# cruizemedia

Cruizemedia Content Server v1.2
Administrator's Guide

# Contents

# **1** **Getting Started**

Cruizemedia Content Server is a enterprise-class video-on-demand platform capable of serving a wide range of publishing and viewing requirements.

## **About this documentation**

This documentation describes how to install and configure Content Server for a wide range of deployment environments.

## **Introduction**

Thank you for your evaluation or purchase of Cruizemedia Content Server. We strive to create the best products in the market for the categorization, searching, processing and delivery of digital media.

## **What's Included**

Cruizemedia Content Server is a purely self-contained package including:

- Web application server
- Embedded database server
- Search engine
- Server-side components for the categorization, searching and delivery of digital media
- A sample library with Learning Series content, ready to play

A typical install can be accomplished in just a few minutes resulting in a base system that's ready to use. The base system is perfect for demonstration purposes or for smaller installations. However, Content Server can easily integrate with external streaming, database and directory servers for medium to large deployments, as you'll discover in the following chapters.

# 2 System requirements

The Cruizemedia Platform is entirely Java-based and capable of running on Java Runtime Environment (JRE) 1.6 or higher. In some instances, the software may include 3rd party libraries which are platform-specific. The following list of operating systems, database and streaming media servers are officially supported.

## Operating Systems

The following 64 bit operating systems are supported:

- Windows 2003 Server
- Windows 2008 Server
- Windows 2008 R2 Server
- Red Hat Enterprise Linux 5 and 6
- SUSE Linux Enterprise Server 11
- Ubuntu 10.04 LTS Server
- Ubuntu 12.04 LTS Server
- OS X Server 10.8

## Streaming Media Servers

The following streaming media servers are optionally supported with Cruizemedia Content Server Standard and Enterprise editions:

- Flash Media Server 3.0 and 4.0
- Wowza Media Server 2.0 and 3.0

## Directory Servers

The following directory servers are optionally supported with Cruizemedia Content Server Enterprise Edition:

- Microsoft Active Directory
- Novell eDirectory
- OpenLDAP

# Database Servers

The following database servers are supported with Cruizemedia Content Server Enterprise Edition:

- H2 1.3 (included in Standard and Enterprise Edition)
- Oracle 11g 11.1 and 11.2
- Microsoft SQL Server 2005 (Express Edition not supported)
- Microsoft SQL Server 2008
- MySQL 5.5

# 3 Definitions

The following definitions are essential in understanding how objects are structured, categorized and used throughout the system.

### Chapter
A logical division in media containing an logical start and end time. Chapters apply to media types with linear playback such as audio and video.

### Learning Correlation
A qualified associated between a Learning Objective and the learning material (Media).

### Learning Objective
A statement describing what knowledge a learner is expected to gain from the material. Objectives are usually written in a hierarchical structure from general to very specific language.

### Learning Standard
A collection of related Learning Objectives typically organized in a hierarchical structure.

### Library
A library is the highest level virtual container that holds media. Access control can be applied to libraries on a per user or group basis.

### Library Section
A library section is a virtual container of media. A library may have zero or more sections. Media that belongs to a section is accessible only to that section and to the library containing the section.

### Lightweight Directory Access Protocol (LDAP)
LDAP is an open industry standard for accessing directory services. It is commonly used as an external source of authentication.

### Media
Media is an upper most container which describes an asset. Media contains metadata describing it along with chapters, media files, artwork and resources.

### Media File
For every media, one or more media files can be added. Cases where multiple media files may be of benefit include multiple resolutions of video or

images, or the same media encoded using different codecs targeting different devices.

**Media Type**
Three media types are supported in Content Server; audio, video and images.

**Mime Type**
A mime type (aka content type) is a unique identifier used to communicate the type of file being sent over a MIME-based protocol.

**Resource**
A resource is an additional asset added to a media. Each title may contain an unlimited number of additional resource. Resources may include images, PDF's, word processing or spreadsheets. Virtually any file can be added as a resource to a media.

**Storage Location**
A storage location defines the base streaming URL and base file storage location used when adding or accessing media. Storage locations are defined on a per media file basis.

**System Property**
A name/value pair used for global configuration.

**User Account**
Contains user information and credentials. User accounts may be managed internally using a built-in user management and authentication system, or externally via LDAP.

