Chapter

1

Introducing Actuate Information Console

This chapter contains the following topics:

- About Actuate Information Console
- About Actuate Information Console architecture

About Actuate Information Console

Actuate Information Console is a web application that supports accessing and working with report information using a web browser. Web developers and designers use Actuate Information Console's industry-standard technology to design custom e.reporting web applications that meet business information delivery requirements.

Actuate Information Console technology is platform-independent and customizable. By separating user interface design from content generation, Information Console ensures that reporting web application development tasks can proceed simultaneously and independently. You deploy Actuate Information Console on a network with Actuate BIRT iServer. Information Console accesses and stores documents on an Encyclopedia volume managed by iServer. Actuate Information Console technology is also scalable and supports clustering. On a Windows system, the default context root for Information Console is C:\Program Files\Actuate11\iPortal\iportal for Information Console installed separately or C:\Program Files\Actuate11SP4\iServer\servletcontainer\iportal for Information Console embedded in the BIRT iServer application. On a UNIX-based system, the default context root for Information Console is $Home/iPortal/iportal for Information Console installed separately or $Home/iServer/servletcontainer/iportal for Information Console embedded in the BIRT iServer application.

Actuate Information Console technology includes the following features:

- JavaServer Pages (JSPs) support creating HTML or XML pages that combine static web page templates with dynamic content.
- Distributing requests to multiple Actuate BIRT iServer machines in an Actuate BIRT iServer System System cluster balances server loads.
- Simple Object Access Protocol (SOAP) standards provide plain text transmission of XML using HTTP.
- Actuate Information Delivery API supports direct communication between the pages' custom tags and Actuate BIRT iServer.
- The full range of authentication and authorization functionality that Actuate BIRT iServer provides is available.
- Secure HTTP (HTTPS) supports secure information transfer on the web.
- JSR 168 compliant portlets provide access to reports through portal servers that support the JSR 168 standard.
- Licensed options on BIRT iServer provide additional functionality. To use these options on a BIRT iServer System, the BIRT iServer System must be licensed for the options. For example, to use browser-based tools, such as BIRT

Interactive Viewer or BIRT Data Analyzer, the BIRT iServer requires the appropriate license options.

The BIRT 360 Option for BIRT iServer is required to use dashboard and gadget files. If these options are not available, users cannot open dashboards or gadgets in Information Console.

Setting up Actuate Information Console

You install Information Console in either of two ways:

- As a separate web application. This method enables native load-balancing for iServer clusters, redundancy to support constant report services over the web, and secure networks using firewalls and proxy severs as described in Chapter 10, "Using Actuate Information Console security."
- Automatically on the same host with iServer. This method provides reports locally on each iServer machine.

For enterprise architectures, installing Information Console on several web servers is recommended.

To deploy a report to the web, you need:

- An Actuate Information Console installation.
- An application server or JSP or servlet engine such as Actuate embedded servlet engine or IBM WebSphere.
- One or more Actuate designer tools and Actuate BIRT iServer System with Actuate Management Console.
- Actuate BIRT iServer administrator privileges.
- Permission to read, write, and modify operating system directories as necessary. For example, the directory Java uses to hold temporary files is defined by the java.io.tmpdir property and is by default the value of the TMP system variable in the Windows environment and /var/tmp in the UNIX and Linux environments. Read and write permission must be provided to the application server running Information Console for this directory.

This section discusses deployment concerns that may affect your Information Console installation and how you wish to deploy reports to the web. For more information about installing Information Console, see *Installing BIRT iServer for Windows* or *Installing BIRT iServer for Linux and UNIX*.

Generating a web archive (WAR) for installation

To deploy Information Console on an application server, you can use a WAR file of your Information Console application. Generating Web Archive is a feature of Actuate Information Console that is available to Administrator-level users. This feature creates a WAR file of your entire Actuate Information Console system.

Information Console streams the WAR file to your browser. You select a file name and location to save the file. After you customize your system, you can create a WAR file to deploy the customized Information Console on other machines. The customizations can include any modifications of JavaScript, JavaServer Pages (JSP) and other web pages, and skins. Later chapters in this book provide detailed information about customizing JavaScript and JSPs.

If Actuate Information Console is deployed as a WAR file, you cannot further customize skins, add pages, or make any other changes that affect the Actuate Information Console file structure in the WAR file. Instead, install Actuate Information Console as a directory structure with the installation wizard on your product CD and make your changes to that installation. Then use Generate Web Archive to create a new WAR file and deploy that WAR file to your application server.

How to customize and deploy Actuate Information Console in a cluster

To customize Actuate Information Console and deploy it to application servers in a clustered environment, use the following general procedure.

1. Install Actuate Information Console on one of the machines in your cluster.
2. Customize the Actuate Information Console JavaScript, skins, and web pages as desired.
3. Open Information Console. On the landing page, choose My Documents.
4. Log in as an Administrator-level user. On the Information Console banner, choose Customization.
5. Choose Generate Web Archive. At the prompt, provide a location for the WAR file. For example, provide the location where your application server accesses WAR files. By default, the name of the WAR file of your customized Actuate Information Console installation is acweb.war.
6. Deploy the WAR file to each remaining machine in your cluster.

Understanding Actuate Information Console load balancing

Actuate Information Console supports two kinds of load balancing, as illustrated in Figure 1-1, to ensure high availability and to distribute tasks for efficient processing:

- Actuate Message Distribution Service (MDS) balances the request load among Actuate BIRT iServer machines in an Actuate BIRT iServer cluster.

  The Message Distribution service eliminates the need for a third-party network load balancer in front of the Actuate BIRT iServer tier. Actuate Information Console determines which machines in a cluster have MDS running and detects when the MDS machines go offline. MDS distributes the

load among the available servers and does not attempt to send a request to an offline machine.

