

# Actuate e.Reporting Suite 4



Building an e.Reporting Web Site  
Release 4.1

Information in this document is subject to change without notice. Examples provided are fictitious. No part of this document may be reproduced or transmitted in any form, or by any means, electronic or mechanical, for any purpose, in whole or in part, without the express written permission of Actuate Corporation.

© 1995 - 2000 by Actuate Corporation. All rights reserved. Printed in the United States of America.

Contains information proprietary to:

Actuate Corporation  
701 Gateway Boulevard  
South San Francisco, CA 94080  
<http://www.actuate.com>

The software described in this manual is provided by Actuate Corporation under an Actuate License agreement. The software may be used only in accordance with the terms of the agreement.

Actuate Corporation trademarks and registered trademarks:  
Actuate, e.Report, e.Reporting, Live Report Document, Live Report Extension, ReportBlast, ReportCast, Report Encyclopedia, SmartSearch, Transporter, Virtual Report Distribution, and XML Reports.

Third party trademarks:

ActiveX, Microsoft, MSN, The Microsoft Network, Windows, Windows NT, Windows 95/98 and/or other Microsoft products are either trademarks or registered trademarks of Microsoft. CT-Library is a trademark of Sybase, Inc. Digital Creations and DCLC, are trademarks of Digital Creations, LC. Graphics Server is a trademark of Bits Per Second, Ltd. and Pinnacle Publishing, Inc. HDK is a registered trademark of Virtual Media Technology Pty Ltd. INFORMIX-ESQL/C is a trademark of Informix Software, Inc. InstallShield is a registered trademark of InstallShield Corporation. LEADTOOLS is a registered trademark of LEAD Technologies, Inc. Netscape, Netscape Communications, Netscape Communicator, Netscape Enterprise Server, Netscape FastTrack Server, and Netscape Navigator are either trademarks or registered trademarks of Netscape Communications Corporation. NobleNet and WinRPC are trademarks of NobleNet, Inc. ONC, Solaris, SPARC, Sun, and Sun Microsystems are trademarks of Sun Microsystems. Oracle Call Interface is a trademark of Oracle Corporation. Progress is a registered trademark of Progress Software Corporation. Sheridan Calendar Widgets is a trademark of Sheridan Software Systems, Inc. SmartHeap is a trademark of MicroQuill Software Publishing, Inc. Tools.h++ is a trademark of Rogue Wave Software, Inc. UNIX is a registered trademark of X/Open Company, Ltd. XPrinter is a trademark of Bristol Technology, Inc. XPAT, created by James Clark, is licenced under the Mozilla licence agreement. WinWidgets is a trademark of Simple Software, Inc.

All other brand or product names are trademarks or registered trademarks of their respective companies or organizations.

Document No. 000501-2-630301 May 19, 2000

# Contents

<b>Introduction</b> .....	<b>vii</b>
About Actuate e.Reporting Suite 4. ....	vii
About <i>Building an e.Reporting Web Site</i> .....	x
About Actuate e.Reporting Server product .....	xi
Online documentation. ....	xii
Using online manuals .....	xiii
Online help .....	xiii
Using context-sensitive online help .....	xiii
Using the Actuate online help system. ....	xiii
Using report-specific online help .....	xvi
Typographical conventions .....	xvii
Syntax conventions .....	xviii
Chapter 1	
<b>Using ReportCast</b> .....	<b>1</b>
About Actuate ReportCast .....	2
Setting up ReportCast .....	3
Navigating the Report Encyclopedia on the Web .....	4
Accessing the Report Encyclopedia on the Web. ....	4
Accessing channels and reports. ....	6
Accessing, generating, and searching reports on the Web. ....	7
Generating a report. ....	7
Setting report viewing options .....	10
Searching DHTML reports. ....	11
Navigating through the Report Encyclopedia. ....	14
Working with ReportCast channels .....	14
Subscribing to a ReportCast channel .....	14
Unsubscribing from a ReportCast channel. ....	14
Chapter 2	
<b>Administering ReportCast</b> .....	<b>17</b>
About ReportCast administration .....	18
Starting and stopping ReportCast .....	18
Starting and stopping ReportCast on an NT system .....	18
Starting and stopping ReportCast on a UNIX system .....	19
Starting a UNIX ReportCast service .....	19
Stopping a UNIX ReportCast Server .....	19
Setting ReportCast default e.Reporting Server .....	20
Using non-default web server ports .....	21

Configuring ReportCast port numbers . . . . .	21
Creating custom configurations . . . . .	23
Mapping custom configurations to locales . . . . .	24
Supporting locales with e.Reporting Server and ReportCast. . . . .	26
Accessing UNIX ReportCast with CGI scripts . . . . .	27
Working with reports on Windows web sites . . . . .	28
Using CGI scripts for security. . . . .	28
Administering ReportCast channels . . . . .	29
About Actuate ReportCast channels . . . . .	29
URLs in e-mail notification . . . . .	34
Tuning ReportCast performance . . . . .	35
Setting user connection timeouts . . . . .	35
Setting request timeouts . . . . .	36
Tuning viewing performance . . . . .	36

## Chapter 3

### **Working with ReportCast templates . . . . . 39**

About ReportCast templates . . . . .	40
About HTML template files . . . . .	40
About internal template structure. . . . .	43
About template placement and scope. . . . .	43
About the ReportCast object model. . . . .	45
ReportCast objects . . . . .	45
Working with the ReportCast object model. . . . .	46
Understanding objects available to a template . . . . .	47
About the web server template directory structure . . . . .	47
Creating and customizing ReportCast template files . . . . .	49
Template file elements. . . . .	49
Modifying HTML template files. . . . .	50
Locating ReportCast template files . . . . .	51
Understanding ReportCast searches for templates. . . . .	51
Creating custom requester forms for a specific report. . . . .	53
Reloading template files . . . . .	54
Naming template files . . . . .	55
Placing custom templates in a Report Encyclopedia. . . . .	57
Creating custom templates . . . . .	61
Customizing the DHTML and SmartSearch displays. . . . .	65

## Chapter 4

### **ReportCast directives . . . . . 67**

About ReportCast directives . . . . .	68
Sending directives. . . . .	68

Managing directives . . . . .	.69
Formatting directives . . . . .	.70
Types of ReportCast directives . . . . .	.72
ReportCast directives . . . . .	.73
Directives syntax . . . . .	.75
Folder directives . . . . .	.75
Viewing DHTML reports . . . . .	.78
Search directives . . . . .	.82
File directives . . . . .	.83
Factory directives . . . . .	.85
Using submit . . . . .	.89
Using status directives . . . . .	.95
Using information directives . . . . .	.97
About Channel directives . . . . .	.98
Listing ReportCast channels . . . . .	.99
Administering ReportCast channels . . . . .	.99
Displaying channel contents . . . . .	100
Creating channels using channel URLs . . . . .	100
Subscribing to channels using channel URLs . . . . .	101
Examples of ReportCast directives . . . . .	102

## Chapter 5

### **ReportCast scripting language reference . . . . . 105**

About the ReportCast scripting language . . . . .	106
ReportCast scripting language reference . . . . .	106
List command . . . . .	107
List files command . . . . .	110
List versions command . . . . .	112
List folders command . . . . .	112
List items command . . . . .	113
List requestforms command . . . . .	113
List requests . . . . .	114
List paramgroups, list parameters commands . . . . .	115
List values command . . . . .	117
List status command . . . . .	118
List members command . . . . .	119
List grants command . . . . .	120
List privileges command . . . . .	121
List notifications command . . . . .	122
List schedules command . . . . .	122
List printerprops command . . . . .	123
List channels command . . . . .	123

List reportservers command . . . . .	124
List pathlinks command . . . . .	124
List expirationpolicies command . . . . .	126
List tocitems command . . . . .	126
List criteria command . . . . .	126
List resulthits command . . . . .	127
List resultfields command . . . . .	127
If command . . . . .	128
Include command . . . . .	128
With command . . . . .	129
Set command . . . . .	130
ReportCast scripting language variables . . . . .	131
ReportCast variables . . . . .	131
Web server variable . . . . .	132
Web request variables . . . . .	133
e.Reporting Server variables . . . . .	133
View session variables . . . . .	135
Table of contents item variables . . . . .	135
Search criterion variables . . . . .	135
Search result set variables . . . . .	136
Search result hit variables . . . . .	136
Search result field variables . . . . .	137
Current folder variables . . . . .	137
Folder item variables . . . . .	137
Current request folder . . . . .	139
Report definition . . . . .	139
Request item variables . . . . .	140
Member variables . . . . .	142
Privilege variables . . . . .	143
Request status variables . . . . .	143
Parameter group variable . . . . .	143
Parameter variables . . . . .	143
Schedule variable . . . . .	144
Printer variables . . . . .	145
ReportCast Channel variables . . . . .	145
Error variables . . . . .	146

## Chapter 6

### **Internal ReportCast security . . . . . 147**

About ReportCast security . . . . .	148
Setting the ReportCast security variables . . . . .	148
Providing security for ReportCast administration . . . . .	150
Logging on and authenticating users . . . . .	150
Logging on as anonymous . . . . .	150

Logging on as an e.Reporting Server user . . . . .	151
Authenticating users using forms. . . . .	151
Security and the Apache web server. . . . .	152
Customizing the Login dialog box . . . . .	152
Setting the idle user session timeout. . . . .	152
Modifying the HTTP unauthorized status code . . . . .	153
About the ReportCast Security Extension. . . . .	153
Overview . . . . .	153
Developing the ReportCast Security Extension . . . . .	157
AcWebSecurityStart function . . . . .	158
AcWebSecurityAuthenticate function . . . . .	158
AcWebSecurityStop function . . . . .	160
Development considerations . . . . .	160
NT-specific information . . . . .	162
Server-specific information . . . . .	162
Netscape server . . . . .	162
Microsoft Internet Information server. . . . .	162
Configuring the ReportCast Security Extension . . . . .	163
AC_WEBAGENT_SECURITY_LIBRARY variable . . . . .	164

## Chapter 7

### **Security for Internet deployment . . . . . 165**

About network security and Actuate . . . . .	166
Data protection strategies . . . . .	166
Protecting corporate data . . . . .	166
Understanding Actuate component communication . . . . .	167
Establishing the e.Reporting Server port. . . . .	168
Configuring the e.Reporting Server port. . . . .	169
Configuring cached connection management . . . . .	170
Web browser to web server communication. . . . .	170
Connecting to and browsing the Report Encyclopedia. . . . .	170
Running and viewing reports . . . . .	171
Running reports . . . . .	171
Configuring data reply ports. . . . .	171
Viewing e.reports as DHTML, PDF, or XML . . . . .	173
Configuring the view process port . . . . .	174
Viewing e.reports using the LRX. . . . .	175
Connecting to the Report Encyclopedia with the split ReportCast configuration for UNIX	176
Configuring the split ReportCast configuration for UNIX . . . . .	176
Deploying Actuate products in Internet environments . . . . .	178
Understanding configuration options and firewall setup . . . . .	178
Dual firewalls with ReportCast in the DMZ. . . . .	179

NAT considerations for dual firewalls with ReportCast in the DMZ . . . . .	181
Dual firewalls with a proxy server . . . . .	182
NAT considerations for dual firewalls with a proxy server . . . . .	183
Split ReportCast installation for UNIX . . . . .	183
NAT considerations for the split ReportCast configuration. . . . .	184
Placing the e. Reporting Server in the DMZ . . . . .	184
<b>Index . . . . .</b>	<b>185</b>

---

## About Actuate e.Reporting Suite 4

e.Business customers use Actuate® e.Reporting Suite 4 to develop and deploy high resolution structured content to tens of thousands of users. Actuate takes web reporting to the next level by providing options for various needs as varied as seamless personalized web pages and traditional online and printed reports.

Customers building e.Business sites face challenges where Actuate e.Reporting Suite 4 offers the following solutions.

---

<b>Challenge</b>	<b>Actuate solution</b>
Developing custom content using HTML	DHTML provides a fast, no download option
Using plug-ins to view structured content	No need to distribute and install plug-ins for tens of thousands of users
Compromising information display due to lack of integrated tools	Provides template based design and display
Exploding use of web-based content delivery	Ability to support a million hits per day per CPU
Demand for personalized secure content	Page security for tens of thousands of users
Reuse of existing content	Open server provides access to content from other applications

<b>Challenge</b>	<b>Actuate solution</b>
Maintaining data integrity on hard copy	PDF provides high resolution printed copy
Portability of electronic data or content to other applications	XML output provides access to data across applications

Actuate tools and reports:

- Solve complex data access problems
- Solve formatting problems that go beyond the scope of other tools
- Scale to support thousands of users

The following summary describes the products in the Actuate e.Reporting Suite 4.

<b>Product name</b>	<b>Use</b>
Actuate Developer Workbench	An object-oriented application used by professional developers of structured content to design, build, and distribute report object designs and components throughout the enterprise. The Actuate Basic Language and Actuate Foundation Class Library support extensive customization capabilities.
Actuate e.Report Designer	An application that complements the Developer Workbench and is used by business users to design and distribute a variety of reports. These reports require no programming. This application supports both modification of complex reports and using sophisticated components from libraries.
Actuate End User Desktop	An application used by end users to request, generate, view, and print live report documents on networked client systems. The ReportQuery™ capabilities enable seamless transfer of data from an Actuate report to any productivity tool or analysis tool.
Actuate Viewer	Application for end users to find, view, and print report documents. The ReportQuery™ capabilities are also part of the Actuate Viewer.

Product name	Use
Actuate e.Reporting Server	<p>A server application that generates Live Report Documents, manages them in the Report Encyclopedia®, and makes them available to users.</p> <p>This product includes with Actuate Administrator Desktop, an application for system and network administrators to manage and control one or more Actuate report servers.</p> <p>This product also includes Actuate ReportCast™ that transforms the Report Encyclopedia into a dynamic, secure web site. ReportCast provides the foundation for Channels and seamless integration with other web sites.</p> <p>This product includes Actuate Live Reports Extension (LRX) that works with both Microsoft Internet Explorer and Netscape Navigator to support report viewing and printing.</p>
Actuate Advanced e.Reporting Server	<p>An application that adds two capabilities to the basic e.Reporting Server, open server and page security.</p> <ul style="list-style-type: none"> <li>■ Open server supports the use of third-party report generators with the Actuate e.Reporting Server.</li> <li>■ Page security supports personalized viewing of parts of reports for different users.</li> </ul>

Product name	Use
Actuate Software Development Kit	<p data-bbox="705 210 1198 296">Actuate ActiveX Controls embed Actuate reporting functionality into custom applications.</p> <p data-bbox="705 305 1229 510">Actuate Requester API accesses attributes and values of report parameters, changes the values of report parameters, controls how and when a report is generated, displays and prints reports, and configures report print setup. Access the Requester API using Actuate Basic, Visual Basic, C, or C++.</p> <p data-bbox="705 520 1182 635">Actuate search extension API supports developing search extensions to transfer data to any third-party productivity or analysis tool.</p> <p data-bbox="705 645 1208 878">Actuate report server API implements common Report Encyclopedia tasks, integrates report server features into existing corporate applications, automates routine or time-consuming tasks, and implements new feature groupings for custom business processes. Access the report server API using C++.</p> <p data-bbox="705 888 1198 975">Actuate Report Server Security Extension supports the use of third-party security tools.</p> <p data-bbox="705 986 1208 1065">Actuate archive driver supports the use of third-party archiving software and hardware.</p>

Actuate Viewer and Actuate Live Report Extension (LRX) are included with all products except Actuate Software Development Kit.

---

## About *Building an e.Reporting Web Site*

*Building an e.Reporting Web Site* is a guide to providing access to the Actuate Report Encyclopedia on the World Wide Web.

*Building an e.Reporting Web Site* includes the following chapters:

- *Introduction.* This chapter provides an overview of this guide, the Actuate e.Reporting Suite, and the typographical conventions used.

- *Chapter 1. Using ReportCast.* This chapter introduces the e.reporting web site, and explains what you need to deploy, navigate, and access e.reports over the Web. This chapter describes how to work with ReportCast channels.
- *Chapter 2. Administering ReportCast.* This chapter describes how to administer ReportCast. This chapter explains how to start and stop ReportCast, configuration tasks, using CGI with ReportCast, and tuning ReportCast performance.
- *Chapter 3. Working with ReportCast templates.* This chapter describes ReportCast templates, and explains how to create custom templates.
- *Chapter 4. ReportCast directives.* This chapter describes the ReportCast directives.
- *Chapter 5. ReportCast scripting language reference.* This chapter describes ReportCast's template scripting language and the ReportCast scripting language variables.
- *Chapter 6. Internal ReportCast security.* This chapter describes the ReportCast user authentication and security facilities.

---

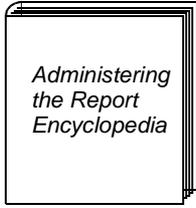
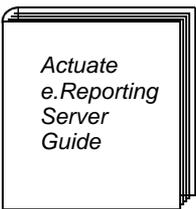
## About Actuate e.Reporting Server product

Actuate e.Reporting Server documentation includes a printed manual, an installation card, online help, user documentation in PDF files and in HTML format, and release notes. Information about the product that could not be included before the book printing deadline is in the release notes.

The Actuate web site, <http://www.actuate.com>, contains late-breaking news about the product and its features, as well as product update information. To obtain the password necessary to access the portion of the web site available only to customers, telephone Actuate Customer Support. The engineers in Actuate Customer Support can also help you with technical questions about the product according to your service contract. The Customer Support telephone number, fax number, and e-mail information can be found among the printed materials in the product box.

The printed documentation includes the following manuals.

<b>For information about</b>	<b>See the following</b>
Installation	Installation card
Late-breaking information about the software and documentation	Release notes

For information about	See the following
<p>Users, groups, privileges, and roles</p> <p>Printers and printing requests</p> <p>Process groups</p> <p>Building and managing your first Report Encyclopedia</p>	 <p><i>Administering the Report Encyclopedia</i></p>
<p>Viewing, running, finding, and printing reports</p>	 <p><i>Using e.Reports</i></p>
<p>Overview of the Actuate report server architecture</p> <p>Report server administration</p> <p>Database connections</p> <p>Report Encyclopedia utilities</p>	 <p><i>Actuate e.Reporting Server Guide</i></p>
<p>Actuate ReportCast, its templates, URL directives, scripting language, security and user authentication facilities.</p> <p>Actuate Live Report Extension (LRX)</p>	 <p><i>Building an e.Reporting Web Site</i></p>

## Online documentation

The information in the printed manuals is also available as online books in Adobe Acrobat PDF format and in the online help system for the Actuate products. For products without a Windows interface such as the Actuate e.Reporting Server and the Actuate ReportCast, we provide HTML help files.

## Using online manuals

The online manuals do not install automatically with the product. On the product CD, you find those files in the Manuals directory. Open the introductory PDF file to get an overview of the manuals. Copy this file and the files for each book you want to be able to use online to your local drive. The items in the table of contents and the page numbers in the index both contain links to the appropriate topics in the text. In the index, you access the link by positioning your cursor over the page number, not the topic.

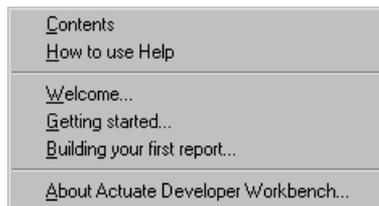
The HTML files install automatically with the product. These files can be viewed with standard browsers.

## Online help

Actuate products provide both context-sensitive online help about the product and report-specific online help about the report you are viewing. The Actuate e.Reporting Suite makes it possible for developers to create customized report-specific online help. Report-specific online help is only available when you view a report using an Actuate desktop product, such as Actuate End User Desktop, Actuate Viewer, or one of the Actuate LRXs.

## Using context-sensitive online help

Sections from the printed manuals have been linked directly to the software interface to make relevant information available while you work. Dialog boxes that need additional explanation about how to use them have a help button. To access online help, use the Help menu in the menu bar.



An alternative method is to use the F1 key.

## Using the Actuate online help system

Use two windows to access and view information in the e.Reporting Suite help system. The window on the right displays the contents of the online help

topics. The window on the left displays the table of contents or the index of the online help system.

Use the Sort button to view an alphabetical listing of help topics

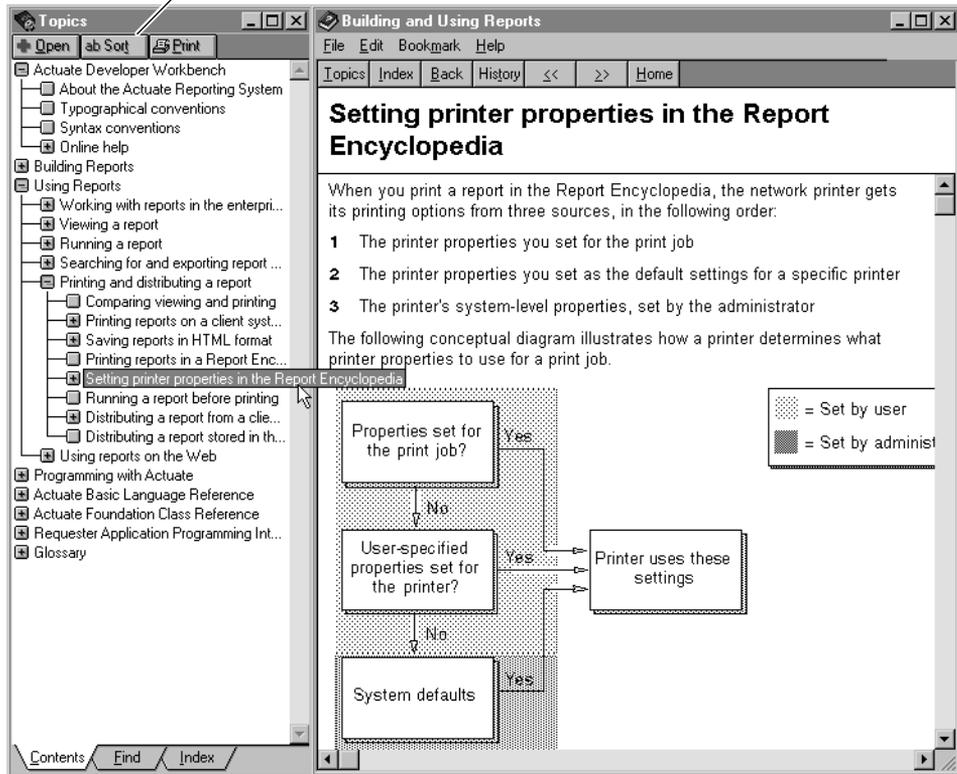


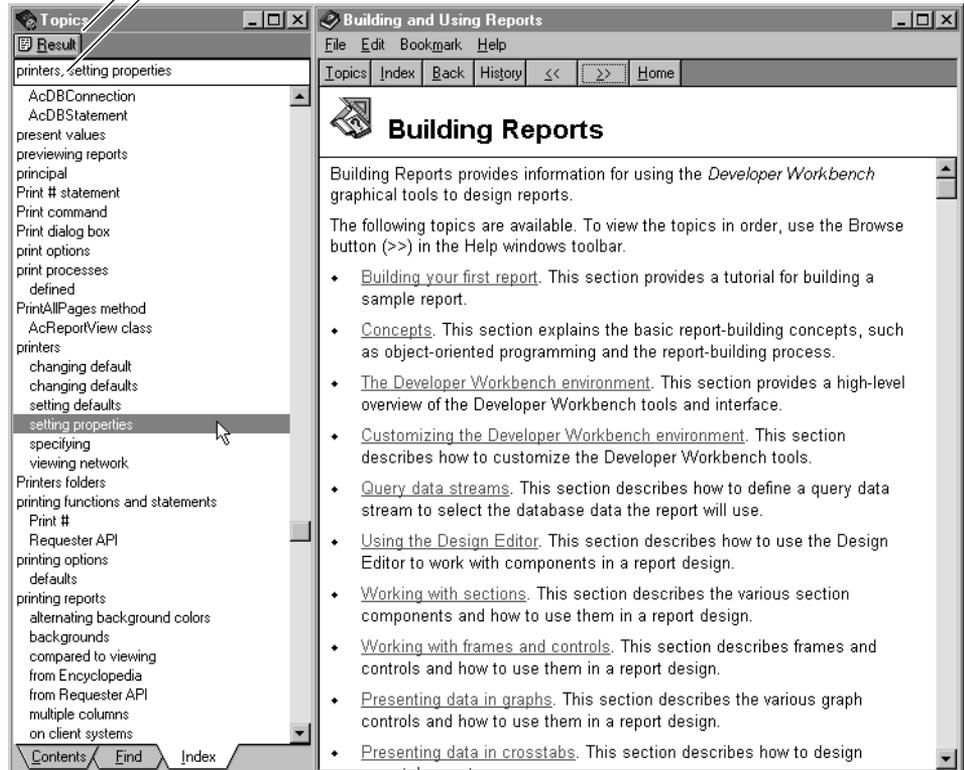
Table of Contents, Find, and Index tabs

The tabs at the bottom of the left window access different views. Use these tabs to switch views among the Table of Contents, Find, and Index. The buttons at the top of the window change according to the tab you select at the bottom.

The following illustrations show an example of the Index and the result of an Index search.

Use the Results button to view the topics

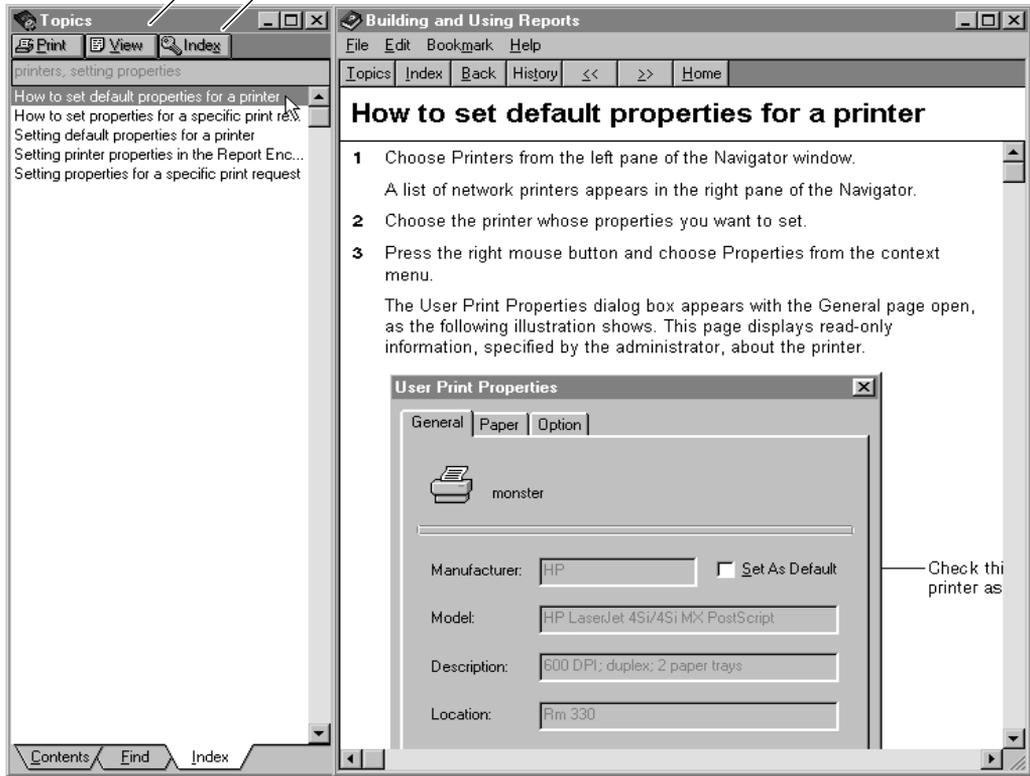
Enter the keyword to search the Index



The following illustration shows the result of the search as it appears in the window on the left. To view the topic in the right window double-click the topic from the list. The topic displays in the window on the right.

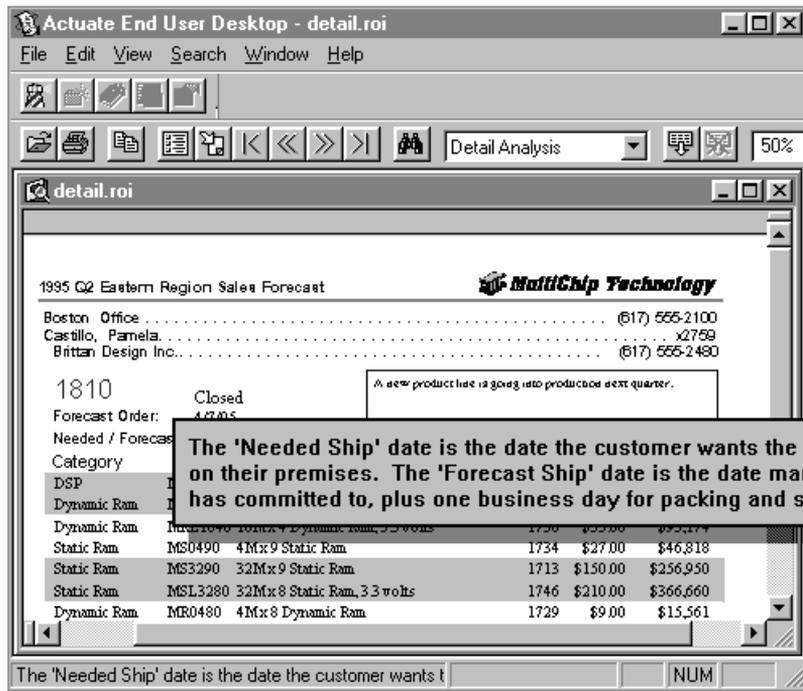
Use the View button to view the topic in the window on the right

Use the Index button to return to the keyword index



## Using report-specific online help

During the design phase, report developers have the option to include report-specific online help. For example, the report developer can add comments to give further detail about specific report objects or to explain how calculations were created.



For detailed information about report-specific online help see Chapter 3, "Viewing a report from the desktop Navigator," in *Using e.Reports*.

## Typographical conventions

The following table describes the typographical conventions.

Item	Convention	Example
Code examples	Sans serif	Dim As String
File names	All uppercase letters	DETAIL.ROI
Key combination	A + sign between the keys means to press both keys at the same time	Ctrl+Shift
Menu items	Capitalized. No bold	File

<b>Item</b>	<b>Convention</b>	<b>Example</b>
Submenu items	Separated from main menu item with small arrow	File→New
User input or user response	Enclosed in quotation marks	"M*16*"

---

## Syntax conventions

The following table describes the symbols used to present the syntax of Actuate Basic language elements.

<b>Symbol</b>	<b>Description</b>	<b>Example</b>
[]	Optional item	[Alias<alias name>]
<>	Argument you must supply	<expression to format>
{}	Groups two or more mutually exclusive options or arguments	{While   Until}
	Separates mutually exclusive options or arguments in a group	Exit {Do   For   Function   Sub}

# Using ReportCast

This chapter contains the following topics:

- About Actuate ReportCast
- Setting up ReportCast
- Navigating the Report Encyclopedia on the Web
- Accessing, generating, and searching reports on the Web
- Working with ReportCast channels

# About Actuate ReportCast

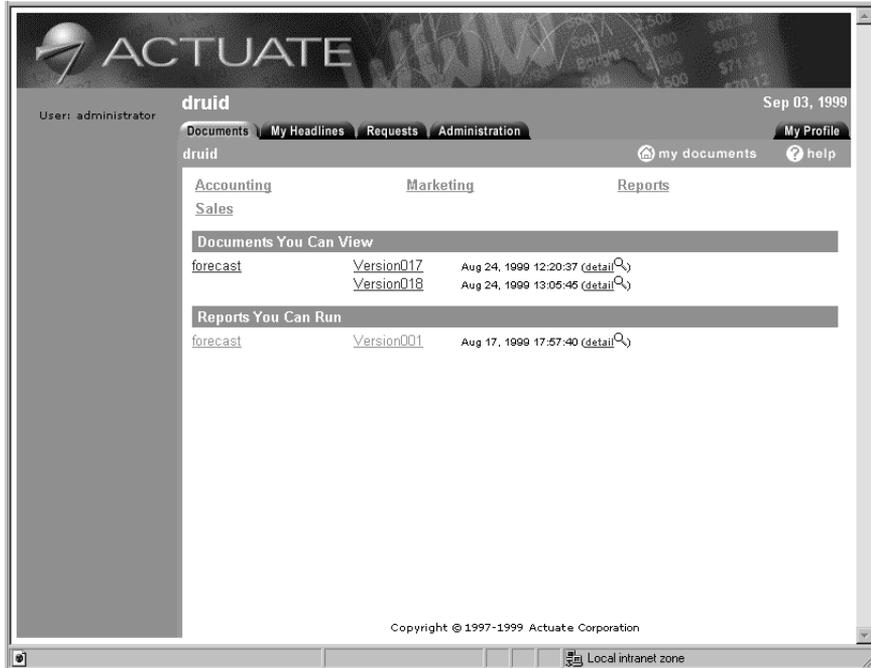
Actuate ReportCast provides access to the Actuate e.Reporting Server from the World Wide Web. Using ReportCast, you can access and work with reports in the Actuate Report Encyclopedia with any web browser.

ReportCast technology is a dynamic-link library (DLL) that runs as part of your web server. There is no user interface to the DLL. You confirm the successful installation by viewing the newly-installed reporting web site in your web browser.

Previously, to view Actuate reports over the Web you used Actuate LRX (Live Report Extension) for Netscape or Microsoft plug-ins installed with your web browser. With ReportCast, you can view e.reports directly using any web browser. The navigation options of the two viewing methods differ slightly.

ReportCast generates HTML pages that are adaptations of the Actuate Navigator interface, then sends those HTML pages to the web browser. For example, an HTML form that accepts parameter values for report generation replaces the Requester dialog box of the Navigator. Using ReportCast technology you can access HTML reports and DHTML reports without needing any additional software installed on your desktop system.

The following illustration shows the default reporting web site.



The URL format of the default reporting web site is:

`http://<webserver>/acweb/<reportserver>`

- <web server> is the name of the web server running the web interface to the Report Encyclopedia.
- acweb is a keyword indicating that you want to access the Report Encyclopedia.
- <report server> is the name of the e.Reporting Server with the encyclopedia you want to access.

Users navigate the reporting web site by clicking links on the reporting web page, or by entering ReportCast directives as part of a URL.

ReportCast provides a set of default web page templates that determine the appearance of the pages that display in the browser. Commands can be written in a combination of the ReportCast scripting language, Javascript, and standard HTML tags and are embedded in the template. Webmasters have these options for using the templates:

- Use the default templates
- Customize the default templates
- Create entirely new templates

---

## Setting up ReportCast

As you create structured content from databases, you deploy e.reports over the World Wide Web. A typical setup includes:

- A web server
- An Actuate installation consisting of:
  - e.Reporting Server,
  - an e.report development tool such as Developer Workbench or e.Report Designer
  - Administrator Desktop
- ReportCast

A webmaster must have administrator privileges to the web server and report server to effectively manage the e.reporting web site.

---

## Navigating the Report Encyclopedia on the Web

Actuate's web interface to the Report Encyclopedia allows you to access and work with reports on the Report Encyclopedia using a web browser. The interface provides web-based Actuate Navigator functionality similar to that provided by the End User Desktop and Administrator Desktop. The web interface also provides a service called ReportCast channels. ReportCast channels use push technology to provide reports to particular web channels. By subscribing to the channels that interest you, you can easily find and view reports on topics of particular importance to you.

Check with your e.Reporting Server administrator to find out if you can access reports on the Web.

To navigate through the Report Encyclopedia using a web browser, click hyperlinks contained on the Encyclopedia's web pages, or enter special URLs or directives, in the Address or Location window of your web browser. For more information about directives, see Chapter 4, "ReportCast directives."

For more information about navigating the Report Encyclopedia, see Chapter 2, "Working with the desktop Navigator," in *Using e.Reports*, the LRX online help, or the online manual in .PDF format. The illustrations in this chapter use the standard web pages provided by Actuate. Your e.Reporting Server administrator may customize the design of the web pages for your site, so your pages may appear different from those in the illustrations.

### Accessing the Report Encyclopedia on the Web

To access your Report Encyclopedia, bring up your web browser and enter a URL similar to the following:

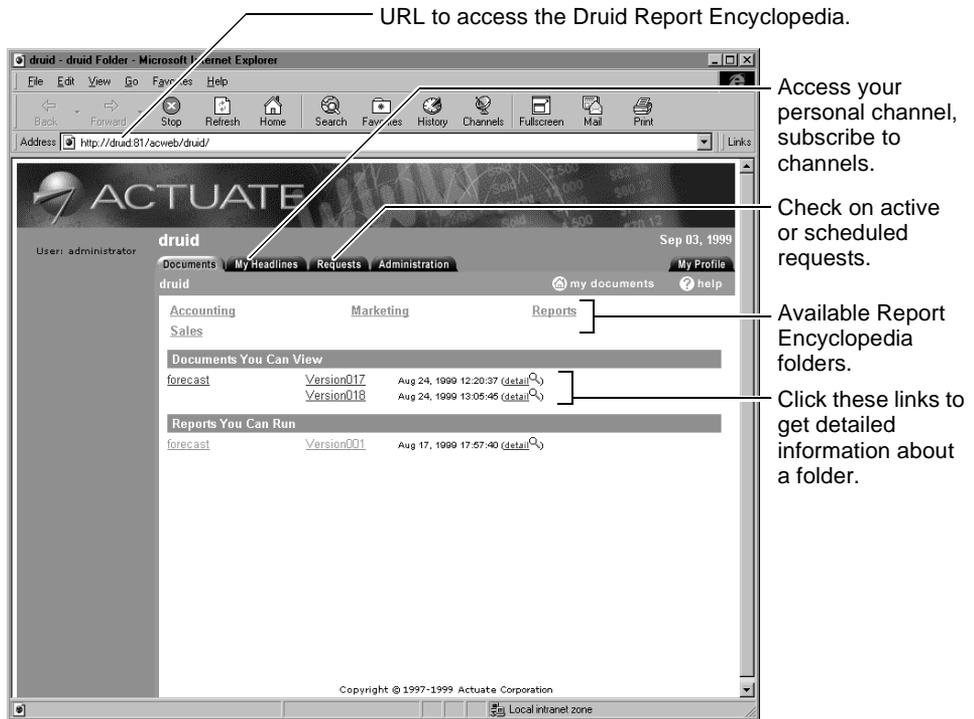
`http://<web server>/acweb/<report server>`

- `<web server>` is the name of the web server running the web interface to the Report Encyclopedia.
- `acweb` is a keyword indicating that you want to access the Report Encyclopedia.
- `<report server>` is the name of the e.Reporting Server with the encyclopedia you want to access.

For example, to access reports on an e.Reporting Server called Druid from a web server called Druid, running on port 81:

`http://druid81/acweb/druid`

Depending upon your setup, a Login dialog box may appear. If it does, enter the user name and password you normally use to access your Report Encyclopedia. A Report Encyclopedia web page displays in the browser.



From this main page, you can:

- View the contents of the Report Encyclopedia.
- Access your personal channel (your completed reports folder).
- Access the available Report Encyclopedia folders.
- Get detailed information about folders.
- Sort the folders by name, type, or creation date and time.
- Request report runs.
- Get status information about requests to run reports.

Requests to run reports or to get status information about requests time out after 120 seconds if ReportCast cannot contact the e.Reporting Server.

To ensure that you always see the most current web reporting page, set your web browser to check the date of web pages every time you issue a request to the e.Reporting Server.

To see changes to the Report Encyclopedia that do not involve adding or deleting an item or folder (for example, updating an existing item), force-reload the web reporting page. To force-reload a web reporting page in Netscape, hold the shift-key down at the same time as clicking the reload button. In Microsoft Internet Explorer, clear the cache and then reload the page.

## Accessing channels and reports

When you click the My Headlines tab on the main Report Encyclopedia web page, a page such as the following displays in the browser.

Subscribe to other ReportCast channels. The Administration tab displays only if you have administrator privileges.

Click these links to access the channels you are subscribed to.

These are the completed reports you can view.

Click the Details icons to examine a report's status, view the report, or delete the status information.

From My Headlines page, you can:

- Click the Personalize button to subscribe to and unsubscribe from ReportCast channels.
- Click the channel icons to access particular channels to which you are subscribed.

- View completed reports.
- Click the Details icon to view a report's status, view the report, or delete status information for the report.

---

## Accessing, generating, and searching reports on the Web

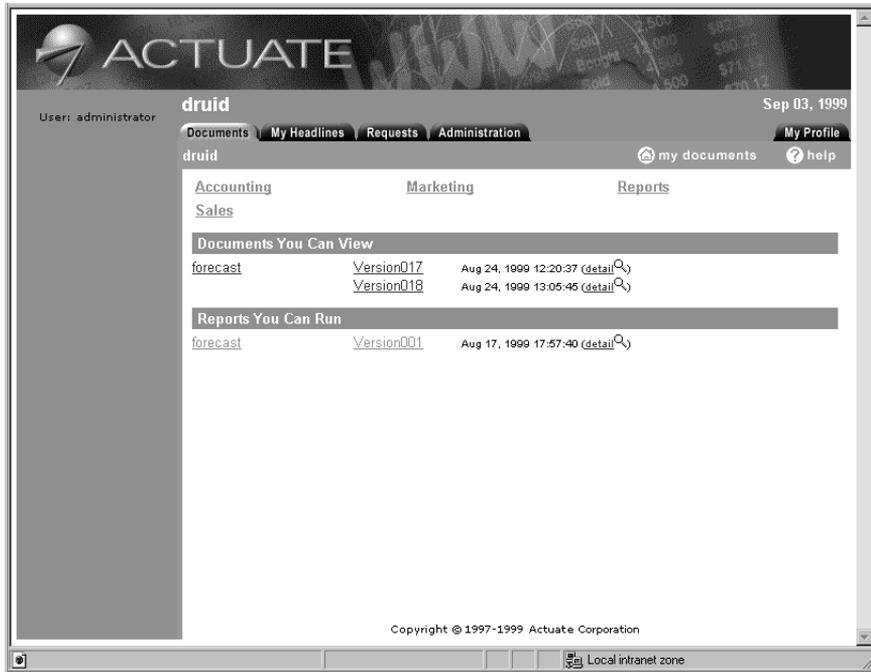
To access and generate reports on the Web:

- Click report executable links in a Report Encyclopedia folder to generate a report.
- Click report links on the My Headlines web page or from other ReportCast channels' pages to view a report.
- Navigate through the Report Encyclopedia folders from the main Report Encyclopedia web page.

### Generating a report

When the main web page for the Report Encyclopedia displays, the same list of folders that you normally see in the Administrator Desktop's Navigator window displays. Click a folder's link to display its contents.

Folder contents pages are similar to the following image.



The folder web page lists the folder's contents, including any subfolders, as well as links to your personal folder and to the Active and Scheduled requests web pages.

The difference between this page and your personal channel web page is that the folder contents page lists report executables. To generate a report, click the report executable's link. A requester page similar to the following displays in the browser.