**User Group**
A user account can belong to zero or more user groups. User groups are used as a container to provide access control across multiple user accounts.

# 4 Installation

The following topics discuss installing Cruizemedia Content Server. Also covered is starting the server the first time and logging on to the user interface in a browser.

For a  platforms, a graphical installation wizard will guide you through the installation process. Additionally, the installer will detect the absence of a desktop environment and present a console-mode installer if necessary.

## Installing on Windows

When installing on Windows with UAC enabled, the following dialog may be displayed:
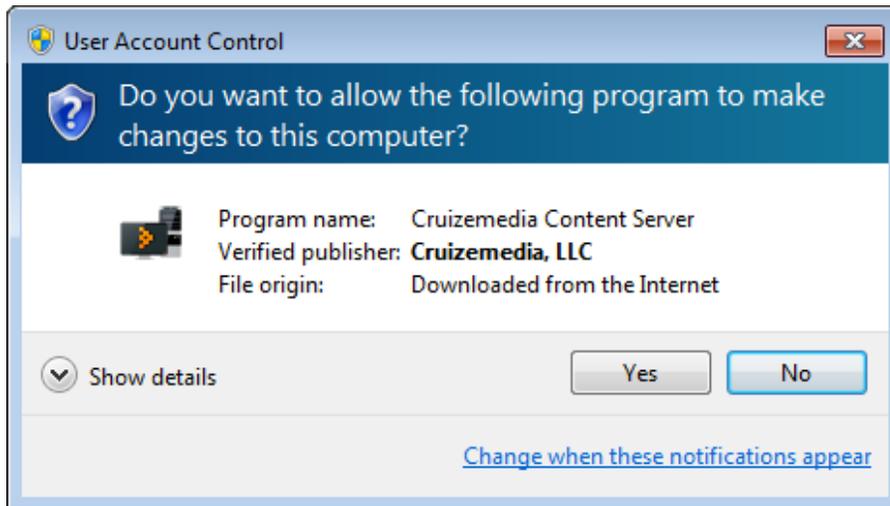


Ensure the publisher is verified as depicted in the image above and click 'Yes' to begin the installation.

The default installation directory on Windows is:
`C:\Cruizemedia\ContentServer`

It is highly recommended that Content Server <u>not</u> be installed into Program Files (x64 or x86), rather to it's own directory as shown above.

Upon a successful installation, the installer will automatically create a Windows system service to start Content Server the next time the operating system loads and will automatically start the server when installation completes.

# Installing on OS X

When installing Content Server on OS X, the default installation directory is:
`/Applications/Cruizemedia/ContentServer`

This path may be changed to any directory accessible during operating system startup. The installer will install Content Server as the current logged in user. As a final step of the installation process, the installer will attempt to install a Daemon Launcher which is responsible for starting Content Server automatically each time OS X loads. Installing a Daemon Launcher requires root administrative privileges and will force the following dialog to display:



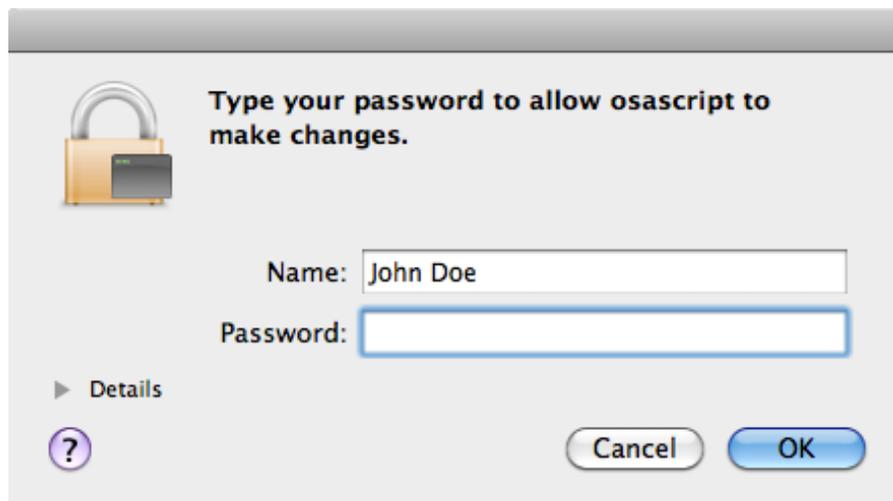After entering the users password, the Daemon Launcher will be installed. When installation has completed, Content Server will start automatically.