- Clustered Actuate Information Console machines can use a third-party application to balance the load among the application servers.

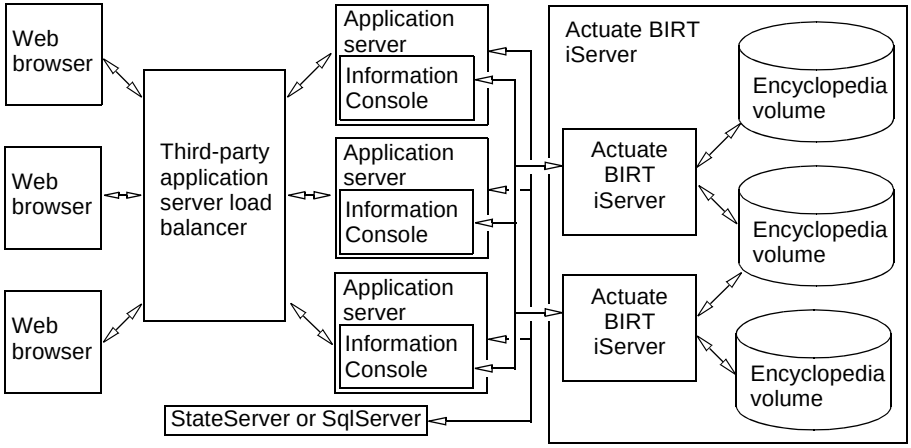

Figure 1-1 Load-balancing architecture for Information Console

Deploying a load-balancer for an Actuate BIRT iServer cluster

To deploy a load-balancer or proxy layer in front of the Actuate BIRT iServer tier, disable the Actuate load-balancing support by setting the MDS_ENABLED configuration parameter to False in the web.xml Actuate Information Console configuration file.

About using a cluster of application servers

If the application servers running Information Console support session state management, you can configure Actuate Information Console and the application servers to share and maintain a web browsing session state across a cluster of Information Console instances. Configuring the application servers to track the state of each Information Console instance supports reusing authentication information. In other words, you can log in to an Information Console instance and send a request using another Information Console instance without logging in again using the second instance.

If you do not use an application server to track session state information, managing the session state is fast, but you lose a user's state information when you restart Actuate Information Console or your application server.

Sharing session state information takes advantage of the application servers' failover features. If a user is on a cluster application server running Information

Console and that application server fails, another application server running Information Console can manage the user's session.

An application server works with one or more database servers to manage session state information. All application servers must have access to the database server to store and retrieve session state information. For specific information about configuring your installation, see your application server documentation.

About Actuate Information Console architecture

This section describes the general operation, authentication, and structure of Information Console as a web application.

The Actuate Information Console architecture is illustrated in Figure 1-2.

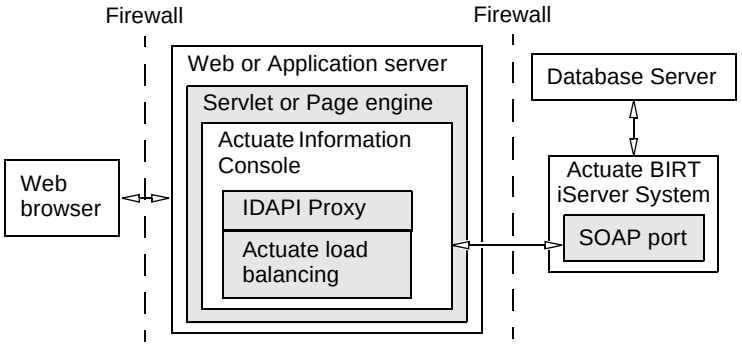

Figure 1-2 Actuate Information Console architecture overview

A user submits a request by choosing a link on a web page that specifies an Actuate Information Console URI. As shown in Figure 1-2, the web or application server receives the URI as an HTTP request and passes the request to the servlet or page engine. The engine invokes Actuate Information Console, interprets the URI, and communicates with the Actuate BIRT iServer using the Actuate Information Delivery API (IDAPI). The IDAPI manages the request and returns the results to Actuate Information Console and the servlet or page engine. The web server returns the results to the web browser. Then, the web browser displays the results for the user.

Actuate Information Console manages requests as part of a JSP engine within a web or application server. There is no default user interface for the engine. On a Windows system, Actuate Information Console installation places an Actuate Information Console link on the Start menu.

Using proxy servers with Actuate Information Console

When setting up a proxy server with Actuate Information Console, there are steps you must take if your internal application server port is protected by a firewall. In this situation, when the proxy server changes the URL to point to the new context's port, that port is unavailable due to the firewall. The usual solution is to configure a reverse proxy, but if you are using multiple proxies and a reverse proxy is not practical for your installation, Actuate Information Console can perform the redirection.

To redirect a page without using a reverse proxy, Actuate Information Console forwards the URL to redirect to the processRedirect.jsp page and updates the browser's location bar accordingly. This action processes on the client. The browser takes the current URL location and updates the rest of the URI using the redirected URL. You must also set the ENABLE_CLIENT_SIDE_REDIRECT configuration parameter to True and modify the redirect attributes in the <context root>/WEB-INF/struts-config.xml file. The necessary modifications are included in the file. You just need to comment out the lines that have the redirect attribute set to True and uncomment the lines that forward to the processRedirect.jsp page.

