ORACLE®

ATG WEB COMMERCE

Version 10.0.2

Business Control Center Administration and Development Guide

Oracle ATG
One Main Street
Cambridge, MA 02142
USA

ATG Business Control Center Administration and Development Guide

Document Version

Doc10.0.2 BCCADMINv1 04/15/2011

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 to the ATG Business Control Center

The ATG Business Control Center is a browser-based interface that allows marketers and others in your organization to create and edit many of the elements required to maintain one or more ATG-based Web sites. The activities you can perform in the ATG Business Control Center are divided into the following general areas:

- ATG Content Administration activities. The ATG Business Control Center is the primary interface for performing ATG Content Administration tasks, including the creation and deployment of Web site content.
- Management of ATG Personalization assets. You use the ATG Business Control Center to create and manage the components that are responsible for delivering personalized Web site content (content that changes dynamically for each site visitor). These components include user segments, content targeters, and content groups.
- Management of the profiles required by ATG Business Control Center users, including their organization and role assignments.

The ATG Business Control Center is also used as the starting point for launching several ATG applications, including the following:

- ATG Merchandising
- Search Administration
- The Site Administration utility
- ATG Outreach
- ATG Campaign Optimizer
- ATG Service applications

The ATG Business Control Center is installed on the asset management server, which is the server where ATG Content Administration is also installed and where you develop content before deploying it to your live sites. It is also installed on the agent server, which is used by ATG Service applications. In general terms, the steps you complete to configure the ATG Business Control Center are part of the process you follow to configure ATG Content Administration: setting up the ATG Content Administration database, configuring deployment targets, and so on. This manual assumes that you have completed the setup process for ATG Content Administration as described in the *ATG Content Administration Programming Guide*.

For more information on the servers where the ATG Business Control Center runs, refer to the architecture diagram in the *ATG Multiple Application Integration Guide*.

About This Manual

This manual is intended for administrators and developers. It includes the following chapters:

- Introduction to the ATG Business Control Center – describes setting up and starting the ATG Business Control Center. Includes information on login requirements and user profile architecture. (Administrators)
- Configuring Profile Repositories – shows the internal, external, and preview profile repository model used in the ATG product suite and describes various configuration steps for these repositories. (Administrators)
- Setting Up Preview – describes how to configure the preview feature, which allows ATG Business Control Center users to view proposed changes before deploying them to a live site. (Administrators)
- ATG Business Control Center Security – describes how to manage access control to specific areas of the ATG Business Control Center. (Administrators)
- Managing User Profiles – describes how to create and maintain profiles for ATG Business Control Center users. (Administrators)
- Creating Organizations and Roles – explains how to set up a hierarchy of organizations and roles for your ATG Business Control Center users, which you can then use to manage site access, among other activities. (Administrators)
- Configuring Segment Lists – explains how to use the UI to specify the segments that you want to include in activities such as reporting. (Administrators or developers)
- Configuring the Asset Manager – describes how to change the way specific asset types are displayed and edited in the ATG Business Control Center. (Developers)
- Customizing the UI for Specific Tasks and Activities – shows how to use a task configuration file to modify the UI for specific user activities. (Developers)
- Appendix: Tags in a Task Configuration File – describes the tags defined in the `taskConfiguration_1.0.dtd`. (Developers)

Additional Reading

In order to perform the tasks described in this manual, you need to be familiar with ATG Content Administration and ATG Personalization concepts. References are made throughout this guide to the documentation for these products. In particular, it is strongly recommended that you review the database and deployment setup information in the *ATG Content Administration Programming Guide* and the user profiling chapters of the *ATG Personalization Programming Guide* in addition to this manual.

For information on creating user segments, content targeters, and content groups, refer to the *ATG Business Control Center User's Guide*.

For information on using the ATG Business Control Center within a multiple-application environment, refer to the *ATG Multiple Application Integration Guide*.

For information about using ATG products in a multisite environment (one that supports several Web sites that share items such as user profiles and shopping carts), refer to the *ATG Multisite Administration Guide*.

Installing the ATG Business Control Center

The ATG Business Control Center requires ATG Content Administration and is installed automatically with the ATG platform if your product set also includes ATG Content Administration. For information, refer to the *ATG Installation and Configuration Guide*.

Importing Data Required by the ATG Business Control Center

When you are configuring the ATG Content Administration (versioned) database, note in particular that if you are using any database other than the SOLID evaluation database you must run the following scripts to import data necessary for using the ATG Business Control Center. This step is part of the longer setup procedure described in the *ATG Content Administration Programming Guide* in the section *Configuring the ATG Content Administration Database*. Particular attention is called to it here because running these scripts is often overlooked. Forgetting to run them causes a number of errors to occur in the ATG Business Control Center UI:

- `<ATG10dir>/BIZUI/install/importBizui.sh` (or `.bat` for Windows). Imports default data for the ATG Business Control Center framework.
- `<ATG10dir>/AssetUI/install/importAssetUI.sh` (or `.bat`). Imports view mapping data used by the `AssetUI` module.
- `<ATG10dir>/DPS-UI/install/import/importDPSUI.sh` (or `.bat`). Imports Personalization data for the ATG Business Control Center.

Login Requirements

To log into the ATG Business Control Center and access specific areas within it, you need the following user configuration:

- A user profile, defined in the Internal Users interface of the ATG Business Control Center. When you log into the application (see below), you are prompted to enter a user name and password. The values to enter are the login name and password specified in your user profile. A default profile is provided – see Starting the ATG Business Control Center.

- An ATG Portal role, for example `100001-member`. This role is required for access to the ATG Business Control Center framework. ATG Portal roles are assigned through the Internal Users interface. In this case, the appropriate roles are located by default in `Global Roles > Root Folder > Bizui`. If you have only this role, you have access to the Users options in the ATG Business Control Center but not to any other option on the BCC Home page. For more information on assigning roles, see Creating Roles.
- An ATG Content Administration role, for example `EPub-User` or `EPub-Manager`. These are also added to your user profile through the Internal Users interface in the ATG Business Control Center. Although you do not need one of these roles to start the ATG Business Control Center, you do need one to access all ATG Content Administration features, which include projects. Because the targeting features you work with in the ATG Business Control Center use projects, you need this role to be able to work with them. If you do not have this role, the Targeting and Segmentation option on the Home page does not appear.
- Some ATG applications use ATG Customer Intelligence to provide reports. For example, ATG Outreach includes reports that business managers can use to view data about e-mail campaigns. If you want to view any reports, you need one of the roles that allows access to the Report Center, for example `Report Viewer` or `Reporting Administrator`. These roles are located in the `Global Roles > Root` folder. The roles appear only if ATG Customer Intelligence is installed.
- A Web browser. The ATG Business Control Center supports various browsers and operating systems. For up-to-date information, see the Supported Environments page on atg.com: `http://www.atg.com/en/products-services/commerce/supported-environments/`.

Starting the ATG Business Control Center

To start the ATG Business Control Center:

1. Make sure that the appropriate database is running. If you are evaluating ATG products on a local machine running Windows, you can start the SOLID database installed with ATG 10.0.2 by selecting Start > Programs > ATG 10.0.2 > ATG Content Administration > Start SOLID Server.
2. Assemble and start the ATG products and modules that you want to use with the ATG Business Control Center. The BIZUI module is required to run the ATG Business Control Center, so the assembly list must contain a product or module that calls BIZUI, for example ATG Content Administration or an application that is launched through the ATG Business Control Center, such as ATG Merchandising. The ATG Business Control Center is started automatically as part of this process.

   In order to enable Preview, the `runAssembler` command must include the `-layer Preview` switch. For example

   ```
   runAssembler –layer Preview –m PubPortlet MyVersionedApp MyPreviewApp
   ```

Note that the `PubPortlet` module includes most of the functionality required to perform basic ATG Content Administration tasks, and it calls a number of other modules that support the ATG Business Control Center, including BIZUI.

For information on startup and assembler commands, refer to the documentation for the appropriate ATG product. For general information on assembling applications, including a basic description of ATG platform modules, refer to the *ATG Programming Guide*.

For evaluation purposes on a local Windows machine, start ATG Content Administration as follows: select Start > Programs > ATG 10.0.2 > ATG Content Administration > Start ATG 10.0.2.

3. Point your Web browser to the server where the ATG Business Control Center is running. The URL takes the following form:

```
http://<hostname>:<port>/atg/bcc
```

where `<hostname>` is the name of the server, and `<port>` is the number of the port which that server uses for the ATG Business Control Center.

```
http://anycorp.androcles:8080/atg/bcc
```

If you are running the ATG Business Control Center on a local machine, enter the following URL:

```
http://localhost:<port>/atg/bcc
```

The default port numbers for JBoss, IBM WebSphere, and Oracle WebLogic are 8080, 9080, and 7001, respectively. See the *ATG Installation and Configuration Guide* for details.

4. When the ATG Business Control Center login dialog box appears, enter the login name and password defined in your user profile. The fields are case sensitive. For evaluation purposes, or to set up profiles for other users, use the following login name and password:

```
admin
admin
```

ACC Support

The Web-based ATG Business Control Center is intended to supersede the ACC interface used in earlier versions of the ATG platform. In ATG 10.0.2, the ACC is still available and can be used to edit the ATG Personalization components described in this guide. However, please note the following:

- Content targeters and content groups that you create in the ACC can be opened and edited in the ATG Business Control Center. However, once you edit these items in the ATG Business Control Center, you cannot subsequently use them in the ACC.
- Content targeters and content groups can be configured to support a multisite environment (one in which a single installation of ATG products is used to run more than one Web site). However, the multisite features for these items are not available in

the ACC. To support multiple sites, manage these items through the ATG Business Control Center.

- User profiles, organizations, and roles can be created and edited in either the ATG Business Control Center or the ACC. The ATG Business Control Center is the recommended interface. Note also that the ACC references the external user profile repository by default, so any changes you make through the Users menu in this application apply to external profiles (live profiles on your production sites). For information on how to use the ACC to edit internal profiles, refer to the ATG Service documentation.
- The ATG Business Control Center requires ATG Content Administration. If your ATG product suite does not include ATG Content Administration, you must continue to use the ACC to create and edit the components described in this guide. See the *ATG Personalization Guide for Business Users*.
- Slots, scenarios, and workflows must be edited in the ACC.

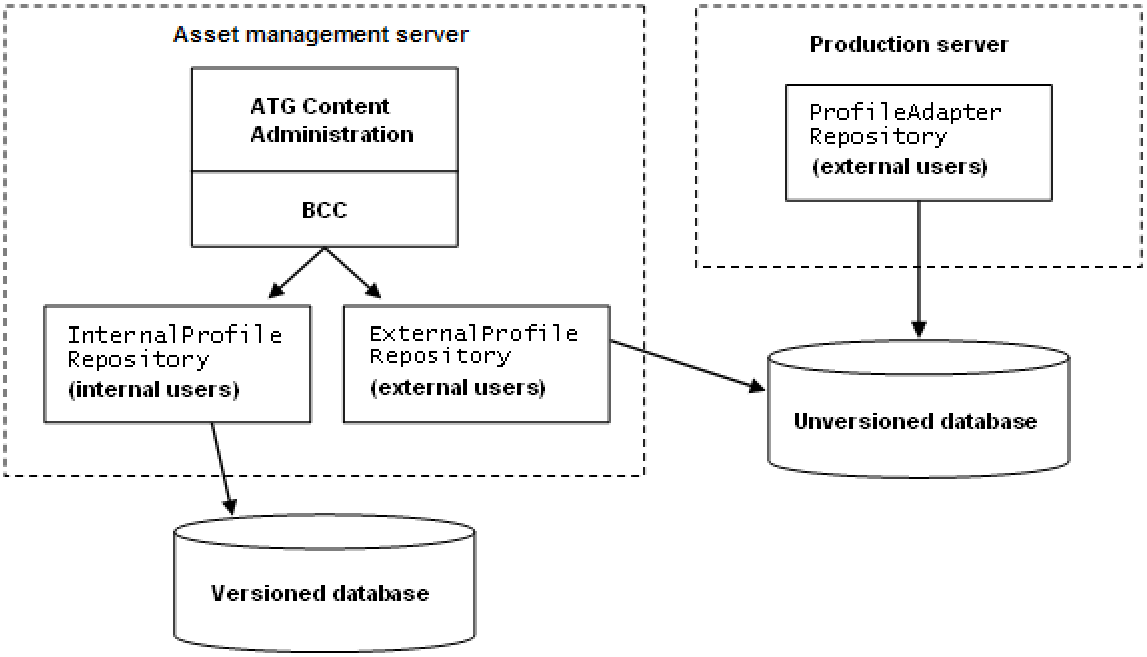
  Note: In order to use the ACC on the asset management server to edit these items, you must assemble the application for preview.

2 Configuring Profile Repositories

As explained in the Managing User Profiles chapter of this guide, an ATG product environment can contain three kinds of user profiles:

- Internal profiles—profiles for ATG application users, including users of the ATG Business Control Center. Examples – people who use the ATG Business Control Center to manage targeted content; administrators who configure organizations and roles; Web site managers who deploy content to your production sites; ATG Outreach users; and customer service representatives (ATG Service users).
- External profiles – profiles for Web site customers (for ATG Commerce or ATG Service installations), or visitors to non-commerce sites.
- Preview profiles – if an ATG installation uses preview features, the environment can also include preview profiles, which are used to test content against a non-live profile before it is deployed.

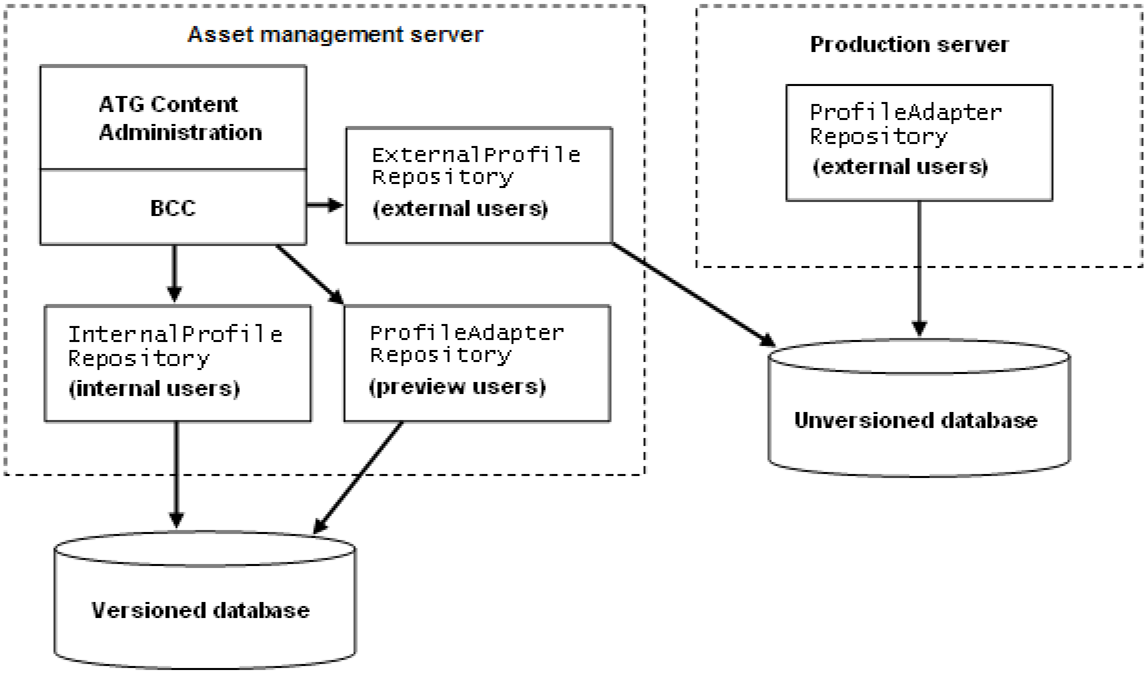
If preview features are not enabled, the following profile repository model is used:



Profile Repository Model without Preview

The important thing to note about this model is that there are two instances of the `ProfileAdapterRepository`, both of which store external profiles and point to the **production (unversioned)** database. The ATG Business Control Center uses the instance on the asset management server to edit and otherwise manage external profiles. In addition, there is an `InternalProfileRepository` on the asset management server, which stores internal (ATG application) user profiles and references dpi* tables (for example `dpi_user`) in the versioned ATG Content Administration database.

If preview features are enabled, an ATG installation uses the profile repository model shown below:



Profile Repository Model with Preview Enabled

As the diagram illustrates, in this configuration there are three profile repositories on the asset management server:

- An `InternalProfileRepository`
- An `ExternalProfileRepository` that points to the **production (unversioned)** database. This repository has the same function as the `ProfileAdapterRepository` on the management server in the non-preview model. It references external (customer) profiles and gives the ATG Business Control Center access to those profiles.
- An instance of the `ProfileAdapterRepository` that manages a set of sample profiles used by preview features. The preview `ProfileAdapterRepository` uses the dps* tables (for example, `dps_user`) in the versioned ATG Content Administration database. It uses the same `userProfile.xml` definition file as the

ProfileAdapterRepository on the production server. See Creating Profiles for information on how to populate this repository.

One profile repository is located on the production server: a ProfileAdapterRepository points to the production (unversioned) database. This repository stores external (customer) profiles.

Viewing External Profile Properties Added by Other Applications

As described in the *ATG Personalization Programming Guide*, the standard profile supplied with the ATG platform contains properties such as first name, last name, login, and e-mail address. Other ATG applications add their own properties to the external profile definition. ATG Outreach, for example, adds opt-in properties that track whether a customer has chosen to receive mailings on a particular subject.

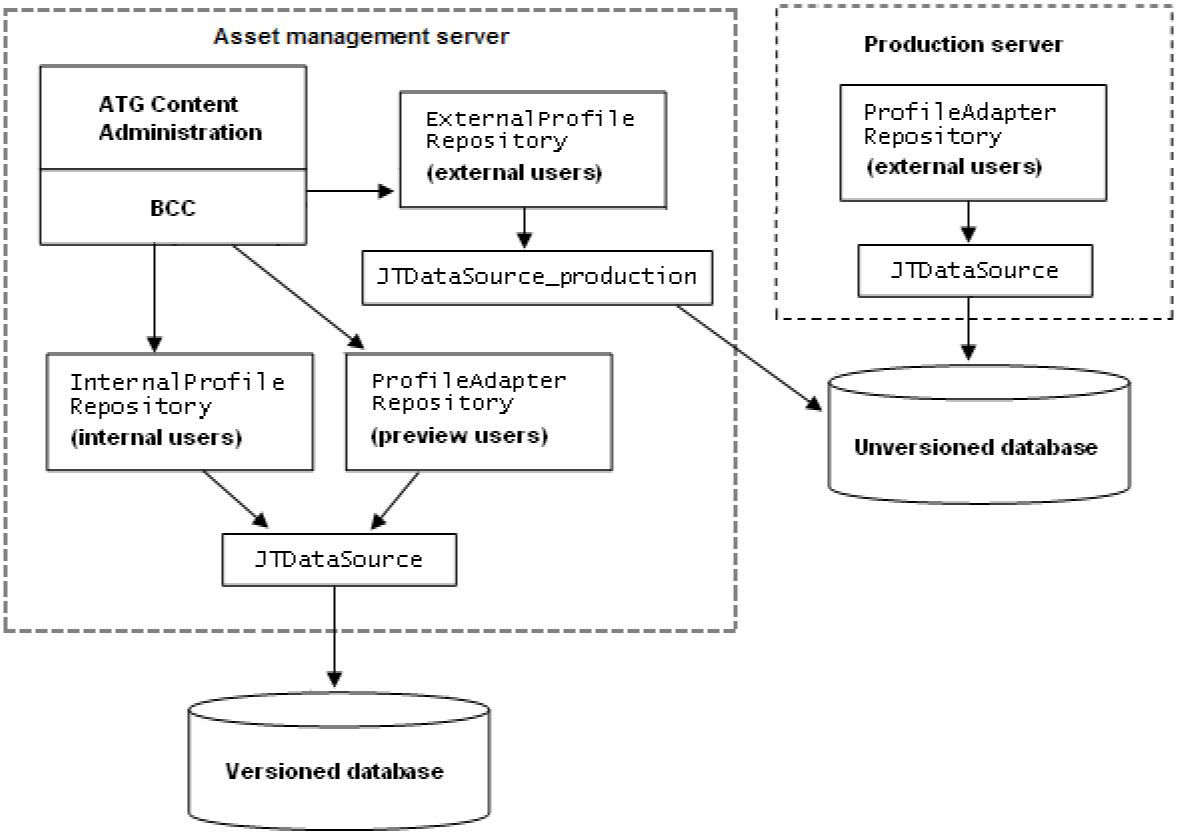
The extensions that each application makes to the external profile definition are stored in that application's ExternalUsers sub-module. Including these modules when you start each application allows ATG Business Control Center users on different servers to see all the profile properties added by other applications. For detailed information, refer to *Sharing External Profile Definitions Across Multiple Applications* in the *ATG Multiple Application Integration Guide*.

Profiles in a Multisite Environment

In a multisite environment, a single installation of ATG products supports more than one Web site, and the sites are configured to share data such as a shopping cart. In such an environment, user profiles are automatically shared across all your Web sites. The profile repository model is identical in a multisite-enabled installation to the model used by an environment that supports only one Web site. For more information, refer to the *ATG Personalization Programming Guide*. For more information on configuring a multisite environment, see the *ATG Multisite Administration Guide*.

Configuring Profile Repository Data Sources

Like any ATG repository, profile repositories use data source components to connect to a specified set of tables in the target database. The following diagram shows the default data source components used by the profile repositories in ATG 10.0.2. (Note that the diagram shows the profile model with preview enabled, but the data source configuration is the same if you don't use preview.)



By default, the `/atg/dynamo/service/jdbc/JTDataSource_production` component on the asset management server points to the versioned database. After installation, change this component so that it points to the production database, as shown in the diagram. Note that this step is necessary even if you are using the SOLID evaluation database.

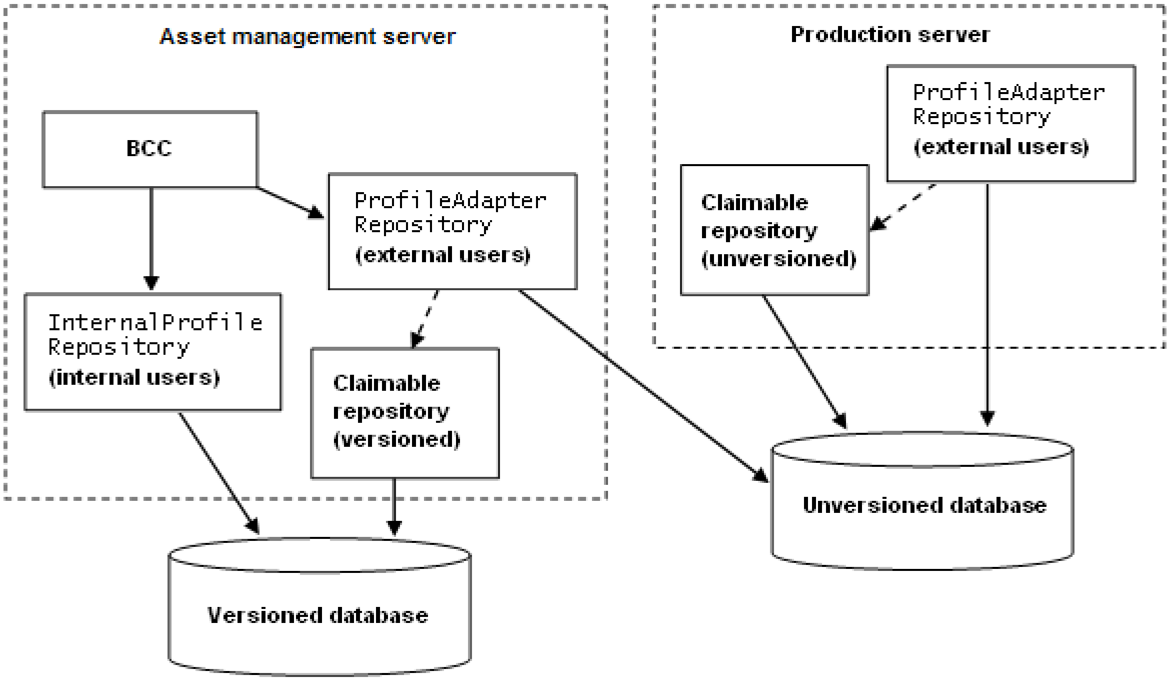
If you are using the SOLID evaluation database, you do not need to perform any other configuration steps for the profile repository data sources. However, when you switch to a production-quality database, set the remaining two data source components as follows:

- Configure the `/atg/dynamo/service/jdbc/JTDataSource` component on the asset management server to connect to the versioned database.
- Configure the `/atg/dynamo/service/jdbc/JTDataSource` component on the production server to connect to the production database.

For more information, including instructions for replacing the default data source components for use with specific application servers, refer to *Configuring Data Sources and the Transaction Manager* in the *ATG Installation and Configuration Guide*.

Configuring Foreign Repository References in External Profiles

If your external profiles contain any references to versioned repositories, you must configure the profiles so that the references point to the appropriate database. A customer profile on an ATG Commerce site, for example, can have properties that reference the Claimable repository, which stores information about coupons. The Claimable repository is a versioned repository, whose assets are edited on the management server and deployed to its unversioned instance on the production server. Without additional configuration, the instance of the external profile repository on the asset management server would point by default to the **versioned** instance of the Claimable repository. The following diagram illustrates this relationship:


Asset management server
BCC
ProfileAdapter Repository (external users)
InternalProfile Repository (internal users)
Claimable repository (versioned)
Versioned database
Production server
ProfileAdapter Repository (external users)
Claimable repository (unversioned)
Unversioned database

Unconfigured References to Foreign Repositories

In this unconfigured setup, if you used the ATG Business Control Center to edit an external profile that had references to any versioned repository, it would be possible to drill down to properties in the versioned repository on the asset management server. You could not edit any such properties this way (the UI would prevent you from doing so because you would not be working within the context of a project). However, if any of these properties represented an asset that you select through a picker, it would be possible to add the asset to the profile. You would thus be attaching a versioned asset to an external profile on the production server, but the asset might not have been deployed. Doing so could cause undesirable results.

To avoid this problem, identify any references from your external profiles to custom versioned repositories. Then complete the steps shown below. Note that standard ATG repositories are preconfigured appropriately—you need to complete the procedure only for repositories that you have added.

1. Add versioned and unversioned instances of the foreign repository to the asset management server (for example, (for example, `/atg/myApp/MyRepository` and `/atg/myApp/MyRepository_production`).
2. In the non-versioned (production) instance of the foreign repository on the asset management server (for example, `MyRepository_production`), set the `foreignRepositoryMapper` property to `/atg/repository/ProductionRepositoryMapper`.
3. In the `localconfig` layer on the asset management server, add a `/atg/repository/ProductionRepositoryMapper.properties` file. In the `RepositoryMappings` property, specify the versioned repository as a key and the matching non-versioned (production) repository as the value, as shown in the example:

   ```
   RepositoryMappings+=\
      /atg/myApp/MyRepository=/atg/myApp/MyRepository_production
   ```

4. In the `localconfig` layer on the asset management server, add a `/atg/registry/ContentRepositories.properties` file. Add the versioned repository to the `initialRepositories` property, as shown below:

   ```
   initialRepositories+=\
      /atg/myApp/MyRepository
   ```

5. In the `localconfig` layer on the asset management server, add a `/atg/dynamo/service/AssetResolver.properties` file. Add the non-versioned (production) repository to the `additionalAssetSources` property, as shown below:

   ```
   additionalAssetSources+=\
      /atg/myApp/MyRepository_production
   ```

   This step is necessary so that the ATG Business Control Center can create and resolve asset URIs for items in the new repository. If you see a message similar to the following when you try to display an item that contains a reference to a foreign repository, make sure that this step has been completed.

   ```
   "Cannot find asset with URI atgasset:/<repository>_production/
   <repository ID>.You may not have permission to view the asset."
   ```

Note that the same situation could occur for unversioned repositories as well, particularly if preview user profiles could reference items in those repositories. Essentially, whenever you have a repository that is referenced by an external or preview user profile, you need to configure it appropriately with production and non-production instances on the asset management server that point to the correct database.

Configuring Caching for Viewing Profile Changes

To ensure that a user viewing profiles through the ATG Business Control Center on one server can see changes made to a profile on another server, use distributed hybrid cache mode for profile repository items. This cache mode is described in the *ATG Repository Guide*.

3 Setting Up Preview

ATG Content Administration and the Asset Manager provide a preview feature that lets you test project assets on a preview server before deploying them to your live sites. You can evaluate the assets from the perspective of different user profiles.

This chapter provides an overview of preview capabilities and how to set it up in your environment. For information about previewing assets, see the *ATG Business Control Center User's Guide* or the documentation for your application.

Note that this chapter applies to both JSP-based and Flex-based applications, including ATG Merchandising and Site Administration.

Supported Asset Types

You can preview the following asset types:

- `GSARepository` assets—for example, product catalog assets.
- Personalization assets: user segments, targeters, content groups, slots, and scenarios.

Preview Benefits

Preview is best suited for environments where content developers and reviewers need to evaluate their work regularly on a production-like site before deployment. Configured through Nucleus components, preview is easy to set up and can support a rich matrix of user profiles and preview sites that are readily accessible to users. You can deploy the preview application on ATG servers that do not run the ATG Business Control Center, and thereby safely provide preview access to users who lack ATG Business Control Center login privileges.

