

ContentDirector™ 2.0

API Guide

Network Appliance, Inc.
495 East Java Drive
Sunnyvale, CA 94089
Telephone: +1 (408) 822-6000
Fax: +1 (408) 822-4501
Support telephone: +1 (888) 4-NETAPP
Information email: info@netapp.com
Information Web: <http://www.netapp.com>

Part number 215-00124
November 2001

Copyright and trademark information

Copyright information

Copyright © 1994–2001 Network Appliance, Inc. All rights reserved. Printed in the U.S.A.

No part of this book covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Portions of this product are derived from the Berkeley Net2 release and the 4.4-Lite-2 release, which are copyrighted and publicly distributed by The Regents of the University of California.

Copyright © 1980–1995 The Regents of the University of California. All rights reserved.

Portions of this product are derived from NetBSD, which is copyrighted by Carnegie Mellon University.

Copyright © 1994, 1995 Carnegie Mellon University. All rights reserved. Author Chris G. Demetriou.

Permission to use, copy, modify, and distribute this software and its documentation is hereby granted, provided that both the copyright notice and its permission notice appear in all copies of the software, derivative works or modified versions, and any portions thereof, and that both notices appear in supporting documentation.

CARNEGIE MELLON ALLOWS FREE USE OF THIS SOFTWARE IN ITS “AS IS” CONDITION. CARNEGIE MELLON DISCLAIMS ANY LIABILITY OF ANY KIND FOR ANY DAMAGES WHATSOEVER RESULTING FROM THE USE OF THIS SOFTWARE.

Software derived from copyrighted material of The Regents of the University of California and Carnegie Mellon University is subject to the following license and disclaimer:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notices, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment:

This product includes software developed by the University of California, Berkeley and its contributors.
4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER

IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Portions of the software were created by Netscape Communications Corp.

The contents of those portions are subject to the Netscape Public License Version 1.0 (the “License”); you may not use those portions except in compliance with the License. You may obtain a copy of the License at <http://www.mozilla.org/NPL/>.

Software distributed under the License is distributed on an “AS IS” basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is Mozilla Communicator client code, released March 31, 1998.

The Initial Developer of the Original Code is Netscape Communications Corp. Portions created by Netscape are Copyright © 1998 Netscape Communications Corp. All rights reserved.

Software derived from copyrighted material of Network Appliance, Inc. is subject to the following license and disclaimer:

Network Appliance reserves the right to change any products described herein at any time, and without notice. Network Appliance assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by Network Appliance. The use and purchase of this product do not convey a license under any patent rights, trademark rights, or any other intellectual property rights of Network Appliance.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

Trademark information

NetApp and the Network Appliance design are registered trademarks of Network Appliance, Inc. in the United States, Canada, and the European Union. Network Appliance is a registered trademark of Network Appliance, Inc. in Monaco and a trademark of Network Appliance, Inc. in the United States and Canada. FAServer is a registered trademark of Network Appliance, Inc. in the United States and the European Union. NetCache is a registered trademark of Network Appliance, Inc. in the European Union and Japan, and a trademark of Network Appliance, Inc. in the United States. SnapCopy is a registered trademark of Network Appliance, Inc. in the European Union and a trademark of Network Appliance, Inc. in the United States. WAFL is a registered trademark of Network Appliance, Inc. in the European Union and a trademark of Network Appliance, Inc. in the United States and Canada. FilerView and SecureShare are registered trademarks of Network Appliance, Inc. in the United States. Data ONTAP is a trademark of Network Appliance, Inc. in the United States and Canada. Snapshot is a trademark of Network Appliance, Inc. in the United States and the European Union. ApplianceWatch, BareMetal, ContentDirector, ContentReporter, SecureAdmin, Serving Data by Design, Smart SAN, SnapManager, SnapMirror, SnapRestore, and Web Filer are trademarks of Network Appliance, Inc. in the United States.

Apple is a registered trademark and QuickTime is a trademark of Apple Computer, Inc. in the United States and/or other countries.

Microsoft is a registered trademark and Windows Media is a trademark of Microsoft Corporation in the United States and/or other countries.