For example, the following code is the struts-config.xml entry for the login action:

```
<!-- Process a user login -->
<action path="/login" name="loginForm"
   scope="request" input="/iportal/activePortal/private/login.jsp"
   type="com.actuate.activeportal.actions.AcLoginAction"
   validate="false">
<forward name="loginform"
   path="/iportal/activePortal/private/login.jsp" />
<!--
<forward name="success" path="/iportal/activePortal/private/common
   /processredirect.jsp?redirectPath=/getfolderitems.do" />
-->
<forward name="success" path="/getfolderitems.do"
   redirect="true" />
<forward name="dashboard" path="/dashboard" redirect="true" />
<forward name="ajcLogin" path="/ajclanding.jsp" redirect="true" />
<forward name="landing" path="/landing.jsp" redirect="false" />
</action>
```

By default the forward statement for success points to getfolderitems.do with the redirect attribute set to True. This code instructs the application server to send a redirect with the getfolderitems.do URL when the user logs in.

From behind a firewall and proxy, this redirect method fails because the redirect sent by the application server points to the application server port instead of the firewall and proxy port. For success, comment out the line having redirect="true".

Uncomment the line that points to processRedirect.jsp. The following code shows the updated entry in struts-config.xml:

```
<!-- Process a user login -->
<action path="/login" name="loginForm"
   scope="request" input="/iportal/activePortal/private/login.jsp"
   type="com.actuate.activeportal.actions.AcLoginAction"
   validate="false">
<forward name="loginform"
   path="/iportal/activePortal/private/login.jsp" />
<forward name="success" path="/iportal/activePortal/private/common
   /processredirect.jsp?redirectPath=/getfolderitems.do" />
<!--
<forward name="success" path="/getfolderitems.do"
   redirect="true" />
-->
<forward name="dashboard" path="/dashboard" redirect="true" />
<forward name="ajcLogin" path="/ajclanding.jsp" redirect="true" />
<forward name="landing" path="/landing.jsp" redirect="false" />
</action>
```

This change needs to be made for all the actions in struts-config.xml that send a redirect to the browser.

About Actuate Information Console pages

Actuate Information Console uses JSPs to generate web pages dynamically before sending them to a web browser. These JSPs use custom tags, custom classes, and JavaScript to generate dynamic web page content. The JavaScript, classes, and tags provide access to other pages, JavaBeans, and Java classes. For example, application logic in Actuate Information Console can reside on the web server in a JavaBean.

Web browsers can request a JSP with parameters as a web resource. The first time a web browser requests a page, the page is compiled into a servlet. Servlets are Java programs that run as part of a network service such as a web server. Once a page is compiled, the web server can fulfill subsequent requests quickly, provided that the page source is unchanged since the last request.

The dashboards servlet and JSPs support the dashboards and gadgets interface for Information Console. The dashboard pages reside in <context root> \dashboard\jsp. To provide dashboard access, enable the BIRT 360 license option.

The channels JSPs and custom tags support viewing reports submitted to channels. The channels pages reside in <context root>\iportal\activePortal \private\channels. Users access channels by clicking Channel in the sidebar.

The filesfolders JSPs and custom tags support accessing repository files and folders. These JSPs and custom tags reside in <context root>\iportal \activePortal\private\filesfolders.

The submit request JSPs and custom tags support submitting new jobs. The submit request JSPs reside in <context root>\iportal\activePortal\private \newrequest. For specific information about running jobs using Actuate Information Console, see *Using Information Console.*

The options JSPs and custom tags support managing user option settings. The options pages reside in <context root>\iportal\activePortal\private\options.

The viewing JSPs and custom tags support the following functionality, depending on the report type:

- Searching report data
- Using a table of contents to navigate through a report
- Paginating or not paginating a report
- Fetching reports in supported formats

For specific information about viewing reports using Actuate Information Console, see *Using Information Console.*

Use the default pages, customize the pages, or create entirely new pages to deploy your reporting web application.

Working with Actuate Information Console URIs

Actuate Information Console Uniform Resource Identifiers (URIs) convey user requests to the Actuate BIRT iServer System. URIs access functionality including generating and storing reports, managing volume contents, and viewing reports.

About Actuate Information Console URIs

Actuate Information Console URIs consist of the context root and port of the web server where you install and deploy the JSPs or servlets. Actuate Information Console URIs have the following syntax:

```
http://<web server>:<port>/iportal/<path><page>.<type>
   [?<parameter=value>{&<parameter=value>}]
```

- <web server> is the name of the machine running the application server or servlet engine. You can use localhost as a trusted application's machine name if your local machine is running the server.
- <port> is the port on which you access the application server or page or servlet engine. The default port for Information Console installed separately is 8700, while the BIRT iServer embedded version uses 8900 by default.

- iportal is the default context root for accessing the Actuate Information Console pages.
- <path> is the directory containing the page to invoke.
- <page> is the name of the page or method.
- <type> is jsp or do.
- <parameter=value> specifies the parameters and values that the page requires.

For example, to view the login page, Actuate Information Console uses a URI with the following format:

```
http://<web server>:<port>/iportal
   /login.jsp?TargetPage=<folder/file>
```

- iportal/login.jsp is the JSP that provides default login functionality for Information Console.
- TargetPage is the viewframeset.jsp parameter that specifies the page to direct the user to after the login completes.
- <folder/file> is the complete pathname for the file that the client opens after the login completes.