# Installing on Linux

Installation of Content Server on Linux may be performed via a graphical wizard identical to OS X and Windows, or via the command line. The installer will automatically determine if a desktop environment is present and if so, will launch the graphical installer. If no desktop environment is detected, as is the case with a typical Linux server, a text-based console installer will launch. In either case, the process of installing is identical.

When installing Content Server on Linux, the default installation directory is:
`~/Cruizemedia/ContentServer`

This path may be changed to any directory accessible to the operating system. The installer will install Content Server as the current logged in user.

Upon the completion of a successful installation, the installer will automatically start Content Server. Unlike the OS X and Windows installers, the Linux installer will not automatically create a startup daemon. Refer to your Linux distributions' guidelines on proper startup daemon creation.

# Configuring external database

The following topics explain how to configure Content Server to use an external database. The use of an external database server is recommended for instances where high-traffic is expected or for medium to large media libraries. Smaller, low-traffic deployments may use the embedded H2 database without any further configuration. External database support is available in Cruizemedia Content Server Enterprise Edition.

## Read-committed mode required

The database must run in read-committed transaction isolation mode. Read-committed transaction isolation mode is the default mode for Oracle and SQL Server, but not MySQL.

If you use MySQL, you must change to read-committed transaction isolation mode before starting the server for the first time after installing. The MySQL configuration file is my.cnf (UNIX) or my.ini (Windows). The setting in the server section under [mysqld] must be set to READ-COMMITTED. For example:

```
[mysqld]
transaction-isolation=READ-COMMITTED
```

## Enable case sensitivity

If you plan to run Content Server on a computer with a UNIX or Linux operating system, enable case sensitivity for the database. This applies to recognizing case sensitivity of data; it does not apply to the case of database table or column names. Case sensitivity must be enabled when the database is created and not at a later time.

# Oracle database guidelines

After creating the schema user, execute the following statements:

```
GRANT CREATE PROCEDURE TO [user];
GRANT CREATE SESSION TO [user];
GRANT CREATE TABLE TO [user];
GRANT CREATE VIEW TO [user];
GRANT CREATE SEQUENCE TO [user];
ALTER USER [user] QUOTA UNLIMITED ON [tablespace];
```

# MySQL tuning guidelines

The following are tuning guidelines for MySQL.

### Read-committed transaction isolation mode
MySQL must be run in read-committed transaction isolation mode. For details, refer to the previous page.

### InnoDB Storage Engine & Barracuda
Content Server only supports MySQL when MySQL has the InnoDB Storage Engine set as default and when used in conjunction with the Barracuda file format. The Barracuda file format enables many advanced features which Content Server takes advantage of.

```
[mysqld]
default-storage-engine=INNODB
innodb_file_per_table=1
innodb_file_format=barracuda
```

### Server name case sensitive
If you run MySQL on a UNIX or Linux computer, the database server name is case sensitive. For instance, if there is a table named *media* and you write the query:

```
SELECT * FROM MEDIA;
```

MySQL reports that it cannot find the table MEDIA.  The solution for dealing with this is recommended whether you run MySQL on UNIX, Linux or Windows.

Set the following in the my.cnf file for the entire MySQL database server instance. On UNIX and Linux the configuration file is names my.cnf and on Windows the file is my.ini.

```
[mysqld]
lower_case_table_names=1
```

**Large Transaction Size**
Media artwork is stored in the database as binary objects. The default packet size for a SQL statement in MySQL is insufficient. Set the following in the my.cnf file for the entire MySQL database server instance.

```
[mysqld]
max_allowed_packet=24M
```

# Changing database configuration

To change database configuration, execute the dbConfig script in the /bin/ directory of the Content Server installation. On Windows, this can be accessed from the Start Menu or from the dbConfig.cmd command script and on Linux and OS X systems from dbConfig.sh.

Database configuration changes should be made when the server is not running. The next time the server is started, the changes will take effect.

If the database is empty, Content Server will automatically create the necessary data model and initialize default values for all of the required objects.