RealAudio, RealNetworks, RealPlayer, RealSystem, RealText, and RealVideo are registered trademarks and RealMedia, RealProxy, and SureStream are trademarks of RealNetworks, Inc. in the United States and/or other countries.

Network Appliance is a licensee of the CompactFlash and CF Logo trademarks.

All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such.

Network Appliance NetCache is certified RealSystem compatible.

Table of Contents

	Preface	iii
Chapter 1	Overview of ContentDirector.	1
	ContentDirector function and key features	2
	ContentDirector architecture	3
	Resources for distributing content	5
Chapter 2	System Requirements	7
Chapter 3	Overview of the ContentDirector API	9
	Typical applications for the API	10
	Summary of API capabilities	11
	ContentDirector features supported by the API	12
	API usage basics	14
Chapter 4	API Demo	17
Chapter 5	Assignment, Status, and Exception Codes	21
	Assignment types	22
	Status codes	23
	Exception codes.	25
	Glossary	27
	Index	33

Preface

Introduction

This guide provides a general overview of the Network Appliance™ ContentDirector™ 2.0 API. For detailed information about the packages, classes, and methods contained in the API, see the Javadoc on the ContentDirector CD.

Audience

This guide is for programmers who use the ContentDirector API. It assumes you are familiar with your operating system, your network, the basic concepts of how data moves over a network, database concepts, and programming in the Java language. This guide does not cover basic system, programming, or network administration topics, such as IP addressing, routing, and network topology; it emphasizes the characteristics of the ContentDirector API software.

Terminology and conventions

The term “ContentDirector” refers to a complete configuration, including all components of the software.

This guide uses the term “type” to mean pressing one or more keys on the keyboard. It uses the term “enter” to mean pressing one or more keys, and then pressing the Enter key.

Keyboard conventions

When describing key combinations, this guide uses the hyphen (-) to separate individual keys. For example, “Ctrl-D” means pressing the “Control” and “D” keys simultaneously. Also, this guide uses the term “Enter” to refer to the key that generates a carriage return, although the key is named “Return” on some keyboards.

Typographic conventions

The following table describes typographic conventions used in this guide.

Convention	Type of information
<i>Italic type</i>	Words or characters that require special attention. File names. Placeholders, for information you must supply. Book titles in cross-references.
Monospaced font	Command names. Information displayed on the system console or other computer monitors. The contents of files.
Bold monospaced font	Words or characters you type. What you type is always shown in lowercase letters, unless you must type it in uppercase letters.

Special messages

This guide contains special messages that are described as follows:

Note

A note contains important information that helps you install or operate the system efficiently.

Caution

A caution contains instructions that you must follow to avoid loss of data.

About this chapter

This chapter provides an overview of ContentDirector features and architecture, and how the API interacts with other components of the software.

For more detailed information about using ContentDirector, see the *ContentDirector 2.0 Administration Guide*.

Topics in this chapter

This chapter contains the following topics:

- ◆ [“ContentDirector function and key features”](#) on page 2
- ◆ [“ContentDirector architecture”](#) on page 3
- ◆ [“Resources for distributing content”](#) on page 6

ContentDirector function and key features

Function and key features

ContentDirector automates the distribution of content over a network from a central source to multiple destinations. The network can be a local area network, intranet, or internet. Reusable, user-defined rules govern content distribution tasks, including the location of source files and destinations, scheduling, and how content is handled during distribution.

Supported content types

ContentDirector distributes content for the following protocols.

Protocol	Name
http://	HTML
mms://	Microsoft Media Streaming
rtsp://	Real-Time Streaming Protocol

