COREMEDIA CONTENT CLOUD

Deployment Manual

COREMEDIA

CoreMedia GmbH

Altes Klöpperhaus, 5. OG

Rödingsmarkt 9

20459 Hamburg

July 11, 2024 (Release 2406.0)

List of Tables

1. Preface

1.1 Audience

This manual is intended for architects and developers who want to learn about the concepts of CoreMedia's flexible deployment. You should be familiar with configuration management and application deployment concepts.

Additionally, you should be familiar with *CoreMedia CMS*, *Spring Framework* and *Apache Maven*.

1.2 Typographic Conventions

CoreMedia uses different fonts and types in order to label different elements. The following table lists typographic conventions for this documentation:

Element	Typographic format	Example
Source code Command line entries Parameter and values Class and method names Packages and modules	Courier new	`cm systeminfo start`
Menu names and entries	Bold, linked with |	Open the menu entry **Format|Normal**
Field names CoreMedia Components Applications	Italic	Enter in the field *Heading* The *CoreMedia Component* Use *Chef*
Entries	In quotation marks	Enter "On"
(Simultaneously) pressed keys	Bracketed in "<>", linked with "+"	Press the keys <Ctrl>+<A>
Emphasis	Italic	It is *not* saved
Buttons	Bold, with square brackets	Click on the **[OK]** button
Code lines in code examples which continue in the next line	\	`cm systeminfo \` `-u user`

Table 1.1. Typographic conventions

In addition, these symbols can mark single paragraphs:

Pictograph	Description
	Tip: This denotes a best practice or a recommendation.
	Warning: Please pay special attention to the text.
	Danger: The violation of these rules causes severe damage.

Table 1.2. Pictographs

1.3 CoreMedia Services

This section describes the CoreMedia services that support you in running a CoreMedia system successfully. You will find all the URLs that guide you to the right places. For most of the services you need a CoreMedia account. See Section 1.3.1, “Registration” [5] for details on how to register.

NOTE

CoreMedia User Orientation for CoreMedia Developers and Partners

Find the latest overview of all CoreMedia services and further references at:

http://documentation.coremedia.com/new-user-orientation

- Section 1.3.1, “Registration” [5] describes how to register for the usage of the services.
- Section 1.3.2, “CoreMedia Releases” [6] describes where to find the download of the software.
- Section 1.3.3, “Documentation” [7] describes the CoreMedia documentation. This includes an overview of the manuals and the URL where to find the documentation.
- Section 1.3.4, “CoreMedia Training” [10] describes CoreMedia training. This includes the training calendar,the curriculum and certification information.
- Section 1.3.5, “CoreMedia Support” [10] describes the CoreMedia support.

1.3.1 Registration

In order to use CoreMedia services you need to register. Please, start your initial registration via the CoreMedia website. Afterwards, contact the CoreMedia Support (see Section 1.3.5, “CoreMedia Support” [10]) by email to request further access depending on your customer, partner or freelancer status so that you can use the CoreMedia services.

1.3.2 CoreMedia Releases

Downloading and Upgrading the Blueprint Workspace

CoreMedia provides its software as a Maven based workspace. You can download the current workspace or older releases via the following URL:

https://releases.coremedia.com/cmcc-12

Refer to our Blueprint Github mirror repository for recommendations to upgrade the workspace either via Git or patch files.

> NOTE
>
> If you encounter a 404 error then you are probably not logged in at GitHub or do not have sufficient permissions yet. See Section 1.3.1, "Registration" [5] for details about the registration process. If the problems persist, try clearing your browser cache and cookies.

Maven artifacts

CoreMedia provides parts of its release artifacts via Maven under the following URL:

https://repository.coremedia.com

You have to add your CoreMedia credentials to your Maven settings file as described in section Section 3.1, "Prerequisites" in *Blueprint Developer Manual* .

npm packages

CoreMedia provides parts of its release artifacts as npm packages under the following URL:

https://npm.coremedia.io

Your pnpm client first needs to be logged in to be able to utilize the registry (see Section 3.1, "Prerequisites" in *Blueprint Developer Manual*).

License files

You need license files to run the CoreMedia system. Contact the support (see Section 1.3.5, "CoreMedia Support" [10]) to get your licences.

1.3.3 Documentation

CoreMedia provides extensive manuals, how-tos and Javadoc as PDF files and as online documentation at the following URL:

https://documentation.coremedia.com

The manuals have the following content and use cases:

Manual	Audience	Content
Adaptive Personalization Manual	Developers, architects, administrators	This manual describes the configuration of and development with *Adaptive Personalization*, the CoreMedia module for personalized websites. You will learn how to configure the GUI used in *CoreMedia Studio*, how to use predefined contexts and how to develop your own extensions.
Analytics Connectors Manual	Developers, architects, administrators	This manual describes how you can connect your CoreMedia website with external analytic services, such as Google Analytics.
Blueprint Developer Manual	Developers, architects, administrators	This manual gives an overview over the structure and features of *CoreMedia Content Cloud*. It describes the content type model, the *Studio* extensions, folder and user rights concept and many more details. It also describes administrative tasks for the features. It also describes the concepts and usage of the project workspace in which you develop your CoreMedia extensions. You will find a description of the Maven structure, the virtualization concept, learn how to perform a release and many more.
Connector Manuals	Developers, administrators	This manuals gives an overview over the use cases of the eCommerce integration. It describes the deployment of the Commerce Connector and how to connect it with the CoreMedia and eCommerce system.
Content Application Developer Manual	Developers, architects	This manual describes concepts and development of the *Content Application Engine (CAE)*. You will learn how to write Freemarker templates that access the other CoreMedia modules and use the sophisticated caching mechanisms of the CAE.

Manual	Audience	Content
Content Server Manual	Developers, architects, administrators	This manual describes the concepts and administration of the main CoreMedia component, the *Content Server*. You will learn about the content type model which lies at the heart of a CoreMedia system, about user and rights management, database configuration, and more.
Deployment Manual	Developers, architects, administrators	This manual describes the concepts and usage of the CoreMedia deployment artifacts. That is the deployment archive and the Docker setup. You will also find an overview of the properties required to configure the deployed system.
Elastic Social Manual	Developers, architects, administrators	This manual describes the concepts and administration of the *Elastic Social* module and how you can integrate it into your websites.
Frontend Developer Manual	Frontend Developers	This manual describes the concepts and usage of the Frontend Workspace. You will learn about the structure of this workspace, the CoreMedia themes and bricks concept, the CoreMedia Freemarker facade API, how to develop your own themes and how to upload your themes to the CoreMedia system.
Headless Server Developer Manual	Frontend Developers, administrators	This manual describes the concepts and usage of the *Headless Server*. You will learn how to deploy the Headless Server and how to use its endpoints for your sites.
Importer Manual	Developers, architects	This manual describes the structure of the internal CoreMedia XML format used for storing data, how you set up an *Importer* application and how you define the transformations that convert your content into CoreMedia content.
Multi-Site Manual	Developers, Multi-Site Administrators, Editors	This manual describes different otions to desgin your site hierarchy with several languages. It also gives guidance to avoid common pitfalls during your work with the multi-site feature.

Manual	Audience	Content
Operations Basics Manual	Developers, administrators	This manual describes some overall concepts such as the communication between the components, how to set up secure connections, how to start application.
Search Manual	Developers, architects, administrators	This manual describes the configuration and customization of the *CoreMedia Search Engine* and the two feeder applications: the *Content Feeder* and the *CAE Feeder*.
Studio Developer Manual	Developers, architects	This manual describes the concepts and extension of *CoreMedia Studio*. You will learn about the underlying concepts, how to use the development environment and how to customize *Studio* to your needs.
Studio User Manual	Editors	This manual describes the usage of *CoreMedia Studio* for editorial and administrative work. It also describes the usage of the *Adaptive Personalization* and *Elastic Social* GUI that are integrated into *Studio*.
Studio Benutzerhandbuch	Editors	The Studio User Manual but in German.
Supported Environments	Developers, architects, administrators	This document lists the third-party environments with which you can use the CoreMedia system, Java versions or operation systems for example.
Unified API Developer Manual	Developers, architects	This manual describes the concepts and usage of the *CoreMedia Unified API*, which is the recommended API for most applications. This includes access to the content repository, the workflow repository and the user repository.
Utilized Open Source Software & 3rd Party Licenses	Developers, architects, administrators	This manual lists the third-party software used by CoreMedia and lists, when required, the licence texts.
Workflow Manual	Developers, architects, administrators	This manual describes the *Workflow Server*. This includes the administration of the server, the development of workflows using the XML language and the development of extensions.

Table 1.3. CoreMedia manuals

If you have comments or questions about CoreMedia's manuals, contact the Documentation department:

Email: documentation@coremedia.com

1.3.4 CoreMedia Training

CoreMedia's training department provides you with the training for your CoreMedia projects either live online, in the CoreMedia training center or at your own location.

You will find information about the CoreMedia training program, the training schedule and the CoreMedia certification program at the following URL:

http://www.coremedia.com/training

Contact the training department at the following email address:

Email: training@coremedia.com

1.3.5 CoreMedia Support

CoreMedia's support is located in Hamburg and accepts your support requests between 9 am and 6 pm MET. If you have subscribed to 24/7 support, you can always reach the support using the phone number provided to you.

To submit a support ticket, track your submitted tickets or receive access to our forums visit the CoreMedia Online Support at:

http://support.coremedia.com/

Do not forget to request further access via email after your initial registration as described in Section 1.3.1, “Registration” [5]. The support email address is:

Email: support@coremedia.com

Create a support request

Support request

CoreMedia systems are distributed systems that have a rather complex structure. This includes, for example, databases, hardware, operating systems, drivers, virtual machines, class libraries and customized code in many different combinations. That's why CoreMedia needs detailed information about the environment for a support case. In order to track down your problem, provide the following information:

- Which CoreMedia component(s) did the problem occur with (include the release number)?
- Which database is in use (version, drivers)?
- Which operating system(s) is/are in use?
- Which Java environment is in use?
- Which customizations have been implemented?
- A full description of the problem (as detailed as possible)
- Can the error be reproduced? If yes, give a description please.
- How are the security settings (firewall)?

In addition, log files are the most valuable source of information.

Support checklist

To put it in a nutshell, CoreMedia needs:

1. a person in charge (ideally, the CoreMedia system administrator)
2. extensive and sufficient system specifications
3. detailed error description
4. log files for the affected component(s)
5. if required, system files

Log files

An essential feature for the CoreMedia system administration is the output log of Java processes and CoreMedia components. They're often the only source of information for error tracking and solving. All protocolling services should run at the highest log level that is possible in the system context. For a fast breakdown, you should be logging at debug level. See Section 4.7, "Logging" in *Operations Basics* for details.

Which Log File?

In most cases at least two CoreMedia components are involved in errors: the *Content Server* log files together with the log file from the client. If you know exactly what the problem is, solving the problem becomes much easier.

Where do I Find the Log Files?

By default, application containers only write logs to the console output but can be accessed from the container runtime using the corresponding command-line client.

For the *docker* command-line client, logs can be accessed using the **docker logs** command. For a detailed instruction of how to use the command, see docker logs. Make sure to enable the timestamps using the `--timestamps` flag.

```
docker logs --timestamps <container>
```

For the *kubectl* command-line client in a Kubernetes environment you can use the **kubectl logs** command to access the logs. For a detailed instruction of how to use the command, see kubectl logs. Make sure to enable the timestamps using the `--timestamps` flag.

```
kubectl logs --timestamps <pod>
```

1.4 Changelog

In this chapter you will find a table with all major changes made in this manual.

Section	Version	Description
Chapter 4, *Encryption Service Setup* [203]	Core-Media Content Cloud v11 – 11.2307.1	Added a new chapter about configuring the encryption service with the keystore based encryption service.

Table 1.4. Changes

2. Docker Setup

CoreMedia Content Cloud offers an application modularity that supports individual application lifecycles, including a modularized development, release and deployment process. Using Spring Boot standards and container technologies at its core, our goal is to enable our applications to become more cloud native and easier to integrate with available cloud technologies.

The docker setup provided with *CoreMedia Content Cloud* consists of:

- A Maven build process to create the images
- A Docker compose setup to start a development environment

To use *CoreMedia Content Cloud* in production with Docker images, you have one of the following choices:

- Use the Docker images to start containers and benefit from the process abstraction containers provide.
- Implement a Docker Swarm setup to benefit from the simplicity of a Swarm orchestration layer.
- Implement a Kubernetes Deployment to benefit from the flexibility and power, Kubernetes provides to create a cloud native cluster deployment.

Obviously Kubernetes is the most promising technology, but the initial complexity and the steep learning curve in addition with the requirement to build and maintain the knowledge of this technology may not fit each ones requirements.

To help you with this choice, this chapter will provide you with an overview of the build process of our Docker images, the conventions and configuration options the Docker images provide and how to use them for a production deployment using the containers as a simple process abstraction.

2.1 Container Images Build Process

To build the container images for Spring Boot applications with Maven, there are currently two plugins to consider:

- Spring-Boot Maven plugin
- Google Jib Maven plugin

Both provide a tight integration into Maven but are completely different from their approach. The Spring Boot Maven plugin uses CloudNative Buildpacks whereas the Jib plugin provides a more direct approach but with the benefit of not requiring Docker at all.

We chose the Jib plugin, because it allows us to provide more customizability without the need of a custom buildpack. It also allows us to use our custom base image to provide a seamless migration. See java-application-base image [18] for more information.

2.1.1 Google Jib

Google Jib integrates with Spring Boot applications and the image will be build by the same Maven module that builds the Spring-Boot application. To activate the image build process, the Maven profile `default-image` needs to be activated.

2.1.2 Maven Structure

2.1.2.1 Properties

```
<properties>
 <jib.goal>dockerBuild</jib.goal> ❶
 <application.image-base>gcr.io/distroless/java@11</application.image-base>
 ❷
 <application.image-prefix>coremedia</application.image-prefix> ❸
 <application.image-suffix>my-app</application.image-suffix> ❹
 <application.image-tag>${project.version}</application.image-tag> ❺
 <application.image-arch>amd64</application.image-arch> ❻
</properties>
```

❶ The goal to build. Choose between either `dockerBuild` or `build`, visit the jib documentation for more details.

❷ The base image, here a Google distroless image. For deterministic build, the sha256 digest should be used. If you want to retrieve the base image from your local Docker daemon, you need to prefix the image source with `docker://`.
❸ The image prefix. You should set your registry here.
❹ The image suffix, that is the application name.
❺ The tag of the image.
❻ The architecture of the base image. In case a multi-arch OCI manifest is referenced, this determines the target platform image architecture.

2.1.2.2 Plugin

The Jib plugin has an extension mechanism and by default there are some extensions active. One of them is the Spring Boot extension, which takes care of reading the layer manifest of the Spring Boot plugin. The layered JARs mechanism was introduced with version `2.3` of Spring Boot to add application resources in layers, grouped by their nature to change. For more information, see the section about Layered Jar in the official plugin documentation.