Content Server requires its own database separate from other applications.

# Starting Content Server

The follow describes the process of starting and shutting down Content Server for each supported platform.

## Windows

A system service is created as a final step of the installation process. The service will automatically start each time the operating system loads. This behavior can be modified by setting the service from 'automatic' to 'manual' in the Windows Control Panel. If the service is set to manual, Content Server may still be started and shutdown from the respective shortcuts on the Windows Start Menu. The shortcuts on the Start Menu are for starting and shutting down the server manually and should not be used in conjunction with a system service which automatically starts. The start and shutdown command are:

```
startServer.cmd
stopServer.cmd
```

## OS X

A Launch Daemon is created in /Library/LaunchDaemons and will automatically start Content Server each time the operating system loads. Additionally, it automatically starts upon the completion of installation. Shell scripts to start and shutdown the servers are located in the /bin/ directory of the Content Server installation. The start and shutdown commands are:

```
startServer.sh
stopServer.sh
```

## Linux

Content Server is started upon completion of installation. Unlike Windows and OS X, a daemon is not automatically created due to the wide variety of distribution-specific variances. System administrators may easily create the daemons manually. Startup and shutdown scripts are located in the /bin/ directory of the Content Server installation. The start and shutdown commands are:

```
startServer.sh
stopServer.sh
```

# First Time Logging In

By default, Content Server will configure the embedded web application container to listen on TCP port 8080. The Content Server user interface may be accessed via:

```
http://[hostname]:[port]
```

The default URL is:
http://localhost:8080

When Content Server starts up for the first time, it automatically creates the necessary objects, including the default admin account.

Username:     admin
Password:     admin

It's highly recommended that the admin password is changed. Additionally, secondary user accounts should be created with administrative privileges in the event the admin credentials are lost or forgotten.

# **5** Configuration

This chapter describes the various configuration options available.

## Libraries

Cruizemedia Content Server supports one or more libraries. Each library can:

- Contain media unique to each library
- Contain media shared across multiple libraries
- Permit or deny access on a per user or group basis
- Contain logical containers called library sections

Planning how to structure libraries may be the most important decision made in the configuration process.

Two example library structures are:

- Example Company Library
  - Human Resources
  - Sales & Marketing
  - Information Technology
  - CEO Messages
- Pre-release Training Library

- High School
  - Science
  - Social Studies ...
- Middle School
  - Science
  - Social Studies ...

In the first example, two libraries are created. The first library may allow everyone access to all content, while the second library may be restricted because it contains information about products or services that have not been announced yet.

The second example is similar, two libraries are created but this time each library could contain nearly identical library sections. Age or grade restrictions could be used to limit access to age appropriate material.

In both examples, library sections are used to help logically divide content. Access control is performed at the library level, not the section level.

# Storage Locations

A storage location defines base paths and URL's used to access content. Content Server must have at least one storage location defined. Using multiple storage locations is possible. The publishing process detailed in Chapter 6, will further describe how locations are applied on a per media file basis. For most installations, one storage location should be sufficient.

# User Management

Content Server includes built-in user management which allows the creation, management, authentication and authorization of users to the server and its libraries. When creating or editing users, the following fields are available:

- Username
- Password
- Name
- Email
- URL

Users can be assigned one of three available roles which grant additional privileges.

- Editor
- Publisher
- Administrator

**Editor**
This role can add new media to the server. Media added by this role may be subject to approval.

**Publisher**
This role, like the editor, can add new media to the server. Media added by this role does not require approval. In addition, this role can approve content additions made by editors.

**Administrator**
This role has full administrator access. This role is not restricted to libraries and can perform all global changes.

If a user does not have a role assigned, a default 'user' role is assumed and no additional privileges are granted.

Users can be members of zero or more groups. Advantages of assigning users to groups include simplified library access control and automatic LDAP group-to-role mapping.

# Directory Services

Content Server supports Active Directory, eDirectory, and OpenLDAP for external authentication. Enabling LDAP authentication does not turn off the internal user authentication scheme, rather, they work together to provide a unified authentication system.

