create a properties file for the following component:
   `/atg/search/config/LanguageDimensionService`
2. Set the `testProfileLocaleProperty` to null, as follows:
   `testProfileLocaleProperty^=/Constants.null`

To use a different preview profile property to determine the language dimension:

1. In your local configuration, create a properties file for the following component:
   `/atg/search/config/LanguageDimensionService`
2. Set the `testProfileLocaleProperty` to the name of the property you want to use.

   The default setting is:
   `testProfileLocaleProperty=Locale`

Using Grouping

Commerce sites may use the grouping-by-product feature to collapse all SKU results associated with a given product into a single result. In order to use grouping, you indicate in the configuration of your Search Index Output Configuration file that you want to index by SKU rather than index by product.

In order for grouped items to display the way you prefer in Search Testing, make the necessary configurations described in Displaying SKUs Grouped by Product.

By default, if you index by SKU, your items will be grouped. To view results as ungrouped, you need to disable grouping in Search Testing, as described in Disabling Grouping. Grouping is unavailable if you index by product.

Displaying SKUs Grouped by Product

Whether groupings display as a single product with information about one SKU or individual products in the Search Testing results list depends on the usage of the sub-item Results input. For more information about setting these properties, see the *ATG Commerce Search Guide*.

If your site displays groupings and you'd like to replicate your site behavior in a search test, follow these steps:

1. Locate the Query Request component by checking the `searchRequest` property of the `/atg/commerce/search/testing/SearchTestingQueryFormHandler` for the

name and location of the Query Request component. The default is `/atg/commerce/search/catalog/QueryRequest`.

2. Set the following Query Request properties:

   ```
   sorting=property
   sortProperty=string:$repositoryId:1
   ```

3. Configure the `responseNumberSettings` map property to return groupings in the way you prefer:

   - `prop` determines the maximum number groupings to return. Keep in mind that 10 results display in the first page of the result list.
   - `perProp` establishes the maximum number of SKUs per product.

   For example, the following settings return a maximum of 100 groupings each containing no more than 20 SKUs:

   ```
   QueryRequest.responseNumberSettings=prop=100,perProp=20
   ```

For more information on grouping and setting up grouping, see the *Standard Query Results* section of the *Standard Query* chapter in the *ATG Search Query Reference Guide*.

Disabling Grouping

If you index by SKU and don't want results grouped by product in search test results, you need to replace the following property values:

1. Update the following properties of the Search Testing UI Property Mapping component, located in `/atg/search/testing`:

   ```
   propertyMapping=itemId=childSKUs.$repositoryId,\
   itemDisplayName=role:childskus.displayname,\
   itemDescription=role:childskus.description,\
   viewItemDisplayName=childskus.displayname
   ```

2. Locate the Query Request component by checking the `searchRequest` property of the `/atg/commerce/search/testing/SearchTestingQueryFormHandler` for the name and location of the Query Request component. The default is `/atg/commerce/search/catalog/QueryRequest`.

3. To display only SKUs in the results list, set the following Query Request properties:

   ```
   sorting=document
   sortProp=""
   ```

4. Save your changes.

Setting up Search Environments

A Search environment designates a set of one or more machines as being responsible for one of the following tasks:

- Indexing your content. Each Search project has a single environment for indexing, which can be distributed across multiple hosts.
- Answering end-user queries using the deployed index. A project can have any number of environments for this purpose. These environments should be hosted on a machine optimized for high availability and performance.

Before you run Search tests, you must identify the Search environments that contain the products, SKUs, search configurations, facets, and other search settings you want to test.

Managing Environments in the Search Testing Service

The Search Testing Service specifies the testing environments available to the ATG Search Testing UI. Configure the following properties for the `SearchTestingService` component, located in the `/atg/search/testing` directory:

If your ATG Search Environment's Global Settings include content labels other than Catalog or multiple target types, set the `defaultContentLabel` and `defaultTargetType` properties to the components represent the content label and target type that should appear as preselected in the Search Testing UI. For example:

```
defaultContentLabel=Catalog
defaultTargetType=Production
```

For more information about content labels and target types, see *Before You Begin – Global Settings* in the *ATG Search Administration Guide*.