**ACTUATE**

druid Aug 30, 1999

User: administrator Documents My Headlines Requests Administration My Profile

**Submit Request** New Request ? help

---

**Parameter Groups**

**Output Parameters**

Headline  (String)

**Request**

Wait for results  (when running now)

---

**Advanced**

**Schedule**

Right Now

Once  (mm/dd/yyyy) at  (hh:mm:ss)

Recurring  at  (hh:mm:ss)

---

**Priority**

High (600)  Medium (500)  Low (200)  Other (1-1000)

---

**Version**

Create new version  Keep only the latest  version(s)

Overwrite existing version

Version Name:

Output Name:

---

**Archive Policy**

Use the archive policy for the distribution folder(s)

Delete objects older than  days  hours

Delete objects on  (mm/dd/yyyy) at  (hh:mm:ss)

Archive items before deletion (only applies if not using archive policy for distribution folder)

---

**Notify Channels**

News

Copyright © 1997-1999 Actuate Corporation

**Submit Request**

Fill out the requester fields as you would in the Administrator Desktop, then click the Submit Request button to submit the request. A confirmation form displays in the browser, similar to the following.



Click the Request Status button to check on your request's status. To cancel the request, click the Cancel Request button.

## Setting report viewing options

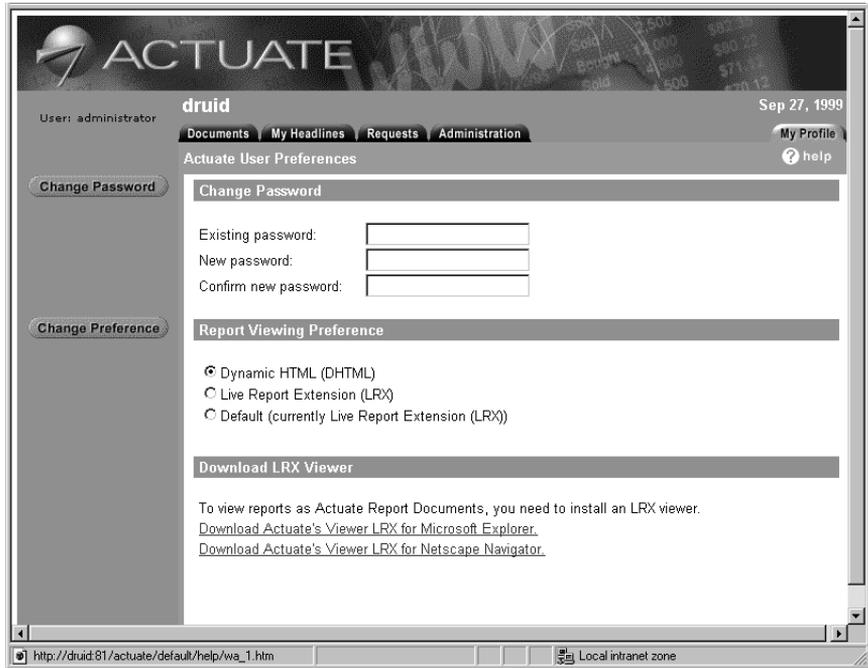
You can tell ReportCast how you want to view reports. Use the User Preferences page to set your report viewing options. Either click the Report Viewing Options link in the ReportCast page banner, or enter a URL similar to the following:

`http://<web server>/acweb/<e.reporting server>/__requestUserPreferences`

For example, if your web server's name is Druid, and your e.Reporting Server also resides on Druid (running on port 81):

`http://druid:81/acweb/druid/__requestUserPreferences`

A page similar to the following image appears.



Choose whether to view reports in Dynamic HTML (DHTML) format, or as Actuate reports using the Actuate Live Report Extension (LRX). The default viewing format is DHTML. Your system administrator may have set an e.Reporting Server-wide default, as well.

You can also download the Actuate Viewer LRX or change your password from the User Preferences page.

For more information about the User Preferences page, see *Using e.Reports*.

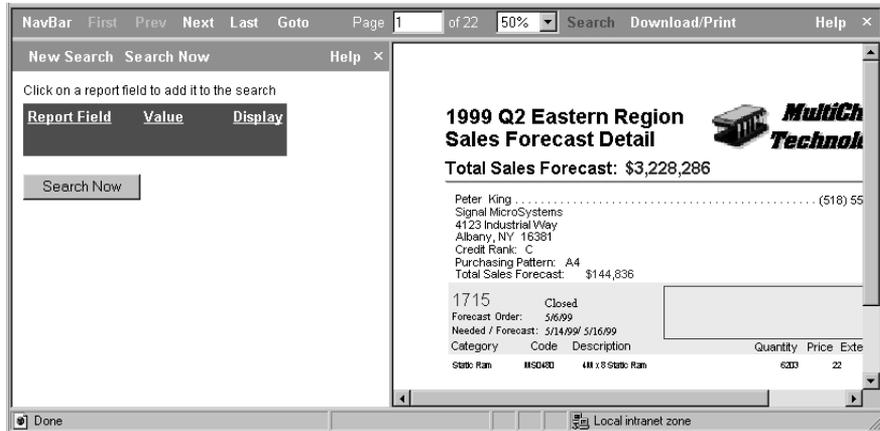
## Searching DHTML reports

You can search for information while viewing DHTML reports using the Search Request page. To perform a search:

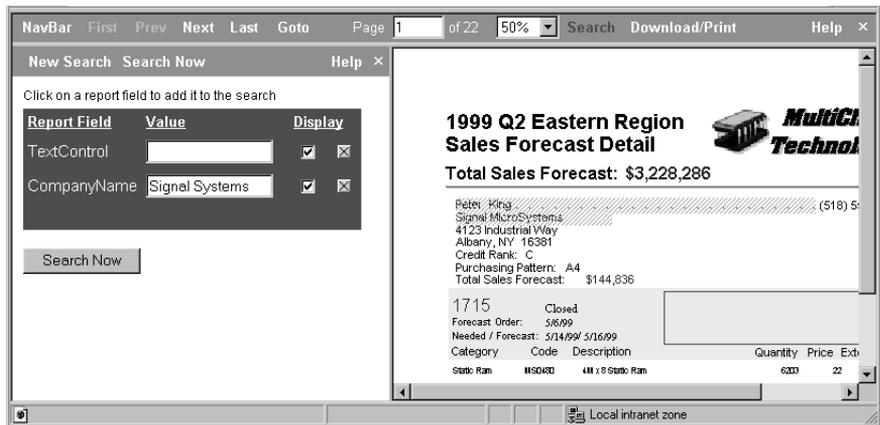
- 1 From the DHTML report viewing page, choose the Search link.



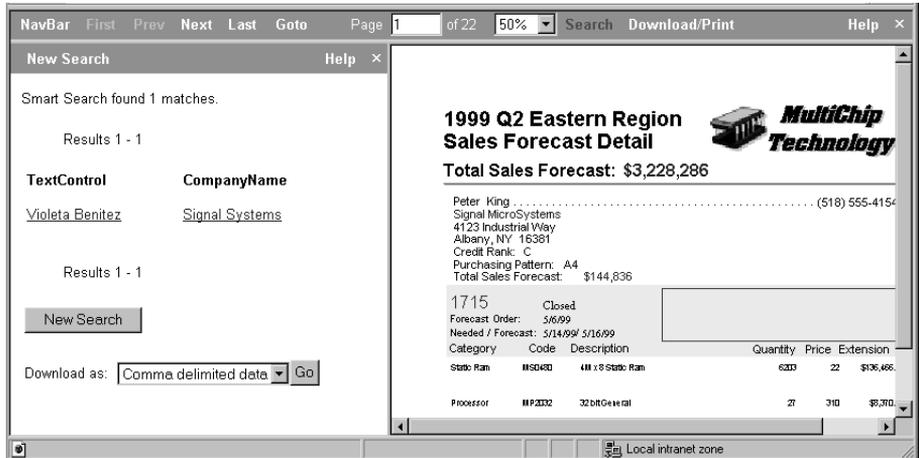
The Search Request page appears.



- 2 Select fields in the DHTML report to add them to your search criteria. The fields appear in the Search List.
- 3 Type the expression or value to search for. The Search Request page looks similar to the following image.



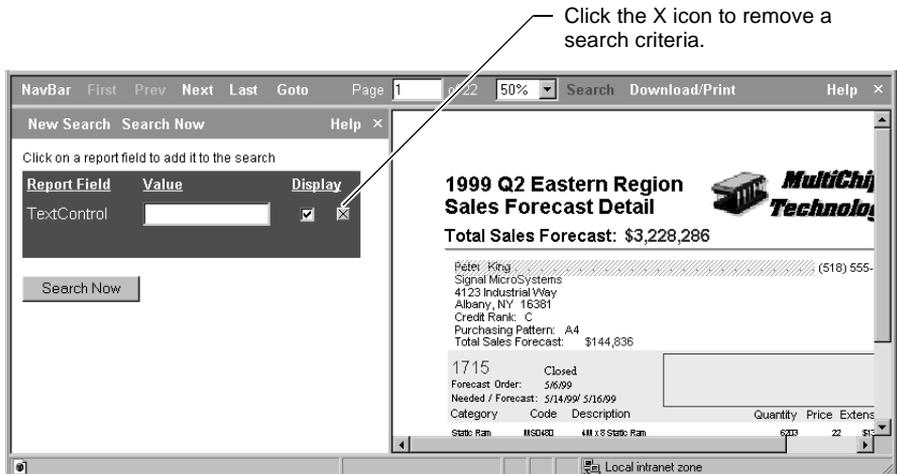
- 4 Click the Search Now button. The Report query results page appears in the browser.



5 Click the search result links to examine specific search results.

To remove search criteria:

- Click a selected field to de-select it, or
- Click the X icon next to the selected field.



For more information about searching DHTML reports on the Web, see *Using e.Reports*.

## Navigating through the Report Encyclopedia

You can navigate through the Report Encyclopedia in a way similar to navigating through the Administrator Desktop. From the main Report Encyclopedia web page, click on a folder's link. The folder contents page displays in the browser. You can continue clicking subfolder links to display their contents, click a report executable to generate a report, or click report .ROIs, .ROVs, or .ROWS to view reports.

---

## Working with ReportCast channels

ReportCast channels provide easy access to particular reports of interest on the Web. When you subscribe to a channel, its link appears in your personal folder. Click the link to check the channel's contents and view reports as you wish.

### Subscribing to a ReportCast channel

To subscribe to a ReportCast channel:

- 1 Choose the My Headlines tab. Your personal folder page appears.
- 2 Click the Personalize button. The channel subscription page appears.  
The subscription page contains a list of available channels, with checkboxes to subscribe to them. Select the checkboxes for the channels to which you wish to subscribe. Click the Details icon to find out more about the channel.
- 3 Click the Apply button. A confirmation page similar to the subscription page displays in your browser.
- 4 Return to your personal channel. Reload the page. The channels you just subscribed to appear on your personal channel web page.

### Unsubscribing from a ReportCast channel

To unsubscribe from a ReportCast channel:

- 1 Choose the My Headlines tab. Your personal folder page appears.
- 2 Click the Personalize button. The channel subscription page appears.  
The subscription page contains a list of available channels, with checkboxes. A checked checkbox indicates that you are subscribed to a particular channel. Select the checkboxes for the channels you wish to unsubscribe from. The checkbox is unchecked.

- 3** Click the Apply button. A confirmation page similar to the subscription page displays in your browser.
- 4** Return to your personal channel. Reload the page. The channels you just unsubscribed from no longer appear on your personal channel web page.



# Administering ReportCast

This chapter contains the following topics:

- About ReportCast administration
- Starting and stopping ReportCast
- Setting ReportCast default e.Reporting Server
- Using non-default web server ports
- Configuring ReportCast port numbers
- Creating custom configurations
- Mapping custom configurations to locales
- Accessing UNIX ReportCast with CGI scripts
- Working with reports on Windows web sites
- Using CGI scripts for security
- Administering ReportCast channels
- URLs in e-mail notification
- Tuning ReportCast performance

---

## About ReportCast administration

Administering ReportCast requires:

- Web server and report server administrator privileges
- Familiarity with web server and report server administration tasks

ReportCast administration tasks include:

- Starting and stopping ReportCast
- Setting the default e.Reporting Server
- Configuring communication port numbers
- Creating custom configurations
- Mapping configurations to system-defined language locales
- Accessing ReportCast using CGI scripts
- Working with reports on Windows sites
- Using CGI scripts for ReportCast security
- Administering ReportCast channels
- URLs in e-mail notification
- Tuning performance

---

## Starting and stopping ReportCast

ReportCast is a service that runs as part of your web server:

- There is no user interface to ReportCast.
- There is no obvious result of installation.

### Starting and stopping ReportCast on an NT system

A ReportCast running on an NT system operates as part of the web server. The ReportCast starts and stops automatically when you start and stop the web server.

## Starting and stopping ReportCast on a UNIX system

UNIX ReportCast has two components:

- NSAPI extension or CGI component
- ReportCast service

The NSAPI or CGI components start automatically when the web server starts, or when a user navigates to a web reporting site URL. These components stop automatically when the web server stops.

The ReportCast service starts and stops separately.

### Starting a UNIX ReportCast service

The ReportCast installation adds the ReportCast service startup script to the `actuate/bin` directory in the `netscape/plugins` directory for the web server that runs ReportCast. To start the ReportCast service, run the following script:

```
<web server directory>/startreportcast.sh
```

- `<web server directory>` is the `netscape/plugins/actuate/bin` directory path for the web server that runs ReportCast.
- `startreportcast.sh` is the ReportCast service startup script.

### Stopping a UNIX ReportCast Server

To stop the UNIX ReportCast Server, run the following script:

```
<web server directory>/stopreportcast.sh
```

- `<web server directory>` is the `netscape/plugins/actuate/bin` directory path for the web server that runs ReportCast.
- `stopreportcast.sh` is the ReportCast Server shutdown script.

### Installation process for Sun Solaris and IBM-AIX

The ReportCast installation process for the Sun Solaris and IBM-AIX ReportCast service adds the service startup script to the Netscape web server startup script. Sun Solaris and IBM-AIX ReportCast services start when the web server starts.

---

## Setting ReportCast default e.Reporting Server

You can define a default e.Reporting Server for ReportCast. This allows you to make use of the `__` default e.Reporting Server specification in ReportCast directives. You use the registry key on NT or environment variable on UNIX `AC_WEBAGENT_DEFAULT_REPORT_SERVER` to set a default e.Reporting Server for ReportCast.

### How to set the default e.Reporting Server on NT

Use Regedit to create a new registry key on an NT system. Before you begin, it is essential to back up your system.

- 1 From the Start menu, run Regedit.

The Registry Editor window displays.

- 2 Choose `HKEY_LOCAL_MACHINE\Software\Actuate\Actuate Web Agent\4.1`.

- 3 Select Actuate Web Agent.

- 4 Select Edit.

The Edit menu appears.

- 5 Choose `New→String Value`.

A new value item displays in the right-hand side of the registry.

- 6 Type: `AC_WEBAGENT_DEFAULT_REPORT_SERVER` for the name of the new item.

- 7 Set the default e.Reporting Server:

- 1 Select `AC_WEBAGENT_DEFAULT_REPORT_SERVER`.

- 2 Right-click on `AC_WEBAGENT_DEFAULT_REPORT_SERVER`.

- 3 Choose Modify.

The Edit String dialog box appears.

- 4 Type the value of the string in the Value data field.

- 5 Choose OK.

For more information about specifying e.Reporting Servers in ReportCast directives, see “Formatting directives” in Chapter 4, “ReportCast directives.”

### How to set the default e.Reporting Server on UNIX

- 1 Use a text editor to edit the `startreportcast.sh` script.

- 2 In the script, above the line that calls `webagentsrvr`, add lines with the following syntax:

```
export AC_WEBAGENT_DEFAULT_REPORT_SERVER
AC_WEBAGENT_DEFAULT_REPORT_SERVER=<ReportServerName>
```

<ReportServerName> is the name of the default e.Reporting Server.

For example, to set ReportCast's default e.Reporting Server to an e.Reporting Server called Corporate, the lines you add read as follows:

```
export AC_WEBAGENT_DEFAULT_REPORT_SERVER
AC_WEBAGENT_DEFAULT_REPORT_SERVER=Corporate
```

For more information about specifying e.Reporting Servers in ReportCast directives, see "Formatting directives" in Chapter 4, "ReportCast directives."

---

## Using non-default web server ports

The default port for most web servers is 80, or 8080. If your web server uses a non-default port, you must either specify the port as part of the URL, or set up an alias to redirect web browsers to another URL. The ReportCast installation process automatically installs ReportCast to communicate with the web server's port. However, when you set up links to your reporting web site, or define base URLs for accessing files on your reporting web site, specify the web server port.

For example:

```
http://mustique:81/acweb/Sales
```

indicates that the web server runs on a machine called Mustique, using port 81.

See your web server documentation for information about setting up web server aliases and redirecting web browsers.

---

## Configuring ReportCast port numbers

You can specify a range of port numbers for ReportCast to use to connect to the e.Reporting Server. You might need to configure the port numbers if your web server uses a specific port number, such as a firewall.

On Windows NT, you configure the start of the range of port numbers by setting a registry string value with the name `AC_SERVER_SOCKET_BASE`. The range of port numbers is set using a string value with the name `AC_SERVER_SOCKET_COUNT`. The string value names are added to the key

HKEY\_LOCAL\_MACHINE\Software\Actuate\Actuate Report Server\4.1.  
The values must be positive integers and the maximum value for both is 32767.

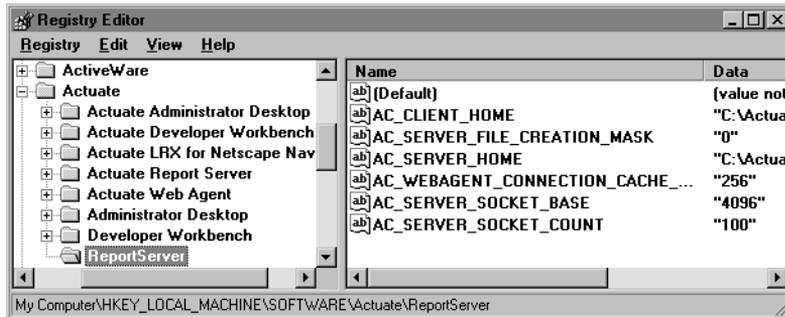
On UNIX, you set the AC\_SERVER\_SOCKET\_BASE and AC\_SERVER\_SOCKET\_COUNT environment variables.

If the start of the range of port numbers is not set, or the start of the range of port numbers is less than 1, the default value is 4096. If the count is not set, the count default value is 8192. If the count is set to less than 1, the count value defaults to 1.

## How to configure ReportCast port range on NT

- 1 If necessary, stop the e.Reporting Server and web server processes. (ReportCast is part of the web server processes.)
- 2 Start the registry editor REGEDIT.EXE.
- 3 Navigate to HKEY\_LOCAL\_MACHINE\Software\Actuate\Actuate Report Server\4.1.
- 4 Create the string value name AC\_SERVER\_SOCKET\_BASE and the value of first port number.
- 5 Create the string value name AC\_SERVER\_SOCKET\_COUNT and the value of the range of port numbers.

This example shows the base port number set to 4096 and the range of port numbers set to 100.



- 6 Close the registry editor.
- 7 Restart the e.Reporting Server and web server processes.

For more information about how to set registry keys, see "How to set a new locale registry key on an NT system," later in this chapter.

## How to configure ReportCast port range on UNIX

- 1 If necessary, stop the e.Reporting Server and web server processes. ReportCast is part of the web server processes.
- 2 Add lines to set the environment variables to e.Reporting Server request server startup script `$AC_SERVER_HOME/bin/reqsrvr.sh`.

Add the new line before the first line that uses the export command.

For example, add these lines to change the `AC_SERVER_SOCKET` base port number to 4096 and the range to 100:

```
AC_SERVER_SOCKET_BASE=4096
export AC_SERVER_SOCKET_BASE
AC_SERVER_SOCKET_COUNT=100
export AC_SERVER_SOCKET_COUNT
```

- 3 Start the e.Reporting Server.

For more information about how to set environment variables, see “How to set locale environment variable on a UNIX system,” later in this chapter.

---

## Creating custom configurations

ReportCast comes with standard configurations such as `enu` (U.S. English) and `jpn` (Japanese). The default configuration is called `default`. To work with custom configurations, you must add `<config>` to your URL. If you do not add `<config>`, the ReportCast uses `default` as the configuration.

The URL has the following syntax:

```
http://<webserver>/acweb/<config>/<reportserver>
```

- `<webserver>` is your web server name.
- `acweb` is a literal string indicating a directive for ReportCast.
- `<config>` is the name of the configuration to use.
- `<reportserver>` is the name of the e.Reporting Server to access.

### How to create a custom configuration

You can create custom configurations, for example, for a particular division or department:

- 1 Go to the web server’s `Actuate` directory.
- 2 Add a new directory named for the custom configuration.
- 3 In the custom configuration directory, add a directory named `Standard`.

- 4 Copy the standard ReportCast templates to the custom configuration's Standard directory.

### **How to set up directories for the custom templates in a custom configuration**

- 1 In the custom configuration directory, create a directory called Custom.
- 2 Copy the templates you want to customize into the Custom directory.
- 3 Customize the templates as appropriate.

For more information about customizing templates, see Chapter 5, "ReportCast scripting language reference."

---

## **Mapping custom configurations to locales**

ReportCast maps the name of its standard configurations to localization settings, also known as locales, to determine how to print numbers and dates and how to sort strings. The differences are significant to users in those language groups. For example, the French, English, and Japanese locales display date and time values differently.

The names of configurations and locales are usually the same.

Locales are standard operating system features. They have standard three-letter names, such as enu, jpn, and frn, for U.S. English, Japanese, and French. When you create a custom configuration that does not have the same name as its locale, you map the custom configuration to the appropriate locale to enable the appropriate functionality. On an NT system, you create a new registry key. On a UNIX system, you set an environment variable.

Folder items and other lists sorted by string values display in order according to the locale of the machine running ReportCast, not necessarily according to the locale specified by the configuration in the URL.

### **How to set a new locale registry key on an NT system**

Use Regedit to create a new registry key on an NT system. Before you begin, it is essential to back up your system.

- 1 From the Start menu, run Regedit.  
The Registry Editor window displays.
- 2 Choose HKEY\_LOCAL\_MACHINE\Software\Actuate\Actuate Web Agent\4.1.
- 3 Select Actuate Web Agent.

4 Choose Edit.

The Edit menu appears.

5 Choose New→String Value.

A new value item displays in the right-hand side of the registry.

6 Type: AC\_WEBAGENT\_<config>\_LOCALE for the name of the new item.

<config>, the name of the custom configuration, must be typed in uppercase letters.

7 Set the location:

1 Select AC\_WEBAGENT\_<config>\_LOCALE.

2 Right-click on AC\_WEBAGENT\_<config>\_LOCALE.

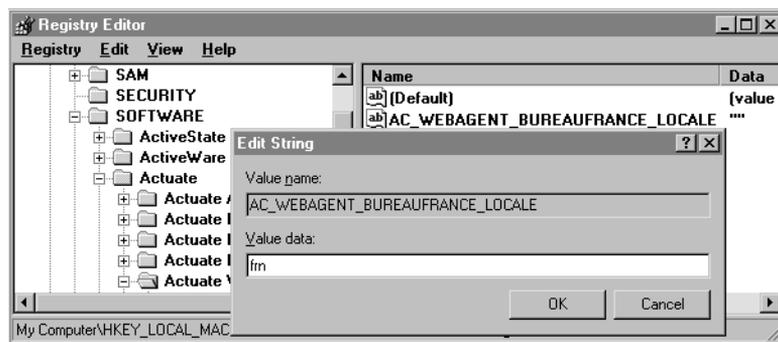
3 Choose Modify.

The Edit String dialog box appears.

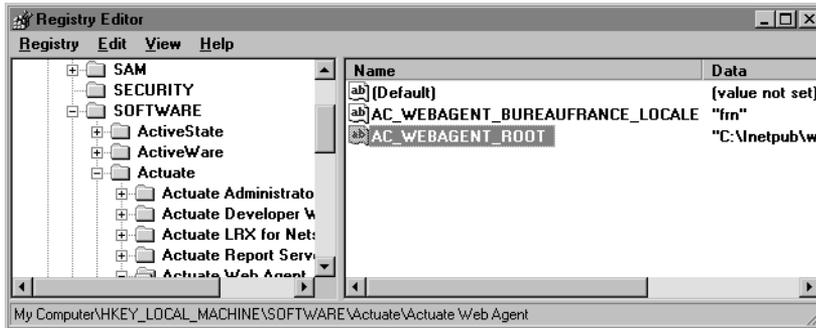
4 Type the value of the string in the Value data field.

5 Choose OK.

For example, if you have a custom configuration for your company's French office called BUREAUFRANCE, and you want the configuration to use the frn locale information, set AC\_WEBAGENT\_BUREAUFRANCE\_LOCALE to frn. The following illustration shows you how to modify the registry key.



The following illustration shows the modified registry key.



## How to set locale environment variable on a UNIX system

- 1 Use a text editor to edit the startwebagent.sh script.
- 2 In the script, above the line that calls webagentsrvr, add lines with the following syntax:

```
export AC_WEBAGENT_<config>_LOCALE
AC_WEBAGENT_<config>_LOCALE=<LocaleName>
```

<config> is the name of the custom configuration.  
 <LocaleName> is the name of the locale.

For example, to set the BUREAUFRANCE locale on your UNIX system, the lines you add read as follows:

```
export AC_WEBAGENT_BUREAUFRANCE_LOCALE
AC_WEBAGENT_BUREAUFRANCE_LOCALE=frn
```

## Supporting locales with e.Reporting Server and ReportCast

The Actuate e.Reporting Server uses a localemap file to support different locales when converting reports in DHTML, XML, or PDF formats. For more information about localemap file, see *Actuate e.Reporting Server Guide*.

ReportCast also uses a localemap file to support different locales. The localemap file is located in the web server's document root directory, for example, <webroot>\docs\actuate\etc.

Any modifications made to the localemap file on an e.Reporting Server system must also be made on any ReportCast system that connects to the e.Reporting Server. The same system locales on the e.Reporting Server system must also be installed on the ReportCast system to allow Actuate locale support to work with Actuate reports in DHTML, XML, and PDF formats.

---

## Accessing UNIX ReportCast with CGI scripts

If your UNIX system uses CGI scripts to access ReportCast, consider the following issues:

- If the ReportCast service is not running and a user attempts to connect to ReportCast, the CGI implementation returns an error. The NSAPI implementation returns a blank page.
- If your web server automatically generates response headers, such as cookies, consider using a parsed-header CGI script to access ReportCast.
- CGI scripts do not automatically resolve escape sequences into the special characters they represent. Your CGI script must include code that translates escape sequences. An escape sequence begins with the percent character (%), followed by the ASCII two-digit hexadecimal code for the special character. For example, add code to translate the following URL:

`http://RS%22Main/acweb/SalesRpts`

to:

`http://RS_Main/acweb/SalesRpts`

CGI scripts are supported only with UNIX ReportCast. There is no CGI counterpart supported for NT.

### How to use a parsed-header CGI script

- 1 Export the environment variable:  
`HTTP_AC_WEBAGENT_PARSED_HEADERS`
- 2 In the CGI script, set the environment variable to True.

The ReportCast installation process generates two CGI scripts:

- `nph-actuate.cgi`
- `ph-actuate.cgi`

The script `ph-actuate.cgi` is a parsed-header script. A URL that contains this script name generates output to which the web server can add its own headers.

If you use a parsed-header CGI script to access ReportCast, your URL must reference `ph-actuate.cgi` to produce parsed-header output.

---

## Working with reports on Windows web sites

To work with reports on a Windows-based web reporting site, register a browser plug-in or helper application for the MIME types of Report Encyclopedia items. You can register plug-ins and helper applications in your browser. For example, in the Netscape browser, register a helper application from Options→General Preferences→Helpers.

If no browser plug-in or helper application is registered for a requested Report Encyclopedia item, the web browser prompts you to save the item to the local file system. The default file name provided in the Save dialog box includes a version number, for example, REPORT.ROI;1. Windows systems use the filename extension to recognize the file type. To ensure that Windows can recognize the file type, delete the version part of the filename, including the semicolon. In this example, you save the file as REPORT.ROI because Windows does not accept the ;1 convention. You can also modify the name to something like REPORT1.ROI to add versioning information.

If the name of the Report Encyclopedia item contains spaces or other special characters, the default file name contains character sequences beginning with % instead of the actual characters.

---

## Using CGI scripts for security

You may need to mask the location on the report server of reports that a user views on the web reporting site.

For example, a custom HTML page can display a link to a report at:

```
http://webserver/cgi/secure.cgi/myreport
```

instead of the actual location. For example:

```
http://webserver/cgi/secure.cgi/acweb/reportserver/accounts/trusted/  
myreport.roi?View
```

The CGI script masks the report URL on the web reporting site. To mask the report URLs and use the CGI script to reconstruct the complete URLs:

- Set the CGI script to change the incoming, disguised values of the PATH\_INFO and QUERY\_STRING CGI variables to construct a valid report-viewing request URL.
- Set the CGI script to set and export the environment variable HTTP\_AC\_WEBAGENT\_BASE\_URL to the unmasked URL.

The Base URL is that part of the URL up to and including the name of the machine running the report server through which the report is normally accessed on the web reporting site.

In this example:

- The incoming value of PATH\_INFO is /myreport. The CGI script resets it to /acweb/reportserver/accounts/trusted/myreport.roi.
- The incoming value of QUERY\_STRING is empty. The CGI script resets it to View.
- The Base URL value is /cgi/secure.cgi/acweb/reportserver, or, alternatively, http://webserver/cgi/secure.cgi/acweb/reportserver.

---

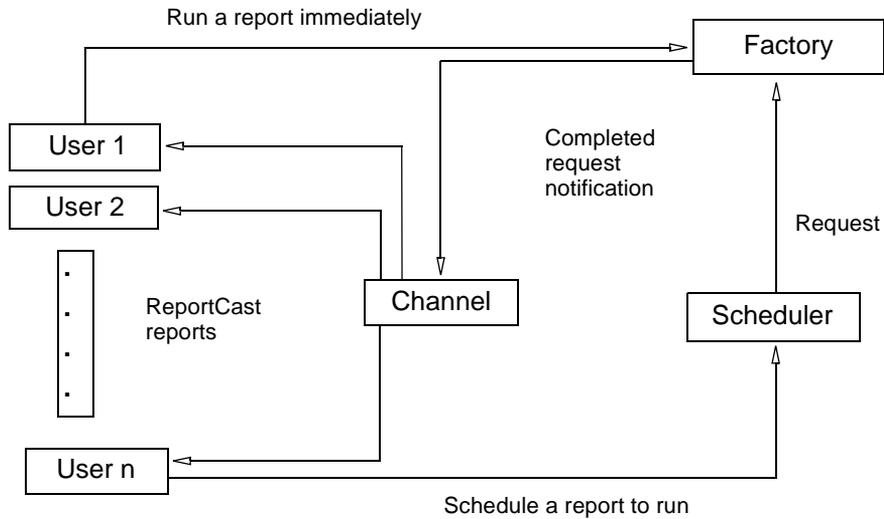
## Administering ReportCast channels

### About Actuate ReportCast channels

Actuate ReportCast channels is a service that allows users to subscribe to particular web channels and view reports available on those channels. ReportCast channels uses push technology to deliver reports to particular web channels. Push technology is also known as web casting, push distribution, or publish and subscribe. When new reports become available, the appropriate channel receives a notification. Subscribed users check the channel's web page to get a listing of available reports. When users click on a report link, the report appears in the web browser window.

ReportCast channels allows you to subscribe to and unsubscribe from particular channels, list the channels to which you are currently subscribed, and view the contents of those channels. If you have administrator privileges, you can also create, delete, and modify channels.

The following illustration shows how ReportCast channels fit into the Actuate Internet architecture.



Users make requests to run reports immediately, or to run reports at a scheduled time. The requester form allows users to select the users and channels to notify when the report is complete. Once the report runs successfully, the completed request notification goes to the appropriate users and channels.

For more information about factory processing, see Chapter 8, “The report generation process,” in *Programming e.Reports*. For more information about scheduling reports, see Chapter 4, “Running a report from the desktop,” in *Using e.Reports*.

Users with administrative privileges on e.Reporting Server can administer ReportCast channels. Choose the Administration tab on the Report Encyclopedia web page, or use the `__channels?admin` directive to access the channel administration page. The complete syntax for the directive is:

`http://<webserver>/acweb/<reportserver>/__channels?admin`

The following table lists the ReportCast channels administrative directives. All commands are available from the channels administration page.

Command	Description
<code>__channels?admin</code>	Accesses the ReportCast channels administration web page.
<code>__completed</code>	Lists the channels to which this user subscribes.
<code>__channels/&lt;channel&gt;</code>	Lists the contents of channel <code>&lt;channel&gt;</code> .

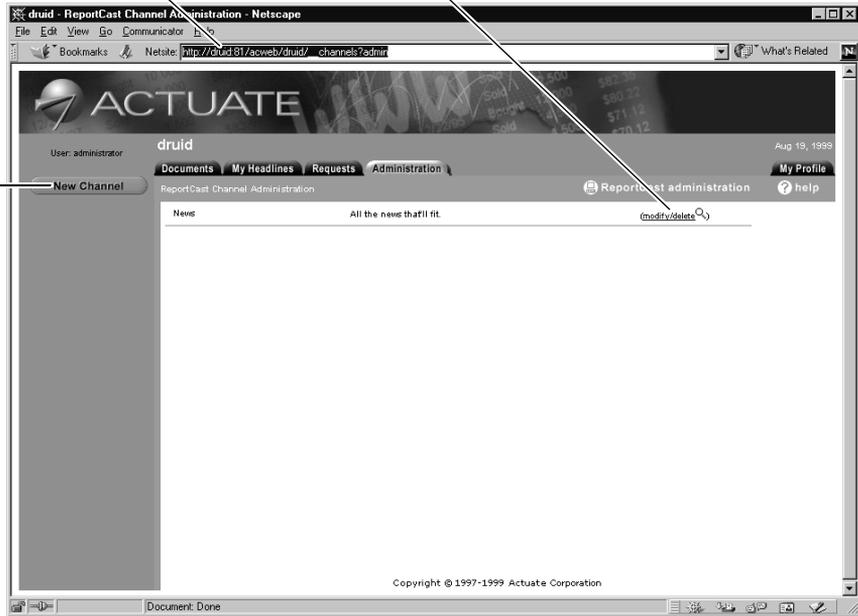
<b>Command</b>	<b>Description</b>
__channels?retrieveForm	Retrieves and displays the channels subscription form.
__channels?submitForm	Submits the subscription form to ReportCast.
__channels/<channel>?drop	Removes the specified channel from the e.Reporting Server.
__channels/ <channel>?retrieveDetails	Displays the details form for the specified channel. Users with administrator privileges can modify the channel using this form.
__channels/ <channel>?submitDetails	Submits the form that modifies details of a channel.
__channels?retrieveCreation	Displays the form that allows users with administrator privileges to create a new channel.
__channels?submitCreation	Submits the form that creates a new channel.

The following illustration shows the channels administration page.

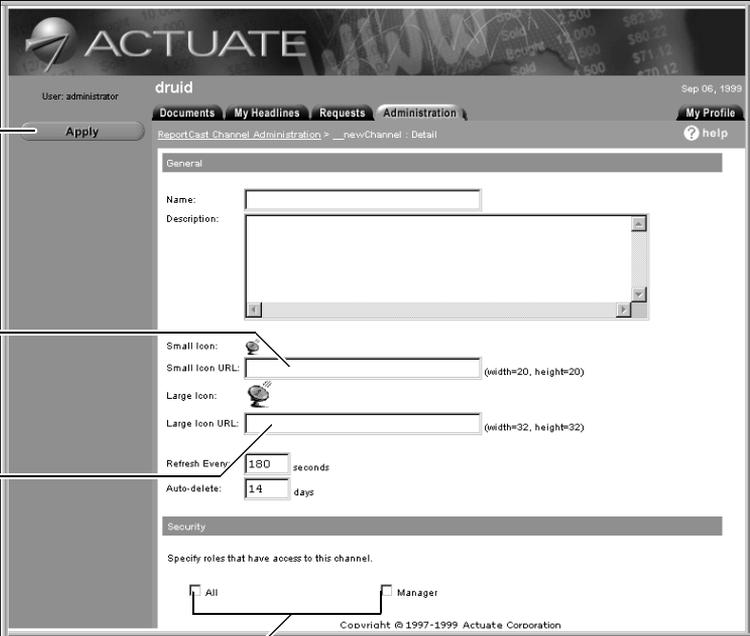
Directive that generates the channels admin page.

Choosing the New Channel button generates the retrieveCreation directive.

Click Modify/Delete to make changes to or to remove the channel.



When you click the New Channel button, the Channel Administration page appears.



The screenshot shows the ACTUATE Channel Administration page. The page has a header with the ACTUATE logo, user information (User: administrator), and navigation tabs (Documents, My Headlines, Requests, Administration). The main content area is titled "ReportCast Channel Administration > \_newChannel - Detail". It contains a form with several sections: "General" with fields for Name and Description; "Small Icon" with a satellite dish icon and a text input for Small Icon URL (width=20, height=20); "Large Icon" with a satellite dish icon and a text input for Large Icon URL (width=32, height=32); "Refresh Every" with a text input for 180 seconds; "Auto-delete" with a text input for 14 days; and "Security" with a section titled "Specify roles that have access to this channel." containing checkboxes for "All" and "Manager".

Click Apply to create the new channel.

Enter the URL for the scaled-down image to use for this channel's icon.

Enter the URL for the large image to use for this channel's icon.

Select a Security checkbox to set the privileges needed to access the new channel.

## How to create a new channel

To create a new channel from the channels administration page:

- 1 Choose New Channel.  
The new channel creation page displays in the browser.
- 2 Fill out the channel creation form:
  - 1 Name is the name of the channel as it appears in channel subscription lists.
  - 2 Description is text describing the channel's purpose, audience, or other pertinent information.
  - 3 Small Icon URL is the URL to an image, for example, .GIF, or .JPG, to be used as the small-scale icon for this channel. The default is the satellite dish icon.

- 4 Large Icon URL is the URL to an image, for example, .GIF, or .JPG, to be used as the large icon for this channel. The default is the satellite dish icon.
  - 5 Refresh Every determines how often the channel's contents should be updated.
  - 6 Auto-delete determines how long an item may remain in the channel before it is automatically deleted.
  - 7 Use the security section to determine which roles may access the channel.
- 3 Click Apply to create the channel.
- A confirmation page displays in the browser, showing the new channel's name and details.

### **How to modify an existing channel**

To modify an existing channel from the channel's administration page:

- 1 Click the Modify/Delete icon for the channel you wish to modify.  
The channel's Details page displays in the browser.
- 2 Edit the channel's information on the Details page.
- 3 Click Apply.  
The updated channel's Details page displays in the browser.

### **How to delete an existing channel**

To delete an existing channel from the channel's administration page:

- 1 Click the Modify/Delete icon for the channel you wish to modify.  
The channel's Details page displays in the browser.
- 2 Click Delete Channel.  
The Successful command completion page displays in the browser.

---

## **URLs in e-mail notification**

Report Encyclopedia administrators can configure the e.Reporting Server to send an e-mail notification when a request is complete. Using ROTP syntax to specify the report's location is the default. ROTP syntax is a report object protocol that runs a report on a server. For example, an e-mail sent to users that contains the line:

```
rotp://bimini/Sales/goals/quarterly.roi;7
```

specifies the seventh version of the report QUARTERLY.ROI is in the folder Sales\Goals in the Report Encyclopedia on Bimini.

As administrator, you can configure the e.Reporting Server to specify a URL for the location of the report when notifying users by e-mail. For example, if users can access the Report Encyclopedia using ReportCast on the web server Paradise, you can configure the e.Reporting Server to specify:

```
http://paradise/acweb/bimini/Sales/goals/quarterly.roi;7?view
```

For more information about specifying URLs, see Chapter 6, “Administering the Report Encyclopedia in the web environment,” in *Administering the Report Encyclopedia*.

---

## Tuning ReportCast performance

This section describes some considerations and options for tuning ReportCast performance.

### Setting user connection timeouts

Each time a user logs on to the web reporting site, ReportCast creates a connection between itself and the e.Reporting Server. The connection stays open indefinitely unless you specify a timeout period.

Registry keys on NT, or environment variables on UNIX, allow you to set the timeout period:

- `AC_WEBAGENT_CONNECTION_CACHE_SIZE` sets the maximum number of e.Reporting Server connections that ReportCast holds open at any time. Set `AC_WEBAGENT_CONNECTION_CACHE_SIZE` to the number of actual users that will be accessing ReportCast, not necessarily the number of distinct e.Reporting Server users. If the connection cache size is not set, the number of connections is unlimited. Set `AC_WEBAGENT_CONNECTION_CACHE_SIZE` to 0 (zero) to ensure that connections are never cached from one request to another. The default value is 256.
- `AC_WEBAGENT_CONNECTION_CACHE_TIMEOUT` sets the approximate time, in seconds, after which an idle e.Reporting Server connection is dropped. If connection cache timeout is not set, connections remain on the e.Reporting Server indefinitely.

On NT, set the registry keys as string values in `HKEY_LOCAL_MACHINE\Software\Actuate\Actuate Web Agent\4.1`. Use Regedit to set the registry keys. For more information about how to set registry keys, see “How to set a

new locale registry key on an NT system,” and “How to set locale environment variable on a UNIX system,” earlier in this chapter.

## Setting request timeouts

You can set the length of time that ReportCast waits before a request times out. The `AC_WEBAGENT_REQUEST_TIMEOUT` registry key on NT or environment variable on UNIX specifies the request timeout period in seconds. It affects only those report requests for which users checked the Wait checkbox (wait for completion). If `AC_WEBAGENT_REQUEST_TIMEOUT` is set and the Wait checkbox is checked on the requester form, ReportCast waits the specified number of seconds. If the report request has not completed, ReportCast displays the request status page, rather than waiting for the request to complete.

The default is that there is no timeout.

On NT, set the registry keys as string values in `HKEY_LOCAL_MACHINE\Software\Actuate\Actuate Web Agent\4.1`. Use Regedit to set the registry keys. For more information about how to set registry keys, see “How to set a new locale registry key on an NT system,” and “How to set locale environment variable on a UNIX system,” earlier in this chapter.

## Tuning viewing performance

If your users are waiting too long for reports to display over the net, set the `DemandCachePageSize` registry key to tune performance.

`DemandCachePageSize` controls the size of data packets sent to an Actuate Viewer LRX over the network. Specify the data packet size in bytes. Packet size must be a multiple of 2048:

- Minimum packet size is 2048 bytes
- Maximum packet size is 128K bytes
- Default packet size is 64K

Consider the following items when deciding how to set the `DemandCachePageSize` registry key:

- Smaller packet sizes cause more reads to be made to the network. If you have a fast network, larger packet sizes may work well for your users.
- Larger packet sizes cause fewer reads to be made to the network. In general, read time is shorter than connect time.
- You may need to experiment to find the packet size that works best for your users. Variables such as network speed, number of users, and so on affect how much particular packet sizes increase report viewing efficiency for your users.