Constraints

The following restrictions apply to preview:

- Preview is project-based: it only shows asset changes in the project where the preview session is launched; assets outside the project show the state they were in when last checked in. Review of cross-project dependencies requires deployment to a staging site, where multiple projects can be deployed and reviewed.
- `WWWFileSystem` assets such as binary documents and media files (for example, MPEG and GIF) are not supported by preview; you must test these on a staging server.
- Faceted Search and Search Merchandising are currently not supported for preview. For information on previewing search results, see the Testing Search Queries chapter in the *ATG Merchandising User Guide.*

- Login and logout functionality is not supported for preview.

Preview Setup Steps

In order to preview assets, you must set up the asset management server designated to run the ATG Business Control Center to support preview. You can also set up additional non-BCC servers to support preview, so multiple applications are available to preview project changes.

In order to enable preview in your environment, perform the following steps:

1. Create a preview application.
2. Configure preview components.
3. Assemble the application for preview and deploy it.
4. Create preview profiles.

You can also use the view mapping system to configure or suppress preview for individual asset types. This is discussed in the last section of this chapter, Asset Preview.

Create a Preview Application

A preview application must be set up as a Web application module on each preview-enabled server. You use a versioned instance of an application that runs on the production server, and deploy this module on a server where the ATG Business Control Center is running; if desired, you can also deploy this module on a non-BCC server. In both cases, you set up versioned instances of the application modules as described in the *ATG Content Administration Programming Guide.*

For example, if an ATG Commerce store in the module `MyCommerceStore` runs on the production server, a preview-enabled server might have a versioned instance of that module called `MyCommerceStoreVer`. You should configure the preview application for startup so when you enter the application's context root as a URL in a browser, the application home page is rendered.

A preview application does not always need to be fully functional—for example, the checkout process might not need to be fully operational. Only the pages required for preview must be accessible and rendered correctly. However, make sure that all resources referenced by preview pages are available on the server.

Note: A Web application must be enabled for preview by its `web.xml` file. By default, all Web applications are enabled for preview. You can disable a Web application for preview as follows:

```
<context-param>
    <param-name>atg.preview</param-name>
    <param-value>false</param-value>
</context-param>
```

Configure Preview Components

in order to enable preview on a non-BCC server, configure the following components:

- PreviewHost
- PreviewPath
- PreviewURL
- PreviewURLManager

Each `PreviewURL` component combines `PreviewHost` and `PreviewPath` components to construct a preview URL; the `PreviewURLManager` determines which `PreviewURLs` are made available to users. By mixing and matching different combinations of preview hosts and paths—for example, the same URL on different hosts—you can provide users a wide variety of preview options.

Preview Configuration on the BCC Server

In order to configure preview on the BCC/asset management server, you only need to set its `Localhost` component properties `hostName` and `port` to the same host and port used by the asset management server. If setting these properties manually, you can find the properties file at this location:

```
<ATG10dir>/home/localconfig/atg/dynamo/service/preview/Localhost.properties
```

PreviewHost

In order to run preview applications on multiple hosts, you must create a `PreviewHost` component from this class:

```
atg.service.preview.PreviewHost
```

This component has the following properties:

Property	Description
`hostName`	The name of the host where the preview application runs
`port`	The port used by `hostName`.

Note: `hostName` must be different from the host used by the asset management server.

PreviewPath

The `PreviewPath` component encapsulates the relative Nucleus path to a preview URL and is created from this class:

```
atg.service.preview.PreviewPath
```

The component has a single property that must be set:

Property	Description
`defaultPath`	The relative Nucleus path to the preview URL. This string can include parameters that are passed to the preview page, including any of the dynamic parameters that preview supports (see Preview URL Parameters). This string must not include the URL's host name or port number. **Note:** If you use WebSphere for a preview application server, the setting must have a trailing slash (/). For example: `defaultPath=/ondemand/`

For example:

```
# /atg/dynamo/service/preview/SketchPath.properties
$class=atg.service.preview.PreviewPath
defaultPath=/ondemand/browse/productDetailWithPicker.jsp?id=$id
```

PreviewURL

The `PreviewURL` component encapsulates the URL to a preview application and is created from this class:

```
atg.service.preview.PreviewURL
```

This component has the following properties:

Property	Description
`name`	The user-defined name of this preview URL, used by the Preview dialog if you do not set the properties `displayNameResourceBundle` and `displayNameResourceKey`.
`resourceBundleName`	The resource bundle that contains the preview site's display name, specified by `displayNameResourceKey`.
`displayNameResourceKey`	The key in `resourceBundleName` that defines the preview site's display name.
`host`	The Nucleus path of a PreviewHost component, set only for preview-enabled servers that run on a host other than one where the BCC server runs. A null setting indicates the local (BCC server) host.

`path`	The Nucleus path of a PreviewPath component. This property must be set.
`protocol`	The protocol to use for this URL—for example, `https`. If this property is not set, the default protocol is `http`.

For example:

```
# /atg/dynamo/service/preview/Sketch.properties
$class=atg.service.preview.PreviewURL
name=Sketch
host=
path=SketchPath
```

PreviewURLManager

The `PreviewURLManager` component maintains a list of `PreviewURL` components and is created from this class:

```
atg.service.preview.PreviewManager
```

This component has one property that must be set:

Property	**Description**
`previewURLs`	A list of PreviewURL components. This property determines which URLs are displayed to Asset Manager or ATG Content Administration users in the Preview Launcher dialog, for previewing project assets.

For example:

```
# /atg/dynamo/service/preview/PreviewURLManager
previewURLs+=\
   /atg/dynamo/service/preview/SketchRemoteHomeURL,\
   /atg/dynamo/service/preview/SketchRemoteCategoryURL,\
   /atg/dynamo/service/preview/SketchRemoteSubCategoryURL,\
   /atg/dynamo/service/preview/SketchRemoteProductURL,\
   /atg/dynamo/service/preview/SketchLocalHomeURL,\
   /atg/dynamo/service/preview/SketchLocalCategoryURL,\
   /atg/dynamo/service/preview/SketchLocalSubCategoryURL,\
   /atg/dynamo/service/preview/SketchLocalProductURL
```

Preview URL Parameters

A preview URL contains dynamic parameters, which at runtime, are replaced by property values, then passed to the page. Parameters can reference special properties and asset properties.

Special Properties

The following table shows the special properties that the preview URL can include:

Property	Description
`$id`	The asset's repository ID or the absolute path of a file.
`$projectId`	The project ID, empty if there is no current project.
`$componentPath`	The Nucleus path of the repository or virtual file system.
`$itemType`	The item descriptor name for repository items or file type for files.

For example, given the following URL, `$id` resolves to a repository item's ID:

```
http://www.myStore/women/Clothing.jsp?prodId=$id
```

Referencing Asset Properties

Dynamic parameters can be set to any property of the previewed asset with the following notation:

`parameter``=$[`*`property-name`*`]`

For example:

```
http://www.myStore/women/Clothing.jsp?prodID=$id&name=$[displayName]
```

The `prodID` parameter resolves to the value of `$id`, while the `name` parameter is assigned the asset's `displayName` property.

Referencing Asset Subproperties

Asset Manager preview lets you set parameters to asset subproperties—that is, properties that are themselves the values of other properties. This lets you access product items with greater specificity. For example, the following URL supplies the parameter `$[template.url]`, in order to obtain the value of the subproperty `url`:

```
http://www.myStore/women/$[template.url]?prodId=$id&name=$[displayName]
```

If you have several product templates, saving each template to the product itself and using a pointer to the template property in the URL ensures that the correct template is used.

Assemble the Application for Preview

To use preview, you assemble the versioned preview application and the ATG Business Control Center with the `–layer Preview` switch. For example:

```
runAssembler –liveconfig acme.ear
   –layer Preview –m PubPortlet DCS-UI DCS.Versioned MyVrsApp
```

You can also supply the `–layer Preview` switch to the `StartDynamoOnJBOSS.bat` script. For example:

```
startDynamoOnJBoss.bat –liveconfig –ear acme.ear
   –layer Preview -m PubPortlet DCS-UI DCS.Versioned MyVrsApp
```

Note: In order to simulate a production-like environment, assemble with the `–liveconfig` option.

For full information on `runAssembler` command options and syntax, see *Developing and Assembling Nucleus-Based Applications* in the *ATG Programming Guide*.

Assembling for Non-BCC Servers

If you run the preview application on multiple servers, two requirements apply:

- For servers that do not run the ATG Business Control Center, assembly must omit the modules `PubPortlet` and, if applicable, `DCS-UI`. In this case, assembly must explicitly include the WebUI module.
- The data of versioned repositories must be consistent across all servers. In order to ensure that asset and user profile changes on one server are replicated on all, you must enable caching—preferably, distributed or distributed hybrid caching modes—on all server instances. For more information about repository caching modes, see *SQL Repository Caching* in the *ATG Repository Guide*.

  Note: Never disable caching on versioned repositories for preview applications or use simple caching mode.

For example, you might assemble an application for preview on a non-BCC server as follows:

```
runAssembler –liveconfig acme.ear –layer Preview –m WebUI DCS.Versioned MyVrsApp
```

Create Preview Profiles

Preview features reference a set of profiles stored in the `ProfileAdapterRepository` on the asset management server. The profiles let you simulate the experience of various users. Preview profiles reside in a preview profile repository, which you first define, then populate.

Defining the Repository

Preview profiles should have the same properties as external profiles in your system, so you can accurately simulate the experience of external users. For example, if you extend the default profile definition so external user profiles have an income property, and you create a targeter that displays different content according to that property's value, the same property must exist in the preview profile. To achieve this, make sure that the preview profile repository uses the same userProfile.xml definition file as the external profile repository.

Populating the Repository

After you define the profile repository, you populate it by creating profiles like other user profiles (see Creating Profiles in the next chapter). For more information on profile repositories, refer to the *ATG Personalization Programming Guide*.

Asset Preview

You can set up preview for specific asset types by configuring one or more URLs as extensions of the view mapping system (described in *Customizing Asset Display* in the *ATG Content Administration Programming Guide*). Preview URLs map repository item descriptors to asset preview pages. If an asset type is configured with a preview URL, the ATG Business Control Center displays that URL in the Preview Launcher dialog, invoked by clicking Preview As in the asset's properties window. If other URLs display (as configured through preview components), the asset preview URL is included among them and is differentiated by a yellow background.

You configure asset preview by setting the atgPreviewURL attribute on the appropriate itemView or itemViewMapping:

- Set itemView to display the same preview button on all tabs of the view.
- Set itemViewMapping to display a different preview button on each tab of a view.

The atgPreviewURL attribute takes the following form:

```
atgPreviewURL=preview-url
```

where *preview-url* points to the JSP where the asset is previewed. This setting is parsed by the page fragment previewURLParser.jspf, located in:

```
<ATG10dir>/PubPortlet/PubPortlets.ear/portlets.war/html/ProjectsPortlet
```

The preview URL can reference an independent page that you create for previewing assets; it can also be a page within a Web application that is installed on the ATG BUSINESS CONTROL CENTER server.

Procedure

To add preview functionality to a Properties tab for an asset type, follow these steps:

1. If necessary, create a JSP to use for the asset preview.
2. In the ACC, choose Publishing > View Mapping
3. Click on List in order to list items of type itemMapping.

4. Click New Item to create a unique `itemMapping` for this asset type.

5. Provide the required mapping information: `itemName`, `itemPath`, and `mode` (set to `edit`).

   Note: Set the name field to * (asterisk).

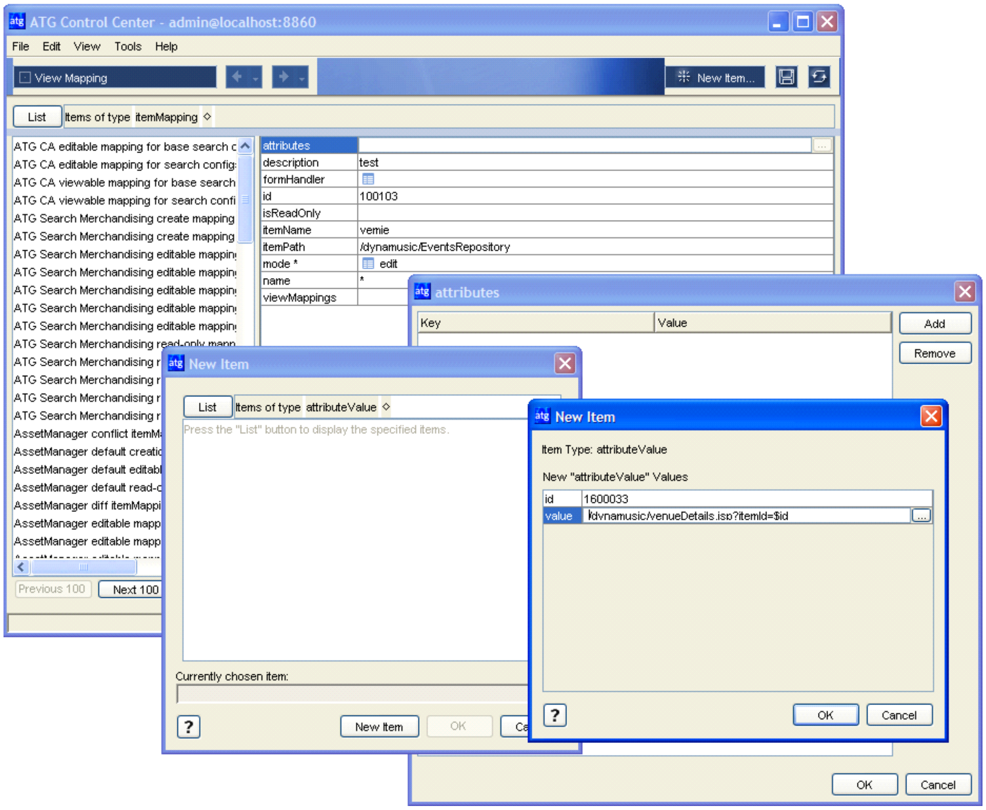
6. Edit the attributes property by clicking on `...`

7. In the Attributes dialog, click Add.

8. In the dialog, click New Item.

9. In the New Item dialog, enter the relative Nucleus path to the preview page URL. For example:

   ```
   /dynamusic/venueDetails.jsp?itemID=$id
   ```



ATG Control Center - admin@localhost:8860

Items of type itemMapping

ATG CA editable mapping for base search c
ATG CA editable mapping for search config:
ATG CA viewable mapping for base search
ATG CA viewable mapping for search confi
ATG Search Merchandising create mapping
ATG Search Merchandising create mapping
ATG Search Merchandising editable mappin
ATG Search Merchandising editable mappin
ATG Search Merchandising editable mappin
ATG Search Merchandising editable mappin
ATG Search Merchandising editable mappin
ATG Search Merchandising read-only mapp
ATG Search Merchandising r
ATG Search Merchandising r
ATG Search Merchandising r
ATG Search Merchandising r
AssetManager conflict itemM
AssetManager default creatio
AssetManager default editabl
AssetManager default read-c
AssetManager diff itemMappi
AssetManager editable mapp
AssetManager editable mapp

attributes	
description	test
formHandler	
id	100103
isReadOnly	
itemName	vemie
itemPath	/dynamusic/EventsRepository
mode *	edit
name	*
viewMappings	

attributes

Key	Value

New Item

Items of type attributeValue

Press the "List" button to display the specified items.

Currently chosen item:

New Item

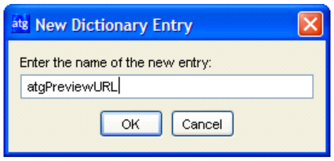
Item Type: attributeValue

New "attributeValue" Values

id	1600033
value	/dynamusic/venueDetails.jsp?itemId=$id

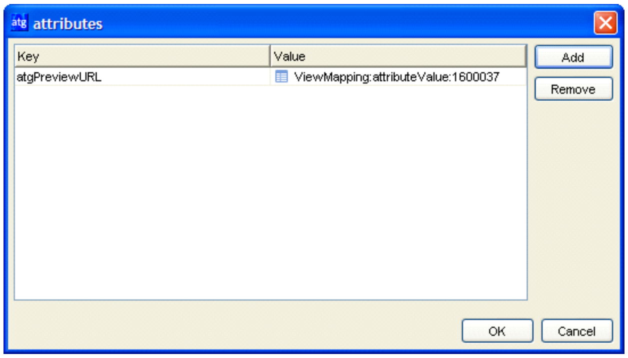
10. Click OK.

11. In the New Dictionary Entry dialog, enter `atgPreviewURL` as the dictionary key, then click OK:



Enter the name of the new entry: atgPreviewURL

The Attributes dialog redisplays with the following settings:



Key	Value
atgPreviewURL	ViewMapping:attributeValue:1600037

12. Click OK to close the Attributes dialog.
13. In the New Item dialog, click OK to save the new `itemMapping`:

In the ATG Business Control Center, reload the asset's editing page and verify that a Preview button appears.

Using Multiple Preview URLs for the Same Asset

If an `itemMapping` has multiple `itemViewMappings`, you can specify a unique preview URL for each one. For example, an `itemMapping` might specify two `itemViewMappings`—one in English, and the second in Spanish. In this case, the preview URL can be language-specific for each `itemViewMapping`.

Suppressing Preview

If you configure your environment to make preview generally available, you can suppress it for a specific asset type through the view mapping system. Create an item mapping for that asset type and set its `disablePreview` attribute to `true`:

1. In the ACC, choose Publishing > View Mapping.
2. Click on List in order to list items of type `itemMapping`.
3. Click New Item to create a unique `itemMapping` for this asset type.
4. Provide the required mapping information: `itemName`, `itemPath`, and `mode` (set to `edit`).

   Note: Set the name field to * (asterisk).
5. Edit the attributes property by clicking on . . .

6. In the Attributes dialog, click Add.
7. In the dialog, click New Item.
8. In the New Item dialog, enter `true`.
9. Click OK.
10. In the New Dictionary Entry dialog, enter `disablePreview` as the dictionary key, then click OK. The Attributes dialog redisplays with the new attribute and its setting.
11. Click OK to close the Attributes dialog.
12. In the New Item dialog, click OK to save the new `itemMapping`.

4 ATG Business Control Center Security

The ATG Business Control Center provides various levels of security, which you can use to control access to the entire UI, to specific activities, or to assets managed within it.

- **Login permission.** As described earlier in this manual, anyone who wants to use the ATG Business Control Center needs an ATG internal user profile with specific roles in order to log in. By default, users also need a password for login access. See Login Requirements for more information.
- **Task access.** The tasks that a user can complete for any project within the ATG Business Control Center are determined by the roles assigned to each task in the underlying workflow. For example, you can configure the Deploy task so that it can be completed only by people who have a Web Administrator role. For information on how to configure task access, refer to *Project and Workflow Security* in the *ATG Content Administration Programming Guide*.
- **BCC Home page security.** You can configure the menu items contained in the Operations list on the BCC Home page so that they appear only to specific users, or to users whose profile is associated with a given role or organization. See Configuring BCC Home Page Security for more information.
- **Asset security.** You can provide different levels of access to specific assets. Assets include folders as well as individual items such as user segments and content targeters. For example, you can allow users with the role Manager to view any targeter in a particular folder, but you can give Write and Delete access to that folder only to users who have a Marketer role. For information, see Configuring Access Rights for Assets.

Configuring BCC Home Page Security

Security access to different parts of the ATG Business Control Center can be managed at a high level through permissions that you set on the options that appear in the BCC Home page. All ATG Business Control Center users require several roles, as described earlier in this manual. You can use those roles to control whether specific options appear in the Operations list.

BCC Home page security is configured through `<generic-activities>` tags in XML definition files for each application that uses the ATG Business Control Center. The tags contain access control list (ACL) entries specifying the roles that can access each option in the Operations list. For detailed information, see *Access to Generic Activity* in the *ATG Content Administration Programming Guide*.

Note also that some BCC Home page links cause the project creation page to be displayed. These links are visible if the user has permission to start the project workflow associated with the links.

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, or delete all types of assets. The ATG Business Control Center uses Access Control Lists (ACLs) to manage asset security and determine the operations that a given user can perform on an asset. 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.

The following example shows the default security policy for ATG Personalization assets, which is defined in `<ATG9>\Publishing\base\config\atg\epub\file\publishingFileSecurity.xml`:

```
<!-- fileFolder security attributes -->
  <item-descriptor name="fileFolder">
    <acl-property name="acl"/>

    <descriptor-acl value="
Profile$role$epubSuperAdmin:read,write,create,delete;
Profile$role$epubAdmin:read,write,create,delete;
Profile$role$epubManager:read,write,create,delete;
Profile$role$epubUser:read,write,create,delete;
Admin$role$administrators-group:read,write,create,delete;
Admin$role$everyone-group:read;"/>
  </item-descriptor>

  <!-- targeter security attributes -->
  <item-descriptor name="targeter">
    <acl-property name="acl"/>

    <descriptor-acl value="
Profile$role$epubSuperAdmin:read,write,create,delete;
Profile$role$epubAdmin:read,write,create,delete;
Profile$role$epubManager:read,write,create,delete;
Profile$role$epubUser:read,write,create,delete;
Admin$role$administrators-group:read,write,create,delete;
Admin$role$everyone-group:read;"/>
  </item-descriptor>

  <!-- profileGroup security attributes -->
  <item-descriptor name="profileGroup">
```

```
    <acl-property name="acl"/>

    <descriptor-acl value="
Profile$role$epubSuperAdmin:read,write,create,delete;
Profile$role$epubAdmin:read,write,create,delete;
Profile$role$epubManager:read,write,create,delete;
Profile$role$epubUser:read,write,create,delete;
Admin$role$administrators-group:read,write,create,delete;
Admin$role$everyone-group:read;"/>
  </item-descriptor>

  <!-- contentGroup security attributes -->
  <item-descriptor name="contentGroup">
    <acl-property name="acl"/>

    <descriptor-acl value="
Profile$role$epubSuperAdmin:read,write,create,delete;
Profile$role$epubAdmin:read,write,create,delete;
Profile$role$epubManager:read,write,create,delete;
Profile$role$epubUser:read,write,create,delete;
Admin$role$administrators-group:read,write,create,delete;
Admin$role$everyone-group:read;"/>
  </item-descriptor>
```

You can set access rights for a role or an organization, in which case the rights are inherited by any user who is assigned to that role or organization (as shown in the example above). You can also set access rights for individual user profiles. When a user attempts to work with an asset in the ATG Business Control Center, the secured repository is checked for access rights on the item the user is trying to manipulate.

As well as setting the default access rights for an asset type through the repository definition file, you can define access rights for individual assets through the Security tab in the ATG Business Control Center. These rights replace the default security policy for that asset. This type of access control can be implemented for the following assets:

- Individual user segments, content groups, and content targeters. For example, assume you have a content targeter that shows content to all users who have registered at your sites within the last month. You can set security access on that targeter so that it can be deleted or modified only by someone with a Marketer role.
- Folders that contain user segments, content groups, or targeters. Controlling access to a folder gives you an efficient way to manage access for all items in that folder. For example, you could have a folder called New Members that contains all the content targeters set up for new site members. Then you could set access rights for that folder so that it can be viewed only by users with a specific role. Note, however, that users may still be able to view an asset contained in a folder by querying for it directly through a search field.

If your Web sites use ATG Merchandising, you can define access control for additional asset types, for example products, SKUs, and pricelists.

You can specify the following access rights for assets:

Access right	Description
Create	Controls whether a user can create new instances of an asset type. This right cannot be set through the Security tab in the ATG Business Control Center. You can set it only by modifying the ACL in the definition file for the secured repository used to store the assets. 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 tab. 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 Business Control Center terms, this means that the user can add the asset to a project for the purposes of deleting it from the system.
View Owner	Corresponds to READ-OWNER in the Access Control List. Not currently used in the ATG Business Control Center UI.
Set Owner	Corresponds to WRITE-OWNER in the Access Control List. Not currently used in the ATG Business Control Center UI.
View Access Rights	Controls the ability to view the access control list for this item. If a user does not have this access right, the Security tab does not appear. (This access right is automatically granted to the owner of the item.) Corresponds to READ-ACL in the Access Control List.
Set Access Rights	Controls the ability to change the access rights for this item through the Security tab. (This access right is automatically granted to the owner of the item.) Corresponds to WRITE-ACL in the Access Control List.

Setting Access Rights in the ATG Business Control Center

The procedure shown here describes how to set access rights for a new user segment folder. The steps are similar for any of the asset types listed in the previous section.

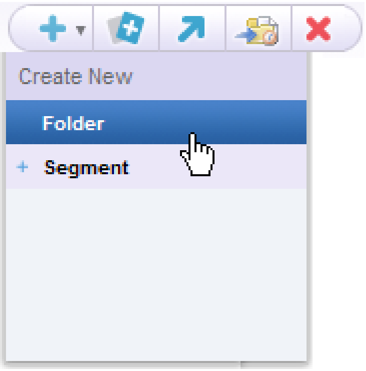
Important: Keep the following points in mind before you change access rights:

- To be able to change access rights, your own user profile must have Write, View Access Rights, and Set Access Rights privileges for the asset you are changing. Otherwise, the Security tab does not appear for the asset (see the procedure below).

- In ATG 10.0.2, some parts of the Asset Manager framework use JavaServer Page (JSP) technology and others, including ATG Merchandising, are built in Flex. The information in this section applies only to the JSP-based framework. The user interface for Flex-based applications does not include a Security tab. Change access rights for ATG Merchandising assets, including Promotions and Coupons, with ACLs, as described earlier in this chapter. To learn which access rights users must have to view and work with assets in ATG Merchandising, see *Configuring Access Rights for Assets* in the *ATG Merchandising Administration Guide*.

To set access rights for a user segment folder:

1. On the BCC Home page, select Personalization > Targeting and Segmentation.
2. Create a project that you will use to deploy the folder after you have set its access rights. (You can add the folder to an existing project if you prefer.) For detailed information on creating projects, see the *ATG Business Control Center User's Guide*.
3. Select User Segments from the Show list. Any user segments and folders that already exist in your system appear in the left pane.
4. If you want to edit access rights for an existing folder, proceed to step 5. If you want to create a folder:
   - Click the Create New button, as shown in the image, and select Folder.

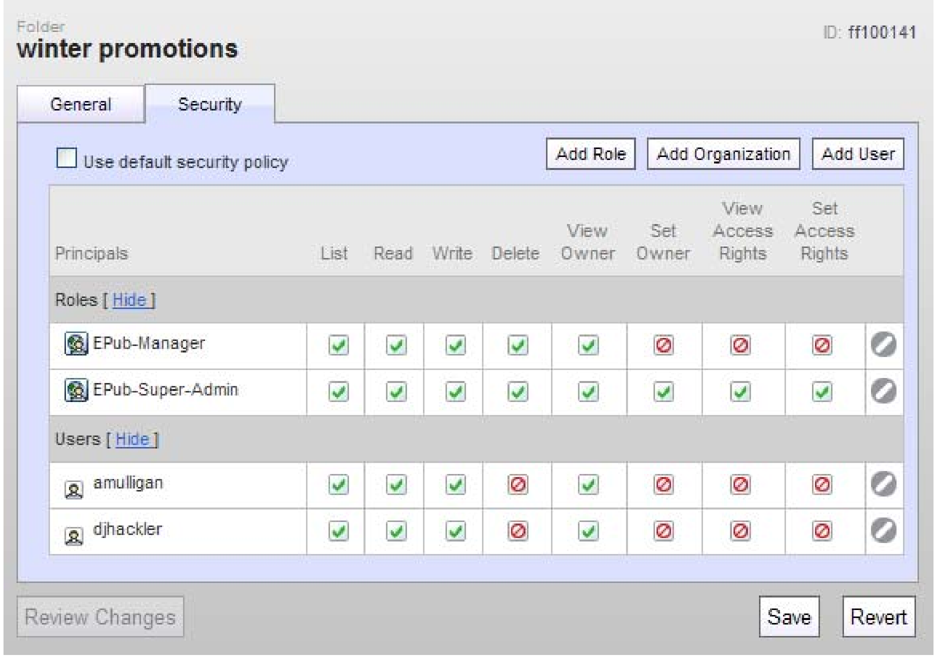


   - Specify any required properties for the folder (by default, the Folder Name property).
   - Click Create.
   - The folder is created, and a Security tab appears next to the General tab. (Note that the Security tab does not appear for any asset until the asset has been created.)
5. Display the Security tab. The default access rights for this folder appear.
6. 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 folder, you are replacing the default rights.

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

- Add Role: Allows you to define access rights to this folder 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 folder 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 folder 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]								
EPub-Manager	✓	✓	✓	✓	✓	⊘	⊘	⊘
EPub-Super-Admin	✓	✓	✓	✓	✓	✓	✓	✓
Users [Hide]								
amulligan	✓	✓	✓	⊘	✓	⊘	⊘	⊘
djhackler	✓	✓	✓	⊘	✓	⊘	⊘	⊘

7. Click the appropriate button and use the asset picker that appears 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 item, including your own profile if necessary.

8. 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.

9. 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.

10. Complete the project so that the folder is deployed. For detailed information, see the see the *ATG Business Control Center User's Guide*.

5 Managing User Profiles

Profile repositories are a fundamental ATG concept that is discussed in detail in the *ATG Personalization Programming Guide*. Briefly, a profile contains properties that describe the characteristics of your ATG application users. There are two kinds of profile properties: *explicit* and *implicit*. Web site users (for example, the customers of a commerce site) provide explicit properties themselves, typically by filling out a registration form or editing a Preferences page. Examples of explicit properties are a person's name and address. Implicit properties are information that the system gathers about visitors by tracking their behavior at your sites. A list of recently browsed Web pages is an example of an implicit property.