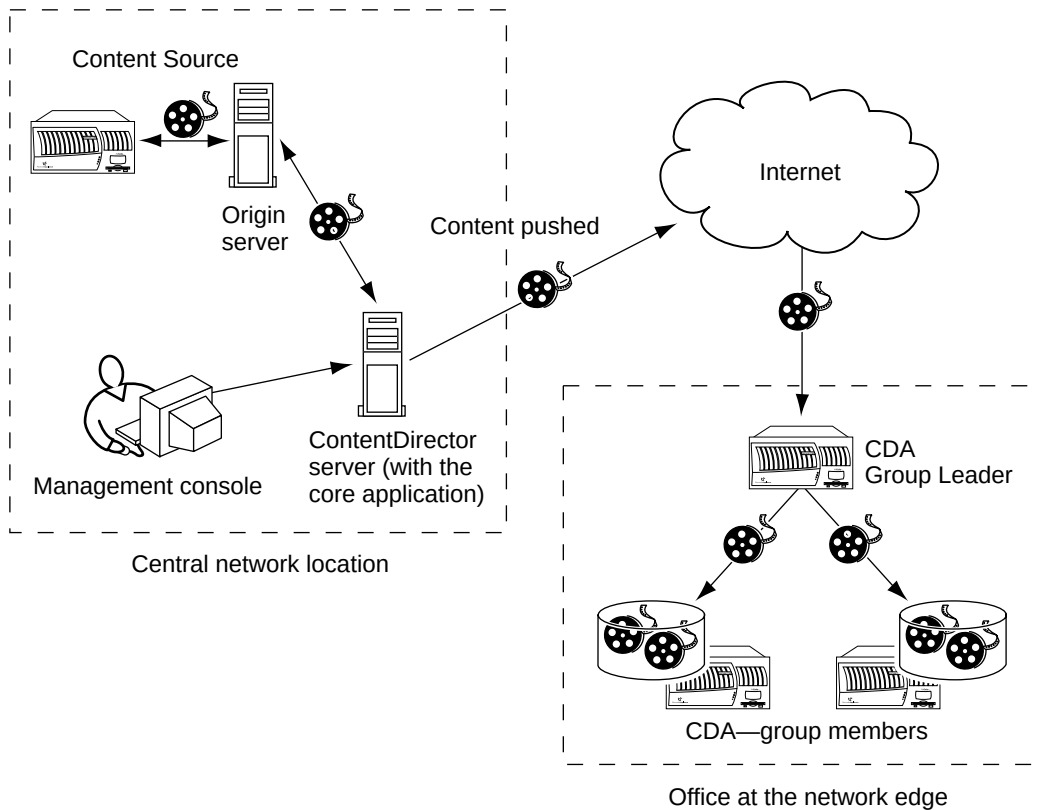
ContentDirector architecture

Components of ContentDirector distributed architecture

ContentDirector has a distributed architecture based on the following main components:

- ◆ *Core application* on a Windows or Solaris server, consisting of
 - ❖ Content control manager that originates and manages content distribution tasks
 - ❖ Scheduler that schedules content distribution tasks
 - ❖ Logger
 - ❖ Integrated Sybase database
- ◆ *ContentDirector Agents* (CDAs) on NetCache™ appliances, that can perform one or both of the following roles:
 - ❖ Group members (GMs) that receive content and serve it to clients
 - ❖ Group leaders (GLs) that manage communications between the core application and group members
- ◆ Optional administrator applet on a remote Java-enabled client, to enable remote administration of the core application

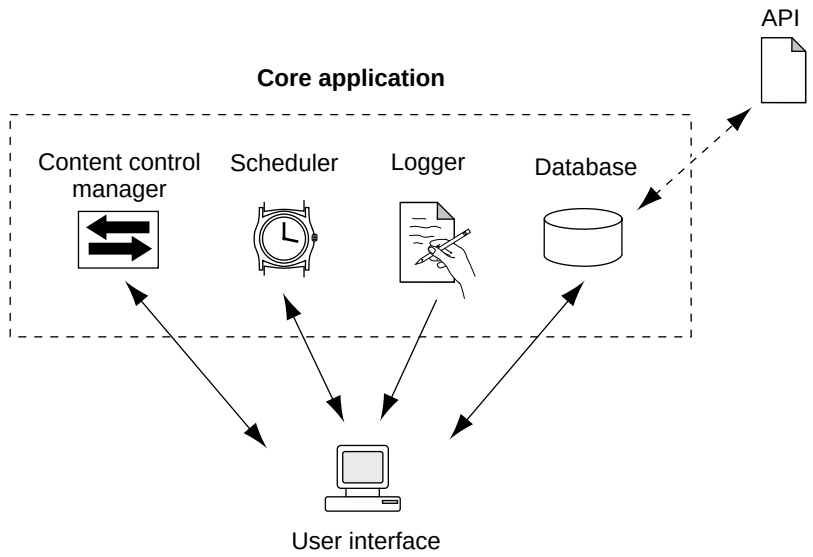
The following illustration shows how information flows through the components of ContentDirector.