Another extension, we make use of in our setup is the `jib-ownership-extension`. This extension can be used to set the ownership of files, see the jib ownership extension documentation for more details.

```
<plugin>
  <groupId>com.google.cloud.tools</groupId>
  <artifactId>jib-maven-plugin</artifactId>
  <version>3.4.0</version>
  <dependencies>
    <dependency>
      <groupId>com.google.cloud.tools</groupId>
      <artifactId>jib-ownership-extension-maven</artifactId>
      <version>0.1.0</version>
    </dependency>
  </dependencies>
  <configuration>
    <pluginExtensions>
      <pluginExtension>

<implementation>com.google.cloud.tools.jib.maven.extension.ownership.JibOwnershipExtension

        </implementation>
        <configuration
implementation="com.google.cloud.tools.jib.maven.extension.ownership.Configuration">

          <rules>
            <rule>
              <glob>/coremedia**</glob>
              <ownership>1000:1000</ownership>
            </rule>
          </rules>
        </configuration>
      </pluginExtension>
    </pluginExtensions>
```

```
  </configuration>
</plugin>
```

The configuration of the plugin is split up into two parts:

- A generic configuration
- The configuration to build the default image.

The Generic Configuration

```
<configuration>
  <from>
    <image>${application.image-base}</image> ❶
    <platforms>
      <platform>
        <os>linux</os>
        <architecture>${application.image-arch}</architecture>
      </platform>
    </platforms>
  </from>
  <to>
    <image>${application.image-prefix}/${application.image-suffix}</image>
❷
    <tags>
      <tag>${application.image-tag}</tag> ❸
    </tags>
  </to>
  <container>
    <appRoot>/coremedia</appRoot> ❹
    <workingDirectory>/coremedia</workingDirectory> ❺
    <ports> ❻
      <port>8080</port>
      <port>8081</port>
    </ports>
  </container>
</configuration>
```

❶ The base image to start from.
❷ The image to build. If that is a remote registry and goal is set to `build`, jib will build remotely without docker.
❸ The image tag.
❹ The application root, by default Jib will assign all files to root without write rights to others.
❺ The working directory all relative paths are calculated to.
❻ The ports that can be exposed.

The java-application-base Configuration

```
<configuration>
  <container>
    <entrypoint>INHERIT</entrypoint> ❶
```

```
    <args>
      <arg>application</arg> ❷
    </args>
    <user>coremedia:coremedia</user> ❸
    <environment> ❹
      <JAVA_HEAP>256m</JAVA_HEAP>
      <JAVA_PARALLEL_GC>true</JAVA_PARALLEL_GC>

<SPRING_BOOT_EXPLODED_MAINCLASS>@jib-main-class-file</SPRING_BOOT_EXPLODED_MAINCLASS>
 ❺

<SPRING_BOOT_EXPLODED_CLASSPATH>@jib-classpath-file</SPRING_BOOT_EXPLODED_CLASSPATH>
 ❻
    </environment>
    <ports>
      <port>8080</port>
      <port>8081</port>
    </ports>
  </container>
</configuration>
```

❶ With this directive, we tell Jib to inherit the entrypoint information from the base image. In this case it is `/coremedia/entrypoint.sh`.

❷ The argument to the entrypoint script. The `application` script encapsulates the startup logic. Visit the DockerHub page of the coremedia/java-application-base image for more details.

❸ The user to start the application. With this base image, we need to use the `coremedia` user to start the application.

❹ The environment variables the `application` entrypoint chain script can be configured.

❺ The application starter class we defined in the properties section above.

❻ The classloader classpath. This mimics the classloading, the Spring-Boot Jib extension automatically sets.

Extra files and directories

Some images require extra files and directories. A good example is the `workflow-server`. In this case and when using the `jib-compat` profile, we want to add the server-tools to start the workflowconverter in advance of starting the server itself. To do so we use the `extraDirectories` directive of the Jib plugin. For more details about how to copy additional files to the image visit the official Jib documentation about it.

2.1.3 The java-application-base Image

One of the major inventions Docker brought to software development is the ability to package applications as layers of filesystems, that when stacked upon each other result

in the final application image. This technique allows docker to reuse and share common parts between multiple images at build and startup time and reduces the disk and network footprint drastically.

In the CoreMedia Docker setup, this technique is used by providing a common base image on the public DockerHub repository. The `java-application-base` image is the common base image for all our Spring-Boot application based images. It encapsulates our best practice to configure, start and manage CoreMedia applications in most secure way. It is provided in different flavors for different JDK base images on DockerHub and it is possible to fork the source repository on GitHub, to build a customized image.

To summarize the capabilities of the image, it provides:

- A JVM to run our applications
- A startup script chain to integrate arbitrary initialization steps before starting the Spring-Boot application
- A set of convenient script chain steps to let applications delay their startup until service dependencies are available
- A monitoring integration for Prometheus

For a detailed description of the base image, please visit the java-application-base DockerHub page.

2.1.3.1 Building Distroless Images

For Kubernetes deployments, it might be interesting to build distroless images. These are images, that contain only the JVM and the application but no shell. These images are, by nature, more secure. To build a distroless image, a second Maven profile should be added:

```
    <profile>
      <id>distroless-image</id>
      <properties>
        <jib.skip>false</jib.skip>
        <application.image-base>
gcr.io/distroless/java@sha256:65aa73135827584754f1f1949c59c3e49f1fed6c35a918fadba8b4638ebc9c5d

        </application.image-base> ❶
      </properties>
      <build>
        <plugins>
          <plugin>
            <groupId>com.google.cloud.tools</groupId>
            <artifactId>jib-maven-plugin</artifactId>
            <configuration>
              <container>
                <jvmFlags> ❷
                  <jvmFlag>-XX:+UseParallelGC</jvmFlag>
                  <jvmFlag>-XX:+ExitOnOutOfMemoryError</jvmFlag>
                  <jvmFlag>-XX:IntitialRAMPercentage=45.0</jvmFlag> ❸
                  <jvmFlag>-XX:MaxRAMPercentage=70.0</jvmFlag> ❹
```

```
                <!--
                  | <jvmFlag>-XX:+HeapDumpOnOutOfMemoryError</jvmFlag>
                  | <jvmFlag>-XX:HeapDumpPath=/tmp/app.hprof</jvmFlag>
                -->
              </jvmFlags>
            </container>
          </configuration>
        </plugin>
      </plugins>
    </build>
  </profile>
```

❶ Go to https://console.cloud.google.com/gcr/images/distroless/GLOBAL/java and pick a hash

❷ When building a distroless based image, there is only the JVM included and only environment variables and jvm flags are supported.

❸ In Kubernetes the JVM memory is controlled by Kubernetes. In order to allow that, we need to define the upper and lower bound of the allocated pod memory using `IntitialRAMPercentage` and `MaxRAMPercentage`.

2.1.3.2 Building ARM Images

To build ARM images with maven, you need to set the maven property `application.image-arch` to `arm64`. If the referenced OCI manifest has a matching image in its list, the build process will be based on that architecture. Be advised, that jib is currently not able to create multi image manifests. For that reasons only one architecture can be build at a time.

2.1.3.3 Building Dockerless

Jib allows building images without Docker being installed. To do so, you need to replace the build goal from `dockerBuild` to `build`. This can be done by setting the property `jib.goal`.

When building dockerless directly into a registry, make sure your registry supports the OCI format. If not, you need to set the `<format>OCI</format>` settings in the build poms of the app modules to `<format>docker</format>`.

To start a registry locally you can start with:

```
docker run -d \
  -e REGISTRY_HTTP_ADDR=0.0.0.0:5001 \
  -p 5001:5001 \
  --name registry \
  registry:2.8
```

The port here is switched to `5001` to not collide with Apple AirPlay listener on MacOS, a common mistake you can run into.

To build the images into this registry, run:

```
mvn install -Pdefault-image \
  -Djib.goal=build
  -Djib.allowInsecureRegistries=true
  -Dapplication.image-prefix=localhost:5001
```

2.2 Docker Container Startup

2.2.1 Startup Entrypoint and Command chain

When the Docker container is started, the command of the process is created by inspecting the `Entrypoint` and the `Cmd` metadata fields of the image. If you build the images will all defaults and you run

```
docker image inspect --format='{{.Config.Entrypoint}} - {{.Config.Cmd}}'
coremedia/cae-live
```

You will get a response like this:

```
[./entrypoint.sh] - [application]
```

This is the entrypoint script chain that will be processed at startup.

2.2.2 Health Check

Docker has a built-in health check mechanism that, if defined for an image, probes a defined command regularly to determine the health status of the container. Because containerization is all about automation of processes, health checks are not only important for monitoring purposes but even more important for triggering events in an automation chain in your cluster. Many tools depend on the status information the container daemon provides. The Traefik reverse proxy uses in our Docker development setup for example automatically removes a routing to a backend, when the backends health status is not `healthy`.

If you for example run

```
docker ps --format '{{.Names}} - {{.Status}}'
```

you will get a status report for all running containers like this:

```
elastic-worker - Up 15 hours (healthy)
user-changes - Up 15 hours (healthy)
cae-preview - Up 15 hours (healthy)
cae-live - Up 15 hours (healthy)
studio-server - Up 15 hours (healthy)
overview - Up 15 hours
content-feeder - Up 15 hours (healthy)
cae-feeder-live - Up 15 hours (healthy)
commerce-adapter-wcs - Up 15 hours (healthy)
```

```
headless-server-preview - Up 15 hours (healthy)
master-live-server - Up 15 hours (healthy)
cae-feeder-preview - Up 15 hours (healthy)
headless-server-live - Up 15 hours (healthy)
studio-client - Up 15 hours (healthy)
workflow-server - Up 15 hours (healthy)
content-management-server - Up 15 hours (healthy)
```

The health check command is defined in the `Dockerfile` of the application, that is

```
HEALTHCHECK --start-period=30s --interval=30s --timeout=3s \
 CMD curl -Lf http://localhost:8081/actuator/health || exit 1
```

2.3 Container Configuration

There are three ways to configure an application:

- Build time configuration
- Runtime configuration
- Start time configuration

Please review the official Spring-Boot documentation for further details how to externalize configuration.

2.3.1 Build Time Configuration

Build time configuration can be achieved by simply predefining properties in one of the configuration files in the workspace. The location is `application.properties` in the `apps/<app name>/spring-boot/<app name>-app/src/main/resources` directory

2.3.2 Start Time Configuration

Start time configuration can be achieved using

- environment variables
- mounted configuration files

Configuring Spring-Boot applications using environment variables is probably the most common case and is the default choice in a container based environment. With Docker you simply add each environment variable on the command-line or pass in a path to a file with the environment variables. With Spring environment variables are defined in upper case and transformed by Spring to the java properties syntax.

- `REPOSITORY_URL` is transformed to `repository.url`
- `LINK_URLPREFIXTYPE` is transformed to `link.urlPrefixType`

To pass in environment variables use the `-e` or the `--env-file` options.

Passing environment variables on the command-line:

```
docker run \
 -e REPOSITORY_URL=http://my-host:40180/ior coremedia/cae-preview
```

Configuring the application by mounting a configuration file should only be used if passing variables in does not work or if replacing an existing configuration file is desired. To mount a file into the container, the path to the file or directory on the host and in the container must be defined. I.e.

Mounting configuration files:

```
docker run \
 -v $PWD/override.properties:/coremedia/config/application.properties \
 coremedia/cae-live
```

2.3.3 Runtime Configuration

To support runtime (re)configuration, the application's configuration had to be backed by a k/v store and the application would have to support it. This kind of configuration is not supported by CoreMedia applications except for some special properties that can be configured using struct content items in the content repository.

2.4 Containerized Tools

With the Docker setup all CoreMedia tools are packaged within the `coremedia/management-tools` image. You will find the Maven module to build the image at:

```
global/management-tools/management-tools-image
```

The image contains the following tools:

- content-management-server tools
- master-live-server tools
- replication-live-server tools
- workflow-server tools
- cae-feeder tools
- theme-importer

2.4.1 Running the Tools

To run the tools, you can either:

- Start the `management-tools` container in interactive mode and run the tools *from within*. This approach has the advantage, that it provides a familiar user experience as if the tool is installed locally.
- Start a tool from the `management-tools` container directly, using the tool as the command argument and remove the container when the tool finishes. This approach has the advantage, that it can be used in the context of a script as it requires no interaction.

Regardless of the approach, you need to keep in mind the following common issues with containers:

- In case you need to read from or write to files on your host, you need to mount volumes or create a bind mount.
- In case the containers are placed within a user-defined network, you need to add the container to that network to access the servers you want to connect. This network configuration depends on the network plugins being used. For the localhost `docker compose` development setup, this network is named `backend` and prefixed with either the value of the `COMPOSE_PROJECT` environment variable or the name of the directory, where the compose files lies. In the development setup this will be `compose` but you can find out what networks are available, by running:

```
docker network list
NETWORK ID     NAME              DRIVER  SCOPE
181f1af50e6e   bridge            bridge  local
2788cdf4bd7a   compose_backend   bridge  local ❶
3227b513840e   compose_web       bridge  local ❷
494f6bdaafa8   host              host    local
3d8b08335372   none              null    local
```

❶ The network, where all services are put in
❷ The network, where all services with Traefik ingress are put in

Mounting a file system path: You can mount a file system path to a docker container by using the `--volume <local-path>:<container-path>` command-line option.

Adding networks: You can attach the docker container to a network using the `--network <network name>` command-line option.

2.4.1.1 Start an Interactive Shell in the Container

To start the `management-tools` container with an interactive shell, you need to run:

```
docker run -it --rm coremedia/management-tools
```

2.4.1.2 Start a Tool Directly With a Custom Entrypoint

To start a tool, for example the `version` tool directly, you need to run:

```
docker run --rm \
  coremedia/management-tools \
  tools/bin/cm version
```

Here the tool and its arguments are passed to the entrypoint. Because the default working directory is set to `/coremedia` by the image, you only need to pass the relative path to the `cm` executable. All arguments that follow will be handled as arguments to the `cm` executable. The results of the tools are written to stdout and can easily be processed using standard shell means.

Parse an IOR URL:

```
docker run --rm \
  coremedia/management-tools \
  tools/bin/cm ior <ior string>
```

2.4.2 Configuring the Tools

Many tools have special configuration files and their values cannot be passed into the tool using environment variables. In order to configure the tools, you have two options:

- Create configuration files on your host and mount them into the container when executing the tool
- Use them embedded `confd` entrypoint chain script together with confd template configurations and render the configuration at startup using one of the many configuration backends, confd provides.

2.4.2.1 Mount Configuration Files

1. Create the config file

```
cat << EOF > capclient.properties
cap.client.server.ior.url=http://<host>:<port>/ior
EOF
```

2. Execute the tool with mounted configuration file

```
docker run --rm \
  --entrypoint /coremedia/tools/bin/cm \
  --volume \
    $PWD/capclient.properties:\
    /coremedia/tools/properties/corem/capclient.properties
  coremedia/management-tools \
  dump -u <user> -p <password> <id>
```

With mounted configuration files, you can create a separate configuration folder for each content server and for each environment you want to connect to and mount the whole folder to switch a tool from one environment to another.