Configuring LDAP requires the following connection parameters:

- Host:      The hostname or IP address of the LDAP server
- Base DN:   Fully qualified base distinguished name
- Port       TCP port (typically 389 or 636 for SSL)
- User DN:   Fully qualified distinguished name of an admin or proxy user
- Password:  The users password

The mapping of attributes to the specific LDAP implementation can easily be accomplished via the drop down entitled LDAP Schema. Selecting the schema that matches the LDAP server being connected to will automatically populate the attributes with the default values for that schema. OpenLDAP users may need to manually adjust these settings as OpenLDAP implementations vary between vendors.

Finally, access control via LDAP is accomplished by mapping specific LDAP groups to the pre-defined Content Server roles of user, editor, publisher and administrator. When an LDAP user authenticates to Content Server, a lookup is performed to query what groups the user is a member of and references it to the role the group is mapped to.

**Hybrid Authentication**
It is recommended to have one or more internal users defined who have administrative privileges in the event the defined LDAP server becomes unavailable or changes are made to its configuration. Doing so will allow you to login to Content Server as an internal user in order to manage these changes. If a hybrid approach is implemented, unique usernames must be used. It is not possible to have both an internal user and an LDAP user with the same username. There are no specific changes that need to be made to enable hybrid authentication. Content Server will always look at its internal database of non-LDAP users first, if it finds the account and the account is enabled, it will allow logins using it. If no internal account matching the username is found, it will attempt authentication via LDAP.

# System Properties

System properties are name/value pairs which have a global impact on the server. Content Server has the following system properties.

**MAX_RESOURCEFILE_UPLOAD_SIZE**
Specifies the maximum upload size for resources attached to media. Value specified is in megabytes. Defaults to 2 (MB).

**MAX_NUMBER_RESPONSE_ITEMS**
Specifies the maximum number of objects to return in a response. This value affects paging and should not be changed unless instructed to do so by Cruizemedia Support. Defaults to 100.

**COMPRESS_RESPONSE_WITH_GZIP**
Specifies whether the server should send it's response GZIP compressed. By enabling this option, the server will first check to see if the client can accept compressed responses. Enabling GZIP compression conserves bandwidth and decrease the overall response times. Defaults to true.

**LUCENE_FSDIRECTORY_LOCATION**
Specifies the directory the internal search engine should store the index files. This option is used when ENABLE_LUCENE_RAMDIRECTORY is disabled. Defaults to ~/.cruizemedia/index

**ENABLE_LUCENE_RAMDIRECTORY**
Specifies if the internal search engine should keep the index resident in memory. For most deployments, enabling this option will increase performance and reduce disk I/O when performing searches. If this option is disabled, the index will be read from disk from the location specified by LUCENE_FSDIRECTORY_LOCATION. Defaults to true.

**TEMP_DIRECTORY**
Specifies the directory to be used as a temporary directory. This location should have enough free space to accommodate all file uploads as they occur. Completed uploads are removed. Defaults to ~/.cruizemedia/temp

**LDAP_ENABLED**
Specifies if external LDAP authentication should be used. Changing this value is not recommended. To configure LDAP, see the Directory Services section of Chapter 6. Defaults to false.

**VALIDATE_ALL_XML_PAYLOADS**
Specifies if XML validation should occur when receiving and when generating XML payloads. Under normal production circumstances this value should be false. For development or API integration tasks, enabling this value will check payloads against the XML schema. Defaults to false.

**VALIDATE_ALL_XML_RESPONSES**
Specifies if XML validation should occur when the server attempts to send an XML response. Under normal production circumstances this value should be false. For development or API integration tasks, enabling this value will check all XML documents prior to actually sending them. Defaults to false.

# **6** Content Publishing

This chapter describes content publishing, formats and advanced streaming options.

## Supported Formats

Content Server supports standards-based image, audio and video formats including:

MP3 Audio
MPEG-4 Audio (AAC)
MPEG-4 Video (H.264/AVC + AAC)
Flash Video (FLV)
JPEG
PNG

Most production and encoding workflows already produce files in these supported formats.

## Roles

Content Server includes built-in roles for both editors and publishers.

• **Editors**: May edit existing content in libraries for which the user has access to.

• **Publishers**: May add, edit and delete content and may change library assignments of content in libraries for which the user has access to.

• **Administrator**: May change any content on the system. May access the administrative functions interface to manage libraries, sections, storage locations, users, groups and change the systems configuration.