How the API fits into the ContentDirector architecture

The API communicates only with the database component of the core application. It has no direct communication with the other components. By communicating with the database, the API can control objects that trigger events in the other components. This allows it to act as foundation for a customized user interface or as a mechanism for other applications that utilize the API to interface and communicate with ContentDirector.

The following illustration compares how the standard ContentDirector user interface and the API interact with the core application.



Resources for distributing content

About resources

This section describes the resources in the ContentDirector database that are used in the content distribution process.

API rights to resources

The following table lists types of resources in ContentDirector, their purpose, and the rights of the API to each type. The API can have one of three rights to a resource:

- ◆ Full—create, modify, delete, and retrieve information on existing objects
- ◆ Read—only retrieve information about existing objects
- ◆ None—cannot access this resource

In addition to content distribution resources, ContentDirector has general preferences and a system of user rights to which the API has no access.

Resource	Definition	API Rights
Assignment	Task type performed by job	Read
Contact	Personnel who automatically receive email from ContentDirector regarding job status	None
Content source	Directory at the center of the network, containing content to be distributed through ContentDirector to NetCache appliances at the edge	None
Default policy	Preconfigured policy shipped with ContentDirector	Read
Destination	Node at the edge of the network that receives content through ContentDirector	Read
Destination group	Group of network edge nodes that are targeted to receive content from the same source in ContentDirector	Read
Jobs	Reusable set of parameters that specify the content and schedule for content distribution events	Full
Policy	Reusable set of rules in ContentDirector governing how content is handled during distribution events	Read

Types of system requirements

The system requirements for the ContentDirector API include

- ◆ Hardware and operating system
- ◆ ContentDirector
- ◆ Java tools

Hardware and operating system requirements

The following table provides hardware and operating system requirements for ContentDirector and the API.

	Windows NT®	Windows® 2000	Solaris	NetCache (GL and GM only)
OS	Service Pack 4 and later	Server and Advanced Server, all versions	Version 2.7 and later	Version 5.2 and later
Processor	800 MHz	800 MHz	400 MHz	
RAM	Core application: 256 MB (512 MB –1GB preferred) ContentDirector Agent: 256 MB	Core application: 256 MB (512 MB–1GB preferred) ContentDirector Agent: 256 MB	512 MB	

ContentDirector

ContentDirector 2.0 or higher is required to use the API. There are three requirements related to ContentDirector for using the API:

- ◆ To compile programs written with the API, the contentdirector.jar file must be in the classpath.
- ◆ To use programs written with the API, the ContentDirector NT service or Solaris daemon must be running.
- ◆ To successfully distribute content, ContentDirector must recognize at least one destination as online and responding.

Java tools

Java Development Kit (JDK) 1.3 or later must be installed on the system. JDK 1.3 is distributed on the ContentDirector CD, and you can download the most recent version from the Sun Microsystems Web site. The ContentDirector installation program does not automatically install the JDK.

About this chapter This chapter contains an overview of the ContentDirector API, its purpose, and its capabilities.

Topics in this chapter This chapter contains the following topics:

- ◆ [“Typical applications for the API”](#) on page 10
- ◆ [“Summary of API capabilities”](#) on page 11
- ◆ [“ContentDirector features supported by the API”](#) on page 12
- ◆ [“API usage basics”](#) on page 14

Typical applications for the API

Example applications

The API is for users who need to use ContentDirector in a customized manner. Some examples are

- ◆ Creating a custom user interface for ContentDirector, for security or standardization. This interface can present a subset of ContentDirector features to users, such as only destinations and existing jobs.
- ◆ Including ContentDirector functionality in the user interface for another application, such as an HTML authoring application that includes pushing updated content to selected destinations.
- ◆ Automatically initiating content distribution based on an event-driven trigger, for example, pushing updated content to a specified destination group when a sales figure is incremented by a threshold amount.