## How to set the DemandCachePageSize registry key

For more information about using Regedit, see “How to set a new locale registry key on an NT system,” and “How to set locale environment variable on a UNIX system,” earlier in this chapter.

- 1 Run Regedit:
  - For the Netscape browser, navigate to HKEY\_CURRENT\_USER→Software→Actuate→Netscape LRX 4.0→General
  - For the Microsoft Internet Explorer browser, navigate to HKEY\_CURRENT\_USER→Software→Actuate→MSIE LRX 4.0→General
- 2 Select DemandCachePageSize.
- 3 Set the value of DemandCachePageSize.



# Working with ReportCast templates

This chapter contains the following topics:

- About ReportCast templates
- Creating custom templates

---

## About ReportCast templates

As you submit directives, ReportCast generates reporting web pages dynamically from HTML templates. The template files contain scripting language commands embedded in HTML comments. The scripting language commands tell ReportCast how to generate and format its web pages.

You can customize the templates to meet your needs. Only the downloading and viewing directives cannot use custom templates. You can also customize default forms, perhaps modifying a standard form to display only some of the available parameters, add a background color or graphic, or to display your company logo.

To customize ReportCast templates, you should have knowledge of the following:

- Directives you can send to ReportCast. For information about ReportCast directives, see Chapter 4, "ReportCast directives."
- Contents of the template files. For information about template files, see "About HTML template files," later in this chapter.
- ReportCast scripting language. For information about ReportCast scripting language, see Chapter 5, "ReportCast scripting language reference."
- HTML, the HyperText Markup Language. Information about HTML is available from various sites on the World Wide Web and in many books available at local bookstores.

### About HTML template files

All ReportCast template files must have the extension `.ACHTML`, for Actuate template HTML. Template files can be specific to display the contents of a particular folder on a particular report server. Template files can also be general to display e.Reporting Server objects on a particular web server. Whether a template applies to one e.Reporting Server folder or to the entire web server depends on its placement in the e.Reporting Server, the web server, or on Unix systems only, the ReportCast Server directory structure.

Custom HTML requester forms are the exception. These are HTML files that you create to run specific reports. ReportCast does not provide these forms. Custom HTML requester forms must:

- Reside in the same e.Reporting Server folder as the report executable (`.ROX`) file
- Have the file extension `.HTML`
- Depend upon the `.ROX` file

Custom HTML requester forms that reside on the e.Reporting Server in a Report Encyclopedia folder are static. Since custom HTML requester forms reside in the Report Encyclopedia, ReportCast cannot fill in variables such as the e.Reporting Server name and the current date and time.

For more information about the placement of template files, see “About template placement and scope,” later in this chapter.

For more information about creating custom HTML requester forms for specific reports, see “Creating custom requester forms for a specific report,” later in this chapter.

The following table lists and describes the web pages that ReportCast generates.

<b>Generated HTML page</b>	<b>Template file name</b>	<b>Description</b>
Folder Listing	FOLDERLIST.ACHTML	Content of an Encyclopedia folder.
Requester Menu	REQUESTERMENU.ACHTML	Request form that runs reports.
Standard Requester	REQUEST.ACHTML	Standard request form.
Custom Folder	INDEX.ACHTML	Custom folder page created by an administrator and stored directly on an e.Reporting Server.
Request Folder	REQUESTLIST.ACHTML	Content of one or all request folders.
Request Confirmation	CONFIRM.ACHTML	Response page returned after a user submits a request to run a report.
Request Detail	REQUESTDETAIL.ACHTML	Response page used to display detailed information about a request.
Request Status	STATUS.ACHTML	Status page returned after a user asks for request status.
Cancelled Request Confirmation	CANCEL.ACHTML	Response page returned when a request to cancel an active request is successful.
File Detail	FILEDETAIL.ACHTML	Lists file properties.

<b>Generated HTML page</b>	<b>Template file name</b>	<b>Description</b>
Confirmation	SUCCESS.ACHTML	Response page returned when a request to drop or rename a report is successful.
Error	ERROR.ACHTML	Response page generated after an error condition is detected.
Administration	ADMIN.ACHTML	Page generated to perform general, non-e.Reporting Server-related, administrative tasks.
Channel Subscription	CHANNELSUBSCRIPTION.ACHTML	Subscription form to subscribe to and unsubscribe from channels.
Channel Administration	CHANNELADMIN.ACHTML	Top-level channel administration page.
Channel Contents	CHANNELCONTENTS.ACHTML	Page that displays contents of a specified channel.
Channel Detail	CHANNELDETAIL.ACHTML	Administrators use this page to create, delete, and modify a specified channel.
Report Viewing Options	USERPREFERENCES.ACHTML	Choose how to view reports.
View Frameset	VIEWFRAMESET.ACHTML	The initial set of frames that display the DHTML report, including the navigation toolbar, search window, table of contents, and the DHTML report.
Navigation	VIEWNAV.ACHTML	The navigation toolbar used for viewing the DHTML report.
Table of Contents	VIEWTOC.ACHTML	The DHTML report's table of contents.
File Versions	FILEVERSIONS	Display the list of available versions of the current file.

## About internal template structure

The standard ReportCast pages include many elements that several templates use, such as the copyright section and page headers. The ReportCast templates directory contains files with names such as contentactive.achtml, contentbegin.achtml, contentfileversions.achtml. These subtemplate files generate elements common to many ReportCast pages. ReportCast templates include these subtemplate files, using ReportCast scripting language include statements, to provide elements such as the copyright notice, page headers, and folder contents listing. To create custom templates, you customize the individual subtemplate files as well as the standard template files.

## About template placement and scope

A template's placement determines its scope. A template can apply to a specific report, report server folder, or to an entire web server.

### Placement in an e.Reporting Server folder for a specific report

Placing an HTML requester form in an e.Reporting Server folder and creating a dependency from that requester form to a particular .ROX file provides a custom requester form for that specific report executable. If more than one HTML file in the e.Reporting Server folder depends upon a particular .ROX, ReportCast displays a requester menu form. The user chooses the custom requester to use.

### Placement in an e.Reporting Server folder

Placing a template called index.achtml in an e.Reporting Server folder provides a folder-specific template. Such templates apply only to a particular folder's contents on a particular e.Reporting Server.

### Placement in an e.Reporting Server's \_\_WebAgent folder

Creating a \_\_WebAgent folder on an e.Reporting Server, then placing ReportCast templates inside, provides e.Reporting Server-specific templates. Such templates apply only to objects on a particular e.Reporting Server.

### Placement in the Custom directory

Placing ReportCast templates in a directory called Custom in the web server's Actuate directory tree, or in the UNIX ReportCast Server's Actuate directory tree, provides web server configuration-specific templates. Such templates apply only to objects for a particular configuration of a particular web server, for example, the Japanese-language or French-language configuration.

For NT installations and UNIX installations with both ReportCast Agent and ReportCast Server on the same machine, the Custom directory is located

under your web server's document root directory, in the Actuate directory. For UNIX installations with separate ReportCast Agent and ReportCast Server installations, the Custom directory is located in the Actuate distribution's ReportCast directory:

```
wwwroot\actuate\<config>\standard
```

where `<config>` is the name of the configuration. For example, if your NT web server configuration is on `wwwroot`, and you are providing a Japanese-language configuration, the directory path is:

```
wwwroot\actuate\jpn\custom.
```

If your UNIX ReportCast Server is in the Actuate distribution directory `/usr/actuate` and you are providing a Japanese-language configuration, the directory path is:

```
/usr/actuate/ReportCast/templates/jpn/custom
```

If you plan to provide custom templates for a particular configuration, you must create the Custom directory yourself and place the custom templates in it.

### **Placement in the Standard directory**

ReportCast templates located in a directory called Standard in the web server's Actuate directory tree, or in the UNIX ReportCast Server's Actuate directory tree, are the default ReportCast templates supplied by Actuate. For NT installations and UNIX installations with both ReportCast Agent and ReportCast Server on the same machine, if the web server's document root is `wwwroot`, the path to the default templates is:

```
wwwroot\actuate\<config>\standard
```

where `<config>` is the name of the configuration. For UNIX installations with separate ReportCast Agent and ReportCast Server installations, if the Actuate distribution directory is `/usr/actuate`, the path to the default templates is:

```
/usr/actuate/ReportCast/templates/<config>/standard
```

where `<config>` is the name of the configuration.

### **Search order for web templates**

ReportCast searches for template files in a specific order:

- e.Reporting Server folder
- e.Reporting Server's `_ _WebAgent` directory
- The Custom directory for a particular configuration
- The Standard directory for the configuration

For more detailed information about the search order, see “Understanding ReportCast searches for templates,” later in this chapter.

## About the ReportCast object model

To understand how ReportCast’s directives, scripting language commands, and templates work together, it is useful to understand something about the objects with which ReportCast works. ReportCast objects represent the URL requests, web server, e.Reporting Server, and objects on the e.Reporting Server. ReportCast performs actions on e.Reporting Server objects or displays their properties.

### ReportCast objects

ReportCast operates on these types of objects.

Object	Description
Web server	Web server running ReportCast.
ReportCast	ReportCast process.
Web request	Request that ReportCast is currently handling.
e.Reporting Server	Actuate e.Reporting Server on which the requested object is located.
Folder	Folder in a Report Encyclopedia. Folders can contain other folders as well as files.
Folder item	One of the contents of a folder.
Report definition	.ROX, .ROV, .ROI, .ROW file types.
Request folder	A folder containing active, scheduled, or completed requests.
Request	A request to run a report. Requests are located in Request folders.
Member	A user, role, or group.
Parameter group	A group containing parameters for a specific report request.
Parameter	A specific parameter used to create a request.
Schedule	The part of a request’s schedule that describes the times at which the request is to run.
Printer property	The part of the print request that provides information about the printer that will print the requested report.
Channel	An object that contains a set of requests.

<b>Object</b>	<b>Description</b>
Error	An object that contains information about an error encountered by ReportCast.
TOC item	An object that consists of a displayable name and a component ID.
View session	An object that tracks the current page, navigation mode, format, and other viewing properties for the current report.

## Working with the ReportCast object model

The following table shows the relationships among ReportCast objects. The “Provides a list of” column lists the objects with which the objects in the Object column are associated.

<b>Object</b>	<b>Provides a list of...</b>
Web server	Nothing
ReportCast	Nothing
Web request	Nothing
e.Reporting Server	Members, request folders, channels
Folder	Folder items, members (those granted privileges)
Folder item	Depends on the kind of item
Report definition	Parameters, parameter groups, parameter values
Request	Request status
Request folder	Requests
Member	Privileges within the context of a folder item or channel
Parameter	Nothing
Parameter group	Parameter
Parameter value	Nothing
Privilege	Nothing
Channel	Requests
Error	Nothing
View session	
TOC item	
Search criterion	
Search result set	

<b>Object</b>	<b>Provides a list of...</b>
Search result hit	
Search result field	

## Understanding objects available to a template

When you create templates, it is convenient to know when particular objects are available.

These objects are always available:

- e.Reporting Server (if specified in the URL)
- ReportCast
- Web request
- Web server

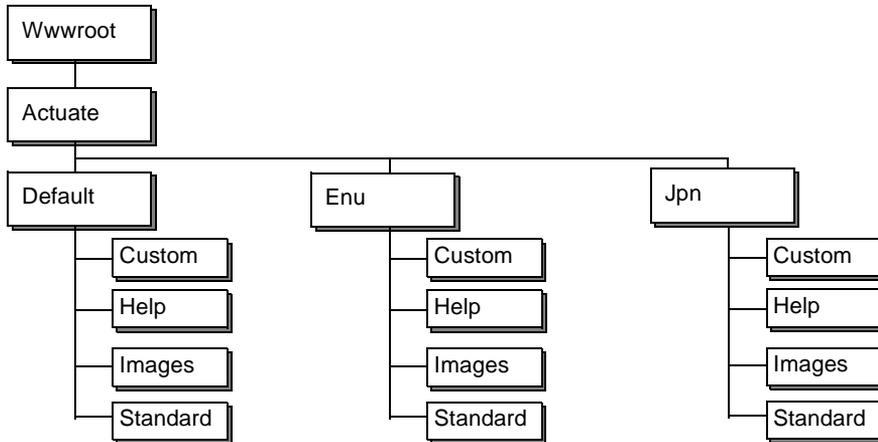
These rules determine when other objects are accessible. If the object does not meet one of the following requirements, it is not accessible:

- If an object is on the web page of the web server, ReportCast, request, or e.Reporting Server, it is available for that page's template.
- List objects are available to list all objects related to a specific object, such as a list of privileges, e.Reporting Servers, and so on. Each iteration through the list loop makes the next object in the list available.
- If an object is accessible, you can make its variables and list statements accessible as well using the With command. The With command has no effect if the object is not already accessible. For more information about the With command, see "With command," in Chapter 5, "ReportCast scripting language reference."

## About the web server template directory structure

When you install ReportCast, the installation process sets up a template directory structure. The structure depends upon the type of ReportCast installation.

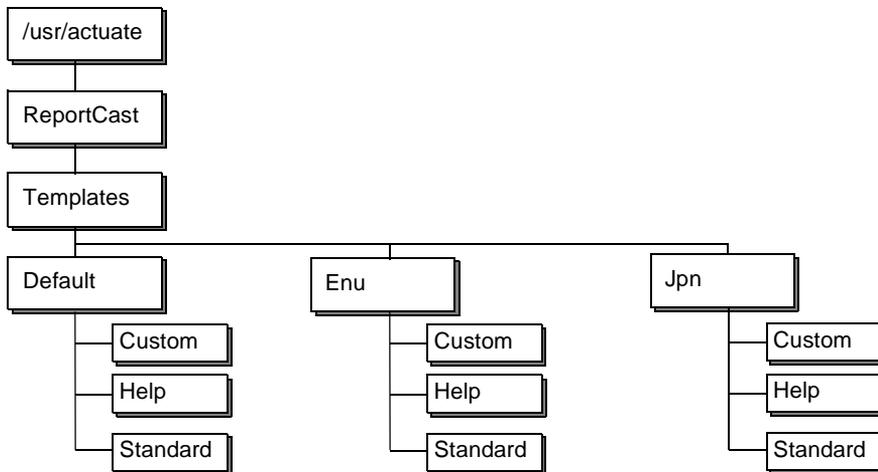
For NT ReportCast, UNIX ReportCast Agent, and UNIX installations with ReportCast Agent and ReportCast Server on the same machine, the structure is similar to the following diagram.



In this diagram, `wwwroot` represents the document root for your web server.

The directories `jpn` and `enu` are standard abbreviations for localized languages. The directory `default` is the default ReportCast configuration. If you do not specify a configuration, ReportCast uses the default directory.

For UNIX ReportCast Server installations, the structure is similar to the following diagram.



In this diagram, `/usr/actuate` represents the root of your Actuate distribution directory structure.

UNIX ReportCast installations with ReportCast Agent and ReportCast Server on separate machines have the following additional considerations:

- Static content, such as Javascript files and images, reside on the web server with the ReportCast Agent.
- The .ACHTML template files reside in the Actuate distribution directory structure with the ReportCast Server.
- To create custom ReportCast configurations with a split ReportCast installation, you must customize Javascript files and images on the ReportCast Agent machine, and customize templates on the ReportCast Server machine.

The installation process automatically creates a directory structure for each language that ReportCast supports. The default template files go in the standard directories. Copy the templates you want to customize to the appropriate custom directory, and make changes to them there. This process ensures that you always have the original default templates. The help directories contain online help files for ReportCast. The images directories contain the images and icons that display on the reporting web pages, such as the folder, schedule, and search icons.

## Creating and customizing ReportCast template files

Create and customize ReportCast template files in the same way that you work with any web page: use an editing tool or an HTML authoring tool. Most authoring tools do not support the ReportCast scripting language. You can use authoring tools to design your ReportCast pages, then add the scripting commands later. If your authoring tool supports adding HTML comments to your web pages, use that feature to add the scripting commands.

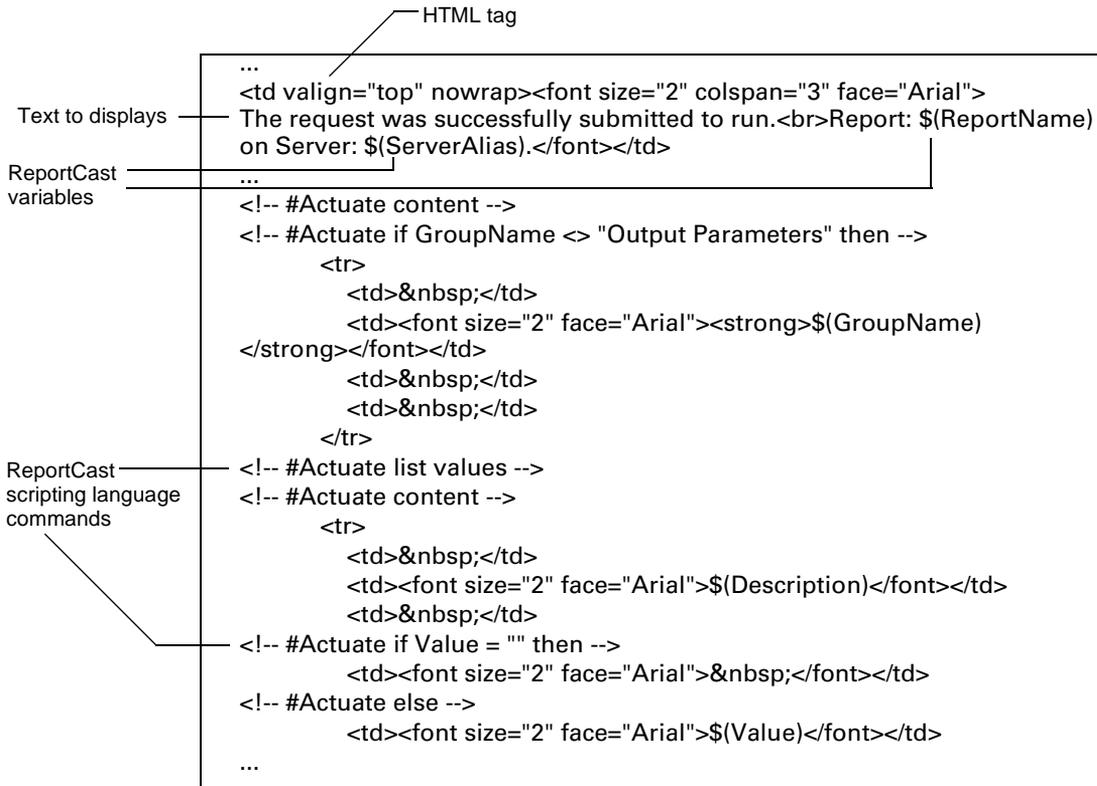
### Template file elements

The default HTML template files contain these elements:

- HTML tags that determine text format
- Text
- Variables, which ReportCast substitutes with data according to the directive you issue. These variables appear in the following format:
  - \$ (VARIABLENAME)
- ReportCast scripting language directives. These directives appear in the following format:
  - <!-- #Actuate <directive> [options] -->
- Javascript

The following is a fragment of the HTML document source for the CONTENTCONFIRM.ACHTML file, a template file that generates the content

portion of the request confirmation web page. The request confirmation page displays after a user submits a request to run a report.



## Modifying HTML template files

You can easily modify template files written in HTML, to change the content and format of the HTML pages generated by ReportCast. For example, you can change the background pattern or color, or replace the Actuate bitmap in the HTML page. You can also add or delete variables in the template files to change the data.

Before editing any of these template files, you should:

- Be well-versed in HTML.
- Understand how to use ReportCast variables in templates.
- Be familiar with the ReportCast scripting language.
- Be familiar with Javascript.

For more detailed information about ReportCast scripting language variables, see “ReportCast scripting language variables,” in Chapter 5, “ReportCast scripting language reference.” For more detailed information about descriptions of the ReportCast scripting language, see Chapter 5, “ReportCast scripting language reference.”

## Locating ReportCast template files

ReportCast expects:

- To find the templates in specific locations
- Templates to have specific names

If you put your templates in different locations, or give them different names, ReportCast will not find them. If ReportCast does not find the appropriate template, that template cannot be used to generate web pages. If you change the name of an included subtemplate file, you must change all references to that subtemplate file in all the ReportCast templates that refer to it. For more information about naming ReportCast template files, see “Naming template files,” later in this chapter.

## Understanding ReportCast searches for templates

When it receives a directive, ReportCast first determines whether or not to look for custom templates. It then searches for custom templates in a specific order:

- 1 Determine the search type. The request may be to display folder contents or to run a report:
  - If this is a request to display folder contents, look in the e.Reporting Server folder for an INDEX.ACHTML file. If there is an INDEX.ACHTML file, use it to generate the folder contents web page.
  - If this is a request to run a report, look in the e.Reporting Server folder for an HTML file that depends on the report’s ROX. If there is one, assume that it is a custom requester form for the report and display it as the requester web page. If there is more than one HTML file that depends on a particular ROX, display a requester menu form. The user chooses the form.
- 2 Search for template files in the e.Reporting Server’s `__WebAgent` folder. Templates in the `__WebAgent` folder generate customized web pages for this e.Reporting Server.
- 3 For NT installations and UNIX installations with ReportCast Agent and ReportCast Server on the same machine, search for template files in the web server’s Custom directory for the configuration currently in use. For UNIX installations with ReportCast Agent and ReportCast Server on

separate machines, search for template files in the ReportCast Server's Templates\`<config>`\Custom directory in the Actuate distribution directory, where `<config>` is the configuration currently in use.

Templates in the Custom directory generate web pages for this particular configuration. For example, if the configuration in use is NT ReportCast's French configuration, ReportCast uses the custom templates in `wwwroot\Actuate\Frn\Custom` to generate French-language web pages.

- 4 For NT installations and UNIX installations with ReportCast Agent and ReportCast Server on the same machine, search for template files in the web server's Standard directory. For UNIX installations with ReportCast Agent and ReportCast Server on separate machines, search for template files in the ReportCast Server's Templates\`<config>`\Standard directory in the Actuate distribution directory, where `<config>` is the configuration currently in use.

If no other template files exist, ReportCast uses the default templates in the Standard directory.

You can develop custom templates for a particular e.Reporting Server or web server to create a distinctive look for a departmental e.Reporting Server or web server. Custom templates ensure that the department's web pages have a consistent design across your company's intranet.

For NT installations and UNIX installations with ReportCast Agent and ReportCast Server on the same machine, the default template files are located in the Actuate directory within the web server's document root directory. For example, if your web server's document root is `wwwroot`, the default template files are in `wwwroot\Actuate\Default\Standard`.

For UNIX installations with ReportCast Agent and ReportCast Server on different machines, the default template files are located in the ReportCast server's `ReportCast\Templates\Default\Standard` directory of your Actuate distribution directory structure.

The following table lists the expected locations for ReportCast template files. Docroot refers to the document root, whether on the web server, or in the UNIX ReportCast Server directory structure.

Location	Description
Docroot\ <code>[config]</code> \Standard	Location of the default templates for configuration <code>[config]</code> of this web server. The default directory contains the default Actuate ReportCast configuration.
Docroot\ <code>[config]</code> \Custom	Location of custom templates that you create for configuration <code>[config]</code> of this web server.

Location	Description
__WebAgent	Location of custom templates that you create for this e.Reporting Server. Use the e.Reporting Server Navigator to create this folder on your e.Reporting Server.
INDEX.ACHTML	The custom template you create specifically for a particular folder on your e.Reporting Server. Use the e.Reporting Server Navigator to place this template file in the appropriate folder on your e.Reporting Server.

A template's location—in an e.Reporting Server's WebAgent folder, in a custom directory, and so on—determines its scope. For more information about placement of templates for customizing part or all of the ReportCast interface, see "About template placement and scope," earlier in this chapter.

## Creating custom requester forms for a specific report

You can create custom requester forms for a particular report. If an .ROX or an .ROV has one or more HTML files that depend on it, ReportCast treats those files as custom requester forms. When a request to run one of those reports occurs, ReportCast displays the custom requester form rather than the standard form. If there is more than one custom requester form, ReportCast displays a menu and allows the user to choose the form to use.

### How to create a custom requester form

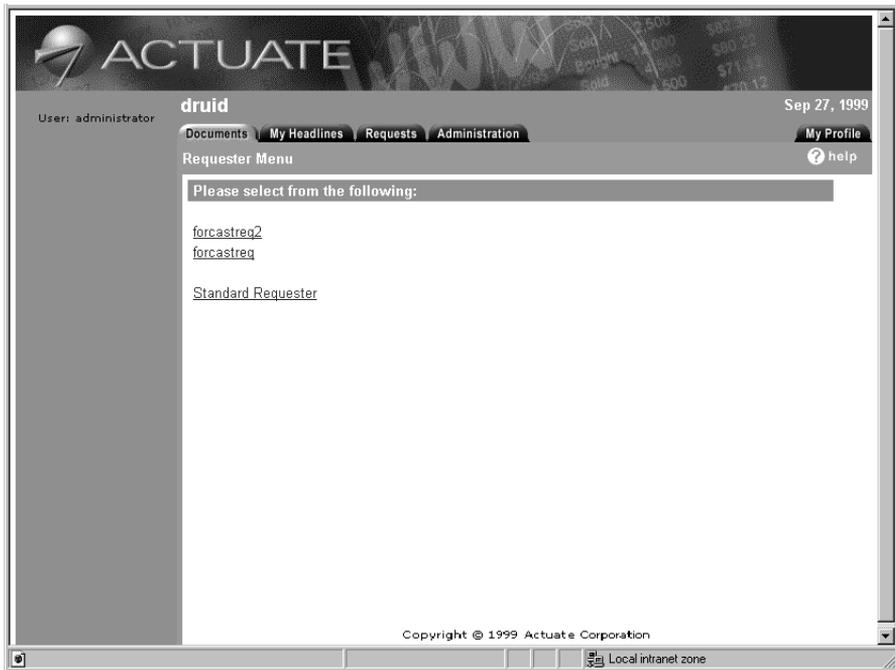
- 1 Use the editing tool of your choice to create a custom HTML requester form, perhaps using the standard requester, REQUEST.ACHTML, as a model.
- 2 Save the requester form with the extension .HTML.
- 3 In the Administrator Desktop, navigate to the e.Reporting Server folder containing the report. That is where you attach the custom HTML requester form.

### How to attach a custom requester form to a report

- 1 Select the e.Reporting Server folder that contains the report.
- 2 Right-click to access the Context menu.
- 3 Choose New Report Item from the Context menu.  
The Save To Volume dialog box displays.
- 4 Locate the custom requester. Save it to the current folder.

- 5 Select the custom requester.
- 6 Right-click to access the Context menu.
- 7 Choose Properties from the Context menu.
- 8 Choose the Dependency tab.  
The Dependency window displays.
- 9 Choose Add to add the .ROX as a dependency for the custom requester.
- 10 Choose OK.

If there is more than one custom HTML requester form for a particular .ROX, ReportCast displays a form similar to the following when it receives a request to run a report.



### How to select a requester form

The user selects the appropriate requester form link to use that form.

### Reloading template files

Changes to the template files are not reflected in the generated HTML pages immediately. You must reload the templates from the ReportCast

Administration page to see the changes. Only users with administrative privileges can access the ReportCast Administration page.

## How to reload templates

- 1 To access the ReportCast Administration page, either choose the Administration tab, or type the URL in your browser. The URL is similar to:  
`http://<webserver>/acweb/_ _admin.`  
where <webserver> is the name of the machine that runs your web server. If you choose the Administration tab, the ReportCast Channel Administration page appears. Click the ReportCast administration link on the left side of the page. The ReportCast administration page appears.
- 2 Click the Reload Templates link.  
The Confirmation page displays if the templates reload successfully.
- 3 Verify your changes by displaying the appropriate ReportCast page.

## Naming template files

The following table lists the names of the ReportCast template files. All templates have the file extension .ACHTML. All templates are in the Standard directory of the web server's Actuate directory. You must follow this naming convention for any custom template files you create or ReportCast will not find them.

Template	Description
Active	Displays active requests.
Activedetail	Lists the properties for an active request.
Admin	Displays the ReportCast Administration page.
Cancel	Displays confirmation that a request has been cancelled.
Channeladmin	Displays the ReportCast Channel Administration page.
Channelcontents	Displays the contents of a particular channel.
Channeldetail	Lists the properties for a particular channel.
Channelitemdetail	Lists the properties for a particular channel item.
Channelsubscription	Displays the ReportCast channels Subscription page.
Confirm	Displays confirmation that a request to run a report was submitted to e.Reporting Server.

<b>Template</b>	<b>Description</b>
Error	Displays information about an error condition.
Filedetail	Lists the properties for a particular report item.
Fileversions	Lists the available versions of the current file.
Folderlist	One-page view of a Report Encyclopedia folder.
Index	Custom folder page created by the administrator, and stored directly on the e.Reporting Server.
Personalchcontents	Displays the contents of a particular user's Personal Channel folder.
Request	Requester page generated by ReportCast.
Requestdetail	Displays the details for a particular request.
Requestermenu	Lists the available requester forms. This page displays only if there are custom requester forms available.
Requestlist	Lists the Active, Scheduled, or Completed requests.
Scheduled	Lists all scheduled requests on the current e.Reporting Server.
Scheduleddetail	Lists the properties of a scheduled request.
Status	Displays the status of an active or completed request.
Success	Displays confirmation that a Drop or Rename operation completed successfully.
Viewframeset	Displays the frameset for viewing the DHTML report.
Viewnav	Composes the toolbar used for viewing DHTML reports.
Viewtoc	Displays the DHTML report's table of contents.

The following table maps ReportCast directives to the template(s) they generate or affect.

<b>Directive</b>	<b>Template</b>
__active	Requestlist
__completed	Requestlist
__scheduled	Requestlist
Cancel	Cancel

<b>Directive</b>	<b>Template</b>
__channels?admin	Channeladmin
__channels?retrieveForm	Channelsubscription
__channels?submitForm	Confirm
__channels?retrieveCreation	Channeldetail
__channels?retrieveDetails	Channeldetail
__channels/channel?drop	Success
DeleteStatus	Success
Detail	Filedetail
Request	Requestermenu
Sort and Direction	Folderlist
StdRequest	Request
Submit	Confirm
Drop	Cancel
FlushConnections	Admin
ViewFrameset	Viewframeset
GetDynamicData	Viewframeset
ViewPage	Viewframeset
GetReportData	
GetStyleSheet	Viewframeset
ViewTOC	Viewtoc
ViewNavigation	Viewnav
Versions	Fileversions
requestSearch	Searchrequest
submitSearch	Searchresults

## **Placing custom templates in a Report Encyclopedia**

When you want to create a distinctive design for a particular department's e.Reporting Server web pages, use the Administrator Desktop's Navigator to place custom template files directly in the e.Reporting Server Encyclopedia. You must have administrator privileges on both the web server and the e.Reporting Server to perform this procedure.

### **How to place custom templates in a Report Encyclopedia**

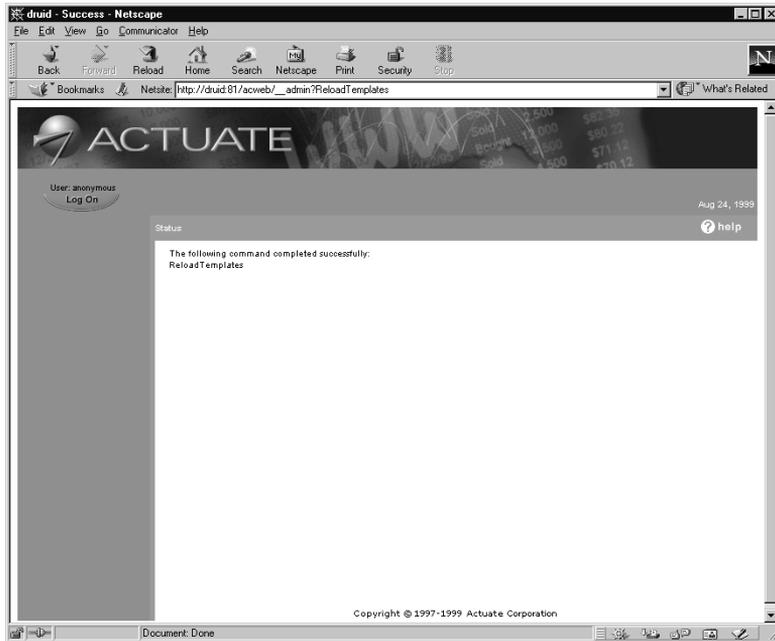
- 1 Create the \_\_WebAgent folder on your Report server:

- 1 Start Actuate Administrator Desktop.
  - 2 Choose File→Navigator.  
The Report Encyclopedia Login dialog box appears.
  - 3 Enter your login information.
  - 4 Choose OK.
  - 5 Choose the e.Reporting Server destination for the custom ReportCast templates.
  - 6 Choose File→New Encyclopedia Item→Folder.  
The Create a new folder dialog box appears.
  - 7 Type `__WebAgent` in the Enter folder name box. Choose OK.  
The `__WebAgent` folder appears in the Navigator windows.
- 2** Create custom templates for the Report Encyclopedia:
- 1 Copy the default templates from the web server configuration's Standard directory to a work directory.
  - 2 Modify the templates.
  - 3 In the Navigator, open the `__WebAgent` folder.
  - 4 Choose File→New Encyclopedia Item→Report Item.  
The Save to Volume dialog box appears.
  - 5 In the Files of Type box, choose All Files (\*.\*) .
  - 6 Browse to the location of your custom templates.
  - 7 Select a template to place in the `__WebAgent` folder.
  - 8 Choose Save to Volume.  
The template appears in the Navigator window's `__WebAgent` folder contents listing.

Repeat steps 4 and 5 for each template you want to place in the `__WebAgent` folder.

To view the HTML source of a page you see in the Netscape browser, choose View→Document Source. The source document identifies the template file ReportCast used to generate the page.

The following illustrations show the source for the SUCCESS.ACHTML template file and the web page it generates. The complete Success web page is similar to the following page.



The following sections show the template source code that generates specific parts of the Success web page.

### Web page title, background, banner, page header, username, date, and sidebar graphics



```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html>
<!-- Actuate Template File: $(TemplateName) -->
<!-- #Actuate Include "header.achtml" -->
<head>
<!-- #Actuate Include "browserstest.achtml" -->
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<Meta Name="Generator" Content="Actuate Web Agent 3.0">
<title>$(Host) - Success</title>
<!-- #Actuate Include "headerscript.achtml" -->
</head>
<!-- #Actuate Include "bodybegin.achtml" -->
<!-- #Actuate Include "banner.achtml" -->
```

```

<!-- PageHeader Begin -->
<!-- #Actuate Set CurrentTab = tabNone -->
<!-- #Actuate Set ServerString = "" -->
<!-- #Actuate Include "pageheader.achtml" -->
<!-- PageHeader End -->
<!-- SideBar Begin -->
<!-- #Actuate Include "sidebarbegin.achtml" -->
<!-- #Actuate Include "sidebarsuccess.achtml" -->
<!-- #Actuate Include "sidebarend.achtml" -->
<!-- SideBar End -->

```

## Display Status, result of command execution, command executed



```

<!-- #Actuate Include "contentheader.achtml" -->
<!-- PageTitle Begin -->
<!-- #Actuate Set PageTitle1 = "Status" -->
<!-- #Actuate Set PageTitle2 = "" -->
<!-- #Actuate Set PageTitle3 = "" -->
<!-- #Actuate Set PageTitle4 = "" -->
<!-- #Actuate Set PageTitle5 = "" -->
<!-- #Actuate Include "pagetitle.achtml" -->
<!-- PageTitle End -->
<!-- Content Begin -->
<!-- #Actuate Include "contentbegin.achtml" -->
<!-- #Actuate Include "contentsuccess.achtml" -->
<!-- #Actuate Include "contentend.achtml" -->
<!-- Content End -->

```

## Display copyright information, page footer, close page elements



```

<!-- #Actuate Include "contentfooter.achtml" -->
<!-- #Actuate Include "pagefooter.achtml" -->
<!-- #Actuate Include "bodyend.achtml" -->
<!-- #Actuate Include "metaend.achtml" -->
</html>

```

For more information about ReportCast scripting language variables, see “ReportCast scripting language variables,” in Chapter 5, “ReportCast scripting language reference.”

---

## Creating custom templates

This section provides examples of customizing the HTML pages generated by ReportCast.

### How to customize a particular template

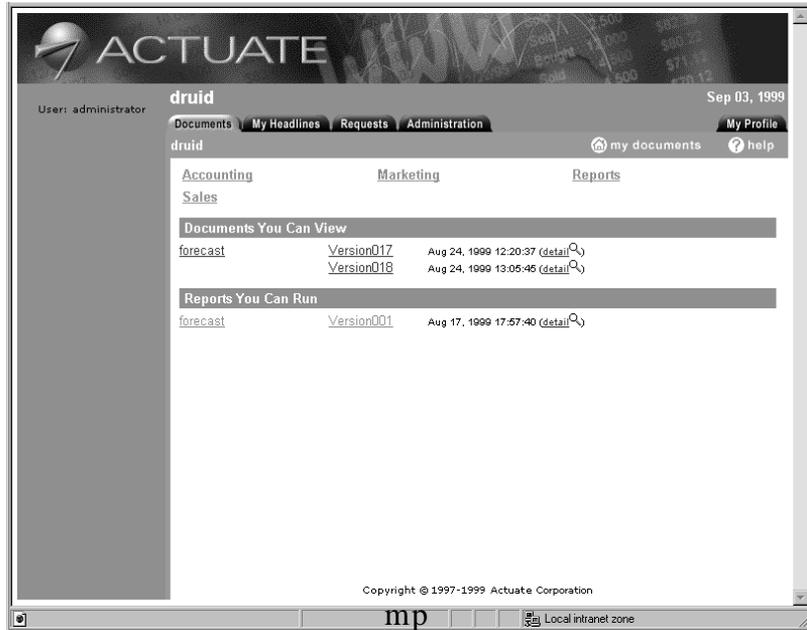
- 1 Display the standard web page for the template you want to customize.
- 2 In your web browser, go to a Report Encyclopedia folder containing report executables.
- 3 Click a report executable.  
The Requester web page displays in your browser.
- 4 Choose File→Save As.  
Save the web page in your configuration’s custom directory.
- 5 Edit the file in the Custom directory.

The standard templates include other template files and use variable substitution to display appropriate banners and icons on the resulting web page. The advantage to this method is that various elements are easily reused by several templates. The disadvantage is that elements such as nested included files can be confusing.

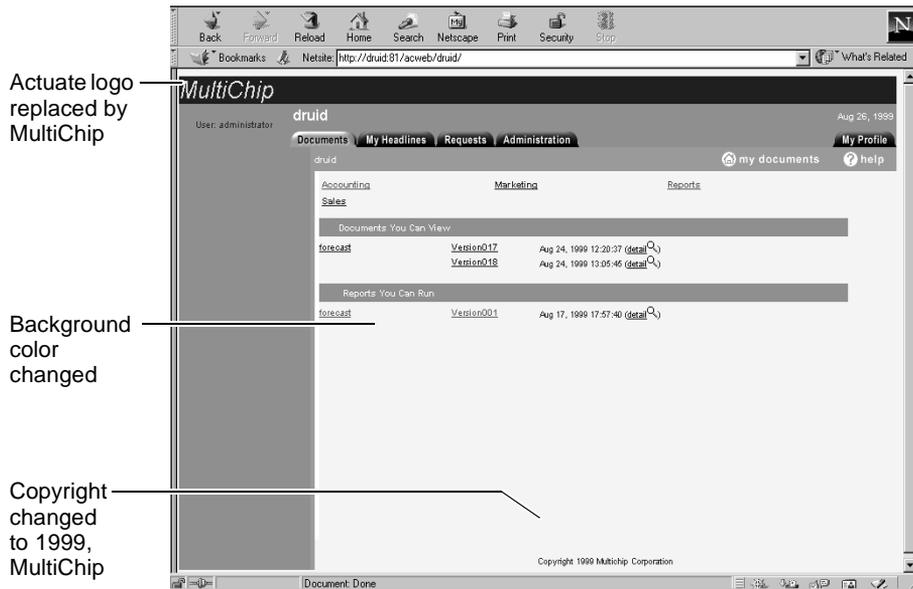
You can display the page you wish to customize in the web browser, then save it. The web page source is much more straightforward, substitutions have been made and the contents of included files are specifically included. You can edit and customize the complete web page.

**Example** This example shows how you can modify template files to change the format and data that ReportCast generates when it receives an `http://.../<folder>` directive. The following illustration shows the HTML pages generated from the default and custom template files.

## Default HTML page generated from a http://.../ <folder> directive



## Customized HTML page



## Modifying templates

To make the changes shown in this example, you modify these files:

- FOLDERLIST.ACHTML, the template that generates Folder listing pages.
- CONTENTFOLDERLIST.ACHTML, included by FOLDERLIST.ACHTML to generate the content for the Folder listing pages.
- HEADER.ACHTML, included by FOLDERLIST.ACHTML to define reporting web page colors, element names, commonly used URL's, and locations of images.
- HEADERSCRIPT.ACHTML, a Javascript file included by FOLDERLIST.ACHTML to preload images for the reporting web page.
- BANNER.ACHTML, included by FOLDERLIST.ACHTML to generate the company logo.
- COPYRIGHT.ACHTML, included by FOLDERLIST.ACHTML to generate the copyright.

Before you get started, create a directory called Custom in your configuration directory. For example, if you are using the U.S. English configuration, create the custom directory in your web server's `actuate\enu` directory.

You must reload the templates to activate your changes. Reload templates from the ReportCast administration page, or use the `ReloadTemplates` directive.

## How to replace one logo with another

Because the following is such a common requirement, the following steps take two sample logos and describe how to replace the Actuate logo with the Multichip logo:

- 1 Copy BANNER.ACHTML from the Standard directory to the custom directory.
- 2 Edit the file. The default contents are:

```
<tr>
  <td valign="top" colspan=1 bgcolor="$(colorLogoBack)" nowrap>
    </td>
  <td width="100%" valign="top" colspan=6 bgcolor="$(colorLogoBack)"
    nowrap>
  </td>
</tr>
```

The images `imgLogo1` and `imgLogo2` generate the default Actuate logo and banner for all the reporting web pages. The `HEADER.ACHTML` template defines `imgLogo1` and `imgLogo2`.

- 3 Replace the HTML code:

```

```

as follows:

```
<font color="yellow" size="6" face="Courier"><strong><em>  
MultiChip Corporation</em></strong></em></font>
```

- 4 Replace the HTML code:

```

```

as follows:

```
&nbsp;
```

- 5 Save BANNER.ACHTML.

- 6 View the changes:

- 1 Reload the templates.

- 2 Return to the ReportCast home page. The home page is now similar to the following image.



## How to change the border color

- 1 Copy the FOLDERLIST.ACHTML, PAGEHEADERX.ACHTML, and HEADER.ACHTML templates to the custom directory.