Keeping track of profile properties makes it possible for a Web application to maintain persistent information about users and to deliver personalized content to them accordingly. For example, you could have an option on your registration form that asks visitors to indicate their income range, which the system stores as a profile property. You could then set up your site content so that you display different items according to the income range of each site visitor.

The profiles of site visitors or customers are **external user profiles**. However, anyone who uses ATG applications to manage activities related to your customer-facing sites also has a user profile. For example, marketers who are responsible for creating user segments or other targeting assets must have a profile in order to access the targeting interface in the ATG Business Control Center. The profiles of ATG application users are **internal user profiles**. Unlike external profiles, internal profiles are not created or updated automatically. Instead, an administrator creates and manages them manually through the ATG Business Control Center.

If an ATG installation uses preview features, the environment can also include **preview profiles**, which are used to test how content will appear to a sample user.

For information on how to configure the repositories used to store user profiles, including an architecture diagram, see Configuring Profile Repositories earlier in this guide.

The procedures for creating and managing user profiles through the ATG Business Control Center are essentially the same for all three types of profile. This chapter describes these procedures.

Understanding Profile Sub-Types

It is possible to have more than one type of user profile in the same profile repository. You can create a base set of profile properties for every user, and then create sub-types that include additional properties unique to each subset. For example, you could create a base set of profile properties that are common to all internal users, and then add two sub-types, “administrator” and “marketer.” Each sub-type has its own distinct properties in addition to the common ones.

These sub-types are called “profile types,” and you can use them in most places in the ATG Business Control Center where you are required to define groups of users; for example, you can use them as filters in content targeter rules.

For more information, refer to the *ATG Personalization Programming Guide*.

Displaying or Hiding Profile Properties

The ATG Business Control Center is configured so that not all properties in the default profile repository appear in the UI. The display is controlled by settings in the View Mapping repository. You can add or hide specific properties by editing the default view mapping settings. For instructions, see *Specifying Properties to Display* in the *ATG Merchandising User Guide*.

Viewing User Profiles

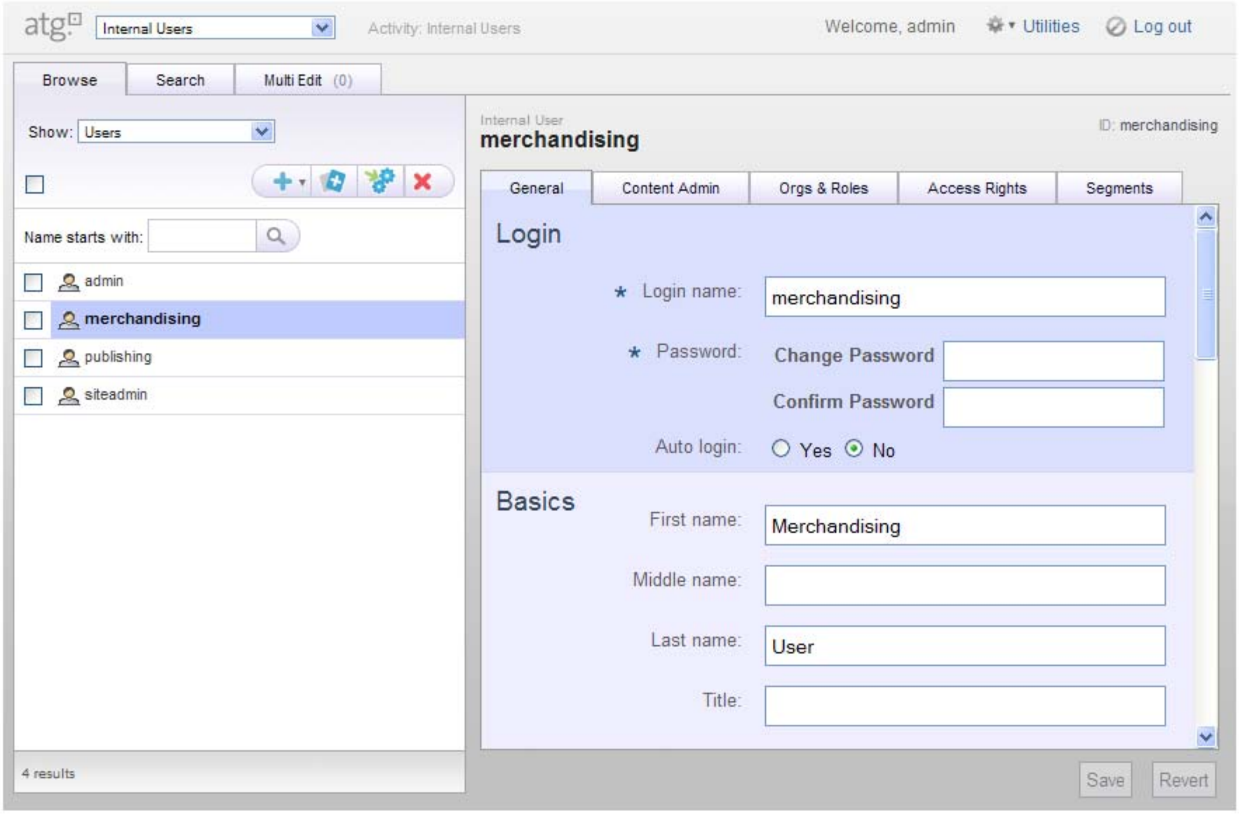
To view the user profiles in your system, complete the following steps:

1. Display the BCC Home page.
2. Expand the Personalization item and select the appropriate Users option. **Caution:** If you choose External Users, be aware that you will be viewing live profiles on your production server.
3. Select the Users option from the Show menu.

Important: If your product suite includes other ATG applications that add properties to external user profiles, such as ATG Outreach or ATG Service applications, and you cannot view all these properties through the ATG Business Control Center, ensure that the correct application modules are running and that your external profile definition files are correctly located. For more information, see Viewing External Profile Properties Added by Other Applications.

If changes made to a profile on one ATG instance do not appear in the ATG Business Control Center on another instance, make sure that caching is set up appropriately for profiles. See Configuring Caching for Viewing Profile Changes for more information.

The following image shows the Users interface with some sample data:



★ Login name: merchandising

★ Password: Change Password

Confirm Password

Auto login: ○ Yes ● No

First name: Merchandising

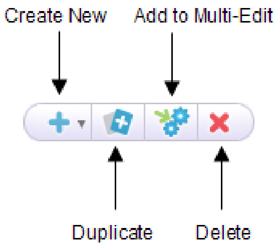
Middle name:

Last name: User

Title:

Elements in the Users interface are described below:

- The Activity field corresponds to the Users option selected on the BCC Home page (in this case, Internal Users). It indicates the profile repository whose contents are displayed below.
- The Browse tab displays a list of all existing items of the type indicated by the Show menu (users, organizations, or roles). In this case, the tab displays all internal user profiles. By default, profiles are displayed and sorted by login name, but an environment can be configured to display profiles according to another property, such as last name or email address.
- The Search tab allows you to locate a specific profile or group of profiles. See the *ATG Business Control Center User's Guide* for information on how to use it.
- The Multi Edit tab gives you several ways to edit large numbers of profiles. See the *ATG Business Control Center User's Guide* for more information.
- The icons above the list of profiles allow you to do the following:
  - **Create a New** profile.
  - **Duplicate** the selected profile(s).
  - **Add** the selected profile(s) to the **Multi Edit** tab for batch editing.
  - **Delete** the selected profile(s) See Deleting Profiles.



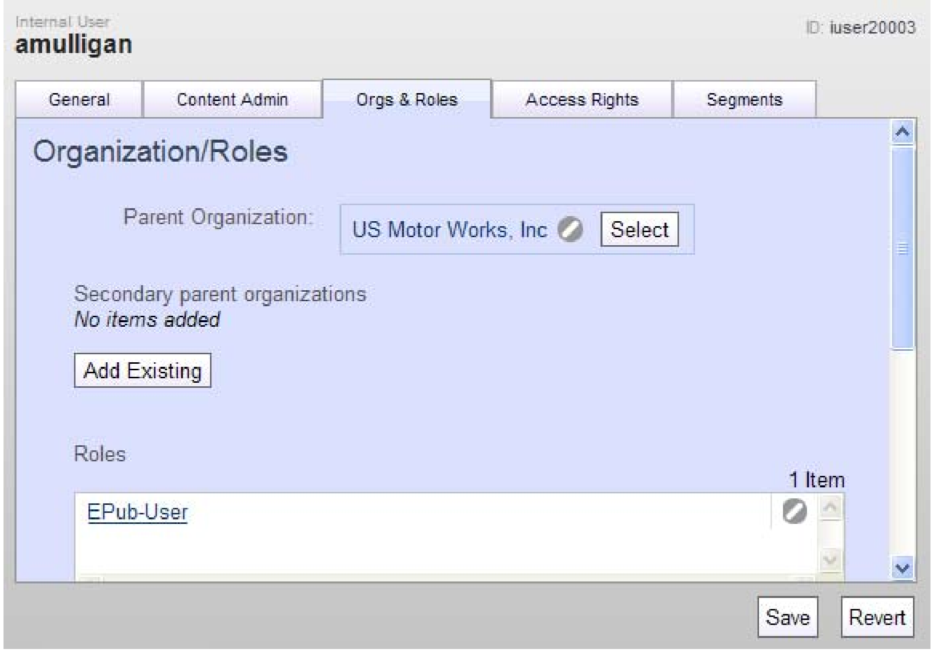
In all cases except Create New, you use the checkboxes in the Browse list to select the profiles to which you want the icon action to apply. Use the checkbox next to the icons to unselect all checked profiles.

- The Name Starts With field lets you narrow the display of profiles to ones that start with specific characters. Type the character or characters you want and then click Go. In the example above, entering "c" would narrow the list of profiles to two items, `carlaslattery` and `celiaJ`. To restore the full list, empty the Name Starts With field and click Go again. For more advanced queries, use the Search tab.
- The General tab contains fields where you can view or edit the current values of basic properties for this profile. The properties vary depending on the definition of your profile repository. A value is required for any property with an asterisk (in the example, the `Login Name` and `Password` properties). All other values are optional. For more information, see Creating Profiles.
- The Content Admin tab contains settings that control the display of items in the ATG Content Administration areas of the ATG Business Control Center.
- The Access Rights tab is used by ATG Service applications to configure security for options in the ATG Service UI. For more information, refer to the ATG Service documentation. The tab does not appear if you are viewing external or preview profiles.
- The Orgs and Roles tab shows the organizations this profile belongs to and any roles that have been assigned to it. See Using the Orgs and Roles Tab.
- The Segments tab shows the user segments that have been created for your system and indicates which ones the current profile belongs to. See Using the Segments Tab.
- The Save and Revert buttons allow you to preserve or undo any changes you make to this profile. If you click Revert, the profile is restored to the last saved version. **Note:** When you start creating a new profile, a Create button appears instead of Save.

Using the Orgs and Roles Tab

The Orgs and Roles tab displays the organizations and roles that have been assigned to the selected profile and allows you to assign new ones or edit the current assignments.

The images below shows the Orgs and Roles tab with some sample data. You may have to scroll down to see the area shown in the second image.



Internal User

amulligan

ID: iuser20003

Organization/Roles

Parent Organization: US Motor Works, Inc

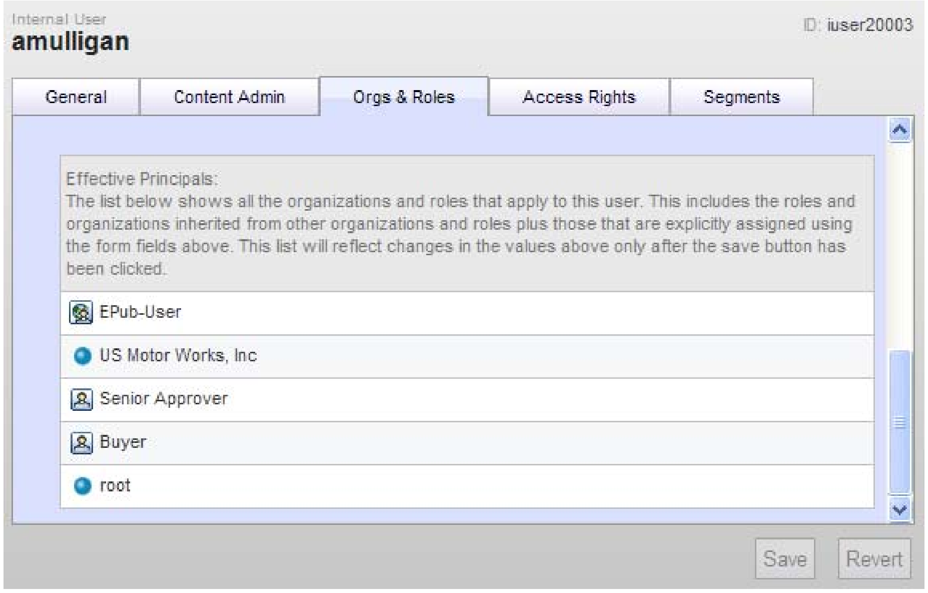
Secondary parent organizations

No items added

Roles

1 Item

EPub-User



Internal User

amulligan

ID: iuser20003

Effective Principals:
The list below shows all the organizations and roles that apply to this user. This includes the roles and organizations inherited from other organizations and roles plus those that are explicitly assigned using the form fields above. This list will reflect changes in the values above only after the save button has been clicked.

- EPub-User
- US Motor Works, Inc
- Senior Approver
- Buyer
- root

The Parent Organization field shows the organization to which this user is directly assigned. To assign a parent organization, click Select and use the asset picker to select the item you want.

If you are viewing internal profiles, a Secondary Parent Organizations field also appears. Secondary parent organizations are used by ATG Service applications to implement security. For more information, see the ATG Service documentation.

The Roles list shows the global and organizational roles that have been directly assigned to this user. To assign a role, click Add Existing and use the asset picker to select the role. For organizational roles, you can choose only those items that are accessible to the user's parent organization. (As explained later in this guide, organizational roles must be created for a specific organization. See Creating Organizations.)

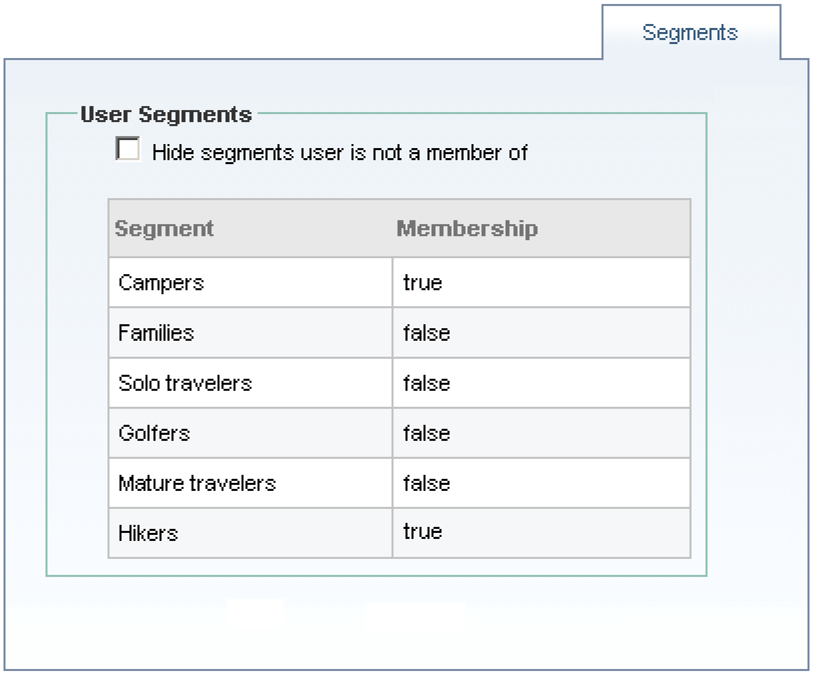
"Principals" are the entities that make up a user directory. Users, organizations, and roles are examples of user directory principals. The Effective Principals list shows all organizations and roles that have been assigned to this user, either directly or through inheritance. In the example above, the user has inherited membership in two organizations through the organization to which he is directly assigned.

If the user is a member of any segments, these also appear in the Effective Principals list, identified by the icon shown below:

my segment

Using the Segments Tab

The Segments tab lists the user segments that have been created for your environment and shows the ones to which the selected profile belongs.



User Segments

□ Hide segments user is not a member of

Segment	Membership
Campers	true
Families	false
Solo travelers	false
Golfers	false
Mature travelers	false
Hikers	true

Segment membership is determined automatically by the system, so you cannot edit the information in this tab. When you add a new segment, a property with the same name as the segment is added to each profile. If a profile meets the segment's rule criteria, the property is automatically set to true for that profile. For example, assume you create a segment called Campers that is configured to include all profiles whose `interest` property includes “camping.” A property called `Campers` is added to each profile, and the system sets that property to true for any profile whose `interests` property includes the value “camping.”

Unless you are running ATG Service applications, segments usually only apply to external user profiles. For information on creating user segments, see the *ATG Business Control Center User’s Guide*.

Creating Profiles

Profiles for external users (typically site visitors and customers) are usually created automatically by the system. However, you need to create profiles manually for internal users and preview users.

When you create a profile, you specify values for the properties defined for the user item descriptor in the profile repository. A default set of profile properties is provided with the ATG Business Control Center. If they do not meet your business requirements, you can extend the profile definition to include different or additional properties. For information on how to do this, refer to the *ATG Personalization Programming Guide*.

To create a profile, complete the following steps:

1. Display the BCC Home page.
2. Expand the Personalization option and select Internal Users.
3. Click the Create New icon and select the type of profile you want to create, typically User. (If your environment includes profile sub-types, the base profile type and sub-types appear as options. See Understanding Profile Sub-Types for more information.)
4. Enter values for any required properties, which are shown with an asterisk. For most environments, a login name and password are required.

   Login Name: Login names must be unique within each profile repository. You can enter any combination of letters, numbers, and special characters, including spaces. For ATG Business Control Center users, the value you specify here is the value the user specifies in the Username field when he or she logs in. Typically, customer-facing Web sites include a form that allows users to change their login names.

   Login names are **case sensitive** for all databases except MS SQL. Use caution when creating profiles on MS SQL so that you do not create the same profile twice with different capitalization.

   Password: The default profile repository requires each profile to have a password, so you must enter one when you create a profile. The password can contain up to 32 characters. ATG Business Control Center users enter the value of this property in the Password field when they log in.

   Passwords are **case sensitive** for all databases except MS SQL.

   The password appears as asterisks when you type it. The next time you or anyone else displays this screen, the property will be blank. You cannot view the value, but if necessary you can change the password by entering a new one.

Typically, customer-facing Web sites include a form that allows users to change their passwords.

5. When you have finished specifying property values for all tabs, click Create, which adds the profile to the system. For information on editing specific properties, see Editing Profiles.

Editing Profiles

For external users (typically site visitors and customers) you do not usually have to edit profile property values because the ATG system does this automatically. For example, as part of creating a personalized Web site, you set up sensors or scenarios that track a visitor's behavior at your Web sites and change profile properties accordingly, or you could create a Preferences page that updates the profile with information entered in a form. However, there may be cases where you need to edit these profiles. You may have to make frequent edits to the profiles of ATG Business Control Center users or preview users.

To edit a profile, complete the following procedure.

1. Display the BCC Home page.
2. Expand the Personalization item and select the appropriate Users option. **Caution:** If you choose the External Users option, be aware that you will be editing live profiles on your production server. See also Viewing Changes Made to External Profiles below.
3. Find and display the profile that contains the value you want to change. See the *ATG Business Control Center User's Guide* for information on using the Search tab to locate a specific profile.
4. Change the value as needed.
5. Click Save.

For more information on editing specific types of property or on editing large numbers of profiles, see the *ATG Business Control Center User's Guide*.

For information on how to add new properties to all profiles, see the *ATG Personalization Programming Guide*.

Viewing Changes Made to External Profiles

In some circumstances, changes made to an external profile on the production server may not appear immediately in the ATG Business Control Center on the asset management server, and vice versa. For example, a customer could change his or her address on the production server, and the new address would not appear immediately in the profile UI on the asset management server. The delay occurs because of profile item caching – until the item cache expires on the server where the changes were **not** made, the changes do not appear on that server.

If necessary, use the ATG Dynamo Server Admin to flush the item cache. See the *ATG Installation and Configuration Guide* for more information.

Deleting Profiles

To delete a user profile, select the checkbox next to the profile you want to delete, and click the Delete icon. This action removes the profile from the repository. It cannot be undone.

Important: Deleting profiles is not recommended. In particular, avoid deleting profiles for any users who have created or edited items in the ATG Business Control Center. The system tracks asset ownership, so deleting the profile of someone who created a project or an asset can result in errors. Deleting the profiles of any logged-in users (including your own profile) can cause data corruption.

6 Creating Organizations and Roles

In addition to setting up profiles for individual ATG Business Control Center users, you can set up additional profiles for abstract entities called *organizations* and *roles* and use them to create a multi-level organization of site users grouped by function.

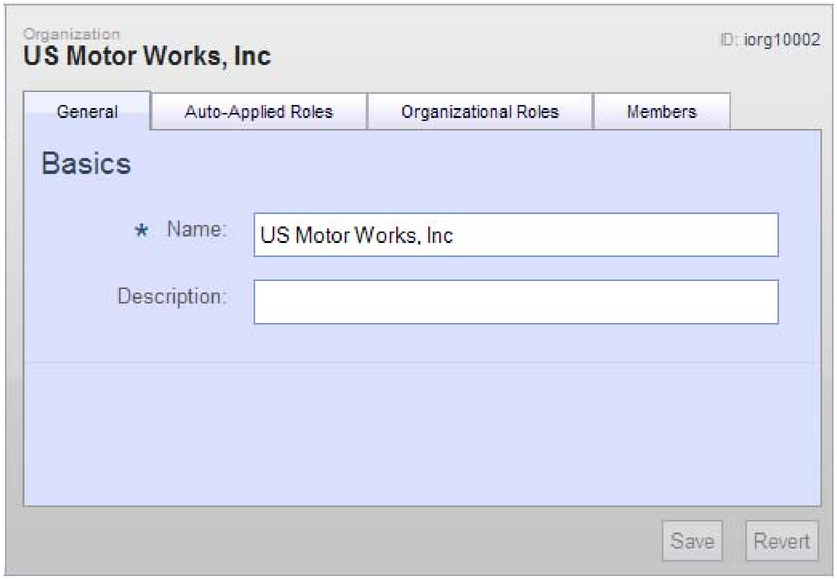
Some Web sites require only a one-level or flat organizational structure for their users. For example, a large retail site may have many different types of visitor (new customers, frequent buyers, customers with a high income, residents of a specific area), and those visitors may have dramatically different needs in terms of the content of the site. Regardless of their different requirements, however, they perform essentially the same role – retail customer – at the site. For a site such as this, the organizational structure of users requires one level only.

A profile repository configuration that includes organizations and roles is useful for Web sites that serve a variety of users with widely differing functions. For example, a business-to-business commerce site could have some users who are buyers and others who have the supervisor role of approving purchasing decisions. In addition, other users may act as administrators for the site. In this case, the site's organizational structure could have two or possibly three levels. Other examples of sites that could involve multi-level user relationships are sites that manage interactions between a business and its partners, or sites that function as an Internet community or discussion forum.

In addition, organizations and roles are also often used in combination with user profiles to configure secure access to different parts of a site or an ATG application. Some ATG applications require you to set up organizations and roles for this purpose. The ATG Business Control Center, for example, uses roles to manage access permission for different areas of its UI. For information on using organizations and roles for security in ATG applications, refer to the documentation of the appropriate application. For information on using roles to control access to the ATG Business Control Center, see Configuring BCC Home Page Security.

How Organizations and Roles Work

Organizations correspond to the different ways you can group users. For example, you could set up organizations that correspond to the different business units in a company, or to different geographic areas. Like users, each organization has its own profile in the profile repository. The following example shows part of the profile for an organization called US Motor Works:



US Motor Works, Inc

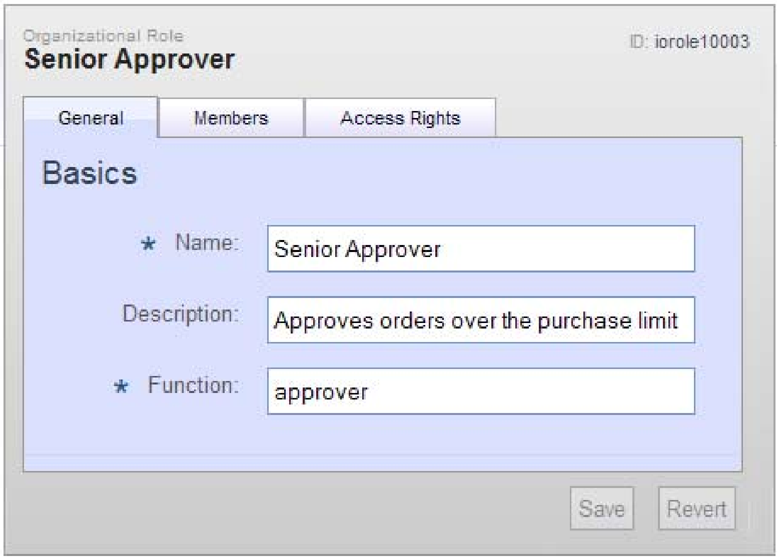
Basics

* Name: US Motor Works, Inc

Description:

Organization profile

Roles correspond to specific functions that a person can perform within an organization, such as "vice president" or "administrator." Like users and organizations, each role has its own profile in the repository. The following example shows the profile of a role called "Senior Approver":



Senior Approver

Basics

* Name: Senior Approver

Description: Approves orders over the purchase limit

* Function: approver

Role profile

(Note that there are two types of role—global and organizational. The example above is an organizational role. See the section on roles for descriptions of both types.)

After you have set up organizations and roles, you assign users to them. Users then inherit the profile properties of the assigned organization or role. In the same way, you can assign roles to organizations; in this case, any user who is a member of an organization inherits any roles you have assigned to it. You can also create a hierarchy by assigning organizations to other organizations; for example, you could assign two organizations called "North West" and "South West" to an organization called "Western Region." Each child organization inherits the profile properties of the parent organization.

As mentioned at the beginning of this chapter, you can use a combination of users, organizations, and roles (also called a user directory) to control access to various areas of the site. You can also use them as you would use segments – you can set up scenarios and targeters that deliver appropriate content to members of a given organization, for example, or to users who have a specific role. (For more information on scenarios, see *ATG Personalization Guide for Business Users*. For more information on segments, see the *ATG Business Control Center User's Guide*.)

For information on how to use organizations and roles for site security and access control, refer to the *ATG Programming Guide*.

Organization Hierarchy

Organizations are configured in a hierarchy, with child organizations inheriting properties from the organizations above them (the parent organizations). In addition, parent organizations inherit members from child organizations—if you have a top-level organization called Head Office, and a child called Sales, any user who is a member of Sales is also a member of Head Office. Any roles assigned to Head Office can therefore be assigned to members of Sales.

When you create an organization, you are required to specify the position in the hierarchy where you want the new organization to be located. All organizations must be located under the root organization, which is the top-level organization in any system. Note that the default root organization is a usable organization like any other. It cannot be deleted, but it can be renamed if desired.

Global and Organizational Roles

As mentioned above, an ATG environment can include two types of role—global and organizational. This section describes the roles and explains the differences between them.

Global Roles

Global roles represent any type of generic or specific function or activity that you choose, for example "admin," "sales rep," or "webmaster." You can do the following with a global role:

- Assign it to any organization as an auto-applied role. Users then inherit the properties of that role as additions to their own profiles.
- Group similar global roles together into folders to help you manage them more easily.

Organizational Roles

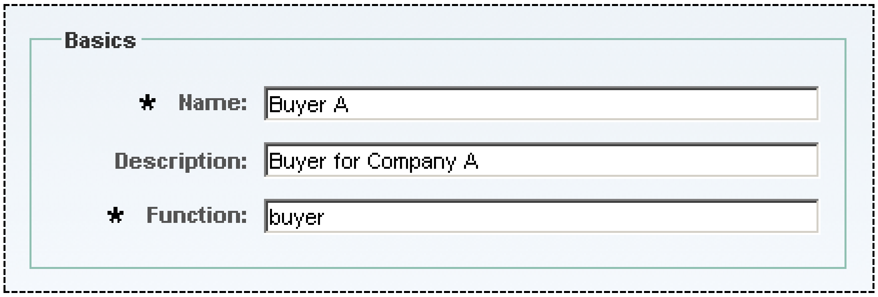
Organizational roles are specific to a particular organization. You can do the following with an organizational role:

- Assign it as an auto-applied role to the organization under which you created it. (Note that you cannot assign an organizational role that you created for Company Y to Company Z.) The role is then automatically assigned to all users who belong to that organization.
- Assign it to any individual user who is a member of the organization to which the role belongs. In this case, other members of the organization do not automatically inherit the role's properties.

Organizational roles contain a property called `function` that you can use as a way of identifying similar roles in different organizations. Suppose you have a business-to-business commerce site. The companies with whom you do business are defined as organizations in the ATG Business Control Center—you create an organization called Company A and another called Company B. Each company has buyers who are responsible for buying products from your site. You decide that you want to offer different prices to each buyer depending on his or her organization; for example, you want buyers from Company A to have a 15% discount on any backordered item, but you do not want to offer this discount to buyers from Company B.