Summary of API capabilities

Capabilities and limitations of the API

The ContentDirector API provides a Java interface to automate the scheduling and creation of jobs. It specifically supports the ContentDirector features that relate to running jobs, including

- ◆ Opening and closing the connection to the ContentDirector database
- ◆ Retrieving the predefined components of a job
- ◆ Retrieving predefined jobs
- ◆ Creating new jobs from the predefined components
- ◆ Running jobs: distributing content according to the parameters specified by the job

After the API retrieves a job or job component object and its ID number, it can also retrieve the parameters of that object.

Because the API is specifically focused on jobs, it does not offer full access to other features of ContentDirector. The API enables you to retrieve policies, destination groups, and destinations and their respective parameters, but not to create or modify them. Because you cannot create or modify these components in the API, they must be defined in the ContentDirector user interface before you run the application that uses the API.

ContentDirector features supported by the API

About API classes

The API includes classes that define the ContentDirector objects used by the API. These classes are limited to those pertaining to the assignment and those used to define the job. The methods for the job set and return job parameters. The methods for the destination group, destination, and policy objects return the parameters only.

Connection interfaces

The API can perform the following tasks in conjunction with the ContentDirector database:

- ◆ Establish a secure connection to the ContentDirector server and return a handle used to authenticate subsequent transactions
- ◆ Retrieve objects from the database
- ◆ Transfer local objects to the database
- ◆ Close the connection to the server

The user name and password must be associated with an account that has system administrator privileges. However, for security reasons, the default system administrator user/password of super/super cannot be used to access the API.

Job component interfaces

These interfaces retrieve lists of component objects with associated IDs. The IDs can then be used in subsequent calls to include the component in a job or to retrieve the parameters of the component. For example, after retrieving a policy ID, you can find out the commit quorum set in that policy.

The job component interfaces can retrieve the following components:

- ◆ Destination groups
- ◆ Destinations
- ◆ Policies
- ◆ Default policy
- ◆ List of all policies associated with the user

Job interfaces

The job interfaces are used to retrieve job IDs. The job IDs are used in content distribution calls, in job modification and deletion calls, and to retrieve job components and assignments.

The job interfaces can perform the following tasks:

- ◆ Create jobs
- ◆ Run jobs
- ◆ Modify jobs
- ◆ Delete jobs
- ◆ Retrieve all job IDs associated with the user
- ◆ Retrieve all job IDs of a particular status
- ◆ Retrieve the history of a job
- ◆ Retrieve the ID of the last assignment that ran for a specific job

Content distribution interface

The content distribution interface initiates the content distribution event. This call returns a job Assignment ID that can be used to retrieve the status of the assignment.

Assignment interfaces

The assignment interfaces enable you to retrieve the status code and assignment type of a job assignment. See `CdAssignment > getAssignmentTypeId()` in the Javadoc for a list of assignment types.

Features not supported by the API

The ContentDirector API does not access the following features of ContentDirector:

- ◆ User rights scheme—the API user must be a ContentDirector system administrator
- ◆ Content sources
- ◆ Preferences
- ◆ Contacts

The API has one additional limitation: It supports only compiled Java code. Scripts and command line input are not supported.

API usage basics

Logging into the API

The API accepts a username and password as arguments. The simpleApiDemo.bat demo batch file contains the following example of a login to the API:

```
C:\Program Files\Network Appliance
Inc\ContentDirector\CCM\jakarta-
tomcat\lib\jasper.jar;" -DPROP_FILE="C:\Program
Files\Network Appliance
Inc\ContentDirector\CCM\config\cm.properties"
SimpleApiDemo -id demo -pw super -rm RMI -server local
```

Compiling programs written with the API

Use ContentDirector.jar to compile programs written with the API. For example, to compile SimpleApiDemo.java, at the \API directory enter the following command:

```
javac -classpath "C:\Program Files\Network Appliance
Inc\ContentDirector\CCM\lib\ContentDirector.jar"
SimpleApiDemo.java
```

Using the demo as a template

You can use the SimpleApiDemo.java file as a template for how to execute many common calls to the API. For example, the following lines of code in the demo determine which jobs are currently running:

```
* 8) get the jobs of status running

*/

cdJobs =
this.getAllJobsOfStatus(conn, IConstantsExternal.STAT_R
UNNING);

System.out.println("*****");
System.out.println("Running jobs");
System.out.println("*****");
printVectorItems(cdJobs);
```

**Where to find
additional
information**

A comprehensive Javadoc Help file is available on the ContentDirector CD. It details the packages, classes, and methods included in the API.