Specifying Response Languages

For multi-language catalogs, you can filter results by language, that is, display results that use a subset of the languages you support. The response language restricts the search results to products and SKUs in a specific language. For example, if the response language is English, and the search text is “dress,” your results could include a number of sundresses as well as formal gowns, all of which are described in English.

There are two ways to structure a localized catalog. You can define one asset per product/SKU with properties representing different languages, or several assets per product/SKU, each in a different language. Regardless of the structure you use, the process of indexing produces the same document output, that is, one document for each language-to-product/SKU combination.

When you specify response languages, you can include any number of languages as long as those languages are specified in the search index. Unsupported response languages are ignored by ATG Search. If all response languages that you define are unsupported, ATG Search discards the language filter, returning all results that match the input settings.

There are two places to set response languages, one for search text and another for category. When a test includes search text and a category, the search text language is used. It's a good idea to provide the same languages to both the category and search text response language settings. Search text response languages are applied to all tests that include search text, even those for which a category and a category response language have been provided. The category response language, in this case, is ignored.

Adding Search Text Response Languages

You can explicitly provide search text response languages, but if you don't, ATG Merchandising uses the language a user selects from the Search Text Language drop-down list, which means that you'll see only the products and SKUs that are indexed in the same language as the search text. A search for "robe" when the search text and search text response languages are English will return the silk and terrycloth coverings that one wears over pajamas.

Because you can specify one search text language at a time, if you want a test to find products and SKUs in multiple languages, you should specify multiple response languages. Consider a multi-language sporting equipment store. A search for "Specialized" reveals all Specialized bicycles sold by the store, which is helpful when each language catalog sells a unique set of SKUs.

Another reason to specify search response languages explicitly is if your site supports multiple languages and you'd like to search in one, but see results in another, perhaps more familiar language. For example, a French-speaking merchandiser, who wants to see "robes" from the English catalog, but doesn't know that in English they are called "dresses" could search for "robe" when the search text language is French and the response language is English to see dresses sold on the English-language site.

To specify response languages, you provide the languages to the Parser Options component as described in the following procedure. The procedure instructs you to create a properties file; however, if you prefer to use the ACC, select the corresponding components of class `atg.search.routing.utils.Language` instead of providing locales to `targetLanguages`.

1. Create a properties file for the Parser Options component with a `targetedLanguages` property set to a comma-separated list of languages or locales that represent the languages you want to support:

   ```
   $class=atg.search.routing.command.search.ParserOptions
   $scope=request

   targetLanguages=en_US,en_GB
   ```

2. Save the properties file to the local configuration directory as follows:

   - If you already have a Parser Options component, name this properties file for that component, and save it to the same location.
   - Otherwise, create a new component by naming the file anything you choose and saving to a directory, such as:

     ```
     <ATG9dir>/home/localconfig/atg/commerce/search
     ```

     If you've created your own Parser Options component, you also need to add it to the `parserOptions` property of the Query Request component, which you find by checking the `searchRequest` property of `/atg/commerce/search/testing/SearchTestingQueryFormHandler` for the name and location of the Query Request component.

For more information on the form handler class and Parser Options component, see the *Search Form Handlers* chapter of the *ATG Search Query Guide*.

Adding Category Response Languages

Category response languages are applied for tests that include a category as an input and not search text. Search Testing should support the exact same languages as your production site, if you want the complete set of results returned by the search engine to be visible in Search Testing. Note that it's unlikely a site visitor would see all of those results, because most sites are coded to filter out results that aren't in a site visitor's language. If you'd like for Search Testing to display what a given site visitor sees, or results in a subset of the supported languages, specify those languages directly. Otherwise, copy your response language settings from your production site to see results in all languages.

Viewing Results in All Languages

To display results in all languages supported by your production site:

1. Copy all components of the following classes that exist in the production environment:

   ```
   atg.search.routing.command.search.StringConstraint
   atg.search.routing.command.search.ConstraintsGroup
   atg.search.routing.command.search.ConstraintsGroup
   ```

2. Copy the value in the `documentSetConstraints` property of the production Query Request component to the corresponding component in `/atg/commerce/search/catalog`.

Viewing Results in a Subset of Languages

The way in which you provide category response languages that are a subset of those supported by the production environment depends on the number of languages you want to use.