- 2 Edit PAGEHEADERX.ACHTML:

- 1 Search for the string BGCOLOR. Search results include lines such as this one:

```
<TD HEIGHT="10" COLSPAN="6" VALIGN="bottom"  
BGCOLOR="$(colorLightBlue)">
```

- 2 Replace the color (\$colorLightBlue) with turquoise. For example:

```
<TD HEIGHT="10" COLSPAN="6" VALIGN="bottom" BGCOLOR="turquoise">
```

- 3 Save PAGEHEADERX.ACHTML.

- 3 Edit FOLDERLIST.ACHTML:

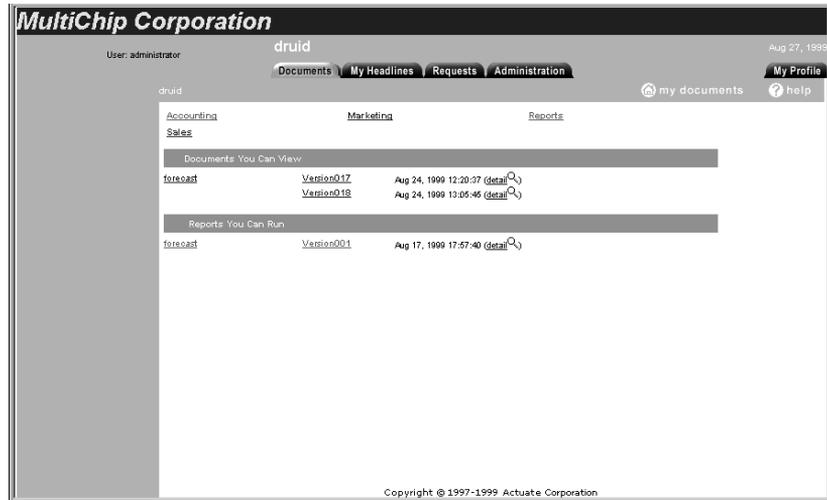
Search for the string BGCOLOR. Search results include lines such as this one:

```
<TD WIDTH="149" BGCOLOR="$(colorLightBlue)" VALIGN="TOP"  
NOWRAP><IMG SRC="$(imgSpace)" HEIGHT="5" WIDTH="149"></TD>
```

Replace the color (\$colorLightBlue) with turquoise. For example:

```
<TD WIDTH="149" BGCOLOR="turquoise" VALIGN="TOP" NOWRAP>  
<IMG SRC="$ (imgSpace)" HEIGHT="5" WIDTH="149"></TD>
```

- 4 Save FOLDERLIST.ACHTML.
- 5 Reload the templates. The web page looks similar to the following graphic.



The steps in the previous procedure result in a turquoise border for the Folder listing page. You can also include custom background images or set custom background colors for your Folder listing page.

The HEADER.ACHTML file contains settings for all the templates' image variables.

### How to change the copyright information

The standard templates use a copyright image to display copyright information on the reporting web pages. To change the copyright information, change the image file, IMG\_COPYRIGHT.GIF, in the images directory.

## Customizing the DHTML and SmartSearch displays

This section describes how to customize the DHTML and SmartSearch displays.

The following table lists the files that affect DHTML and SmartSearch display:

<b>File</b>	<b>Description</b>
<b>Frameset</b>	Determine the DHTML frameset look and feel
VIEWFRAMESET.ACHTML	
<b>Toolbar</b>	Determine the look and feel of the DHTML toolbar
VIEWNAV.ACHTML	
VIEWNAV.JS	
<b>Search</b>	Determine the SmartSearch look and feel
SEARCHREQUEST.ACHTML	
SEARCHRESULTS.ACHTML	
SEARCH.JS	
<b>Navbar (Table of Contents)</b>	Determine the look and feel of the DHTML table of contents
VIEWTOC.ACHTML	
TOCTREE.JS	
Download as PDF	
Save as ahtml	

# ReportCast directives

This chapter contains the following topics:

- About ReportCast directives
- Types of ReportCast directives
- About Channel directives
- Examples of ReportCast directives

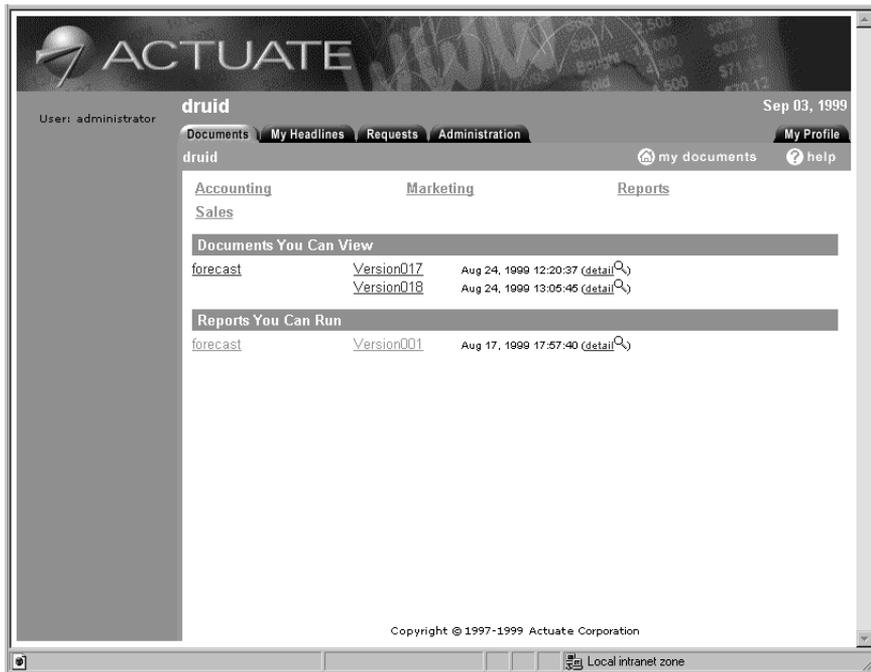
# About ReportCast directives

ReportCast performs tasks based on directives from the web browser. Directives are URLs that contain the keyword acweb followed by the directive. When the web server encounters the acweb keyword, it passes the entire URL to ReportCast for processing.

## Sending directives

Directives are often associated with buttons or links in the HTML pages generated by ReportCast. Choosing one of those buttons or links sends a directive. Another way to send a directive is to type the URLs in the Address or Location window of the web browser.

The following illustration shows a sample page generated by ReportCast. The page contains many links that, when clicked, send the appropriate URL directive to ReportCast.



You can see what directive is associated with a link by placing the cursor over the link.

## How to view Netscape link information

The Netscape server displays complete URLs automatically.

## How to view Microsoft Internet Explorer link information

To display complete URLs in Microsoft's Internet Explorer:

- 1 Select View→Options.
- 2 Choose Advanced.
- 3 Deselect the Show friendly URLs.

The URL appears in the status line at the bottom of the browser window.

## Managing directives

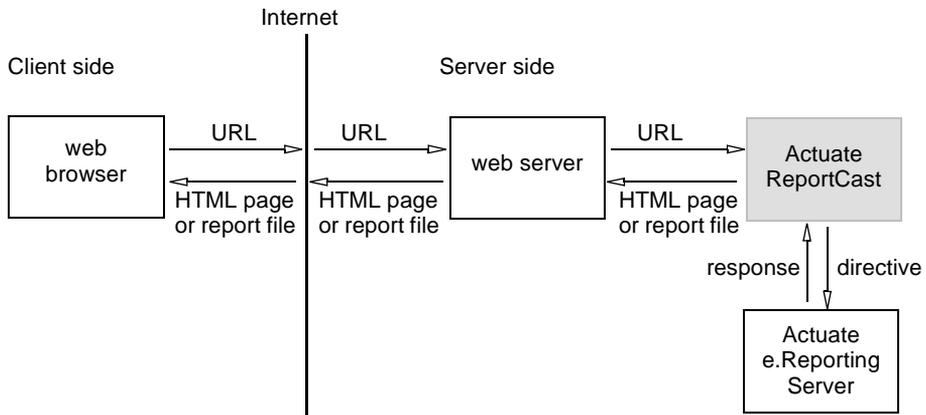
ReportCast uses HTML template files to generate HTML pages in response to directives sent from the web browser. Templates can be report-specific, folder-specific, e.Reporting Server-specific (apply to generating HTML pages on a particular e.Reporting Server), or web server-specific. ReportCast locates these template files by searching in a specific order:

- 1 The e.Reporting Server folder.
- 2 The e.Reporting Server's \_ \_WebAgent directory.
- 3 The web server's Custom directory for a particular configuration of the web server.
- 4 The web server's Standard directory.

For more detailed information about the search order, see "Understanding ReportCast searches for templates," in Chapter 3, "Working with ReportCast templates."

For detailed information about the HTML template files, see "About ReportCast templates," in Chapter 3, "Working with ReportCast templates."

The following illustration shows the flow of communication from the web browser to ReportCast and from ReportCast back to the web browser.



## Formatting directives

ReportCast directives use one of the following formats:

```
{http | https}://<webserver>/acweb/ [<configuration>/] <reportserver> | __local |
__default/ [<path>] [<resource>] [;<version>] [ ?<command> ]
```

or:

```
{http | https}://<webserver>/acweb/_ _admin
```

In the UNIX environment, for CGI programs only, the web agent directives use one of the following formats:

```
{http | https}://<webserver>/<cgi script location>/acweb/ [<configuration>
] <reportserver> | __local | __default/
[<path>] [<resource>] [;<version>] [ ?<command> ]
```

or:

```
{http | https}://<webserver>/<cgi script location>/acweb/_ _admin
```

The following section provides additional information about each of the parameters of the directives.

### http | https

- Required.
- Specifies that the web browser and web server communicate using the HTTP communications protocol.

If your web server runs on the Secure Sockets Layer, URLs begin with https instead of http.

### **<webservice>**

- Required.
- Name of the web server machine.

### **<cgi script location>**

- Required to run ReportCast on UNIX using CGI programs.
- Not applicable to other platforms.

### **acweb**

- Required.
- Tells the web server to pass the rest of this URL's fields to ReportCast.

### **\_\_admin**

- Optional.
- Directive that accesses the ReportCast Administration page.

### **<configuration>**

- Optional.
- Name of the set of templates ReportCast should use to generate web pages for this request.

### **<reportserver> | \_\_local | \_\_default**

- Required.
- Name of the e.Reporting Server you want to access.

If the e.Reporting Server and the web server are installed on the same machine, specify the literal `__local` instead. If the e.Reporting Server is the default e.Reporting Server for ReportCast, specify the literal `__default` instead.

For information about setting the default e.Reporting Server, see "Setting ReportCast's default e.Reporting Server" in Chapter 1, "About Actuate ReportCast."

### **<path>**

- Optional.
- Full path to the Report Encyclopedia folder or file to access.

### **<resource>**

- Optional.

- A file or other resource in the Report Encyclopedia folder specified by <path>. If you do not specify this field, ReportCast accesses the folder specified by <path>.

**;<version>**

- Optional.
- Version number of the folder, file, or other resource to access.

**?<command>**

- Optional.
- Action to execute.

---

## Types of ReportCast directives

The following table lists the types of ReportCast directives.

Type of directive	Description
Folder directive	Specifies how the contents of a Report Encyclopedia display.
File directive	Downloads report files from the e.Reporting Server.
Factory directive	Generates new reports from report executables (.ROX).
Status directive	Gets report request status.
Channel directive	Administers channels and retrieves channel contents.
User preferences directive	Displays the user preferences form and submits user preferences changes.
Viewing directive	Determines how to view completed reports.
Search directive	Performs searches on DHTML reports.

## ReportCast directives

The following table lists the ReportCast directives.

Directive	Description
__admin	Accesses the ReportCast administration page. Do not specify a report server in the URL when accessing the administration page.
__home	Accesses the current user's home folder in the Report Encyclopedia, as set in the Administrator Desktop.
?DeleteStatus	Deletes status of a report-generation request.
?Detail	Generates detailed information about a Report Encyclopedia object.
?Drop	Cancels the report-generation request. Deletes a Report Encyclopedia object.
?Files	Lists only the files the current folder contains. Does not list subfolders.
?Reports	Displays the reports in the specified folder, sorted by name.
<report URL>	Downloads the entire report document.
?Request	Generates requester forms.
?StdRequest	Generates the standard requester form.
?Submit	Submits a request. Starts the report generation process.
?Status	Displays status for the current report-generation request.
?Sort and Direction	Specifies how to display the contents of a Report Encyclopedia.
Versions	Displays a list of the available versions of a file when the file the user is viewing in DHTML becomes unavailable.
?View	Displays a report, one page at a time, in a browser. For HTML reports, ?View displays the entire report in the browser.
<report URL> ?ViewDefault	Displays a report in a format determined by the user's preference and viewing privileges.
<report URL> ?ViewFrameset	Displays the report in DHTML.

<b>Directive</b>	<b>Description</b>
<code>--_scheduled</code>	Displays the list of currently scheduled factory requests.
<code>--_active</code>	Displays the list of currently active factory requests.
<code>--_completed</code>	Displays the list of completed factory requests.
<code>--_channels?admin</code>	Displays the channels administration page.
<code>--_channels?retrieveForm</code>	Displays the subscription form for available channels.
<code>--_channels?submitForm</code>	Submits subscription information.
<code>--_channels?retrieveCreation</code>	Displays the form used to add new channels.
<code>--_channels?submitCreation</code>	Submits the new channel form.
<code>--_channels/channel?drop</code>	Removes an existing channel.
<code>--_channels/channel?retrieveDetails</code>	Displays a form used to modify channel attributes.
<code>--_channels/channel?submitDetails</code>	Submits the channel details form.
<code>--_requestUserPreferences</code>	Requests the user preferences page.
<code>--_submitUserPreferences</code>	Submits the user preferences page as an HTML form.
<code>&lt;report URL&gt;?GetDynamicData</code>	Retrieves data for a specific component, such as a graph or image in a report.
<code>&lt;report URL&gt;?ViewPage</code>	Retrieves a specific page of a report, either from the navigation bar or from a set of search results.
<code>&lt;report URL&gt;?GetReportData</code>	Retrieves report data in a particular format.
<code>&lt;report URL&gt;?GetStyleSheet</code>	Retrieves the style sheet for the report in a particular format, such as CSS or XMLStyle.
<code>&lt;report URL&gt;?ViewTOC</code>	Retrieves the report's table of contents from the view process.
<code>&lt;report URL&gt;?ViewNavigation</code>	Displays the navigation bar for viewing a report.

Directive	Description
<report URL>?Versions	Displays the list of available versions of a file, when the file the user is viewing in DHTML becomes unavailable.
<report URL> ?requestSearch	Displays the search request page.
<report URL> ?submitSearch	Submits the search request and returns the search results page.
<report URL> ?extractSearch Results	Submits the search and returns search results in a specified format, such as comma-separated values or tab-separated values.

## Directives syntax

In the following sections, the discussion of directives groups the directives by the objects they affect. All directives begin with one of the following forms of syntax:

http://<webserver>/acweb/<reportserver>/<folder>?

http://<webserver>/acweb/<reportserver>/<folder><file>?

### <webserver>

Names the machine that is running the web server.

### acweb

Literal string indicating that the rest of this string must be passed to ReportCast for processing.

### <reportserver>

Names the e.Reporting Server to access.

### <folder>

Names the folder to access on the e.Reporting Server.

### <file>

Names the report document to access on the e.Reporting Server.

### ?

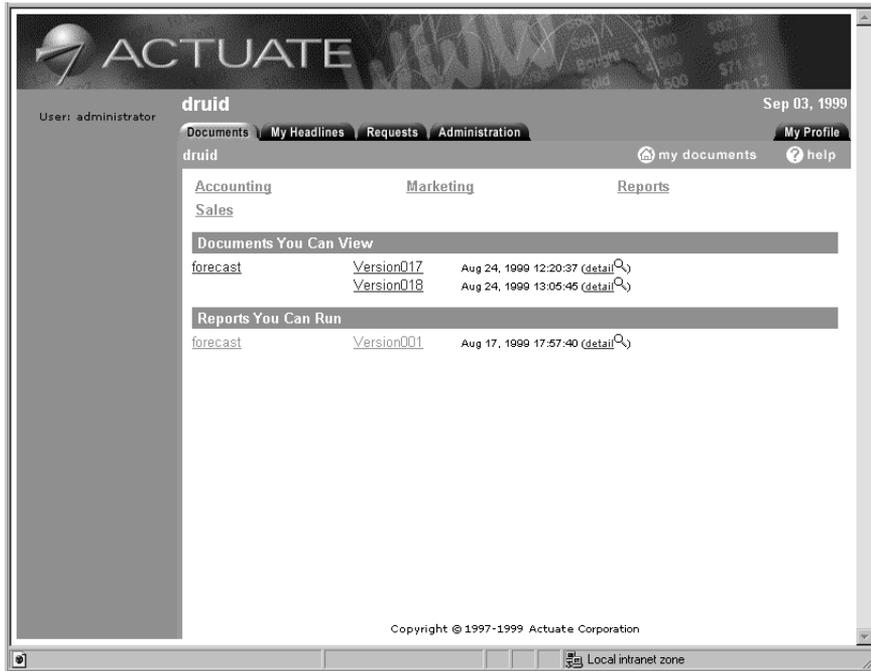
Indicates that the directive and the directive's parameters follow.

## Folder directives

This section describes the folder directives that specify how to display the contents of a Report Encyclopedia.

## About sort and direction directives

When you tell ReportCast to display the contents of a folder on the report server, it displays the folder's files sorted by name and version.



You can choose to display a folder's contents in different ways. For example, you can decide to view reports sorted by size, with the smallest reports listed first:

<http://mustique/acweb/mustique/Marketing?sort=size&direction=desc>

## Using sort and direction directives

Use the sort directive to specify the sort order for a list of Report Encyclopedia folder contents. Use the direction directive to specify whether the sort order should be ascending or descending.

In the previous example, the URL displays Report Encyclopedia reports sorted by size, with the largest reports listed first.

Specify the sort and direction directives in any sequence. Remember to separate directives and their options with ampersands (&). The ampersand (&) is the standard command separator character for URLs.

Here is the syntax for the sort and direction directives:

`http://<webserver>/acweb/<reportserver>/<folder> [?sort=<field>] [&direction = [desc | asc] ] [files]`

For information about the general syntax of directives, see “Directives syntax,” earlier in this chapter. The following section provides additional information about each of the parameters of the sort and direction directives.

### **?sort=<field>**

The sort directive, followed by the sort field. The following table lists the valid sort fields. These sort fields are associated with the Name, Type, Owner, Date, and Size links on the HTML page that displays the contents of a folder.

<b>&lt;field&gt; value</b>	<b>Description</b>
Name	Sorts the folder contents by name and version number
Type	Sorts the folder contents by file type
Owner	Sorts the folder contents by owner
Date	Sorts the folder contents by creation date
Size	Sorts the folder contents by size

### **direction = [desc | asc]**

The direction of the sort, descending (desc) or ascending (asc). Ascending is the default.

### **files**

Lists only the files that this folder contains. Does not list subfolders.

## **Displaying only report files in a folder**

The following table lists the directives you can send if you want to view only report files, not subfolders.

<b>Directive</b>	<b>Displays ...</b>
<code>http://&lt;webserver&gt;/acweb/&lt;reportserver&gt;/&lt;folder&gt;/?files</code>	Reports in the specified folder
<code>http://&lt;webserver&gt;/acweb/&lt;reportserver&gt;/&lt;folder&gt;/?files&amp;sort=Name</code>	Reports in the specified folder, sorted by name
<code>http://&lt;webserver&gt;/acweb/&lt;reportserver&gt;/&lt;folder&gt;/?files&amp;sort=Type</code>	Reports in the specified folder, sorted by file type

Directive	Displays ...
<code>http://&lt;webserver&gt;/acweb/&lt;reportserver&gt;/&lt;folder&gt;/?files&amp;sort=Owner</code>	Reports in the specified folder, sorted by creator
<code>http://&lt;webserver&gt;/acweb/&lt;reportserver&gt;/&lt;folder&gt;/?files&amp;sort=Date</code>	Reports in the specified folder, sorted by creation date
<code>http://&lt;webserver&gt;/acweb/&lt;reportserver&gt;/&lt;folder&gt;/?files&amp;sort=Size</code>	Reports in the specified folder, sorted by size
<code>http://&lt;webserver&gt;acweb/&lt;reportserver&gt;/&lt;folder&gt;/?files&amp;sort=Date&amp;direction=desc</code>	Reports in the specified folder, sorted by creation date in descending order

## Viewing DHTML reports

If Dynamic HTML (DHTML) is your report viewing preference, ReportCast displays reports in your web browser in DHTML format. This section describes the DHTML report viewing directives.

### Displaying DHTML reports

The ViewFrameset directive generates the toolbar for report navigation and displays the initial frame set for viewing a report in DHTML format. Use the following syntax to display reports in DHTML format:

```
http://<webserver>/acweb/<reportserver>/<folder>/
<file>?ViewFrameset&page=<page>
```

For information about the general syntax of directives, see “Directives syntax,” earlier in this chapter.

Use the page keyword to specify the number of the initial page to display. The default initial page number is the first page of the report.

You can bookmark DHTML report pages.

Users generally view DHTML report pages using the navigation bar. The navigation bar includes buttons for next, previous, and first and last pages.



### Displaying available versions of DHTML reports

The Versions directive displays a list of the available versions of a report. Use the following syntax to display the list of available versions:

```
http://<webserver>/acweb/<reportserver>/<folder>/<file>?Versions
```

For information about the general syntax of directives, see “Directives syntax,” earlier in this chapter.

Available reports may expire after specific lengths of time. It is possible that a particular DHTML report could expire and become unavailable as you view it. When a report becomes unavailable, ReportCast displays a list of the available versions of the report. Click on the version you want to view. ReportCast displays that version of the report in DHTML format.

## Getting dynamic content of DHTML report components

The GetDynamicData directive retrieves data for a component that the view process generates dynamically. The component is usually a graph or image in a report. Use the following syntax to retrieve a dynamically-generated component:

```
http://<webserver>/acweb/<reportserver>/<folder>/  
  <file>?GetDynamicData&ComponentID=<ID>&scalingFactor=<scale>
```

For information about the general syntax of directives, see “Directives syntax,” earlier in this chapter.

Use the componentID keyword to specify the ID of the dynamically-generated component.

Use the scalingFactor keyword to specify the size of the component, or the zooming percentage. A scaling factor of 100 indicates the original size of the component. A scaling factor of 50 displays the component at 50% of its original size. 100 is the default scaling factor.

## Viewing a page from the navigation bar

The ViewPage directive allows printing or displaying a specific page of a DHTML report. Use the following syntax to prepare print or display output of a DHTML report page from the navigation bar:

```
http://<webserver>/acweb/<reportserver>/<folder>/  
  <file>?ViewPage&format=<format>&page=<page>&mode=<mode>  
  &scalingFactor=<scale>&operation=<operation>
```

For information about the general syntax of directives, see “Directives syntax,” earlier in this chapter.

Users generally display a specific page by typing the page number in the Page field of the navigation bar.

Users generally prepare print output of a specific page or range of pages by:

- 1 Clicking the Download/Print link in the navigation bar
- 2 Specifying the page or pages they wish to print in the Custom Report Saving Option form
- 3 Clicking the Apply button, then saving the print output to their local disk

The print output is PDF format. Preparing print output is the default. If your site requires that viewing is the default choice when using the Download/Print link, you must change the SAVEAS.ACHTML template. Change the following code:

```
operation=print
```

to

```
operation=view
```

Use the following syntax to display a page from a set of search results or from the report's table of contents:

```
http://<webserver>/acweb/<reportserver>/<folder>/  
  <file>?ViewPage&format=<format>&componentID=<ID>  
  &scalingFactor=<scale>
```

Users generally display a page from a set of search results by clicking a link in the search results list.

The format keyword specifies the format in which to display the report. Format is any format that the view process supports, such as XML structure or page-view formats, or DHTML. The valid values for format are:

- DHTML—a compressed DHTML format that uses Cascading Style Sheets (CSS). This is the default format.
- DHTMLLong—an uncompressed DHTML format. Use this format if your browsers do not support CSS.

The mode keyword indicates the page to display, relative to the current page number. Values for mode are:

- First—displays the first page of the report
- Last—displays the last page of the report
- Previous—displays the page that precedes the current page
- Next—displays the page that follows the current page
- Specific—displays the page specified by the page keyword

The scalingFactor keyword functions as it does for the GetDynamicData directive, described in "Getting dynamic content of DHTML report components," earlier in this chapter.

The operation keyword indicates whether to prepare PDF output for printing or to view PDF format report pages. The default is to prepare PDF format for printing reporting pages. The user then saves the prepared print output to disk and prints the saved output. If your site requires viewing PDF report pages as the default, you must change the SAVEAS.ACHTML template. Values for operation are:

- **Print**—Prepare PDF format print output and save the specified report page or pages to disk.
- **View**—Save the specified report page or pages as PDF and view the PDF output.

### Getting report data

The GetReportData directive retrieves data for a report in a specified format. Use the following syntax to retrieve report data:

```
http://<webserver>/acweb/<reportserver>/<folder>/
<file>?GetReportData&format=<format>[&componentID=<ID>]&scalingFac
tor=<scale>&operation=<operation>
```

For information about the general syntax of directives, see “Directives syntax,” earlier in this chapter.

The scalingFactor keyword functions as it does for the GetDynamicData directive, described in “Getting dynamic content of DHTML report components,” earlier in this chapter.

The operation keyword functions as it does for the ViewPage directive, described in “Viewing a page from the navigation bar,” earlier in this chapter.

If you do not specify a componentID, Actuate retrieves the entire report.

### Getting the report style sheet

The GetStyleSheet directive retrieves the style sheet for the report in the specified format. Use the following syntax to retrieve the style sheet:

```
http://<webserver>/acweb/<reportserver>/<folder>/
<file>?GetStyleSheet&format=<format>
```

For information about the general syntax of directives, see “Directives syntax,” earlier in this chapter.

The format keyword specifies the format in which to retrieve the style sheet. Valid values are CSS, for cascading style sheets for DHTML output, and XMLStyle, for style information specified in XML.

### Viewing the table of contents

The ViewTOC directive displays the DHTML report’s table of contents. Use the following syntax to display the table of contents:

```
http://<webserver>/acweb/<reportserver>/<folder>/
<file>?ViewTOC&format=<format>&componentID=<ID>
```

For information about the general syntax of directives, see “Directives syntax,” earlier in this chapter.

The format keyword specifies the format of the table of contents. XMLDisplay is currently the only supported format.

The componentID keyword specifies the component to use as the root of the table of contents. 0 is the default component ID.

Users generally display a table of contents for a DHTML report by clicking on the TOC icon in the navigation bar.

### **Viewing the navigation bar**

The ViewNavigation directive displays the navigation bar, used for viewing DHTML reports. Use the following syntax to display the navigation bar:

```
http://<webserver>/acweb/<reportserver>/<folder>/  
    <file>?ViewNavigation&page=<page>
```

For information about the general syntax of directives, see “Directives syntax,” earlier in this chapter.

The page keyword specifies the DHTML report page to display.

The navigation bar displays when a user clicks on a DHTML report document in a report encyclopedia folder.

### **Viewing reports according to user preferences**

The ViewDefault directive displays the report according to the user’s preferences and privileges. Use the following syntax to display the report according to the user’s settings:

```
http://<webserver>/acweb/<reportserver>/<folder>/<file>?ViewDefault
```

For information about the general syntax of directives, see “Directives syntax,” earlier in this chapter.

If the user prefers viewing reports using the LRX, but does not have read permission on the report, ReportCast displays the report in DHTML format.

### **Search directives**

This section describes the directives that provide searching capabilities on DHTML reports. For more information about performing searches on DHTML reports using the search request page, see “Searching DHTML reports” in Chapter 1, “Using ReportCast.”

#### **Requesting a search**

The requestSearch directive displays the search request page. Users use the search request page to enter criteria for searching a DHTML report. The syntax is:

```
http://<webserver>/acweb/<reportserver>/<folder>/<file>?requestSearch
```

For information about the general syntax of directives, see “Directives syntax,” earlier in this chapter.

Users display the search request page by clicking the Search link on the navigation bar.



## Submitting a search

The `submitSearch` directive submits a search request and returns the search results page. The search criteria are included in the search form data, contained in the URL.

Users generally submit search requests by filling out the search request page, then clicking the Search Now button.

## Extracting search results

The `extractSearchResults` directive submits a search request and returns the search results in a specified format. The search criteria are included in the search form data, contained in the URL.

Searches are case-sensitive.

## File directives

This section describes the file directives that access report files on the e.Reporting Server.

### Downloading report files

The syntax to use for downloading directives is as follows:

```
http://<webserver>/acweb/<reportserver>/<folder>/<file>
```

Send this directive to download an entire report file from the e.Reporting Server. If you configured the web browser to run the Actuate LRX or if you chose Live Report Extension as your report viewing preference, and you are downloading an Actuate report document (.ROI), ReportCast launches the Actuate LRX to display the report. If you chose Dynamic HTML (DHTML) as your report viewing preference, ReportCast does not download the report. The report displays in your web browser in DHTML format.

### Downloading report document pages on demand

To view a report through the LRX, you can download an entire report document, or download only the pages you need. For large reports, the second

method, called demand paging, saves time and network resources. The syntax to use to download only select pages, is as follows:

`http://<webserver>/acweb/<reportserver>/<folder>/<file>?View`

This directive runs the Actuate LRX, which displays the report in a browser window.

You can use the view directive to view HTML reports (.ROW files). Demand paging does not apply to HTML reports. In HTML reports, ?View simply displays the entire report in the browser. Use the LRX's navigation buttons to go to particular pages in a report, or move to the report's first, next, previous, or last pages.

The view directive is associated with links in the Name column of the table that displays all the report files in a Report Encyclopedia folder. What happens when you use the link depends upon the type of document. For example, in this case, you can only view a document that is a report document but you can run a document that is a report executable. If you hold your cursor over the icon, the status bar displays information about the link.

Display items in the Reports folder

View the report document

Run the report executable

The screenshot shows the Actuate Druid web interface. At the top, there is a navigation bar with 'Documents', 'My Headlines', 'Requests', and 'Administration' tabs. Below this, there are sections for 'Accounting', 'Marketing', and 'Reports'. The 'Reports' section contains a table with two parts: 'Documents You Can View' and 'Reports You Can Run'. The 'Documents You Can View' table lists 'forecast' with two versions: 'Version017' (dated Aug 24, 1999 12:20:37) and 'Version018' (dated Aug 24, 1999 13:05:45). The 'Reports You Can Run' table lists 'forecast' with 'Version001' (dated Aug 17, 1999 17:57:40). Annotations with lines point to the 'Reports' folder, the 'Documents You Can View' table, and the 'Reports You Can Run' table.

Documents You Can View		
forecast	<a href="#">Version017</a>	Aug 24, 1999 12:20:37 (detail)
	<a href="#">Version018</a>	Aug 24, 1999 13:05:45 (detail)

Reports You Can Run		
forecast	<a href="#">Version001</a>	Aug 17, 1999 17:57:40 (detail)

## Specifying file versions

When you issue a directive to download a file, you can also specify a specific version number. For example, the following directive downloads version 2 of `DETAIL.ROI` in the Reports folder on the Paradise e.Reporting Server:

```
http://xanadu/acweb/paradise/reports/detail.roi;2
```

If a folder contains multiple versions of a report and you do not specify a version number, ReportCast downloads the latest version. If you specify a version number that does not exist, ReportCast displays an error message.

## Factory directives

This section describes the ReportCast factory directives. Factory directives generate new reports from report executables (`.ROX`) or report parameter values (`.ROV`). Factory directives also submit requests for execution.

## Using request

The syntax for the `?Request` and `?StdRequest` is as follows:

```
http://<webserver>/acweb/<reportserver>/<folder>/<.rox | .roi | .rov |  
.rov>?Request | ?StdRequest
```

The `?Request` and `?StdRequest` directives generate requester forms that run reports. The requester forms display in the web browser. Use those request forms to fill in request parameters.

Use the `?Request` directive to generate custom requester forms. Use the `?StdRequest` directive to generate the default requester form. If more than one requester form is available, ReportCast displays a list of the forms to choose from, as shown in the following illustration.



Custom requester forms make running reports simpler for your users. For example, suppose that your regional sales departments run a particular generic report on a regular basis. If you use a standard request, each time the sales staff runs the report, they have to fill in the region-specific information on the form. That information is always the same, so filling it in each time can be tedious. You can create a requester form for each region that supplies the appropriate parameter values for that region.

You can create as many custom requester forms as you need, but there must always be a default requester form for ReportCast to use. The type of requester form ReportCast generates depends on the type of file.

Report file type	Form
.ROX	Generates a blank requester form from the .ROX file parameter definitions.
.ROV	Generates a blank requester form using the .ROV file values as defaults.
.ROI	Looks for the .ROV file and uses it to generate a blank requester form. If ReportCast cannot find an .ROV file, it uses an .ROX file.

---

**Report file type****Form**

---

.ROW

Looks for the .ROV file and uses it to generate a blank requester form. If ReportCast cannot find an .ROV file, it uses an .ROX file.

.HTML

Generates a custom HTML requester form for a particular report, depending on a specific .ROX or .ROV file.

---

The following illustration shows the standard requester form for the Byfr report. The Byfr report is in your Actuate4\Devwb\Examples\FirstRpt directory.

The screenshot displays the Actuate druid web interface. At the top, the 'ACTUATE' logo is visible on the left, and the user 'administrator' is logged in. The date is 'Aug 30, 1999'. Navigation tabs include 'Documents', 'My Headlines', 'Requests', and 'Administration'. A 'Submit Request' button is located on the left sidebar. The main content area is titled 'New Request' and contains the following sections:

- Parameter Groups**
- Output Parameters**: A text input field for 'Headline' with '(String)' next to it.
- Request**: A checkbox for 'Wait for results' which is checked, with '(when running now)' next to it.
- Advanced**
- Schedule**: Radio buttons for 'Right Now', 'Once', and 'Recurring'. 'Once' is selected with a date of '08/30/1999' and time of '15:57:59'. 'Recurring' is also visible with a dropdown set to 'Every Day' and time '15:57:59'.
- Priority**: Radio buttons for 'High (800)', 'Medium (500)', 'Low (200)', and 'Other (1-1000)'. 'Medium (500)' is selected. A text input field contains the value '500'.
- Version**: Radio buttons for 'Create new version' (selected) and 'Overwrite existing version'. A checkbox for 'Keep only the latest' is present with a text input field for the number of versions. Below are text input fields for 'Version Name' and 'Output Name' (containing '/Reports/byfr.roi').
- Archive Policy**: Radio buttons for 'Use the archive policy for the distribution folder(s)' (selected), 'Delete objects older than' (with fields for days and hours), and 'Delete objects on' (with date and time fields). A checkbox for 'Archive items before deletion' is also present.
- Notify Channels**: A checkbox for 'News'.

A 'Submit Request' button is located at the bottom left of the form area. The footer of the page reads 'Copyright © 1997-1999 Actuate Corporation'.

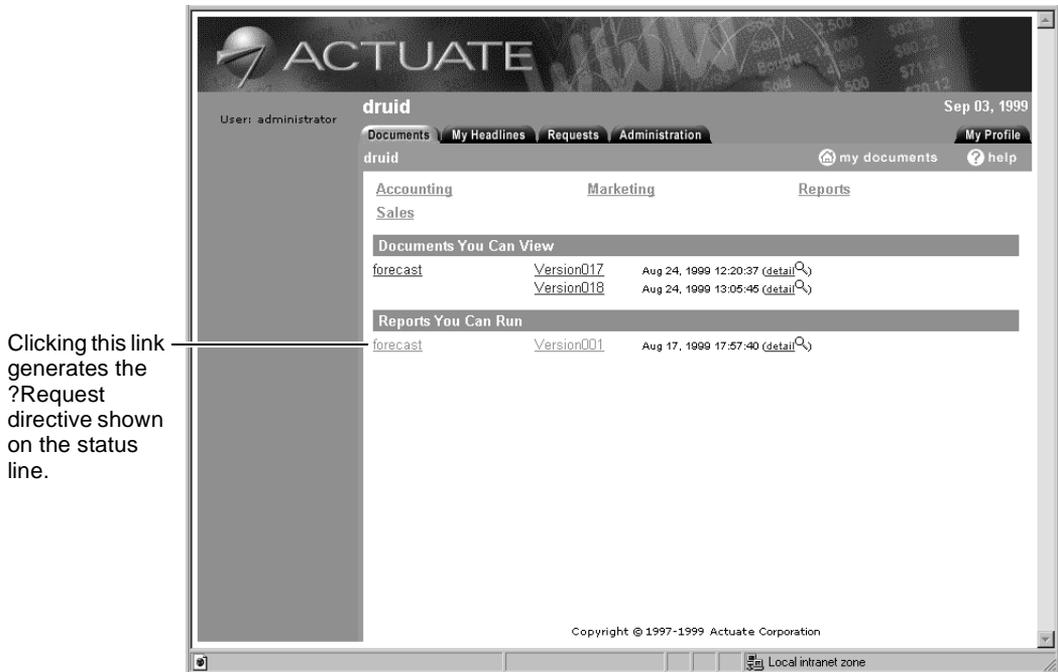
The following illustration shows the VIP custom requester form for the Byfr report. The VIP requester does not wait for the report to finish, runs the report at the highest priority level by default, and overwrites the existing report version rather than creating new versions each time.

The screenshot displays the ACTUATE druid web interface. At the top, the logo 'ACTUATE' is visible on the left, and the date 'Aug 30, 1999' is on the right. Below the logo, the user is identified as 'administrator'. The main navigation bar includes 'Documents', 'My Headlines', 'Requests', and 'Administration'. A 'Submit Request' button is located in the top left corner. The page title is 'New Request', and there is a 'help' link in the top right.

The form is divided into several sections:

- Parameter Groups:** Includes 'Output Parameters' with a 'Headline' text input field (marked '(String)') and a 'Request' section with a 'Wait for results' checkbox (checked) and a '(when running now)' label.
- Advanced:** Contains the 'Schedule' section with three options: 'Right Now' (selected), 'Once' (with date '08/30/1999' and time '15:57:59'), and 'Recurring' (with 'Every Day' selected and time '15:57:59').
- Priority:** Features radio buttons for 'High (800)' (selected), 'Medium (500)', 'Low (200)', and 'Other (1-1000)', along with a text input field containing '500'.
- Version:** Includes radio buttons for 'Create new version' and 'Overwrite existing version' (selected). A 'Keep only the latest' checkbox is present with an empty text input field. Below are 'Version Name' and 'Output Name' text input fields, with the latter containing '/Reports/byfr.roi'.
- Archive Policy:** Features radio buttons for 'Use the archive policy for the distribution folder(s)' (selected), 'Delete objects older than' (with 'days' and 'hours' inputs), and 'Delete objects on' (with date '08/30/1999' and time '15:58:00'). A checkbox for 'Archive items before deletion' is also present.
- Notify Channels:** Includes a 'News' checkbox.

At the bottom left, there is another 'Submit Request' button. At the bottom center, the copyright notice reads 'Copyright © 1997-1999 Actuate Corporation'.



## Using submit

The ?Submit directive submits requests for execution in the e.Reporting Server. Users typically submit requests by clicking a Submit button in a requester form, but you can also generate requests and submit the parameters directly in a submit URL.

The syntax for the ?Submit directive is as follows:

`http://<webserver>/acweb/<reportserver>/<folder>/<rox>?Submit`

## Using a submit directive

Choosing the Submit Request button at the bottom of the form starts the report generation process. When you choose Submit Request, the form submits the parameter values you specified to ReportCast. ReportCast then issues the following directive, which displays a page with information about the report generation request:

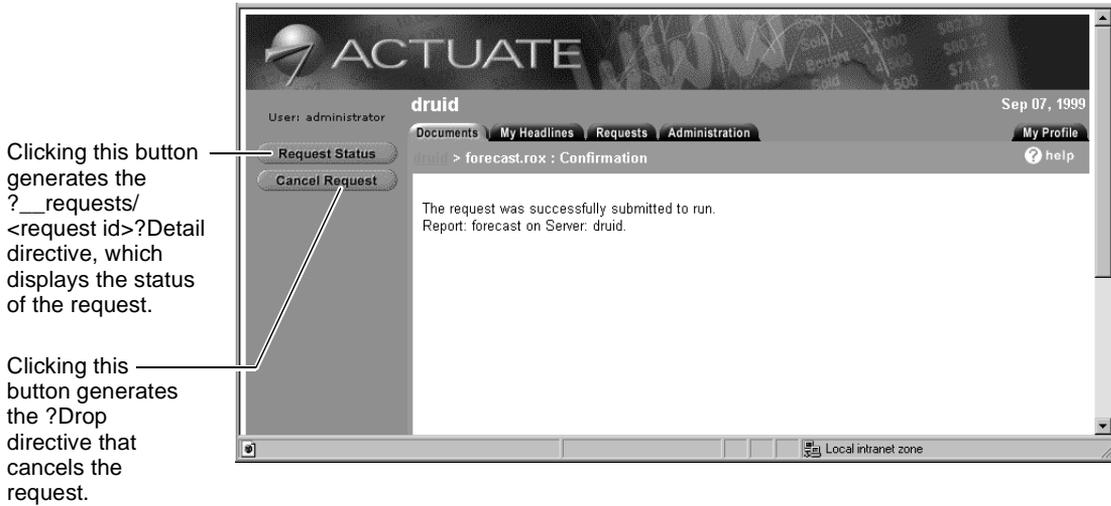
`http://<webserver>/acweb/<reportserver>/<folder>/<.rov> | .<rox>?Submit`

The e.Reporting Server places the resulting .ROX in the user's home folder, if the user has a home folder, or in the same folder as the .ROI.

Always include the ?Submit directive when you create custom requester forms. For example, the following directive sets up a submit button:

```
<form method="POST" action="$(REPORTURL)?Submit">
  <input type="image" value="Submit Request" img src="MySubmitBtn.gif">
```

This directive is typically associated with a button or link. The following illustration shows the results of the ?Submit directive.



## Using a submit URL

You can submit report generation requests directly to ReportCast using submit URLs. The syntax is:

```
<reportserver path>/<folder>/<rox> | <rov>
  ?Submit&variable=value[&variable=value...]
```

### <reportserver path>

The HTTP path for your report server.

### <folder>

The e.Reporting Server folder that contains the report executable from which to generate the report.

### <rox> | <rov>

The report executable or parameter values file from which to generate the report.

### ?Submit

The submit directive.

## &

The HTTP separator that separates the parameter values the e.Reporting Server uses to generate the report.

## variable=value

The variable=value pairs define the report parameters and the values to use to generate the report. Values may never include quotes. Use %20 to represent spaces in any string values you define for the report. Variables are case-sensitive. You must specify date and time values in the format appropriate for your locale.

The following table lists some standard submit parameter variables. You define report parameters as part of your report design in the Developer Workbench. Report parameters are case-sensitive. Refer to them exactly as you define them in the Developer Workbench.