About the API demo The API demo shows how the API accesses the ContentDirector database to create and run a job. To view the program code for the demo, open the ContentDirector\API\SimpleAPIDemo file with a text editor.

Before you start Before you run the demo, you need to do some preliminary setup:

- ◆ Prepare the environment for the demo to run
- ◆ Create objects in ContentDirector required by the demo

Environment setup: To set up the environment for the demo, complete the following steps.

Step	Action
1	For Solaris, create a /ApiDemoFiles directory, or for Windows, create C:\ApiDemoFiles.
2	Create a testfile.txt dummy text file in the directory you just created.

ContentDirector setup: To set up required database objects in ContentDirector, complete the following steps.

Note

You must access ContentDirector through the management console (standard user interface) to create these objects. For detailed instructions, see the *ContentDirector 2.0 Administration Guide*.

Step	Action
1	Create a user named demo with the password super, and select System Administrator for type of user.
2	Create a source called www.apidemo.com. Browse and select the ApiDemoFiles folder for the root source directory.
3	Create a destination group called filePushDemoGroup, select www.apidemo.com as the source site, and select any destinations as members of this group.

Step	Action
4	Ensure that ContentDirector recognizes all members of the new destination group as online and responding in the Destinations view of the main screen.

Running the API demo

To run the API demo, complete the following steps.

Step	Action
1	Ensure that the ContentDirector service is running.
2	At the machine running the ContentDirector core application, navigate to the ContentDirector\API directory.
3	Run the simpleApiDemo.bat batch file. Connecting to the database can require about a minute.
4	After the API demo runs, at the prompt, press a key to close the demo window.

Activity in the demo window

As the demo runs, it executes the following steps (all printing is to the console):

Step	Action
1	Connects to the database. Prints connection status: Connection established!
2	Retrieves the default policy and prints its name: ***** Default Policy *****

Step	Action
3	Checks whether the database is valid, and prints ***** Database is valid *****
4	Retrieves the list of destination groups, and finds the destination group named filePushDemoGroup.
5	Retrieves and prints the destination servers of the filePushDemoGroup destination group. Printout starts as follows: ***** List of destinations for filePushDemoGroup *****
6	Creates a demo job with the policy ID being the default policy ID and the content group ID being the ID of filePushDemoGroup.
7	Prints the list of jobs. Print out starts as follows: ***** List of jobs *****
8	Runs the new demo job.
9	Retrieves the list of jobs with the status running, and prints them out. Print out starts as follows: ***** Running jobs *****
10	Modifies the new demo job.

Step	Action
11	Prints the list of jobs. Printout starts as follows: ***** List of jobs *****

**Activity in the
ContentDirector
management
console**

The API demo creates and runs a job called ApiDemoJob. You can view this job in the Jobs view of the ContentDirector main screen. A job that ran successfully has the status of Completed. To view a history of this job, right-click the job, then click History. You can also view details in the Status window at the bottom of the main screen.

About this chapter This chapter provides lists of ContentDirector assignment types, status codes, and error codes that are referenced by the API.

Topics in this chapter This chapter contains the following topics:

- ◆ [“Assignment types”](#) on page 22
- ◆ [“Status codes”](#) on page 23
- ◆ [“Exception codes”](#) on page 25

Assignment types

About assignments and assignment types

Assignments define the specific tasks that jobs perform. Each job runs a sequence of assignments that represents sequential phases of content distribution. See the following table for definitions.

Retrieving the assignment type

Use the `GetAssignmentTypeID()` method of the `CDAssignment` class to retrieve the assignment type for any assignment. See `com.netapp.contentdirector.api.jdk.CdObjects > CdAssignment > getAssignmentTypeId` in the API Javadoc.

Commonly used assignment types

A job typically has the assignment type of `publish`. If a user aborts a job before it is completed, the assignment type is `abort`. The following table contains the five assignment types you see through the API. For more information about replicate and repush, see the *ContentDirector 2.0 Administration Guide*.