To view results in a single language, follow the steps in this procedure, which specify support for English:

1. Create a properties file for a component of class `atg.search.routing.command.search.StringConstraint` as follows:

   ```
   $class=atg.search.routing.command.search.StringConstraint
   $scope=request

   property=language
   operation=equal
   value=English
   ```

2. Save the properties file in a Search Testing directory in your local configuration layer, such as `<ATG9dir>/home/localconfig/atg/commerce/search`.

3. Register the new component with the Query Request component in `/atg/commerce/search/catalog` by setting the `documentSetConstraints` property to the type of constraint being used. For example:

   ```
   documentSetConstraints=/atg/commerce/search/StringConstraint
   ```

To view results in multiple languages that are a subset of the total list of languages supported for production, you need to create several components, one for each language you want to support. For example, to view results in French and English, follow these steps:

1. Create a properties file for an English String Constraint component of class `atg.search.routing.command.search.StringConstraint` in a Search Testing directory in your local configuration layer, such as `<ATG9dir>/home/localconfig/atg/commerce/search`:

   ```
   $class=atg.search.routing.command.search.StringConstraint
   $scope=request

   property=language
   operation=equal
   value=English
   ```

2. In the same location, create a similar component called French String Constraint with French as the language:

   ```
   $class=atg.search.routing.command.search.StringConstraint
   $scope=request

   operation=equal
   property=language
   value=French
   ```

3. In the same location, create a Constraint Group component that will combine the constraints in the String Constraint components that you created.

   ```
   $class=atg.search.routing.command.search.ConstraintsGroup
   $scope=request

   constraints=EnglishStringConstraint,FrenchStringConstraint
   operation=or
   ```

4. Register the new component with the Query Request component in `/atg/commerce/search/catalog` by setting the `documentSetConstraints` property to the Constraints Group component being used. For example:

   ```
   documentSetConstraints=StringConstraint
   ```

Customizing Search Testing Inputs and Outputs

You can customize Search Testing to add search inputs to tests and to modify which properties appear in the list of search results.

This section contains the following topics:

[Customizing the Results List Output](#)

[Property Mappings between Search Testing and the Search Response XML](#)

Adding Inputs to Search Testing

You can modify the Search Testing UI to add an “Advanced Inputs” section that contains your custom Search Testing inputs. This allows you to test searches using properties that are not included in the default Search Testing configuration. When you save a search test, the value of your input is automatically saved as long as it follows the format provided in the default `advancedIputs.jsp` file.

Note: ATG does not support changes to the out-of-the-box search test inputs.

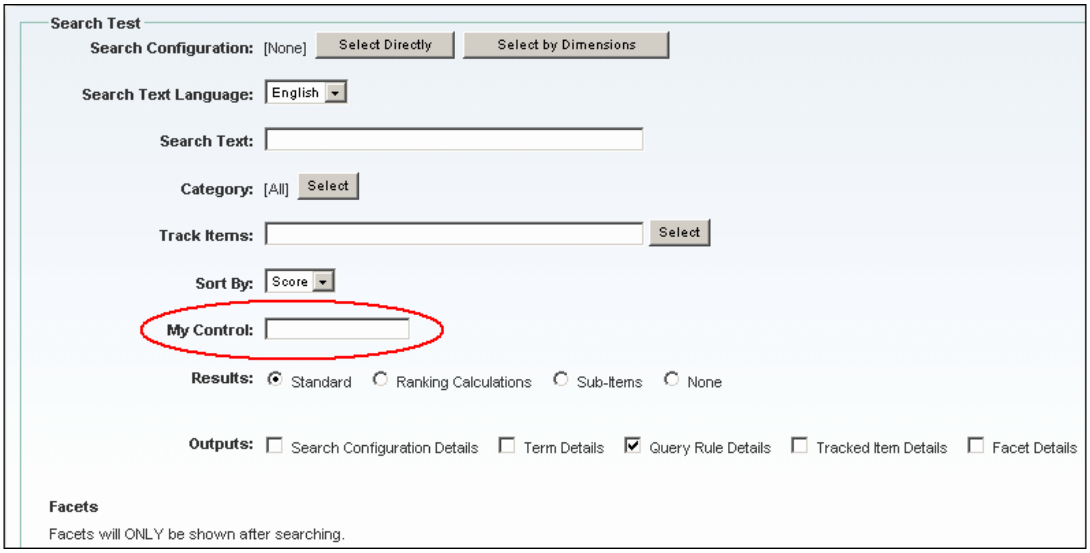
This topic contains the following information:

[Steps to Add an Input to Search Testing](#)

[Sample Search Testing Form Handler Subclass](#)

[Sample Advanced Inputs JSP](#)

The following figure illustrates how custom inputs appear in Search Testing:



Search Configuration: [None]

Search Text Language: English

Search Text:

Category: [All]

Track Items:

Sort By: Score

My Control:

Results: ● Standard ○ Ranking Calculations ○ Sub-Items ○ None

Outputs: □ Search Configuration Details □ Term Details ☑ Query Rule Details □ Tracked Item Details □ Facet Details

Facets

Facets will ONLY be shown after searching.

Steps to Add an Input to Search Testing

To add an input to Search Testing:

1. Create an application module for your customizations. See *Creating a New Application Module* in the *ATG Programming Guide*.

   ATG recommends that you create a new module to ensure that you always have a working application and to avoid issues when you upgrade.

2. If the input you are adding requires a change to the search query request, for example, a date range input that narrows the search results by availability date, you must extend the form handler class used by the `atg/search/web/assetmanager/SearchTestingFormHandler` component, as follows:

   - Create a subclass of the underlying form handler class, `atg.search.web.assetmanager.SearchTestingFormHandler` located in `DCS-UI.Search`.
   - In the subclass, override the `preSearch` method of the class to implement the search request logic required for your custom search test inputs.

     See Sample Search Testing Form Handler Subclass for sample code that you can use as a starting point.

3. Copy the `advancedInputs.jsp` file to your module and give it a custom name, such as `myAdvancedInputs.jsp`. The file is located in the `DCS-UI.Search` module at:

   ```
   <ATG9dir>/DCS-UI9.1/DCS-UI/Search/j2ee-apps/DCS-UI-Search.ear/
   DCS-UI-Search.war/searchTesting/input
   ```

4. Edit the JSP file to add your custom inputs. Sample code is provided in the default `advancedInputs.jsp` file as a starting point.

   See Sample Advanced Inputs JSP for an example.

5. Override the `SearchTestingFormHandler` component to update the `pageConfigMap` property with your new JSP file. There are two things you must specify in the mapping:

   - The name and location of the new JSP file.
   - The Web application context root for the file. (The context identifies your web application so the DSP include can include that file.)

   The following shows an example of the `pageConfigMap` property with an updated advanced inputs JSP file:

   ```
   pageConfigMap+=\
     advancedInputsPage=/custom/myAdvancedInputs.jsp,\
     advancedInputsContext=DCS-UI-Search,\
   ```

   For more information about overriding a component, see the *ATG Programming Guide*.

6. In your `SearchTestingFormHandler` component, update the `$class` value to reference your new subclass.

7. (Optional) Customize the results list output to display search result data related to your custom inputs. See Customizing the Results List Output.

8. Assemble the ATG Merchandising application and include your module in the modules list. See Assembling an ATG Merchandising Application.

9. Deploy and start ATG Merchandising.

10. Verify that your new search test inputs work as expected.

Sample Search Testing Form Handler Subclass

Use the following sample code as a starting point for creating a subclass of the `atg.search.web.assetmanager.SearchTestingFormHandler` class:

```
Public void preSearch(
  DynamoHttpServletRequest pRequest,
  DynamoHttpServletResponse pResponse
  ) throws ServletException, IOException {

    QueryFormHandler queryFormHandler =
       getQueryFormHandler();
    QueryRequest request =
       (QueryRequest)queryFormHandler.getSearchRequest();
    String myControlValue =
       getAdvancedParameters().get( "myControl" );

    /*
    logic to set something in request based on myControlValue
    */

    super.preSearch( pRequest, pResponse );
  }
```

Sample Advanced Inputs JSP

The `advancedInputs.jsp` file contains the following sample code as a starting point for writing the JSP for your custom inputs:

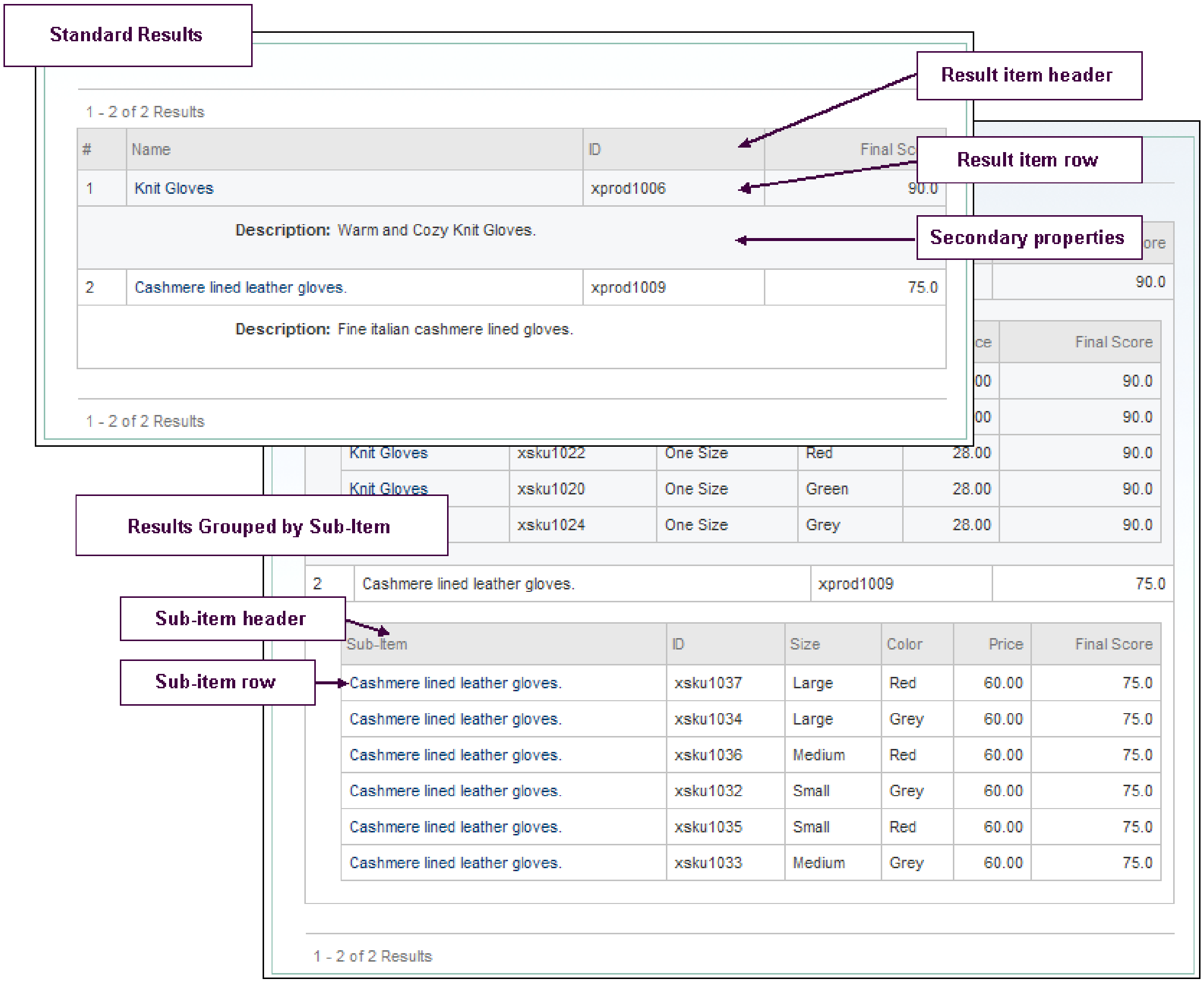
```
<%@ taglib prefix="c"     uri="http://java.sun.com/jstl/core" %>
<%@ taglib prefix="dspel" uri="http://www.atg.com/taglibs/daf/dspjspELTaglib1_0"%>
<%@ taglib prefix="fmt"   uri="http://java.sun.com/jstl/fmt" %>
<%@ taglib prefix="assetui-search"
     uri="http://www.atg.com/taglibs/assetui search" %>
<%@ taglib prefix="web-ui" uri="http://www.atg.com/taglibs/web-ui" %>

<%-- dspel:page>

<c:set var="formHandlerPath" value="${requestScope.formHandlerPath}"/>
<c:set var="formHandler" value="${requestScope.formHandler}"/>

<dt>My Control:</dt>
<dd>
  <dspel:input id="myControl" type="text"
    bean="${formHandlerPath}.advancedParameters.myControl" />
</dd>

</dspel:page--%>
```

Customizing the Results List Output

You can add or remove properties from the search test results list output. For example, if your site includes manufacturer name in search results, you can add that property to the results list output of your search tests.