Using a special character in a URI

Actuate Information Console URIs use encoding for characters that a browser can misinterpret. The following example uses hexadecimal encoding in the Information Console URI to display the report, Msbargph.roi, from an Encyclopedia volume:

```
http://infoconsole:8900/iportal/activePortal/viewer
   /viewframeset.jsp?name=%2fmsbargph%2eroi%3b1&__vp=server1
```

You do not have to use hexadecimal encoding in all circumstances. Use the encoding only when the possibility of misinterpreting a character exists. The following unencoded URI displays the same report as the preceding URI:

```
http://infoconsole:8900/iportal/activePortal/viewer
   /viewframeset.jsp?name=\msbargph.roi;1&__vp=server1
```

Always encode characters that have a specific meaning in a URI when you use them in other ways. Table 1-1 describes the available character substitutions. An ampersand introduces a parameter in a URI, so you must encode an ampersand that appears in a value string. For example, use:

```
&company=AT%26T
```

instead of:

```
&company=AT&T
```

Table 1-1 Encoding sequences for use in URIs

Character	Encoded substitution
ampersand (&)	%26
asterisk (*)	%2a
at (@)	%40
backslash (\)	%5c
colon (:)	%3a
comma (,)	%2c
dollar sign ($)	%24
double quote (")	%22
equal (=)	%3d
exclamation (!)	%21
forward slash (/)	%2f
greater than (>)	%3e
less than (<)	%3c
number sign (#)	%23
percent (%)	%25
period (.)	%2e
plus (+)	%2b
question mark (?)	%3f
semicolon (;)	%3b
space ( )	%20
underscore (_)	%5f

If you customize Actuate Information Console by writing code that creates URI parameters, encode the entire parameter value string with the encode() method. The encode() method is included in encoder.js, which is provided in the Actuate Information Console <context root>/js directory. The following example encodes the folder name /Training/Sub Folder before executing the getFolderItems action:

```
<%-- Import the StaticFuncs class. --%>
<%@ page import="com.actuate.reportcast.utils.*" %>

<%
   String url =
   "http://localhost:8900/iportal/getfolderitems.do?folder=" +
   StaticFuncs.encode("/Training/Sub Folder");
   response.sendRedirect(url);
%>
```

The encode() method converts the folder parameter value from:

```
/Training/Sub Folder
```

to:

```
%2fTraining%2fSub%20Folder
```

About UTF-8 encoding

All communication between Information Console and BIRT iServer uses UTF-8 encoding. UTF-8 encoding is the default encoding that web browsers support. For 8-bit (single-byte) characters, UTF-8 content appears the same as ANSI content. However, if extended characters are used (typically for languages that require large character sets), UTF-8 encodes these characters with two or more bytes.

UTF-8 encoding support is encoded for all Information Console web pages. When customizing these pages or adding customized web pages to an Information Console web application, provide UTF-8 encoding support using the following code:

```
<META
   HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">
```

About Actuate Information Console functionality levels

Actuate Information Console provides functionality levels that control which features are available to a user. Each user has an allocated functionality level. Functionality levels are mapped to security roles, much like privileges. Four functionality levels are supplied with Actuate Information Console. Table 1-2 shows the supplied functionality levels and their corresponding security roles.

Table 1-2 Functionality levels mapping to security roles

Functionality level	Security role
Basic	All—default access
Intermediate	Active Portal Intermediate
Advanced	Active Portal Advanced

Table 1-2 Functionality levels mapping to security roles (continued)

Functionality level	Security role
Administrator	Active Portal Administrator

The Actuate Information Console administrator can modify these levels and add additional levels by editing the configuration file. The standard location for the Actuate Information Console configuration file is <context root>\WEB-INF \functionality-level.config. The functionality level description includes the name of its corresponding security role in the Encyclopedia volume.

Make sure that any roles specified in the configuration file also exist in the Encyclopedia volume. Because all users automatically belong to the All security role, all users will receive the functionality associated with the Basic functionality level plus the functionality associated with any other roles they have.

Understanding functionality levels

By default, the provided functionality levels give the following access. Users with the Basic level can perform the following tasks:

- Access Documents, My Jobs, and Channels
- Delete their own files

Basic level users cannot perform any other modifications. The default banner for the Basic level looks like the one in Figure 1-3.

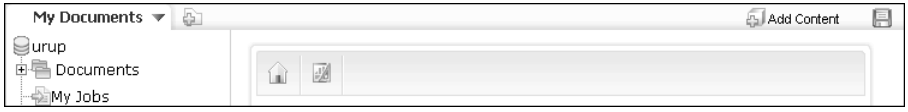

Figure 1-3 Banner menu for a basic level Actuate Information Console user

Users at the Intermediate level have all the Basic level access, and can also perform the following tasks:

- Search documents.
- Create their own job notifications with attachments.
- Subscribe to channels.
- Upload and download files.
- Use the interactive viewer, if this option is licensed.

Users at the Advanced level have all the Intermediate level access, plus they can perform the following tasks:

- Create and delete folders.
- Share files and folders.

- Set job priority.

The default banner for the Intermediate and Advanced levels adds a Search link and looks like the banner in Figure 1-4.

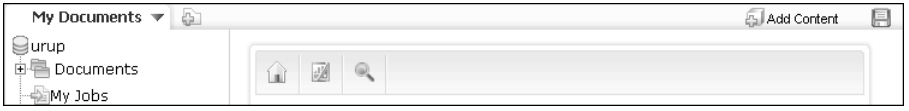

Figure 1-4 Banner menu for advanced level Actuate Information Console user

Users at the Administrator level can perform all Advanced level tasks and can also clone and customize Actuate Information Console skins. The default banner for the Administrator level adds a Customization link, activates the add content function, and looks like the banner in Figure 1-5.

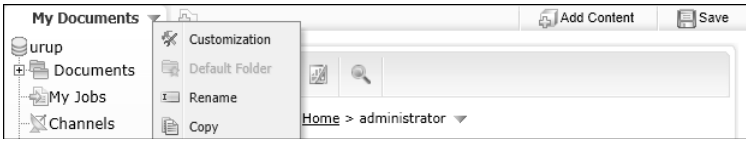

Figure 1-5 Banner menu for an administrator Actuate Information Console user

Use Actuate Management Console to associate the levels with users in the Encyclopedia volume by assigning the appropriate roles to each user.