Variable	Description
__accessToGrant	The type of access to grant automatically to those roles that have permission to view the report. Permission is granted by a combination of the __channels and __exclude values. If the output is a secure DHTML report, grant Secure Read access. Otherwise, grant Read access.
BundleRox	True to bundle the data along with the generated .ROI. False otherwise. False is the default value.
Headline	A descriptive tagline for a report. Displays on the Channel Contents page. Use the character string %20 to represent spaces in the headline string.
__scheduleType	Required. The type of schedule: immediate, once, or recurring. Immediate is the default.
__wait	Optional. Wait while the report generates. Can be used only if the report's __scheduleType is immediate.
__onceDate	Required for once schedules. The date on which to run the report, for reports with __scheduleTypes of once. Must be in the appropriate format for your locale. For example, the format for the U.S. (enu) locale is mm/dd/yy. The current date is the default.

Variable	Description
<code>__onceTime</code>	<p>Required for once schedules.</p> <p>The time at which to run the report, for reports with <code>__scheduleTypes</code> of once. Must be in the appropriate format for your locale. For example, the format for the US (enu) locale is hh:mm:ss. The current time is the default.</p>
<code>__schedulePeriod</code>	<p>Required for recurring schedules.</p> <p>How often to run the report, and on which day(s). Choose a day of the week.</p> <p><code>__schedulePeriod</code> values are Every Day, Weekdays, Mondays, Tuesdays, Wednesdays, Thursdays, Fridays, Saturdays, Sundays, First Day of the Month, Last Day of the Month. All values are case-sensitive.</p> <p>Every Day or Weekdays. Set only if <code>__scheduleType</code> is recurring.</p>
<code>__recurringTime</code>	<p>Required for recurring schedules.</p> <p>The time at which to run the report. Set only if reports <code>__scheduleTypes</code> is recurring.</p> <p>Must be in the appropriate format for your locale. For example, the format for the U.S. (enu) locale is hh:mm:ss.</p>
<code>__priority</code>	<p>Optional.</p> <p>The importance that generating this report has in the e.Reporting Server priority.</p> <p>An integer value between 1 and 1000, or Other. 200 is low, 500 is medium, 800 is high. 500 is the default.</p> <p>To set <code>__priority</code>, use a radio button group in the HTML requester form.</p>
<code>__priorityValue</code>	<p>Optional.</p> <p>Indicates request priority, if the value of: <code>__priority</code> is not Other</p> <p><code>__priorityValue</code> is an integer between 1 and 1000</p>
<code>__overwrite</code>	<p>Optional.</p> <p>New to create a new version of this report. New is the default.</p> <p>Old otherwise.</p>

Variable	Description
<code>__versionname</code>	Optional. A string value for the new version name of this report.
<code>__outputname</code>	Optional. Name of the output. ROI file for this report. The default is <code>&lt;report&gt;.roi</code> , where <code>&lt;report&gt;</code> is the name of the .ROX file.
<code>__channels</code>	Optional. Name of a channel to notify of this request. You can notify more than one channel.
<code>__exclude</code>	Optional. The name of a role that must not have access to the report. You can exclude more than one role.
<code>__groups</code>	Optional. The name of the group to notify of this request. You can notify more than one group.
<code>__users</code>	Optional. The name of the user to notify of this request. You can notify more than one user.
<code>__wait</code>	Optional. Indicate whether or not to wait for the report to complete, then immediately display the first page. Use only if <code>__scheduleType</code> is set to "immediate". Default is to wait for the report to complete.
<code>__limit</code>	Optional. Indicate whether or not to limit the number of versions of the output files for the current request. Set <code>__wait</code> to "limit" to limit the number of versions. Any other value means that the number of versions is unlimited.
<code>__limitNumber</code>	Optional. The number of versions to which to limit the output files for the current request. Use only if <code>__limit</code> is set to "limit". <code>__limitNumber</code> can be any positive number.

Variable	Description
<code>__archivePolicy</code>	Optional. The archive policy to implement for the objects created as output for the current request. Values are folder, age, and date. Set "folder" to use the archive policy that is already set for the folders to which the output is distributed. Set "age" to delete objects older than a specific time period. Set "date" to delete objects on a specific date.
<code>__ageDays</code>	Optional. Use with <code>__ageHours</code> to determine how long output objects exist before they are deleted. Use only if <code>__archivePolicy</code> is set to "age". <code>__ageDays</code> can be any positive number.
<code>__ageHours</code>	Optional. Use with <code>__ageDays</code> to determine how long output objects exist before they are deleted. Use only if <code>__archivePolicy</code> is set to "age". <code>__ageHours</code> can be any positive number.
<code>__dateToDelete</code>	Optional. The date on which to delete the output objects of the current request. Use only if <code>__archivePolicy</code> is set to "date". <code>__dateToDelete</code> must be a date in a locale-specific format. The default format is "mm/dd/yyyy".
<code>__timeToDelete</code>	Optional. The time on the date specified by <code>__dateToDelete</code> on which to delete the output objects of the current request. Use only if <code>__archivePolicy</code> is set to "date". <code>__timeToDelete</code> must be a time in a locale-specific format. The default format is "hh:mm:ss".
<code>__archiveBeforeDelete</code>	Optional. Indicate whether or not to archive the output objects of the current request before deleting them, according to <code>__archivePolicy</code> 's setting. This has no effect if <code>__archivePolicy</code> is set to "folder". Set to TRUE to archive objects before deleting them. The default value is FALSE.

For example:

```
http://mustique/acweb/caligari/Test/
MoSales.rox?Submit&Headline=Monthly%20Sales%20Report&__scheduleTy
pe=recurring&__schedulePeriod=Last%20Day%20of%20the%20Month&__re
curringTime=01:30:00&__users=Admin&__groups=Sales&__groups=Mktg
```

This example submits the report executable MOSALES.ROX to the e.Reporting Server, Caligari. Its headline is Monthly Sales Report. It is on a recurring schedule to run on the last day of each month at 1:30 am. The user Admin and groups Sales and Mktg will be notified of the report run.

## Using status directives

Status directives retrieve status about specific report-generation requests, and about all report-generation requests handled by the factory on the specified e.Reporting Server.

### Getting status of a report-generation request

The syntax to initiate a request for status of a report-generation request is as follows:

```
http://<webserver>/acweb/<reportserver>/__requests/<request id>?detail
```

This directive displays a page that shows detailed status information about the report-generation request. Typically, you do not enter this URL directly. Instead, you click a link that generates this directive.

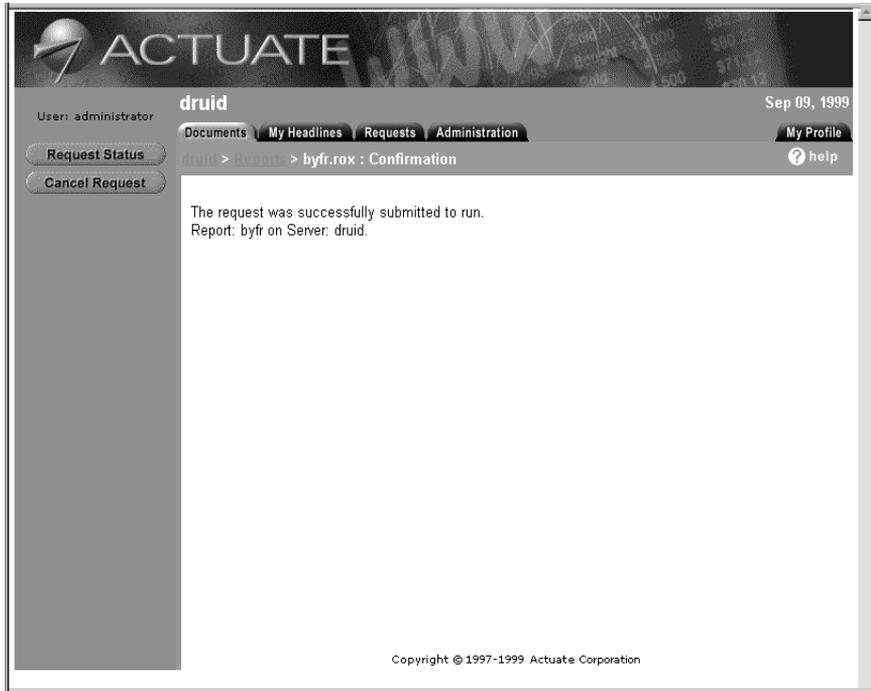
### Cancelling a report-generation status request

The syntax to cancel a request for status of a report-generation request is as follows:

```
http://<webserver>/acweb/<reportserver>/__requests/<request id>?drop
```

This directive cancels the request for status of a report-generation request just submitted. Typically, you do not enter this URL directly. Instead, you click a link that generates this directive, such as the Cancel Request button on the Request Successfully Submitted page.

The following illustration shows a status page for a report-generation request. At the bottom of that page are links you can select to cancel the report-generation job, get more status information, or view the report when it is generated.



## Deleting status for a request

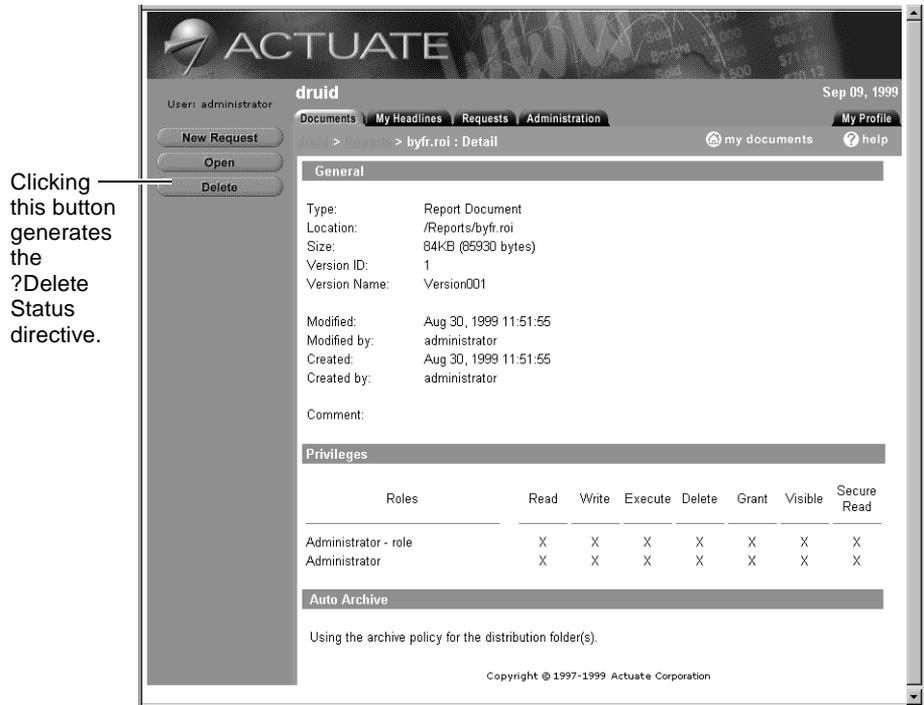
The syntax to delete the request for status of a report-generation request is as follows:

```
http://<webserver>/acweb/<reportserver>/_ _requests/<request id>/  
?DeleteStatus
```

This directive deletes any status the current user has for the request specified by <request id>. ReportCast creates status information automatically as a report generates.

For example, if you generate a particular report or set of reports, ReportCast creates status for each of them. The status is available from a report's details page. When you view the list of completed requests on your report server, you access the status by clicking on the report's Details link. The Details page contains a Delete Notice button. Click that button to delete the status for that report.

The following illustration shows a report's Details page and the Delete Notice button.



## Using information directives

Information directives provide detail about files, folders, and requests. The syntax for the ?Detail directive is as follows:

`http://<webserver>/acweb/<reportserver>/<folder>/<file>?Detail`

?Detail provides precise information about a Report Encyclopedia object. The information provided is usually too extensive and complex to display in ReportCast's various summary views. The previous illustration of Active Requests Details on Mustique shows detail information for a .ROX file.

Information includes:

- All operations allowed on that object
- User privileges for accessing the object

ReportCast uses a format that is similar to the Navigator's property pages format to display the information. For more information about the Navigator, see Chapter 1, "Understanding the Administrator Desktop," in *Administering the Report Encyclopedia*.

## Getting information about factory requests

You can get information about factory requests by using the status directive URLs. The syntax is:

`http://<webserver>/acweb/<reportserver>/<status directive>`

The following table lists the status directives.

Directive	Displays ...
<code>__scheduled</code>	Factory requests currently scheduled
<code>__active</code>	Factory requests currently active
<code>__completed</code>	Factory requests completed

These directives are associated with the Scheduled, Active, and Completed links on the HTML page that displays the contents of a folder.

---

## About Channel directives

Use channel directives to administer, view the contents of, subscribe to, and unsubscribe from ReportCast channels. For more information about channel directives, see “Working with ReportCast channels,” in Chapter 1, “Using ReportCast.”

The syntax of channel directives is:

`http://<webserver>/acweb/<reportserver>/<channel directive>`

The following table lists the channel directives.

Directive	Description
<code>__channels?admin</code>	Displays the channel administration page
<code>__completed</code>	Displays the list of channels to which the user is subscribed
<code>__channels/&lt;channel&gt;</code>	Lists channel contents <channel>
<code>__channels?retrieveForm</code>	Displays the subscription form for available channels
<code>__channels?submitForm</code>	Submits the subscription information

Directive	Description
<code>__channels?retrieveCreation</code>	Displays a form to create a new channel Available to administrators only
<code>__channels?submitCreation</code>	Submits the new channel creation form Available to administrators only
<code>__channels/channel?drop</code>	Removes an existing channel Available to administrators only
<code>__channels/channel?retrieveDetails</code>	Displays channel attributes modification form Available to administrators only
<code>__channels/channel?submitDetails</code>	Submits the channel details modification form Available to administrators only

## Listing ReportCast channels

The syntax of the channel directive is:

`http://<webserver>/acweb/<reportserver>__completed`

This directive lists all available ReportCast channels. Channels to which the user already subscribes display in a different color than those to which the user does not subscribe. Users with administrator privileges can customize this list using the ReportCast templates.

## Administering ReportCast channels

The syntax of the available administrative directives are:

`http://<webserver>/acweb/<reportserver>/__channels?admin`

`http://<webserver>/acweb/<reportserver>/__channels?retrieveCreation`

`http://<webserver>/acweb/<reportserver>/__channels?submitCreation`

`http://<webserver>/acweb/<reportserver>/__channels/  
<channel>?retrieveDetails`

`http://<webserver>/acweb/<reportserver>/__channels/  
<channel>?submitDetails`

`http://<webserver>/acweb/<reportserver>/__channels/<channel>?drop`

These directives allow users with administrator privileges to display the channel administration page, add new channels, modify existing channels, and drop existing channels.

For example, to display the channel administration page for a ReportCast service running on Mustique's web server, use the following URL:

```
http://mustique/acweb/mustique/___channels?admin
```

To drop a channel called prodmkt, use the following URL:

```
http://mustique/acweb/mustique/___channels/prodmkt?drop
```

## Displaying channel contents

The syntax of the directive is:

```
http://<webserver>/acweb/<reportserver>/___channels/<channel>
```

This directive displays the channel contents <channel>. The contents are a list of completed requests. Some channels may have restricted access. If a particular user is not allowed to access a particular channel, an error message displays in the web browser.

## Creating channels using channel URLs

You can create new channels or change existing channels directly using channel URLs. The syntax is:

```
<reportserver path>/  
___channels?submitCreation&variable=value[&variable=value...]
```

or:

```
<reportserver path>/___channels/  
<channel>?submitDetails&variable=value[&variable=value...]
```

### <reportserver path>

The HTTP path for your e.Reporting Server.

### \_\_\_channels?submitCreation

The channels directive followed by the submitCreation command.

### &

The URL separator character. Separates the parameter values that the e.Reporting Server uses to create or modify the channel.

### variable=value

Specifies the channel parameters and the values to use to create or modify the channel properties. Values can never include quotes. Use %20 to represent spaces in any string values you define for the channel, such as the channel

description. Specify numbers and times in the format appropriate for your locale.

The following table lists the channel parameter variables.

<b>Variable</b>	<b>Description</b>
<code>__ChannelName</code>	Required for channel creation. Names the new channel or the channel to modify.
<code>__Comment</code>	Optional. Describes the channel.
<code>__Expiration</code>	Required. Number of days before item expires from the channel.
<code>__LargeIconURL</code>	Optional. URL to a 32x32-pixel icon that represents the channel.
<code>__PollingInterval</code>	Required. Refresh frequency, in seconds, of the channel contents page.
<code>__Roles</code>	Optional. Names a role that has permission to subscribe to this channel. More than one role can subscribe to a channel.
<code>__SmallIconURL</code>	Optional. URL to a 20x20 pixel icon that represents the channel.

For example:

```
http://mustique/acweb/caligari/  
__channels?submitCreation&__ChannelName=News&__Comment=News%20  
you%20can%20use.&__Expiration=3&__PollingInterval=30&__Roles=All
```

This example creates a new channel called News on the report server Caligari. The channel description is News you can use. Items expire out of the channel when they are three days old. The channel checks for new items every 30 seconds. Users with the role All can subscribe to the channel.

## Subscribing to channels using channel URLs

You can subscribe to channels directly using channel URLs. The syntax is:

<reportserver path>/  
\_\_channels?submitForm&variable=value[&variable=value...]

**<reportserver path>**

The HTTP path for your e.Reporting Server.

**\_\_channels?submitForm**

The channels directive, followed by the submitForm command.

**&**

The URL separator character. This character separates the parameter values that the e.Reporting Server uses to create or modify the channel.

**variable=value**

Specifies channel parameters and the values used to subscribe to the channel

The following table lists the channel subscription variables.

---

Variable	Description
channels	Optional. Names of the channel to which to subscribe. You can subscribe to more than one channel.

---

For example:

`http://mustique/acweb/caligari/  
__channels?submitForm&channels=News&channels=Sales`

This example subscribes the user to the News and Sales channels on the Caligari report server.

---

## Examples of ReportCast directives

The following section provides several examples of directives you enter in the browser's Location or Address box.

The following directive displays the Paradise e.Reporting Server contents. Actserv is the name of the machine on which the web server and ReportCast are installed:

`http://actserv/acweb/paradise/`

The following directive displays the contents, sorted by type, of the Reports folder in the Paradise e.Reporting Server:

`http://actserv/acweb/paradise/reports/?sort=Type`

The following directive displays only the report files excluding subfolders, sorted by type, in the Reports folder on the Paradise e.Reporting Server:

<http://actserv/acweb/paradise/reports/?sort=Type&files>

The following directive displays the request form for running the DETAIL.ROX report:

<http://actserv/acweb/paradise/reports/detail.rox?Request>

The following directive submits the request form for running the DETAIL.ROX report:

<http://actserv/acweb/paradise/reports/detail.rox?Submit>

The following directive displays all the scheduled factory requests on the Paradise e.Reporting Server:

[http://actserv/acweb/paradise/\\_ \\_scheduled](http://actserv/acweb/paradise/_ _scheduled)

The following directive displays a list of all the channels to which the user is subscribed. This same directive displays the contents of the user's completed folder:

[http://actserv/acweb/paradise/\\_ \\_completed](http://actserv/acweb/paradise/_ _completed)

The following directive displays the contents of channel Reports. The contents take the form of a list of the completed requests available for viewing:

[http://actserv/acweb/paradise/\\_ \\_channels/Reports](http://actserv/acweb/paradise/_ _channels/Reports)

The following directive displays the channel subscription form:

[http://actserv/acweb/paradise/\\_ \\_channels?retrieveDetails](http://actserv/acweb/paradise/_ _channels?retrieveDetails)

The following directive submits the channel subscription form:

[http://actserv/acweb/paradise/\\_ \\_channels?submitDetails](http://actserv/acweb/paradise/_ _channels?submitDetails)



# ReportCast scripting language reference

This chapter contains the following topics:

- About the ReportCast scripting language
- ReportCast scripting language reference
- ReportCast scripting language variables

---

## About the ReportCast scripting language

ReportCast scripting language commands tell ReportCast how to fill in information when it generates web pages from the template files. ReportCast templates determine the design and presentation of generated web pages. The templates contain a mixture of normal HTML tags and Actuate extensions. The Actuate extensions are the ReportCast scripting language.

The ReportCast scripting language allows you to create custom reporting applications and integrate them with web applications. You can create sets of templates for different audiences and uses, such as:

- Custom templates for specific departments, such as Sales or Accounting
- Internationalized templates for generating web pages in a particular language, such as Japanese, French, or English

The elements of the ReportCast scripting language are:

- **Commands.** Commands tell ReportCast how to insert information when it generates web pages. Commands are case insensitive and must appear in an HTML comment that starts with #Actuate. Each scripting language command must start at the beginning of the line in the template file. Do not place spaces or tabs before the HTML comment containing the scripting language command.
- **Variables.** ReportCast substitutes variable names with real data when it generates web pages. Variable names are case insensitive and may appear anywhere in a line.

---

## ReportCast scripting language reference

This section presents an alphabetical listing of the ReportCast scripting language commands and their syntax. Each command entry includes a general description of the command and a summary of its options.

Scripting language commands are of the following types:

- **Sequential.** These are commands that are executed in sequence, from beginning to end, as ReportCast encounters them in a template.
- **Iteration.** These are commands that loop through lists of objects. Objects include files, folders, properties, and requests. Iteration commands provide some of the same features as the Developer Workbench. You can specify the equivalents of Before, After, Content, and IfEmpty sections of reports.
- **Selection.** These commands provide structures for case statements and conditional statements.

Scripting language commands use the following syntax:

```
<!-- #Actuate <directive> [options] -->
```

### **#Actuate**

- Required.
- Tells ReportCast that this is a command.

### **<directive>**

- Required.
- Tells ReportCast what to do.

### **[options]**

- Optional.
- Lists one or more options for the control directive.

The following table lists the scripting language commands.

<b>Command</b>	<b>Description</b>
List	Iterate through the contents of a list. This command is the most complicated. ReportCast provides many types of lists.
If	Conditionally includes a particular HTML fragment.
Include	Includes another scripting language file.
With	Works with a specified web object.
Set	Sets the value of a scripting language variable.

---

## List command

Loop through the contents of a list.

### Syntax

```
list keyword [options]
```

```
    ifempty  
    before  
    content  
    after  
end list
```

## Description

List command provides the ability to loop through a set of lists defined by ReportCast. The ifempty, before, content, and after sections are optional. You can define sections of lists using these optional keywords. Lists begin with list keyword and end with end list. You can define sections of lists using the optional ifempty, before, content, and after keywords. If they are not present, anything between the list and end list keywords is considered to be the content section. Each section begins with its keyword, and ends at the next keyword.

The following table describes each list element.

List element	Description
keyword	Required. Specifies the type of list to process.
ifempty	Optional. Specifies an action to take if the list is empty.
before	Optional. Specifies content to appear before the main list section. Similar to Developer Workbench's Before slot.
content	Optional. Specifies content to appear in the main list section.
after	Optional. Specifies content to appear after the main list section. Similar to Developer Workbench's After slot.
end list	Required. Specifies the end of the list.

## List keywords

List keywords specify the type of list to process. The next sections describe the List command's keywords in more detail. The following table describes the List keywords.

Keyword	Description
Files	Lists the files in the current Report Encyclopedia folder.
Versions	Lists the versions, if any, for the current file.

<b>Keyword</b>	<b>Description</b>
Folders	Lists the folders in the current Report Encyclopedia folder.
Items	Lists the folders and files in the current Report Encyclopedia folder.
RequestForms	Lists the available custom and standard request pages that run the report associated with the current .ROI, .ROW, .ROX, or .ROV file.
Requests	Lists all requests.
ParamGroups	Lists all parameter groups for a request.
Parameters	Lists all parameters for a request or group.
Values	Lists the values for a submitted request.
Status	Lists the status items for a request.
Members	Lists all members on the current e.Reporting Server. Members include all users, roles, and groups.
Grants	Lists all members that have explicitly been granted permissions for the current item.
Privileges	Lists the privileges each user and role has for the current item.
Notifications	Lists the users and groups to notify when the current request completes.
Schedules	Lists the string descriptions of schedules for the current request.
PrinterProps	Lists the properties of the printer being used for the current request.
Channels	Lists the channels available on the report server or the channels to which the current user is subscribed.
ReportServers	Lists the report servers to which ReportCast has open connections.
Expirationpolicies	Lists the expiration policies for the current file.
Pathlink	Lists the path links that apply to the current folder, file, or .ROD.
Tocitems	Lists the table of contents items to display.
Criteria	Lists the search criteria that were submitted as part of a search request.

Keyword	Description
Resulthits	Lists the search result hits returned by the view process.
Resultfields	Lists the fields that make up the search results.

---

## List files command

Lists the objects in the current Report Encyclopedia folder.

### Syntax

list files [(type [, type...])] [sort by field [desc]]

### Description

List files lists Report Encyclopedia objects in the current encyclopedia folder. You can list all files, or just files of a particular type. You can specify the sort order for the files, or use the sort order specified in the URL directive that generates the file list web page. This command lists only the files in the current folder. It does not list the contents of other folders within the current folder.

### Example

The following example shows how to list the files in a Report Encyclopedia folder:

```
<!-- Start the list -->
<!-- #Actuate list files -->
<!-- Define the list heading -->
<!-- #Actuate before -->
<TABLE>
<TR>
<TH>Name</TH> <TH>Owner</TH> <TH>Date</TH>
<!-- Here is the main list content -->
<!-- #Actuate content -->
<TR>
<TD> $(FileName) </TD>
<TD>$(OwnerName)</TD>
<TD>$(Created)</TD>
</TR>
<!-- Now, define the end of the list -->
</TABLE>
```

```

<!-- Here's what to do if the list is empty -->
<!-- #Actuate ifempty -->
This folder does not contain any report items.
<P>
<!-- #Actuate end list -->

```

## List files type clause

Specify the kinds of files to list. File types are indicated by the suffix of the file in the Report Encyclopedia, for example, .ROW, .ROI, .ROX. Types are case insensitive. You can specify wildcard characters in the type clause.

Wildcard	Matches
?	Any one character
*	Any number of characters
...	All files not previously matched by a preceding pattern

### Example

The following example lists all the report executables in the current folder:

```
<-- #Actuate list files (ROX)
```

## List files sort by clause

Specifies how to sort the files in the list.

### Description

Takes the sort order from the URL directive, by default. The default sort direction is ascending. You can override the default sort order by specifying the sort by clause in your list files command. Use the desc keyword to change the direction of the sort to descending. You can sort by only one field at a time. ReportCast cannot sort by multiple fields.

The following table lists the available sort fields.

Sort field	Description
Name	Sorts by file name and version number.
Type	Sorts by file type.
Owner	Sorts by file owner name.
Date	Sorts by file creation date.
Size	Sorts by file size.

## Example

The following example lists all report executables, sorted by creation date, in ascending order:

```
<!-- #Actuate list files (ROX) sort by date
```

---

## List versions command

List the available versions of the current file.

### Syntax

```
list versions
```

### Description

Each time the report runs, the e.Reporting Server creates a new version of the report file unless you specify otherwise. If a user attempts to view a version of a file in Dynamic HTML (DHTML) that is no longer available, list versions lists the available versions of that file.

List versions is available from:

- The file versions page, named FILEVERSIONS.ACHTML
- The file details page, named FILEDETAIL.ACHTML

### Example

```
<!-- #Actuate list versions
```

---

## List folders command

List all the folders in the current Report Encyclopedia folder.

### Syntax

```
list folders [sort by field [desc]]
```

## Description

List folders lists all folders in the current Report Encyclopedia folder. You can sort the list by specific fields. You can also specify whether the sort should be ascending or descending. The default sort order is ascending.

The following example lists all folders in the current Report Encyclopedia folder, sorted by owner, in descending order:

```
<--! #Actuate list folders sort by owner desc -->
```

List folders is available from the folder listing page, named FOLDERLIST.ACHTML.

## See also

List files sort by clause.

---

## List items command

Lists the contents of the current Report Encyclopedia folder.

## Syntax

```
list items [sort by field [desc]]
```

## Example

The following example lists all the items in the current Report Encyclopedia folder, using the default sort order. The default sort order is ascending:

```
<--! #Actuate list items -->
```

List items is available from the folder listing page, named FOLDERLIST.ACHTML.

## See also

List files sort by clause.

---

## List requestforms command

Lists the available requester forms.

## Syntax

list requestforms

## Description

Lists the requester forms available to run the report specified by the URL directive or associated with the file name in the URL directive. List requestforms is available only from ReportCast's Requester Menu page.

List requestforms is available from the requester menu page, named REQUESTERMENU.ACHTML.

---

## List requests

Lists the requests on the current report server or for the current user.

## Syntax

list requests

list {active | scheduled | completed} requests [sort by {name | date | status}]

## Description

ReportCast uses the list requests command to generate a list of requests when it receives a URL directive for the `__active`, `__completed`, or `__scheduled` request folders. List requests is available only from ReportCast's Request Folder page.

The first form lists all requests on the current e.Reporting Server.

The second form lists requests for the current user. Use this form to create a web page other than a request folder page, for example, to display the user's completed requests on an e.Reporting Server's main page, so that their completed reports are immediately available.

List requests is available from the request list page, named REQUESTLIST.ACHTML.

---

# List paramgroups, list parameters commands

Use these commands to create custom requester forms.

## Syntax

list paramgroups

list parameters

## Description

List paramgroups and list parameters provide a requester template for generating a custom requester form.

When users send a request to run a particular report, ReportCast generates a requester form. The requester form lists the available report parameters and allows users to fill in the parameters they are interested in. For example, users can run a standard sales report, filling in parameters such as region, sales office, quarter, and so on.

If many users from particular regions or departments plan to run the standard report regularly, you can provide custom requester forms that provide default values for parameters the users would otherwise have to fill in by hand. For example, you can create a requester form for the Western Region sales group, another for the Eastern Region sales group, and still other requester forms for the European, Asian, and South American sales groups.

If a particular parameter group name is empty—the value is ""—its parameters were defined at the top level in the Developers WorkBench Parameters dialog box. They do not really belong to a particular parameter group.

The following illustration shows a standard requester form for a report. The form lists all the parameters required for the report, including channels to notify when the report finishes running. The parameters are listed by group.

ACTUATE

User: administrator **druid** Aug 30, 1999

Documents My Headlines Requests Administration My Profile help

Submit Request New Request

**Parameter Groups**

**Output Parameters**

Headline  (String)

**Request**

Wait for results  (when running now)

**Advanced**

**Schedule**

Right Now

Once  (mm/dd/yyyy) at  (hh:mm:ss)

Recurring  at  (hh:mm:ss)

**Priority**

High (800)  Medium (500)  Low (200)  Other (1-1000)

**Version**

Create new version  Keep only the latest  version(s)

Overwrite existing version

Version Name:

Output Name:

**Archive Policy**

Use the archive policy for the distribution folder(s)

Delete objects older than  days  hours

Delete objects on  (mm/dd/yyyy) at  (hh:mm:ss)

Archive items before deletion (only applies if not using archive policy for distribution folder)

**Notify Channels**

News

Submit Request

Copyright © 1997-1999 Actuate Corporation

The requester template, REQUEST.ACHTML, can list parameters by group or by name.

Use list parameters to list the parameters alphabetically. Use list paramgroups to list the parameters by group. List parameters can be nested. List paramgroups cannot be nested.

List parmgroups and list parameters are available from the following templates:

- Standard requester page, named REQUEST.ACHTML
- Request confirmation page, named CONFIRM.ACHTML
- Request details page, named REQUESTDETAILS.ACHTML
- Request status page, named STATUS.ACHTML
- Cancelled request confirmation page, named CANCEL.ACHTML

---

## List values command

Generates a request confirmation page.

### Syntax

list values [setonly]

### Description

Lists the values that the user set for request parameters. For more information about request parameters, see “List parmgroups, list parameters commands,” earlier in this chapter.

The setonly option lists only those parameters for which the user entered a value. You can use this with the list parmgroups command to group the values by parameter groups.

List values is available from the following templates:

- Request confirmation page, named CONFIRM.ACHTML
- Request details page, named REQUESTDETAIL.ACHTML
- Request status page, named STATUS.ACHTML
- Cancelled request confirmation page, named CANCEL.ACHTML

List values can be nested inside the list parmgroups command.

### Example

The following example iterates through a values list and displays the values on a request confirmation page. This example is taken from the CONFIRM.ACHTML template:

```

<!-- #Actuate list values -->
<!-- #Actuate content -->
<table>

<!-- Display the values heading -->

    <tr>
<td>&nbsp;</td>
<td><font size="2" face="Arial">Description</font></td>
<td>&nbsp;</td>

<!-- If the value is null, display a space -->
<!-- #Actuate if Value = "" then -->
<td><font size="2" face="Arial">&nbsp;</font></td>

<!-- Otherwise, display the value -->
<!-- #Actuate else -->
<td><font size="2" face="Arial">$(Value)</font></td>

<!-- #Actuate end if -->
</tr>
</table>

<!-- #Actuate end list -->

```

---

## List status command

Generates a request status page.

### Syntax

list status

### Description

The request status page lists the status items available for the current request. List status is available from the following templates:

- Confirmation page, named CONFIRM.ACHTML
- Request status page, named STATUS.ACHTML
- Cancelled request confirmation page, named CANCEL.ACHTML
- Request details page, named REQUESTDETAIL.ACHTML

## Example

The following example iterates through a list of status items for reports on a particular e.Reporting Server and displays them on a status page. This example is taken from the STATUS.ACHTML template:

```
The following is the status of running $(ReportName) on $(Server):
<!-- #Actuate list status -->
<!-- #Actuate before -->
<center>
<table border=0 bgcolor=black>
<thead valign=top>
<tr bgcolor="#804080">
<th>Time<th>Status
<tbody>
<!-- #Actuate content -->
<tr bgcolor="E0E0E0"><td>$(DateAndTime)<td>$(StatusText)</tr>
<!-- #Actuate after -->
</table>
</center>
<!-- #Actuate end list -->
```

---

## List members command

Lists every member on a e.Reporting Server.

### Syntax

```
list members { (users | roles | groups [...])}
```

### Description

The term member refers to any user, role, or group on an e.Reporting Server. You can use list members to generate a pull-down list of users to notify when a particular request completes. The default is to list all members or specify to include only certain types of members, for example, list only users, or roles, or groups. The list is sorted by name.

List members is available from all templates except for the admin page, called ADMIN.ACHTML. It cannot be nested inside another list members command or inside the list grants or list notifications commands.

## Example

The following example iterates through a list of the roles that are members on the current e.Reporting Server, and sets up an input checkbox. This example is based on the CHANNELDETAIL.ACHTML template:

```
<!-- #Actuate list members (roles) -->
  <!-- #Actuate if MemberName <> "Administrator" and MemberName <>
  "All" then -->
    <input type="checkbox" name="__Roles" value="$(MemberName)">
    $(MemberName)
  <!-- #Actuate end if --> <!-- MemberName -->
<!-- #Actuate end list --> <!-- members -->
```

---

## List grants command

Lists all members that are explicitly granted permissions on the current item, either a Report Encyclopedia object or a channel.

### Syntax

```
list grants [all] {(users | roles | groups [,...])}
```

### Description

List grants lists those privileges on an object that are explicitly granted to the current user or role. It does not include those privileges that the current user or role inherited from a role to which they belong. This means that the set of users returned by list grants list does not include the set of users that inherit all their privileges on the object from their roles.

List grants is available from the following templates:

- File detail page, named FILEDETAIL.ACHTML
- Channel detail page, named CHANNELDETAIL.ACHTML
- Request page, named REQUEST.ACHTML

List grants can be nested inside the list versions or list items commands or at the top level in a template for a folder, encyclopedia item, or channel. This command is useful if you want to allow or disallow the current member to execute a report depending upon whether they have privileges on an item.

Use the All option to list all members, not only those that have privileges on the current folder, file, or channel. If the current user does not have

administrator privileges, the list of privileges returned applies only to the current user and the roles to which that user belongs.

For more information about members, see “List members command,” earlier in this chapter.

## Example

The following example lists the roles that are granted privileges on the current item. The example is based on the REQUEST.ACHTML template:

```
<!-- #Actuate list grants (roles) -->
<!-- #Actuate if MemberName <> "Administrator" and MemberName <> "All"
then -->
<!-- #Actuate Set RoleFound = "True" -->
<!-- #Actuate end if -->
<!-- #Actuate end list -->
```

---

## List privileges command

Lists all the privileges granted to the current user or role for the current encyclopedia item or channel.

### Syntax

list privileges

### Description

Lists all the privileges available on the current e.Reporting Server. Use list privileges within a list grants command to list each of the privileges granted to the current user or role for the current Report Encyclopedia item or channel. If you use list privileges in the context of a Report Encyclopedia item, the returned list does not include privileges inherited from a role to which the current user or role belongs.

List privileges is available from the following templates:

- The channel details page, named CHANNELDETAIL.ACHTML
- The file details page, named FILEDETAIL.ACHTML

---

## List notifications command

Creates a list of users and groups to notify when the current request is complete. The list is sorted by name.

### Syntax

list notifications

### Description

Supports user notification when a request to run a report completes. The standard requester form, REQUEST.ACHTML, displays the list of available channels. To notify a particular channel when a request completes, users click the appropriate check box. You can also modify the STATUS.ACHTML template to show the list of channels to be notified when the user checks the status of the request.

The list notifications command is available from the following templates:

- The confirmation page, named CONFIRM.ACHTML
- The cancellation page, named CANCEL.ACHTML
- The request details page, named REQUESTDETAIL.ACHTML
- The status page, named STATUS.ACHTML

---

## List schedules command

Lists the schedule associated with the current request.

### Syntax

list schedules

### Description

Supports specifying simple schedules for running reports, such as daily, right now, or once at a specified time and date. The e.Reporting Server's Administrator Desktop also allows you to specify complex schedules, such as twice weekly plus once a month on a specific day. You can view complex

schedules using the list schedules command, even though you cannot specify those commands using ReportCast.

The list schedules command is available from the request details page, named REQUESTDETAILS.ACHTML.

---

## List printerprops command

Lists property names and values for the printer being used for the current request.

### Syntax

```
list printerprops
```

### Description

Provides information about the printer being used to print out the current request. List printerprops is available from the request details page, named REQUESTDETAIL.ACHTML, if the request was either Print or Execute and Print.

---

## List channels command

Lists channels to which the current user is subscribed.

### Syntax

```
list [all] channels
```

### Description

Provides a list of channels to which the user is subscribed. For example, the standard template for the Personal Channel, PERSONALCHCONTENTS.ACHTML, displays the list of channels to which the user is subscribed in the left sidebar. Use the All option to get a list of all available channels on the current e.Reporting Server.

List channels is available from all templates except for ADMIN.ACHTML.

## Example

The following example lists all channels available on the current e.Reporting Server:

```
<!-- #Actuate list all channels -->
```

---

## List reportservers command

Lists the report servers to which ReportCast has open connections.

### Syntax

```
list reportservers
```

### Description

ReportCast opens a connection to a particular e.Reporting Server each time a user accesses an e.Reporting Server through ReportCast. Users with administrative privileges can check the open connections from ReportCast to a particular e.Reporting Server from the ReportCast administration page. Open connections use system resources. If response time is slower than desirable, you can close open connections from the administration page.

List reportservers is available from the admin page, named ADMIN.ACHTML.

---

## List pathlinks command

### Syntax

```
list pathlinks
```

### Description

Creates a list of the path links that apply to the current folder, file, or .ROD. The list is empty if there is no current folder, file, or .ROD. Use this command to find the individual elements of a folder, file, or .ROD's complete path name. For example, list pathlinks allows cascading style sheet properties to control the colors of Report Encyclopedia objects' links on the reporting web page.

List pathlinks is available from the following templates:

- The folder list page, called FOLDERLIST.ACHTML
- The file details page, called FILEDETAIL.ACHTML
- The confirmation page, called CONFIRM.ACHTML
- The request details page, called REQUESTDETAIL.ACHTML
- The status page, called STATUS.ACHTML
- The cancellation page, called CANCEL.ACHTML

## Example

The following example is a fragment from FOLDERLIST.ACHTML. It displays the folder path name with each directory name separated by a small folder icon:

```
<!-- #Actuate list pathlinks -->
<!-- #Actuate ifempty -->
<tr>
  <td valign="bottom" nowrap>
    <table border="0" cellpadding="0" cellspacing="0">
      <tr>
        <td></td>
        <td><font size="3" face="Arial"><strong>$(ServerAlias)</strong>
</font></td>
      </tr>
    </table>
  </td>
</tr>
<!-- #Actuate before -->
<tr>
  <td valign="bottom" nowrap>
    <table border="0" cellpadding="0" cellspacing="0">
      <tr>
        <td></td>
        <td><font size="3" face="Arial"><strong>
<!-- #Actuate content -->
<!-- #Actuate if Hyperlink <> "" then -->
          <a href="$(Hyperlink)">$(DisplayName)</a>&nbsp;
&nbsp;
<!-- #Actuate else -->
          $(DisplayName)
<!-- #Actuate end if -->
<!-- #Actuate after -->
        </strong></font></td>
      </tr>
    </table>
  </td>
</tr>
```

```
</table>
</td>
</tr>
<!-- #Actuate end list -->
```

---

## List expirationpolicies command

### Syntax

list expirationpolicies

### Description

Creates a list of the object aging expiration policies that apply to the current Report Encyclopedia file. The expirationpolicies command is available from the file detail page, FILEDETAIL.ACHTML.

---

## List tocitems command

### Syntax

list tocitems

### Description

Lists the items in the DHTML report's Table of Contents. The list tocitems command is available from the view table of contents page, named VIEWTOC.ACHTML.

---

## List criteria command

### Syntax

list criteria

## Description

Lists the search criteria for a search query. The list criteria command is available from the search results page, named SEARCHRESULTS.ACHTML.

---

## List resulthits command

### Syntax

```
list resulthits
```

### Description

Lists the search result hits returned by the search query. The list resulthits command is available from the search results page, named SEARCHRESULTS.ACHTML.

---

## List resultfields command

### Syntax

```
list resultfields
```

### Description

Lists the fields that are part of the search query's results.

You can call list resultfields within the scope of a particular search result hit. Doing so allows you to access the value and component ID for each field in that result hit.

You can call list resultfields outside the scope of any particular search result hit. Doing so allows you to access only the name of each field contained in the search query's results.

The list resultfields command is available from search results page, named SEARCHRESULTS.ACHTML.

---

## If command

Conditionally includes HTML in the generated web page.

### Syntax

```
if expr then
  elseif expr then
  else
  end if
```

### expr

An expression.

Expressions have the following formats:

```
expr and expr
expr or expr
(expr)
not expr
var = "string"
var <> "string"
var = var
var <> var
```

### Description

Allows template developers to include HTML in their generated web pages based on specific conditions. For example, use the If command to provide a particular format for particular types of files, or hide files the current user is not authorized to see.

---

## Include command

Includes another template file with the current template file.

### Syntax

```
include filename
```

## Description

Including other template files with the current template file provides reusability for your HTML template code. The included template file can include other template files, which may include other template files, and so on. Included template files must be both readable and located in the current search path's directory tree. In other words, they must be local to the report server. You cannot include templates that reside on other file systems or directory structures.

The Include command is useful for including a standard element in several templates. It simplifies maintenance to make changes to one file, rather than to several files. For example, include the standard company logo in the LOGO.ACHTML template file and standard company copyright information and links to help in the COPYRIGHT.ACHTML template.