You can customize the following sections of the results list output:

- Search result item header and row
- Search result item secondary properties
- Search result sub-item header and row (if you group SKUs by product)



Standard Results

1 - 2 of 2 Results

#	Name	ID	Final Sc
1	Knit Gloves	xprod1006	90.0

Description: Warm and Cozy Knit Gloves.

2	Cashmere lined leather gloves.	xprod1009	75.0

Description: Fine italian cashmere lined gloves.

1 - 2 of 2 Results

Result item header

Result item row

Secondary properties

Results Grouped by Sub-Item

				ore
				90.0

				ce	Final Score
				00	90.0
				00	90.0
Knit Gloves	xsku1022	One Size	Red	28.00	90.0
Knit Gloves	xsku1020	One Size	Green	28.00	90.0
	xsku1024	One Size	Grey	28.00	90.0

2	Cashmere lined leather gloves.	xprod1009	75.0

Sub-item header

Sub-item row

Sub-Item	ID	Size	Color	Price	Final Score
Cashmere lined leather gloves.	xsku1037	Large	Red	60.00	75.0
Cashmere lined leather gloves.	xsku1034	Large	Grey	60.00	75.0
Cashmere lined leather gloves.	xsku1036	Medium	Red	60.00	75.0
Cashmere lined leather gloves.	xsku1032	Small	Grey	60.00	75.0
Cashmere lined leather gloves.	xsku1035	Small	Red	60.00	75.0
Cashmere lined leather gloves.	xsku1033	Medium	Grey	60.00	75.0

1 - 2 of 2 Results

Before you begin, make sure the properties that you are adding to the search test output meet the following conditions:

- The property exists in the search index.

To be in the index, it must be included in the `IndexingOutputConfig` definition file. See *Indexing Repository Data* in the *ATG Search Query Guide*.

- The property value is included in the search response XML.

  Configure the `textProps` property (for text-searchable properties) or the `docProps` property (for metadata properties) in the `QueryRequest` component to include the property you want to add to Search Testing. For information about modifying the `QueryRequest` component, see Updating Components with Production Values in this guide. For information about the `textProps` and `docProps` properties, see *Setting Request Properties* in the *ATG Search Query Guide*.

To customize the search test results list output:

1. Create an application module for your customizations. See *Creating a New Application Module* in the *ATG Programming Guide*.

   ATG recommends that you create a new module to ensure that you always have a working application and to avoid issues when you upgrade.

2. Copy the JSP files that you want to customize to your module.

   The results list output JSP files are located in the `DCS-UI.Search` module at:

   ```
   <ATG9dir>/DCS-UI9.1/DCS-UI/Search/j2ee-apps/DCS-UI-Search.ear/
   DCS-UI-Search.war/searchTesting/output
   ```

   The out-of-the-box files are named:

   - `resultTableHeader.jsp` – The result item table heading.
   - `resultTableRow.jsp` – The result item row.
   - `resultTableSecondaryProperties.jsp` – The result item secondary properties.
   - `subTableHeader.jsp` – The result sub-item table heading.
   - `subTableRow.jsp` – The result sub-item row.

   Rename your version of each file to avoid confusion, for example, `myResultTableHeader.jsp`.

3. Edit the JSP files to add or remove properties from the output. If you edit the result row or sub-item row JSP, be sure to also edit the corresponding table heading JSP.

   The out-of-the-box JSP files rely on property mappings that map the search response XML properties to Search Testing UI properties. See Property Mappings between Search Testing and the Search Response XML.

   You can add new mappings or just refer directly to the search response properties. The following example retrieves the value of a custom property named `myProperty` from the search results:

   ```
   <dspel:getvalueof
   bean="${result.document.properties["myProperty"]}"/>
   ```

   To support localization, the table heading JSP files take heading names from a resource bundle. The resource bundle is located at:

```
<ATG9dir>/AssetUI/Search/lib/classes.jar/atg/search/web/
assetmanager/WebAppResources.properties
```

4. Override the `SearchTestingFormHandler` component to update the `pageConfigMap` property with your new JSP files. There are two things you must specify in the mapping:

   - The name and location of the new JSP file.
   - The web application context root for the file. (The context identifies your web application so the DSP include can include that file.)