2.4.2.2 Use confd to Render Configuration Files at Startup

Since `coremedia/java-application-base:2.1.6` confd is included in the base image. Confd is a tool to render configuration files using GO templates and a special set of template functions from confd.

Confd expects its configuration located at `/etc/confd/conf.d` defined using toml syntax and its templates `/etc/confd/templates`. Each template to render requires a separate configuration file.

The `management-tools` container is by default outfitted with templates for:

- UAPI connections using `capclient.properties`
- resetcaefeeder using `resetcaefeeder.properties
- sql tools using `sql.properties`
- workflowconverter using `workflowserver.properties`

The templates have been designed, so that the environment keys would be the same as if the tool would accept the Spring Boot environment variable to Spring property transformation. For the UAPI connection, this means, that defining `CAP_CLIENT_SERVER_IOR_URL` would result in rendering `cap.client.server.ior.url`.

The advantage of using confd over mounting configuration files is that confd allows setting a key path prefix on the command-line which allows us to define multiple environments within the same configuration source.

For convenience purposes in the development setup, three prefixes are preconfigured within the management-tools image:

- `dev/management`
- `dev/master`
- `dev/replication`

You can easily switch between them by using the convenience function `tools-context` included in the bashrc profile:

```
$ tools-context dev/master
[DOCKER ENTRYPOINT] - entering /coremedia/confd
```

```
[DOCKER ENTRYPOINT] - running confd with backend "env" and prefix "dev/master"
[DOCKER ENTRYPOINT] - entrypoint chain finished
```

Using confd to configure tools within a container can be especially useful when developing custom UAPI clients, without Spring configuration techniques, that should run as daemonized services within a container.

2.4.3 Examples With confd Rendered Configuration

Create config source:

```
cat << EOF > toolconfig.env
UAT_MANAGEMENT_CAP_CLIENT_SERVER_IOR_URL=http://uat-content-management-server:8080/ior
UAT_MASTER_CAP_CLIENT_SERVER_IOR_URL=http://uat-master-live-server:8080/ior
UAT_REPLICATION_1_CAP_CLIENT_SERVER_IOR_URL=http://uat-master-live-server:8080/ior
UAT_REPLICATION_2_CAP_CLIENT_SERVER_IOR_URL=http://uat-master-live-server:8080/ior
# PROD
PROD_MANAGEMENT_CAP_CLIENT_SERVER_IOR_URL=http://prod-content-management-server:8080/ior
PROD_MASTER_CAP_CLIENT_SERVER_IOR_URL=http://prod-master-live-server:8080/ior
PROD_REPLICATION_1_CAP_CLIENT_SERVER_IOR_URL=http://prod-master-live-server:8080/ior
PROD_REPLICATION_2_CAP_CLIENT_SERVER_IOR_URL=http://prod-master-live-server:8080/ior
EOF
```

Export content to your host from UAT content-management-server:

```
docker run --rm \
  --env-file=toolconfig.env \
  -e CONFD_PREFIX=uat/management \
  --volume $PWD/export:/export \
  coremedia/management-tools confd \
  tools/bincm serverexport -u admin -p admin \
  -r --basedir /export /
```

In the example above `toolconfig.env` is passed to docker and the environment variable `CONFD_PREFIX` is set to `uat/management`, which translates to the environment variable prefix `UAT_MANAGEMENT_`. In the env file the connection to the Content Management server is defined using the key `UAT_MANAGEMENT_CAP_CLIENT_SERVER_IOR_URL`. If the prefix had been set to `uat/master` the content repository of the master would have been used for the export.

2.4.4 Examples with Mounted Configuration

Create config:

```
cat << EOF > capclient.properties
cap.client.server.ior.url=http://<host>:<port>/ior
EOF
```

```
cat <<EOF > resetcaefeeder.properties
jdbc.driver=com.mysql.cj.jdbc.Driver
jdbc.url=jdbc:mysql://<host>:3306/cm_mcaefeeder
jdbc.user=cm_mcaefeeder
jdbc.password=cm_mcaefeeder
EOF

cat <<EOF > sql.properties
sql.store.driver=com.mysql.cj.jdbc.Driver
sql.store.url=jdbc:mysql://localhost:3306/cm_management
sql.store.user=cm_management
sql.store.password=cm_management
EOF
```

Export content to your host:

```
docker run --rm \
  --entrypoint /coremedia/tools/bin/cm \
  --volume \
    $PWD/capclient.properties:\
    /coremedia/tools/properties/corem/capclient.properties \
  --volume \
    $PWD/export:/export \
  coremedia/management-tools \
  serverexport -u admin -p admin \
  -r --basedir /export /
```

Import content to your host:

```
docker run --rm \
  --entrypoint /coremedia/tools/bin/cm \
  --volume \
    $PWD/capclient.properties:\
    /coremedia/tools/properties/corem/capclient.properties \
  --volume \
    $PWD/import:/import \
  coremedia/management-tools \
  serverimport -u admin -p admin \
  -r /import
```

Reset the cae-feeder-preview:

```
docker run --rm \
  --entrypoint /coremedia/tools/bin/cm \
  --volume \
    $PWD/resetcaefeeder.properties:\
    /coremedia/tools/properties/corem/resetcaefeeder.properties \
  coremedia/management-tools \
  resetcaefeeder reset
```

Unlock the content-management-server:

```
docker run --rm \
  --entrypoint /coremedia/tools/bin/cm \
  --volume \
    $PWD/sql.properties:\
    /coremedia/tools/properties/corem/sql.properties \
  coremedia/management-tools \
  unlockcontentserver
```

3. CoreMedia Properties Overview

Spring Boot configuration

Properties can be configured via Spring Boot. You can use `application.properties`, system properties, environment variables in uppercase and many more. See the official Spring Boot documentation for details.

For details on specific property data types like `Duration` or `DataSize`, see Spring Boot documentation, section Properties Conversion

Spring relaxed binding

Spring's relaxed binding also allows for different notations of property names like snake or camel case, but the default is 'kebab case' (separating words with dashes). Generally a dot in a property name reflects some kind of logical hierarchy. List-valued properties are zero-based and use bracket notation (x.y.1.* -> x.y[0].*)

NOTE

Configuration properties that are defined in the Blueprint may be missing here. Therefore, also check the Blueprint sources for classes annotated with `ConfigurationProperties` and use their JavaDoc.

3.1 Content Application Engine Properties

3.1.1 General CAE Properties

`cae.cors.allow-credentials-for-url-pattern`

Type	java.util.Map<java.lang.String,java.lang.Boolean>
Default	
Description	Map of whether user credentials are supported, based on URL patterns. Example: `cae.cors.allow-credentials-for-url-pattern[{path\:.*}]=true` See Javadoc for more information on CORS configuration for the CAE.

`cae.cors.allowed-headers-for-url-pattern`

Type	java.util.Map<java.lang.String,java.util.List<java.lang.String>>
Default	
Description	Map of headers that a pre-flight request can list as allowed for use during an actual request, based on URL patterns. A header name is not required to be listed if it is one of: Cache-Control, Content-Language, Expires, Last-Modified or Pragma. Example: `cae.cors.allowed-headers-for-url-pattern[{path\:.*}]=x-requested-with,x-csrf-token` See Javadoc for more information on CORS configuration for the CAE.

`cae.cors.allowed-methods-for-url-pattern`

Type	java.util.Map<java.lang.String,java.util.List<java.lang.String>>
Default	
Description	Map of HTTP methods to allow, based on URL patterns. Example: `cae.cors.allowed-methods-for-url-pattern[{path\:.*}]=GET,POST,PUT` See Javadoc for more information on CORS configuration for the CAE.

`cae.cors.allowed-origins-for-url-pattern`

Type	java.util.Map<java.lang.String,java.util.List<java.lang.String>>
Default	
Description	Map of origins to allow, based on URL patterns. Example: `cae.cors.allowed-origins-for-url-pattern[{path\:.*}]=https://domain1.com,https://domain2.com` In the preview CAE, this property may e.g. be configured with the Studio host in order to allow AJAX requests from the Studio to the CAE. See Javadoc for more information on CORS configuration for the CAE.

`cae.cors.exposed-headers-for-url-pattern`

Type	java.util.Map<java.lang.String,java.util.List<java.lang.String>>
Default	
Description	Map of response headers other than simple headers (i.e. Cache-Control, Content-Language, Content-Type, Expires, Last-Modified or Pragma) that an actual response might have and can be exposed, based on URL patterns. Example:

```
cae.cors.exposed-headers-for-url-pattern[{path\:.*}]=x-requested-with,x-csrf-token
```

See Javadoc for more information on CORS configuration for the CAE.

cae.cors.max-age-for-url-pattern

Type	java.util.Map<java.lang.String,java.time.Duration>
Default	
Description	Map of how long, as a duration, the response from a pre-flight request can be cached by clients, based on URL patterns. Example: `cae.cors.max-age-for-url-pattern[{path\:.*}]=3m` See Javadoc for more information on CORS configuration for the CAE.

cae.cookie.force-http-only

Type	java.lang.Boolean
Default	true
Description	Whether or not to force the 'HttpOnly' attribute on all cookies.

cae.cookie.force-secure

Type	java.lang.Boolean
Default	true
Description	Whether or not to force the 'Secure' attribute on all cookies.

cae.cookie.same-site

Type	java.lang.String
Default	

Description	The value of the cookie's 'SameSite' attribute. Valid values are the ones as defined by the spec. In addition, the value 'Unset' can be used to indicate that the attribute should not be set.

`cae.csrf.ignore-paths`

Type	java.util.List<java.lang.String>
Default	
Description	Ant Paths to ignore for CSRF prevention.

`cae.http-firewall.allow-semicolon`

Type	java.lang.Boolean
Default	false
Description	Determines if semicolon is allowed in the URL (i.e. matrix variables).

`cae.http-firewall.allow-url-encoded-double-slash`

Type	java.lang.Boolean
Default	false
Description	Determines if a double slash (//) that is URL encoded (%2F%2F) should be allowed in the path or not.

`cae.http-firewall.allow-url-encoded-percent`

Type	java.lang.Boolean
Default	false
Description	Determines if a percent (%) that is URL encoded (%25) should be allowed in the path or not.

`cae.http-firewall.allow-url-encoded-period`

Type	java.lang.Boolean

Default	false
Description	Determines if a period (.) that is URL encoded (%2E) should be allowed in the path or not.

`cae.http-firewall.allow-url-encoded-slash`

Type	java.lang.Boolean
Default	false
Description	Determines if a slash (/) that is URL encoded (%2F) should be allowed in the path or not.

`cae.http-headers.csp.directives`

Type	java.lang.String
Default	default-src 'self'; script-src 'self' 'unsafe-inline'; style-src 'self' 'unsafe-inline'; img-src 'self' data:;
Description	The CSP directives to be set. Defaults to "self". Set to empty to omit the CSP header.

`cae.http-headers.csp.report-only`

Type	java.lang.Boolean
Default	false
Description	Flag to control if the content security policy is to be reported only.

`cae.http-headers.frame-options`

Type	com.coremedia.cae.security.CaeHttpHeadersConfigurationProperties$FrameOptions
Default	disable
Description	Configure the X-Frame-Options header.

`cae.http-headers.xss`

Type	com.coremedia.cae.security.CaeHttpHeadersConfigurationProperties$XSS

Default	disabled
Description	Configure the X-XSS-Protection header.

`cae.hashing.secret`

Type	java.lang.String
Default	
Description	A Secret which is used for url parameter hashing. Needs to be at least 32 characters long. If not configured a secret will be generated and exposed via warn log on application startup. If multiple CAEs are used, ensure to set the secret instead of trusting a generated one.

`cae.link-transformer.include-params-appender.uri-paths`

Type	java.util.List<java.lang.String>
Default	/dynamic/
Description	uriPaths the IncludeParamsAppendingLinkTransformer should be applied to.

`cae.link-transformer.serializer-classes`

Type	java.util.List<java.lang.Class<?>>
Default	
Description	A list of fully qualified class names for which a com.fasterxml.jackson.databind.JsonSerializer should be registered for view parameter conversion. Every class which is configured here, should have a proper com.coremedia.id.IdScheme implementation being registered at the com.coremedia.id.IdProvider bean.

`cae.preview.metadata-enabled`

Type	java.lang.Boolean
Default	false
Description	Whether to disable metadata rendering. Disabled by default.

`cae.preview.pbe.include-jquery`	
Type	java.lang.Boolean
Default	false
Description	Configures if jquery should be included when rendering the preview related scripts.

`cae.preview.pbe.studio-url-whitelist`	
Type	java.util.List<java.lang.String>
Default	
Description	Configures a list of valid Studio URLs. The Studio Preview integration does only work for listed Studio instances. If left blank, any Studio instance is considered valid.

`cae.set-unknown-mime-type`	
Type	java.lang.Boolean
Default	false
Description	This property controls if an instance of com.coremedia.blueprint.cae.filter.UnknownMimetypeCharacterEncodingFilter is registered to fix unknown encoding errors in Webshere versions up to and including 8.5.5010.20160721_0036. The UnknownMimetypeCharacterEncoding filter will be used when cae.set-unknown-mime-type is set to true. The default is suitable when using Tomcat or Websphere starting from 8.5.5011.20161206_1434.

`cae.single-node`	
Type	java.lang.Boolean
Default	false
Description	This property is used in com.coremedia.blueprint.cae.handlers.HandlerBase#doCreateModelWithView to control if a possibly outdated resource is served or if a redirect is sent. The redirect is only a valid response when cae.single-node is set to true.

`cae.view.cycle-check`	
Type	java.lang.Boolean

Default	true
Description	Check for cyclic inclusions. You should not disable the check, unless for some good reason, e.g.: • You use a custom ViewDispatcher, whose getView method is not determined only by its arguments. • Your View#render implementation invokes ViewUtils#render with the same bean + viewName, but with an other (delegate) View.

`cae.view.debug-enabled`

Type	java.lang.Boolean
Default	false
Description	If set to true, html comments will be written to the rendered pages around included fragments. This is a development feature. With these comments you can easily see which JSP, bean and view was used to render a fragment.

`cae.view.errorhandler.enabled`

Type	java.lang.Boolean
Default	true
Description	Enables/disables the view exception handler.

`cae.view.errorhandler.output`

Type	java.lang.Boolean
Default	false
Description	If handler is enabled and set to true, exceptions will be displayed in the current page.

`cae.view.filter-lookup-by-predicate`

Type	java.lang.Boolean
Default	false

Description	By convention, templates are written for bean interfaces, but views may be named after any type. If set to true, viewlookup will only be done for views named after interfaces, not classes, with configurable excludes and includes. This option is turned off by default.

`cae.view.max-depth`

Type	java.lang.Integer
Default	200
Description	Maximum depth of inclusions.

`cae.viewdispatcher.cache.enabled`

Type	java.lang.Boolean
Default	true
Description	Defines if the caching of view lookups is enabled. Disabling might be useful when developing templates. Shouldn't be disabled when in production mode!