Customizing functionality levels

Customize or add functionality levels by modifying or creating a level definition in functionality-level.config. A functionality level definition consists of five parts:

- Level name
  The level name must be a unique alphanumeric string, enclosed within <Name> and </Name> tags.
- Matching security role
  The name of the security role that corresponds to the functionality level. Both the security level and the functionality level must exist before the functionality level can be assigned to a user. Enclose the role name with <Role> and </Role> tags.
- Available features
  Table 1-3 describes the five available features.

Table 1-3 Features for functionality levels

Feature	Description
Channels	Provides access to channels
Customization	Provides access to skin customization

Table 1-3 Features for functionality levels (continued)

Feature	Description
Documents	Provides access to files and folders
Jobs	Allows submitting and accessing jobs
Mobile	Provides access to BIRT mobile viewing
Search	Provides access to the file search facility

Features are specified one per line and are enclosed within <FeatureID> and </FeatureID> tags. When a feature is omitted from a functionality level, the corresponding side menu or banner item is hidden to anyone assigned that functionality level. For example, the Search feature is not provided in the Basic functionality level, so the Search link does not appear for users with the Basic functionality level.

- Available subfeatures

  Subfeatures correspond to actions that you can perform through Actuate Information Console. Most subfeatures are associated with a feature. A subfeature cannot be included in a functionality level if its corresponding feature is not included. The subfeatures are described in Table 1-4.

Table 1-4 Subfeatures for functionality levels

Subfeature	Feature	Description
AddFile	Documents	Permits adding files when the user has the appropriate privileges
AdvancedData	NA	Permits the modifying and synchronizing of data sets in BIRT Studio
CreateFolder	Documents	Permits creating folders when the user has the appropriate privileges
Dashboard BusinessUser	NA	Permits use of dashboards
Dashboard Developer	NA	Permits design and administration of dashboards
DeleteFile	Documents	Permits deleting files when the user has the appropriate privileges
DeleteFolder	Documents	Permits deleting folders when the user has the appropriate privileges
DownloadFile	Documents	Permits downloading files when the user has the appropriate privileges
InteractiveViewing	NA	Permits opening Interactive Viewer

(continues)

Table 1-4 Subfeatures for functionality levels (continued)

Subfeature	Feature	Description
JobPriority	Jobs	Permits setting job priority, up to the user's maximum job priority
SelfNotification WithAttachment	Jobs	Activates e-mail notification for successful jobs
ShareDashboard	NA	Permits sharing dashboards when the user has the appropriate privileges
ShareFile	Documents	Permits sharing files when the user has the appropriate privileges
SubscribeChannel	Channels	Permits subscribing to channels

Subfeatures are specified one per line, enclosed within <SubfeatureID> and </SubfeatureID> tags.

- Available Actuate Analytics user experience levels

  Users can select their own Actuate Analytics user experience level on the Actuate Information Console Options page from the levels listed here. The following Actuate Analytics user experience levels are available at this functionality level:

  - Novice
  - Standard
  - Advanced

The following code shows a sample functionality level entry:

```
<Level>
   <Name>ViewAndSearch</Name>
   <Role>All</Role>
   <FeatureID>Jobs</FeatureID>
   <FeatureID>Documents</FeatureID>
   <FeatureID>Search</FeatureID>
   <SubfeatureID>ShareFile</SubfeatureID>
   <SubfeatureID>DeleteFile</SubfeatureID>
   <AnalyticsExperienceLevel>Novice</AnalyticsExperienceLevel>
   <AnalyticsExperienceLevel>Standard</AnalyticsExperienceLevel>
   <AnalyticsExperienceLevel>Advanced</AnalyticsExperienceLevel>
</Level>
```

The level is named ViewAndSearch and is available to all security roles. Users with ViewAndSearch functionality can run jobs, access documents, and search for files. In addition, they can share and delete their own files, and set their Actuate Analytics experience level to any of the available levels.

Customizing functionality level features

Customize functionality level features by modifying the action they perform and the graphic image they use. Features are defined in the functionality-level.config file. A feature definition consists of up to five parts:

- Feature ID
  This is the feature name and must be a unique alphanumeric string, enclosed within <ID> and </ID> tags. This value is used as the feature name in functionality level definitions. Do not change this value, because the IDs are used in the Actuate Information Console code to identify the features.

- Label key
  This key is used in the Actuate Information Console resource files. These files have names of the format, ActivePortalResources_<locale>.properties. The files are located in <context root>\WEB-INF\lib\resources.jar. If this file does not contain a resource file for a locale, the resource file, ActivePortalResources.properties, for the default locale, en_US, is used. The key provides for proper translation in the resource file so that the hyperlink text for the feature is displayed using the current locale. Keys are enclosed within <Labelkey> and </Labelkey> tags. Do not change the key values or the resource string substitution fails.

- Link
  This link is the target URI of the label key hyperlink, which is typically to the page that corresponds to the feature. Table 1-5 shows the targets for each feature. Links are enclosed within <Link> and </Link> tags. Change the link target for the feature by replacing the default page or action name.

Table 1-5 Actuate Information Console targets for features

Feature	Actuate Information Console target
Documents	\getfolderitems.do
Jobs	\selectjobs.do
Channels	\selectchannels.do
Search	\searchfiles.do
Customization	\customize.do

- Large icon and Small icon
  These optional icons are displayed together with the link, depending on the skin. For example, the Classic skin displays the large icons, the Treeview skin uses the small icons, and the Tabbed skin does not use these icons at all. Table 1-6 shows features and their icons. Large icons are 32 pixels square. Their file names are relative to the context root and are enclosed within <LargeIcon> and </LargeIcon> tags. Small icons are 16 pixels square. Their

file names are relative to the context root and are enclosed within <SmallIcon> and </SmallIcon> tags. Replace these file names with the names of your own icons to customize your skin's appearance.