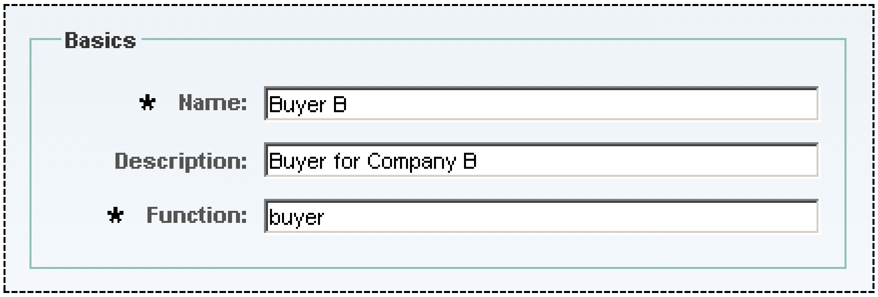
You set up two organizational roles, Buyer A and Buyer B, and assign one to each organization. You can then design your pricing structures and scenarios to differentiate between the two, offering distinct discounts to each role. However, in other ways the two roles are very similar, and you decide you want to establish a connection between them. You specify "buyer" as the Function property for each role—this property acts as a sort of keyword and allows the ATG system to keep track of the fact that the roles are related. You could then write custom applications that use this property in a variety of ways; for example, they could create custom scenario events that are triggered by members of any organizational role with a given function value. (See the *ATG Personalization Guide for Business Users* for information on creating scenarios.)

The following images show the profiles you set up for the two organizational roles. Note that the setting of the Function field is the same for each one.



Basics

* Name:	Buyer A
Description:	Buyer for Company A
* Function:	buyer



Basics

* Name:	Buyer B
Description:	Buyer for Company B
* Function:	buyer

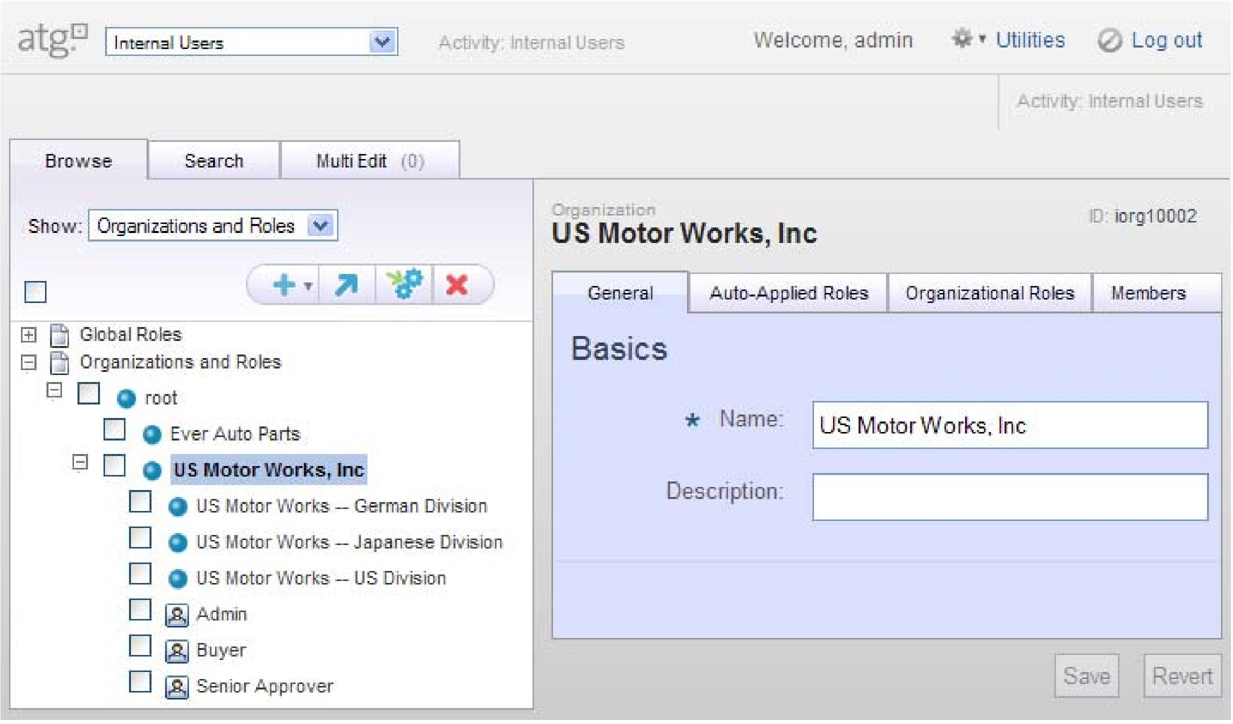
Note: The value of the `function` property must be unique within an organization. For example, you cannot create two organizational roles for the same organization and give them both the function "supervisor."

The Organizations and Roles Interface

You use the organizations and roles interface to view and create organizations and roles and to assign user profiles to them. This section provides an overview of the interface and describes key elements.

(**Note:** For organizations, you can use either the Organizations and Roles interface shown here or the Organizations interface, which shows only organizations. See Viewing Organizations for more information.)

The following image shows the panels and other items that make up the Organizations and Roles interface:



Organization

US Motor Works, Inc

ID: iorg10002

* Name: US Motor Works, Inc

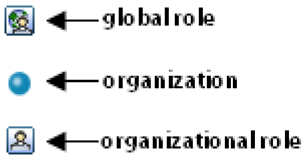
Description:

- The Browse tab displays an expandable tree of folders that contain all existing items of the type indicated by the Show menu. In this case, the tab displays all organizations and roles in the profile repository.

  Three types of item can appear in this display:

  - Organizations
  - Global roles
  - Organizational roles

  These items are identified by the following icons:

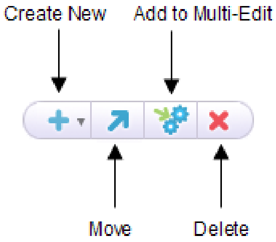
  

  See the previous section for information on the differences between global and organizational roles.

- The Search tab allows you to locate specific organizations or roles. See the *ATG Business Control Center User's Guide* for information on how to use it.

- The Multi Edit tab gives you several ways to edit large numbers of organizations or roles. See the *ATG Business Control Center User's Guide* for more information.

- The icons above the list of organizations and roles allow you to do the following:
    - **Create a New** item.
    - **Move** the selected item(s) to a different position in the tree.
    - **Add** the selected item(s) to the **Multi Edit** tab for batch editing.
    - **Delete** the selected item(s).



  In all cases except Create New, you use the checkboxes in the Browse tab to select the items to which you want the icon action to apply. Use the checkbox in the icon toolbar to unselect all checked items.

- The tabs that appear in the right pane vary according to the type of item selected in the list. For information, see the section that describes how to create an item of that type.
- The Save and Revert buttons allow you to preserve or undo any changes you make to this item. **Note:** When you start creating a new organization or role, Create and Cancel buttons appear instead of Save and Revert.

Viewing Organizations

There are two ways to view organizations in the ATG Business Control Center: through the Organizations and Roles interface, which is shown in the previous section, or through the Organizations interface.

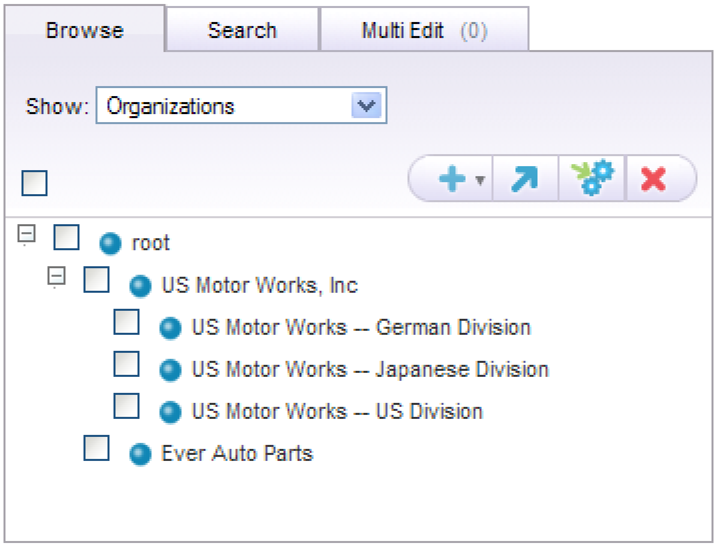
- The **Organizations and Roles** interface shows all organizations in the selected profile repository as well as all roles, which is useful if you want to see the relationships between these items in one display.
- The **Organizations** interface shows only organizations, which is helpful if your environment includes a large number of roles and you prefer not to display these while you are editing organizations.

To view organizations, complete the following steps:

1. Display the BCC Home page.
2. Expand the Personalization item and select the appropriate Users option. **Caution:** If you choose External Users, be aware that you will be viewing live profiles on your production server.

3. From the Show dropdown menu, select Organizations.

The organizations in the selected profile repository appear in a folder hierarchy in the Browse tab, as shown:



Show: Organizations

- ⊟ □ root
  - ⊟ □ US Motor Works, Inc
    - □ US Motor Works -- German Division
    - □ US Motor Works -- Japanese Division
    - □ US Motor Works -- US Division
  - □ Ever Auto Parts

The properties for the selected organization appear in tabs in the right pane:

- The **General** tab shows basic information such as the name and description.
- The **Auto-Applied Roles** tab shows any roles that are automatically applied to any user who is a member of this organization. You also assign auto-applied roles here. See Using the Auto-Applied Roles Tab.
- The **Organizational Roles** tab shows any organizational roles that are accessible by this organization. See Using the Organizational Roles Tab for more information.
- The **Members tab** shows the users or child organizations that belong to this organization. You also use this tab to assign users to this organization. See Using the Members Tab.

Creating Organizations

When you create an organization, you specify values for the set of profile properties defined for the organization in your system. A default set of properties is provided with the ATG Business Control Center. If these properties do not meet your business requirements, you can extend the profile (specifically, the organization item descriptor in the profile repository) to include different or additional properties. For information on how to do this, refer to *Working with the Dynamo User Directory* in the *ATG Personalization Programming Guide*. This step is optional.

To create an organization, complete the following steps:

1. Display the BCC Home page.
2. Expand the Personalization item and select the appropriate Users option. **Caution:** If you choose External Users, be aware that you will be viewing live profiles on your production server.
3. From the Show menu, select Organizations (or Organizations and Roles).
4. In the left pane, select the position in the hierarchy where you want the new organization to be located (the parent organization). All organizations must be located under the root organization, either directly under it or below another organization that is a child of the root. **Note:** To select a parent organization, click its name, not the checkbox next to it. The selection is indicated by blue highlighting.
5. Click the Create New icon and select Organization.
6. In the General tab, fill in all required properties, which are shown with an asterisk. By default, the Name property is required. Organization names can include spaces.
7. Click Create. The new organization is added to the selected profile repository and appears in the list on the left. (You may have to refresh the browser window or expand the appropriate folder to see it.)

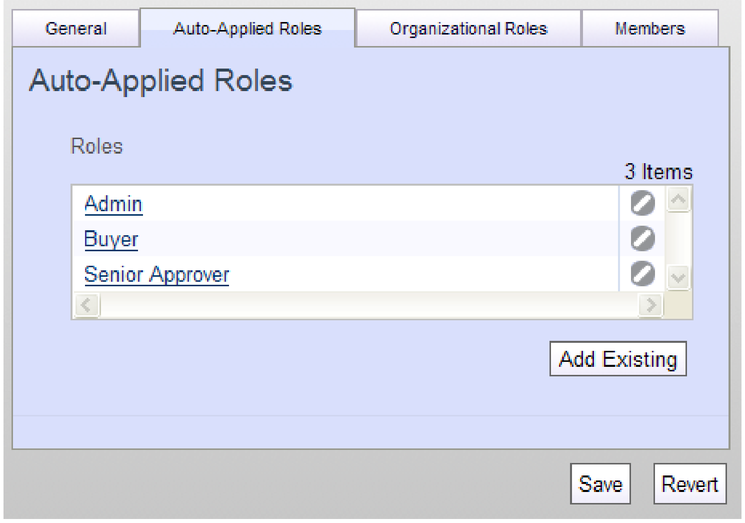
   Note: If you did not select a parent organization in step 4, an error message appears when you click Create. Use the Select Parent button to choose a parent organization.
8. Use the Auto-Applied Roles tab to associate this organization with specific global or organization roles. Users who are members of this organization automatically inherit these roles. Organizations that are children of this organization also inherit them. See Using the Auto-Applied Roles Tab for more information.
9. Use the Organizational Roles tab to associate this organization with specific organizational roles. Users who belong to this organization can optionally be assigned any of these roles. See Using the Organizational Roles Tab for more information.
10. Use the Members tab to assign users to this organization. You can also assign other organizations to it. See Using the Members Tab for more information.
11. Click Save when you have finished making your changes.

Using the Auto-Applied Roles Tab

You can assign roles to organizations so that they are automatically applied to any member of that organization. For example, you could assign a role called “manager” to Organization X as an auto-applied role. Any users you assign to Organization X automatically inherit the manager role.

To assign a role to an organization so that it will be automatically applied to members, display the Auto-Applied Roles tab, click Add Existing, and select the role you want. Then click Save.

The following image shows the Auto-Applied Roles tab with some sample data. The roles admin, supervisor, and approver will be automatically assigned to any user who is a member of this organization (either through direct membership or by inheritance).



You can use both global and organizational roles as auto-assigned roles. You can assign any global role to an organization. However, any organizational role you want to use must be available to the organization to which you want to assign it, either directly or through inheritance.

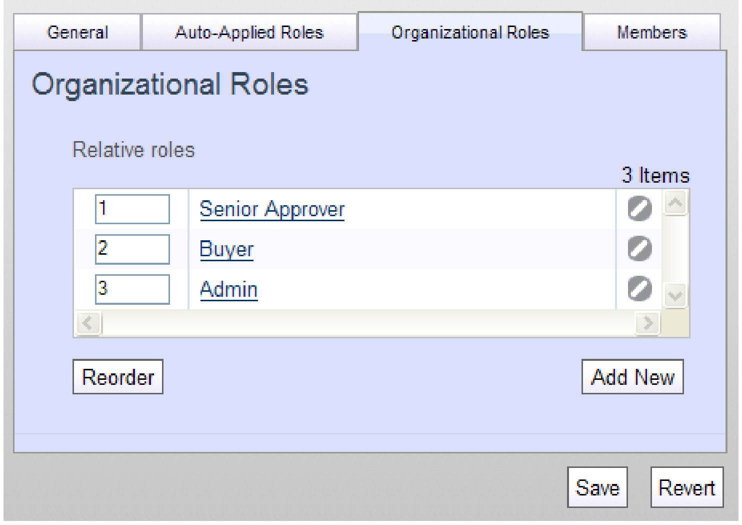
To remove a role from an organization, click the Remove Item icon next to the role you want to delete.

Note: The name of any role you assign here is a link. Clicking it allows you to edit the properties of the given role. Bear in mind that editing a role here is the same as editing it through the roles interface, so any changes you make will apply wherever this role is used.

Using the Organizational Roles Tab

The Organizational Roles tab shows the organizational roles that have been created specifically for the given organization. Any users who are members of the organization can be assigned these roles. The key difference between auto-applied roles and organizational roles is that auto-applied roles are automatically inherited by all members of the organization; in the case of organizational roles, you can choose which ones you assign to each member.

You use the Organizational Roles tab to specify the organizational roles that are available to this organization. The following image shows the tab with some sample data:



Organizational Roles

Relative roles

3 Items

1	Senior Approver
2	Buyer
3	Admin

Note that in the ATG platform, *relative role* is simply another term for *organizational role*.

By default, any organizational roles created under this organization appear automatically in the list. You can remove a role by clicking the Remove Item icon next to the role. **Caution:** Be aware that if you remove a role and click Save, you cannot undo the action.

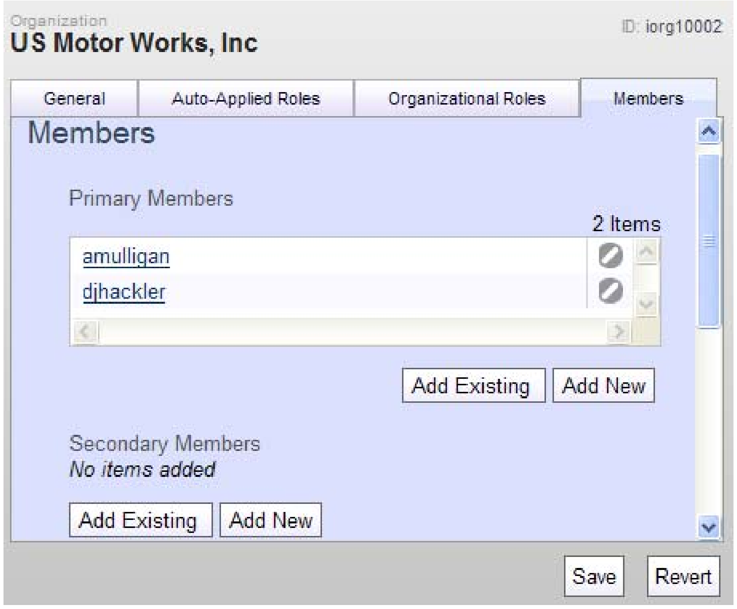
You can also create an organizational role through this tab; to do so, click Add New and define the new role. The role is added to the system just as if you created it through the roles interface. See Creating Roles.

Note: The name of any role you assign here is a link. Clicking it allows you to edit the properties of the given role. Bear in mind that editing a role here is the same as editing it through the roles interface, so any changes you make will apply wherever this role is used.

Using the Members Tab

Use the Members tab to assign users to this organization and to view existing members. (Note that you can also assign users through the Orgs and Roles tab in the Users interface.)

The two images below show the Members tab with some sample data. Note that you have to scroll down to see the All Members display shown in the second image.



Organization
US Motor Works, Inc
ID: iorg10002

Members

Primary Members

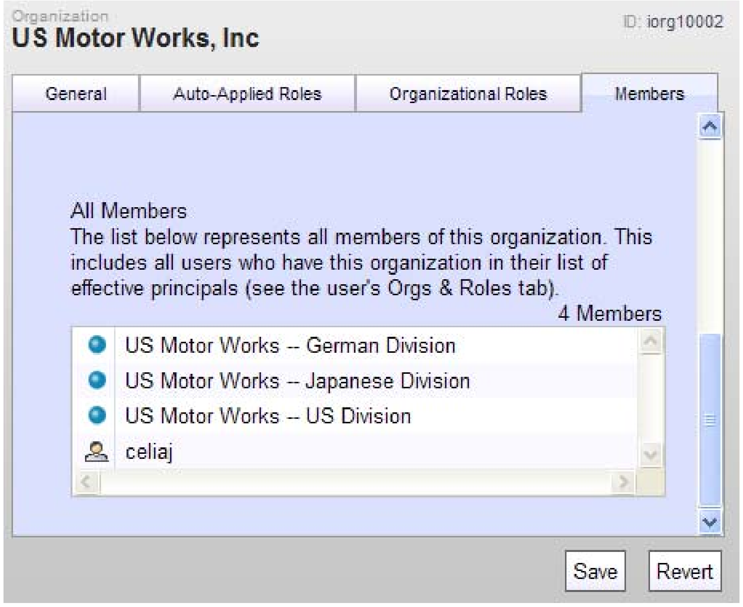
2 Items

amulligan

djhackler

Secondary Members

No items added



Organization
US Motor Works, Inc
ID: iorg10002

All Members

The list below represents all members of this organization. This includes all users who have this organization in their list of effective principals (see the user's Orgs & Roles tab).

4 Members

US Motor Works -- German Division

US Motor Works -- Japanese Division

US Motor Works -- US Division

celiaj

The list shows all members of this organization, which can be any of the following.

- Primary Members: Users who have been assigned membership directly through this tab (`amulligan` and `djhackler`, in the example above).
- Organizations that have been created below this one in the organization hierarchy (in this case, the three US Motor Works divisions).

- Users who have inherited membership from another organization. In the example, the user called `celiaj` is a member of the US Motor Works – US Division organization and therefore inherits membership in the organization shown in this tab.

To assign a member to this organization, click Add Existing and select the user you want. Then click Save. (You can also create a user through this tab; to do so, click Add New and define the new user profile. The profile is added to the system just as if you created it through the Users interface.) Clicking the Remove Item icon removes the user from this organization.

Notes: The name of any user you assign here is a link. Clicking it allows you to edit the properties of the given user. Editing a user here is the same as editing it through the Users interface.

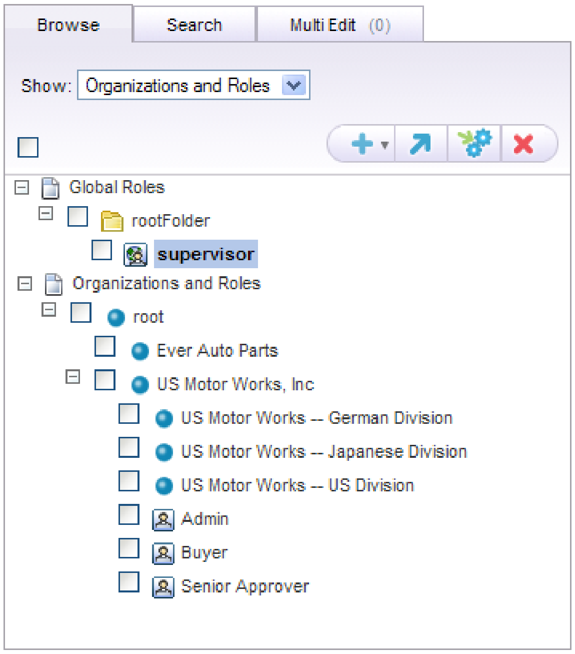
The Secondary Members area is used by ATG Service applications for security administration. For more information, refer to the *ATG Service Installation and Configuration Guide*.

Viewing Roles

To view roles, complete the following steps:

1. Display the BCC Home page.
2. Expand the Personalization item and select the appropriate Users option. **Caution:** If you choose External Users, be aware that you will be viewing live profiles on your production server.
3. From the Show dropdown menu, select Organizations and Roles.

The following image shows the list that appears in the Browse tab:

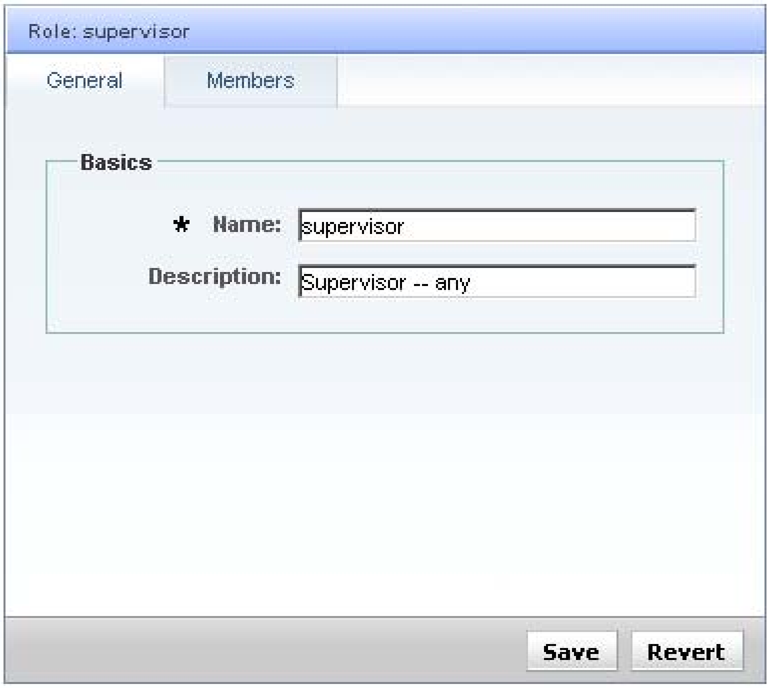


Show: Organizations and Roles

- Global Roles
  - □ rootFolder
    - □ supervisor
- Organizations and Roles
  - □ root
    - □ Ever Auto Parts
    - □ US Motor Works, Inc
      - □ US Motor Works -- German Division
      - □ US Motor Works -- Japanese Division
      - □ US Motor Works -- US Division
      - □ Admin
      - □ Buyer
      - □ Senior Approver

As described earlier in this guide, the list shows all organizations in the selected profile repository as well as all roles. Global roles appear in the Global Roles folder at the top of the list. Organizational roles appear in the Organizations and Roles folder. In the image above, “supervisor” is a global role and “Admin,” “Senior Approver,” and “Buyer” are organizational roles.

The following image shows the tabs that appear on the right when you select a global role.



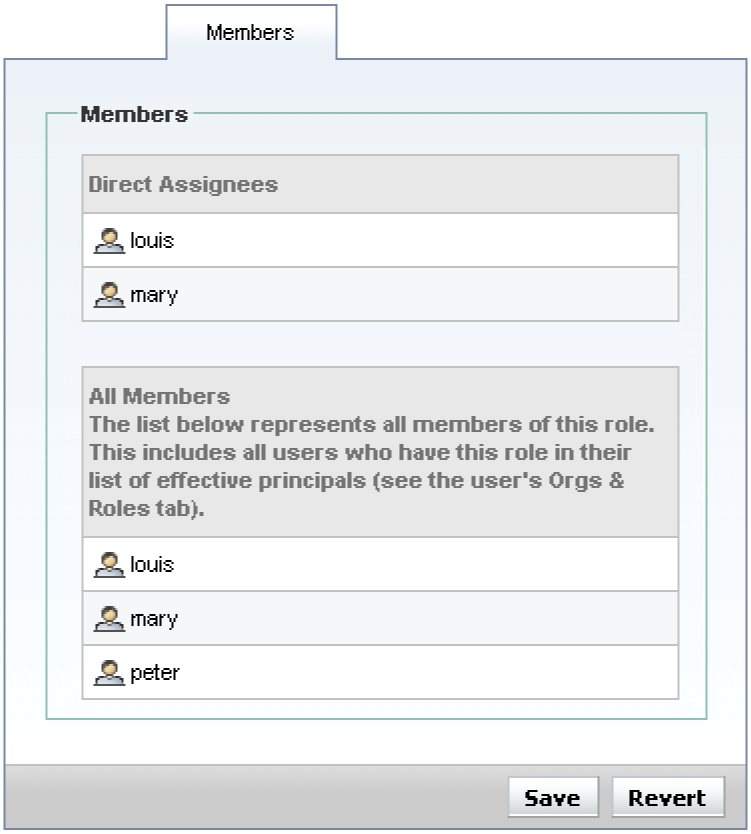
Role: supervisor

* Name: supervisor

Description: Supervisor -- any

Note: For roles that apply to internal users, an Access Rights tab also appears. This tab is used by ATG Service applications to configure security for options in the ATG Service UI. For more information, refer to the ATG Service documentation.

The General tab contains basic setup properties for this role. The Members tab, shown below, displays the users who have this role:



Members

Direct Assignees

louis

mary

All Members
The list below represents all members of this role. This includes all users who have this role in their list of effective principals (see the user's Orgs & Roles tab).

louis

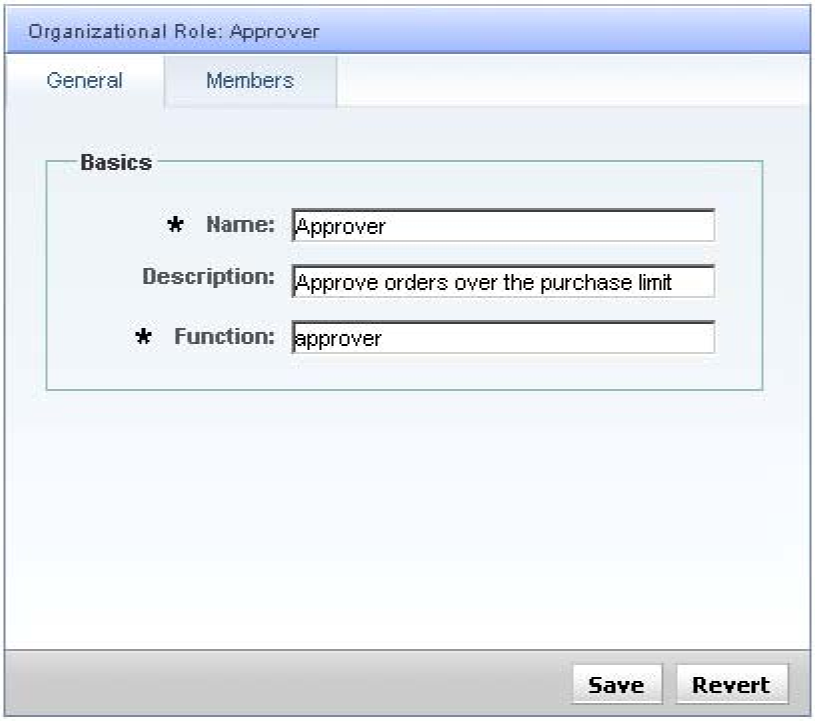
mary

peter

The Direct Assignees list shows users to whom this role has been directly assigned (which you do through the Orgs and Roles tab in the Users interface). The All Members list shows all users who have this role, including anyone who has inherited the role through membership of an organization. In the example above, Peter is a member of an organization to which this role was assigned.

The information in this tab is read only.

For organizational roles, the tabs that appear on the right are almost identical to the global role display. The General tab shows basic setup properties, and includes a required Function property, as shown:



Organizational Role: Approver

* Name: Approver

Description: Approve orders over the purchase limit

* Function: approver

The Function property acts as a keyword that allows you to maintain a connection among organizational roles that are similar to each other. For a detailed discussion, see Organizational Roles at the beginning of this chapter.

For organizational roles, the Members tab shows the same information as it does for global roles.

Creating Roles

When you create a role, you specify values for the set of profile properties defined for the role in your system. Two default sets of properties are provided with the ATG Business Control Center—one for global roles and one for organizational roles. If the properties do not meet your business requirements, you can extend the profiles to include different or additional properties. For information on how to do this, refer to *Working with the Dynamo User Directory* in the *ATG Personalization Programming Guide*. This step is optional.