`cae.viewdispatcher.expose-spring-macro-helpers`

Type	java.lang.Boolean
Default	true
Description	Set whether to expose a RequestContext for use by Spring's macro library, under the name "springMacroRequestContext". Default is **"true"**. Currently needed for Spring's Velocity and FreeMarker default macros. Note that this is *not* required for templates that use HTML forms *unless* you wish to take advantage of the Spring helper macros.

`cae.viewdispatcher.fallback-to-default-view`

Type	java.lang.Boolean
Default	true

Description	Fallback to default view if requested view name raises view exception.

Table 3.1. Configuration Properties with Prefix cae

3.1.2 Delivery CAE Properties

`delivery.developer-mode`	
Type	java.lang.Boolean
Default	false
Description	Setting to enable merging and minification of resources such as JavaScript and CSS. If set to true, these resources will be merged and minfied during delivery and only a single link will be rendered to each type of resource. otherwise, each corresponding script and stylesheet will be rendered as a separate link.

`delivery.local-resources`	
Type	java.lang.Boolean
Default	false
Description	true if links to resources such as JavaScript and CSS should be generated to point to files in the application container (e.g. in modules under META-INF/resources/themes/) instead of the content repository.

`delivery.preview-mode`	
Type	java.lang.Boolean
Default	false
Description	Indicates whether this is a preview cae.

`delivery.standalone`	
Type	java.lang.Boolean
Default	true

Description	True when this is a "standalone" CAE in a servlet container serving requests directly or false when behind some URL rewriting reverse proxy (e.g. Apache).

Table 3.2. Delivery Properties

3.1.3 Http Cache Control Properties

`cae.cache-control.cache-size`

Type	java.lang.Integer
Default	10000
Description	Maximum count of cache configuration entries. Cache cleans up automatically considering LRU strategy. Default is 10000 entries.

`cae.cache-control.for-type`

Type	java.util.Map<java.lang.String,org.springframework.boot.autoconfigure.web.WebProperties$Resources$Cache$Cachecontrol>
Default	

Description	Map of initial cache control configuration objects for Http cache control Header. See WebProperties.Resources.Cache.Cachecontrol for further details. The configuration is type specific, but also can benefit from abstraction along the class hierarchy. That means that you may simply configure a common value for a super class instead of taking care about all the child classes. The type information part of the property name should be the simple name of the class (no package information) and in lowercase. Configuration of classes with the same name in different packages is not supported.
	Example Configuration:
	`cae.cache-control.for-type.cmlinkable.max-age=1m`
	`cae.cache-control.for-type.blob.max-age=180d`
	Default/Fallback:
	`cae.cache-control.for-type.object.max-age=1m`

`cae.cache-control.for-url-pattern`

Type	java.util.Map<java.lang.String,org.springframework.boot.autoconfigure.web.WebProperties$Resources$Cache$Cachecontrol>
Default	
Description	Map of initial cache control configuration objects for Http cache control Header. See Resources.Cache.Cachecontrol for further details. The configuration is URL pattern specific.

`cae.cache-control.initial-max-age-for-type`

Type	java.util.Map<java.lang.String,java.time.Duration>
Default	

Description	Initial max-age value for Cache Control Header. Does not have to be adjusted necessarily. The initial max-age value should be small, since the effective cache duration is provided with the subsequent request. The configuration is type specific, but also can benefit from abstraction along the class hierarchy. That means that you may simply configure a common value for a super class instead of taking care about all the child classes. The type information part of the property name should be the simple name of the class (no package information) and in lowercase. Configuration of classes with the same name in different packages is not supported.
	
	Example Configuration:
	
	`cae.cache-control.initial-max-age-for-type.blob=180d`
	
	Default/Fallback:
	
	`cae.cache-control.initial-max-age-for-type.object=1m`

`cae.cache-control.s-max-age-factor`

Type	java.lang.Double
Default	0
Description	This value is used to compute a s-maxage header in relation to a recognized max-age value. If the value is 0.0 then the s-maxage value is not automatically adjusted. In particular this means the s-maxage value isn't written in a cache-control header unless it was otherwise configured, for example with:

```
cae.cache-control.for-type.cmlinkable.s-max-age=1m
```

But please note, a value other than 0.0 takes precedence over a static configured value. Then s-maxage takes the value of max-age * sMaxAgeFactor.

`cae.cache-control.url-path.remove-semicolon-content`	
Type	java.lang.Boolean
Default	true
Description	Set if everything after a ";" (semicolon) should be cut from the request URI for for url-url-pattern matching. Eg. to see all matrix parameters of fragement urls this should be set to "false". Default is "true".

`cae.cache-control.url-path.url-decode`	
Type	java.lang.Boolean
Default	true
Description	Set if the request URI should be decoded for url-url-pattern matching. Default is "true".

`cae.cache-control.url-path.use-always-full-path`	
Type	java.lang.Boolean
Default	false
Description	Set if the full path within the current web application context is used for url-url-pattern matching. By default this is set to "false" which means "/servlet" is cut off.

Table 3.3. Configuration Properties with Prefix cache.control

3.1.4 CORS Properties

`cae.cors.allow-credentials-for-url-pattern`	
Type	java.util.Map<java.lang.String,java.lang.Boolean>

Default	
Description	Map of whether user credentials are supported, based on URL patterns. Example: `cae.cors.allow-credentials-for-url-pattern[{path\:.*}]=true` See Javadoc for more information on CORS configuration for the CAE.

cae.cors.allowed-headers-for-url-pattern

Type	java.util.Map<java.lang.String,java.util.List<java.lang.String>>
Default	
Description	Map of headers that a pre-flight request can list as allowed for use during an actual request, based on URL patterns. A header name is not required to be listed if it is one of: Cache-Control, Content-Language, Expires, Last-Modified or Pragma. Example: `cae.cors.allowed-headers-for-url-pattern[{path\:.*}]=x-requested-with,x-csrf-token` See Javadoc for more information on CORS configuration for the CAE.

cae.cors.allowed-methods-for-url-pattern

Type	java.util.Map<java.lang.String,java.util.List<java.lang.String>>
Default	
Description	Map of HTTP methods to allow, based on URL patterns. Example: `cae.cors.allowed-methods-for-url-pattern[{path\:.*}]=GET,POST,PUT` See Javadoc for more information on CORS configuration for the CAE.

cae.cors.allowed-origins-for-url-pattern

Type	java.util.Map<java.lang.String,java.util.List<java.lang.String>>
Default	
Description	Map of origins to allow, based on URL patterns. Example: `cae.cors.allowed-origins-for-url-pattern[{path\:.*}]=https://domain1.com,https://domain2.com` In the preview CAE, this property may e.g. be configured with the Studio host in order to allow AJAX requests from the Studio to the CAE. See Javadoc for more information on CORS configuration for the CAE.

cae.cors.exposed-headers-for-url-pattern

Type	java.util.Map<java.lang.String,java.util.List<java.lang.String>>
Default	
Description	Map of response headers other than simple headers (i.e. Cache-Control, Content-Language, Content-Type, Expires, Last-Modified or Pragma) that an actual response might have and can be exposed, based on URL patterns. Example: `cae.cors.exposed-headers-for-url-pattern[{path\:.*}]=x-requested-with,x-csrf-token` See Javadoc for more information on CORS configuration for the CAE.

cae.cors.max-age-for-url-pattern

Type	java.util.Map<java.lang.String,java.time.Duration>
Default	
Description	Map of how long, as a duration, the response from a pre-flight request can be cached by clients, based on URL patterns. Example:

```
cae.cors.max-age-for-url-pattern[{path\:.*}]=3m
```

See Javadoc for more information on CORS configuration for the CAE.

Table 3.4. Configuration Properties with Prefix cae.cors

3.1.5 Blob Transformation properties

`com.coremedia.transform.blob-cache.base-path`	
Type	java.nio.file.Path
Default	
Description	The path to the transformed image blobs cache. If not set (which is the default), then this cache is deactivated and the results of image transformations are stored using the UAPI cache. It is recommended to set this property to a folder that is not cleared during CAE restart, for example, /var/cache/coremedia/persistent-transformed-blobcache.

`com.coremedia.transform.blob-cache.log-interval-seconds`	
Type	java.time.Duration
Default	600s
Description	Delay for logging metrics of the transformed BLOB cache. The default unit is "seconds".

`com.coremedia.transform.blob-cache.size`	
Type	org.springframework.util.unit.DataSize
Default	1GB
Description	The maximum allowed size that the transformed image blob cache can occupy on the disk. This is a separate cache where results of blob transformations are stored persistently and can survive CAE restarts.

	Note that the file system overhead for storing the files does not count towards this value. So the physical space that has to be reserved on the disk for the cache has to be slightly higher than value of this configuration property. If several concurrent threads write large blobs at the same time, the deletion of the folder with the old unused files can be postponed for later, thus this is the second reason why the maximum allowed cache size can grow slightly higher than this configuration property. The size of such deviation depends on the blobs size as well as the amount of parallel threads. The default unit is "bytes".

`com.coremedia.transform.memory-guard.safety-factor`

Type	java.lang.Double
Default	1.2
Description	A safety factor of the memory guard configured for the imageTransformer. Setting this to zero disables memory checks.

`com.coremedia.transform.throttle.fallback-factor`

Type	java.lang.Integer
Default	1
Description	A factor to multiply the blob size with if the size of the in-memory representation needed for transformation cannot be determined. A warning is logged if this happens.

`com.coremedia.transform.throttle.hard-limit`

Type	java.lang.Boolean
Default	true
Description	Specifies whether the configured permits should be considered a hard limit for transformations. Setting this to *false* allows transformations needing more permits (serialized, one at a time). This would allow larger images to be transformed, but would make OutOfMemoryErrors during transformations more likely.

`com.coremedia.transform.throttle.permits`

Type	java.lang.Integer
Default	0
Description	Capacity of the ThrottlingBlobTransformer. Default is 0 (zero), which falls back to 1/4 of the total JVM memory size (in megabytes).

`com.coremedia.transform.throttle.safety-factor`

Type	java.lang.Double
Default	3
Description	A safety factor for the ImageSizePermitStrategy to multiply the memory size of an image with. The default number of 3 takes into account that some image operations need to make a copy of the image, plus one copy for the in-memory cache for loaded images.

Table 3.5. Blob Transformation Properties

`cache.capacities.com.coremedia.transform.image.java2d.LoadedImageCacheKey`

Value	Number
Default	100 MB
Description	The size of the loaded image cache, in bytes.

Table 3.6. Further Configuration Properties

3.1.6 Renamed CAE Properties

The following list contains renamed *CAE* configuration properties. Use the new names, the old names are deprecated.

Deprecated Name	New Name

Table 3.7. Renamed CAE Configuration Properties

3.2 Content Server Properties

3.2.1 General Content Server Properties

`cap.server.allow-synthetic-replay`	
Type	java.lang.Boolean
Default	true
Description	Whether it is allowed for clients to request a synthetic replay of the content repository, for example using the constant Timestamp.SYNTHETIC_REPLAY Unified API. This is a very expensive operation that is rarely used except when setting up a Replication Live Server from scratch.

`cap.server.base-dir`	
Type	java.lang.String
Default	user.dir
Description	Used as base directory for handling relative file-paths in the content-server configuration. Defaults to the value of system-property user.dir.

`cap.server.blob-channel-timeout`	
Type	java.lang.Integer
Default	60
Description	This property sets the timeout for streaming blobs to and from the database in seconds. In general, you don't have to change the default value. It is provided for exceptional cases, when the connection to the database is unreliable.

`cap.server.blob-md5-permission-check`	
Type	java.lang.Boolean

Default	true
Description	Blob access by clients is protected against URL guessing. If you want to use clients from releases < CMCC 2110 with content servers of CMCC 2110 or newer, you must set this flag to false for the new content servers.

`cap.server.blob-url-pattern`

Type	java.lang.String
Default	https?:.*
Description	A regular expression that must match the entire URL string for URL blobs, that is, blobs that are only referenced by their URL in the content repository and are resolved at the client side on access. Using URL blobs can significantly reduce the storage requirements of the Content Server. The pattern must match the entire URL string, starting with the protocol. Before matching, the path component of the URL is normalized according to java.net.URI#normalize() where applicable. URLs with the schemes s3: and classpath: are not normalized. By default, only http: and https: URLs are allowed. Allowing too many URLs may cause security problems. For example, allowing file:.* would also grant access to all configuration files. Instead, a single path like file:///share/blobData/.* should be sufficient in most cases. Alternative patterns can be given according to the Java regular expression syntax as implemented by java.util.regex.Pattern: file:///share/blobData/.*|http://blobstorage.internal/.*

`cap.server.cache.group-cache-size`

Type	java.lang.Integer
Default	500
Description	This property defines the size of the group cache. It limits the maximum number of groups which can be found in one search for groups by CMS clients. Set the property so that all groups connected to rights can be cached in memory.

`cap.server.cache.group-cache-status-interval`

Type	java.lang.Integer
Default	0

Description	The time between two log messages reporting the current state of the group cache, in seconds; the maximum value is 3600, the minimum value is 10.

`cap.server.cache.member-folder-rights-cache-size`

Type	java.lang.Integer
Default	1000
Description	This property defines the size of the folder-specific rights cache. This cache stores the results of right calculations per folder and member, aggregating the results for all content types. This cache might help custom code using APIs other than the Unified API. Change this setting only if you observe the method getRights(MemberKey) in thread dumps of a slow Content Server.

`cap.server.cache.resource-cache-size`

Type	java.lang.Integer
Default	60000
Description	The capacity of the resource cache of the Content Server; the maximum value is 1000000, the minimum value is 100. This property defines the resource cache size, that is, the number of resources the server holds in memory. This value should sometimes be adapted to the increasing number of resources in the actual working set. If the value is too small, the server does not perform well. One resource needs about 2kB of heap space.

`cap.server.cache.resource-cache-status-interval`

Type	java.lang.Integer
Default	300
Description	The time between two log messages reporting the current state of the resource cache, in seconds; the maximum value is 3600, the minimum value is 10.

`cap.server.cache.rights-cache-size`

Type	java.lang.Integer
Default	3000

Description	This property defines the size of the rights cache. This cache stores the results of right calculations per resource, content type and member. If you have lots of different resources, content types and users you might need to adapt the value of the property. Check the proper size of the cache by examining the cache misses and faults in the log. To activate the log output of the rights cache set the cap.server.cache.rights-cache-status-interval property to a value larger than zero.

`cap.server.cache.rights-cache-status-interval`

Type	java.lang.Integer
Default	0
Description	This property defines the interval (in seconds) at which log output of the rights cache is written. "0" means, that no log output is written.

`cap.server.cache.user-cache-size`

Type	java.lang.Integer
Default	500
Description	This property defines the size of the user cache. It limits the maximum number of users which can be found in one search for users by CMS clients. Set the property to the size of the largest user search you want to perform, or the number of concurrently working users, whichever is greater.

`cap.server.cache.user-cache-status-interval`

Type	java.lang.Integer
Default	0
Description	The time between two log messages reporting the current state of the user cache, in seconds; the maximum value is 3600, the minimum value is 10.