Table 1-6 Icons for features

Feature	SmallIcon	LargeIcon
Documents	\images\ filesfoldersicon16x16.gif	\images \filesfoldersicon.gif
Jobs	\images\requestsicon16x16.gif	\images\requestsicon.gif
Channels	\images\channelsicon16x16.gif	\images\channelsicon.gif

The following example shows a sample definition for the Channels feature. This example specifies custom large and small icons. The Classic and Treeview skins, and any skins cloned from them, use these new images for the channel icon.

```
<Feature>
   <ID>Channels</ID>
   <Labelkey>SBAR_CHANNELS</Labelkey>
   <Link>/selectchannels.do</Link>
   <SmallIcon>/images/customIcon16x16.gif</SmallIcon>
   <LargeIcon>/images/customIcon32x32.gif</LargeIcon>
</Feature>
```

Preserving functionality levels and features

The functionality-levels.config file is overwritten during upgrade installations. This change ensures that new levels, features, and subfeatures are available to you with your new Actuate Information Console installation. If you have modified your existing functionality-level.config file, make a backup of the changes before the upgrade. Use the backed-up file to access your changes and merge them into the new functionality-level.config file.

Using Actuate Analytics experience levels

If you have purchased the Actuate Analytics Option, additional customization features are available. The Actuate Analytics Cube Viewer uses experience levels to disable Cube Viewer features based on the experience level that a user chooses.

Understanding experience levels

The following list shows the three standard Actuate Analytics experience levels:

- Novice
- Standard
- Advanced

The Actuate Information Console Administrator can edit the experience.levels configuration file that defines the levels to modify the levels and add additional level definitions. The location for the Actuate Information Console configuration file is <context root>\WEB-INF. The user can also choose their default experience level on the Actuate Information Console Options—General page. Actuate Information Console stores the choice as part of the user's profile. If the experience.levels file is missing, all functionality becomes available to all users.

Every Actuate Information Console functionality level includes a list of Actuate Analytics Cube Viewer experience level names. The experience level names must match the experience level configuration names in the experience.levels file. This list controls the experience levels available to the user for that functionality level. The user can choose among the experience levels available when viewing a cube.

Customizing experience levels

As the Actuate Information Console Administrator, you can create and modify experience levels. You modify a level by adding or removing HIDEITEM entries to hide or not hide a part of the Cube Viewer user interface. The following tables, which are organized by their user interface component, describe these entries. The Experience level column shows the most restricted experience level that displays the user interface component. For example, Standard indicates that both Standard and Advanced show the element but that Novice does not. An experience level of None means that none of the supplied levels show that element.

Display of the Cube Viewer horizontal bars are controlled with the elements in Table 1-7.

Table 1-7 Tags that control Cube Viewer horizontal bars

HIDEITEM tag keyword	Functionality	Experience level
ENTIRE_BANNER	Banner	Novice
ENTIRE_DIMENSIONBAR	Categories bar	Novice
ENTIRE_REPORTBAR	Report bar	None
ENTIRE_TOOLBAR	Toolbar	Novice
ENTIRE_TITLEBAR	Title bar	Novice

Cube Viewer toolbar buttons are controlled with the elements in Table 1-8.

Table 1-8 Tags that control Cube Viewer toolbar buttons

HIDEITEM tag keyword	Functionality	Experience level
ABOUT_TB	About	Novice

(continues)

Table 1-8 Tags that control Cube Viewer toolbar buttons (continued)

HIDEITEM tag keyword	Functionality	Experience level
methodS_TB	Calculate	Advanced
COLLABORATE_TB	Collaborate	Novice
EXCEPTION_TB	Exception highlighting	Advanced
HELP_TB	Help	Novice
HOME_TB	Home	None
HORIZONTAL_BAR_CHART_TB	Horizontal bar chart	Novice
LINE_GRAPH_TB	Line graph	Standard
PIE_CHART_TB	Pie chart	Novice
PREFERENCES_TB	Preferences	Novice
PRINT_TB	Print	Novice
SAVE_AS_TB	Save	Novice
EXPORT_TB	Save as Microsoft Excel	Advanced
EXPORTDOC_TB	Save as Microsoft Word	Advanced
TABLE_VIEW_TB	Table view	Novice
FIT_TO_PAGE_TB	Vertical or horizontal fit to page	Standard
VERTICAL_BAR_CHART_TB	Vertical bar chart	Novice
VIEW_TB	Presentation or analysis view	Standard
VIEWS_TB	Reports combo box	Standard
EDIT_UNDO_REDO_TB	Undo/Redo	Novice

Some toolbar buttons have their own menus. The menu items are controlled with the elements in Table 1-9, grouped by button.