## Example

The following example includes a template in the current template file:

```
<!-- #Actuate include "stdfooter.achtml" -->
```

---

## With command

Makes the specified object's variables and list statements available.

## Syntax

```
with objectname  
  statements  
end with
```

## Description

Includes a named object's variables and list statements in the current object's scope. Objects have the following values.

Object value	Description
ReportDefinition	Makes available the Report Definition for the current request, which provides the parameter, parameter group lists, and other properties.
InputObject	Makes available the Folder Item for the .ROX file of the request.

---

Object value	Description
OutputObject	Makes available the Folder Item for the .ROI or .ROW of the request.
ValuesObject	Makes available the Folder Item for the .ROV file of the request.

---

## Example

The following example makes the ReportDefinition object available, then iterates through its parameter groups list:

```
<!-- #Actuate with ReportDefinition -->
<!-- #Actuate list paramgroups -->
<!-- #Actuate content -->
<!-- (Actions taken on parameter groups here) -->
<!-- #Actuate end list -->
<!-- #Actuate end with -->
```

---

## Set command

Assigns a value to a variable.

### Syntax

```
set var="string"
```

```
set var1=var2
```

**var, var1, var2**

Unique variable names. Variable names are case-insensitive.

### string

The value of the character string to assign to var.

### Description

Assigned values can be character strings or other variables' values. Once a variable is assigned, it is available within the current template using the following syntax:

```
$(VariableName).
```

## Example

The following example checks whether or not a folder contains items, and if so, sets the variable `FolderHasItems` to `True`. The following example is taken from the `FOLDERLIST.ACHTML` template:

```
<!-- #Actuate list items -->
<!-- #Actuate content -->
<!-- #Actuate if FolderPath<>"/" or ( ItemBaseName<>"$$$TempROVs" and
ItemBaseName<>"$$CUTCOPY" ) then -->
<!-- #Actuate Set FolderHasItems = "True" -->
<!-- #Actuate end if -->
<!-- #Actuate end list -->
```

---

## ReportCast scripting language variables

The ReportCast scripting language variables are placeholders for data within the template files. ReportCast substitutes actual data values for the variables when it generates the web page. The variable substitution process enables creating HTML pages dynamically, depending on the directive sent. Dynamic HTML pages support page generation that reflects the constant data changes on the e.Reporting Server.

ReportCast variables provide current, run-time access to information about a wide variety of topics including the ReportCast services, e.Reporting Server, the available folders and files, schedules, ReportCast channels, requests, privileges, and parameters.

Variables use the following syntax:

`$(variable)`

All variable names are case-insensitive. For example, the following command tells ReportCast to generate a page that displays the name of a report and its owner's name:

The name of the report is `$(ReportName)` and the owner is `$(OwnerName)`.

If the report name is `SalesEastQ497.roi`, and the owner is `SalesMgr`, the generated page displays:

The name of the report is `SalesEastQ497` and the owner is `SalesMgr`.

## ReportCast variables

ReportCast variables return information about the machine running ReportCast, as well as information about ReportCast itself. ReportCast

variables are available on all templates. The following tables list the scripting language variables.

<b>Variable</b>	<b>Purpose</b>
Today	Displays the current date on the machine running ReportCast. Date format is localized.
Now	Displays the current time on the machine running ReportCast. Time format is localized.
AgentVersion	Displays the product name and version of ReportCast.
Host	Displays the name of the machine that is running ReportCast.
TemplateLoadDateAndTime	Displays the date and time that ReportCast's template file cache was last cleared.
StartDateAndTime	Displays the date and time that ReportCast was started.
ElapsedTime	Displays the elapsed time since ReportCast was started.
NumberOfCommands	Displays the number of commands that ReportCast has processed since the last time it was started.
AdminURL	Displays the URL to ReportCast's administration page.
ReportServerName	Displays the report server name that the user specified during login. ReportServerName is available only from the login form, and only if ReportCast is not set up with Basic Authentication.

## Web server variable

The following table lists the ReportCast variable that returns information about the web server on which ReportCast is running. This variable is available on all templates.

<b>Variable</b>	<b>Purpose</b>
WebServerVersion	Displays the product name and version of the web server in which ReportCast is running.

## Web request variables

The following table lists the ReportCast variables that return information about the current command that ReportCast is processing. They are available on all templates.

Variable	Purpose
CommandName	Displays the name of the command appended to the URL.
URL	Displays the full URL sent to ReportCast.
ObjectURL	Displays the URL only for the object that is the target of the URL.
UserName	Displays the name of the current user.
UserAgent	Displays the contents of the User Agent, for example, Mozilla/3.01 WinNT.
RemoteAddress	Displays the dotted-decimal IP address of the browser.
RemoteHost	Displays the fully resolved host name of the browser.
TemplateName	Displays the name of the template without the directory path.
PreviousURL	Displays the URL of the web page previous to the current page.

## e.Reporting Server variables

The following table lists ReportCast variables that return information about the e.Reporting Server named in the current request's URL. These variables are available in all templates except the ReportCast administration page.

Variable	Purpose
Server	Displays the name of the e.Reporting Server.
ServerVersion	Displays the full product name and version of the e.Reporting Server.
ScheduledRequestsURL	Displays the URL for the list of scheduled requests on the e.Reporting Server.
ActiveRequestsURL	Displays the URL for the list of active requests on the e.Reporting Server.
CompletedRequestsURL	Displays the URL for the list of completed requests on the e.Reporting Server.

<b>Variable</b>	<b>Purpose</b>
RequestsURL	Displays the URL for the list of all requests on the e.Reporting Server.
HomeFolderURL	Displays the URL for the listing of the current user's home folder on the e.Reporting Server. If no home folder is defined for the current user, HomeFolderURL has the same value as RootFolderURL.
RootFolderURL	Displays the URL to the listing of the root folder of the e.Reporting Server.
ChannelsURL	Displays the URL to the listing of the current user's channels.
ServerAlias	Displays the descriptive string name of the e.Reporting Server. If there is no descriptive name, ServerAlias is the e.Reporting Server's machine name.
IsAdministrator	True if the current user belongs to the administrator group of the current e.Reporting Server. False otherwise.
ServerFeatureLevel	Indicates the e.Reporting Server level. 0 is None, 1 is Standard, and 2 is Advanced.
ViewFormatPreference	Contains the current user's format preference for viewing reports. DHTML, ROI, or SYSTEM DEFAULT. SYSTEM DEFAULT uses the viewing preference set by the e.Reporting Server administrator.
ViewFormatDefault	Displays the system-wide default set by the e.Reporting Server administrator, either DHTML or ROI.

Note that the value for ServerAlias comes from the ACSERVERALIAS property of the root folder on the e.Reporting Server. Use the Navigator to set the e.Reporting Server's name property, which sets the ACSERVERALIAS value.

## View session variables

The following table lists the ReportCast variables that return information about the current view session. The view session variables are available whenever the user views a DHTML report.

Variable	Purpose
CurrentPage	The number of the DHTML report page currently being viewed.
CurrentTOCParentID	The component ID of the current Table of Contents entry, or node.
TotalPages	The total number of pages in the DHTML report.

## Table of contents item variables

The following table lists the ReportCast variables that return information about the current table of contents item. The table of contents item variables are available whenever the user makes use of a DHTML table of contents.

Variable	Purpose
ComponentID	The current table of contents item's component ID.
Expandable	TRUE if the table of contents item has subitems that appear when a user expands the item by clicking on the item's "+" sign, FALSE otherwise.
Text	The displayable name of the current table of contents item.

## Search criterion variables

The following table lists the ReportCast variables that return information about the current search criterion. The search criterion variables are available whenever the user performs a search on a DHTML report.

Variable	Purpose
ControlName	The name of the control that the current search criterion applies to. For example, "Office name."
ComponentID	The component ID of the control that the current search criterion applies to.

<b>Variable</b>	<b>Purpose</b>
SearchValue	The expression that the control's value must match. For example, "Boston," or ">1000." SearchValue is blank if the user did not enter a search expression for the control, but selected it as part of the search results.
Selected	TRUE if the user selected the control as part of the search results, FALSE otherwise.

## Search result set variables

The following table lists the ReportCast variables that return information about the current search result set. The search result set variables are available whenever the user examines the results returned by a search on a DHTML report.

<b>Variable</b>	<b>Purpose</b>
TotalNumberOfHits	The total number of result hits returned by the search.
NumberOfHitsOnPage	The number of hits displayed on the search results page.
StartHitNumber	The index of the first hit displayed on the search results page. The range is 1–TotalNumberOfHits.
EndHitNumber	The index of the last hit displayed on the search results page. The range is 1–TotalNumberOfHits.

## Search result hit variables

The following table lists the ReportCast variables that return information about the current search result hit. The search result hit variables are available whenever the user examines a specific search result hit.

<b>Variable</b>	<b>Purpose</b>
ComponentID	The current search result hit's component ID.

## Search result field variables

The following table lists the ReportCast variables that return information about the current search result field. The search result field variables are available whenever the user examines a specific search result field.

Variable	Purpose
Value	The displayable value of the current search result field.

## Current folder variables

The following table lists the ReportCast variables that return information about the current folder. The current folder variables are available whenever the user displays a page that describes a folder, for example, the folder listing page FOLDERLIST.ACHTML.

Variable	Purpose
FolderPath	Displays the full path name of the current folder.
FolderName	Displays the name of the current folder.
ParentFolderPath	Displays the full path name of the folder that contains the current folder.
ParentURL	Displays the URL of the parent folder. This variable's value is empty ("" ) if there is no parent folder.
FolderHeader	Displays the description of the folder's contents.
FolderTitle	Displays the title to be used in the HTML <TITLE> tag.
LinkedFolderPath	Displays a series of lines that provide a hierarchical path to the current folder.
LastModified	Displays the date and time that the folder was last modified.

## Folder item variables

The following table lists the ReportCast variables that return information about folder items. Folder items are both folders and the files inside a folder.

These variables are available for each item in a list items, list files, and list folders list, and from the File detail page.

<b>Variable</b>	<b>Purpose</b>
LastModified	Displays the date and time that the item was last modified.
LastModifiedBy	Displays the name of the user who last modified the item.
PathName	Displays the full path name of the item on the e.Reporting Server.
ItemName	Displays the name of the current item.
ItemBaseName	Displays the item's base name, without the file name extension.
ItemPathName	Displays the full path name of the item.
Created	Displays the item creation date and time.
OwnerName	Displays the name of the item's owner.
URL	Displays the URL of the item.
ObjectID	Displays the item's unique object ID.
FileType	Displays the item's file type.
AbbreviatedType	Displays the extension of the file, for example, .ROX, .ROI.
FileSize	Displays the size of the item.
AbbreviatedFileSize	Displays the size of the item, abbreviated according to the localized display format. For example, 12.5MB, 923KB.
FileVersion	Displays the version number of the item.
FileVersionName	Displays the descriptive version name of the file item. If there is no descriptive name, ReportCast displays Version <n>, <n> is the version number.
IsBundledOutput	TRUE if the object is a .ROW file or a third-party report consisting of HTML pages, FALSE otherwise. View bundled files using the ?View or ?ViewDefault directives.
IsExecutable	TRUE for .ROX, .ROV, and third-party report executable files, FALSE otherwise. You can request these files using the ?Request directive.
LatestURL	The URL to the most recent version of the file.

<b>Variable</b>	<b>Purpose</b>
LinkedFolderPath	A series of lines that provide a hierarchical path to the current folder item's properties.
NewItem	True if the current item is the first version of a file to be listed in the list items or list files commands, False otherwise. The first version is not necessarily Version 1. It is simply the first in this particular list. Its position depends on the sorting options used.
NumVersions	Displays the number of versions of the file that correspond to the current item.
Comment	Displays a lengthy description of the file, stored as a property on the e.Reporting Server. Displays on detail pages in the Navigator. Comments can be of any length.

## Current request folder

The following table lists the ReportCast variables that return information about the request folder. These variables are available from the Request folder page or the Channel contents page.

<b>Variable</b>	<b>Purpose</b>
FolderName	Displays the name of the request folder: Scheduled, Active, or Completed.
Entries	Displays the number of requests in the request folder.

## Report definition

The following table lists the ReportCast variables that return information about a report. These variables are available from the Standard Requester page and the Request confirmation page.

<b>Variable</b>	<b>Purpose</b>
ReportName	Displays the name of the report executable, without the file extension.
ReportPath	Displays the complete name of the report executable.
ReportOutputPath	Displays the default location of the output file for the report (path and file name).

<b>Variable</b>	<b>Purpose</b>
LastModified	Displays the date and time the report executable was last modified.
DefaultDate	Provides a default date value (the current date) for the report requester form, in the correct localized format.
DefaultTime	Provides a default time value (the current time) for the report requester form, in the correct localized format.
ReportURL	Displays the URL of the report executable.
CancelURL	Displays the URL for cancelling the current request. Available only from the Request Confirmation page.
ViewURL	Displays the URL for viewing the output of the request. Available only from the Request Confirmation page.
DownloadURL	Displays the URL for downloading the output of the current request. Available only from the Request Confirmation page.
StatusURL	Displays the URL for the status page of the current request. Available only from the Request Confirmation page.
RequestURL	Displays the base URL containing the ID of the current request, without the query portion. RequestURL can be combined with queries in other templates to produce URLs other than the request's URL.

## Request item variables

The following table lists the ReportCast variables that return information about items in a list requests list. These variables are available from the Request status and Request detail pages.

<b>Variable</b>	<b>Purpose</b>
RequestID	Displays the unique ID of the current request.
RequestType	Displays the type of the current report-generation request: Print or Execute.
RequestedBy	Displays the name of the person submitting the request.

<b>Variable</b>	<b>Purpose</b>
RequestStatus	Displays the status of the current request: Scheduled, Active, Completed, or Failed.
OutputName	Displays the base name of the report document that will be generated from the current request.
OutputLocation	Displays the name of the folder containing the request's output file.
OutputPathName	Displays the complete name of the output file, including the full path name and file extension.
ReportName	Displays the name of the requested report executable (.ROX).
ReportPathName	Displays the complete name of the requested report's executable, including the full path name and file extension.
Priority	Displays the numeric priority of the request.
StartTime	Displays the date and time at which the current request started to run.
CompletedTime	Displays the date and time at which the current request completed.
Headline	Displays the request headline. This variable is valid only for successfully completed requests.
ReportURL	Displays the URL of the report executable.
CancelURL	Displays the URL of the command to cancel the request.
ViewURL	Displays the URL for viewing the output of the request. This variable is valid only for successfully completed requests.
DeleteURL	Displays the URL for the command to delete status information for the current request.
StatusURL	Displays the URL of the status page for the current request.
RequestURL	Displays the base URL for the ID of the current request, without the query portion.
DetailURL	Displays the URL for the request's detail page.
ReportVersion	Displays the version number of the requested report executable.
ReportVersionName	Displays the version name of the requested report executable.

<b>Variable</b>	<b>Purpose</b>
CurrentSize	Displays the current size of the output of the request, in bytes.
AbbreviatedCurrentSize	Displays the current size of the output of the request, in KB, abbreviated.
CurrentPages	Displays the current number of pages of the request's output.
ElapsedTime	Displays the time expired since the start of the request.
ProcessGroup	Displays the name of the process group running the request.
OutputVersion	Displays the version number of the completed request's output file.
OutputVersionName	Displays the version name of the output file of the completed request.
Size	Displays the size of the completed request's output file.
AbbreviatedSize	Displays the size of the completed request's output file, in KB, abbreviated.
Pages	Displays the number of pages in the completed request's output file.
NextStartTime	Displays the time that the request is scheduled to start.
ValuesFileName	Displays the name of the scheduled request's .ROV file.
PrintRange	Displays the range of pages to be printed for the scheduled request.

## Member variables

The following table lists the ReportCast variables that return information about members including users, groups, and roles. Member lists include the list of all members on an e.Reporting Server, the list of members that have privileges for a folder item or channel, and the list of members to be notified about a request.

<b>Variable</b>	<b>Purpose</b>
MemberName	Displays the name of the current user, group, or role.
MemberType	Displays the member's type: User, Group, or Role.

## Privilege variables

The following table lists the ReportCast variables that return information about privileges for a current folder item or channel and a current member. These variables are available from the list privileges command.

Variable	Purpose
PrivilegeName	Displays the name of the current privilege.
Granted	Displays True or False depending on the current privilege, folder item or channel, and member.

## Request status variables

The following table lists the ReportCast variables that return information about the current request's status. These variables are available from the List status command.

Variable	Purpose
DateAndTime	Displays the date and time of the status element.
StatusText	Displays the text of the status element.

## Parameter group variable

The following table lists the ReportCast variable that returns information about related groups of report parameters. These variables are available from the List paramgroups command.

Variable	Purpose
GroupName	Displays the parameter group's name.

## Parameter variables

The following table lists the ReportCast variables that return information about report parameters. These variables are part of report definitions and parameter groups. These variables are available from the list parameters

command. Use the List parameters command either from the top level of a report definition or within a List paramgroups command.

<b>Variable</b>	<b>Purpose</b>
Default Value	Displays the parameter's default value.
Type	Displays the type of information that the parameter should contain, for example, String, Number.
Alias	Displays the parameter's alias, if one exists, or "" if there is no alias.
Name	Displays the parameter's actual name. That name is what ReportCast passes to the factory.
Description	Displays a description of the parameter's name. This description is the same as Alias if Alias is not empty.
Required	Displays True if this parameter is required. Displays False otherwise.
AdHoc	Displays True if this is an ad hoc parameter. Displays False otherwise.
Hidden	Displays True if this is a hidden parameter. Displays False otherwise.
HideText	Displays True if this parameter's value must be masked. Displays False otherwise. Useful for masking values such as passwords in the authentication dialog box.

## Schedule variable

The following table lists the ReportCast variables that return information about the current request's schedule. These variables are available from the List schedules command.

<b>Variable</b>	<b>Purpose</b>
ScheduleDescription	Displays the text description of the request's schedule.

## Printer variables

The following table lists the ReportCast variables that return information about the printer being used for the current request. These variables are available from the list Printerprops command.

Variable	Purpose
Name	Displays the name of the current printer property.
Value	Displays the value of the current printer property.

## ReportCast Channel variables

The following table lists the ReportCast variables that return information about the current channel. These variables are available from the list channels command, or from the Channel detail or Channel contents pages.

Variable	Purpose
ChannelName	Displays the name of the current channel.
ChannelURL	Displays the URL of the current channel.
Subscribed	Displays True if the current user is subscribed to the current channel. Displays False otherwise.
Expiration	Displays the number of days an item remains on the current channel before it is removed.
PollingInterval	Displays the number of minutes between browser refreshes of the current channel's contents. This number is formatted appropriately for the current configuration. For example, commas separate every three digits in English: 1,000.
RawPollingInterval	Displays the number of seconds between browser refreshes of the current channel's contents. This number is not formatted.
NextPollTime	The time at which the browser must refresh the contents of a channel. This value is an HTTP date, and can be used in a <meta> HTML tag to cause a page to expire from the browser's persistent cache. Current information about HTTP dates is available from the HTTP specification available from the World Wide Web Consortium's web site, <a href="http://www.w3.org/">http://www.w3.org/</a> .

<b>Variable</b>	<b>Purpose</b>
SmallIconURL	Displays the URL of the small version of the custom icon for the current channel. This value is empty ("" ) if there is no custom icon.
LargeIconURL	Displays the URL of the large version of the custom icon for the current channel. This value is empty ("" ) if there is no custom icon.
Comment	Displays a description of the channel, with HTML tags. The description may be of any length.
PlainComment	Displays a description of the channel, without the HTML tags. Use PlainComment in multiline input tags on an HTML form.

## Error variables

The following table lists the ReportCast variables that return information about error conditions that occur on e.Reporting Server requests. Error variables are available from the Error page.

<b>Variable</b>	<b>Purpose</b>
ErrorClass	Displays the type of error that occurred: <ol style="list-style-type: none"> <li>1 e.Reporting Server Connection Error</li> <li>2 e.Reporting Server Message Error</li> <li>3 ReportCast Internal Error</li> <li>4 ReportCast Parser Error</li> </ol>
ErrorText	Displays a description of the specific error condition, if possible. The description is localized.
RawErrorText	Displays a description of the specific error condition. The text is not localized.
ErrorNumber	Displays the numeric code for the error. This is an HTTP status code for all but ReportCast internal errors and parser errors.
Server	Displays the name of the report server on which the error occurred.

# Internal ReportCast security

This chapter contains the following topics:

- About ReportCast security
- Providing security for ReportCast administration
- Logging on and authenticating users
- Security and the Apache web server
- Customizing the Login dialog box
- Setting the idle user session timeout
- Modifying the HTTP unauthorized status code
- About the ReportCast Security Extension
- Development considerations
- Configuring the ReportCast Security Extension

---

## About ReportCast security

A reporting web site is accessible to any user who has a web browser and the URL to the site. How do you ensure that users access only those Report Encyclopedia objects for which they have permission? How do you protect sensitive reports, or ensure that only authorized users can perform administrative tasks for ReportCast? This chapter discusses the ReportCast security features and how to use them to provide secure access to your web reporting site.

The types of security you can provide for ReportCast are:

- **Administrative security.** Assign an administrator login and password to provide security for performing ReportCast's administrative tasks. This security applies to tasks specific to ReportCast only. They are not related to e.Reporting Server administration.
- **Default user authentication.** Use the default ReportCast and e.Reporting Server facilities to ensure that users access only those reports and other Report Encyclopedia objects for which they have permission.
- **User authentication using ReportCast Security Extension (RCSE).** Use the RCSE to customize and control the user login and authentication process.

You can also customize the Login dialog box and specify the unauthorized status code that ReportCast returns when authentication fails.

Actuate fully supports HTTPS (Secure Socket Layers, or SSL). The default port for HTTPS is 443. For information about configuring the port numbers, see "Configuring ReportCast port numbers" in Chapter 1, "About Actuate ReportCast."

### Setting the ReportCast security variables

Enabling the ReportCast Security Extension and setting up secure facilities involves using registry keys on NT or environment variables on UNIX to set values for the RCSE security variables.

#### How to set security variables on NT

Create new registry keys using Regedit. Be sure to back up your system before running Regedit.

- 1 From the Start menu, Run Regedit.  
The Registry Editor window displays.
- 2 Choose HKEY\_LOCAL\_MACHINE\Software\Actuate\Actuate Web Agent\4.1.

3 Select Actuate Web Agent.

4 Choose Edit.

The Edit pull-down menu displays.

5 Choose New→String Value.

A new value item displays in the right side of the registry.

6 Type the name of the security variable.

For example, if you are setting the registry key for the AC\_WEBAGENT\_ADMINISTRATOR\_USER\_NAME security variable, type:

AC\_WEBAGENT\_ADMINISTRATOR\_USER\_NAME

7 Set the security variable:

1 Select the security variable name.

2 Right-click on the security variable name, then choose Modify.

The Edit String dialog box appears.

3 Type the value for the security variable in the Value data: field.

4 Choose OK.

## How to set security variables on UNIX

Set the security environment variables in the user environment in which the ReportCast runs. In other words, you must be logged on as the user that runs ReportCast. That is usually the same user that runs the web server, unless ReportCast runs separately from the web server as on HP-UX systems.

The easiest method is to set the security variables and export them in the ReportCast startup script, startwebagent.sh. The ReportCast installation places the startwebagent.sh script in the <netscape-home>/plugins/actuate/bin directory, where <netscape-home> is the root of your Netscape installation. Add the setenv and export statements to the startup script just before the script invokes the ReportCast server, webagentsrvr.

For example, to set and export the AC\_WEBAGENT\_ADMINISTRATOR\_USER\_NAME security variable:

```
AC_WEBAGENT_ADMINISTRATOR_USER_NAME=<user>
export AC_WEBAGENT_ADMINISTRATOR_USER_NAME
```

<user> is the administrator user name.

---

## Providing security for ReportCast administration

You perform administrative tasks for ReportCast from the ReportCast administration web page:

`http://<webserver>/acweb/___admin`

<webserver> is the name of the web server that is running ReportCast.

The ReportCast administration web page accesses administrative information for ReportCast. That page cannot be used to administer the e.Reporting Server.

To enable secure access to ReportCast's administration page, use these registry keys for NT, or these environment variables for UNIX:

- `AC_WEBAGENT_ADMINISTRATOR_USER_NAME` sets the administrator's user name
- `AC_WEBAGENT_ADMINISTRATOR_PASSWORD` sets the administrator's password

You can set a user name and password or leave both blank. For example, a ReportCast installation used for testing purposes only or accessible only internally, may not require security. Setting these values for ReportCast does not affect access to your Report Encyclopedias. For information about setting these security variables, see "Setting the ReportCast security variables," earlier in this chapter.

---

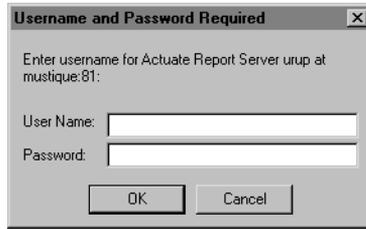
## Logging on and authenticating users

Users can log on using their e.Reporting Server user names and passwords. Another option is using anonymous login. If the anonymous login is set up in the e.Reporting Server, ReportCast automatically logs on users as anonymous.

### Logging on as anonymous

The anonymous user allows users to access reports on a particular e.Reporting Server without having to go through a login screen or any sort of authentication. e.Reporting Server administrators set up the anonymous user in the Administrator Desktop, configure it with View Properties privileges on all but the most sensitive objects, and do not assign a password.

When a user attempts to access a secure object, ReportCast displays the Login dialog box.



The user must enter a valid user name and password to be able to access the secure object. ReportCast reconnects the user to the e.Reporting Server with the new authentication information and privileges.

Most browsers put already-viewed pages into their cache. That makes viewing web pages faster, but the pages could be out of date. If you reconnect to the e.Reporting Server as a different user, make sure that the web reporting pages you view are current. If they are not, perform a forced reload of the page:

- In Netscape Navigator, hold down the Shift while clicking Reload.
- In Microsoft Internet Explorer, clear the cache and then reload the page.

For more information about setting up the anonymous user, see Chapter 2, “Managing Report Encyclopedia security,” in *Administering the Report Encyclopedia*.

## Logging on as an e.Reporting Server user

If the e.Reporting Server administrator did not set up an anonymous user, the Login dialog box appears when the user first accesses the e.Reporting Server. When the user enters a valid user name and password, the initial e.Reporting Server view appears in the browser.

## Authenticating users using forms

To log in as a different user using a dialog box, users must first log out, then shut down their web browser, restart their web browser, and log back in. If you enable form-based login authentication, users can log out and log back in as a different user without shutting down their web browser.

Users with administrator privileges can customize the login and logout forms. Customize the login form by modifying LOGIN.ACHTML. Customize the logout form by modifying LOGOUT.ACHTML.

Form-based authentication is the default for Apache web servers. For other web servers, enable form-based login authentication by setting the AC\_WEBAGENT\_USE\_LOGIN\_FORM registry key (on NT) or environment variable (on UNIX) to TRUE.

---

## Security and the Apache web server

The Apache web server default security process is slightly different than other Unix web servers' default security processes.

Users log in using a form, instead of logging in using a dialog box. For more information, see "Authenticating users using forms," earlier in this chapter.

---

## Customizing the Login dialog box

You can customize the Login dialog box to a certain extent. The Login dialog text is:

- Enter username for <custom text> at <webserver>:
  - <custom text> is the text that you can customize.
  - <webserver> is the name of the web server to which the user wants to connect. You can include the string \$(Server) in your custom text to display the name of the server the user is logging into.

Customize the Login dialog box using the security variable `AC_WEBAGENT_AUTHENTICATION_PROMPT`. For information about setting this security variable value, see "Setting the ReportCast security variables," earlier in this chapter.

For example, setting the following value for `AC_WEBAGENT_AUTHENTICATION_PROMPT`:

```
logging in to MyCorp's $(Server) server
```

results in the following Login dialog prompt to the Sales web server from the ERegion server:

```
Enter username for logging in to MyCorp's Sales server at ERegion:
```

---

## Setting the idle user session timeout

If a user is idle for more than 1800 seconds (30 minutes), they must log in again. Users with administrator privileges can change the idle time value by setting the environment variable `AC_WEBAGENT_LOGIN_TIMEOUT`. Specify the length of time that user sessions can be idle, in seconds. For example, to set the timeout to 15 minutes, add the following line to the startup script just before the script invokes the ReportCast server, `webagentsrvr`:

```
AC_WEBAGENT_LOGIN_TIMEOUT=900
export AC_WEBAGENT_LOGIN_TIMEOUT
```

---

## Modifying the HTTP unauthorized status code

When user authentication fails, ReportCast sends the HTTP unauthorized status code, 401, to the browser. You can modify the HTTP unauthorized status code that ReportCast sends to the browser by setting the `AC_WEBAGENT_UNAUTHORIZED_STATUS_CODE` variable.

Authentication fails if either the ReportCast Security Extension returns a non-zero value from the `AcWebSecurityAuthenticate()` function or login to the e.Reporting Server fails. By default, ReportCast sends back the status code of 401 if authentication fails. The 401 error causes the browser to prompt the user for user name and password. This security variable allows the administrator to prevent clients from trying to authenticate by returning a value other than 401. For information about setting this security variable, see “Setting the ReportCast security variables,” earlier in this chapter.

If you want to customize the text to display if authentication fails, you can customize the `error.achtml` template. For more information about customizing the ReportCast templates, see “About ReportCast templates,” in Chapter 3, “Working with ReportCast templates.”

---

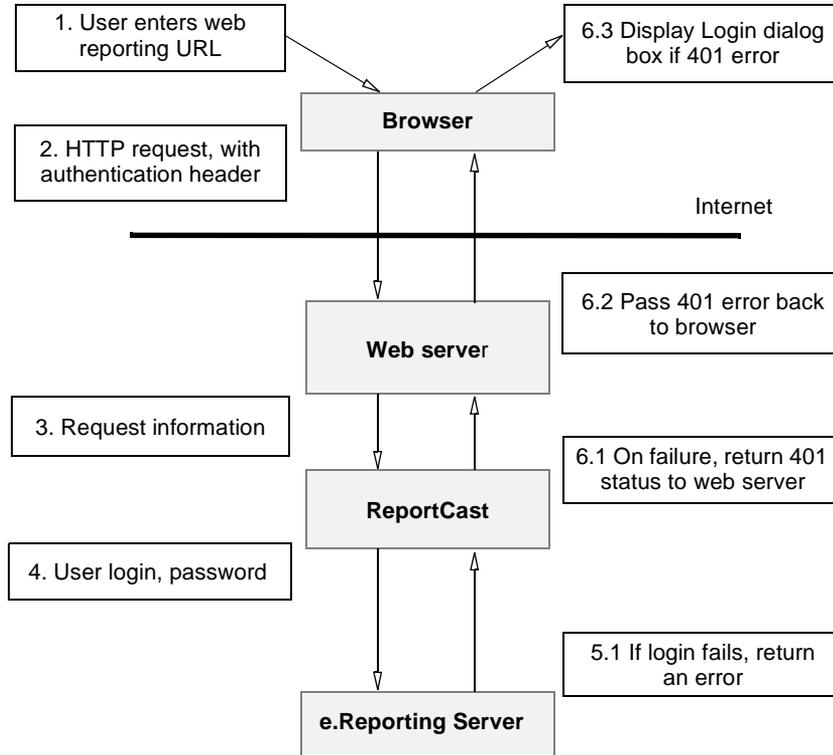
## About the ReportCast Security Extension

The Actuate ReportCast Security Extension controls the way that information associated with requests from the web browser is used by ReportCast to log on to the e.Reporting Server.

### Overview

The Actuate ReportCast Security Extension replaces the default user authentication in ReportCast with an external routine. Using the security extension enables developers to customize the interaction among the web browser, ReportCast, and e.Reporting Server to control the user login and authentication process.

The following diagram illustrates the default authentication process.



The default process is:

- 1 The user enters the web reporting home page URL in the browser.
- 2 The browser passes the HTTP request to the web server, including the authentication header.
- 3 The web server recognizes the acweb keyword and passes the request information to ReportCast.
- 4 ReportCast maps the Basic Authentication header to the user login and password, then passes the user login and password to the e.Reporting Server. If there is no Basic Authentication header, the user login becomes anonymous and the password is null (""). The anonymous login works only if it is already set up on the e.Reporting Server.
- 5 The e.Reporting Server processes the user login and password:
  - 1 If login fails, the e.Reporting Server returns an error to ReportCast.
  - 2 If login succeeds, the e.Reporting Server returns successful to ReportCast.

**6** If login failed:

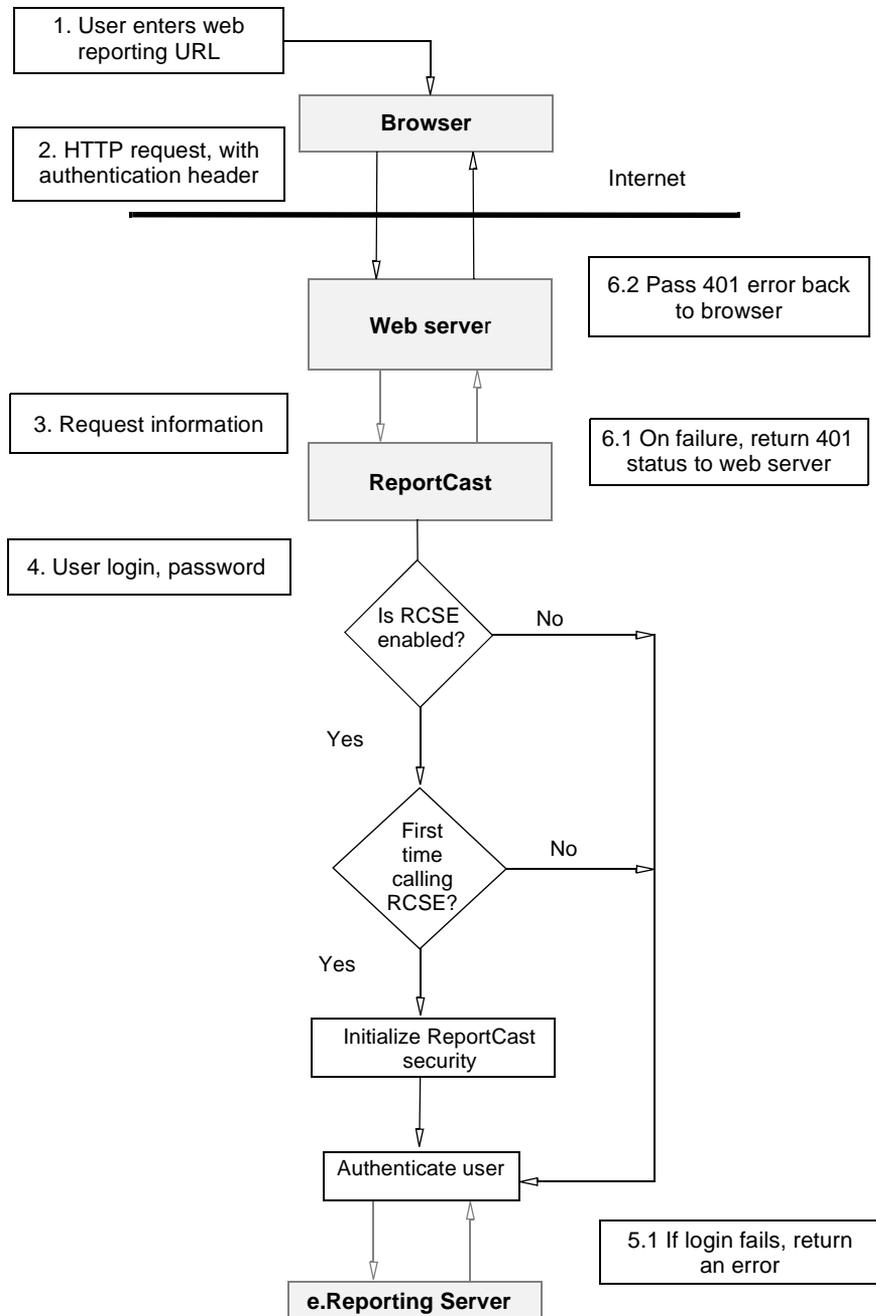
- 1 ReportCast returns the HTTP 401 status to the web server.
- 2 The web server returns the HTTP 401 status to the browser.
- 3 The browser displays the Login dialog box, and the process begins again.

This authentication process is effective most of the time. However, it may require the user to log on multiple times if there is additional security in place on the web reporting site. This authentication process does not support the Microsoft Internet Information Server Windows NT Challenge/Response security feature. It supports only Basic Authentication.

Using the ReportCast Security Extension, you can override the default authentication process and customize the user name and password that ReportCast uses to access the e.Reporting Server. If you enable the ReportCast Security Extension, ReportCast calls it each time it connects to the e.Reporting Server to determine the user name and password to use to log on to the e.Reporting Server. It gets the user name and password from the HTTP headers supplied by the browser.

You enable the ReportCast Security Extension by creating registry entries in NT or by setting environment variables on UNIX. The following illustration shows the authentication process using the ReportCast Security Extension.

The following illustration shows the authentication process using the ReportCast Security Extension.



The process is similar to the default authentication process:

- 1 The user enters the web reporting home page URL in the browser.
- 2 The browser passes the HTTP request to the web server, including the authentication header.
- 3 The web server recognizes the acweb keyword and passes the request information to ReportCast.
- 4 If the ReportCast Security Extension is enabled, and this is the first call to the RCSE, initialize the RCSE.
- 5 Authenticate the user using your custom `AcWebSecurityAuthenticate()`.
- 6 The e.Reporting Server processes the user login and password.
  - 1 If login fails, the e.Reporting Server returns an error to ReportCast.
  - 2 If login succeeds, the e.Reporting Server returns successful to ReportCast.
- 7 If login failed:
  - 1 ReportCast returns the HTTP 401 status to the web server.
  - 2 The web server returns the HTTP 401 status to the browser.
  - 3 The browser displays the Login dialog box, and the process begins again.

## Developing the ReportCast Security Extension

To override the default authentication process and enable the ReportCast Security Extension, you develop a custom DLL to contain the RCSE security library and set a registry key on NT or an environment variable on UNIX.

The ReportCast Security Extension library functions are as follows.

RCSE function	Description
<code>AcWebSecurityStart</code>	Initialize the RCSE library.
<code>AcWebSecurityAuthenticate</code>	Authenticate a user.
<code>AcWebSecurityStop</code>	Shut down the RCSE library.

The ReportCast Security Extension variable is `AC_WEBAGENT_SECURITY_LIBRARY`. Set the `AC_WEBAGENT_SECURITY_LIBRARY` variable to identify the ReportCast Security Extension library for the ReportCast and to enable the ReportCast Security Extension.

When the ReportCast Security Extension is enabled, ReportCast calls the RCSE library functions:

- `AcWebSecurityStart` initializes the ReportCast Security Extension library during startup.
- `AcWebSecurityAuthenticate` translates user security information into e.Reporting Server login information.
- `AcWebSecurityStop` prepares the library for shutdown.

If the `AcWebSecurityStart()` or `AcWebSecurityStop()` functions are not present in the RCSE library, ReportCast continues and does not issue an error message. `AcWebSecurityAuthenticate()` must be present in the RCSE library if the ReportCast Security Extension is enabled.

For more information about enabling the ReportCast Security Extension, see “Configuring the ReportCast Security Extension,” later in this chapter.

For information about setting the security variables, see “Setting the ReportCast security variables,” earlier in this chapter.

---

## AcWebSecurityStart function

Initializes the ReportCast security library.

### Syntax

```
void AcWebSecurityStart( )
```

### Description

Performs internal initialization, such as loading tables. If `AcWebSecurityStart()` is present in the RCSE library, the ReportCast calls it once, when it loads the security library.

---

## AcWebSecurityAuthenticate function

Authenticates a user.

## Syntax

```
int AcWebSecurityAuthenticate(char **httpHeaders, int numberOfHeaders,  
    char *username, char *password)
```

## Description

Called each time ReportCast connects to an e.Reporting Server to service a request. If this function is not present in the RCSE library, or an error occurs during the operating system call to locate the function, ReportCast sends an HTML error page generated from the error.achtml template to the browser.

This function supports using the HTTP headers from the browser to determine the user name and password that ReportCast uses to log on to the e.Reporting Server.

### **httpHeaders [in]**

An array of null-terminated strings, each representing an HTTP header using the format defined in the HTTP 1.1 specification.

### **numberOfHeaders [in]**

Number of elements in the httpHeaders array.

### **username [out]**

Name ReportCast uses to log on to the e.Reporting Server. ReportCast is responsible for allocating and freeing this memory. This memory can hold a string 128 bytes long. Your code must copy the user name into this location.

### **password [out]**

Password to use to log on to the e.Reporting Server. ReportCast is responsible for allocating and freeing this memory. This memory can hold a string 128 bytes long. Your code must copy the password into this location.

## Return value

0 on success, 1 on failure.

If AcWebSecurityAuthenticate() returns 1, ReportCast does not attempt to log on to the e.Reporting Server and returns a status code of 401 or the value specified by the AC\_WEBAGENT\_UNAUTHORIZED\_STATUS\_CODE variable to the web browser.

---

## AcWebSecurityStop function

Shuts down the ReportCast security library.

### Syntax

```
void AcWebSecurityStop( )
```

### Description

Performs internal cleanup for the RCSE library. If `AcWebSecurityStop()` is present in the library, it is called once, when ReportCast shuts down.

---

## Development considerations

Here are some items to consider for developing the Actuate ReportCast Security Extension:

- The functions that the security library exports, `AcWebSecurityStart()`, `AcWebSecurityAuthenticate()`, and `AcWebSecurityStop()`, should all be declared `extern C` if you are writing the library in C++. This procedure prevents the compiler from changing the symbol name into something other than what ReportCast is looking for.
- Your security library must be thread-safe and cannot depend on any one thread handling a particular request.
- The HTTP header names that are passed to the security library should be in the format specified in the HTTP specification. The headers do not use the CGI or ISAPI format. For example, the header `Accept-Charset` is used, rather than `HTTP_ACCEPT_CHARSET`. This practice allows you to write the Security Extension code using the standard HTTP headers so that it is portable among different web servers and platforms.
- Ensure that your code is not case-sensitive.
- When testing your ReportCast Security Extension library, be aware that ReportCast caches connections to the e.Reporting Server. If you are changing user passwords on the server during your testing, you may have to clear cached connections using the ReportCast administration page.

## Example

The following code example shows a template used to develop a security library:

```
#ifndef WIN32
#include <windows.h>
#endif

#ifdef __cplusplus
extern "C"
#endif

void
#ifdef WIN32
WINAPI
#endif
AcWebSecurityStart() {
// Your code here.
}

void
#ifdef WIN32
WINAPI
#endif
AcWebSecurityStop() {
// Your code here.
}

int
#ifdef WN32
WINAPI
#endif
AcWebSecurityAuthenticate(char **httpHeaders,
    int numberOfHeaders,
    char *username,
    char *password) {
// Your code here.
}

#ifdef __cplusplus
}
#endif
```

## NT-specific information

On NT, you must use the WINAPI calling convention in the function definitions for the RCSE functions. Declare the RCSE functions in a DEF file that is linked to the DLL.