`cap.server.check-unique-db-access`

Type	java.lang.Boolean
Default	true

Description	This property determines whether to check for another server that is running concurrently on the same database on server startup.

`cap.server.document-types`

Type	java.util.List<java.lang.String>
Default	classpath*:framework/doctypes/**/*.xml
Description	This property defines where the server finds the XML file(s) containing the content type definitions. The pattern supports resources from classpath or the filesystem. Relative file-paths are resolved against cap.server.base-dir). Examples: `classpath*:/framework/doctypes/**/*.xml (Default), file:///coremedia/doctypes/**/*.xml, config/contentserver/doctypes/**/*.xml`

`cap.server.encrypt-passwords-key-file`

Type	java.lang.String
Default	
Description	The location of the key generated by cm encryptpasswords (absolute or relative to cap.server.base-dir). If empty, defaults to etc/keys/DATABASE_NAME.DATA-BASE_USER.rijndael

`cap.server.http-port`

Type	java.lang.Integer
Default	0
Description	Defines the HTTP(S) port of the application container containing the Content Server. The entry /Server/Service/Connector@port in server.xml has to have the same value as this property.

`cap.server.init-runlevel`

Type	java.lang.String
Default	online
Description	The initial runlevel that the server will try to reach on startup. Possible runlevels are: online, administration, maintenance. This property does not override the default behavior of the Replication Live Server for the initial replication.

`cap.server.initial-password`

Type	java.util.Map<java.lang.String,java.lang.String>
Default	
Description	The initial password to set for the default user with the indicated name. This password is set when the server is started for the first time. You can change the passwords later on at any time.

`cap.server.license`

Type	java.lang.String
Default	license.zip
Description	Defines where the server finds the license file (url or file-path absolute or relative to cap.server.base-dir).

`cap.server.login-service-webserver-privileged`

Type	java.lang.Boolean
Default	false
Description	This property specifies whether client connections of the login service 'webserver' are privileged and may log in as different users without further authentication. The default is false. It may be set to true to make the 'webserver' login service a privileged service as it was the case in releases before 1907.

`cap.server.login.authentication`

Type	java.lang.String

Default	classpath:coremedia-jaas.conf
Description	Defines where the server finds the JAAS login configuration file (url or file-path absolute or relative to cap.server.base-dir). Will be set on the java.security.auth.login.config System property (in case of a URL, a temporary file is created).

`cap.server.login.bouncers`

Type	java.lang.String
Default	
Description	This property points to the optional login bouncer configuration file (absolute or relative to cap.server.base-dir). A login bouncer can grant or deny access to the Content Server based on the characteristics of the user and the set of currently logged in users.

`cap.server.login.password-hash-algorithm`

Type	java.lang.String
Default	bcrypt:10
Description	A specification of the hash algorithm used for storing passwords. Allowed values are md5 for MD5-based password hashing and bcrypt:N with N being an integer between 4 and 30 (inclusive) for bcrypt-based password hashing. In the latter case N denotes the work factor which should adapted to the available CPU resources. This parameter applies to passwords of users defined in the built-in user repository of the Content Server, only. The value md5 is discouraged, because it makes brute-force attacks on passwords of low and medium strength possible. It should only be used if passwords need to be changed by clients (Studio, cm changepassword) that have not been updated to a CMS release that supports configurable password hashing. Old clients can login even after a password change without any restrictions. After changing this property, it is recommended to update the passwords of all users to ensure that all hashes have been computed according to the desired algorithm.

`cap.server.maximum-startup-delay`

Type	java.lang.Integer
Default	60

Description	Configured maximum time for Content Server startup in seconds. This is the maximum time after which the Content Server is treated as initialized if it was started as Windows Service or as web application. Dependent Windows Services / other web applications will be started when the Content Server has reached its initial runlevel or after this time.

`cap.server.multiple-live-servers`

Type	java.lang.Boolean
Default	false
Description	This property defines whether the server publishes to multiple live servers. Note that this flag cannot be easily changed after the first start of the Content Management Server.

`cap.server.naming-policy-allow-at`

Type	java.lang.Boolean
Default	false
Description	Allow '@' in member names (may cause confusion with user domains)

`cap.server.persistent-property-writers`

Type	java.util.List<java.lang.String>
Default	*
Description	A list of names of groups that may write or delete persistent properties. The magic group "*" grants rights to all users. This is the default for compatibility reasons. Connections using the publisher and replicator login service are always allowed to write persistent properties.

`cap.server.repository-home`

Type	java.lang.String
Default	/Home
Description	Defines the folder which will be used to store the home folders of the users. The whole folder hierarchy of the home folders is only visible to the administrator. Other user will

	only see one home folder with the path defined in cap.server.repository-home containing his personal files, such as the preferences. The default folder is /Home which will be automatically created by the system. If you define another folder, you need to create this folder by your own.

`cap.server.repository-system`

Type	java.lang.String
Default	/System
Description	Defines the system folder. It contains for example the public dictionary of the spell checker. The default folder is /System which will be automatically created by the system. If you define another folder, you need to create this folder by your own.

`cap.server.search.enable`

Type	java.lang.Boolean
Default	false
Description	If true full text search is enabled.

`cap.server.session-ping-interval`

Type	java.lang.Integer
Default	60
Description	The maximum number of seconds that a ping is delayed when there are no available events

`cap.server.termination-timeout-seconds`

Type	java.lang.Integer
Default	30
Description	Timeout for waiting on running threads on shutdown

`cap.server.unique-db-access-write-interval`

Type	java.lang.Integer
Default	0
Description	If unique DB access is checked and if this property is positive, this property determines the number of seconds between two writes of the current timestamp to the database to indicate the liveliness of the server; if 0 or negative the server neither writes a timestamp regularly nor expects a timestamp to be written;

`cap.server.use-strict-workflow`

Type	java.lang.Boolean
Default	false
Description	This property enforces the strict workflow mode. That is, the approver of a resource must be different from the editor. This is checked independently of the workflow engine, and should only be used in cases where a custom workflow definition is not an option.

`cap.server.userproviders`

Type	java.util.List<hox.corem.server.ServerConfigurationProperties$Userproviders>
Default	
Description	Configurations for UserProviders For details see hox.corem.server.ServerConfigurationProperties.Userproviders.

`cap.server.blobstore.s3.bucketname`

Type	java.lang.String
Default	
Description	The name of the S3 bucket that is used to store blobs. If this property is not specified, the S3 media store is not enabled.

`cap.server.blobstore.s3.rootdir`

Type	java.lang.String

Default	
Description	The name of an S3 path prefix, used to disambiguate multiple media stores in one S3 bucket. If possible, it is recommended to use separate buckets instead, improving performance by avoiding constant path prefixes.

Table 3.8. Content Server Properties

3.2.2 CORBA Properties

`com.coremedia.corba.client.local-socket`

Type	java.lang.Boolean
Default	false
Description	Redirect all outgoing connections to localhost. This is useful for running UAPI clients locally, while forwarding HTTP and CORBA connections to the servers through an SSH tunnel. A typical SSH invocation that allows the use of this socket factory would look like this, forwarding all ports of the *Content Management Server* and the *Workflow Server*: ssh -L 40180:SERVER:40180 -L 40183:SERVER:40183 -L 40380:SERVER:40380 -L 40383:SERVER:40383 HOST When using this option, you should also activate com.coremedia.corba.server.setNoSocket(boolean) in order to reduce the attack surface of the JVM.

`com.coremedia.corba.client.redirect.original-host`

Type	java.lang.String
Default	
Description	Redirect IIOP invocations to other addresses. Useful in tunnelling scenarios: imagine a server X on port P running in a network you cannot reach; but you are able to create a tunnel that forwards your address Y:Q to X:P, e.g. you do user@Y % ssh -g -LQ:X:P Then you can configure your clients by the com.coremedia.corba.client.redirect property group:

- originalHost: The original server's name (X). If set, only IORs pointing to that server will be redirected.
- redirectHost: The new server name (Y). If set, this name will be placed in the new target address.
- originalPort: The original server's port (P). If set, only IORs pointing to that port will be redirected.
- redirectPort: The new server port (Q). If set, this port will be placed in the new target address.

`com.coremedia.corba.client.redirect.original-port`

Type	java.lang.Integer
Default	-1
Description	Redirect IIOP invocations to other addresses. For details see com.coremedia.corba.client.redirect.originalHost.

`com.coremedia.corba.client.redirect.redirect-host`

Type	java.lang.String
Default	
Description	Redirect IIOP invocations to other addresses. For details see com.coremedia.corba.client.redirect.originalHost.

`com.coremedia.corba.client.redirect.redirect-port`

Type	java.lang.Integer
Default	-1
Description	Redirect IIOP invocations to other addresses. For details see com.coremedia.corba.client.redirect.originalHost.

`com.coremedia.corba.client.ssl.clear-text-ports`

Type	java.lang.String

Default	
Description	The Corba clear text ports Usually one, maybe two (comma separated) values for *Content Server* and *Workflow Server*. In case of multiple values, the order must correspond to the com.coremedia.corba.client.ssl.sslPorts.

`com.coremedia.corba.client.ssl.keystore`

Type	java.lang.String
Default	
Description	The keystore for SSL encrypted communication

`com.coremedia.corba.client.ssl.passphrase`

Type	java.lang.String
Default	
Description	The passphrase for SSL encrypted communication

`com.coremedia.corba.client.ssl.ssl-ports`

Type	java.lang.String
Default	
Description	The Corba SSL text ports Usually one, maybe two (comma separated) values for *Content Server* and *Workflow Server*. In case of multiple values, the order must correspond to the com.coremedia.corba.client.ssl.clearTextPorts.

`com.coremedia.corba.server.host`

Type	java.lang.String
Default	
Description	The host of the ORB.

	This hostname is exposed to the client via the IOR. Normally, you do not need to care about this, but in runtime environments with special DNS configurations it may be necessary to set a certain name that the client is able to resolve.

`com.coremedia.corba.server.no-socket`

Type	java.lang.Boolean
Default	true
Description	Disable ORB server sockets. This is useful when no incoming requests are expected, so that the server socket would only pose a security risk. The default is true, i.e. server sockets are disabled. This is appropriate for all client applications and must be set to false only for *Content Server* (CMS, MLS and RLS) and *Workflow Servery* applications.

`com.coremedia.corba.server.port`

Type	java.lang.Integer
Default	-1
Description	The clear text port of the ORB.

`com.coremedia.corba.server.single-ip`

Type	java.lang.String
Default	
Description	Bind ORB sockets to this IP address. Useful to keep control on multi-homed hosts.

`com.coremedia.corba.server.ssl.enforce`

Type	java.lang.Boolean
Default	false

Description	Enforce SSL communication By default, the ORB opens an SSL port additionally to the clear text port. With this flag you can suppress the clear text port and thus ensure that clients use SSL connections.
`com.coremedia.corba.server.ssl.keystore`	
Type	java.lang.String
Default	
Description	The keystore for SSL encrypted communication
`com.coremedia.corba.server.ssl.passphrase`	
Type	java.lang.String
Default	
Description	The passphrase for SSL encrypted communication
`com.coremedia.corba.server.ssl.ssl-port`	
Type	java.lang.Integer
Default	-1
Description	The port of the ORB for SSL encrypted communication. The orb will open the SSL socket *in addition* to the clear text port. There is no way to disable this.

Table 3.9. CORBA Properties

3.2.3 Properties for the Publisher

In the properties given below, some properties take a different form when using *CoreMedia Multi-Master Management*. In that case, an index number is used to indicate the publication target that is configured using this property value. The index numbers are consecutive integers starting with 1 and running up to the number of publication targets.

`publisher.autoextend.latest-approved-version`

Type	java.lang.Boolean
Default	true
Description	This property affects the automatic extension of publication sets. The default behavior ("true") is as follows: If the referenced content item is not published yet, publish its latest (= newest) approved version. If the referenced content item is already published, do nothing If set to "false", not the latest but the earliest approved version is published.

`publisher.autoextend.max-failures`

Type	java.lang.Integer
Default	0
Description	The maximum number of implicitly added content items and folders whose publication may fail before no further content items and folders may be added implicitly. If set to 0 (the default), the publisher never stops adding content items and folders.

`publisher.autoextend.update-linked-documents`

Type	java.lang.Boolean
Default	false
Description	This property affects the automatic extension of publication sets. By default, ("false") only direct linked and approved content items are added to the publication set. If set to "true", all linked content items are recursively added to the publication set. The recursion stops when a version is already published and may lead to surprisingly large publication sets. Setting updateLinkedDocuments implicitly also sets latestApprovedVersion to "true".

`publisher.connection-timeout-seconds`

Type	java.lang.Integer
Default	300
Description	The timeout in seconds for the connection to the Master Live Server ior.

`publisher.destroy-intermediate-versions`

Type	java.lang.String
Default	strict
Description	Whether intermediate content item versions between two publications will be destroyed or not on the Content Management Server. Example for dumb mode: Version 1 of content item A was published. In the meantime, the versions 2, 3, 4 and 5 have been created. When you now publish version 5, the versions 2, 3, 4 are destroyed and only version 1 and 5 remain on the Content Server. Old flags true and false are supported, too. false maps to off while true maps to strict. Available Modes • off Turns destruction of intermediate versions off. • dumb Simply destroys all versions between the currently published one and the previously published one; if you run into performance issues on publication or you do not use a multi-site set up this should be chosen. Mind that for multi-site set up this setting might break translation states. • strict Destroys all versions between the currently published one and the previously published one, but only if the versions are not referenced by master/masterVersion properties. This ensures that for a multi-site set up your translation state will be valid. This setting is recommended for multi-site set ups.

`publisher.destroy-older-versions-on-live-servers`

Type	java.lang.Boolean
Default	true
Description	Whether older published versions should be destroyed ("true") on the Master Live Server. That is, only two published versions (the current and the one before) of each content item remain on the Master Live Server. Only change if you have a valid reason.

`publisher.enable-bypass-previews`

Type	java.lang.Boolean

Default	true
Description	Whether publication previews bypass ("true") or not ("false") the publication queue for faster response times.

`publisher.local.domain`

Type	java.lang.String
Default	
Description	The domain for logging in locally to the Content Management Server.

`publisher.local.password`

Type	java.lang.String
Default	publisher
Description	The password for logging in locally to the Content Management Server.

`publisher.local.user`

Type	java.lang.String
Default	publisher
Description	The user name for logging in locally to the Content Management Server.