Assignment type	Description
CCMEVENT_ABORT	User aborting job
CCMEVENT_PUBLISH	Delivering command buffer
CCMEVENT_REMOVE	Deleting job
CCMEVENT_REPLICATE	System created
CCMEVENT_REPUSH	Resending job to destinations that did not receive it the first time

Status codes

About status codes Status codes display the current state of each assignment while it is running, and the final state after the assignment has ceased running.

Retrieving the status code Use the `GetStatusID()` method of the `CDAssignment` class to retrieve the status of any assignment.

The following table lists status codes, sorted by message, with brief descriptions.

Status	Description
STAT_ABORTED	User aborted job while it was running
STAT_ASGNCREATE	Creating assignment
STAT_AWAITINGCCM	Shows only at ContentDirector startup— job is queued waiting for content control manager to be ready
STAT_CANCELED	Not currently used
STAT_CHANCREATE	Creating channels
STAT_CMDDELIVER	Delivering command buffer
STAT_COMMITING	Committing assignment
STAT_COMPLETED	Succeeded
STAT_CONFLICT	Conflicted with another assignment
STAT_CRAWLING	Not currently used
STAT_DATADELIVER	Delivering data
STAT_EXPIRED	Last scheduled run time is in the past
STAT_FAILED	Job failed; review log messages to determine reason
STAT_IDLE	Waiting to run

Status	Description
STAT_INITIATED	Job started to run
STAT_PENDING_EXECUTION	Scheduler sent job to content control manager
STAT_PROCESSING	Destination is processing job
STAT_RUNNING	Content control manager started job
STAT_SUSPENDED	Not currently used
STAT_UNSCHEDULED	Not currently used

Exception codes

About exception codes

The API can cause exceptions if there are problems connecting to the ContentDirector database or accessing an object.

The following table contains the exception messages that are seen by the API.

Exception	Description
CdApiVersionException	API incompatibility issue
CdAuthenticationException	Problem verifying account and authenticating
CdPermissionException	Problem accessing a CD object
CdServerException	Problem connecting to CD services

Glossary

appliance	A device that performs a single, well-defined function and is simple to install and operate, for example, a NetCache appliance or NetApp® filer.
authentication	The process of verifying the identity of a user who is logging in to a computer system.
cache	<i>Verb.</i> To store data temporarily for expedited access. <i>Noun.</i> The location in which data is stored temporarily. There are a variety of types of caches, including disk cache, memory cache, and Web cache.
checkpoint	A copy of content pushed by a specific distribution event in ContentDirector. Checkpoints are stored at the center of the network as backups. They are used to restore content at the edge to a known state.
client	A workstation or PC in a client-server architecture; that is, a computer system or process that requests services from and accepts the responses of another computer system or process.
console	A physical or virtual terminal that is used to monitor and control a NetApp appliance.
ContentDirector	A Network Appliance scalable software solution that automates the distribution of content from the center to the edge of networks.
ContentDirector agent (CDA)	A component of ContentDirector that acts as distribution or destination point at the edge of the network, for content pushed from the center of the network.

content provider	Any company in the business of aggregating or producing content for delivery through the Internet.
core application	The controlling component of ContentDirector at the center of the network that schedules, selects content, and creates policies for the distribution of content to the edge.
Data ONTAP	A software product designed exclusively for NetApp appliances that optimizes file service by combining patented file-system technology and a microkernel design dedicated to network data access. Developed by Network Appliance.
destination	A node at the edge of the network that receives content through ContentDirector and serves it to clients.
destination group	A group of network edge nodes that are targeted to receive content from the same source in ContentDirector. Destination groups are purely logical and do not need to map to physical groups of nodes.
domain	A subset of ContentDirector content sources and destinations, associated with specific groups of users and visible only to them and administrators.
Domain Name System (DNS)	An Internet service for finding computers on an IP network.
domain administrator	A ContentDirector user with administrative rights to a specific set of program resources.
edge device	Any device, for example a NetCache appliance, that is deployed at the periphery of a LAN and WAN environment to interface traffic from the edge of one network to the other.
file server	A computer that provides file-handling and storage functions for multiple users.