For example, the following code shows the contents of a DEF file to link into the DLL:

```
DESCRIPTION 'Your Security Extension Description'  
EXPORTS  
    AcWebSecurityStart  
    AcWebSecurityStop  
    AcWebSecurityAuthenticate
```

## Server-specific information

The following sections contain additional information that applies to the supported web servers, Netscape and Microsoft.

### Netscape server

The ReportCast Security Extension supersedes any previous ReportCast authentication customization solutions. The security extension feature allows implementation of custom security authentication that is portable and avoids the need to add directives to the obj.conf file to achieve customization.

### Microsoft Internet Information server

By default, the Microsoft Internet Information server (IIS) uses the Windows NT security information from the browser to grant or deny access to all resources on the web server. IIS considers the ReportCast DLL to be the resource to which it must authorize access, rather than the e.Reporting Server. Each user name and password must exist on both the NT system to access ReportCast, and the e.Reporting Server to access the Report Encyclopedia. Since this setup is not practical in some cases, the ReportCast configuration customizes this behavior.

If the security extension is not enabled, ReportCast overrides the default behavior of IIS and a component of ReportCast acts as an authentication filter. This filter instructs IIS to access ReportCast as the NT user designated for IIS anonymous access. In this configuration, IIS must be configured to allow anonymous access and Windows NT Challenge/Response must be disabled. When ReportCast attempts to connect to the e.Reporting Server, it uses Basic Authentication to gather the user name and password information.

The alternative authentication method is to use the ReportCast Security Extension. If the security extension is enabled, ReportCast does not override

the default behavior of IIS to authorize access to the ReportCast DLL. Authentication information for the ReportCast DLL comes from the browser. ReportCast does not filter the user information. The user information coming from the browser must identify a valid NT user. If the browser does not send authentication information, and if IIS is configured to allow anonymous access, IIS accesses ReportCast as the NT user designated for anonymous access.

If you want to access ReportCast using a different Windows NT user than that specified by the browser, install an additional ISAPI filter DLL in Internet Information Server.

Depending on the security model you use, the security extension returns the same user information as the browser sent, or maps the user information to a completely different user. In either case, the user it returns must exist in the e.Reporting Server if ReportCast is to log on successfully.

---

## Configuring the ReportCast Security Extension

To enable the ReportCast Security Extension, you must set an environment variable in UNIX or create a registry entry under Windows NT.

The `AC_WEBAGENT_SECURITY_LIBRARY` variable specifies the ReportCast security library, enables the ReportCast Security Extension, and causes ReportCast to make the calls into your security library.

On Windows NT, set the variable as a string value under the `HKEY_LOCAL_MACHINE\Software\Actuate\Actuate Web Agent\4.1` key.

On UNIX systems, set the `AC_WEBAGENT_SECURITY_LIBRARY` environment variable in the environment of the user that ReportCast runs in. You must be logged on as the user that runs ReportCast. That is usually the same user that runs the web server, unless ReportCast runs separately from the web server as on HP-UX systems. Set this variable before you or the startup script start ReportCast.

For more information about setting the security variables, see “Setting the ReportCast security variables,” earlier in this chapter.

---

## **AC\_WEBAGENT\_SECURITY\_LIBRARY variable**

ReportCast security library name.

### **Description**

Set `AC_WEBAGENT_SECURITY_LIBRARY` to allow ReportCast to locate your security library. If this variable is not set, ReportCast uses the default authentication scheme.

The variable value is the name of the DLL or shared object file that contains the authentication functions. The name can include either an absolute or relative path.

ReportCast loads the library once, the first time a request is made to ReportCast. ReportCast unloads the library when it exits. If the library cannot be loaded, ReportCast sends an error page to the browser.

# Security for Internet deployment

This chapter contains the following topics:

- About network security and Actuate
- Data protection strategies
- Understanding Actuate component communication
- Deploying Actuate products in Internet environments

---

## About network security and Actuate

This chapter discusses configuring Actuate e.Reporting Suite 4 products for deployment in a secured network environment. Project managers, installation engineers, and security team members can plan hardware and software requirements, installation configuration options, and how Actuate fits into the Internet security infrastructure of their organization based on information in this chapter.

The topics discussed include the technologies that deter unauthorized access to corporate data, such as firewalls, Network Address Translation (NAT), and proxy servers.

This chapter is not a guide to securing your network. This chapter is a general guide to how Actuate products work with secure environments. This chapter assumes a working knowledge of TCP/IP communications.

---

## Data protection strategies

Actuate e.Reporting Suite 4 provides a structured content generation solution for web applications. Deploying Actuate applications developed for the Internet requires planning for network security.

Internet applications support access to information within an organization from outside that organization. Because the organization's internal network is connected to the Internet, there is the risk of unauthorized access to the corporate network and to the data that resides on that network.

### Protecting corporate data

Typically companies use firewalls to prevent unauthorized access to corporate networks and data.

A firewall is a system or group of systems that restrict access between two networks, such as an organization's internal network and the Internet. Firewalls keep unauthorized users out. As a result, firewalls prevent damage caused by malicious programs such as worms and viruses from spreading to other parts of your network. At the same time, firewalls allow legitimate business to tunnel through the firewall and be efficiently conducted on your network.

Firewalls can be used to restrict access between two internal networks, for example, the accounting and engineering networks. Security teams configure firewalls to allow traffic using specific protocols, such as HTTP, over specific network addresses and ports.

Companies also use Network Address Translation (NAT). NAT is routers and software that allow private networks using unregistered, private IP (Internet Protocol) addresses to connect to the Internet.

Finally, proxy servers, specialized web servers or hardware that operate on or behind a firewall, improve efficient use of network bandwidth and offer better network security.

Organizations use one or a combination of these technologies to prevent unauthorized access to the corporate network.

---

## Understanding Actuate component communication

This section describes the relationship between Actuate component communication and your secure network environment.

Actuate e.Reporting Server, ReportCast, and all client products use TCP/IP and related protocols for all network communication. The communications process between client products and the e.Reporting Server is the same as the communications process between ReportCast and the e.Reporting Server.

The originating or outbound port is the port that a component uses to initiate two-way connections between itself and another component.

The destination or inbound port is the port to which an originating port initiates two-way connections.

Actuate components have these communications processes:

- Communicating between the web browser and web server
- Establishing the e.Reporting Server port
- Connecting to and browsing the Report Encyclopedia
- Running e.reports
- Viewing e.reports in DHTML format
- Viewing e.reports using the LRX

You configure the ports to use for specific communications tasks at e.Reporting Server and ReportCast startup using registry settings on NT systems or environment variables on UNIX systems.

The types of ports you can configure are:

- e.Reporting Server port. This port is used for the ReportCast to e.Reporting Server connection.
- Cached connection management. This port is used to control the number of ReportCast's cached connections.

- View process port. This port is used for viewing DHTML reports, PDF content, and XML data and XML display.
- Data reply connection. This port is used to view reports using the LRX or Viewer, retrieving parameters for running reports, and notifications for report requests.

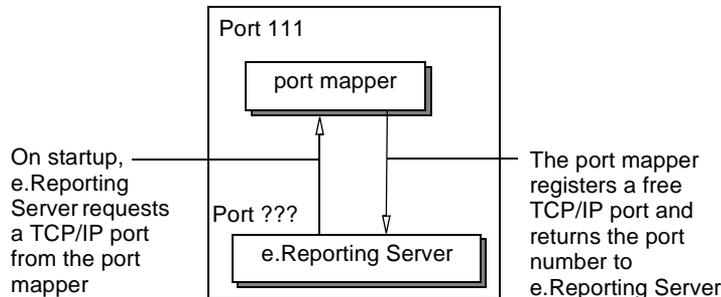
## Establishing the e.Reporting Server port

The e.Reporting Server port is the destination port for connections from ReportCast to the e.Reporting Server. ReportCast and the e.Reporting Server use the connection for:

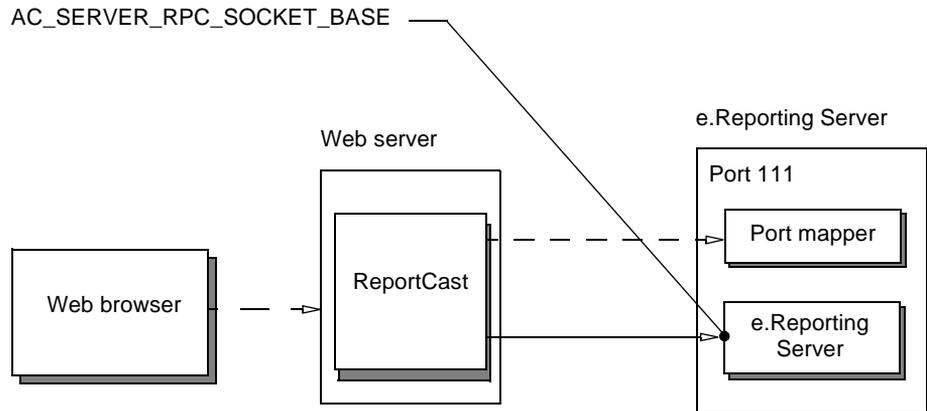
- Login
- Report Encyclopedia browsing
- Report Encyclopedia administration
- Initiating report requests
- Initiating report viewing

e.Reporting Server applications use the TCP/IP port mapper to determine the port through which to communicate with the e.Reporting Server. When the e.Reporting Server starts, it requests a communications port from the port mapper. The port mapper locates an available TCP/IP port, registers the port number, and returns the port number to the e.Reporting Server.

The port mapper listens for queries on port 111. Applications query the port mapper for the port used by the e.Reporting Server. The port mapper responds with the registered port. Applications use the registered port to connect to the e.Reporting Server.



The following illustration shows the connection between ReportCast and the e.Reporting Server.



- - - - - Dashed lines are transient connections  
 \_\_\_\_\_ Solid lines are persistent connections

The e.Reporting Server port uses the RPC protocol over TCP. There is one destination port for all user connections. You can configure the number of cached connections. The peak number of connections depends on the user load.

## Configuring the e.Reporting Server port

You can set the port number to use for connections and the number of ports to search if that port is not available.

To configure the e.Reporting Server port, set the following registry keys on NT or environment variables on UNIX on the e.Reporting Server machine:

- `AC_SERVER_RPC_SOCKET_BASE` sets the port number that the e.Reporting Server attempts to register with the port mapper.
- `AC_SERVER_RPC_SOCKET_COUNT` sets the number of ports search if the registered port is not available.

When the e.Reporting Server attempts to register a port with the port mapper, it searches from `AC_SERVER_RPC_SOCKET_BASE` to `AC_SERVER_RPC_SOCKET_BASE + AC_SERVER_RPC_SOCKET_COUNT - 1` to find an open port. If no ports are available within the configured range, the port mapper allocates a port. The default is to allow the port mapper to allocate the port.

For detailed information about configuring registry keys on NT or environment variables on UNIX, see Chapter 2, “Report server administration on UNIX,” or Chapter 3, “Report server administration on Windows NT,” in the *Actuate e.Reporting Server Guide*.

## **Configuring cached connection management**

ReportCast caches connections to the e.Reporting Server port, `AC_SERVER_RPC_SOCKET_BASE`. To control the number of cached connections, set the following registry keys on NT or environment variables on UNIX on the web server machine:

- `AC_WEBAGENT_CONNECTION_CACHE_SIZE` limits the total number of connections. Actuate drops the oldest connections automatically.
- `AC_WEBAGENT_CONNECTION_CACHE_TIMEOUT` limits the life of the connection.

For more information, see Chapter 2, “Administering ReportCast.”

## **Web browser to web server communication**

Web browser to web server communication uses the standard HTTP or HTTPS (SSL) protocols. Actuate products do not require any special configuration or consideration for HTTP or HTTPS communication. HTTPS is transparent to Actuate.

## **Connecting to and browsing the Report Encyclopedia**

The process for connecting to and browsing the Report Encyclopedia is:

- The web browser connects to the web server using HTTP or HTTPS to a standard port, usually port 80, with a request to view the list of files in a folder.
- On the first connection, ReportCast opens an originating port.
- ReportCast connects to the port mapper using TCP to determine the e.Reporting Server port.
- The port mapper returns the e.Reporting Server port.
- ReportCast connects to the e.Reporting Server using the destination port supplied by the port mapper. ReportCast uses the new connection to retrieve information from the e.Reporting Server. All users share one destination port.
- ReportCast generates the web reporting page and returns it to the web server.

- The web server returns the web reporting page to the web browser using HTTP.

Connections from the web browser to the web server and ReportCast, and from ReportCast to the port mapper, are transient. The connection between ReportCast and the e.Reporting Server remains open.

The desktop products use the same process to communicate with the e.Reporting Server.

## Running and viewing reports

This section describes the process of running and viewing reports and how to configure the view process and data reply ports.

### Running reports

One connection is used for each concurrent user running a report. Each connection opens a pair of ports: one on the originating side, one on the destination side. Data moves in both directions.

The process for running reports is:

- The web browser connects to the web server using HTTP or HTTPS to a standard port, usually port 80.
- ReportCast requests the parameters needed to run the report.
- The e.Reporting Server establishes a new connection to ReportCast for sending the report's parameter data.
- ReportCast generates the web reporting page and returns it to the web server.
- The web server returns the web reporting page to the web browser using HTTP.

The desktop products use the same process to run a report.

The process for running reports uses data reply ports. Configuring data reply ports is described in "Configuring data reply ports," next in this chapter.

### Configuring data reply ports

The LRX and Viewer use the data reply ports to view reports, to retrieve parameters for running reports, and for report request notification.

The data reply connection process is:

- ReportCast logs in and gets an RPC connection to the e.Reporting Server.
- User initiates a request to run a report or to view content using the LRX viewer.
- ReportCast requests parameters from the e.Reporting Server or requests content for the LRX viewer.
- e.Reporting Server acknowledges the request and initiates a connection to ReportCast:
  - e.Reporting Server opens an originating port (AC\_SERVER\_REPLY\_SOCKET\_BASE).
  - e.Reporting Server connects to the destination port on the ReportCast machine (AC\_SERVER\_SOCKET\_BASE).
- e.Reporting Server sends the report data.
- e.Reporting Server closes the connection.

To configure originating data reply ports, set the following registry keys on NT or environment variables on UNIX on the e.Reporting Server machine:

- AC\_SERVER\_REPLY\_SOCKET\_BASE sets the number at which to begin allocating originating ports.
- AC\_SERVER\_REPLY\_SOCKET\_COUNT sets the maximum number of ports to use.

When the e.Reporting Server receives a data reply request, it searches from AC\_SERVER\_REPLY\_SOCKET\_BASE to AC\_SERVER\_REPLY\_SOCKET\_BASE + AC\_SERVER\_REPLY\_SOCKET\_COUNT-1 to find an open port. The default value for AC\_SERVER\_REPLY\_SOCKET\_BASE is 4096. The default value for AC\_SERVER\_REPLY\_SOCKET\_COUNT is 8192.

To configure destination ports, set the following registry keys (on NT) or environment variables (on UNIX) on the ReportCast or Viewer machine:

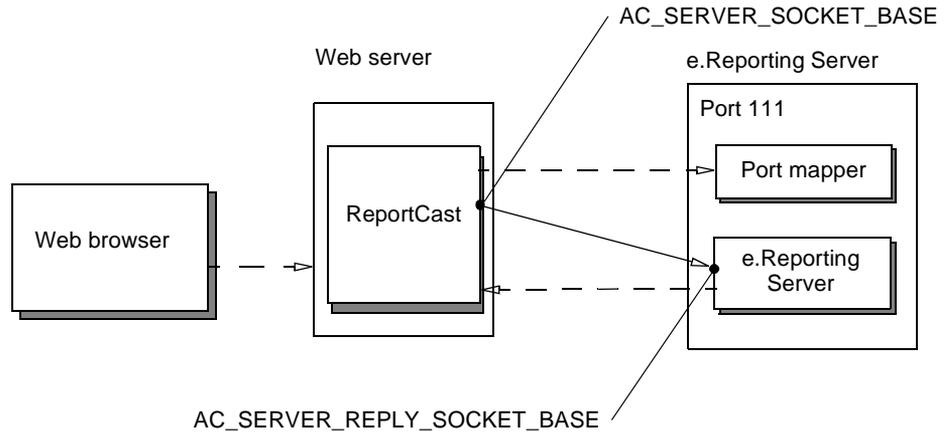
- AC\_SERVER\_SOCKET\_BASE sets the number at which to begin allocating destination ports.
- AC\_SERVER\_SOCKET\_COUNT sets the maximum number of ports to use.

When the e.Reporting Server receives a data reply request, it searches from AC\_SERVER\_SOCKET\_BASE to AC\_SERVER\_SOCKET\_BASE + AC\_SERVER\_SOCKET\_COUNT-1 to find an open port. The default value for AC\_SERVER\_SOCKET\_BASE is 4096. The default value for AC\_SERVER\_SOCKET\_COUNT is 8192.

Data reply ports use the TCP protocol and the UDP protocol for notifications. Actuate opens one port for each concurrent data request. For example, if ten

users simultaneously request a report, ten pairs of ports open. For each pair of ports, one is on the origination side and one is on the destination side. Data reply connections are opened and closed for each data reply request.

The following illustration shows the connections between ReportCast and the e.Reporting Server for data reply requests.



- - - - - Dashed lines are transient connections  
 \_\_\_\_\_ Solid lines are persistent connections

## Viewing e.reports as DHTML, PDF, or XML

Viewing DHTML, PDF, or other structured content using the view process uses one port per view process. Each view process port is shared by all users.

The process for viewing e.reports in DHTML format is:

- The web browser connects to the web server using HTTP or HTTPS to a standard port, usually port 80, with a request to view a report as DHTML.
- ReportCast makes a request to view a DHTML report.
- The e.Reporting Server tells ReportCast which port to use to retrieve the DHTML content.
- ReportCast connects to the e.Reporting Server port for viewing DHTML content. The e.Reporting Server returns DHTML content to ReportCast. Actuate uses one port for each view process. Typically, the number of ports used is one or two. See the *Actuate e.Reporting Server Guide* for more information. All users share the same view process ports.
- ReportCast generates the web reporting page and returns it to the web server.

- The web server returns the web reporting page to the web browser using HTTP.

The process of viewing e.reports in DHTML, PDF, or XML uses the view process port. Configuring the view process port is described in “Configuring the view process port,” next in this chapter.

## Configuring the view process port

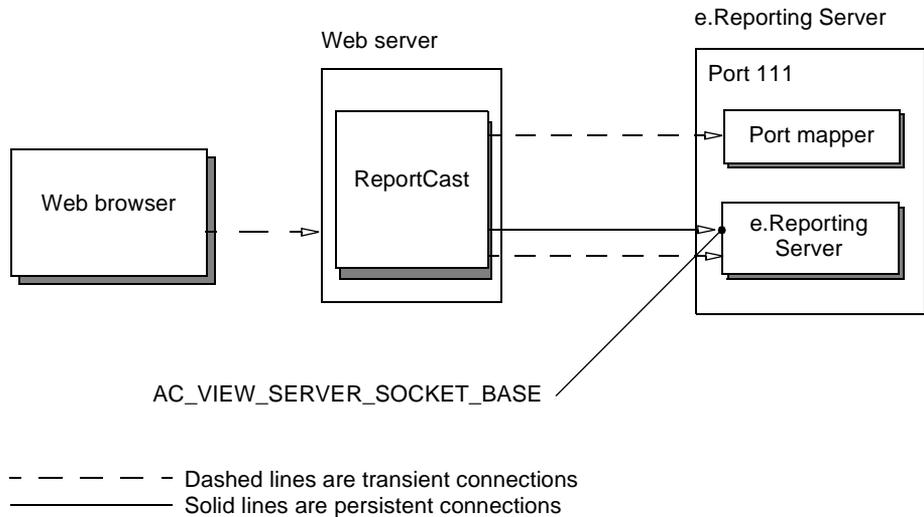
The view process port is used for viewing DHTML, PDF, and XML content. Set the following registry keys on NT or environment variables on UNIX on the e.Reporting Server machine:

- `AC_VIEW_SERVER_SOCKET_BASE` sets the port number at which to begin allocating view process connection ports. The default value is 15500.
- `AC_VIEW_SERVER_SOCKET_COUNT` sets the maximum number of ports to search for open ports. The default value is 200.

When the e.Reporting Server receives a request for the view process, it searches from `AC_VIEW_SERVER_SOCKET_BASE` to `AC_VIEW_SERVER_SOCKET_BASE + AC_VIEW_SERVER_SOCKET_COUNT-1` to find an open port. Set `AC_VIEW_SERVER_SOCKET_BASE` to a number greater than 1024. Set `AC_VIEW_SERVER_SOCKET_COUNT` to a number less than 65535.

The view process port uses the TCP protocol. Actuate uses one port for each view process. View process connections are opened and closed for each request to view DHTML, XML, or PDF content.

The following illustration shows the connections among ReportCast, the view process, and the e.Reporting Server.



## Viewing e.reports using the LRX

One connection is used for each concurrent user viewing a report. Each connection opens a pair of ports: one on the originating side, one on the destination side. Data moves in both directions.

The process for viewing e.reports using the LRX is:

- The web browser connects to the web server using HTTP or HTTPS to a standard port, usually port 80, with the request to view the reporting using the LRX.
- ReportCast makes a request to view a report using the LRX.
- The e.Reporting Server establishes a new connection to a ReportCast port for sending report data for viewing.
- ReportCast generates the web reporting page and returns it to the web server.
- The web server returns the web reporting page to the web browser using HTTP.

The Viewer and other desktop products use the same process to view a report on the e.Reporting Server.

The process for viewing e.reports using the LRX uses data reply ports. Configuring data reply ports is described in “Configuring data reply ports,” earlier in this chapter.

## Connecting to the Report Encyclopedia with the split ReportCast configuration for UNIX

The split ReportCast configuration supports installing the ReportCast Agent on one machine, and the ReportCast Server on another machine. You can simplify the configuration of your firewall by placing the ReportCast Agent with the web server in the DMZ and the ReportCast Server on a separate machine on the corporate net. This configuration is currently available only on UNIX.

The split ReportCast communication process is:

- The web browser connects to the web server using HTTP or HTTPS to a standard port, usually port 80, with a request to view the list of files in a folder, run a report, or view a report.
- The web server passes the request to the ReportCast Agent.
- The ReportCast Agent connects to the ReportCast Server on the port specified by `AC_WEBAGENT_PORT` on the machine specified by `AC_WEBAGENT_HOST` and passes the request to the ReportCast Server.
- The ReportCast Server connects to the e.Reporting Server.
- The ReportCast Server generates the web reporting page and returns it to the ReportCast Agent.
- The ReportCast Agent returns the web reporting page to the web server.
- The web server returns the web reporting page to the web browser using HTTP.

## Configuring the split ReportCast configuration for UNIX

To configure the split ReportCast configuration, set the following environment variables:

- `AC_WEBAGENT_PORT` sets the port to use for communicating between the ReportCast Agent and the ReportCast Server. Set `AC_WEBAGENT_PORT` on both the ReportCast Agent and ReportCast Server machines.
- `AC_WEBAGENT_HOST` sets the name of the machine hosting the ReportCast Server. Set `AC_WEBAGENT_HOST` only on the ReportCast Agent machine.

When the ReportCast Agent receives a request, it connects to the ReportCast Server on the port specified by the `AC_WEBAGENT_PORT` setting, on the host machine specified by the `AC_WEBAGENT_HOST` setting.

The default value for AC\_WEBAGENT\_PORT is 5050. Set AC\_WEBAGENT\_PORT to the same value on both the ReportCast Agent and the ReportCast Server machines.

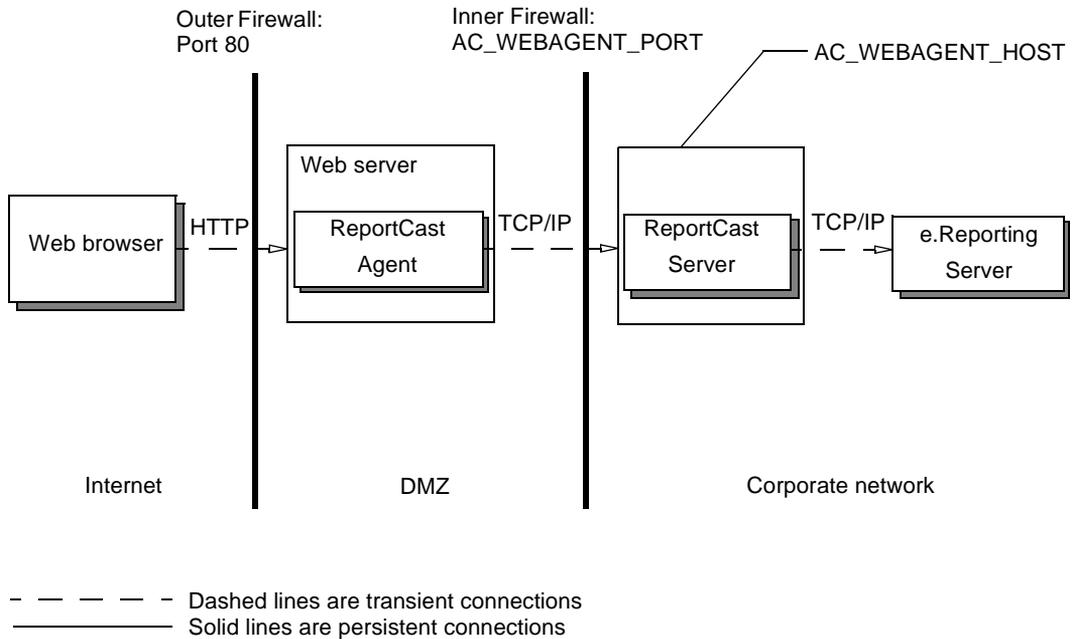
Set AC\_WEBAGENT\_PORT as follows:

- For Netscape web servers only, on the ReportCast Agent side, set AC\_WEBAGENT\_PORT in the Netscape start script.
- For all web servers including Netscape on the ReportCast Agent side, set AC\_WEBAGENT\_PORT in the nph-actuate.cgi and ph-actuate.cgi scripts.
- On the ReportCast Server side, set AC\_WEBAGENT\_PORT in startreportcast.sh.

Set AC\_WEBAGENT\_HOST on the ReportCast Agent side only as follows:

- For Netscape web servers only, set AC\_WEBAGENT\_HOST in the Netscape start script.
- For all web servers, including Netscape, set AC\_WEBAGENT\_HOST in the nph-actuate.cgi and ph-actuate.cgi scripts.

The following illustration shows the connections among ReportCast Agent, ReportCast Server, and the e.Reporting Server.



---

## Deploying Actuate products in Internet environments

There are many options for configuring network security. This section presents some scenarios and discusses requirements for deploying Actuate products in secure Internet environments.

Some routers and firewalls perform Network Address Translation (NAT). You can set up NAT to perform one-to-one IP address mapping in both directions. In a connection from point A to point B, NAT must map both A to B, and B to A.

The following are sample network security configuration options:

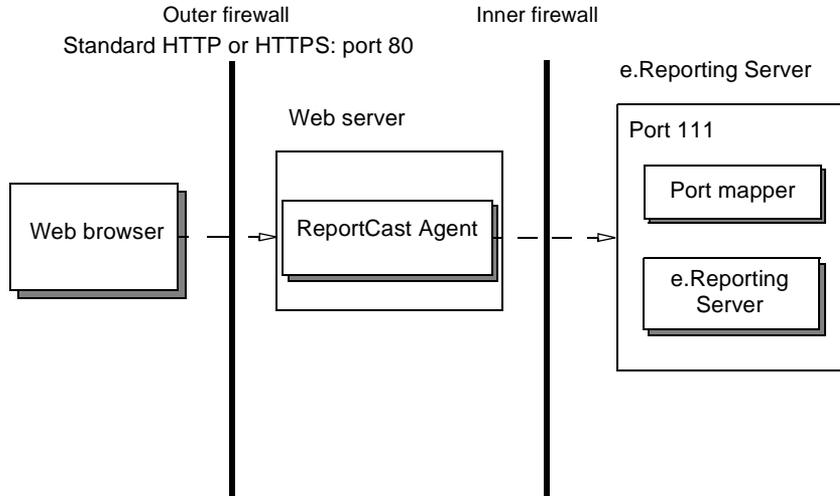
- Web server and ReportCast in the DMZ places the web server and ReportCast between two firewalls. The e.Reporting Server resides behind the inner firewall.
- Proxy server in the DMZ places a proxy server between two firewalls. The web server, ReportCast, and e.Reporting Server reside behind the inner firewall.
- Split ReportCast Installation for UNIX places the web server and ReportCast Agent in the DMZ. The ReportCast Server and e.Reporting Server reside behind the second firewall.

The following sections discuss firewall configuration options, including port configuration instructions.

### Understanding configuration options and firewall setup

Most firewall configurations are set up to accept traffic from specific protocols, such as HTTP or HTTPS traffic only. Generally, corporations expose only limited ports and one IP address to the World Wide Web. All Internet traffic comes through specific ports, usually including port 80, and to the one outside IP address. From port 80, Internet traffic is routed to appropriate internal IP addresses and ports, often using Network Address Translation (NAT).

The web browser to web server communications process is the same for each configuration. If there is a firewall, communication uses standard HTTP or HTTPS protocols, as shown in the following diagram.

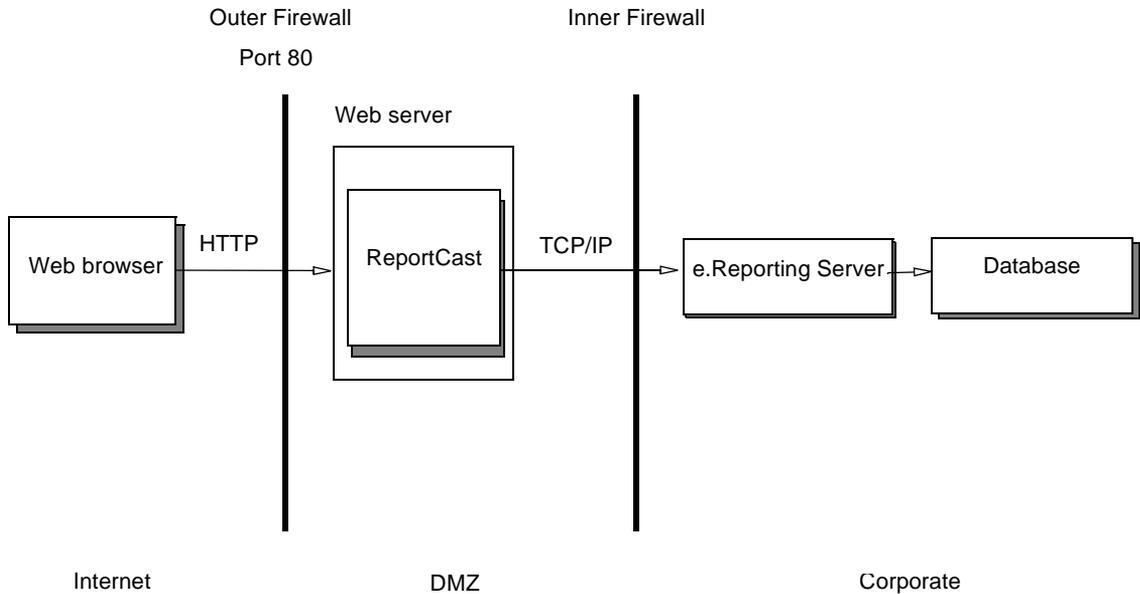


- - - - - Dashed lines are transient connections  
 \_\_\_\_\_ Solid lines are persistent connections

The following sections discuss each firewall configuration option.

## Dual firewalls with ReportCast in the DMZ

The following illustration shows a dual firewall configuration with the web server and ReportCast in the DMZ.

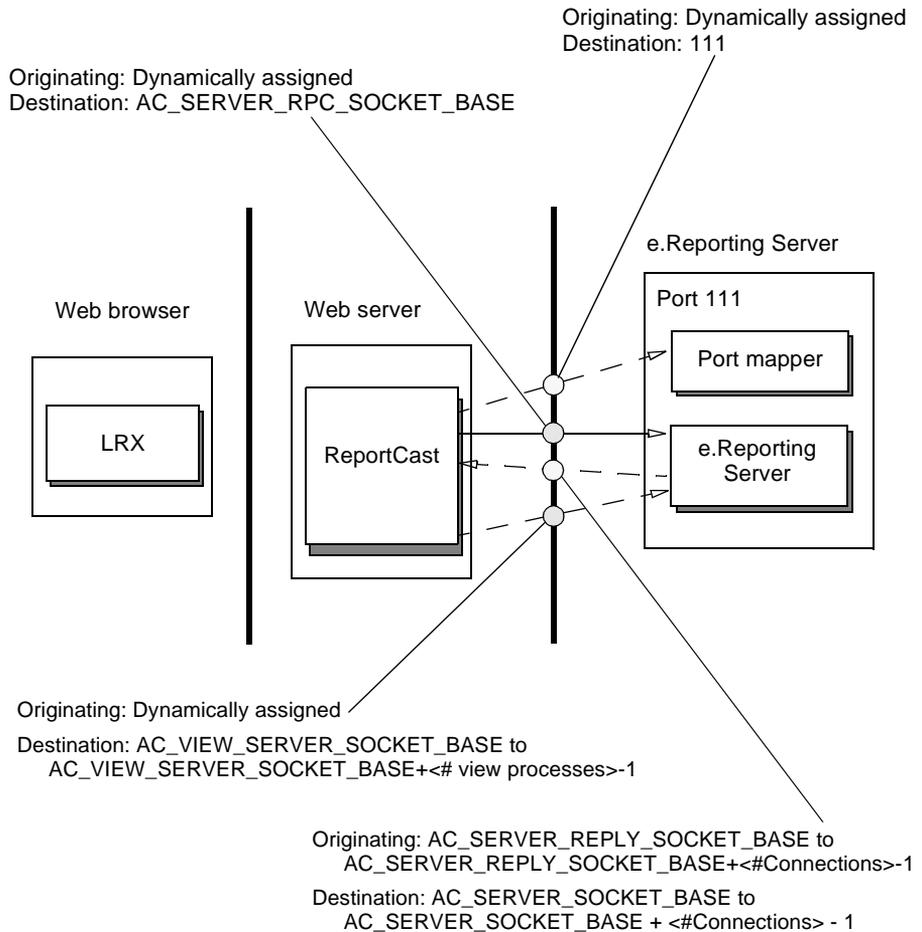


- - - - - Dashed lines are transient connections  
 \_\_\_\_\_ Solid lines are persistent connections

Web browsers communicate with the web server through one well-known port, usually port 80, in the outer firewall. ReportCast passes requests through the inner firewall to the corporate net and the e.Reporting Server using TCP/IP. The e.Reporting Server communicates with the Report Encyclopedia and databases using TCP/IP.

This dual firewall option provides an effective screen against unauthorized access. Internet traffic comes through only one port, generally using only one protocol: HTTP or HTTPS. Users on the Internet cannot gain direct access to the corporate network, because its ports and IP addresses are known only internally. Users on the Internet have limited options for accessing the web server machine, because they can only communicate using HTTP.

The following illustration shows the inner firewall configuration.



- - - - - Dashed lines are transient connections.  
 \_\_\_\_\_ Solid lines are persistent connections.

For information about configuring the originating and destination ports, see “Configuring the e.Reporting Server port,” “Configuring data reply ports,” and “Configuring the view process port,” earlier in this chapter.

## NAT considerations for dual firewalls with ReportCast in the DMZ

Actuate supports the NAT Static Address Translation algorithm, where users establish one-to-one connections between corporate (inside) network addresses and global (outside, Internet) network addresses. Actuate

components initiate connections from either side of the firewall or routers. The IP address mapping must operate and be consistent from both sides of the firewall or routers.

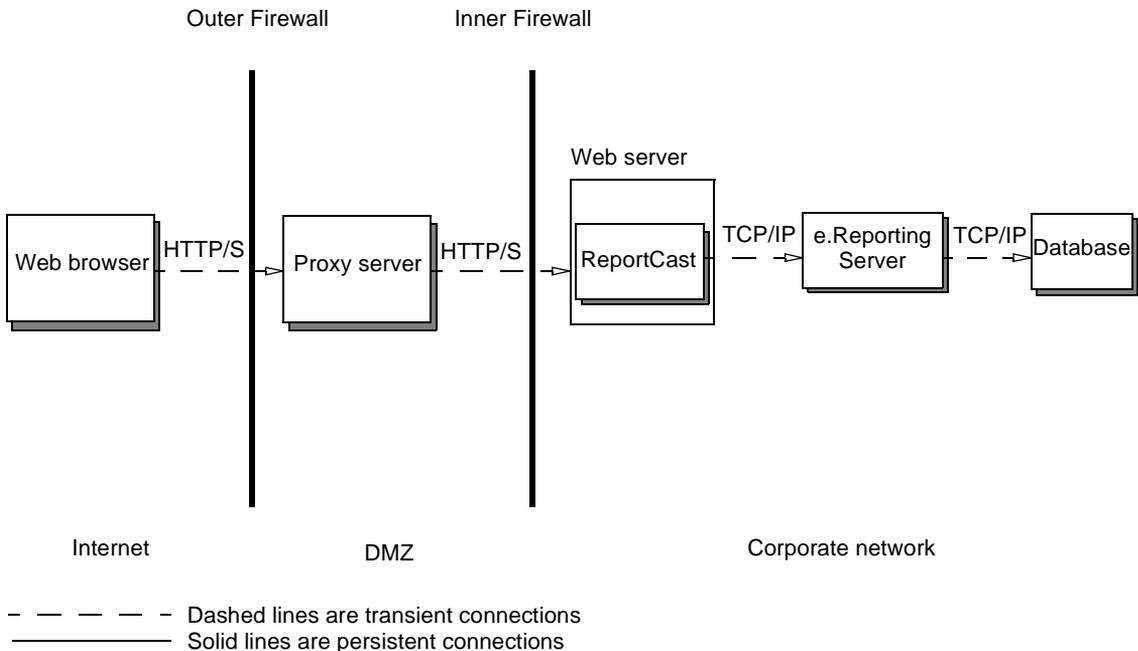
This configuration does not support the following NAT algorithms:

- Dynamic Source Address Translation
- Port Address Translation (PAT)
- Destination Address Rotary Translation

## Dual firewalls with a proxy server

In this configuration, a proxy server resides in the DMZ. The web server, ReportCast, and e.Reporting Server reside behind the inner firewall on the corporate network. The proxy server accepts HTTP or HTTPS traffic from the Internet, then passes the HTTP or HTTPS requests to the web server on the corporate network. ReportCast receives requests from the web server and passes the requests to the e.Reporting Server using TCP/IP. The e.Reporting Server passes the requests to the Database using TCP/IP.

The following illustration shows a dual firewall configuration with a proxy server in the DMZ.



The proxy server handles all communications with the Internet using HTTP through one well-known port, usually port 80. The proxy server communicates with the web server on the corporate network using HTTP through one internally-known port. ReportCast and the e.Reporting Server use TCP/IP to communicate.

This configuration option provides one access point for users on the Internet, through one IP address and well-known port, and one access point to the corporate network. The proxy configuration severely limits the possibilities of accessing the corporate network from the Internet.

## **NAT considerations for dual firewalls with a proxy server**

Configure NAT as you usually would for use with a proxy server. Refer to your NAT and proxy server documentation for configuration information.

## **Split ReportCast installation for UNIX**

In this configuration, available only on UNIX, you can install ReportCast in two pieces:

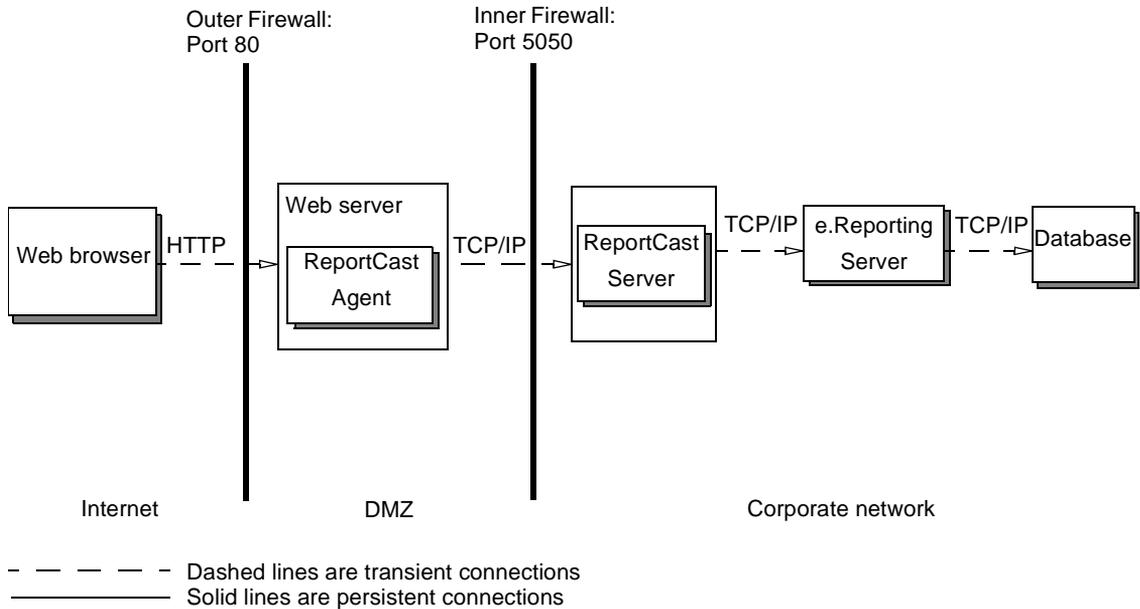
- ReportCast Agent resides on the web server and passes requests to the ReportCast Server.
- ReportCast Server may reside on a machine separate from ReportCast Agent and passes ReportCast requests to the e.Reporting Server.

You install the web server and ReportCast Agent in the DMZ. The ReportCast Server and e.Reporting Server reside behind the inner firewall on the corporate network.

The split ReportCast configuration provides the following advantages:

- Requires only one port through the inner firewall. The default is port 5050. The port is configurable.
- Does not require a proxy server.

The following illustration shows the split ReportCast configuration.



The ReportCast Agent communicates with the ReportCast Server through one inner firewall port using TCP/IP. All requests go through the single inner firewall port.

For information about configuring ports for the split ReportCast configuration, see “Configuring the split ReportCast configuration for UNIX,” earlier in this chapter.

### NAT considerations for the split ReportCast configuration

There are no special NAT considerations for the split ReportCast configuration.

### Placing the e. Reporting Server in the DMZ

This configuration is possible, but not recommended. In this configuration, the web server, ReportCast, and the e.Reporting Server reside in the DMZ. The e.Reporting Server contains corporate data in the form of reports. Corporate data is vulnerable to outside attack in this configuration, which presents an unacceptable security risk.