`publisher.priority`

Type	java.util.Map<java.lang.String,java.lang.Integer>
Default	
Description	Priorities of publications requested through: • guiEditor: 60 • uapi: 40 • generator: 40 • jpython: 20 • importer: 20

- utility: 20
- unknown: 20

`publisher.target`	
Type	java.util.List<hox.corem.server.publish.PublisherConfigurationProperties$Target>
Default	
Description	target is a list-valued property. You configure it with publisher.target[0].name, for example. Values other than "0" are only required for the multi-master feature, that is publication to different Master Live Servers. Target has the following properties: name • **Value:** java.lang.String • **Default:** • **Description:** The permanent and unique name of the publication target. Once set, it should never be changed, as this name is used for target identification in the APIs and in JMX. displayName • **Value:** java.lang.String • **Default:** • **Description:** The display name is shown to users when no localized information about a publication target is available; display names, too, should be unique, but they may well change to better illustrate the current uses of a publication target. user • **Value:** java.lang.String • **Default:** publisher • **Description:** The URL where the publisher can obtain the IOR of the Master Live Server. password • **Value:** java.lang.String • **Default:** publisher • **Description:** The password for logging in to the Master Live Server. domain • **Value:** java.lang.String

- **Default:**
- **Description:** The domain for logging in to the Master Live Server.

iorUrl

- **Value:** java.lang.String
- **Default:**
- **Description:** The URL where the publisher can obtain the IOR of the Master Live Server.

folders

- **Value:** java.lang.String
- **Default:**
- **Description:** The base folders that are assigned to a publication target. This property typically references exactly one top-level folder, either by name or by its numerical ID. If more than one site is generated from a single Live Server, multiple top-level folders may be given, separated by commas. When indicating a folder by name, that name is blocked for rename operations on the top-level folder. Once you have assigned a folder to a publication target, it must not be reassigned to another target. Doing so would result in inconsistencies between Content Management Server and Master Live Server.

Table 3.10. Publisher Properties

3.2.4 Properties for the Connection to the Database

NOTE

sql.properties for Command Line Tools

Be aware that the command line tools that access the database directly (like `cm unlockcontentserver` or `cm schemaaccess`) still need the `sql.properties` file in their `properties/corem` directory, as the tools are not (yet) configurable via Spring.

`sql.pool.check-connection-max-threads`

Type	java.lang.Integer

Default	-1
Description	The maximum number of threads for checking database connections, or -1 for twice the number of sql.pool.max-connections.

`sql.pool.check-timeout`

Type	java.lang.Integer
Default	5
Description	Maximum time in seconds the check is allowed to take.

`sql.pool.log-queries`

Type	java.lang.Boolean
Default	false
Description	If the property is "true", messages concerning queries (search of content item versions) will be generated.

`sql.pool.log-query-statements`

Type	java.lang.Boolean
Default	false
Description	If the property is "true", SQL statements concerning queries will be written to the log.

`sql.pool.log-schedule-messages`

Type	java.lang.Boolean
Default	false
Description	If the property is "true", write messages of the connection pool to the log file.

`sql.pool.log-verbose`

Type	java.lang.Boolean

Default	false
Description	If the property is "true", more debug messages will be generated.
`sql.pool.max-connections`	
Type	java.lang.Integer
Default	4
Description	Maximum number of connections to the database.
`sql.pool.max-queries`	
Type	java.lang.Integer
Default	4
Description	Maximum number of connections used for queries, that is, the maximum number of parallel queries.
`sql.pool.max-retries`	
Type	java.lang.Integer
Default	15
Description	Maximum number of retries on job that failed due to read-only DB or connection loss.
`sql.pool.min-connections`	
Type	java.lang.Integer
Default	2
Description	Minimum number of connections to the database.
`sql.pool.open-timeout`	
Type	java.lang.Integer

Default	30
Description	Maximum time in seconds the opening of connections is allowed to take.

`sql.pool.reaper-interval`

Type	java.lang.Long
Default	120
Description	The interval in seconds in which it is checked if connections can be closed.

`sql.pool.reaper-timeout`

Type	java.lang.Long
Default	180
Description	The time in seconds a connection must be idle before it will be closed.

`sql.pool.retry-delay`

Type	java.time.Duration
Default	15s
Description	Delay interval before a retry is attempted due to a read-only DB or connection loss.

`sql.pool.retry-on-connection-loss`

Type	java.lang.Boolean
Default	false
Description	If "true", will retry failed transactions due to db connection failure.

`sql.pool.retry-on-read-only-db`

Type	java.lang.Boolean
Default	false

Description	If "true", will retry failed transactions due to a read-only database.

`sql.pool.scheduler-pull-up-ratio`

Type	java.lang.Double
Default	0.1
Description	Ratio of non-exclusive elements to pull up in scheduler queue in case of a potential deadlock (in the range]0.0,1.0]).

`sql.pool.scheduler-watch-interval`

Type	java.time.Duration
Default	10s
Description	The interval between two checks for potential scheduler deadlocks.

`sql.pool.scheduler-watchdog-enabled`

Type	java.lang.Boolean
Default	true
Description	Flag to enable scheduler watchdog.

`sql.pool.validator-interval`

Type	java.lang.Long
Default	300
Description	The interval in seconds in which existing connections will be checked for function.

`sql.pool.validator-timeout`

Type	java.lang.Long
Default	120

Description	The time in seconds a connection must be idle before it will be checked for function.

`sql.schema.alter-table`

Type	java.lang.Boolean
Default	false
Description	Setting the property to "true", causes the Content Server to automatically add database columns for newly defined content type properties and to fix wrong widths of String properties. This will only work if you set sql.schema.checkColumns to "true".

`sql.schema.check-columns`

Type	java.lang.Boolean
Default	true
Description	Setting the property sql.schema.checkColumns to "true", causes the Content Server to compare the content type definition with the existing database schema for missing columns and matching widths of String properties. If there are differences, it depends on the setting of sql.schema.alterTables if the Content Server refuses to start ("false") or if it adds and changes the columns automatically ("true"). Checking the tables consumes a considerable amount of time, so that the server starts up more slowly. If sql.schema.checkColumns is set to "false", the Content Server will not check the columns. If there are differences, you will run into Content Server exceptions later on.

`sql.schema.create-drop-indexes`

Type	java.lang.Boolean
Default	false
Description	Setting the property to "true", causes the Content Server to automatically create and drop indexes on content type properties, according to the Index attribute in the doctypes.xml. This flag only affects existing columns. For newly added columns and tables, an index is always created if the Index attribute is set.

`sql.schema.create-table`

Type	java.lang.Boolean

Default	true
Description	The Content Server always checks if tables for the content types are missing. Setting the property to "true", causes the Content Server to create missing tables for new content types. If the setting is "false" and there are missing tables the Content Server refuses to start

`sql.store.collector.blob-preservation-period`

Type	java.lang.Integer
Default	86400
Description	The time in seconds, a blob, that has no reference to a resource, will be preserved.

`sql.store.collector.delay`

Type	java.lang.Integer
Default	86400
Description	The delay between two collections in seconds.

`sql.store.collector.initial-delay`

Type	java.lang.Integer
Default	600
Description	The delay in seconds before starting the first collection.

`sql.store.collector.start-time`

Type	java.lang.Integer
Default	-1
Description	The time of the day, in seconds from 0:00h in the default time zone, when the collector should run every day. If set to -1, it does not run at a fixed time. The initial delay is always enforced as a minimal delay after server startup.

`sql.store.collector.suspend`

Type	java.lang.Boolean
Default	false
Description	If set to true, unused blobs will not be deleted in the blob store. This can be used during backup if you have a non-transactional blob store. Alternatively, the blobcollector actuator endpoint must be used to suspend deletion of unused blobs at runtime before starting a backup of a non-transactional blob store.

`sql.store.convert`

Type	java.util.Map<java.lang.String,java.lang.String>
Default	
Description	Converters which are used to convert custom XML formats which have been defined for the obsolete coremedia-sgmltext.dtd. **Example**: sql.store.convert.DocumentType.PropertyType=com.customer.XMLConverter

`sql.store.convert-correct-rich-text`

Type	java.lang.Boolean
Default	true
Description	The editor in versions before SCI 4.1.38 created XML text, which was not valid according to the coremedia-richtext-1.0.dtd. If you have created content with versions before SCI 4.1.38, set the property to "true" to correct these errors (CoreMedia recommends to always use "true").

`sql.store.driver`

Type	java.lang.String
Default	
Description	The JDBC driver used to connect to the database. **Example:** oracle.jdbc.driver.OracleDriver

`sql.store.folder-index-timeout`

Type	java.time.Duration
Default	1m
Description	Maximum time to wait for an enabled folder index to become up-to-date when a content query with BELOW-clause is executed. Folder index updates may take some time, if large folders were moved. If the maximum time has been reached, the query will be executed as if the folder index was offline. Set to negative value for unlimited.

`sql.store.generate-blob-ids`

Type	java.lang.Boolean
Default	true
Description	Whether the store generates blob ids on its own. Defaults to "true". If set to "false", the store relies on externally provided blob ids. This allows the use of shared media stores for MLS and RLS. This flag must not be changed after the first start of a Content Server.

`sql.store.isolation`

Type	java.lang.String
Default	
Description	Define the transaction isolation level. An empty value uses the driver default setting.

`sql.store.log-driver-messages`

Type	java.lang.Boolean
Default	false
Description	Write messages of the JDBC driver to the logs • hox.corem.server.sql.SQLStore and • com.coremedia.cotopaxi.server.DatabaseProperties.

`sql.store.login-user-name`

Type	java.lang.String

Default	
Description	The user name for a database login. If not set, the value of "sql.store.user" will be used to log in to the database. In some cases the login username differs from the actual user, e.g. with PostgreSQL on Azure a postfix on the user name is necessary to log in. Example: • sql.store.login-user-name=username@domain • sql.store.user=username

`sql.store.password`

Type	java.lang.String
Default	
Description	The password of the user at the database.

`sql.store.prepared-statement-cache-size`

Type	java.lang.Integer
Default	-1
Description	If set, denotes the maximum number of prepared statements that is kept open per database connection.

`sql.store.read-only-media-store-names`

Type	java.util.List<java.lang.String>
Default	
Description	A list of names of read-only media stores. Such stores receive their binary data from another Content Server which also generates the blob ids that must then be used when creating blobs in this Content Server. Because the blob ids are provided externally, sql.store.generate-blob-ids=false is required if this property is non-empty. This makes it possible to share blobs between MLS and RLS. The blob collector will not remove any blobs from shared stores.

`sql.store.replace-substitute`

Type	java.lang.Boolean
Default	false
Description	There are problems with zOS DB2 databases to store characters that are not in the databases character set. Such characters are converted to the character with the code "0x001A" upon read. If the property is set to "true", this character will be replaced with the character defined in sql.store.substituteCharacter, thereby avoiding the replacement character that is illegal in XML texts.

`sql.store.sgml-cache-interval-seconds`

Type	java.lang.Integer
Default	0
Description	Seconds between cache statistics log entries to facility cap.server.store.sgmlcache

`sql.store.sgml-cache-size`

Type	org.springframework.util.unit.DataSize
Default	10MB
Description	Total size of XML objects cached in memory. Unit defaults to bytes.

`sql.store.substitute-character`

Type	java.lang.String
Default	?
Description	The character, with which "0x001A" should be replaced.

`sql.store.url`

Type	java.lang.String
Default	
Description	The URL of the database to connect to.

	Example: jdbc:oracle:thin@HostName:Port:CM Replace HostName and Port with the appropriate values of the database host. Don't replace HostName with "localhost", this may cause problems with some JDBC drivers.
`sql.store.use-ctes-for-below-queries`	
Type	java.lang.Boolean
Default	true
Description	Whether to use recursive common table expressions (aka "WITH RECURSIVE") in SQL statements for content queries with BELOW-clauses (aka descendantOf clauses) to find contents below some folder, if the folder index is not online.
`sql.store.user`	
Type	java.lang.String
Default	
Description	The user name at the database, which must match the schema.

Table 3.11. SQL Properties

3.2.5 Properties for Replicator Configuration

`replicator.auto-restart`	
Type	java.lang.Boolean
Default	true
Description	Restarts the Replicator sessions if lost (true).
`replicator.check-blob-hashes`	
Type	java.lang.Boolean
Default	true

Description	Whether the replicator checks that the hashes of uploaded blobs are the same as the hashes stored on the Master Live Server. A failed check indicates that the blob store configuration differs between the Master Live Server and the Replication Live Server. Defaults to true. Only set to false to force replication to resume in case of an emergency. The resulting inconsistent blobs ids may affect blob retrieval and the affected Replication Live Server should be recreated afterwards.

`replicator.check-stream`

Type	java.lang.Boolean
Default	true
Description	Defines if the Replication Live Server checks the event queue for connection (true). The event stream is not checked during the initial replication.

`replicator.check-timeout`

Type	java.lang.Long
Default	300
Description	The time in seconds after which the Replication Live Server checks the event queue for connection.

`replicator.chunking-threshold`

Type	java.lang.Integer
Default	10000
Description	Sets the maximum number of events that is fetched from the Master Live Server in one chunk during startup; 0 indicates no limit. Lowering the threshold will reduce main memory consumption at the cost of startup times.

`replicator.enable`

Type	java.lang.Boolean
Default	true
Description	Defines if the Replicator should be started on start of the Replication Live Server (true).

`replicator.force-online-switch`	
Type	java.lang.Boolean
Default	false
Description	Switch to runlevel online even if the repository contents have not yet been replicated from the Master Live Server. Ignored if not cap.server.init-runlevel=online.

`replicator.log-events`	
Type	java.lang.Boolean
Default	false
Description	All repository events will be logged (true).

`replicator.max-accepted-lag`	
Type	java.lang.Long
Default	100
Description	If the Replication Live Server is offline after a consistent replication but it is more than the given number of events behind the current timestamp, then the Replication Live Server will not go online until it catches up.

`replicator.packager-flush-size`	
Type	java.lang.Integer
Default	500
Description	Sets the maximum number of events that the packager holds in main memory before flushing to disk. Lowering the flush size will reduce main memory consumption at the cost of write performance.

`replicator.password`	
Type	java.lang.String
Default	replicator

Description	Configures the password to authenticate against the Master Live Server.

`replicator.publication-ior-url`

Type	java.lang.String
Default	
Description	Sets the location where the Replicator can find the IOR of the Master Live Server.

`replicator.restart-replicator-on-error`

Type	java.lang.Boolean
Default	true
Description	Restart the Replicator if an error occurs on Replication Live Server side (true). Otherwise, the Replicator will be stopped.

`replicator.shutdown-server-on-error`

Type	java.lang.Boolean
Default	false
Description	Stop the Replication Live Server if an error occurs on Replication Live Server side.

`replicator.startup-timeout`

Type	java.lang.Boolean
Default	false
Description	Defines if the Replication Live Server waits on start for the Replicator to connect to the Master Live Server.

`replicator.tmp-dir`

Type	java.lang.String
Default	

Description	The path to the folder where the Replication Live Server should write temporary data during replication. The path is relative to cap.server.base-dir of the replicator application, so you will probably use an absolute path. During initial replication of a complete repository, a huge amount of data will be written to this directory, in the order of the repository size. So, this directory should be located on a suitable file system with enough space left. During normal operation, only newly published blob content will be buffered on local disk.

`replicator.user`

Type	java.lang.String
Default	replicator
Description	Configures the user name to authenticate against the Master Live Server.

Table 3.12. Replicator Properties

3.2.6 Properties for Timezone and IOR

`cap.client.server.ior.url`

Value	URL format `http://<server>:<port> /ior`
Default	
Description	This property determines where to get the IOR of the *contentserver*. `<server>` must be the name of the *Content Server* host. For <port> you have to set the server's web server HTTP port.