filer	An appliance that attaches to a computer network and is used for data storage. NetApp filers operate using Data ONTAP™ software.
firewall	A server or device deployed between a LAN and the external WAN to control all inbound and outbound traffic, thus the enhancing security of the network behind the firewall.
fully qualified domain name (FQDN)	The complete name of a specific computer on the Internet, consisting of the computer's host name and its domain name.
group leader	A network node that acts as a ContentDirector distribution point, or passthrough, for content near the edge of a network. Group leaders transmit content to group members.
group member	A network node that acts as a ContentDirector destination point for content at the edge of a network. Group members receive content from group leaders.
IP address	The unique numerical address of a computer that is attached to a TCP/IP network, for example, 198.95.226.66.
job	A reusable set of parameters in ContentDirector that specifies the content and schedule for content distribution events.
MIME	Multipurpose Internet Mail Extensions. A specification that defines the mechanisms for specifying and describing the format of Internet message bodies. An HTTP response containing the MIME Content-Type header allows the HTTP client to invoke the application that is appropriate for the data received.
MMS	Microsoft Media Streaming. The streaming media control protocol used in Windows Media Technologies 3 and 4 streams.

mount	To make the file system on a particular device accessible by attaching it to the file hierarchy.
NetCache	The proprietary software that runs on NetCache appliances. Developed by Network Appliance.
NetCache appliance	An appliance designed specifically to serve Internet and intranet content quickly and reliably. Developed by Network Appliance.
NetCache Manager	The browser interface through which a NetCache appliance is managed.
network interface card (NIC)	A printed circuit board that connects a computer or other node to a network. Also known as a “network adapter.”
NOW	NetApp On the Web. The secure customer service Web site for Network Appliance.
NTFS permissions	The permissions used in NTFS-style security to control access to files and directories.
policy	A reusable set of rules in ContentDirector governing how content is handled during distribution events, criteria for success, and email notifications of job status.
quorum	Threshold criteria for the success of a content push in ContentDirector.
regular user	A nonadministrative ContentDirector user. Regular users have only the rights granted by membership in a group of users, defined by an administrator.

Remote Shell	A program that enables a user on one system to execute a program on another system. Remote Shells are usually established with the <code>rsh</code> command and are not interactive.
resource	An object in ContentDirector that is necessary for using the program. Users can create, modify, use and delete resources. Resources include: jobs, policies, destination groups, destinations, sources, checkpoints, contacts, users, and user groups.
rsh	The command used to establish a Remote Shell connection.
RTSP	Real Time Streaming Protocol. An open standard for delivery of real-time media over the Internet.
Snapshot	A Data ONTAP feature that creates an online, read-only copy of the entire file system, the Snapshot™ feature protects against accidental deletions or modifications of files without duplicating file contents.
source	Directory on a local disk drive at the center of the network, containing content to be distributed through ContentDirector to NetCache appliances at the edge.
stream	A media file that is transmitted in a continuous flow over a network. Streams are of two types, live and on-demand.
streaming media	A term used to describe media files that begin playing while they are being transmitted over the network to the media player on the client computer.
streaming server	A server configured to serve streaming media to clients.
system administrator	A ContentDirector user with administrative rights to all program resources.

user group	A list of nonadministrative ContentDirector users, and the rights that they share.
Time To Live (TTL)	The time limit on how long a copy of an object remains in the cache before NetCache verifies the object with the originating Web server.
volume	An independent file system with its own RAID groups.

Index

A

API

- compiling programs written with 14
- exceptions thrown by 25
- Javadoc for 15

Assignments

- API rights to 5
- definition of 22
- retrieving types 22

C

Classpath requirements 7

Contacts, API rights 5

Content source, API rights to 5

ContentDirector

- components of 3
- function of 2
- key features 2
- minimum system requirements for 7

D

Default policy, API rights to 5

Demo

- activity in console 18
- activity in ContentDirector GUI 20
- how to run 18
- preparing to run 17
- using as template 14

Destination group

- API rights to 5

Destinations

- API rights to 5
- requirements for distributing content 7

E

Exceptions thrown by API 25

J

Java requirements 8

Javadoc for API 15

Jobs, API rights to 5

L

Logging into API 14

N

NetCache

- system requirements 7

P

Policies, API rights to 5

Protocols, supported 2

S

Solaris system requirements 7

Status codes

- definition of 23
- retrieving 23

W

Windows system requirements 7