Table 1-9 Tags that control Cube Viewer submenu items

HIDEITEM tag keyword	Functionality	Experience level
Save As submenu of the Save button		
SAVEAS_PDF	Adobe PDF (.pdf)	Novice
SAVEAS_SYLK	Microsoft Excel (.xls)	Novice
SAVEAS_RTF	Microsoft Word (.doc)	Novice
SAVEAS_CSV	Text (.txt comma separated)	Novice

Table 1-9 Tags that control Cube Viewer submenu items

HIDEITEM tag keyword	Functionality	Experience level
Save Assubmenu of Save button *(continued)*		
SAVEAS_TSV	Text (.txt tab separated)	Novice
SAVE_SESSION_LOCALLY	Work offline	Standard
Collaborate button menu items		
COLLABORATE_MENU_ITEM	Collaborate	Novice
SEND_REPORT_MENU_ITEM	Send report	Novice
Preference button menu items		
COLUMNS_MENUITEM	Columns	Novice
GENERAL_MENUITEM	General	Novice
ROWS_MENUITEM	Rows	Novice
Graphics submenu of Preference button		
BAR_MENUITEM	Bar	Novice
EXPERIENCE_LEVELS_ CASCADE_MENUITEM	Experience levels	Novice
LINE_MENUITEM	Line	Novice
PIE_MENUITEM	Pie	Novice
Calculate button menu items		
AVERAGE	Average	Advanced
DIFFERENCE	Difference	Advanced
GROWTH_PERCENT	%Growth	Advanced
INTERSECT	Intersection (AND)	Advanced
MAX	Maximum value	Advanced
MIN	Minimum value	Advanced
PERFORMANCE_INDEX	Performance index	None
RATIO	Ratio	Advanced
RELATIVE_TIME_PERIOD	Relative time periods	Advanced
SUM	Sum	Advanced
PERCENTAGE_OF_ TOTAL_MENU_ITEM	% of total	Advanced
UNION	Union (OR)	Advanced

(continues)

Table 1-9 Tags that control Cube Viewer submenu items

HIDEITEM tag keyword	Functionality	Experience level
Reports button menu items		
HOME	Home	Standard
ADMIN_LOG_ON	Log on as view administrator	Advanced
REPORTS_BOX	Save	Novice
ADD_TITLE	Save as	Standard

The menu items shown after a right-click on a table view, bar chart view, or line chart view are controlled with the elements that are shown in Table 1-10.

Table 1-10 Tags that control Cube Viewer context menu items

HIDEITEM tag keyword	Functionality	Experience level
Right-click on table view		
COLUMN_PERCENTAGES	% of column total	Advanced
GLOBAL_PERCENTAGES	% of grand total	Advanced
ROW_PERCENTAGES	% of row total	Advanced
DRILL_UP	Collapse	Novice
VIEW_SOURCE	Drill through to details	Advanced
DRILL_DOWN	Expand	Novice
EXPORT_SOURCE_TO_ SPREADSHEET	Export details to spreadsheet	Advanced
FORMAT_SCALE	Format scale…	Standard
GENERAL_PROPERTIES	Preferences	Standard
VIEW_NUMERIC_DATA	Show cell value with calculation results	Novice
Right-click on a bar in the bar view		
HISTOGRAM_PROPERTIES	Bar chart preferences	Standard
Right-click on the line chart view		
LINE_PROPERTIES	Line preferences	Standard

The menu items shown after right-click on various axis components are controlled with the elements shown in Table 1-11.

Table 1-11 Tags that control Cube Viewer axis context menu items

HIDEITEM tag keyword	Functionality	Experience level
Right-click on the column axis header		
COLUMN_PROPERTIES	Column preferences	Standard
Right-click on the row axis header		
ROW_PROPERTIES	Row preferences	Standard
Right-click on axes, submenu under sort		
CUSTOM_SORT	Custom…	Standard
DESCENDING	Highest to lowest	Novice
ASCENDING	Lowest to highest	Novice
TOP_10	Show highest ten	Novice
BOTTOM_10	Show lowest ten	Novice
Right-click on the axes		
APPLY_FILTER	Apply or cancel filter…	Standard
DRILL_UP	Collapse	Novice
Right-click on the axis header		
FILTER_MENU_ITEM	Filter	Standard
Right-click on axis		
EDIT_method	Edit calculation	Standard
DRILL_DOWN_INTO	Expand into	Novice
DRILL_TO_LEVEL	Expand to level	Standard
RELATIVE_DATE_FILTER_MENU_ITEM	Filter by relative time periods…	Advanced
HIDE_SELECTED_UNSELECTED	Hide selected or unselected subcategories	Standard
HIDE_ROWS_COLUMNS_WITH_NO_DATA_IN	Hide rows and columns with no data	Standard
HIDE_ROWS_COLUMNS_WITH_ZEROS_IN	Hide rows and columns with zeros	Standard
SHOW_ALL_CATEGORIES	Cancel hiding	Standard

(continues)

Table 1-11 Tags that control Cube Viewer axis context menu items (continued)

HIDEITEM tag keyword	Functionality	Experience level
Right-click on axis *(continued)*		
COLUMN_TOTALS	Show column totals as	Standard
ROW_TOTALS	Show row totals as	Standard
LABEL_STYLE_CASCADE_ MENU_ITEM	Show labels as	Standard
Right-click on axis header		
SEARCH_MENU_ITEM	Search…	Standard
PIVOT_AXES	Swap rows and columns	Novice
Right-click on pie view		
LIMIT_SLICES_BY_THRESHOLD	Limit slices by threshold	Standard
LIMIT_SLICES_BY_VIEWPORT	Limit slices by view port	Standard
PIE_PROPERTIES	Pie preferences	Standard
SHOW_ALL_SLICES	Show all slices	Standard
SHOW_NEXT_LARGEST_ CATEGORY	Show next largest slice	Standard
Right-click on row or column totals right-click menu item		
LEADING_CASCADE_ITEM	Leading	Novice
NONE_CASCADE_ITEM	None	Novice
TRAILING_CASCADE_ITEM	Trailing	Novice

When modifying elements in experience.levels, do not remove the following entries:

```
<HIDEITEM>EDIT_TITLE</HIDEITEM>
<HIDEITEM>ADMIN_LOG_ON</HIDEITEM>
<HIDEITEM>VIEW_SOURCE</HIDEITEM>
```