`cap.client.timezone.default`

Value	TimeZone
Default	Europe/Berlin

Description	This parameter determines the used timezone. The standard value is *Europe/Berlin*. More time zones are documented in the class `java.util.TimeZone`.

Table 3.13. capclient.properties

3.2.7 Renamed Properties

Deprecated Name	New Name
`sql.store.sgml-cache-size-bytes`	`sql.store.sgml-cache-size`

Table 3.14. Renamed Content Server Properties

3.3 Headless Server Properties

3.3.1 Headless Server Spring Boot Properties

`caas.doctype.doc-type-collection`

Type	java.lang.String
Default	CMCollection
Description	The name of the doctype for Collections.

`caas.doctype.doc-type-linkable`

Type	java.lang.String
Default	CMLinkable
Description	The name of the doctype for Linkables.

`caas.doctype.doc-type-location-taxonomy`

Type	java.lang.String
Default	CMLocTaxonomy
Description	The name of the doctype for Location Taxonomies.

`caas.doctype.doc-type-navigation`

Type	java.lang.String
Default	CMNavigation
Description	The name of the doctype for Navigations.

`caas.doctype.doc-type-picture`

Type	java.lang.String
Default	CMPicture
Description	The name of the doctype for Pictures.

`caas.doctype.doc-type-taxonomy`

Type	java.lang.String
Default	CMTaxonomy
Description	The name of the doctype for Taxonomies.

`caas.graphql.expose-secrets`

Type	java.lang.Boolean
Default	false
Description	Expose encrypted values from settings in their encrypted version. Defaults for security reasons to false.

`caas.graphql.max-execution-timeout`

Type	java.time.Duration
Default	0
Description	Limits the allowed execution time for a query, set in milliseconds. 0 = no timeout.

`caas.graphql.max-query-complexity`

Type	java.lang.Integer
Default	0
Description	Limits the complexity of a graphql query if set to a value greater than 0. 0 = off.

`caas.graphql.max-query-depth`

Type	java.lang.Integer
Default	30
Description	Limits the depth of a graphql query if set to a value greater than 0. 0 = off.

`caas.graphql.repository-path-exclude-patterns`

Type	java.util.List<java.lang.String>
Default	[/Settings/Options/Settings/Internal/**, /Sites/**/Options/Settings/Internal/**, /Settings/Options/Settings/Feedback Hub, /Sites/**/Options/Settings/Feedback Hub, /Settings/Options/Settings/Content Hub, /Sites/**/Options/Settings/Content Hub, /Settings/Options/Settings/GlobalLink, /Settings/Options/Settings/Translation Services, /Sites/**/Options/Settings/Translation Services, /Settings/Options/Settings/Elastic Social Credentials]

Description	Repository paths excluded from retrieval via GraphQL endpoint. The paths may contain glob style expressions. Note, that the document name must be part of the path, if you do not use glob style expressions and want to protect a distinct document.
	Examples:
	`caas.graphql.repository-path-exclude-patterns[0]=/A/B/C/D`
	`caas.graphql.repository-path-exclude-patterns[1]=/A/*/C/*`
	`caas.graphql.repository-path-exclude-patterns[2]=/A/B/**`
	`caas.graphql.repository-path-exclude-patterns[3]=/A/**/C/*`

`caas.media.hash-property-names`

Type	java.util.List<java.lang.String>
Default	[width, height, widthRatio, heightRatio]
Description	List of global or site specific transformation option names defined in image transformation settings whose values are to be considered to compute image hashes.

`caas.search.cache.querylist-search-cache-for-seconds`

Type	java.lang.Integer
Default	-1

Description	Time to cache query list search results in seconds. Set it to "-1" to deactivate the query list cache.

`caas.search.cache.seconds`

Type	java.lang.Integer
Default	-1
Description	Time to cache search query results in seconds. Set it to "-1" to deactivate the search-query cache.

`caas.search.enabled`

Type	java.lang.Boolean
Default	true
Description	Enable graphql search extensions.

`caas.search.load-search-schema`

Type	java.lang.Boolean
Default	true
Description	Load the GraphQL search schema extensions.

`caas.search.max-search-limit`

Type	java.lang.Integer
Default	200
Description	Limit the count of search results hits. Defaults to 200.

`caas.search.solr.collection`

Type	java.lang.String
Default	preview

Description	Solr collection to use for CaaS search.

`caas.stax.context-trace-enabled`

Type	java.lang.Boolean
Default	false
Description	When 'true', wraps ContextHandlers and OutputHandlers while parsing in order to tracelog all parsing events. This is for debugging purposes only. Defaults to false. Additionally this requires the loglevel for 'com.coremedia.caas.richtext.stax.handler.output' to be set to trace level: logging.level.com.coremedia.caas.richtext.stax.handler.output = trace

`caas.stax.max-aliases-for-collections`

Type	java.lang.Integer
Default	50
Description	Configures the maximum number of allowed yaml aliases used by snakeyaml instances while parsing rich text configurations. Defaults to 50.

`caas.stax.suppress-root-tag`

Type	java.lang.Boolean
Default	true
Description	When 'false', the root tag in the string result of a rich text transformation is not suppressed. Defaults to true.

`caas.bypass-filter-predicates`

Type	null
Default	false
Description	If enabled, FilterPredicates my be ignored via request header 'X-Ignore-Filters'. If disabled, the header is ignored.

`caas.cache-specs`

Type	java.util.Map<java.lang.String,java.lang.String>
Default	
Description	[Caffeine Cache] configuration: Cache specs by cache name. Size based eviction: • maximumSize=[long] - eviction occurs when the configured size limit of the cache is exceeded • maximumWeight=[long] - The values are removed from the cache when the weight is exceeded Time based eviction: • expireAfterWrite = entry is expired after period is passed since the last write occurs • expireAfterAccess = entry is expired after period is passed since the last read or write occurs • refreshAfterWrite = refresh entries after a defined period automatically Durations are represented by an integer, followed by one of "d", "h", "m", or "s", representing days, hours, minutes, or seconds respectively.

`caas.cors.allow-credentials-for-url-pattern`

Type	java.util.Map<java.lang.String,java.lang.Boolean>
Default	
Description	Map of whether user credentials are supported, based on URL patterns. Example: `caas.cors.allow-credentials-for-url-pattern[/**]=true` See org.springframework.web.util.pattern.PathPattern> for the syntax of allowed path patterns.

`caas.cors.allowed-headers-for-url-pattern`

Type	java.util.Map<java.lang.String,java.util.List<java.lang.String>>
Default	

Description	Map of headers that a pre-flight request can list as allowed for use during an actual request, based on URL patterns. A header name is not required to be listed if it is one of: Cache-Control, Content-Language, Expires, Last-Modified or Pragma. Example: `caas.cors.allowed-headers-for-url-pattern[/**]=x-requested-with,x-csrf-token` See org.springframework.web.util.pattern.PathPattern for the syntax of allowed path patterns.

`caas.cors.allowed-methods-for-url-pattern`

Type	java.util.Map<java.lang.String,java.util.List<java.lang.String>>
Default	
Description	Map of HTTP methods to allow, based on URL patterns. Example: `caas.cors.allowed-methods-for-url-pattern[/**]=GET,POST,PUT` See org.springframework.web.util.pattern.PathPattern for the syntax of allowed path patterns.

`caas.cors.allowed-origins-for-url-pattern`

Type	java.util.Map<java.lang.String,java.util.List<java.lang.String>>
Default	
Description	Map of origins to allow, based on URL patterns. Example: `caas.cors.allowed-origins-for-url-pattern[/**]=https://domain1.com,https://domain2.com` See org.springframework.web.util.pattern.PathPattern for the syntax of allowed path patterns.

`caas.cors.disable-protection`	
Type	java.lang.Boolean
Default	false
Description	Disable CORS configuration completely. Disabling CORS results effectively in an 'allow all CORS policy'.

`caas.cors.exposed-headers-for-url-pattern`	
Type	java.util.Map<java.lang.String,java.util.List<java.lang.String>>
Default	
Description	Map of response headers other than simple headers (i.e. Cache-Control, Content-Language, Content-Type, Expires, Last-Modified or Pragma) that an actual response might have and can be exposed, based on URL patterns. Example: `caas.cors.exposed-headers-for-url-pattern[/**]=x-requested-with,x-csrf-token` See org.springframework.web.util.pattern.PathPattern for the syntax of allowed path patterns.

`caas.cors.fallback.allow-credentials`	
Type	java.lang.Boolean
Default	true
Description	Allow credentials flag to be used for path pattern '/**', in the case that no specific CORS configuration exists at all. @see org.springframework.web.cors.CorsConfiguration#setAllowCredentials(Boolean)

`caas.cors.fallback.allowed-methods`	
Type	java.util.List<java.lang.String>
Default	

Description	The fallback allowed methods to be used for path pattern '/**', in the case that no specific CORS configuration exists at all. @see org.springframework.web.cors.CorsConfiguration#setAllowedMethods(List)

`caas.cors.fallback.allowed-origin`

Type	java.lang.String
Default	
Description	A fallback allowed origin to be used for path pattern '/**', in the case that no specific CORS configuration exists at all. This may be used to easily configure a minimal CORS config consisting of the FQDN, which is usually set via environment variables. @see org.springframework.web.cors.CorsConfiguration#setAllowedOrigins(List)

`caas.cors.max-age-for-url-pattern`

Type	java.util.Map<java.lang.String,java.time.Duration>
Default	
Description	Map of how long, as a duration, the response from a pre-flight request can be cached by clients, based on URL patterns. Example: `caas.cors.max-age-for-url-pattern[/**]=3m` See org.springframework.web.util.pattern.PathPattern for the syntax of allowed path patterns.

`caas.expose-extensions`

Type	java.lang.Boolean
Default	false
Description	Set to true to expose extensions as described by the GraphQL spec.

`caas.forward-cookies`

Type	java.lang.Boolean
Default	false
Description	Enables http cookie forwarding of incoming requests on endpoint 'graphql' and for persisted queries on '/caas/v1/*' towards outgoing, secondary requests to 'graphql' (triggered by '/preview') and cae remote link resolving.

`caas.forward-header-names`

Type	java.util.List<java.lang.String>
Default	
Description	Comaseparated enumeration of http headernames to be forwarded for incoming requests on endpoint 'graphql' and for persisted queries on '/caas/v1/*' towards outgoing, secondary requests to 'graphql' (triggered by /preview) and cae remote link resolving.

`caas.log-requests`

Type	java.lang.Boolean
Default	false
Description	Enables logging for HTTP requests (excluding OPTIONS)

`caas.media-cache-time`

Type	java.time.Duration
Default	365d
Description	Maximum cache time to set in the com.coremedia.caas.web.controller.MediaController responses via the Cache-Control max-age header.

`caas.media-no-transform`

Type	java.lang.Boolean
Default	true

Description	Flag to set no-transform value for Cache-Control header in the Media Controller responses

`caas.preview`

Type	java.lang.Boolean
Default	false
Description	Enables preview mode

`caas.site-filter-config-properties.excluded-field-names`

Type	java.util.Collection<java.lang.String>
Default	[localizedVariant, localizedVariants, derivedSites]
Description	Fields to be excluded from site restrictions check. To disable all fields (thus prohibiting all objects not belonging to the filtered site), add the config property without any field names (=empty list). Adding additional fields requires to add the defaults as mentioned above to the config property! Defaults to "localizedVariant", "localizedVariants" and "derivedSites".

`caas.strict-transport-security-header.include-subdomains`

Type	java.lang.Boolean
Default	true
Description	If true, adds the 'includeSubdomains' flag to the Strict-Transport-Security HTTP response header.

`caas.strict-transport-security-header.max-age`

Type	java.lang.Integer
Default	63072000
Description	Max-Age of the Strict-Transport-Security HTTP response header in seconds, e.g. max-age=63072000.

caas.strict-transport-security-header.preload	
Type	java.lang.Boolean
Default	false
Description	If true, adds the 'preload' flag to the Strict-Transport-Security HTTP response header.

Table 3.15. Headless Server Properties

3.3.2 Persisted Query Properties

caas.persisted-queries.allow-list	
Type	java.lang.Boolean
Default	false
Description	Query allow listing. Set this to true to disallow any queries not loaded by some of the persisted queries pattern above.

caas.persisted-queries.apollo-query-map-resources-pattern	
Type	java.lang.String
Default	classpath*:graphql/queries/apollo*.json
Description	Resource pattern for persisted queries in Apollo Persisted Queries format

caas.persisted-queries.automatic	
Type	java.lang.Boolean
Default	true
Description	Enable Apollo Automatic Persisted Queries

caas.persisted-queries.exclude-file-name-pattern	
Type	java.lang.String

Default	.*Fragment(s)?.graphql
Description	Resource pattern for persisted queries in Relay Compiler format

`caas.persisted-queries.query-resources-pattern`

Type	java.lang.String
Default	classpath*:graphql/queries/*.graphql
Description	Resource pattern for persisted queries, one query per resource file. The filename w/o extension serves as the query id. See also: Spring PathMatchingResourcePatternResolver class.

`caas.persisted-queries.relay-query-map-resources-pattern`

Type	java.lang.String
Default	graphql/queries/relay*.json
Description	Resource pattern for persisted queries in Relay Compiler format

`caas.rest.jslt-enabled`

Type	java.lang.Boolean
Default	true
Description	En/disables the JSLT transformation processor on REST endpoints

`caas.rest.jslt-transformations-pattern`

Type	java.lang.String
Default	classpath*:transformations/*.jslt
Description	Returns the path pattern where to find the jslt transformation resources within headless.

`caas.rest.query-mapping-pattern`

Type	java.lang.String

Default	classpath*:graphql/rest-mapping/*.properties
Description	Returns the path pattern where to find rest mapping resources within headless.
`com.coremedia.caas.web.persistedqueries.impl.PersistedQueryAutoConfiguration`	
Type	java.util.List
Default	
Description	To disable the feature 'persisted-queries' add the class 'com.coremedia.caas.web.persistedqueries.impl.PersistedQueryAutoConfiguration' to 'spring.autoconfigure.exclude'.
`com.coremedia.caas.web.rest.RestMappingAutoConfiguration`	
Type	java.util.List
Default	
Description	To disable the feature 'persisted-query-rest-mapping' add the class 'com.coremedia.caas.web.rest.RestMappingAutoConfiguration' to 'spring.autoconfigure.exclude'.

Table 3.16. Persisted Query Properties

3.3.3 Metadata Properties

`caas.metadata.property-mapping-default-filename`	
Type	java.lang.String
Default	propertyMapping.json
Description	Name of the default property mapping definition file
`caas.metadata.property-mapping-location`	
Type	java.lang.String
Default	classpath*:graphql/metadata/propertyMapping*.json

Description	Location of the property mapping file(s)

`com.coremedia.caas.web.metadata.MetadataAutoConfiguration`

Type	java.util.List
Default	
Description	To disable the feature 'metadata-query-root' add the class 'com.coremedia.caas.web.metadata.MetadataAutoConfiguration' to 'spring.autoconfigure.exclude'.