# Index

## Symbols

# as command indicator 107  
& command separator 76  
\* wildcard 111  
... wildcard 111  
? as directive indicator 72  
? wildcard 111

## A

AbbreviatedCurrentSize variable 142  
AbbreviatedFileSize variable 138  
AbbreviatedSize variable 142  
AbbreviatedType variable 138  
AC\_SERVER\_REPLY\_SOCKET\_BASE  
  setting 172  
AC\_SERVER\_REPLY\_SOCKET\_COUNT  
  setting 172  
AC\_SERVER\_RPC\_SOCKET\_BASE  
  setting 169  
AC\_SERVER\_RPC\_SOCKET\_COUNT  
  setting 169  
AC\_SERVER\_SOCKET\_BASE setting 21, 22  
AC\_SERVER\_SOCKET\_COUNT setting 21,  
  22  
AC\_VIEW\_SERVER\_SOCKET\_BASE  
  setting 174  
AC\_VIEW\_SERVER\_SOCKET\_COUNT  
  setting 174  
AC\_WEBAGENT\_AUTHENTICATION\_PR  
  OMPT setting 152  
AC\_WEBAGENT\_CONNECTION\_CACHE\_  
  SIZE setting 170  
AC\_WEBAGENT\_CONNECTION\_CACHE\_  
  TIMEOUT setting 170  
AC\_WEBAGENT\_HOST setting 176, 177  
AC\_WEBAGENT\_PORT setting 176, 177  
AC\_WEBAGENT\_SECURITY\_LIBRARY  
  setting 157, 163, 164  
AC\_WEBAGENT\_UNAUTHORIZED\_STAT  
  US\_CODE setting 153  
access restrictions 166  
access rights 91, 120, 121  
  *See also* privileges

accessing  
  e.Reporting Server 2, 153  
  Report Encyclopedia in web  
    environments 4–7  
  Report Encyclopedia items 14  
  ReportCast channels 6  
  ReportCast objects 47  
  ReportCast templates 43–44  
accessing online help xiii  
\_\_accessToGrant variable 91  
.ACHTML files 40  
ACSERVERALIAS property 134  
\_\_active directive 74, 98  
active request  
  displaying URL for 133  
  getting 114  
Active template 55  
Activedetail template 55  
ActiveRequestsURL variable 133  
ActiveX controls x  
Actuate  
  home page xi  
  release notes xi  
Actuate Basic  
  syntax conventions xviii  
#Actuate keyword 107  
Actuate LRX ix  
  viewing reports 83, 175  
Actuate product summary viii  
Actuate Scripting Language  
  command reference 106–131  
  command separator 76  
  command types listed 106  
  commands listed 107  
  directive formats 70–72  
  directives described 72–102  
  directives listed 73–75  
  getting number of processed  
    commands 132, 133  
  overview 106  
  template mappings for 56  
  variables described 131–146  
acweb keyword 68, 71

- AcWebSecurityAuthenticate function 153, 158
  - overview 157
- AcWebSecurityStart function 158
  - overview 157
- AcWebSecurityStop function 160
  - overview 157
- ad hoc parameters 144
- AdHoc variable 144
- \_\_?admin directive 73, 74, 98
- \_\_admin keyword 71
- Admin template 55
- ADMIN.ACHTML 42
- administration page 42, 132
  - implementing security from 150
- administrative groups 134
- administrative security 148
- administrators
  - setting user name and passwords 150
- AdminURL variable 132
- Advanced e.Reporting Server ix
- after keyword 108
- \_\_ageDays variable 94
- \_\_ageHours variable 94
- AgentVersion variable 132
- aging expiration policies 126
  - channel 145
- Alias variable 144
- aliases 134, 144
- analysis tools (third-party) x
- anonymous users 150
- application deployment 178
- application programming interfaces x
- \_\_archiveBeforeDelete variable 94
- \_\_archivePolicy variable 94
- ascending sort order 76, 111
- assigning values to variables 130
- authentication 148
  - bypassing 150
  - default 153, 154
  - enabling 158
  - failing 153
  - overriding 155, 157
  - process described 154, 157
- authentication filters 162
- authoring tools 49
- automating reporting tasks x

## B

- Base URL 29
- Basic Authentication header 154
- before keyword 108
- bookmarks 78
- border colors 64
- browsing 170
- BundleRox variable 91
- bundling flag 91
- buttons 68

## C

- cache 132, 151
- cached connection management 167, 170
- cached connection management for 167, 170
- Cancel template 55
- CANCEL.ACHTML 41
- canceled requests 10, 95
- Cancelled Request Confirmation page 41
- CancelURL variable 140, 141
- cascading style sheets 81
- case sensitivity
  - file types 111
  - report parameters 91
  - search results 83
  - variables 91
- case statements 106
- CGI scripts 27
- changing
  - border colors 64
  - copyright information 65
  - passwords 11
  - ReportCast template files 50, 54
    - example for 61, 63
  - sort orders 111
  - unauthorized access status 153
- channel administration page 42
  - displaying 100
- Channel Contents page 42
- channel creation form 33
- channel creation page 33
- Channel Detail page 42
- channel directives 72, 98–100
- channel objects 45
- channel parameter variables 101
- Channel Subscription page 14, 42

- channel subscription variables 102
- channel URLs 100–102
- channel variables 145
- Channeladmin template 55
- CHANNELADMIN.ACHTML 42
- Channelcontents template 55
- CHANNELCONTENTS.ACHTML 42
- Channeldetail template 55
- CHANNELDETAIL.ACHTML 42
- Channelitemdetail template 55
- \_\_ChannelName variable 101
- ChannelName variable 145
- \_\_channels variable 93
- channels ix, 4, 29–34, 167, 170
  - accessing 6
  - administering 30, 99
  - changing 34
  - checking contents 14
  - creating 33–34
  - deleting 34
  - displaying contents 100
  - getting current subscriptions 123
  - getting granted permissions 120, 121
  - getting information 145
  - getting URLs 134, 145
  - listing available 99
  - overview 29
  - subscribing to 14, 101
  - testing current subscriptions 145
  - unsubscribing from 14
  - updating contents 34
- channels administration page 31, 33
- channels keyword 109
- channels variable 102
- Channelsubscription template 55
- CHANNELSUBSCRIPTION.ACHTML 42
- ChannelsURL variable 134
- ChannelURL variable 145
- character strings
  - assigning to variables 130
- client validation 153
- colors
  - changing border 64
- command separator 76
- CommandName variable 133
- commands 3
- commands *See* Actuate Scripting Language
- \_\_Comment variable 101
- Comment variable 139, 146
- communications processes 167
- communications protocols 170
- company names and logos
  - example for creating 63
- \_\_completed directive 74, 98
- completed requests
  - displaying URLs for 133
  - getting date/time 141
- CompletedRequestsURL variable 133
- CompletedTime variable 141
- componentID keyword 79, 82
- ComponentID variable 135, 136
- conditional expressions 128
- conditional statements 106
- configuration-specific templates 43
- configuring communications ports 167, 169, 170, 171, 174
- configuring ReportCast ports 21
- configuring secured internet environments 178
- configuring security extensions 163
- Confirm template 55
- CONFIRM.ACHTML 41
- Confirmation page 42
- connections
  - getting 124
  - Report Encyclopedia 170, 176
  - securing 167
  - setting registered ports for 168, 169
- content keyword 108
- contents page 7
  - displaying for DHTML reports 81
  - getting information about 135
  - listing DHTML items in 126
  - testing for sub-items 135
- context-sensitive help xiii
- ControlName variable 135
- copyright information
  - changing 65
- Created variable 138
- creating
  - custom copyright information 65
  - customized company logos 63
  - customized Requester forms 53
  - ReportCast templates 47, 49
  - web pages 40
- creating report-specific online help xvi
- criteria keyword 109
- current date 132

- current folder variables 137
- current request variables 139
- current time 132
- CurrentPage variable 135
- CurrentPages variable 142
- CurrentSize variable 142
- CurrentTOCParentID variable 135
- custom default forms 40
- custom directory 49
  - adding templates to 43
- Custom Folder page 41
- custom Requester forms 86, 115
  - attaching to reports 53
  - creating 53
  - examples of 87, 88
- custom templates 40, 49, 52
  - adding to Encyclopedia 57
  - examples 61–65
  - naming 55
- custom web pages 57

## D

- data protection 166–167
- data reply connections 168, 171
- data reply requests 173
- date formats 91
- DateAndTime variable 143
- dates
  - getting 132, 141
  - getting for status information 143
  - last modification 137, 138, 140
  - setting default 140
- \_\_dateToDelete variable 94
- \_\_default keyword 71
- default directories 47
- default forms 40
- default reporting web site 2
- default settings (system-wide) 134
- default templates 3, 49
- default user authentication 148, 153, 154
  - overriding 155, 157
- Default variable 144
- DefaultDate variable 140
- DefaultTime variable 140
- ?DeleteStatus directive 73, 96
- DeleteURL variable 141
- deleting
  - status information 96, 141
- deleting ReportCast channels 34
- demand paging 84
- DemandCachePageSize registry key 36
- dependencies
  - ReportCast template files 43
- deploying online reports 3
- deployment 178
- desc keyword 111
- descending sort order 76, 111
- Description variable 144
- ?Detail directive 73, 97
- DetailURL variable 141
- Developer Workbench viii
- DHTML reports 2
  - displaying 11, 78–82, 173
  - displaying contents page 81
  - displaying navigation bars for 82
  - expiring 79
  - generating navigation bar for 78
  - getting available versions 112
  - getting content for 79
  - getting information on contents 135
  - getting information on current views 135
  - getting style sheets for 81
  - listing items in contents page 126
  - scaling 79
  - searching 11, 12, 82, 136
- ?Direction directive 73
  - issuing 76–78
- directives *See* Actuate Scripting Language
- directories
  - displaying full path names 138, 139, 141
  - returning paths for 124, 137
- directory structures
  - setting for ReportCast templates 47
- displaying
  - available channels 99
  - current users 133
  - DHTML reports 11, 78–82, 173
  - Encyclopedia contents 14, 113
  - folder contents 7, 14, 76, 112, 113
  - folder lists 112, 113
  - HTML source files 58
  - navigation bar 82
  - online documentation xiii
  - report files 77

- reports 2, 14, 83, 171
- request details 97
- status information 96
- Table of Contents 81
- URLs 69
- DLLs 157
- document files
  - getting path links for 124
- Document Source command 58
- documentation x, xi
  - online xii
  - syntax conventions xviii
  - typographical conventions xvii
- downloading Actuate Viewer LRX 11
- downloading files 83, 85
- downloading report documents 73
- DownloadURL variable 140
- ?drop directive 73, 74, 99
  - usage example 95
- dual firewalls 179, 181, 182–183
- dynamic HTML *See* DHTML

## E

- e.Report Designer viii
- e.Reporting Server ix
  - accessing 2, 153
  - caution for setting up 184
  - downloading files 83, 85
  - getting connections 124
  - getting current state 119, 142
  - getting information about 133
  - logging onto 150–151
  - overview xi
  - reporting error conditions for 146
  - securing for networked environments 166
  - setting template dependencies for 43
- e.Reporting Server objects 45
- e.Reporting Server port 167, 168
  - configuring 169–170
- e.Reporting Server variables 133
- e.Reporting Suite vii
  - product summary viii
- editing tools 49
- elapsed time 132, 142
- ElapsedTime variable 132, 142
- end list keyword 108
- End User Desktop viii
- EndHitNumber variable 136

- Entries variable 139
- enu directory 48
- environment variables 149
- error codes and numbers 146
- error messages 146
- error objects 46
- Error page 42
- Error template 56
- error variables 146
- ERROR.ACHTML 42, 153
- ErrorClass variable 146
- ErrorNumber variable 146
- ErrorText variable 146
- \_\_exclude variable 93
- executable files
  - attaching Requester forms to 53
  - getting full paths 141
  - getting information about 139
  - getting version 141
  - running 8, 43
- executing report executables 8, 43
- executing reports 85, 171
  - creating completed notices for 122
  - creating user-specified requests for 115, 117
  - getting schedules for 122
- Expandable variable 135
- \_\_Expiration variable 101
- expiration policies
  - channel 145
  - report files 126
- Expiration variable 145
- Expirationpolicies keyword 109
- expressions 128
- ?extractSearchResults directive 75, 83

## F

- F1 key xiii
- Factory directives 72, 85–89
- Factory requests
  - getting information about 98
- file creation dates/time 138
- file dependencies
  - ReportCast templates 43
- File Detail page 41
- file directives 72, 83–85
- file lists 110
  - sorting items 110, 111

- file names
  - custom ReportCast templates 55
  - getting extension 138, 141
- file types
  - getting 138
  - listing 111
- File Versions page 42
- FileDetail template 56
- FILEDETAIL.ACHTML 41
- ?Files directive 73
- files
  - See also* output files; template files
  - attaching Requester forms to 53
  - changing ReportCast template 50, 54
    - example for 61, 63
  - creating ReportCast template 49
  - downloading 83, 85
  - generating from conditions 128
  - generating report 7–10, 85
  - generation request forms for 115, 117
  - getting aging expiration 126
  - getting available versions 112
  - getting information 138, 139
  - getting path links for 124
  - getting size 138
  - renaming template 51
  - running report executables 8, 43
  - viewing in folders 77
- files keyword 108
- FileSize variable 138
- FileType variable 138
- FileVersion variable 138
- FileVersionName variable 138
- FileVersions template 56
- FILEVERSIONS.ACTML 42
- finding data
  - in DHTML reports 11, 12, 82, 136
- finding template files 51
- firewalls
  - defined 166
  - HTTP traffic and 178
  - setting up 178–183
- folder contents pages 7
- folder directives 72, 75–78
- folder item objects 45
- folder item variables 137
- Folder Listing page 41
- folder objects 45
- folder variables 137, 139
- FolderHeader variable 137
- Folderlist template 56
- FOLDERLIST.ACHTML 41
- FolderName variable 137, 139
- FolderPath variable 137
- folders
  - displaying contents 7, 14, 76, 112, 113
  - displaying current 112
  - displaying report files only 77
  - displaying URLs for 134
  - getting file lists for 110
  - getting information 139
  - getting information on current 137
  - getting information on items 137
  - getting paths for 124, 137
  - modification dates/time 137, 138
  - setting templates dependencies for 43
  - sorting items 76, 110, 113
- folders keyword 109
- FolderTitle variable 137
- forced reloads 6, 151
- format keyword 82
  - valid values 80, 81
- format preferences 134
- formatting ReportCast directives 70–72
- forms 40
  - customizing Requester 53
  - setting dependencies for 43
  - setting up Requester 40

## G

- generating report files 7–10, 85
  - creating completed notices for 122
  - creating user-specified requests for 115, 117
  - getting schedules for 122
  - with conditions 128
- generating web pages 106, 131
- ?GetDynamicData directive 74, 79
- ?GetReportData directive 74, 81
- ?GetStyleSheet directive 74, 81
- globally unique IDs 138, 140
- Granted variable 143
- grants keyword 109

- GroupName variable 143
- \_\_groups variable 93
- groups
  - generating lists for 119, 122, 142
  - getting admin status for 134
  - getting information 142
  - getting report parameter 143
- GUIDs 138, 140

## H

- Headline variable 91, 141
- help xiii–xvii
  - creating report-specific xvi
  - usage overview xiii
- help directories 49
- Help menu xiii
- help topics xiii
- hidden parameters 144
- Hidden variable 144
- HideText variable 144
- hierarchical paths 137
- \_\_home directive 73
- home folder 134
- HomeFolderURL variable 134
- Host variable 132
- hosts 132, 133
- HTML authoring tools 49
- HTML pages 2
  - creating 40
  - generating 106, 131
  - generating from conditions 128
  - predefined 41, 43
  - viewing 84
- HTML reports 2
- HTML requester forms 40
  - See also* Requester forms
  - attaching to reports 53
  - customizing 53, 86, 115
  - examples of 87, 88
  - filling out 9
  - generating 85
  - selecting 54
- HTML scripting language *See* Actuate Scripting Language
- HTML source files 58
- HTML template files
  - caching 132

- changing 50, 54
  - example for 61, 63
- creating 49
- defining variables in 131
- getting names 133
  - listed 41
- loading 128
- naming 55
- overview 40
- reloading 54, 55
- HTML templates 3
  - accessibility 43–44
  - adding to Encyclopedia 57
  - associating objects with 47
  - customizing 49, 52
    - examples 61–65
  - elements described 49
  - mapping to directives 56
  - overview 40, 43
  - referencing subtemplates in 51
  - search paths 44, 52
  - searching for 51
  - setting directory structures 47
- HTTP communications protocol 70
- HTTP unauthorized status codes 153
- HTTP/HTTPS protocols 178
- HTTPS support 148
- hyperlinks 4
  - See also* links
- hypertext markup language *See* HTML
- HyperText Transfer Protocol *See* HTTP

## I

- icons 33
  - getting URLs for 146
- If command 107, 128
- ifempty keyword 108
- images directory 49
- implementing security 147
- inbound ports 167
- Include command 107, 128
- Index template 56
- INDEX.ACHTML 41
- information directives 97–98
- initialization 158
- InputObject constant 129
- installation defaults 47

- installing online documentation xiii
- internationalization 49
- Internet *See* web environments
- Internet Explorer
  - displaying links 69
  - reloading pages 6
  - reloading web pages 151
- Internet Information Server (IIS) 162
- IP addresses
  - displaying dotted-decimal 133
- IsAdministrator variable 134
- ISAPI filters 163
- IsBundledOutput variable 138
- IsExecutable variable 138
- ItemBaseName variable 138
- ItemName variable 138
- ItemPathName variable 138
- items keyword 109
- iterating through lists 107
- iteration scripting commands 106

## J

- jpn directory 48

## K

- keywords 108
  - See also specific*

## L

- language support 49
- large icons 146
  - \_\_LargeIconURL variable 101
- LargeIconURL variable 146
- LastModified variable 137, 138, 140
- LastModifiedBy variable 138
- LatestURL variable 138
- libraries 157
  - \_\_limit variable 93
  - \_\_limitNumber variable 93
- LinkedFolderPath variable 137, 139
- links
  - as navigation tools 4
  - associated with ReportCast channels 14
  - associated with ReportCast directives 68
- List channels command 123
- List command 107–110

- keywords listed 108
- overview 107
- List criteria command 126
- List expirationpolicies command 126
- List files command 110–112
  - specifying file type 111
- List folders command 112
- List grants command 120
- List items command 113
- list keyword 108
- List members command 119
- List notifications command 122
- List parameters command 115
- List paramgroups command 115
- List pathlinks command 124
- List printerprops command 123
- List privileges command 121
- List reportservers command 124
- List requestforms command 113
- List requests command 114
- List resultfields command 127
- List resulthits command 127
- List schedules command 122
- list statements 129
- List status command 118
- List tocitems command 126
- List values command 117
- List versions command 112
- lists
  - available file versions 112
  - channel subscriptions 123
  - current users 119, 122, 142
  - defining 108
  - DHTML contents 126
  - directory paths 124
  - folder contents 110
  - folders 112, 113
  - iterating through 107
  - object expiration 126
  - parameter groups 116
  - permissions status 120, 121
  - printer properties 123
  - report parameters 115, 117
  - request schedules 122
  - request status 118
  - Requester forms 113
  - requests 114, 140

- search criteria 127
- search hits 127
- server state 119, 142
- Live Report Extension (LRX)
  - viewing reports 83, 175
- \_\_local keyword 71
- locales 48
- locating data
  - in DHTML reports 11, 12, 82, 136
- locating template files 51
- Login dialog box 4
  - anonymous users 150
  - customizing 152
  - specifying unauthorized status for 148
- login scripts
  - anonymous users 150
  - failing 153, 155, 157
- logos
  - example for creating 63
- loops 107
- LRX (Live Report Extension) ix
  - viewing reports 83, 175

## M

- Manuals directory xiii
- Masking URLs 28
- masks 144
- member objects 45
- member variables 142
- MemberName variable 142
- members keyword 109
- MemberType variable 142
- mode keyword 80
- modification dates/time
  - folders 137, 138
  - report executables 140
- moving through Report Encyclopedia 4, 14
- multiple sort fields 111
- My Headlines tab 6

## N

- Name variable 144, 145
- naming custom template files 55
- NAT Static Address Translation
  - algorithm 181
- navigating through Report Encyclopedia 4, 14

- navigation bar
  - displaying 82
  - generating 78
  - searching from 83
  - viewing DHTML pages from 79
- navigation buttons 84
- Navigation page 42
- Navigator (Actuate) 4
- Netscape Navigator
  - displaying links 69
  - implementing security for 162
  - reloading web pages 151
- Network Address Translation (NAT) 167, 178, 181
  - proxy servers and 183
- network communications 167–177
- network security 166
- network security configuration options 178
- NewItem variable 139
- NextPollTime variable 145
- NextStartTime variable 142
- non-default web server ports 21
- notification groups 93
- notification lists 119, 122, 142
- notifications 172
- notifications keyword 109
- Now variable 132
- NumberOfCommands variable 132
- NumberOfHitsOnPage variable 136
- NumVersions variable 139

## O

- object IDs 138
- object lists 110
  - sorting items 110, 111
- ObjectID variable 138
- objects 45
  - accessibility 47
  - associating with ReportCast templates 47
  - getting detail information 97
  - getting expiration policies for 126
  - getting granted permissions for 120
  - getting URLs 133
  - initializing values for 129
- ObjectURL variable 133
- on demand paging 84
- \_\_onceDate variable 91
- \_\_onceTime variable 92

- online documentation xi, xii
  - syntax conventions xviii
  - typographical conventions xvii
- online help xiii–xvii
  - creating report-specific xvi
  - usage overview xiii
- online reports vii
  - See also* reports
  - deploying 3
  - getting current 5
  - getting specific version 85
  - overview 7
- open server ix
- operation keyword 80
- outbound ports 167
- output files 139
  - getting location 141
  - getting number of pages 142
  - getting size 142
  - getting version 142
- OutputLocation variable 141
- \_\_outputname variable 93
- OutputName variable 141
- OutputObject constant 130
- OutputPathName variable 141
- OutputVersion variable 142
- OutputVersionName variable 142
- \_\_overwrite variable 92
- OwnerName variable 138
- owners 138

## P

- page keyword 78, 82
- page ranges 142
- Pages variable 142
- parameter group objects 45
- parameter group variables 143
- parameter groups
  - displaying 143
  - generating lists of 116
- parameter objects 45
- parameter values files 142
- parameter variables 91, 101, 143
- parameters 85
  - defining in directives 91
  - displaying default 144
  - generating requester forms for 86
  - getting available 115

- getting information 143
- masking values 144
- setting 117
- specifying default 115
- parameters keyword 109
- ParamGroups keyword 109
- Parent URL variable 137
- ParentFolderPath variable 137
- parsed-header CGI scripts 27
- passwords
  - anonymous users 150
  - changing 11
  - setting 150
- path names
  - current folder 124, 137
  - executable files 141
  - folder items 138
  - output files 141
  - report files 139
- pathlink keyword 109
- PathName variable 138
- PDF files xi, xiii
  - viewing 173
- performance tuning 36
- permissions 91, 120, 121
  - See also* privileges
- personal channel web page 6
- Personalchcontents template 56
- PlainComment variable 146
- \_\_PollingInterval variable 101
- PollingInterval variable 145
- port mapper 168
- ports 167, 178
  - configuring for ReportCast 21
  - registering for secured connections 168, 169
- predefined ReportCast templates 3, 49
- predefined web pages 41, 43
- preferences
  - getting 134
  - setting 10
- PreviousURL variable 133
- printed documentation x, xi
  - syntax conventions xviii
  - typographical conventions xvii
- Printer property objects 45
- printer variables 145
- PrinterProps keyword 109

- printers
  - generating property lists 123
  - getting information for current 145
- printing reports 142
- PrintRange variable 142
- \_\_priority variable 92
- Priority variable 141
- \_\_priorityValue variable 92
- privilege variables 143
- PrivilegeName variable 143
- privileges 143
- privileges keyword 109
- process groups 142
- ProcessGroup variable 142
- product summary viii
- product updates xi
- programming interfaces x
- property lists 123
- protection *See* security
- providing report-specific online help xvi
- proxy servers 167
  - dual firewall configurations for 182–183
- push technology 4

## R

- RawErrorText variable 146
- RawPollingInterval variable 145
- RCSE *See* ReportCast Security Extension
- recurring schedules 92
- \_\_recurringTime variable 92
- refreshes 145
- Regedit 148
- regedt32 37
- registry keys 148
  - DemandCachePageSize 36
- release notes xi
- reloading web pages 6, 151
- RemoteAddress variable 133
- RemoteHost variable 133
- renaming
  - template files 51
- report definition objects 45
- report definition variables 139
- report definitions
  - getting 139
- Report Encyclopedia
  - accessing in web environments 4–7
  - adding custom templates 57
  - connecting to 170, 176
  - displaying folder contents 7, 14, 76, 112, 113
  - displaying report files only 77
  - getting detail information 97
  - getting granted permissions 120, 121
  - getting item expiration lists 126
  - listing items 110
  - logging onto 4
  - navigating through 4, 14
  - sorting folder contents 76, 110, 113
  - updating items 6
- Report Encyclopedia web page 4
- report files
  - downloading 83, 85
  - generating 7–10, 85
  - generating from conditions 128
  - getting aging expiration 126
  - getting available versions 112
  - getting information 138, 139
  - getting size 138
  - viewing 77
- report generation requests
  - canceling 10
  - creating completed notifications for 122
  - generating schedules for 122
  - getting status 95
  - setting conditions for 128
  - submitting 9, 85, 89, 90
  - user-specified 115, 117
- report server API x
- <report URL> directive 73
- report viewing options 10–13
- Report Viewing Options page 42
- ReportCast
  - accessing using parsed-header CGI scripts 27
  - configuring for UNIX environments 183–184
  - configuring port numbers for 21
  - dual firewall configurations for 179, 181
  - getting information about 131
  - overview 1, 2
  - setting up 3
  - starting and stopping 18

- using CGI scripts to access 27
  - viewing options 10–13
- ReportCast channel directives 98–100
- ReportCast channel objects 45
- ReportCast channel URLs 100–102
- ReportCast channel variables 145
- ReportCast channels ix, 4, 29–34
  - accessing 6
  - administering 30, 99
  - cached connection management for 167, 170
  - changing 34
  - checking contents 14
  - creating 33–34
  - deleting 34
  - displaying contents 100
  - getting current subscriptions 123
  - getting granted permissions 120, 121
  - getting information 145
  - getting URLs 134, 145
  - listing available 99
  - overview 29
  - subscribing to 14, 101
  - testing current subscriptions 145
  - unsubscribing from 14
  - updating contents 34
- ReportCast commands 106–131
  - listed 107
  - overview 106
  - syntax 107
  - types listed 106
- ReportCast directives
  - examples 102
  - formatting 70–72
  - getting associated link 68
  - issuing 68
  - listed 73–75
  - mapping to templates 56
  - overview 68, 69
  - syntax 75
  - types described 72–102
- ReportCast objects 45–47
  - accessibility 47
  - getting detail information 97
- ReportCast scripting language *See* Actuate Scripting Language
- ReportCast Security Extension (RCSE) 148
  - administrative tasks 150
  - changing unauthorized access status 153
  - configuring 163
  - customizing Login dialogs for 152
  - development guidelines 160–163
  - enabling 148–149
  - overriding default authentication 155, 157
  - overview 153
  - testing 160
- ReportCast server start-up script 19
- ReportCast template files
  - caching 132
  - changing 50, 54
    - example for 61, 63
  - creating 49
  - defining variables in 131
  - getting names 133
  - listed 41
  - loading 128
  - naming 55
  - overview 40
  - reloading 54, 55
- ReportCast templates
  - accessibility 43–44
  - adding to Encyclopedia 57
  - associating objects with 47
  - customizing 49, 52
    - examples 61–65
  - default 3, 49
  - elements described 49
  - mapping to directives 56
  - overview 40, 43
  - referencing subtemplates in 51
  - search paths 44, 52
  - searching for 51
  - setting directory structures 47
- ReportCast variables 131
- ReportDefinition constant 129
- report-generation requests *See* Requester forms; requests
- Reporting Server
  - accessing 2, 153
  - caution for setting up 184
  - downloading files 83, 85
  - getting connections 124
  - getting current state 119, 142
  - getting information about 133
  - logging onto 150–151
  - overview xi

- reporting error conditions for 146
- setting template dependencies for 43
- Reporting Server objects 45
- Reporting Server variables 133
- reporting sites 2
  - getting current report 5
- reporting solutions vii
- ReportName variable 139, 141
- ReportOutputPath variable 139
- ReportPath variable 139
- ReportPathName variable 141
- ?Reports directive 73
- reports vii
  - creating online help for xvi
  - deploying 178
  - deploying over web environments 3
  - displaying in secured environments 171, 173
  - displaying in web environments 2, 14, 83
  - getting current 5
  - getting information about 139
  - getting specific version 85
  - overview 7
  - running 85, 171
  - viewing 14
- ReportServerName variable 132
- ReportServers keyword 109
- ReportURL variable 140, 141
- ReportVersion variable 141
- ReportVersionName variable 141
- ?Request directive 73, 85
- Request Confirmation page 41
- request confirmation page 117
- Request Detail page 41
- Request folder objects 45
- Request Folder page 41
- request folders 139
- request forms
  - user-specified report generation 115, 117
- request headlines 141
- request IDs 140, 141
- request item variables 140
- request objects 45
- request parameters 85
- request status page 41, 118
- request status variables 143
- Request template 56
- REQUEST.ACHTML 41
- Requestdetail template 56
- REQUESTDETAIL.ACHTML 41
- RequestedBy variable 140
- Requester API x
- Requester forms
  - attaching to reports 53
  - customizing 53, 86, 115
  - examples of 87, 88
  - filling out 9
  - generating 85
  - listing available 113
  - selecting 54
  - setting up 40
- Requester Menu page 41
- Requestermenu template 56
- REQUESTMENU.ACHTML 41
- RequestForms keyword 109
- RequestID variable 140
- Requestlist template 56
- REQUESTLIST.ACHTML 41
- requests 176
  - canceling 10, 95
  - data reply 173
  - deleting 96
  - displaying URLs for 134
  - executing 89
  - generating notification lists for 119, 122, 142
  - generating server-specific 114
  - getting elapsed time 132, 142
  - getting information 140
  - getting information about Factory 98
  - getting number 139
  - getting priority 141
  - getting scheduled date/time 142
  - getting schedules 122, 144
  - getting status 10, 95, 141, 143
  - getting type 140
  - submitting for DHTML searches 83
  - submitting for report generation 9, 85, 89, 90
  - time out interval 5
  - viewing details about 97
- requests keyword 109
- ?requestSearch directive 75, 82
- RequestStatus variable 141
- RequestsURL variable 134
- RequestType variable 140

- RequestURL variable 140, 141
- \_\_requestUserPreferences directive 74
- required parameters 144
- Required variable 144
- resetting unauthorized status 153
- restricting data access 166
- Resultfields keyword 110
- Resulthits keyword 110
- ?retrieveCreation directive 74, 99
- ?retrieveDetails directive 74, 99
- ?retrieveForm directive 74, 98
- \_\_Roles variable 101
- roles
  - getting current 119, 142
  - getting granted permissions 120, 121
- root folder 134
- RootFolderURL variable 134
- routers 182
- RPC protocol 169
- running report executables 8, 43
- running reports 85, 171
  - creating completed notices for 122
  - creating user-specified requests for 115, 117
  - getting schedules for 122
- run-time access 131

## S

- scalingFactor keyword 79, 80, 81
- schedule objects 45
- schedule variables 144
- \_\_scheduled directive 74, 98
- scheduled requests
  - displaying URLs for 133
  - getting start date/time 142
- Scheduled template 56
- Scheduleddetail template 56
- ScheduleDescription variable 144
- ScheduledRequestsURL variable 133
- \_\_schedulePeriod variable 92
- schedules
  - displaying 144
  - listing 122
- schedules keyword 109
- \_\_scheduleType variable 91
- scope

- ReportCast templates 43
- scripting language
  - command reference 106–131
  - command separator 76
  - command types listed 106
  - commands listed 107
  - directive formats 70–72
  - directives described 72–102
  - directives listed 73–75
  - getting number of processed
    - commands 132, 133
  - overview 106
  - template mappings for 56
  - variables described 131–146
- search criteria 83
  - getting information about 135
  - getting list of 127
  - removing 13
  - specifying 12, 82
- search criterion variables 135
- search directives 72, 82
- search extension API x
- search hits 127
  - getting information on 136, 137
- search paths 44, 52
- Search Request page 12
  - displaying 82
- search requests
  - submitting 83
- search result field variables 137
- search result hit variables 136
- search result set variables 136
- searches (help topics) xv
- searching DHTML reports 11, 12, 82, 136
- SearchValue variable 136
- secure network environment 166
- Secure Sockets Layers 70, 148
- secured servers 148
- security
  - administrative tasks 150
  - changing unauthorized access status 153
  - configuring 163
  - customizing Login dialogs for 152
  - data protection strategies for 166–167
  - guidelines for developing 160–163
  - implementing 147

- network communications and 167–177
- overview 148, 153, 165
- server-specific information 162
- testing 160
- security configuration options 178
- security library 157
  - closing 160
  - declaring functions in 160
  - example code 161
  - initializing 158
  - loading 158
  - specifying 163, 164
- security variables 148–149
  - setting for UNIX systems 163
  - setting for Windows NT 148, 163
- Selected variable 136
- selection scripting commands 106
- sending ReportCast directives 68
- separator character 76
- sequential scripting commands 106
- server objects 45
- Server variable 133, 146
- ServerAlias variable 134
- ServerFeatureLevel variable 134
- servers *ix*
  - defining configuration-specific templates 43
  - getting active requests 114
  - getting connections 124
  - getting current state 142
  - getting information about 132, 133
  - getting system-wide defaults 134
  - logging onto 150–151
  - overview to e.Reporting *xi*
  - reporting error conditions for 146
  - securing 148
- ServerVersion variable 133
- Set command 107, 130–131
- setting ReportCast template dependencies 43
- setting user preferences 10
- Size variable 142
- small icons 146
- \_\_SmallIconURL variable 101
- SmallIconURL variable 146
- Software Development Kit *x*
- ?Sort directive 73
  - issuing 76–78
- sort by clause 111
- sort fields 77, 111
- sort order
  - changing 111
  - setting 76, 111
- sorting
  - folder items 76, 110, 113
  - on multiple fields 111
  - reports in file lists 110, 111
- source files
  - displaying HTML 58
- spaces in string values 91
- split ReportCast communications 176
- Split ReportCast configuration 183–184
- standard directory 49
- Standard Requester page 41
- StartDateAndTime variable 132
- StartHitNumber variable 136
- Starting and stopping NT ReportCast 18
- Starting and stopping UNIX ReportCast 19
- StartTime variable 141
- startwebagent.sh 19, 149
- ?Status directive 73
- status codes
  - unauthorized access 148, 153
- status directives 72, 95–97, 98
- status information 95
  - deleting 141
  - deleting requests for 96
  - displaying 96
  - generating lists for 118
  - getting 10, 95, 141, 143
- status keyword 109
- Status template 56
- STATUS.ACHTML 41
- StatusText variable 143
- StatusURL variable 140, 141
- ?StdRequest directive 73, 85
- strings
  - assigning to variables 130
  - spaces in 91
- style sheets 81
- ?Submit directive 73, 89–95
- Submit buttons 90
- submit parameter variables 91
- submit URLs 90
- ?submitCreation directive 74, 99

- ?submitDetails directive 74, 99
- ?submitForm directive 74, 98
- ?submitSearch directive 75, 83
- submitting requests
  - for DHTML searches 83
  - for report generation 9, 85, 89, 90
- \_\_submitUserPreferences directive 74
- Subscribed variable 145
- subscribing to channels 4
  - getting subscribed users 123
  - testing current subscriptions 145
- Subscription page 14
- subscription variables 102
- subscriptions 14, 29, 101
- subtemplates 43
  - renaming 51
- Success template 56
- SUCCESS.ACHTML 42
  - source code for 58
- syntax
  - channel directives 98, 99, 100, 101
  - channel URLs 100
  - Factory directives 85
  - file directives 83, 84
  - GetDynamicData directive 79
  - GetReportData directive 81
  - GetStyleSheet directive 81
  - if command 128
  - include command 128
  - information directives 97, 98
  - list command 107
  - ReportCast commands 107
  - ReportCast directives 70, 75
  - requestSearch directive 82
  - scripting variables 131
  - set command 130
  - sort and direction directives 76
  - status directives 95, 96
  - Submit directive 89, 90
  - Versions directive 78
  - ViewDefault directive 82
  - ViewFrameset directive 78
  - ViewNavigation directive 82
  - ViewPage directive 79
  - ViewTOC directive 81
    - with command 129
- syntax conventions (documentation) xviii

- system dates/time 132
- system-wide defaults 134

## T

- Table of Contents 42
  - displaying for DHTML reports 81
  - listing DHTML items in 126
  - testing for sub-items 135
- taglines 91
- TCP/IP port mapper 168
- template files
  - caching 132
  - changing 50, 54
    - example for 61, 63
  - creating 49
  - defining variables in 131
  - getting names 133
  - listed 41
  - loading 128
  - naming 55
  - overview 40
  - reloading 54, 55
- TemplateLoadDateAndTime variable 132
- TemplateName variable 133
- templates 3
  - accessibility 43–44
  - adding to Encyclopedia 57
  - associating objects with 47
  - customizing 49, 52
    - examples 61–65
  - elements described 49
  - mapping to directives 56
  - overview to ReportCast 40, 43
  - referencing subtemplates in 51
  - search paths 44, 52
  - searching for 51
  - setting directory structures 47
- templates directory 43
- Text variable 135
- third-party analysis tools x
- time
  - getting current 132
  - getting elapsed 132, 142
  - getting for status information 143
  - getting start 132, 141, 142
  - last modification 137, 138, 140
  - setting default 140

- time formats 92
- time out interval 5
- \_\_timeToDelete variable 94
- titles 137
- TOC item objects 46
- TOC item variables 135
- Tocitems keyword 109
- Today variable 132
- TotalNumberOfHits variable 136
- TotalPages variable 135
- tuning performance 36
- Type variable 144
- typographical conventions xvii
  - syntax xviii

## U

- unauthorized status codes 148, 153
- unauthorized users 166
- Uniform Resource Locators *See* URLs
- UNIX environments
  - communications processes for 167
  - configuring security extension for 163
  - formatting ReportCast directives for 70
  - implementing security for 148, 149, 150
  - Report Encyclopedia connections for 176
  - Split ReportCast configuration for 183–184
- unsubscribing from ReportCast channels 14
- updates xi
- updating Encyclopedia items 6
- URL variable 133, 138
- URLs
  - as ReportCast directives 68
  - command separator for 76
  - displaying 69
  - displaying ReportCast channel 134
  - for detail pages 141
  - for Secure Sockets Layers 70
  - for submitting requests 90
  - getting admin 132
  - getting folder items 138
  - getting for canceled notifications 140, 141
  - getting for current channel 145
  - getting for current request 140, 141
  - getting for custom icons 146
  - getting for output 140, 141
  - getting for report executable 140
  - getting for report executables 141
  - getting for scheduled requests 133
  - getting for status page 140, 141
  - getting full 133
  - getting most recent file version 138
  - getting parent folder 137
  - getting previous 133
  - Report Encyclopedia 4
    - setting user preferences from 10
- user authentication 148
  - bypassing 150
  - default 153, 154
  - enabling 158
  - failing 153
  - overriding 155, 157
  - process described 154, 157
- user names
  - generating lists of 119, 122, 142
  - getting 133, 138, 140
  - setting 150
  - setting for notifications 93
- user preferences
  - getting 134
  - setting 10
- user preferences directives 72
- UserAgent variable 133
- user-defined report generation requests 115, 117
- UserName variable 133
- USERPREFERENCES.ACHTML 42
- \_\_users variable 93
- users
  - displaying current 133
  - generating notification lists for 119, 122, 142
  - getting active requests 114
  - getting admin status for 134
  - getting granted permissions 120, 121
  - getting subscribed channels 123
  - logging in as anonymous 150
  - logging in as different 151
  - testing for subscription status 145
  - unauthorized 166

## V

- Value variable 137, 144, 145
- values keyword 109
- ValuesFileName variable 142
- ValuesObject constant 130
- variable substitution 131
- variables
  - assigning values 130
  - directive-specific 91, 101, 102
  - initializing 129
  - scripting language 131–146
- version numbers
  - getting 132, 133, 138, 141, 142
  - specifying 85
- \_\_versionname variable 93
- ?Versions directive 73, 75, 78
- versions
  - getting available 112
  - getting specific report 85
  - testing 139
- versions keyword 108
- ?View directive 73, 84
- View Frameset page 42
- view process ports 168, 173
  - configuring 174
- view session objects 46
- view session variables 135
- ?ViewDefault directive 73, 82
- Viewer viii
  - downloading 11
- ViewFormatDefault variable 134
- ViewFormatPreference variable 134
- ?ViewFrameset directive 73, 78
- Viewframeset template 56
- VIEWFRAMESET.ACHTML 42
- viewing
  - available channels 99
  - current users 133
  - DHTML reports 11, 78–82, 173
  - Encyclopedia contents 14, 113
  - folder contents 7, 14, 76, 112, 113
  - folder lists 112, 113
  - HTML source files 58
  - navigation bar 82
  - online documentation xiii
  - performance 36

- report files 77
- reports 2, 14, 83, 171
- request details 97
- status information 96
- Table of Contents 81
- URLs 69
- viewing directives 72
- viewing options 10–13
- viewing preferences 134
- Viewnav template 56
- VIEWNAV.ACHTML 42
- ?ViewNavigation directive 74, 82
- ?ViewPage directive 74, 79
- ?ViewTOC directive 74, 81
- Viewtoc template 56
- VIEWTOC.ACHTML 42
- ViewURL variable 140, 141
- viruses 166

## W

- \_\_wait variable 91, 93
- \_\_WebAgent folder 43
- web browsers ix, 2
  - communications protocols for 170, 178
  - displaying links 69
  - getting host name for 133
  - getting IP addresses 133
  - refresh intervals 145
  - resetting unauthorized status 153
- web environments
  - accessing e.Reporting Server 2, 153
  - accessing Report Encyclopedia 4–7
  - deploying reports over 3
  - getting server information 132
  - implementing security 147
  - security configuration options for 178
  - setting up firewalls for 178–183
- web page templates 3
- web pages
  - changing border colors 64
  - creating 40
  - customizing 57
  - customizing copyright information for 65
  - generating 106, 131
  - getting current 5
  - predefined 41, 43

- reloading 6, 151
  - viewing document source for 58
- web request objects 45
- web request variables 133
- web server objects 45
- web server variables 132
- web servers 170
  - defining configuration-specific templates 43
- web sites
  - Actuate home page xi
  - default reporting 2
- WebServerVersion variable 132
- wildcards 111
- WINAPI calling convention 162

- Windows environments
  - communications processes for 167
  - configuring security extension for 163
  - implementing security for 148, 150, 162
- With command 107, 129–130
- World Wide Web *See* Web
- worms 166

## **X**

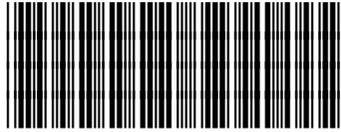
- XML files 174
- XML style sheets 81
- XMLDisplay format 82

## **Z**

- zooming percentage 79







\*100037162\*