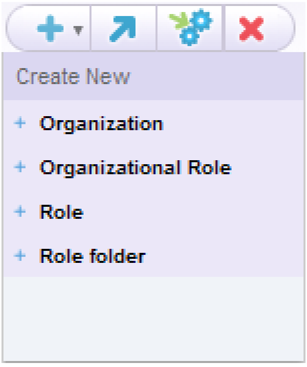
It is important to consider the following information about role names before you start adding roles:

- The names you give global roles must be unique within their folder. However, if you want to use global roles to define membership of a user segment (so that you can then use a person's role as a basis for content targeting, for example), all global roles names must be unique within the profile repository.
- The names you give organizational roles need to be unique only within their organization, even if you plan to use them to define a user segment.

For more information on using roles within a segment or to perform content targeting, see the targeting chapters in the *ATG Business Control Center User's Guide*.

To create a role, complete the following steps:

1. Display the BCC Home page.
2. Expand the Personalization item and select the appropriate Users option. **Caution:** If you choose External Users, be aware that you will be viewing live profiles on your production server.
3. From the Show menu, select Organizations and Roles.
4. In the left pane, select the position where you want the new role to be located. For a global role, select any folder under the Global Roles root folder. For an organizational role, select the organization to which this role applies. **Note:** To select the item, click its name, not the checkbox next to it. The selection is indicated by blue highlighting.
5. Click the Create New icon and select the appropriate option (for a global role, select Role).



6. In the General tab, fill in all required properties, which are shown with an asterisk. By default, the Name property is required. Role names can include spaces.

   For organizational roles, the Function property is also required. See Organizational Roles at the beginning of this chapter.

   The Members tab is read only—you do not need to specify any information here when you are creating a new role.

7. Click Create. The new role is added to the system and appears in the list on the left. (You may have to refresh the browser window and expand the appropriate folder to see it.)

   Note: If you did not select a location in the list on the left, an error message appears when you click Create. Use the Select Parent button to choose a parent folder or organization.

Assigning Roles

After you have created roles, you can assign them to users. To assign a role directly to a user, use the Orgs and Roles tab in the Users interface (see Using the Orgs and Roles Tab for details). You can assign any global role to any user. You can also assign any organizational role to a user as long as the user is a member (either directly or through inheritance) of the organization for which the role was created.

Users can also be assigned roles indirectly through membership in an organization. To set this up, you need to add the role to the organization first so it can then be acquired automatically by the members of that organization. To add a role to an organization, use the Auto-Applied Roles tab.

7 Configuring Segment Lists

Applications that use ATG Customer Intelligence for creating reports, for example ATG Outreach, can choose to have their reports include user segment-related data. In cases where a Web site maintains large numbers of segments, it may be desirable for performance, or more efficient for display purposes, to use only a subset of those segments for reports. Identifying the segments to use is accomplished by maintaining a segment list, which you can edit in the ATG Business Control Center.

Segment lists can also be used for activities other than reporting. In general, any application that needs to identify “interesting” segments can do so through segment lists.

Each application that can use the segment list feature contributes a segment list to the ATG Business Control Center. You can manage this list by following the steps below:

1. From the BCC Home page, select Personalization > Targeting and Segmentation.
2. Create a project.
3. Select Segment List from the Show dropdown menu.
4. Select the appropriate list, for example Outreach Reports.
5. Select the segments you want to use in reports. If no segments have been added previously to the list, click Add and specify the segments you want to include. For some applications, you can prioritize the segments by moving them up or down in the list as required. Segments at the top of the list have the highest priority. For other applications (typically those using segment lists for reporting purposes), the order of the list is ignored.
6. Click Save when you have finished.
7. Complete and deploy the project.

For information on how segment lists are used by ATG applications, refer to *Managing User Segments* in the *ATG Personalization Programming Guide*.

For information on how to create user segments, refer to the *ATG Business Control Center User’s Guide*.

8 Configuring the Asset Manager

Some areas of the ATG Business Control Center use a highly customizable framework called the Asset Manager, which is used for tasks such as creating and editing assets. This framework permits you to create new Navigation pane views and alter the asset properties that appear in the Details pane. It also allows you to change the labels that are displayed for buttons, tabs, and views as well as add or hide UI elements. This chapter describes customization options available for the Asset Manager.

Important: In ATG 10.0.2, some parts of the Asset Manager framework use JavaServer Page (JSP) technology and others are built in Flex. The information in this chapter applies only to the JSP-based framework. The user interface for Flex-based applications cannot be customized. Flex-based applications are as follows:

- ATG Merchandising, including the Promotions and Coupons interface
- Site Administration

This chapter includes the following sections:

Customizing the Duplication Identifier

When you create a duplicate of an item, the ATG Business Control Center determines the name of the duplicate by appending `(copy)` to the name of the original. The duplicate’s name must incorporate the name of the original, but you can replace `(copy)` with another duplicate identifier and change its location in respect to the original name.

The Basic Clone Helper component contains two properties that control the naming format used for duplicated items: `cloneNamePrefix` and `cloneNameSuffix`. By altering the contents of these properties, you specify a prefix or suffix that, in combination with the original item's name, are used for the name of duplicated items. Currently, `cloneNameSuffix` contains `(copy)`, which adds `(copy)` to the end of the original name: `blender(copy)`, for example.

When deciding how you want duplicates to be named, consider that items in lists, such as those that display in the Users or User Segments view of the Browse tab, are alphabetized by default. If you specify a prefix for duplicated items, all duplicated items will be grouped together automatically because they begin with the same letters, whereas specifying a suffix will place each duplicate beside the original it is derived from.

For example, you can configure all duplicates to take the format of `dupe original_name` by following these steps:

1. In the ATG Control Center **Pages and Components** > **Components by Path** task area, double-click the following component to open it in the component editor: `atg/web/assetmanager/action/NameChangingCloneHelper`
2. Find the `cloneNameSuffix` property. Delete the value `(copy)` from both the Configured Value and Live Value columns.
3. Find the `cloneNamePrefix` property. Enter `dupe` in both the Configured Value and Live Value columns.
4. Save your changes.

Preventing Duplicate Values in Lists and Arrays

Users of the ATG Business Control Center can add the same item more than once to a collection property type if the collection is a list or an array. For example, in ATG Merchandising, a SKU can be added twice to the same product, or a category can be added twice to the same parent category.

You can prevent this type of duplication by setting the `prohibitDuplicates` attribute to `true` in the item descriptor of the asset that contains the collection property. The asset item descriptor is located in the repository definition XML file for the repository that contains the asset. By default, duplicates are allowed for lists and arrays.

Caution: Setting the `prohibitDuplicates` attribute to `true` only prevents addition of duplicate values through the user interface. This setting does not prevent the direct import of duplicate values. Also, this setting has no effect on existing duplicate values.

You can set the `prohibitDuplicates` attribute individually for any repository item asset type. Continuing the ATG Merchandising example, you can allow a product to be added multiple times to a category, but prohibit a SKU from being added multiple times to a product.

In the following sample XML, the `prohibitDuplicates` attribute is set to prevent users from adding the same fixed child product more than once to a single category:

```
<property category-resource="categoryProducts" name="fixedChildProducts"
 data-type="list" component-item-type="product" column-name="child_prd_id"
 queryable="true" display-name-resource="fixedChildProducts">
 <attribute name="propertySortPriority" value="-4"/>
 <attribute name="references" value="true"/>
 <attribute name="prohibitDuplicates" value="true"/>
</property>
```

For more information about repository definition XML files, item descriptors, and collection property types, see the *ATG Repository Guide*.

Modifying the Apply to All Behavior

When you are using the Apply to All mode in the Multi Edit tab, you indicate whether you want to change, clear, or keep the existing value for each property that appears in the Details pane. If you select Change as the Update Action, you are required to provide a new value, and an error message appears if you do not. Because you cannot edit properties in the Multi Edit tab that contain images, you receive an error whenever you set an image property to Change.

The `checkForEmptyChangeProperties` property of the `/atg/web/assetmanager/multiedit/ApplyToAllFormHandler` component controls this behavior and is set to `true` by default. If you set this property to `false`, you change the Apply to All behavior as follows:

- Selecting Change and leaving a property empty removes the values currently held by it.
- Image properties are editable.

Changing UI Labels

All labels visible in the Personalization area of the ATG Business Control Center are defined in a resource bundle called `WebAppResources.properties` located in one of the following places:

Module	Resource Bundle Path and Name	Description
`WebUI`	`<ATG10dir>/WebUI/lib/classes.jar/atg/web`	Navigation pane tree
`AssetUI`	`<ATG10dir>/AssetUI/lib/classes.jar/atg/web/assetmanager`	Tabs and general UI infrastructure used by all ATG Business Control Center applications

DPS-UI	<ATG10dir>/DCS-UI/lib/classes.jar /atg/web/personalization	Views and other UI elements designed for targeters, users, roles, and other Personalization items.

Keeping labels in a resource bundle makes your application available for localization. If you want to add or change a label, do so in a custom resource bundle rather than the default resource bundle so that your changes are not overwritten when you install a new version of ATG Business Control Center. Resource bundles should exist in a directory such as `<ATG10dir>/home/locallib` that automatically adds files to your CLASSPATH.

Labels for custom screen elements should be defined in a resource bundle. If you created a view in the Browse tab that lists all manufacturers, for example, you can provide a label to that view by completing these two tasks:

1. Update a custom task configuration file to identify the appropriate resource bundle (if you are using a resource bundle that isn't the default) and the resource bundle key for the view:

```
<tabs>
  <tab id="browse">
    <views>
      <view-order>
        <view-id>manufacturer</view-id>
        <view-id>catalogTree</view-id>
        <view-id>catalogMedia</view-id>
        <view-id>catalogOrphans</view-id>
        <view-id>promotions</view-id>
        <view-id>priceLists</view-id>
        <view-id>pricelistFolders</view-id>
        <view-id>coupons</view-id>
      </view-order>
      <initial-view>
        manufacturer
      </initial-view>
      <view id="manufacturer">
        <resource-bundle>
          atg.commerce.web.MyResourceBundle
        </resource-bundle>
        <display-name-resource>
          browseTab.view.manufacturer
        </display-name-resource>
        <configuration>
          /atg/commerce/web/assetmanager/
          ManufacturerViewConfiguration
        </configuration>
      </view>
    </views>
  </tab>
<tabs>
```

For more information on the task configuration file, see Customizing the UI for Specific Tasks and Activities.

2. In the resource bundle, specify the key from the task configuration file equal to the name you want to use in the UI:

```
browseTab.view.manufacturer=Manufacturers
```

Customizing the Asset Picker

Note that this section does not apply to Flex-based applications (ATG Merchandising, the Promotions and Coupons interface, and Site Administration). See the introduction to this chapter for more information.

The Asset Picker is the mechanism that the ATG Business Control Center provides to allow users to search or browse for assets. It is presented whenever users need to choose an asset—for example, when they select the Add Asset to Project option in the Assets tab or specify an asset as the target of a link from another asset. The Asset Picker is also used as the starting point for creating several types of assets.

The Asset Picker is extensible, allowing you to add custom plugin components that support asset search or browse functionality for custom asset types.

The Asset Picker consists of the following pieces:

- A set of JSPs that provide the base-level pages of the Asset Picker and function as a container for the plugin search or browse components. The default pages are packaged in the `AssetUI.war` file in `<ATG10dir>\AssetUI\j2ee-apps\assetUI.ear`.
- Container JavaScript API: an API implemented by the container pages that is accessed by the plugin page and returns information on the specific asset type included in the search.
- Plugin JavaScript Interface: an API implemented by the plugin page and called by the container page to communicate with the Asset Picker plugins.
- Client JavaScript API: an API implemented by the ATG Business Control Center. It is used to determine the possible asset types selectable by the picker, to configure its many options, and to invoke it.
- Items in the View Mapping repository, which determine the appropriate plugin to present to the user.
- JSPs that represent the plugin components. Default plugins are provided for repository search, virtual file system search, and virtual file system browse.

You can change the default tab that is presented to users when they open the Asset Picker. To do so, edit the `assetPickerDefaultView` property in the `/atg/web/assetmanager/Preferences` component. Possible values are `Search`, `Browse`, or the display name of any custom asset picker tabs you have added to your environment. (Specifically, the value you enter is the `defaultValue` in the `itemViewAttributeDefinition` configured for the tab’s display name.) If no value is set, the Browse tab is displayed.

Asset Picker Container JavaScript API

The `getAssetType()` method is implemented by the Asset Picker container page and is called by the plugin page, which is included in the container through the `dspel:include` tag.

`getAssetType()` returns a JavaScript associative array with the following keys:

Key	Description
`itemType`	The asset item descriptor type.
`componentPath`	The Nucleus path to the repository component for the asset.
`encodedType`	The `itemType` and the component path, separated by a semi-colon.
`itemDisplayProperty`	The display property of the asset item type as specified in the repository definition file.
`typeCode`	Specifies the type of asset being searched for. Can be either "file" or "repository".

The plugin uses these values to perform the search or browse on the item returned by the `getAssetType()` method.

The following example is from `assetPickerContainer.jsp` in `<ATG10dir>\AssetUI\j2ee-apps\assetui.ear\assetui.war`.

```
<script language="JavaScript">
   //
   // Returns the selected asset type to the plugin
   //
   function getAssetType() {
     var assetType             = new Object();
       assetType.itemType      = "<c:out value='${itemType}'/>";
       assetType.componentPath = "<c:out value='${componentPath}'/>";
       assetType.encodedType   = assetType.componentPath
                                   + ":"
                                   + assetType.itemType;
       assetType.itemDisplayProperty
         = getItemDisplayProperty( assetType.componentPath, assetType.itemType );

       for (var i=0; i<assetTypes.length; i++) {
         if ( assetTypes[i].componentPath == assetType.componentPath
           && assetTypes[i].typeName == assetType.itemType ) {

           assetType.typeCode = assetTypes[i].typeCode;
         }
       }
```

```
      return assetType;
    }
```

The next example, from `repositorySearchResults.jsp`, shows how this method is called by a plugin page:

```
function performSearch() {
  var assetType           = getAssetType();

  var itemType            = assetType.itemType;
  var componentPath       = assetType.componentPath;
  var itemDisplayProperty = assetType.itemDisplayProperty;
  var encodedType         = assetType.encodedType;

  document.repositorySearchForm.componentPath.value =
    componentPath;
  document.repositorySearchForm.itemType.value =
    itemType;
document.repositorySearchForm.encodedType.value =
  encodedType;

if ( itemDisplayProperty != null ) {
  document.repositorySearchForm.sortProperty.value =
    itemDisplayProperty;
}
else {
  document.repositorySearchForm.sortProperty.value =
    "displayName";
}
  document.repositorySearchForm.submit();
}
</script>
```

Asset Picker Plugin JavaScript Interface

Asset picker plugins must implement the following JavaScript interface:

Method	Description
`getSelectedAssets()`	Returns a JavaScript array. Each element in the array is itself an associative array consisting of the following keys and values: `id`: The asset's repository ID `displayName`: The asset's display name. `uri`: The asset's Version Manager URI.

The container page calls this method when the user selects an appropriate control in the Asset Picker UI—for example, when the user clicks the Add button to add an asset to a project. The control's function is performed against the assets returned by the `getSelectedAssets()` method.

The following example is from `repositorySearchResults.jsp` in `<ATG10dir>\AssetUI\j2ee-apps\assetui.ear\assetui.war`:

```
<script language="JavaScript">

    function getSelectedAssets() {

      var selectedResults = new Array();
      var resultsForm     = document.resultsForm;

      if ( resultsForm.selectedAssets != null ) {
        // this is the condition where there is one result
        if ( resultsForm.selectedAssets.length == null ) {
          if ( resultsForm.selectedAssets.checked == true) {
            var result = results[ resultsForm.selectedAssets.value ];
            selectedResults[0] = result;
          } // if
        } // if
        // multiple results
        else {
          var j = 0;
          for (var i=0; i < resultsForm.selectedAssets.length; i++) {
            if (resultsForm.selectedAssets[i].checked == true) {
              var result = results[ resultsForm.selectedAssets[i].value ];
              selectedResults[j] = result;
              j++;
            }  //if
          } // for
        } // else
      }

      return selectedResults;
    }
</script>
```

Asset Picker Client JavaScript API

The Asset Picker client (the ATG Business Control Center, in the case of ATG Content Administration) uses a JavaScript `AssetBrowser` object to populate the possible asset item types and repositories that can be selected by the picker. The client also uses the object to configure various options available to the picker.

The following example of the `AssetBrowser` object is taken from the ATG Business Control Center page `<ATG10dir>\PubPortlet\PubPortlets.ear\portlets.war\html\ProjectsPortlet\projectAssets.jsp`, which is used to add assets to a project:

```
var picker = new AssetBrowser();
    picker.browserMode        = "project";
    picker.createMode         = "redirect";
    picker.closeAction        = "refresh";
    picker.assetTypes         = assetTypesList;
    picker.formHandler        = "/atg/epub/servlet/ProjectFormHandler";
    picker.formHandlerAction  = "1";
    picker.assetInfoPath      = "/atg/epub/servlet/ProcessAssetInfo";
    picker.defaultView        = "Search";
    picker.invoke();
```

The `browserMode` property can be set to `"project"`, used for pages where multiple assets can be selected, or `"pick"`, used for pages that allow selection of individual assets.

The `defaultView` property specifies the tab that is active when the page is displayed to the user.

For another example, see `<ATG10dir>\PubPortlet\PubPortlets.ear\portlets.war\html\views\property\item\popupPickerEdit.jsp`, which is an ATG Business Control Center page that allows users to create a link that points to an asset.

The `AssetBrowser` object also has a property called `viewConfiguration` that makes customized data defined in the invoking JavaScript available to the plugin JSP.

```
var viewConfig = new Array();
viewConfig.key1 = 'value1';
viewConfig.key2 = 'value2';
var picker = new AssetBrowser();
picker.viewConfiguration = viewConfig;
```

The `viewConfig` associative array is available to the plugin via a `getViewConfiguration()` JavaScript method implemented by the container page (see `assetPicker.jsp`). The key/value pairs in the array are also available as page variables in the plugin page. For example, these key/value pairs:

```
        <p>key1 = <c:out value='${key1}' />

        <p>key2 = <c:out value='${key2}' />
```

render as follows:

```
        key1 = value1

        key2 = value2
```

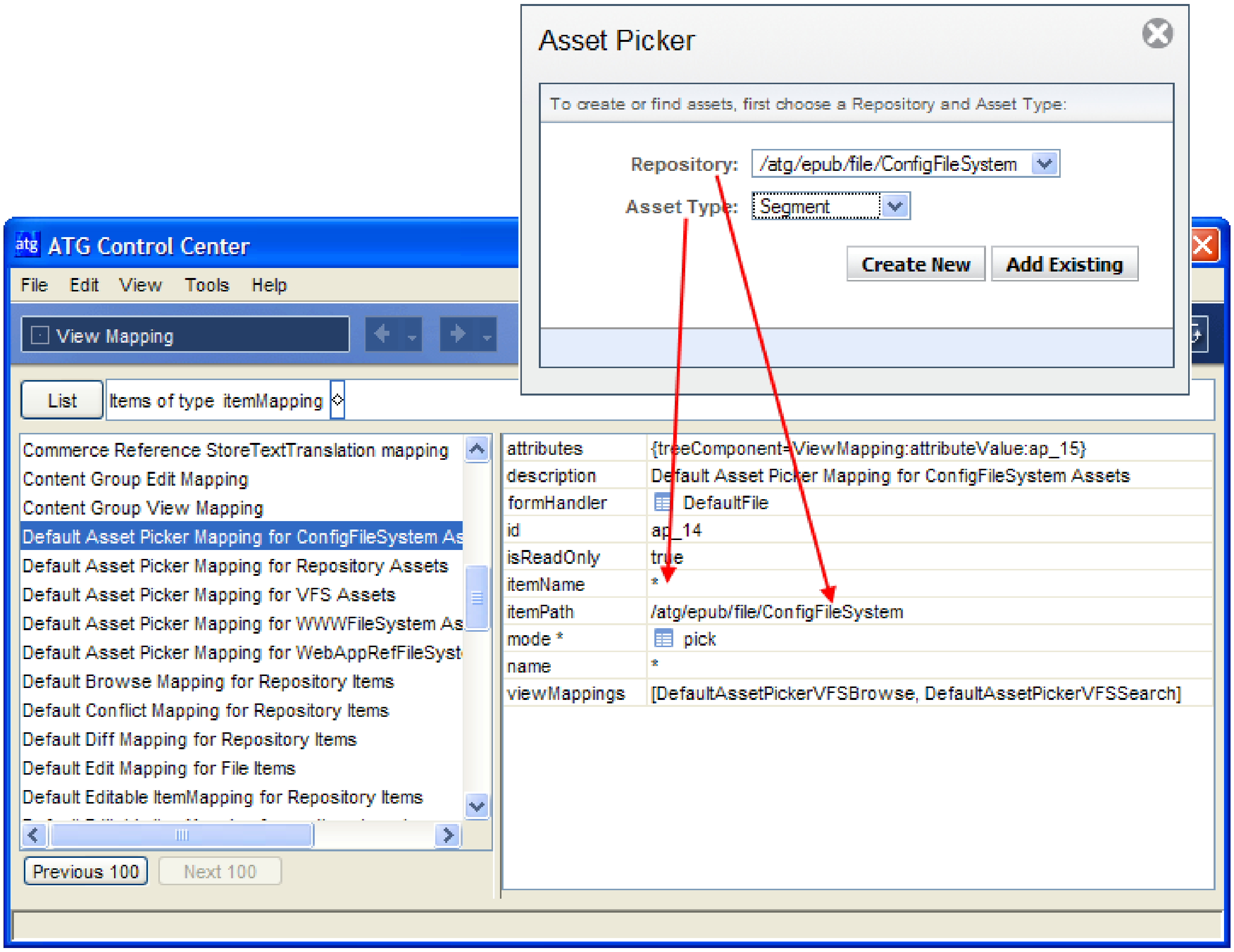
View Mapping Items Used by the Asset Picker

The page or pages that are displayed for a particular repository and asset type are determined by the View Mapping system. Each plugin page is registered as an `itemView` in the View Mapping repository (component `/atg/bizui/viewmapping/ViewMappingRepository`). One or more `itemViews` are mapped to an `itemMapping` via one or more `itemViewMappings`.

The Asset Picker determines the `itemMapping` to use for a given repository and asset type by evaluating three `itemMapping` properties that are specified as attributes in the `<biz:getItemMapping>` tag of the container page, for example `assetPickerContainer.jsp`:

- `Mode`: Always `pick` for search or browse plugins.
- `itemName`: The name of the asset item descriptor for which the plugin is applicable.
- `itemPath`: Nucleus path to the asset repository for which the plugin is applicable.

The images that follow show the relationships among the various view mapping items required by an Asset Picker plugin. The first one shows the `itemMapping` that corresponds to the user's choice of repository and asset type in the initial Asset Picker dialog box. Note that the asterisk in the `itemName` field acts as a wildcard that includes any asset in the repository specified in the `itemPath` field.



To create or find assets, first choose a Repository and Asset Type:

Repository: /atg/epub/file/ConfigFileSystem

Asset Type: Segment

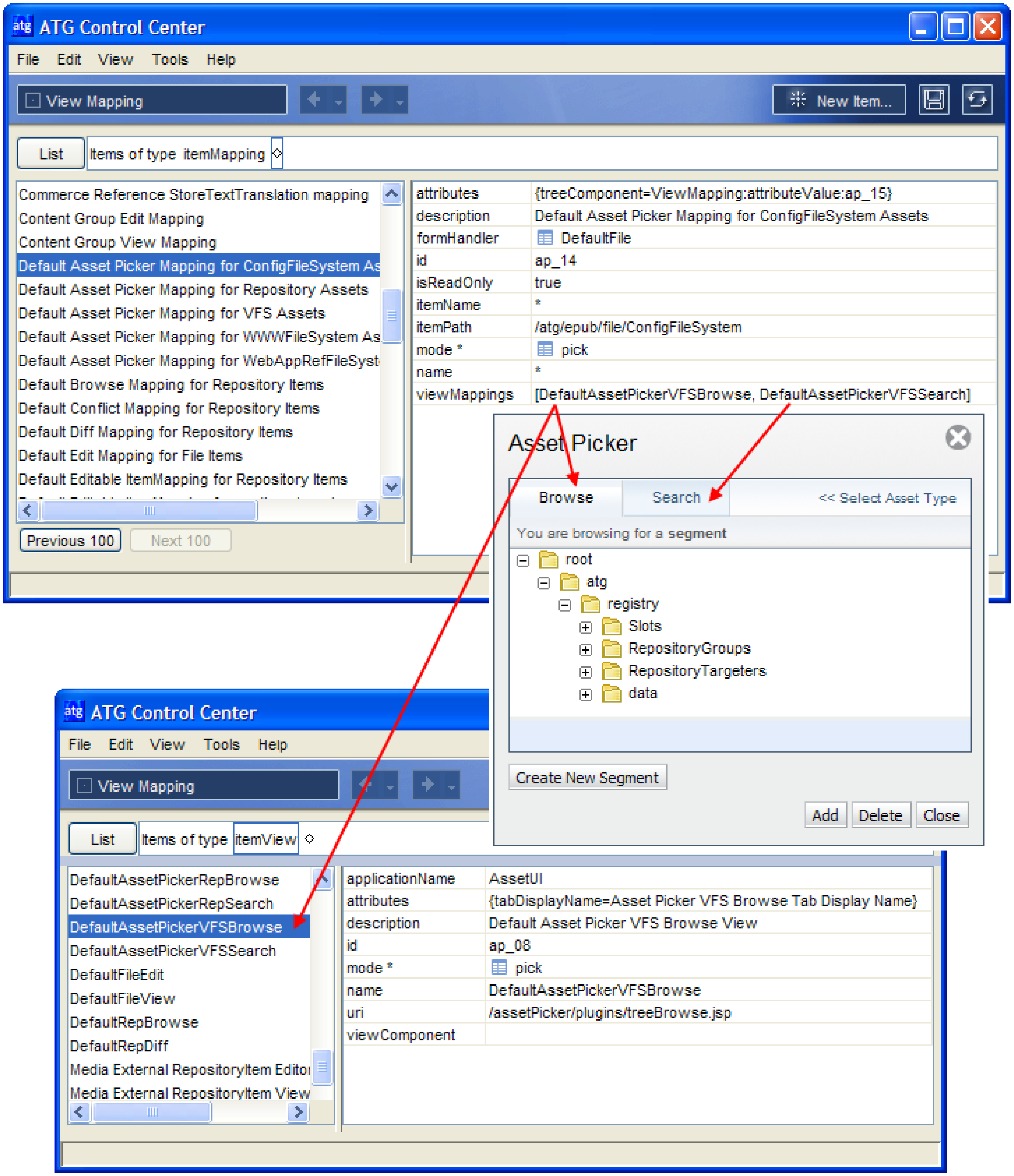
Items of type itemMapping

- Commerce Reference StoreTextTranslation mapping
- Content Group Edit Mapping
- Content Group View Mapping
- Default Asset Picker Mapping for ConfigFileSystem As
- Default Asset Picker Mapping for Repository Assets
- Default Asset Picker Mapping for VFS Assets
- Default Asset Picker Mapping for WWWFileSystem As
- Default Asset Picker Mapping for WebAppRefFileSyst
- Default Browse Mapping for Repository Items
- Default Conflict Mapping for Repository Items
- Default Diff Mapping for Repository Items
- Default Edit Mapping for File Items
- Default Editable ItemMapping for Repository Items

attributes	{treeComponent=ViewMapping:attributeValue:ap_15}
description	Default Asset Picker Mapping for ConfigFileSystem Assets
formHandler	DefaultFile
id	ap_14
isReadOnly	true
itemName	*
itemPath	/atg/epub/file/ConfigFileSystem
mode *	pick
name	*
viewMappings	[DefaultAssetPickerVFSBrowse, DefaultAssetPickerVFSSearch]

Note also that the name of the asset in the Asset Type field (in this case, `Segment`) is determined by the `display-name` attribute for the corresponding item descriptor in the repository definition file.

The second image shows how the `viewMappings` field in the `itemMapping` specifies the tabs that should appear in the plugin. The `itemView` defines the page that is used to display the contents of each tab.



Commerce Reference StoreTextTranslation mapping
Content Group Edit Mapping
Content Group View Mapping
Default Asset Picker Mapping for ConfigFileSystem As
Default Asset Picker Mapping for Repository Assets
Default Asset Picker Mapping for VFS Assets
Default Asset Picker Mapping for WWWFileSystem As
Default Asset Picker Mapping for WebAppRefFileSyst
Default Browse Mapping for Repository Items
Default Conflict Mapping for Repository Items
Default Diff Mapping for Repository Items
Default Edit Mapping for File Items
Default Editable ItemMapping for Repository Items

attributes	{treeComponent=ViewMapping:attributeValue:ap_15}
description	Default Asset Picker Mapping for ConfigFileSystem Assets
formHandler	DefaultFile
id	ap_14
isReadOnly	true
itemName	*
itemPath	/atg/epub/file/ConfigFileSystem
mode *	pick
name	*
viewMappings	[DefaultAssetPickerVFSBrowse, DefaultAssetPickerVFSSearch]

You are browsing for a segment

- root
  - atg
    - registry
      - Slots
      - RepositoryGroups
      - RepositoryTargeters
      - data

DefaultAssetPickerRepBrowse
DefaultAssetPickerRepSearch
DefaultAssetPickerVFSBrowse
DefaultAssetPickerVFSSearch
DefaultFileEdit
DefaultFileView
DefaultRepBrowse
DefaultRepDiff
Media External RepositoryItem Editor
Media External RepositoryItem View

applicationName	AssetUI
attributes	{tabDisplayName=Asset Picker VFS Browse Tab Display Name}
description	Default Asset Picker VFS Browse View
id	ap_08
mode *	pick
name	DefaultAssetPickerVFSBrowse
uri	/assetPicker/plugins/treeBrowse.jsp
viewComponent	

Default Search and Browse Plugins

The following plugins are provided by default for use with the Asset Picker:

- Repository search (`repositorySearch.jsp`)
- Virtual File System search (`vfsSearch.jsp`)
- Virtual File System browse plugin (`vfsBrowse.jsp`) that displays assets as an expandable tree of directories. The plugin is configured for the following repositories:
  - `WWWFileSystem`
  - `ConfigFileSystem`
  - `WebAppRefFileSystem`

  These components appear as tabs in the Asset Picker in the appropriate editing context—for example, the VFS browse tab appears when the user chooses to browse existing file-based assets. Note that a default repository browse component is not included with ATG Content Administration, but the section Creating a Tree-Based Asset Picker Plugin describes how to configure one.