These features are not supported in Actuate Analytics, and these elements must appear under every <EXPERIENCE_LEVEL> element. The NUMBER_OF_LEVELS element value must correspond to the number of experience levels defined in the file. The DEFAULT_EXPERIENCE_LEVEL element value specifies the default experience level to use if no level is specified for a user. For information about using experience level items in the Cube Viewer, see *Working with Cube Reports using Actuate Analytics Option.*

How to add an experience level

1. Using a text editor that supports UTF-8 encoding, open experience.levels. In some system configurations, this file does not already exist in the WEB-INF directory for your application. Some editors, such as Microsoft Notepad, add a marker string to the file to identify the UTF-8 encoding. Do not use an editor that adds hidden information to the file.
2. Find the <NUMBER_OF_LEVELS> tag and increase the number of levels by one. There are three levels in the standard experience.levels, so set the new value to 4:

   ```
   <NUMBER_OF_LEVELS>4</NUMBER_OF_LEVELS>
   ```

3. Find the last </EXPERIENCE_LEVEL> tag
   1. Insert the following code after the </EXPERIENCE_LEVEL> tag and before the </EXPERIENCE_LEVELS> tag:

      ```
      <EXPERIENCE_LEVEL>
         <SKIN_NAME>SampleLevel</SKIN_NAME>
      ```

      The skin name is used by Actuate Information Console on the General—Options page.
   2. Insert code in one of the following formats for the display name that Actuate Analytics Cube Viewer uses:
      - To use a static value for the display name:

        ```
        </DISPLAY_NAME>
           <LOCALE_ID>en_US</LOCALE_ID>
           <NAME>Sample</NAME>
        </DISPLAY_NAME>
        ```

      - To use a resource key to access the display name in the appropriate locale:

        ```
        <DISPLAY_NAME>SAMPLE_EXP_LEVEL_ID<DISPLAY_NAME>
        ```

   3. Insert the following code after the <DISPLAY_NAME> tag line:

      ```
      <HIDEITEM>HOME_TB</HIDEITEM>
         <HIDEITEM>EDIT_TITLE</HIDEITEM>
         <HIDEITEM>ADMIN_LOG_ON</HIDEITEM>
         <HIDEITEM>VIEW_SOURCE</HIDEITEM>
      </EXPERIENCE_LEVEL>
      ```

4. Save and close experience.levels.
5. Using a text editor, open functionality-level.config.
6. Add the following line to the list of Analytics experience levels in every level:

   ```
   <AnalyticsExperienceLevel>Sample</AnalyticsExperienceLevel>
   ```

For example, the Basic level should look like the following code:

```
<Level>
   <Name>Basic</Name>
   <Role>All</Role>
   <FeatureID>Jobs</FeatureID>
   <FeatureID>Documents</FeatureID>
   <FeatureID>Channels</FeatureID>
   <SubfeatureID>DeleteFile</SubfeatureID>
   <SubfeatureID>InteractiveViewing</SubfeatureID>
   <AnalyticsExperienceLevel>Novice</AnalyticsExperienceLevel>
   <AnalyticsExperienceLevel>Standard</AnalyticsExperienceLevel>
   <AnalyticsExperienceLevel>Advanced</AnalyticsExperienceLevel>
   <AnalyticsExperienceLevel>Sample</AnalyticsExperienceLevel>
</Level>
```

7 Save and close functionality-level.config.

8 To apply these configuration changes, restart your application server or JSP engine. For example, to restart Information Console's embedded servlet engine on a Windows XP system, perform the following steps:

   1 From the Windows Start menu, choose Settings➤Control Panel.

   2 Choose Administrative Tools.

   3 Choose Services.

   4 On Services, select Actuate 11 Apache Tomcat for Information Console.

   5 From the menu, choose Action➤Restart.

   6 Close Services.

How to select a new experience level

1 Choose Start➤Programs➤Actuate 11➤Information Console.

2 Log in to Actuate Information Console.

3 In Documents, choose Options.

4 In General, select the Analytics Experience Level and view the levels that appear in the list, as shown in Figure 1-6. The new Sample level is in the list.

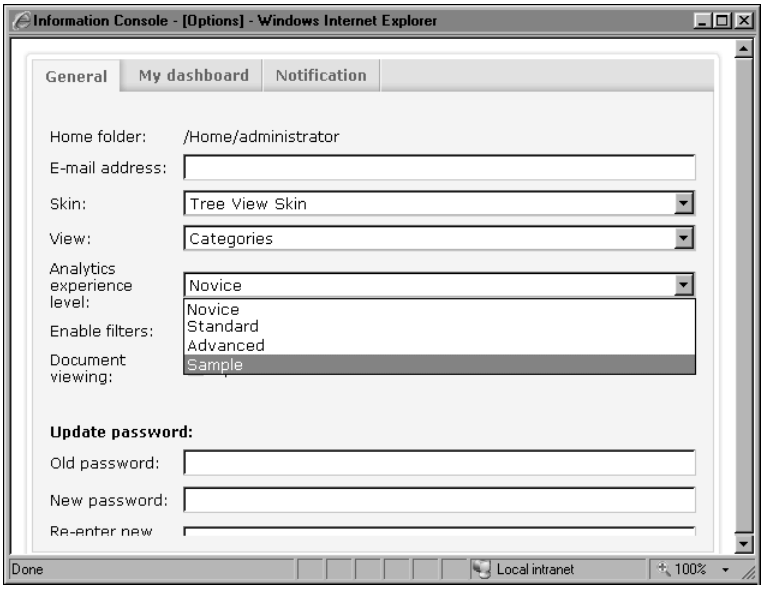

Figure 1-6 Customized Analytics experience levels