## Adding Content

Content may be added to the server by a user with the publisher or administrator role. Simply right click anywhere in the library pane and select 'Add Media'. A dialog will appear asking for the title, initial library and media type. The initial library is the library where the content will first be published. Depending on your organizations' workflow, the initial library may be where you want the content accessible by all, or it may be a staging library created

in advance for the purpose of going through an internal approval process before being moved to its final destination.

After the content is added, the Media Editor will be displayed. The Media Editor allows the editing of all the contents metadata, most of it indexed and used by Content Servers internal search engine.

Artwork may be added provided that its in a supported JPEG or PNG format.

The Media tab is where content can be uploaded. Multiple files may be uploaded simultaneously. Upon completion, the files metadata including the file size, duration, bitrate and dimensions are displayed.

Chapters may also be defined. These provide logical start and end times as well as keywords used by the search engine. The use of chapters may be limited by the type of streaming being performed. By default, Content Server utilizes a technique known as 'progressive download'. Simply speaking, this means that content may be played while it is downloading. If a chapter start time is selected and the file has not yet downloaded up until that point, chapter playback will not start. This deficiency along with many other advantages may be alleviated by the integration of a supported streaming media server such as Wowza Media Server or Flash Media Server.

Resource material may be added to each media asset. Resources are meant to enhance the content by providing access to additional material. Use cases where resources are advantageous are the inclusion of policies, sign-off sheets, assessments and references and links to related content.

Unlike media itself, resources are stored in the database as binary large objects. Therefore, there is a built-in limit of 2MB for each resource. This limit may be overwritten by setting the MAX_RESOURCEFILE_UPLOAD_SIZE system property.

# 7 Educational Applications

This chapter describes the use of Content Server in various education applications.

## Introduction

Content Server includes support for Learning Standards, Objectives and Correlations and is ideal for use in primary and secondary education systems as well as corporate learning environments.

Multiple learning standards may be added by the system administrator. Each learning standard may contain it's own set of learning objectives, typically organized in a hierarchy.  Existing standards, such as those used in public education systems, may use different terminology at each level of the hierarchy. Content Server is not dependent on any pre-defined nomenclature.

## Correlations

Media may be correlated to zero or more objectives. A search can then be performed by navigating to the standard and drilling down into the objectives. Selecting an objective will present the list of search results containing all media which is correlated to the selected objective.

Reverse correlations may also be performed by selecting media, either by browsing the library or performing a keyword search. If the media has been correlated to learning objectives, those objectives will be presented in the media details pane under the Objectives tab.

## Importing Existing Standards

Standards may be easily imported into the system using the Software Development Kit. Since each standard may use a different taxonomy, a general purpose import procedure is not included. Content Server has been successfully tested and developed with the Common Core State Standards by using the API's available at the Achievement Standards Network.

Cruizemedia offers system integration options which include the importing of existing learning standards. Contact Cruizemedia support for details.

# 8 Integration

This chapter describes the integration of Content Server with third-party applications.

## RESTful Web Services API

Communication with Content Server takes place over standard HTTP using an XML-based API using RESTful web services. The API uses CruizeML, an XML schema developed by Cruizemedia specifically for Content Server. Messages sent to and received from Content Server must conform to the CruizeML specification.

Contact Cruizemedia support for details on API usage. The CruizeML XSD can be accessed from http://[*hostname*]:[*port*]/api/xsd

Access to the XSD requires the client is authenticated.

## Software Development Kit

Content Server Enterprise Edition includes a Java SDK with support for most functionality available in the in-browser client. The SDK is located in the /sdk directory where Content Server is installed. Use of the Java SDK is the recommended method of integration. The SDK requires Java 6 or higher.

The following directory structure is used by the SDK:

[*install directory*]/sdk/jars    SDK binary and documentation
[*install directory*]/sdk/lib    Third-party libraries required by the SDK

Documentation for the SDK is in JavaDoc format and is located in contentserver-sdk-doc.jar in the /sdk/jars directory.

The SDK is compatible with the same major and minor version of Content Server only. Cruizemedia uses the following versioning format:

v1.2.0 (major version = 1.  minor version = 2.  revision = 0)

All attempts are made to ensure compatibility is maintained between revisions in the same major/minor version.