Configuring an Asset Picker Plugin

This section describes how to configure a custom plugin for use with the Asset Picker.

In most cases, you can use the default Asset Picker container pages to call your plugin component without any additional configuration of these pages. Also, if your objective is simply to make assets from a custom repository available for searching or browsing in the ATG Business Control Center, you can use the default plugin pages to do so (see Default Search and Browse Plugins). You need to create new pages only if the default UI handling does not meet your business requirements.

In all cases, to allow your plugin to be displayed and used in the ATG Business Control Center, you must register it with the view mapping repository. To do so, you need to configure the following view mapping items:

- An `itemView` specifying the JSP that represents the plugin.
- An `itemViewMapping` representing the tab—for example, a Search tab—that presents the plugin to the user.
- An `itemMapping`, which declares the asset type and view mode for which the plugin is used.

You create view mapping items through the Publishing > View Mapping option in the ACC. For detailed instructions, see *Creating View Mapping Repository Items* section of the *Customizing Asset Display* chapter in the *ATG Content Administration Programming Guide*.

Information on key properties for each of the view mapping items that you need for the plugin is given below.

Plugin itemView

The following list describes key properties in the `itemView` you create for an Asset Picker plugin. As a model, use the `DefaultAssetPickerVFSSearch itemView`.

- `applicationName`: Specify the display name of the J2EE Web application in which the plugin page is stored. The display name is defined in the application's `web.xml` file.

You can store plugin pages in any J2EE Web application that runs with the AssetUI Web application. (Default Asset Picker pages are stored in AssetUI.war, which has the display name AssetUI.) The applicationName value is used to resolve the context root of the application.

- attributes: Create an itemViewAttributeDefinition (a Map) with a key value of tabDisplayName whose defaultValue is set to the name you want to display on the tab for this plugin (frequently Search).
- mode: specify pick
- uri: specify the path to the plugin page in the contained Web application. Do not include the context root, which is resolved from the applicationName, as mentioned earlier. Example: the itemView representing the default Asset Picker plugin for searching file-based assets has the following value: /assetPicker/plugins/vfsSearch.jsp.

Plugin itemViewMapping

The itemViewMapping represents the tab that you want to appear on the page. As a model for the itemViewMapping, use DefaultAssetPickerVFSSearch.

There is one key property to define in the itemViewMapping:

- view : specify the itemView you created for the plugin.

Plugin itemMapping

The following list describes key properties in the itemMapping you create for the Asset Picker plugin. As a model, use the Default Asset Picker Mapping for ConfigFileSystem Assets.

- itemPath: Nucleus path to repository for which this plugin is applicable. Specify Repository Item for all repositories or File for all virtual file systems.
- itemName: the name of the item descriptor for which this plugin is applicable. Enter an asterisk to specify all items in the given repository.
- mode: specify pick.
- viewMappings: Specify the itemViewMapping you created for the plugin, as well as any other itemViewMappings you want to appear in this display. (itemViewMappings represent tabs in the user interface, so if your custom plugin is a Search tab, you might want a corresponding browse tab to appear as well.) Default itemViewMappings are DefaultAssetPickerRepSearch (searches repository assets), DefaultAssetPickerVFSSearch, and DefaultAssetPickerVFSBrowse (which displays assets as an expandable tree).

Note that the Default Asset Picker Mapping for ConfigFileSystem Assets uses the attributes property to identify a component that allows users to browse assets as an expandable tree; this behavior is optional. For more information, see Creating a Tree-Based Asset Picker Plugin.

Creating a Tree-Based Asset Picker Plugin

This section describes how to configure a plugin browse component that displays repository assets as an expandable tree of folders and items. (See the image of the Browse tab in the previous section for an example .) Note that a similar plugin for browsing file-based assets as a tree is provided with ATG Content Administration.

If you want to add your tree to a custom view, see Implementing a New Tree View. Many of the resources you create and configure here are also described in the views section. You need only create these resources once and they can be used by both the view and Asset Picker.

The procedure for configuring the repository browse plugin has these steps, which are described in the sections that follow:

1. Create a Mutable Repository Tree Definition component, which defines the repository item relationships that make up the tree. Also, use this component to identify the property whose value determines the order of items in the tree. See Mutable Repository Tree Definition Component.
2. A default component, `/atg/web/tree/CountBasedBucketer`, manages the display of large numbers of items by dividing them into buckets. If all trees use the same bucketing specifications, use the existing component, modifying its settings as needed. If bucketing specifications vary, create Count Based Bucketer components for each tree. See Count Based Bucketer Component.
3. Create a Bucketing Tree Impl component to manage the resources used for buckets. See Bucketing Tree Impl Component.
4. Create a Tree State component to handle persistence of the tree state for each user in the appropriate project context. See Tree State Component.
5. Register the tree.
   - If you are adding the Asset Picker tree to the Asset Manager part of the ATG Business Control Center, register the tree in a Tree Registry Component. See Tree Registry Component.
   - If you are adding the Asset Picker tree to the base ATG Business Control Center (ATG Content Administration), register the tree in the View Mapping system. See Registering the Browse Plugin with the View Mapping System.

Mutable Repository Tree Definition Component

Create a global-scoped component of class `atg.web.tree.repository.MutableRepositoryTreeDefinition` to configure the parent-child relationships of the repository whose items you want to display. The parent-child relationships correspond to the tree structure that will appear to the user. By defining them here, you allow the client to request the children of any repository item without needing any knowledge of the structure of the repository itself. For example, if you wanted to show related products, you would provide the following value to the `treeLinks` property:

```
product=fixedRelatedProducts;dynamicRelatedProducts;
catalogsRelatedProducts
```

Configure one component of this type for each tree plugin you are adding.

A tree display can have top-down or bottom-up structures. In a top-down structure, an item has one or more properties that point to its children. In a bottom-up example, an item has a property that points to its parent. For example:

```
folder=folder:parentFolder;media:parentFolder
```

Given the previous example, this means that items of type `parentFolder` have two types of children:

- `folder`
- `media`

For these items, the `parentFolder` property equals the `folder` whose children are displayed in this tree.

For a top-down relationship, the `treeLinks` property is specified with slightly different syntax, where the keys are the names of `itemDescriptors` and the values are semicolon-delimited lists of the property names of that `itemDescriptor's` children. In the following example, the children of the `parentFolder` item type are held in properties `folder` and `media`.

```
parentFolder=folder;media
```

The way items are sorted depends on whether those items use top-down or bottom-up relationships. Most top-down items appear in the order they were created or added and can be rearranged manually in the UI. For items with bottom-up relationships, the `sortProperties` property of this component holds the property whose values determine the sort order. By default, items are sorted by display name in alphabetical order.

There are two important things to know about sorting:

- Regardless of whether the tree uses top-down or bottom-up relationships, you can use a Node Sorter component to define a complex sorting strategy that, for example, incorporates multiple levels of sorting. See Customizing the Order of Items in a Tree.
- If your tree branches contain a large number of assets, they will be automatically grouped into buckets. See the Count Based Bucketer Component section for configuration details.

All Mutable Repository Tree Definition classes provide the following configurable properties:

Property	**Description**
`hideRootNodes`	Indicates whether root nodes are hidden from the Browse tab and Asset Picker views. When you create new items, a root node is not automatically assigned as the parent.
`repository`	The Nucleus component that represents the repository whose assets you want to display in this tree.

`rootNodeRQL`	The RQL query that is used to get the root nodes. It might be as simple as ALL or might include some collection criteria—for example, `parent IS NULL`.
`sortProperties`	Provides, for each item type in the tree, a property whose value determines an item's position in the tree as well as the direction – ascending or descending – the values are sorted in. The format is *`item type=property:direction`*.
`treeLinks`	Describes the parent to child relationships of the tree in a map. The exact syntax varies depending on whether the tree has a top-down or bottom-up hierarchy. See the examples below.

Count Based Bucketer Component

A session-scoped Nucleus component of the class `atg.web.tree.CountBasedBucketer` manages the display of large numbers of assets by chunking them into groups, or "buckets." The following diagram illustrates this concept, which is similar to paging:

```
+ folder a
- folder b
   + folder 1
   + folder 2
   + folder 3
   - folder 4
       o items 1-25
       o items 26-50
       o items 51-75
+ folder c
```

Folder 4 contains more than a specified number of items, which are grouped into buckets (shown with the symbol "o" here) that contain 25 items each. Each bucket is named for the type of items it contains. Note that a bucket can contain one type of item only.

In cases where very large numbers of assets exist, buckets are created within buckets (also called double bucketing).

The `bucketRatio` and `bucketSize` properties determine the number of items that appear in a bucket and the number of items that triggers bucketing and double bucketing.

The default component of this class, `/atg/web/tree/CountBasedBucketer`, has a `bucketingRatio` of 1.5 and `bucketingSize` of 40. Modify these values as needed.

Buckets use the sorting strategy defined for the items they contain. If you sort items by creation date (using the `sortProperties` property of the `MutableRepositoryTreeDefinition` component), buckets group items by creation date. Each bucket has a label that shows the range of its contents identified by display name, such as City Girl Boots – Spring Showers Boots, regardless of the property used to sort the items within it. You can change the bucket label so it names a range of property values that represent the property used for sorting. To do this, extend the Count Based Bucketer class and override the putLabelsOnBuckets(List pBuckets) method, replacing it with a method that locates the property held by the sortProperties property.

If you prefer to have multiple levels of bucketing, each of which organizes items based on a different property value, for example, you can create a Node Sorter component as described in Customizing the Order of Items in a Tree.

A Count Based Bucketer component has the following configurable properties:

Property	**Description**
bucketRatio	Determines when to bucket items and when to simply display all items without bucketing them. Bucketing begins when the number of items is greater than the bucketSize multiplied by the bucketRatio. For example, if the bucketSize is 40 and the bucketRatio is 1.5, as in the sample code below, bucketing occurs only if there are 60 items or more. If the number of items is greater than (bucketSize x bucketSize) x bucketRatio, (bucketSize squared, multiplied by the bucketRatio) a second layer of bucketing, called double bucketing, occurs. For example, if the bucketSize is 40 and the bucketRatio is 1.5, and there are more than 2400 items (40 X 40 X 1.5), they are shown in double buckets.
bucketSize	The number of items per bucket.
useDisplayNameInBucketLabels	Determines what displays in the bucket label. If true, the label is the name of the first and last items in the bucket separated by a dash. For example: Crunchproof Helmet – Young Children's Helmet. When set to false, the label is named for the type of item it contains and includes the quantity of items as a range. For example: Products 1-40. Buckets will be generated more quickly when this property is set to false.

The following example is from CountBasedBucketer.properties, which defines the buckets for all trees, such as the Organizations and Roles view of the Browse tab.

```
$class=atg.web.tree.CountBasedBucketer

$scope=global

bucketRatio=1.5

bucketSize=40
```

Bucketing Tree Impl Component

Create a global component of class `atg.web.tree.BucketingTreeImpl` to control the display of items in the tree. Configure one component of this type for each tree plugin you are adding.

All Bucketing Tree Impl classes provide the following configurable properties:

Property	Description
`nodeBucketerPath`	Names the Count Based Bucketer component that controls the generation and contents of buckets.
`nodeSorterPaths`	Names the Node Sorter component that defines a multi-level sorting strategy for items in the tree. This component is optional. See Customizing the Order of Items in a Tree for more information about it.
`treeDefinition`	Names the Bucketing Tree component that defines the parent-child relationship for this repository.
`treeNodeFilterPaths`	Names the Tree Node Filter that defines rules for determining items to exclude from the tree. This component is optional. See Excluding Items from a Tree for more information.

The following example is from `WWWFileSystemTree.properties`, which defines the Bucketing Tree Impl component for the default VFS browse plugin.

```
$class=atg.web.tree.BucketingTreeImpl

$scope=global

treeDefinition=WWWFileSystemTreeDefinition

nodeBucketerPath=/atg/web/tree/CountBasedBucketer
```

Tree State Component

Create a session-scoped Nucleus component of the class `atg.web.tree.TreeState` that monitors opened, checked, and selected nodes. Keeping track of the tree state ensures that when a user switches between views tree information persists.

Configure one component of this type for each tree plugin you are adding.

A Tree State component has the following configurable properties:

Property	Description
`checkableTypesArray`	Optional. An array of the item descriptors that you want to appear with the tree check box control in the UI. This property is useful for situations where the user is not already prompted to specify the asset types he or she wants to browse. In the example below, the property has a dummy value.
`globalTree`	Specifies the Bucketing Tree component that supplies the tree data.
`selectableTypesArray`	Optional. An array of the item descriptors that you want to be highlighted on selection in the UI. The example below uses a dummy value so no highlighting occurs.

The following example is from `TargeterTreeState.properties`, which defines the Tree State component for the Targeters view.

```
$class=atg.web.tree.TreeState
$scope=session

globalTree=/atg/web/assetmanager/configuration/targeting/TargeterTree
```

Tree Registry Component

A component of class `atg.web.tree.TreeRegistry` makes tree views available to the Asset Manager Asset Picker. Update the existing component of this class, `atg/web/assetmanager/TreeRegistry`, by adding the Bucketing Tree Impl and Tree State components you created earlier to the `treeComponents` property. These components should be the key and value, respectively.

If your tree displays items of a type that is visible in another tree, create a component, add the two mentioned resources to it, and add the component to the `/atg/web/assetmanager/SessionInfo.treeRegistries` property.

The following example is from `TreeRegistry.properties`:

```
$class=atg.web.tree.TreeRegistry

$scope=global

treeComponents=\
    /atg/web/assetmanager/configuration/profile/FilteredCombinedOrgRoleTreeState
    =/atg/web/assetmanager/configuration/profile/CombinedOrgRoleTree\
    /atg/web/assetmanager/configuration/targeting/ContentGroupTreeState=/atg/web/
    assetmanager/configuration/targeting/ContentGroupTree\
    /atg/web/assetmanager/configuration/targeting/SegmentTreeState=/atg/web/asset
    manager/configuration/targeting/SegmentTree\
    /atg/web/assetmanager/configuration/targeting/TargeterTreeState=/atg/web/asset
```

```
manager/configuration/targeting/TargeterTree\
/atg/web/assetmanager/configuration/targeting/nonver/ContentGroupTreeState=/
atg/web/assetmanager/configuration/targeting/nonver/ContentGroupTree\
/atg/web/assetmanager/configuration/targeting/nonver/SegmentTreeState=/atg/
web/assetmanager/configuration/targeting/nonver/SegmentTree/
/atg/web/assetmanager/configuration/targeting/nonver/TargeterTreeState=/atg
/web/assetmanager/configuration/targeting/nonver/TargeterTree
```

Registering the Browse Plugin with the View Mapping System

As part of setting up a repository browse plugin for the Asset Picker in the base ATG Business Control Center, you must register the plugin with the view mapping system by creating an itemMapping for it. The itemMapping includes an attribute that points to the Tree State component. The following table describes how to configure the properties of the itemMapping. If necessary, use the Default Asset Picker Mapping for ConfigFileSystem Assets as a model.

Property	Enter this value
attributes	Specify treeComponent as the key and the Nucleus path to the VersionedRepositoryTreeState component as the value.
formHandler	DefaultRepository
isReadOnly	true
itemName	Specify the repository items to which the plugin applies. Enter an asterisk to specify all items in the repository.
itemPath	The path to the repository component.
mode	Specify pick as the mode.
name	Enter an asterisk.
viewMappings	Specify the tabs you want to include, in other words the itemViewMappings that represent the browse tab and the search tab for repository items. You can use the default itemViewMappings, which are DefaultAssetPickerVFSBrowse and DefaultAssetPickerRepSearch. (**Note:** DefaultAssetPickerVFSBrowse can be used for both VFS and repository browse tabs.)

The DefaultAssetPickerVFSBrowse itemViewMapping calls treeBrowse.jsp by default.

For a detailed description of how to create view mapping items, see *Creating View Mapping Repository Items* section of the *Customizing Asset Display* chapter in the *ATG Content Administration Programming Guide*.

Customizing a Tree-Based Asset Picker Plugin

You can define your Asset Picker further by providing additional configuration:

1. Arrange the assets in your tree using any property they all share. See Customizing the Order of Items in a Tree.
2. Exclude additional assets from the Asset Picker. See Excluding Items from a Tree.
3. Change the icon provided to a custom or an out-of-the-box item type. See Implementing a Custom Icon.
4. By default, nodes are named by an item's display name. Use a different property value as the node identifier. See Customizing the Node Text.
5. Change the color, size, or style of the node label. See Customizing the Node Style.
6. All nodes have checkboxes beside them. You can remove these checkboxes for all items in a given view, thereby making items unavailable for selection. See Removing the Checkboxes Beside Nodes.

Customizing the Order of Items in a Tree

By default, the order of items in a given branch depends on how that item's relationship to other items is defined in the Mutable Repository Definition component. Typically, each item type is organized in the order it was added or the position it was manually moved to. Otherwise, items are ordered by their values for a given property.

A custom Node Sorter component permits you to create a complex sort order that can incorporate multiple levels of sorting: for example, items can be organized first by type, and those with the same type are organized by name.

It is best to implement a Node Sorter component in trees that have a small number of items or a large number of items split across a number of branches. For trees that have 1000 or more items in a given branch, using a custom Node Sorter could affect performance, especially if you use secured repositories.

Follow these steps to implement an alternative sort order:

1. Create a class that implements the `atg.web.tree.sort.NodeSorter` interface.
2. Use the `List sortNodes(List pNodes)` method to specify a property whose value determines an asset's position in the tree.
3. Save the class to the directory that contains your custom code. Restart ATG Content Administration.
4. Create a globally scoped component of the new class.
5. To define multiple levels of sorting, repeat steps 1-4.
6. Add the Node Sorter component name to the `nodeSorterPaths`property of the Bucketing Tree Impl component you created in the Bucketing Tree Impl Component section. If you have multiple Node Sorter components, the `nodeSorters` property should hold a comma-separated list of components that reflects the order in which they should be applied.

Excluding Items from a Tree

By default, the assets included in a tree are derived from the `treeLinks` and `rootNodeRQL` properties on the Mutable Repository Tree Definition component. You might want to exclude additional assets from a view in particular circumstances only. For example, you might want trees to show users only those items that they created. You can exclude assets from a tree by writing a Tree Node Filter class.

Using a Tree Node Filter class might degrade performance in trees where a branch holds 1000 items or more, especially when using secured repositories.

Follow these steps:

1. Create a class that implements the `atg.web.tree.TreeNodeFilter` class.
2. Use the `public boolean accept(TreeNode pNode)` method to specify the assets you want to include from the tree by supplying a property and value.
3. Save the class to the directory that contains your custom code. Restart ATG Content Administration.
4. Create a globally scoped component of the new class.
5. Add the Tree Node Filter component name to the `treeNodeFilterPaths` property of the Bucketing Tree Impl component you created in the Bucketing Tree Impl Component section.

There are several Tree Node Filter subclasses included with ATG Content Administration. One example is the `atg.web.tree.targeting.TargetingTreeNodeFilter` class used to limit the items in the tree to content repository items in a given path. Also, this component changes the node labels.

Implementing a Custom Icon

To replace icons for out-of-the-box item types or provide icons for custom types, follow these steps:

1. Add new icons to your Web application. It is good practice to keep all custom icons in one place, for example

   ```
   <ATG10dir>/MyWebApp/atg/ui/images
   ```

2. Locate the following part of the item descriptor that defines the item type.

   ```
     <item-descriptor name="product">
       <attribute name="icon" value="myIcons"/>
       <attribute name="iconWebAppName" value="MyApp"/>
     </item-descriptor>
   ```

3. Change the value (currently `MyApp`) for the `iconWebAppName` attribute to the name of your Web application.
4. Edit the appropriate custom resource bundle to map the icon name to the icon’s path. To do this, you enter the value of the `icon` attribute (`myIcons`) as the key and the icon’s actual path and name as the value.

Customizing the Node Text

By default, the label for a tree node is the display name for the item it represents. You can change the label to display any property of the item, such as the item ID, by modifying the item's item descriptor. The `<display-property>` attribute identifies the property whose value is used as a label. For more information, see the *SQL Repository Definition Tag Reference* section of the *SQL Repository Reference* chapter in the *ATG Repository Guide*.

Customizing the Node Style

The type face, color, size, and other elements that determine the style used for the label of a tree node are specified in a Cascading Style Sheet (CSS). You can change the appearance of all nodes or design a style for nodes in a specific circumstance. Follow these steps:

1. Create a style sheet that defines the style for the node label. Save the CSS to your Web application.
2. Extend the `atg.web.service.CustomStyleFinder` class. In the new class, override the `getStyle(TreeNode pNode, String pKey)` method, coding pNode to locate the custom CSS you created.
3. Create a properties file for the Custom Style Finder component that uses the class you created. Save the properties file to the `/atg/web/service` directory of your local configuration location (such as `<ATG10dir>/home/localconfig`). Other components that refer to Custom Style Finder component will automatically use the correct implementation of it.
4. Update the `/atg/web/assetmanager/ConfigurationInfo.additionalStyleSheets` property to include the path and name of the custom CSS you created.

Removing the Checkboxes Beside Nodes

The Navigation pane tabs and views contain checkboxes used for item selection. You can hide all such checkboxes for items in a given view in the Asset Picker or Navigation pane. Follow these steps:

1. Locate a component that represents the view (or tab as is the case for the Project tab) you want to work with:
    - For User Segments view, use the `/atg/web/assetmanager/configuration/targeting/SegmentViewConfiguration` component.
    - For Targeters view, use the `/atg/web/assetmanager/configuration/targeting/TargeterViewConfiguration` component.
    - For Content Groups view, use the `/atg/web/assetmanager/configuration/targeting/ContentGroupViewConfiguration` component.
    - For Organizations and Roles view, use the `/atg/web/assetmanager/configuration/profile/OrgRoleViewConfiguration` component.
    - For Organizations view, use the `/atg/web/assetmanager/configuration/profile/OrgViewConfiguration` component.

- For Users view, use the `/atg/web/assetmanager/configuration/profile/UserViewConfiguration` component.
- For Project tab, use the `/atg/web/assetmanager/list/ProjectAssetBrowser` component.

2. Set the `editable` property to `false`.
3. Save the properties file to a path in the local configuration directory that's identical to the component's path. For the Users view, you save the properties file to:

```
<ATG10dir>/home/localconfig/atg/web/assetmanager/configuration/profile
```

Creating Views

Note that this section does not apply to Flex-based applications (ATG Merchandising, the Promotions and Coupons interface, and Site Administration). See the introduction to this chapter for more information.

A view is a mechanism that displays specific types of assets organized by folders (called list view) or by hierarchy (called tree view).

A list view displays the items you create and organizes them in the order in which they are created or manually positioned. When you have a large number of items, you can use paging controls to navigate between ranges of items.

A tree view is made up of items that have hierarchical relationships. Some items contain nested items. Others have a child-parent dependency, meaning they exist within other items of a specific type. These relationships restrict the view structure overall, although you can rearrange the order of items within a given tree branch. When a branch holds a large number of items, those items are automatically grouped into buckets that you must expand in order to view the items themselves.

To decide the type of view to create, consider whether the items are in part defined by their relationships to other items or have dependencies on other items. If so, create a tree view. Otherwise, create a list view.

Instructions for creating views are provided in the following sections:

- For list views, see Implementing a New List View.
- For tree views, see Implementing a New Tree View.

Implementing a New List View

Creating a list view is a three-step process:

1. Create a component of class `FilterableBrowseListViewConfiguration` as described below.
2. Add the view to the task configuration file so that it is visible to users. Be sure to map the view name in a resource bundle to a key that you use in the task configuration file. If you skip this step, the view will not display in the Browse tab Show drop-down list. See Changing UI Labels for instructions.

3. Customize the list as necessary. For example, you can provide a custom style to the nodes in the Navigation pane.

Filterable Browse List View Configuration Component

To organize your items in a list, create a component of class `atg.web.assetmanager.configuration.FilterableBrowseListViewConfiguration` and save it to the `/atg/web/assetmanager/configuration` directory. This class creates the list structure used by your view and implements a filtering tool that restricts the contents to items that begin with the letters you type in a text field. For instructions on creating Nucleus components in the ACC, see *Appendix C: Managing Nucleus Components* in the *ATG Page Developer's Guide*.

This section uses a view that lists manufacturers as an example. Set the following properties in this component:

Property	**Description**
`assetTypeName=manufacturer`	Holds the type of items displayed in the list. Although a list displays one item type only, it will also display that item's subtypes. Provide the name defined for the item type in the item descriptor.
`createableTypesList=/atg/commerce/catalog/SecureProductCatalog:manufacturer`	Holds the type of items you can create in the view. Provide the name defined for the item type in the item descriptor. Specify the full path to prevent naming conflicts.
`itemsPerPage=50`	Holds the number of items that are visible in a page. A higher number reduces the number of overflow pages, but may require users to use the scroll bar in order to view all items on a given page. A lower number lets users view all items without scrolling but generates more overflow pages.
`queryRQL=ALL`	Holds the query used to locate items to display in the list. A query for `ALL` indicates that all items of type `manufacturer` are returned. For a list of other queries and operators, see the *Repository Query Language* section of the *Repository Queries* chapter of the *ATG Repository Guide*.
`repositoryPath=/atg/commerce/catalog/SecureProductCatalog:manufacturer`	Holds the name and path of the repository that contains the manufacturer item descriptor.

Implementing a New Tree View

In order to create a view that organizes items into a tree hierarchy, you need to complete the following tasks. If you want to add your tree to the Asset Picker, see Creating a Tree-Based Asset Picker Plugin. Many of the resources you create and configure here are also described in the Asset Picker section.

Follow these steps:

1. Create a component of class `atg.web.tree.repository.MutableRepositoryTreeDefinition`, which defines the relationship between the repository items in the tree. See Mutable Repository Tree Definition Component for more information. There are two properties specific to the process of creating a Navigation pane tree:

   - The `cloneHelper` property holds the Nucleus component that provides the naming strategy for duplicated items.
   - The `useRootNodeAsDefaultParent` property indicates, for a tree with one type of root node, whether that node is hidden from the Browse view, but visible in the Asset Picker (`true`) or visible in both locations (`false`). If set to `true`, the root node is the default parent of newly created items.

2. Create a component of class `atg.web.tree.CountBasedBucketer`, which organizes a large numbers of items into buckets. See Count Based Bucketer Component for more information.

3. Create a component of class `atg.web.tree.BucketingTreeImpl`, which controls the display of items in the tree. See Bucketing Tree Impl Component for information.

4. Create a component of class `atg.web.tree.TreeState`, which manages tree state persistence for each user.

   To learn about the properties of Tree State, see the discussion of its subclass, `atg.web.assetpicker.tree.repository.ContextClearingTreeState`, provided in the Tree State Component section. All properties described there are defined in Tree State.

5. Update the `atg/web/assetmanager/TreeRegistry` component as described in the Tree Registry Component section.

6. Create a Browse Tree View Configuration component, which manages the other tree-related components and specifies the types of items you can create in this view. See below for instructions.

7. Add an entry for the new view to the task configuration file. You can find code that implements a fictitious Manufacturer view in the Changing UI Labels. For general information about task configuration files and the tags they contain, see Customizing the UI for Specific Tasks and Activities.

You can define your tree further by providing additional configuration described in the Creating a Tree-Based Asset Picker Plugin. All options available to the Asset Picker dialog box are available to tree views in the Browse tab.

Browse Tree View Configuration Component

To organize your items in a tree hierarchy view, create a component of class `atg.web.assetmanager.configuration.BrowseTreeViewConfiguration`, and save it in a directory that holds other resources for that view, such as the `/atg/web/assetmanager/configuration` directory. For instructions on creating Nucleus components in the ACC, see *Appendix C: Managing Nucleus Components* in the *ATG Page Developer's Guide*.

For the purpose of this instruction, assume that you are creating a view that lists assortments. The component you create is called Assortments Browse Tree View Configuration. Set the following properties on this component:

Property	Description
`createableTypes`	Holds the list of item types that can be created in this view, generated based on the values from the `createableTypesList` property. This property is read-only.

`createableTypesList=\`
`/atg/commerce/catalog/Assortments,\`
`atg/commerce/catalog`
`/SecureProductCatalog:products,\`
`atg/commerce/catalog`
`/SecureProductCatalog:skus`
	Holds the list of item types that can be created in this view. Note that you can include the item descriptor name only, but you're encouraged to specify the full path to safeguard against naming conflicts. Leaving this property empty permits users to create items of all types that are made visible to the tree by the Mutable Repository Tree Definition component. The list that appears in the Create New drop-down list is automatically alphabetized.