   The following shows an example of the `pageConfigMap` property with updated JSP files for a custom result table header and result row:

   ```
   pageConfigMap+=\
     resultTableHeaderPage=/custom/myResultTableHeader.jsp,\
     resultTableHeaderContext=DCS-UI-Search,\
     resultTableRowPage=/custom/myResultTableRow.jsp,\
     resultTableRowContext=DSC-UI-Search,\
   ```

   For more information about overriding a component, see the *ATG Programming Guide*.

5. Assemble the ATG Merchandising application and include your module in the modules list. See Assembling an ATG Merchandising Application.

6. Deploy and start ATG Merchandising.

7. Verify that the properties you added to the results list output appear as expected.

Property Mappings between Search Testing and the Search Response XML

ATG Merchandising uses a mapping component to map Search Testing properties to properties returned in the search response XML file. This provides an easy way to configure Search Testing to support different search options, such as SKU indexing or grouping by product (for more information on these options, see Using Grouping). You can also update the mapping to accommodate custom property names in your search response XML.

The property mapping component is:

```
atg/search/testing/SearchTestingUIPropertyMapping
```

The sections that follow describe the default property mappings for each supported search configuration:

Mappings when ATG Search Is Indexed by Product

The following property mappings are used if you index your catalog by product:

```
propertyMapping=itemId=$repositoryId,\
itemDisplayName=role:displayname,\
itemDescription=role:description,\
viewItemDisplayName=displayname
```

The following table describes each item in the mapping:

Description	Search Testing UI Property	Search Response XML Property
Product ID Results list ID column	`itemId`	`$repositoryId`
Product name Results list Name column	`itemDisplayName`	`role:displayname`
Product Description Results list Description secondary property	`itemDescription`	`role:description`
Product name Content Item Details popup	`viewItemDisplayName`	`displayname`

Mappings when ATG Search Is Indexed by SKU

The following property mappings are used if you index your catalog by SKU but you do not group SKUs by product:

```
propertyMapping=itemId=childSKUs.$repositoryId,\
itemDisplayName=role:childskus.displayname,\
itemDescription=role:childskus.description,\
viewItemDisplayName=childskus.displayname
```

The following table describes each item in the mapping:

Description	Search Testing UI Property	Search Response XML Property
SKU ID Results list ID column	`itemId`	`childSKUs.$repositoryId`
SKU name Results list Name column	`itemDisplayName`	`role:childskus.displayname`
SKU Description Results list Description secondary property	`itemDescription`	`role:childskus.description`
SKU name Content Item Details popup	`viewItemDisplayName`	`childskus.displayname`

Mappings when ATG Search Is Indexed by SKU and Grouped by Product

The following property mappings are used if you index your catalog by SKU and group SKUs by product:

```
propertyMapping=itemId=$repositoryId,\
itemDisplayName=role:displayname,\
itemDescription=role:description,\
subItemId=childSKUs.$repositoryId,\
subItemDisplayName=role:childskus.displayname,\
subItemSize=role:childskus.size,\
subItemColor=role:childskus.color,\
subItemPrice=price,\
viewItemDisplayName=displayname,\
viewSubItemDisplayName=childskus.displayname
```

The following table describes each item in the mapping:

Description	Search Testing UI Property	Search Response XML Property
Product ID Results list ID column	`itemId`	`$repositoryId`
Product name Results list Name column	`itemDisplayName`	`role:displayname`
Product Description Results list Description secondary property	`itemDescription`	`role:description`
SKU ID Sub-item results list ID column	`subItemId`	`childSKUs.$repositoryId`
SKU name Sub-item results list Name column	`subItemDisplayName`	`role:childskus.displayname`
SKU size Sub-item results list Size column	`subItemSize`	`role:childskus.size`
SKU color Sub-item results list Color column	`subItemColor`	`role:childskus.color`
SKU price Sub-item results list Price column	`subItemPrice`	`Price`
Product name Content Item Details popup	`viewItemDisplayName`	`Displayname`
SKU name Content Item Details popup	`viewSubItemDisplayName`	`childskus.displayname`

Index