`treeComponentName=/atg/commerce/web`
`/browse/AssortmentTreeState`
	Holds the component that configures the version of the tree that displays in tabs such as the Browse tab.

Preventing the Creation of Items

Note that this section does not apply to Flex-based applications (ATG Merchandising, the Promotions and Coupons interface, and Site Administration). See the introduction to this chapter for more information.

In general, each view provides resources that manage the types of items you can create in it. When you click the Create New button in the asset editing UI, it displays a list of item types from which you select the type you want to create. Usually you specify the types in this list by explicitly identifying them in a `createableTypesList` property of a Browse Tree View Configuration (tree views) or Filterable Browse List View Configuration (list views) component. If you leave these properties empty, all types visible in the view are included in the Create New list.

You can prevent a type from appearing in the Create New list of all views, regardless of settings on an individual view, by updating the `nonCreatableTypes` property of the Configuration Info component.

Note that this setting prevents you from creating items using Navigation pane controls, but it does not affect Details pane controls with the same purpose. For example, you can remove roles from the Create New drop-down list in the Organizations and Roles view, but you can still create roles using the Add New button beside the Roles property that displays in the Orgs & Roles tab when a user is selected.

Follow these steps to prevent an item type from appearing in the Create New list:

1. Create a file called `ConfigurationInfo.properties` and save it to `/atg/web/assetmanager` within your local configuration directory, such as: `<ATG10dir>/home/localconfig/atg/web/assetmanager`.
2. Set the `nonCreateableTypes` property to the types of items you want excluded from the Create New button:

```
$class=atg.web.assetmanager.AssetManagerConfiguration

nonCreateableTypes=\
  /atg/userprofiling/PersonalizationRepository:folder
```

Configuring Support for Repository Customizations

The repository item types that are provided with ATG applications can be modified to meet the needs of your Web sites. When you customize item types and subtypes, you typically need to configure the Asset Manager to display them. **Note:** Although item types that are versionable are called asset types in ATG Business Control Center, the term item types is used here because this discussion applies to both versionable and nonversionable types. See the following sections for instructions:

Supporting Custom Item Types

Supporting Custom Item Subtypes

Supporting Custom Properties For Item Types

For information on repositories, item types, and properties, see the *ATG Repository Guide*.

This section does not apply to Flex-based applications (ATG Merchandising, the Promotions and Coupons interface, and Site Administration). See the introduction to this chapter for more information.

Supporting Custom Item Types

A custom item type is a repository item type created from scratch. An example of a custom item type is an alert that displays the weather for a specific place and time.

To make a custom item type visible in Asset Manager applications:

1. Create your custom item type based on instructions provided in the *SQL Content Repository* chapter of the *ATG Repository Guide*.
2. Configure the item type for the versioned database by adding new properties to the repository, adding columns for those properties in the appropriate database tables, and informing the versioning system about the new item type.

   For more information, see the *Configure Repository Asset Support* section of the *Setting Up an Asset Management Server* chapter in the *ATG Content Administration Programming Guide*.
3. Add the new type to the Browse tab views. See *Modifying Tree Views* section of the *Configuring the Merchandising UI* chapter of the *ATG Merchandising User Guide*.
4. Make the new type searchable in the Search tab. See Creating and Defining a Task Configuration File.
5. Create item mappings that control which properties appear for the item type in the Details pane in a given context. See Custom Item Types and View Mappings.
6. Create property groups so that you can use the Multi Edit tab to edit items in bulk. See Creating and Defining Multi Edit Property Groups.
7. A default icon represents all custom repository items in the Navigation pane. If you provide a different icon, see the Implementing a Custom Icon section for instructions.

Items of a custom type are automatically visible in the Project tab, Multi Edit tab, and all relevant dialog boxes.

Supporting Custom Item Subtypes

A custom item subtype is an item type based on an existing item type. For example, you could create an external user subtype called representative that describes users who act on behalf of a company to sell items in specialty parties.

To use a custom item subtype in Asset Manager applications:

1. Create your custom item subtype according to the instructions in the *SQL Content Repository* chapter of the *ATG Repository Guide*.
2. Configure the subtype for the versioned database by adding new properties to the repository, adding columns for those properties in the appropriate database tables, and informing the versioning system about the new item type.

   See the *Configure Repository Asset Support* section of the *Setting Up an Asset Management Server* chapter in the *ATG Content Administration Programming Guide* for instructions.
3. Add the new subtype to the searchable list of item types in the Search tab. This step is optional; by default, a search on a parent type also returns sub-typed values. Perform this task if you want the subtype to be visible in the item types list and you want users to be able to locate only items of this subtype. See Creating and Defining a Task Configuration File.

4. Create new item mappings for the item subtype. This step is necessary only if the item mappings defined for the parent type are not appropriate for the subtype. See Custom Item Subtypes and View Mappings.

5. If you created new item mappings or made any other changes that affect which item mapping should be used for an item of this type, be sure to represent those changes in the task configuration file. See Updating View Mapping Settings in a Task Configuration File.

6. Create property groups so that you can use the Multi Edit tab to edit items in bulk. See Creating and Defining Multi Edit Property Groups.

7. By default, the icon that appears for items of the parent item type is also used for the custom subtype. See the Implementing a Custom Icon section for instructions on changing the icon.

Items of a custom subtype are automatically visible in the Project tab, Multi Edit tab, and all relevant dialog boxes.

Supporting Custom Properties For Item Types

Custom properties are new properties that you add to an existing item type. For example, you could add Referral Source and Age properties to the external user profile.

Depending on how your item mappings are configured, custom properties may automatically display in the Details pane when an item of that type is selected. For information on how item types specify properties, see the Specifying Properties to Display section. Locate the parent item view mapping and item view mapping and adjust their settings accordingly.

If necessary, update property groups so the new properties are visible to the Multi Edit tab. See Creating and Defining Multi Edit Property Groups.

Controlling Details Pane Properties

This section does not apply to Flex-based applications (ATG Merchandising, the Promotions and Coupons interface, and Site Administration). See the introduction to this chapter for more information.

The View Mapping system is a repository that contains items used to assemble the Details pane for a given item in a particular context. For example, the View Mapping system controls display of Details pane properties for a content group. The View Mapping system is described the *Customizing Asset Display* chapter of the *ATG Content Administration Programming Guide*.

The Asset Manager extends the View Mapping system to include `itemMappings` that support properties of personalization items. It uses these `itemMappings` to construct the Details pane and display item properties as follows:

1. At runtime, a JSP interacts with the parsed task configuration file to determine the appropriate mapping mode. For more information, see How the Task Configuration File is Processed.

2. The JSP passes the map mode, item type, and item mapping name to the View Mapping system.
3. The View Mapping system returns an `itemMapping` object, which contains the underlying objects that are used to compose the item's Details pane: `itemViewMappings` and `itemViews`, `propertyViewMappings`, `propertyViews`, and form handler items.

In the ACC, `itemMappings` that are used by the Asset Manager typically have display names that are prefixed `Asset Manager`. These `itemMappings` support items that are specific to ATG Merchandising and ATG Personalization. When you create custom view mapping objects, adopt a similar convention so all of your objects are easily recognized as your customizations.

Note: The term "view mappings" refers in general to items in the View Mapping repository.

Where to Find and Create View Mapping Repository Items

When you configure your database, you import the `viewmapping.xml` files that contain the view mapping items necessary for running your applications. You can see view mapping items in the ATG Control Center under **Publishing > View Mapping**. When you want to modify existing items, locate them in the ACC. Creating new items is also performed in the ACC as described in the *Creating View Mapping Repository Items* section of the *Customizing Asset Display* chapter in the *ATG Content Administration Programming Guide*.

The process for creating item view mappings used by Asset Manager applications is slightly different from the one recommended for ATG Content Administration. See the instructions below.

Creating Item View Mappings

When you create an item view mapping for an Asset Manager application, you must specify its display name in a resource bundle. Using resource bundles makes it possible for you to localize resources; it also simplifies the process of changing UI labels. Follow these steps:

1. In a custom resource bundle, include a key and value pair for the item view mapping (key) and its display name (value). For example: `customtab=Manufacturing`. For more information on resource bundles, see Changing UI Labels.

   Note: It is recommended that you create a new resource bundle that contains your implementation-specific information rather than modifying one that came with an ATG application. Resource bundles should exist in a directory such as `<ATG10dir>/home/locallib` that automatically adds files to your CLASSPATH.

2. Create an item view mapping by following the steps provided in the *Creating View Mapping Repository Items* section of the *Customizing Asset Display* chapter in the *ATG Content Administration Programming Guide* with the following modifications:

   - Add a new attribute value to the item view mapping. To do so, click the Attribute Values text box, then the **...** button beside it. In the Attribute Values dialog box, click New Item, set the Value property to the path and name of your resource bundle, and then click OK. When prompted for a name for the new entry, enter `resourceBundle`, then click OK.

- In the Display Name text box, enter the resource key that you used in the resource bundle (`customtab`).

Modifying the Default View Mappings

There are four ways to control the display of item properties:

- Add or remove properties from a given tab. See Specifying Properties to Display.
- Re-arrange the order of properties on a given tab. See the *Sorting Properties* section of the *SQL Repository Item Properties* chapter of the *ATG Repository Guide*.
- Make a property read-only in an application. To do this, create a property view mapping for the property and set the property view mapping to refer to a read-only mode property view (`AssetManager.view`).
- By default, all item types are available in view and edit mode. If you want to restrict the item types visible in one mode, adjust the item types used in conjunction with an item mapping and map mode in the task configuration file. See Updating View Mapping Settings in a Task Configuration File section for instructions.

Custom Item Types and View Mappings

When you create an item type, consider how you want the item's properties to display in the Details pane. Options include displaying properties on tabs; making some properties read-only; and displaying only a subset of the properties. If your item type requires none of these configurations, you can use the default item mapping defined for all item types, which displays all properties in the Details pane when an item of that type is selected.

If you want to configure the property display, create an item mapping and other view mapping resources as follows:

1. Create two item mappings for each custom item type: one that works in conjunction with `AssetManager.view` mode and one that works in conjunction with `AssetManager.edit` mode. When a user requests a page that includes the custom item type, one of these item mappings is returned containing the appropriate other view mapping objects.
2. Provide values to the properties as necessary. Set the `formHandler property` to `Asset Manager AssetRepositoryFormHandler`. Save your changes.
3. If you want to split the display of the item's properties across several tabs, create one item view mapping for each mode the tab supports. Then add the item view mappings of the edit map mode in the order they should appear in the Details pane. Add them to the `viewMappings` property of the item's edit map mode item mapping. Do the same for the view map mode item view mappings and item mapping.
4. Specify the properties that should display as described in Specifying Properties to Display.
5. Update the task configuration file to support your new item mapping as described in Updating View Mapping Settings in a Task Configuration File.

Custom Item Subtypes and View Mappings

Item subtypes use the view mapping resources defined for their parent item types automatically. If your subtype is appended to an out-of-the-box item type, its new properties will display by default on the first tab that you see when an item of that type is selected. If the item doesn't use tabs, the properties will appear along side the other properties in the Details pane.

For subtypes of custom item types, examine the parent types' item mappings and item view mappings:

- Locate the item mappings named for the item type. If you can't find them, the item uses the default item mapping for all item types, and all the subtypes' properties automatically display in the Details pane.
- If you find item mappings, check the `attributes` property for restrictions on the properties that can display. For information on displaying properties, see Specifying Properties to Display.
- Also, check the `viewMappings` property of the item mapping to see if it has any item view mappings. If it does, the item's properties are split across tabs defined by the listed item view mappings. Review the `attributeValues` property of each item view mapping to see if your new properties are added automatically. If not, add them manually. See the Specifying Properties to Display section.

For instructions on making other changes, see Custom Item Types and View Mappings.

Creating and Defining Multi Edit Property Groups

A property group contains a set of properties defined for a given item type that business users can edit at the same time in the Multi Edit tab. There are no property groups provided by default, so if you want to edit items in bulk, you need to define one or more property groups yourself.

Before you create property groups, consider how they will be used. Typically the property groups you create should be appropriate for a single task performed by the same user.

Properties can be part of as many property groups as you like. When you build your property groups, keep in mind that you can make some properties read only as a point of reference, even if they are editable in other tabs. See below for instructions.

If a property group is designed for a given workflow task, you can configure it to display only to the users performing tasks that require it. By default, all property groups are available to all users. When users can see a property group, it is visible to Step Edit, List Edit and Apply to All modes.

Consider a property group that lets users update the auto-applied roles and organizational roles in Step Edit mode. The steps below show how to create such a property group:

1. Create one item mapping for organizations that supports the multi edit map mode. Provide values to the following properties as follows:
   - In the Name text box, provide a name that identifies your custom property group. The name is used by the task configuration file to determine the circumstances in which the property group is visible to users. If all property groups should be visible at the same time, provide the same name to all

property groups. Note that the item mapping is identified in the list to the left by an automatically generated ID, not by the name you are providing for it now.

- In the Map mode text box, specify `AssetManager.multiEdit`.
- In the Item name text box, enter the item type or subtype. In this case, enter `organization`.
- In the Item path text box, enter the path and name to the repository that holds the item type or subtype you entered in the Item name property. In this case, enter `/atg/userprofiling/ProfileAdapterRepository:organization`.
- In the Form handler text box, enter the default form handler, `AssetManager AssetRepositoryFormHandler`.

Note: In general you need one item mapping per item type that supports the multi edit map mode. If your item type has subtypes, you can also create an item mapping for each subtype, so your property groups can be tailored to your subtypes.

2. Click OK.
3. Create an item view mapping for each property group.
   - In the Name text box, provide a name that identifies your property group. Note that the item mapping is identified in the list to the left by an automatically generated ID not the name you are providing for it now.
   - In the View text box, provide the name of the item view used in the item view mapping for this item type with `AssetManager.edit` mode.
4. Click OK.
5. Identify the properties that should display in the property group as described in the Specifying Properties to Display section. In this case, you use the `specificProperties` key and define an object to hold `roles` and `relativeRoles` properties.
6. Add the name of the new item view mapping to the `viewMappings` property in the new item mapping.
7. Click Save.

To make editable properties read only when viewed in a property group, follow these steps:

1. Create the property group as shown above.
2. In the Property mappings text box, enter the property view mapping used for items of that type in `AssetManager.view` mode. The best way to find the property view mapping is to locate the item view mapping used for items of that type in view mode and check the `propertyMappings` property.
3. Click OK.

Specifying Properties to Display

All properties defined in an item descriptor can display in the Details pane when an item of that type is selected. For out-of-the-box items, a subset of those properties display. You can modify the properties that display for existing items, or specify the list of properties that you want to display for custom or customized items.

There are two ways to manage the display of properties.

- Control the properties visible on a given tab using an item view mapping.
- Prevent properties from appearing anywhere in the UI using an item mapping.

By default, all properties defined in the custom item descriptor will display.

Regardless of where you specify properties (item view mapping or item mapping), the process is the same: you update a map property by setting the type of action you want to perform as the key and creating a component that holds the actual property names as the value. For item view mappings, the map property is called `attributeValues`; for item mappings the map property is called `attributes`.

Use the following keys in an item view mapping `attributeValues` property to indicate properties that you want to be visible in the Details pane:

- `specificProperties` maps to an object containing properties that should display.
- `categorizedGrouping` maps to an object containing categories that have properties that should display. Categories are groupings of properties defined in an item descriptor.
- `showNewProperties` maps to an object that holds a `true` value to indicate that any properties you added to the application should display by default. This setting is configured for all out-of-the-box item view mappings that represent the first tab for an item type. Add this setting to custom item view mappings when appropriate.

To prevent properties from displaying, set one of the following keys in the item view mapping `attributeValues` property or the item mapping `attributes` property of an item mapping:

- `excludedProperties` maps to an object containing properties that won't display, even if they are explicitly permitted by another setting, such as `categorizedGrouping` or `showNewProperties`.
- `excludedCategories` maps to an object containing categories with properties that won't display, even if they are permitted by another setting, such as `specificProperties`.

Note that you can specify as many keys as you want for an item view mapping and item mapping. If you indicate excluded properties only, all other properties on a given item descriptor will display. The reverse is true when you designate properties for inclusion. If one key includes a property that's excluded by another, the exclusion setting prevails.

When you specify properties, use the name defined for the property in the item descriptor, not the display name you see in the Details pane. You can find a description of item types in the *ATG Personalization Programming Guide*.

Making New Properties Visible on the First Tab: Item View Mapping Example

If you add new properties to the external users item descriptor and want to be sure that all new properties display on the first tab automatically, perform the following tasks:

1. In the ACC, open the Publishing > View Mapping task area.
2. In the Items of Type drop-down list, select `itemViewMapping`, and click **List**.
3. From the list of item mappings, click the name of the item view mapping you created, such as AssetManager Editable Mapping for External User Items.
4. Click the text box beside the `attributeValues` property, then click the "…" button.
5. In the Attributes dialog box, click Add. You need to provide a key and a value to the map. First, you'll provide an object that represents the value.
6. In the New Item dialog box, click New Item. Always create an object when you are providing a value to a new key.
7. Beside the Value property, enter `true`. Click OK.
8. In the New Dictionary Entry dialog box, enter `showNewProperties` as the key. Click OK.
9. Once you see the new key and value pair in the Attributes dialog box, click **OK** to save your changes.

Hiding a Category's Properties: Item Mapping Example

Assume that you created several new external user properties in a category you called `MyCategory` and you don't want any of those properties to display when the Details pane is editable. Because you want custom properties in other categories to display, and this tab permits new properties, you need to disallow the properties in `MyCategory`. The steps below show how to adjust your view mapping settings:

1. In the ACC, open the Publishing > View Mapping task area.
2. In the Items of Type drop-down list, select `itemMapping`, and click List.
3. From the list of item mappings, click AssetManager editable mapping for external user items. Notice that each item mapping is named for an item type and map mode.
4. Click the text box beside the attributes property, then click the "…" button.
5. In the Attributes dialog box, click Add. You need to provide a key and a value to the map. First, you'll provide an object that represents the value.
6. In the New Item dialog box, click New Item. Always create an object when you are providing a value to a new key.
7. Beside the Value property, enter `MyCategory`. Click OK.
8. In the New Dictionary Entry dialog box, enter `excludeCategories` as the key. Click OK.
9. Once you see the new key and value pair in the Attributes dialog box, click OK to save your changes.

Controlling the Display of IDs

Although names are used to identify items in the ATG Business Control Center, each item has an ID that's also visible. If you want to prevent users from viewing or editing (where editing is permitted) repository IDs, change the `displayID` property of the `/atg/bizui/Configuration` component to `false`.

Note that the repository definition for the ID property takes precedence over this setting; even if displayID is set to true, if the repository definition specifies expert=true or hidden=true, IDs are not visible.

If the writable attribute is set to true in the repository definition, users can edit the IDs as well as view them (assuming displayID is not overridden by the repository definition). However, users can edit the ID field only until they click “Apply,” which saves the asset. After that, the ID property cannot be edited.

Updating View Mapping Settings in a Task Configuration File

A task configuration file determines the map mode used for each item type and item mapping combination. By default, when a project is present, all items are editable. In this case, the map mode is AssetManager.edit, the item type is set to a * (indicating all types), and the item mapping is Asset Manager.

The case is slightly more complicated for activities. An activity involving an nonversionable item, such as a user profile, uses the same settings as a project. For versionable assets, the item type and item mapping values are the same, but the map mode restricts users to view-only mode because AssetManager.view is used. To learn more about the default configuration, see the Examining the AssetUI Task Configuration File and Examining the DPS-UI Task Configuration File sections. Familiarize yourself with the view mapping settings in the task configuration file before proceeding with this discussion.

There are two customization options available. Rather than keeping the same behavior for all item types, you could make some assets view-only and others editable. To implement this for the AssetManager.view mode, replace the current item type value (*), which applies the settings to all types, with a listing of each item you would like to appear as view-only. Keeping the editable map mode set to * applies that map mode to all remaining item types:

```
<view-mappings>
  <view-mapping mode="AssetManager.view">
    <item-mapping>
      <item-type>/atg/userprofiling/ProfileAdapterRepository:user</item-type>
      <item-type>/atg/userprofiling/ProfileAdapterRepository:role</item-type>
      <item-mapping-name>Asset Manager</item-mapping-name>
    </item-mapping>
  </view-mapping>
<view-mapping mode="AssetManager.edit">
    <item-mapping>
      <item-type>*</item-type>
      <item-mapping-name>Asset Manager</item-mapping-name>
    </item-mapping>
  </view-mapping>
</view-mappings>
```

It is more likely, however, that your custom item types will have custom item mappings that need to be implemented in a task configuration file so items of those types are visible in the Asset Manager parts of the ATG Business Control Center. For example:

```
<view-mappings>
  <view-mapping mode="AssetManager.edit">
    <item-mapping>
      <item-type>/atg/userprofiling/ProfileAdapterRepository:guest
      </item-type>
      <item-mapping-name>MyItemMapping</item-mapping-name>
    </item-mapping>
  </view-mapping>
</view-mappings>
```

The code samples included in this section assume you are customizing view mapping settings for a specific task, workflow, or activity. To modify the view mapping settings for the default task or activity, you use a slightly different syntax. Both methods are described in Creating and Defining a Task Configuration File.

9 Customizing the UI for Specific Tasks and Activities

Note that the information in this chapter does not apply to Flex-based applications (ATG Merchandising, the Promotions and Coupons interface, and Site Administration).

You can define some areas of the ATG Business Control Center so that the UI displays only the controls necessary for the actions you need to perform. For example, the Browse display for Targeting and Segmentation displays fewer options than the UI that appears when you use the Targeting and Segmentation link to create or edit items. Some buttons and tabs are hidden from users in Browse mode.

This behavior is defined through task configuration files, two of which are included with the ATG Business Control Center. One task configuration file, located in the `AssetUI` module, provides the general appearance for default activities. These settings are inherited by tasks and activities in the second task configuration, located in the `DPS-UI` module, that represents Personalization resources. The link you use to access ATG Personalization is associated with settings defined in this task configuration file:

- An "activity" permits you to access parts of the UI that are not associated with version control.
- A "task" is an action you perform that creates or manages versions of assets.

Using a task configuration file, you can alter the default configurations or tailor the UI specifically for the following contexts:

- All tasks in a particular workflow
- A particular activity
- A specific task in a particular workflow
- A specific task, which may appear in multiple workflows

A task configuration file can specify values for the following elements:

- Parts of the UI that are rendered through JSPs
- Tabs in the Navigation pane, which include specifying tab ordering, the first tab to display, and related resource bundle information as well as views and buttons that appear in each tab.
- Views that display in tabs, which include ordering, the first view to display, view buttons, Nucleus configuration components, and related resource bundle information.

- Whether the UI is view-only or editable for the items it displays. This setting is provided by item mappings that correspond with a given view mapping mode and item type.

ATG takes the default configuration file and combines it with your custom configuration file to determine the settings to use for tasks, workflows, and activities.

This chapter contains the following sections:

Task Configuration and Secured Repositories

As described in the introduction to this chapter, task configuration files control access to the tabs, views, buttons and other pieces that make up the UI. Although you can use the task configuration file to control how users interact with data (for example, by removing buttons to prevent users from editing assets such as content groups), the data itself is visible to all users. Secured repositories, by contrast, control a user's access to data, for example by hiding items from view if a user does not have the appropriate permissions.

The secured repositories feature provides item-level security through access control lists, which you use to grant and deny read, write, and other permissions to internal users. The restrictions can be applied by profile, role, or organization. You can also use secured repositories to manage access to item descriptors and individual properties.

Secured repositories can also provide high-level restrictions by hiding links in the Operations list on the ATG Business Control Center home page as well as controlling access to workflows. For information, see the *Managing User Access and Security* chapter of the *ATG Content Administration Programming Guide*.

Exercise caution when configuring access to areas of the UI that allow users to search for items. Users may be able to search for a particular type of item, such as a content group, but no items of that type will be returned unless the user has appropriate access rights (view).

How the Task Configuration File is Processed

When the ATG Business Control Center starts, the Task Configuration Manager component locates the default task configuration file, `taskConfiguration.xml`, as well as any custom task configuration files you have created. All task configuration files managed by a given Task Configuration Manager are collapsed into one file using a process called XML File Combination. During this process, XML File Combination determines the tags to use from the default configuration file and the tags to override with values provided by a custom file. Then, the Task Configuration Manager parses the combined file, organizing the information so that when a page is requested, the ATG Business Control Center can determine what UI elements to display based on the current activity, task, or workflow.

For more information about XML File Combination, refer to the *XML File Combination* section of the *Nucleus: Organizing JavaBean Components* chapter in the *ATG Programming Guide*.

You can use more than one Task Configuration Manager if you want one or more task configuration files to be combined together without incorporating settings from other task configuration files. You may find it easier to have one Task Configuration Manager per application to reduce the scope of tag inheritance.

The tags in the task configuration file are evaluated as follows:

- When there are general settings that apply to all activities, those settings are enclosed in a `<default-activity>` tag. To define the UI appearance for a particular activity, you specify the activity's ID and, if you want the activity to inherit settings from another activity, in addition to the default, you include that activity's ID as well. Inherited settings can be overwritten – replaced, removed, or added to – by new settings you supply in a custom task configuration file.
- General settings for tasks and workflows may be specified using the `<default-task>` tag, which inherits settings from the `<default-activity>` tag. Similar to activities, you can specify independent settings for particular workflows and tasks, inherit settings from other resources or, most commonly, use a combination of both methods.

  Keep in mind that a workflow is made up of tasks. If you define some settings on the workflow-level and others on the task-level for tasks in that workflow, both sets of settings are used unless a conflict occurs, in which case the task setting is used.

Changes to settings for existing activities, tasks or workflows as well as new settings for other ones are specified in a custom task configuration file. It's likely your application will use multiple task configuration files and have several layers of inheritance defined within them. The following example demonstrates how inheritance works. Consider a Replace Small Images task, which is part of the Image Update workflow. If settings are provided to each of the following entities, they are applied in this order:

- Replace Small Images task
- Image Update workflow, which specifies the `assetManager.defaultBrowse` activity as a resource from which it inherits settings
- `assetManager.defaultBrowse activity`
- `<default-task>`
- `<default-activity>`

Examining the AssetUI Task Configuration File

It's a good idea to familiarize yourself with the default configuration settings so you can learn what parts of the UI are configurable and how to work with the tags in the task configuration file. Keep in mind that the default task configuration files cover a subset of the configuration options available to you.

As mentioned earlier, the ATG Business Control Center comes with two task configuration files. The file in the AssetUI module provides basic settings; the file in DPS-UI module defines the appearance for additional activities and tasks. You can find the AssetUI taskConfiguration.xml in the <ATG10dir>/AssetUI/config.jar/atg/web/assetmanager/configuration directory.

The contents of the file are enclosed in a <task-configuration> tag. Inside it, the file is divided into three sections: one enclosed in a <default-activity> tag and two enclosed in <activity> tags.

The <default-activity> tag encloses the default resource bundle that contains the labels you see in the UI. An <activity> tag defines settings for a particular activity, which are combined with those provided by the <default-activity> tag. An activity can involve any operation, such as browsing targeters or modifying user items, as long as it doesn't produce or affect versioned items. There are two types of activities:

- Activities that display items in view-only mode. For example, clicking the Segments and Targeters Operations list link uses the personalization.browseSegmentsAndTargeters activity to access targeters, content groups, and user segments in view-only mode.
- Activities that display nonversionable repository items in edit mode. For example, clicking any of the Users options under Operations accesses the personalization.users activity to display profiles in edit mode.

The two activities defined in this task configuration file provide the foundation for the two types of activities used to access the Personalization parts of the ATG Business Control Center. The first <activity> tag defines the structure of the UI—a JSP and a configuration component—and sets the mode to view-only. The second activity (assetManager.defaultEdit) inherits settings from the first (assetManager.defaultBrowse), but replaces view-only mode with edit.

The purpose of activities like these is to define settings that other activities may inherit. Generic activities exist as a mechanism for holding settings inherited by other resources and are not visible in the ATG Business Control Center. The assetManager.defaultBrowse is the baseline for activities that provide view-only access to versionable items, and assetManager.defaultEdit is the baseline for activities that provide editable access to nonversionable items.

Some of the many configuration values specified in the default task configuration file are pointers to files used by various parts of the UI. All JSPs are located in the AssetManager.war file that you find in <ATG10dir>/AssetUI/j2ee-apps/AssetUI.ear. All other file locations are specified below.

Default Activity Settings

As mentioned earlier, the <default activity> tag defines one tag that applies to all activities, tasks, and workflows. Other resources can overwrite this setting.

```
<default-activity>
    <resource-bundle>
      atg.web.assetmanager.WebAppResources
    </resource-bundle>
</default-activity>
```

The main resource bundle is called `atg.web.assetmanager.WebAppResources`. Its properties file is located in `<ATG10dir>/AssetUI/lib/classes.jar`.

assetManager.defaultBrowse Settings

The `assetManager.defaultBrowse` activity maintains general settings that other activities inherit. This activity provides view-only access to items.

Defining General Activity Settings

The file begins as follows:

```
<activity id="assetManager.defaultBrowse">
   <page>
     /assetManager.jsp
   </page>
   <asset-editor>
     <page>
       /assetEditor/editAsset.jsp
     </page>
   </asset-editor>
   <configuration>
     /atg/web/assetmanager/ConfigurationInfo
   </configuration>
```

The first set of tags identify the page that provides the structure to the Asset Manager UI Framework, `assetManager.jsp`. Among other settings, `assetManager.jsp` defines a two-panel structure for the UI. The structure for the Details pane is determined by `editAsset.jsp`. The `ConfigurationInfo` component specifies the Task Configuration Manager as well as other resources used by the Asset Manager framework to generate the UI. Note that both `AssetUI` and `DPS-UI` files use the same Task Configuration Manager, which means an inheritance dependency exists between them. Any other uses of a `<configuration>` tag in these files provide view-specific settings.

Defining View Mappings

The view mapping settings specify item mappings that control the display of properties in the Details pane. They also let you override the view mapping mode. Specify one `<view-mapping>` tag for each `<item mapping>` tag. These tags identify the UI resource (item mapping name) and type of access (view mapping mode) provided to properties of the specified item types.

Keep in mind that sometimes you may want to provide view-only access to some items and edit access to others. For example, you might want an activity to provide read-only access to items that are otherwise editable. To accommodate this requirement, the JSP uses the task configuration file to assign the map mode based on the item type, giving you the option to pass a different map mode to the JSP.

The following view mappings are specified in `assetManager.defaultBrowse`:

```
  <view-mappings>
    <view-mapping mode="AssetManager.edit">
      <item-mapping>
        <item-type>*</item-type>
        <item-mapping-name>AssetManager</item-mapping-name>
        <view-mapping-mode-override>AssetManager.view</view-mapping-mode-override>
      </item-mapping>
    </view-mapping>
    <view-mapping mode="AssetManager.multiEdit">
      <item-mapping>
        <item-type>*</item-type>
        <item-mapping-name>AssetManager</item-mapping-name>
        <view-mapping-mode-override>AssetManager.view</view-mapping-mode-override>
      </item-mapping>
    </view-mapping>
    <view-mapping mode="AssetManager.view">
      <item-mapping>
        <item-type>*</item-type>
        <item-mapping-name>AssetManager</item-mapping-name>
      </item-mapping>
    </view-mapping>
    <view-mapping mode="AssetManager.diff">
      <item-mapping>
        <item-type>*</item-type>
        <item-mapping-name>AssetManager</item-mapping-name>
      </item-mapping>
    </view-mapping>
    <view-mapping mode="AssetManager.conflict">
      <item-mapping>
        <item-type>*</item-type>
        <item-mapping-name>AssetManager</item-mapping-name>
      </item-mapping>
    </view-mapping>
  </view-mappings>
</activity>
```

When `AssetManager.edit` or `AssetManager.multiedit` is passed from the JSP, the task configuration specifies the item mapping as Asset Manager and changes the mode from edit to view, which prevents buttons like Save from appearing in the UI. Thus the UI is in view-only mode regardless of the map mode initially received by the task configuration file.

Note the use of * as a wildcard value to indicate all items.

assetManager.defaultEdit Settings

Like the `assetManager.defaultBrowse` activity, the `assetManager.defaultEdit` activity provides general settings for activities that let users view and edit nonversionable items in the ATG Business Control Center.

The following tags specify the activity and its relationship to others:

```
 <activity id="assetManager.defaultEdit"
inherit-from="assetManager.defaultBrowse">
```

The first attribute in this tag identifies the resource for which all settings enclosed in the `<activity>` tag apply. The second attribute indicates the resource from which this activity inherits settings. Tags supplied by `assetManager.defaultEdit` then contribute additional settings, which are described below. These tags are designed to overwrite the view-only restriction of the view mapping tags in the `assetManager.defaultBrowse` activity:

```
<view-mappings>
      <view-mapping mode="AssetManager.edit" combine="replace">
        <item-mapping>
          <item-type>*</item-type>
          <item-mapping-name>AssetManager</item-mapping-name>
        </item-mapping>
      </view-mapping>
      <view-mapping mode="AssetManager.multiEdit" combine="replace">
        <item-mapping>
          <item-type>*</item-type>
          <item-mapping-name>AssetManager</item-mapping-name>
        </item-mapping>
      </view-mapping>
    </view-mappings>
  </activity>
```

To determine the item properties to display, the JSP specifies a view mapping mode based on the whether a project is present and editable (`AssetManager.edit`), or is not present or in a view-only state (`AssetManager.view`). The view mapping settings provided for `assetManager.defaultBrowse` are used in all cases except when the JSP specifies `AssetManager.edit` or `AssetManager.multiEdit` modes, in which the input mode is also the output mode. The tags here replace the tags for `AssetManager.edit` and `AssetManager.multiEdit` modes in the `assetManager.defaultBrowse` activity because this activity is designed to provide edit access to the ATG Business Control Center for items that are nonversionable.

Examining the DPS-UI Task Configuration File

The second task configuration file is part of the ATG Business Control Center `DPS-UI` module.

The DPS-UI `taskConfiguration.xml` file is located in the `<ATG10dir>/DPS-UI/config.jar/atg/web/assetmanager/configuration` directory. `AssetUI` provides the basic components used by the Asset Manager to configure all activities, tasks, and workflows. The `DPS-UI` file inherits settings from `AssetUI` and applies them to actual resources that are visible in the ATG Business Control Center.

- The personalization.users activity corresponds to the Users links in the ATG Business Control Center Operations list.
- The personalization.browseSegmentsAndTargeters activity corresponds to the Browse link next to the Targeting and Segmentation option in the Operations list.
- The personalization.editSegmentsAndTargeters task is a generalized task from which other resources inherit settings. This task is not visible in the ATG Business Control Center
- The Personalization/editSegmentsAndTargeters.wdl workflow corresponds to the Segments and Targeters workflow that is accessed when you click the Targeting and Segmentation link in the Operations list.

personalization.users Activity Settings

The activity definition starts as follows:

```
<activity id="personalization.users" inherit-from="assetManager.defaultEdit">

    <activity-name>
      personalization.users
    </activity-name>
```

The activity ID indicates the activity to which the enclosing tags apply. This activity defines the functions available to users who enter the Asset Manager by clicking the Users links in the Operations list. This tag also specifies the resource, assetManager.defaultEdit, from which this activity inherits settings. Note that an activity can inherit settings from another activity, and a task or workflow can inherit from all three types of resources.

In this case the name of the activity is the same as the activity ID. These two identifiers are independent of each other. The ID is defined in the task configuration file and is not used elsewhere, whereas the name corresponds to the name given to the activity in the Asset Manager configuration file. Activities created for the purpose of holding settings inherited by other resources do not have names.

In this case there is no <configuration> tag that provides information about the Task Configuration Manager in this activity or any other resource configured by this file. When this information is omitted, the settings provided for the default Task Configuration Manager are used. The default Task Configuration Manager is /atg/web/assetmanager/configuration/ConfigurationManager.

See the sections below for an explanation of the remaining tags in the file.

Defining Common Buttons and General Tab Settings

Common buttons are defined as follows:

```
<operations>
  <operation>create</operation>
  <operation>move</operation>
  <operation>duplicate</operation>
  <operation>delete</operation>
  <operation>addToMultiEdit</operation>
</operations>
```

The operation tags specify the buttons that appear on the Navigation pane toolbar. Unlike other screen elements, the placement and ordering of buttons is defined in the JSP, not determined by the `<operations>` tag. The buttons named here are subset of the list of buttons available in the Asset Manager.

The next section specifies tabs:

```
<tabs>
   <tab-order>
     <tab-id>browse</tab-id>
     <tab-id>search</tab-id>
     <tab-id>multiEdit</tab-id>
   </tab-order>
   <initial-tab>
     browse
   </initial-tab>
```

The `<tabs>` tags correspond to the first organizational screen elements in the Navigation pane, which are defined in `assetManager.jsp`. The first such element is defined as a tab of which three are specified: Browse, Search, and Multi Edit. These tabs display in the order they are provided to this list. When you use this activity to access the UI, the Browse tab is preselected as determined by the `<initial-tab>` tag.

Defining the Browse Tab

This section describes the Browse tab:

```
      <tab id="browse">
        <display-name-resource>
          assetManager.tab.browse
        </display-name-resource>
        <page>
          /browse/browseTab.jsp
        </page>
```

The Browse tab name is specified to the `assetManager.tab.browse` key in the global resource bundle named in the `AssetUI` configuration file `<resource bundle>` tag. The `browseTab.jsp` file provides the tab content.

The views on the Browse tab are organized and defined as follows:

```
<views>
  <view-order>
    <view-id>personalization.users.users</view-id>
    <view-id>personalization.users.orgsAndRoles</view-id>
    <view-id>personalization.users.orgs</view-id>
  </view-order>
  <initial-view>
    personalization.users.users
  </initial-view>
```

The names of views map to names in the UI as follows:

View Name	Default View Name in the UI
`personalization.users.users`	Users
`personalization.users.orgsAndRoles`	Organizations and Roles
`personalization.users.orgs`	Organizations

The views display in a drop-down list in the order specified here, and when you open the UI, the Users view displays in the Browse tab because it is specified as the initial view.

All views are similar in structure. The code for the `personalization.users.users` view is as follows:

```
<view id="personalization.users.users">
  <resource-bundle>
    atg.web.personalization.WebAppResources
  </resource-bundle>
  <display-name-resource>
    users.browseTab.view.users
  </display-name-resource>
  <configuration>
    /atg/web/assetmanager/configuration/profile/UserViewConfiguration
  </configuration>
  <page>
    /browse/list.jsp
  </page>
</view>
```

Each view has a resource bundle key that maps to the name used for the view in the drop-down list. In addition to a resource bundle key, a new resource bundle is specified because labels for views are stored in a resource bundle that is different from the one used for the rest of the application.

Because there are many configuration options used to structure and operate a view, those options are stored in a Nucleus component to simplify and reduce clutter in the task configuration file. You can view the Nucleus component in the ACC using the path provided in the task configuration file. See the Creating Views section for more information about this component. The list structure used by this view is provided from `list.jsp`.

Defining the Search Tab

The Search tab is defined as follows:

```
<tab id="search">
  <display-name-resource>
    assetManager.tab.search
  </display-name-resource>
  <page>
    /search/searchTab.jsp
  </page>
  <views>
    <initial-view>
      form
    </initial-view>
```

The Search tab definition begins with the resource bundle key that is mapped to the name used for the tab. Note that two resource bundles have been specified: one in the `<default-activity>` tag in the `AssetUI` task configuration file and the other in the `<view>` tags included in this activity. The first resource bundle is the default and used here; the second is used only for the tags that specifically include it. The content displayed in the Search tab is contained in `searchTab.jsp`. Note that although the Search tab has views, they aren't displayed in an ordered drop-down list because they aren't available for user selection. The `searchTab.jsp` file is designed to display one view (`form`) for entering search criteria and another (`results`) for displaying search results. The `form` is defined below:

```
<view id="form">
  <page>
    /search/searchForm.jsp
  </page>
  <item-types>
    <item-type>/atg/userprofiling/ProfileAdapterRepository:user</item-type>
    <item-type>/atg/userprofiling/ProfileAdapterRepository:organization
    </item-type>
    <item-type>/atg/userprofiling/ProfileAdapterRepository:organizationalRole
    </item-type>
    <item-type>/atg/userprofiling/ProfileAdapterRepository:role</item-type>
    <item-type>/atg/userprofiling/ProfileAdapterRepository:roleFolder</item-type>
  </item-types>
</view>
```

The form view supplies the fields and buttons for building a query in searchForm.jsp. When you want to perform a search, you start by selecting the type of item you want to locate from a drop-down list. The types available to the drop-down list are provided here by the <item-type> tags. The actual names that are used in the list are the display names defined for the item type in the item descriptor. The item types in the default list include a subset of the types defined for Personalization module.

The Find, Add Criteria, and New Search buttons are not defined in the task configuration file. Because it seems unlikely you would want to remove or replace these buttons, they are hardcoded into the search JSPs.

The second view, results, appears as follows:

```
        <view id="results">
          <page>
            /search/searchResults.jsp
          </page>
          <operations>
            <!-- users/orgs/roles don't need link or unlink -->
            <operation combine="remove">link</operation>
            <operation combine="remove">unlink</operation>
          </operations>
        </view>
```

The results view uses searchResults.jsp to display the items resulting from a query. The results page lets you select and delete items as well as add them to the Project and Multi Edit tabs.

Defining the Multi Edit Tab

The Multi Edit tab is defined as follows:

```
<tab id="multiEdit">
  <display-name-resource>
    assetManager.tab.multiEdit
  </display-name-resource>
  <page>
    /multiEdit/multiEditTab.jsp
  </page>
  <operations>
    <operation combine="remove">addToMultiEdit</operation>
    <operation>removeFromMultiEdit</operation>
    <operation>stepEdit</operation>
    <operation>applyToAll</operation>
    <operation>listEdit</operation>
  </operations>
  <configuration>
    /atg/commerce/web/assetmanager/MultiEditTabConfiguration
  </configuration>
  </tab>
</tabs>
```

Similar to the Browse tab, the Multi Edit tab stores its tab name in a resource file that maps the string provided here to the tab name. The tab structure is defined in `multiEditTab.jsp`. The Add to Multi Edit button is unnecessary in the Multi Edit tab, so it is removed. Buttons specific to Multi Edit are added to the tab. The `MultiEditTabConfiguration` component defines additional settings that are described in the Browse Tab Configuration information in Creating Views section.

personalization.browseSegmentsAndTargeters Activity Settings

This activity lets you access user segments, content groups, and targeters in view-only mode. It corresponds with the Browse link next to Targeting and Segmentation in the Operations list. The definition starts as follows:

```
<activity id="personalization.browseSegmentsAndTargeters" inherit-
 from="assetManager.defaultBrowse">
  <activity-name>
    personalization.browseSegmentsAndTargeters
  </activity-name>
```

This activity inherits settings from the `assetManager.defaultBrowse` activity.

When you access this activity in the ATG Business Control Center, you see the Browse tab only:

```
<tabs>
  <tab-order>
    <tab-id>browse</tab-id>
  </tab-order>
  <initial-tab>
    browse
  </initial-tab>

  <tab id="browse">
    <display-name-resource>
      assetManager.tab.browse
    </display-name-resource>
    <page>
      /browse/browseTab.jsp
    </page>
```

This activity specifies one tab and does not inherit others; however, it is still necessary to specify tab order and the initial tab.

The resource bundle key and JSP provided for this activity are the same as those identified in the prior activity defined in this task configuration file, `personalization.users`.

The views for the Browse tab are included as follows:

```
<views>
  <view-order>
    <view-id>personalization.segmentsAndTargeters.segments</view-id>
    <view-id>personalization.segmentsAndTargeters.contentGroups</view-id>
    <view-id>personalization.segmentsAndTargeters.targeters</view-id>
  </view-order>
  <initial-view>
    personalization.segmentsAndTargeters.segments
  </initial-view>
```

The three views are visible in the Browse tab and accessible by the Show drop-down list in the order specified above. The mapping of view names in this file to views in the drop-down list is as follows:

View Name	Default View Name in the UI
`personalization.segmentsAndTargeters.segments`	User Segments
`personalization.segmentsAndTargeters.contentGroups`	Content Groups
`personalization.segmentsAndTargeters.targeters`	Targeters

The User Segments view displays first when you view this activity. The same types of information are provided for all three views. The example below shows `personalization.segmentsAndTargeters.segments`:

```
<view id="personalization.segmentsAndTargeters.segments">
  <resource-bundle>
    atg.web.personalization.WebAppResources
  </resource-bundle>
  <display-name-resource>
    segmentsAndTargeters.browseTab.view.segments
  </display-name-resource>
  <configuration>
    /atg/web/assetmanager/configuration/targeting/
    SegmentViewConfiguration
  </configuration>
  <page>
    /browse/tree.jsp
  </page>
</view>
```

This code describes the personalization.segmentsAndTargeters.segments view, which corresponds to the view you see when you select User Segments from the Show menu in the Browse tab. The User Segments label is defined in the resource bundle atg.web.personalization.WebAppResources. The key for this view is segmentsAndTargeters.browseTab.view.segments. Configuration details that determine the resources used to construct the view are provided in the SegmentViewConfiguration component, although the tree structure is provided by tree.jsp.

Note that none of the views have buttons because this is a view-only activity, so tools for moving, creating, and deleting are not necessary.

personalization.editSegmentsAndTargeters Task Settings

The personalization.editSegmentsAndTargeters task is a generalized resource that doesn't map to a particular task or workflow in the ATG Business Control Center. However, this task provides settings that are inherited by the configuration for the editSegmentsAndTargeters workflow.

The task definition starts as follows:

```
<task id="personalization.editSegmentsAndTargeters" inherit-
 from="personalization.browseSegmentsAndTargeters">
```

This tag specifies that the resource being configured is a task called personalization.editSegmentsAndTargeters. Because this task inherits settings from the personalization.browseSegmetnsAndTargeters activity that determine how UI displays user segments, content groups, and targeters in view-only mode, the task tag encloses only those setting required to overwrite previous settings or to supplement them with tools for editing those items.

The general settings are as follows:

```
<asset-editor>
  <page>
    /assetEditor/editAsset.jsp
  </page>
  <configuration>
    /atg/web/assetmanager/configuration/targeting/TargetingEditorConfiguration
  </configuration>
</asset-editor>
```

The Navigation pane obtains its structure from editAsset.jsp. The configuration resources are specified in the TargetingEditorConfiguration component.

The necessary buttons are added to the tab defined in this task (Project) as well as the tab inherited from other resources (Browse), unless overwritten elsewhere in this file:

```
<operations>
  <operation>create</operation>
  <operation>move</operation>
  <operation>duplicate</operation>
  <operation>delete</operation>
  <operation>addToProject</operation>
</operations>
```

These buttons are the same as those added to the `personalization.user` activity, but since this task doesn't inherit settings from that activity, the buttons need to be specified here. Remember that these are a subset of the buttons available to the UI. For a complete list, see the *ATG Merchandising UI Basics* section of the *Using the ATG Merchandising User Interface* chapter in the *ATG Merchandising User Guide*.

The Browse tab settings are inherited by this task from the `personalization.browseSegmentsAndTargeters` activity. The Project tab is added as follows:

```
<tabs>
  <tab-order>
    <tab-id>project</tab-id>
  </tab-order>
```

When a resource is evaluated, inherited tags are ordered before directly specified ones. Hence, the Browse tab configuration is inherited by this resource, so in the Show drop-down list, the order is Browse tab, then Project tab. The initial tab is the Browse tab, which is another setting inherited by the task. This file could have provided a different tab order or initial tab by overwriting the inherited settings.

The Project tab is defined as follows:

```
    <tab id="project">
     <display-name-resource>
       assetManager.tab.project
     </display-name-resource>
     <page>
       /project/projectTab.jsp
     </page>
     <operations>
       <operation combine="remove">addToProject</operation>
       <operation>removeFromProject</operation>
     </operations>
  </tab>
</tabs>
```

The Project tab, named as such by the `assetManager.tab.project` key in the resource bundle provided in the `<default-activity>` tags, relies on `projectTab.jsp` for its Navigation pane structure. The buttons defined in the beginning of this task are visible in the Project tab, although two changes are made to that list. There's no reason to have an Add to Project button on the Project tab because all items visible in the tab have already been added to it, so that button is hidden from view. A Remove from Project button is provided in its place, permitting users to undo changes made to items in the current project.

The view mapping settings are modified as follows:

```
  <view-mappings>
    <view-mapping mode="AssetManager.edit" combine="replace">
      <item-mapping>
        <item-type>*</item-type>
        <item-mapping-name>AssetManager</item-mapping-name>
      </item-mapping>
    </view-mapping>
  </view-mappings>
</task>
```

From the `personalization.browseSegmentsAndTargeters` activity, the current task inherits view-only access to all parts of the UI, meaning that regardless what map mode is specified by the JSP, `AssetManager.view` is returned to it. For `AssetManager.edit` map mode, this setting overrides the previous one. When a page requests this mode, it is used verbatim.

personalization/editSegmentsAndTargeters.wdl Workflow Settings

The `personalization/editSegmentsAndTargeters` workflow is the workflow you access when you click the Targeting and Segmentation link in the Operations list. You use this workflow to create projects that manage interactions with user segments, content groups, and targeters.

The following tags are defined for this workflow:

```
<task id="/Personalization/editSegmentsAndTargeters.wdl"
 inherit-from="personalization.editSegmentsAndTargeters">
    <workflow-name>
      /Personalization/editSegmentsAndTargeters.wdl
    </workflow-name>
  </task>
```

This workflow inherits all settings from the `personalization.editSegmentsAndTargeters` task. The two resources are separate, so other tasks and workflows can also inherit settings from `personalization.editSegmentsAndTargeters` without being affected by any changes made to this workflow.

Creating and Defining a Task Configuration File

You may want to modify the settings provided for existing activities, tasks, and workflows or define settings for custom resources you create for your application. You can adapt the UI by creating a custom task configuration file. To do so, follow these steps:

1. Create a file called `taskConfiguration.xml`.
2. Save the file to your local configuration directory, using the same path as that used by the out-of-the-box configuration file. All task configuration files use the same path within their modules. Example:

   ```
   <ATG10dir>/home/localconfig/atg/web/assetmanager/configuration
   ```

3. Open the file and insert the following text:

   ```
   <?xml version="1.0" encoding="ISO-8859-1" ?>

   <task-configuration>
   </task-configuration>
   ```

4. Inside the `<task-configuration>` tag, insert the tags used to identify the settings you are defining:
   - For the default activity, use the `<default-activity>` tag.
   - For the default task, use the `<default-task>` tag.
   - For a specific activity, use the `<activity>` tag.
   - For a specific task, use a `<task>` tag with a nested `<task-name>` tag.
   - For a specific workflow, use a `<task>` tag with a nested `<workflow-name>` tag.
   - For a specific task in a specific workflow, use a `<task>` tag with nested `<task-name>` and `<workflow-name>` tags.

     For information on tags and tag attributes, see Appendix: Tags in a Task Configuration File.
5. Insert tags for modifying specified activity, task, or workflow configurations. You can make the following modifications:
   - Add support for new views, item types, or properties.
   - Remove existing tabs, buttons, views, item types, and properties.
   - Modify existing settings, such as the item types supported in the Search tab.

Appendix: Tags in a Task Configuration File

The tags described here are used to configure how the Asset Manager UIs of the ATG Business Control Center appear to users based on the activity, workflow, or task they use to access it. This table includes tag names, descriptions, attributes, and nested tags. Note that the rules described here apply to each individual taskConfiguration.xml file as well as the composite file that ATG assembles the individual files into. For example, every tab must have an initial view when the composite file is generated, but an initial view need not be specified in each individual task configuration file.

These tags are defined in the taskConfiguration_1.0.dtd located in the <ATG10dir>/AssetUI/lib/classes.jar/atg/dtds/web/assetmanager directory. Refer to the following table to learn about the tags available to you:

Tag	Description	Additional Information
<activity>	Encloses the set of tags that define how a customized task behaves.	Attributes: id is a required attribute that identifies the name of the activity for which you are providing settings. The name you provide has no significance outside of this file. inherit-from is an optional attribute used to identify an activity from which the activity set to the id attribute inherits settings. Nested Tags: <activity-name> is required <resource-bundle> <page> <asset-editor> <tabs> <view-mappings> <operations> <configuration>

<activity-name>	Encloses the name of an activity as it is defined in a generic activities file. Use this tag if you define an activity that's visible in the ATG Business Control Center.	
<asset-editor>	Encloses tags that describe the Details pane.	Nested Tags: <resource-bundle> <page> <configuration> You are required to use at least one of these tags when you use <asset-editor>.
<configuration>	Encloses the component path and name that provides configuration settings. When nested in <activity> or <task> tags, it identifies a Configuration Manager. When nested in <view> tags, it identifies a component of class atg.web.assetmanager.configuration.FilterableBrowseListViewConfiguration (for list views) or atg.web.assetmanager.configuration.BrowseTreeViewConfiguration (for tree views).	
<configuration>	Encloses the component path and name that configures the Navigation pane to support a given view. This configuration tag is nested in <view> tags.	
<default-activity>	Encloses the set of tags that define settings applied to all activities and tasks by default.	Nested Tags: <resource-bundle> <page> <asset-editor> <tabs> <view-mappings> <operations> You are required to use at least one of these tags when you use <default-activity>.

`<default-task>`	Encloses the set of tags that define the settings applied to all tasks and workflows.	Nested Tags:
		`<resource-bundle>`
		`<page>`
		`<asset-editor>`
		`<tabs>`
		`<view-mappings>`
		`<operations>`
		You are required to use at least one of these tags when you use `<default-task>`.

`<display-name>`	Encloses the name used to identify a tab or view. Use this tag rather than `<display-name-resource>` when you want the same name to display to all users regardless of locale.	
`<display-name-resource>`	Encloses the key that is mapped to the tab's or view's name in a resource bundle. Use this tag, rather than `<display-name>` when you want to provide localized names.	
`<initial-tab>`	Encloses the tab identifier for the tab that is visible when the application opens.	
`<initial-view>`	Encloses the view identifier for the view that is visible when the tab that contains it is accessed for the first time.	
`<item-mapping>`	Encloses the set of tags that determine, when a view mapping mode is requested for a given item type, the item mapping name and view mapping mode that are returned.	Nested Tags: `<item-type>` `<item-mapping-name>` or `<view-mapping-mode-override>` is required
`<item-mapping-name>`	Encloses the name that identifies an item mapping. An item mapping represents a set of screen elements.	
`<item-type>`	Encloses an item type in this format: *`path/repository_definition:item_type_name`*	Attribute: `combine` is an optional attribute used to override inherited item type settings. Set `combine` to `remove` or `append`.

`<item-types>`	Encloses the types of items that the item mapping settings apply to.	Attribute: `combine` is an optional attribute used to override inherited item type settings. Set `combine` to `replace`. Nested tag: `<item-type>` is required
`<operation>`	Encloses the identifier for a button.	Attribute: `combine` is an optional attribute used to override inherited operation settings. Set `combine` to `remove` or `append`.
`<operations>`	Encloses `<operation>` tags that identify button available in a given context.	Attribute: `combine` is an optional attribute used to override inherited operation settings. Set `combine` to `replace`. Nested tag: `<operation>` is required
`<page>`	Encloses a JSP that contains content for a screen element.	
`<resource-bundle>`	Encloses the name of the resource bundle that maps a UI element to its label. The value of this tag must include the path to the resource bundle where each directory is separated by a period instead of a slash.	

`<tab>`	Identifies a tab and contains nested tags that define how it displays.	Attributes:
		`id` is a required attribute that identifies the tab.
		`combine` is an optional attribute used to override inherited tab settings. Set `combine` to `replace` or `remove`.
		Nested Tags:
		`<resource-bundle>`
		`<display-name-resource>`
		`<page>`
		`<configuration>`
		`<views>`
		`<operations>`
		You are required to use at least one of these tags when you use `<tab>`.

<tab-id>	Encloses the identifier for a tab. This tag is used alongside other <tab-id> tags, all of which are nested in a <tab-order> tag in order to specify the order of tabs in the UI.	Attributes: combine is an optional attribute used to add values to or override inherited tab settings. Set combine to remove or append.
<tab-order>	Encloses tags that list tabs in the order used to display them in the UI.	Attributes: combine is an optional attribute used to override an inherited ordering. Set combine to replace. Nested tag: <tab-id> is required
<tabs>	Encloses a series of nested tags that define how tabs display.	Nested tags: <tab> <tab-order> or <initial-tab> is required
<task>	Encloses the set of tags that defines how a task appears.	Attributes: id is an optional attribute used to identify a specific task. Use this attribute when you are defining a generic task or a task that will inherit settings from another resource. inherit-from is an optional attribute used to identify a resource from which this task inherits settings. Nested tags: <workflow-name> or <task-name> is required <resource-bundle> <page> <asset-editor> <tabs> <view-mappings> <operations> <configuration> You are required to use at least one of these tags when you use <task>.
<task-name>	Encloses the name of a task as it is defined in a workflow .wdl file.	
<task-configuration>	Encloses all tags in the task configuration file.	Nested tags: <default-activity> or <default-task> is required <activity> <task>

`<view>`	Identifies a view and contains nested tags that define how it displays.	Attributes:
		`id` is a required attribute that identifies the view.
		`combine` is an optional attribute used to override inherited view settings. Set `combine` to `replace` or `remove`.
		Nested Tags:
		`<resource-bundle>`
		`<display-name-resource>`
		`<display-name>`
		`<page>`
		`<configuration>`
		`<item-types>`
		`<operations>`
		You are required to use at least one of these tags when you use `<view>`.

`<view-id>`	Encloses the identifier for a view. This tag is used alongside other `<view-id>` tags all of which are nested in a `<view-order>` tag in order to specify the order of views in the UI.	Attributes: `combine` is an optional attribute used to append or override inherited ordering settings. Set `combine` to `remove` or `append`.
`<view-mapping>`	Identifies the input view mapping mode used to determine, in combination with the item type, the output view mapping mode and item mapping name.	Attributes: `mode` is a required attribute that identifies the input map mode. `combine` is an optional attribute used to override inherited settings for the input mode. Set `combine` to `replace` or `remove`. Nested Tag: `<item-mapping>` is required
`<view-mappings>`	Encloses a series of nested tags that specify view mapping settings.	Nested Tag: `<view-mapping>` is required
`<view-mapping-mode-override>`	Encloses the name of the output map mode to use.	
`<view-order>`	Encloses `<view-id>` tags that list views in the order used to display them in the UI.	Attributes: `combine` is an optional attribute used to override inherited ordering settings. Set `combine` to `replace`. Nested Tag: `<view-id>` is required

`<views>`	Encloses tags that define how views display.	Nested Tags: `<view>` `<view-order>` or `<initial-view>` is required
`<workflow-name>`	Encloses the name of a workflow used by the `.wdl` file. Include the path of the `.wdl` file relative to the workflow registry.	

Index