Table 3.17. Metadata Root Properties

3.3.4 Remote Service Adapter Properties

`caas.remote.base-url`

Type	java.lang.String
Default	http://localhost:8080/webapp/servlet/service-endpoint
Description	Base URL to the remote service handler

Table 3.18. Headless Server Remote Properties

3.3.5 Headless Server Cache Control Properties

`caas.cache-control.for-url-pattern`

Type	java.util.Map<java.lang.String,org.springframework.boot.autoconfigure.web.WebProperties$Resources$Cache$Cachecontrol>
Default	
Description	Map of cache control configuration objects for HTTP Cache-Control header. The configuration is URL pattern specific. Patterns are matched using

```
org.springframework.util.AntPathMatcher
```

. For every URL the content validity dates are taken into account to compute the max-age value. A negative configured max-age value disables cache control for the URL pattern. A configured positive max-age value is taken as maximum value for the URL pattern. Thus, if X is configured and Y is computed, then min(X,Y) is returned as max-age.

Note that max-age values must not be higher than 31536000 (365 days in seconds).

Also note that, by consequence, a response including properties of a content that is valid for more than a year according to its content settings, and for whose URL pattern no max-age value is defined, will include a max-age value of a year.

caas.cache-control.s-max-age-factor

Type	java.lang.Double
Default	0
Description	This value is used to compute a s-maxage header in relation to a recognized max-age value. If the value is 0.0 then the s-maxage value is not automatically adjusted. In particular this means the s-maxage value isn't written in the Cache-Control header unless it was otherwise configured, for example with: `caas.cache-control.for-url-pattern[/example/**].s-max-age=1m` But please note, a value other than 0.0 takes precedence over a static configured value. Then s-maxage takes the value of max-age * sMaxAgeFactor.

caas.cache-control.url-path.remove-semicolon-content

Type	java.lang.Boolean
Default	true
Description	Set if everything after a ";" (semicolon) should be cut from the request URI for path matching. E.g. to see all matrix parameters of fragment urls this should be set to "false". Default is "true".

caas.cache-control.url-path.url-decode

Type	java.lang.Boolean

Default	true
Description	Set if the request URI should be decoded for path matching. Default is "true".

`caas.cache-control.url-path.use-always-full-path`

Type	java.lang.Boolean
Default	false
Description	Set if the full path within the current web application context is used for path matching. By default, this is set to "false" which means that the servlet path is cut off.

Table 3.19. Headless Server Cache Control Properties

3.3.6 Headless Server Cache Key Properties

`cache.capacities.com.coremedia.caas.search.solr.SolrQueryCacheKey`

Type	java.lang.Long
Default	5000
Description	Sets the cache size for the solr query cache.

Table 3.20. Headless Server Cache Key Properties

3.3.7 Properties of External Frameworks

Table 3.21. Headless Server External Framework Properties

3.3.8 Renamed Properties

Deprecated Name	New Name
`caas.graphql.introspectionEnabled`	`spring.graphql.schema.introspection.enabled`
`caas.graphql.max-search-limit`	`caas.search.max-search-limit`
`caas.querylist-search-cache-for-seconds`	`caas.search.cache.querylist-search-cache-for-seconds`
`caas.solr.collection`	`caas.search.solr.collection`
`caas.graphql-restmapping-controller.enabled`	`spring.autoconfigure.exclude`

Table 3.22. Renamed Headless Server Properties

3.4 Studio Properties

3.4.1 Studio Configuration

The following list contains configuration properties for the *Studio*.

`studio.auto-logout.delay`

Type	java.time.Duration
Default	1800s
Description	The delay before a Studio session is automatically disconnected after inactivity. Inactivity is checked on the client. Set to 0 to disable auto logout (not recommended). For clients that cannot reach the Studio web application, the timeout of the server session determines the timeout of the Studio session.

`studio.bulk-operation-thread-pool.max-threads`

Type	java.lang.Integer
Default	10
Description	Maximum number of threads to use for the bulk operation thread pool.

`studio.cloud.enable`

Type	java.lang.Boolean
Default	false
Description	Whether to enable cloud support.

`studio.dashboard-refresh-interval`

Type	java.time.Duration
Default	30000ms

Description	The interval at which the dashboard is refreshed when visible (0 to disable automatic refresh).

`studio.default-picture.content.paths`

Type	java.util.Map<java.lang.String,java.lang.String>
Default	
Description	Configure {@link com.coremedia.cap.content.Content} property paths that point to picture blobs. Paths are given per {@link com.coremedia.cap.content.ContentType}, e.g. 'studio.default-picture.content.paths.CMPicture=data'. The mechanism is recursive: it is possible to configure a path to other {@link com.coremedia.cap.content.Content} items, e.g. 'studio.default-picture.content.paths.CMTeasable=pictures'.

`studio.default-picture.resolving.enabled`

Type	java.lang.Boolean
Default	true
Description	Configure whether server-side default picture resolving is enabled.

`studio.default-time-zone`

Type	java.lang.String
Default	Europe/Berlin
Description	Default time zone in CoreMedia Studio. Make sure that the default time zone is included in the studio.timeZones list. Defaults to 'Europe/Berlin'.

`studio.differencing.max-markup-size`

Type	org.springframework.util.unit.DataSize
Default	300000B
Description	The maximum size of a markup object (in bytes as default unit) for which differences with other markup can be computed. By default, this value is set to 300,000 bytes, which amounts to approximately 15,000 words.

`studio.expose-license-info`	
Type	java.lang.Boolean
Default	false
Description	Whether to expose license information along with the system information.

`studio.jobs.max-threads`	
Type	java.lang.Integer
Default	10
Description	Maximum number of threads to use for the jobs service thread pool.

`studio.locales`	
Type	java.util.List<java.lang.String>
Default	en,de
Description	A comma-separated list of locales from which the user can choose the locale in which the Studio is displayed. If unset, the list locales defaults to English or German.

`studio.login-url`	
Type	java.lang.String
Default	
Description	If non-empty, a URL to which the Studio client should redirect a user who is not yet logged in. This supports single-sign-on solutions using an external login page. If empty, Studio shows a login form itself.

`studio.preview-controller-pattern`	
Type	java.lang.String
Default	preview?id={0}

Description	Get the preview controller pattern. If it is empty or not defined, then use the default preview controller pattern.

`studio.preview-url-prefix`

Type	java.lang.String
Default	
Description	The URl prefix of the CAE preview web app. This prefix is used to compose preview URIs for contents, assuming that a standard CAE preview controller is mapped at the path 'preview'. Since studio and cae are usually deployed as independent spring boot application or as a containerized application (docker/kubernetes), it is necessary to provide an absolute URL to load the preview, like 'https://your-host-name.com:40980/blueprint/servlet'

`studio.preview-url-whitelist`

Type	java.util.List<java.lang.String>
Default	
Description	Configures a list of valid preview origins. The preview integration does only work for contents from listed origins. Wildcards (*) may be used for valid origin entries. If left blank (or if the property is not listed at all), the origin of studio.previewUrlPrefix is the only accepted origin. The URLs of the preview origins must not contain a trailing slash. NOTE: Once a whitelist is configured, the Studio has no chance to set a target origin in outgoing messages anymore. Be aware that this is a minor security drawback. EXAMPLE: studio.previewUrlWhitelist=https://host1:port1,\ https://host2:port2,\ http://localhost.*,\ *.company.com Enabling Elastic Social tenants in the embedded preview requires including a placeholder CoreMedia Studio then replaces the token with the current tenant. In a Blueprint related project, this could be: studio.previewUrlPrefix=http://{0}.localhost:40081/blueprint/servlet

`studio.publication-priority`

Type	java.lang.Integer
Default	60
Description	The priority of publications started in Studio. For an overview of relevant priorities take a look at "com.coremedia.cap.content.publication.Publication".

`studio.rest.changeset-max-iterations`	
Type	java.lang.Integer
Default	10
Description	Maximum number of iterations to complete a change set.

`studio.rest.invalidation-source.caplist.capacity`	
Type	java.lang.Integer
Default	10000
Description	The capacity of the invalidation source.

`studio.rest.invalidation-source.content.capacity`	
Type	java.lang.Integer
Default	10000
Description	The capacity of the invalidation source.

`studio.rest.invalidation-source.model.capacity`	
Type	java.lang.Integer
Default	10000
Description	The capacity of the invalidation source.

`studio.rest.invalidation-source.notification.capacity`	
Type	java.lang.Integer
Default	10000
Description	The capacity of the invalidation source.

`studio.rest.invalidation-source.project.capacity`

Type	java.lang.Integer
Default	10000
Description	The capacity of the invalidation source.

`studio.rest.invalidation-source.system.capacity`

Type	java.lang.Integer
Default	10000
Description	The capacity of the invalidation source.

`studio.rest.invalidation-source.timeout`

Type	java.time.Duration
Default	20000ms
Description	The timeout to wait for new events to be propagated via the invalidation sources.

`studio.rest.invalidation-source.todo.capacity`

Type	java.lang.Integer
Default	10000
Description	The capacity of the invalidation source.

`studio.rest.invalidation-source.workflow.capacity`

Type	java.lang.Integer
Default	10000
Description	The capacity of the invalidation source.

`studio.rest.network.max-requests`

Type	java.lang.Integer

Default	15
Description	The maximum amount of parallel pending requests executed to one host.

`studio.rest.search-service.default-search-limit`

Type	java.lang.Integer
Default	5000
Description	The default search limit;

`studio.rest.search-service.max-search-limit`

Type	java.lang.Integer
Default	80000
Description	The maximum search limit

`studio.rest.search-service.replace-special-characters-in-prefix-or-wildcard-query`

Type	java.lang.Boolean
Default	true
Description	Whether special characters in prefix and wildcard queries (queries ending with * or containing *) are replaced by spaces. Replacing characters makes sense because Solr does not analyze prefix and wildcard queries in the same way as it indexes text when configured with the StandardTokenizer. By replacing these characters, indexed text can still be matched and queries return the expected hits. However, it can make sense to disable this setting, if Solr is configured to preserve special characters in indexed text.

`studio.struct-as-blob-property-names`

Type	java.util.List<java.lang.String>
Default	
Description	List of struct property names which are to be handled as struct properties by the client.

`studio.supported-locales-variable-name`	
Type	java.lang.String
Default	joo.localization.supportedLocales
Description	The variable name of the supported locales javascript snippet as expected by the client. Defaults to 'joo.localization.supportedLocales'.

`studio.time-zones`	
Type	java.util.List<java.lang.String>
Default	Europe/Berlin
Description	Supported time zones in CoreMedia Studio. Provide a comma separated list of supported Java time zone IDs. Defaults to 'Europe/Berlin'.

`studio.translation.max-dependent-content-iterations`	
Type	java.lang.Integer
Default	100
Description	Defines how deep links should be followed, when calculating dependent content. Can be overridden via content setting.

`studio.translation.max-dependent-contents`	
Type	java.lang.Integer
Default	100
Description	Defines how many dependent contents will be returned in "Translation Workflow Start window" when starting a translation workflow. Can be overridden via content setting.

`studio.translation.show-start-window`	
Type	java.lang.Boolean
Default	true

Description	Configures Pull Translation (translation into preferred site) behavior: When set to true the workflow dialog is displayed, otherwise the workflow is started automatically.

`studio.upload.default-blob-property`

Type	java.lang.String
Default	data
Description	The default blob property name of the default content to be used for bulk uploads, used if no mapping is found in the upload settings defined in content. Defaults to 'data'.

`studio.upload.default-content-type`

Type	java.lang.String
Default	CMDownload
Description	The default content type to be used for bulk uploads, used if no mapping is found in the upload settings defined in content. Defaults to 'CMDownload'.

`studio.upload.upload-settings-struct-property`

Type	java.lang.String
Default	settings
Description	The name of the struct property to be used when the UploadSettings document is read. Defaults to 'settings'.

`studio.validate-before`

Type	com.coremedia.rest.cap.config.ValidatedActionLevel
Default	
Description	Defines which content actions are not allowed if the content has error issues. As some actions are in a dependency relationship (e.g. APPROVE depends on CHECKIN), only the least inclusive actions need to be declared (e.g. CHECKIN entails the check of CHECKIN and APPROVE actions). Currently, the only supported OPTIONS are CHECKIN, APPROVE or nothing for no restriction. Defaults to APPROVE, i.e. content approval and publication is not allowed in case of error issues.

`studio.workflow.validation.start-validators.simple-publication`

Type	java.util.List<java.lang.String>
Default	publicationContentRightsWorkflowValidator,publicationNoAssigneeValidator,publicationSessionUserRightsWorkflowValidator,publicationWorkflowUndoWithdrawValidator,publicationContentIssuesValidator
Description	List of bean names to use in simple publication workflow start validation. The bean names must map to `com.coremedia.rest.cap.workflow.validation.WorkflowValidator` beans. Validators will be executed in the order given in the list. The list must not be empty.

`studio.workflow.validation.start-validators.synchronization`

Type	java.util.List<java.lang.String>
Default	emptyParametersValidator,derivedSyncSitesWritableValidator,contentSitesValidator,contentStateValidatorForTranslationWFStart,contentHasMasterAndLocalePropertyValidator,masterSiteManagedByCurrentUserValidator
Description	List of bean names to use in synchronization workflow start validation. The bean names must map to `com.coremedia.rest.cap.workflow.validation.WorkflowValidator` beans. Validators will be executed in the order given in the list. The list must not be empty.

`studio.workflow.validation.start-validators.translation`

Type	java.util.List<java.lang.String>
Default	emptyParametersValidator,sitesManagedByCurrentUserValidator,contentSitesValidator,contentStateValidatorForTranslationWFStart,contentHasMasterAndLocalePropertyValidator,contentAlreadyInTranslationValidator
Description	List of bean names to use in translation workflow start validation. The bean names must map to

	`com.coremedia.rest.cap.workflow.validation.WorkflowValidator` beans. Validators will be executed in the order given in the list. The list must not be empty.
`studio.workflow.validation.start-validators.two-step-publication`	
Type	java.util.List<java.lang.String>
Default	publicationContentRightsWorkflowValidator,publicationNoAssigneeValidator,publicationSessionUserRightsWorkflowValidator,publicationWorkflowUndoWithdrawValidator,publicationContentIssuesValidator
Description	List of bean names to use in two-step publication workflow start validation. The bean names must map to `com.coremedia.rest.cap.workflow.validation.WorkflowValidator` beans. Validators will be executed in the order given in the list. The list must not be empty.
`studio.xml-filters.enable-multi-class-span-filter`	
Type	java.lang.Boolean
Default	false
Description	Enables a filter for CoreMedia RichText 1.0, which splits up spans with multiple classes to dedicated spans, each with a single class only. This filter should be enabled for CKEditor 4 and should be disabled for all later versions of CKEditor or alternative rich text editing component frameworks.

Table 3.23. Studio Properties

3.4.2 Available Locales Configuration

The following list contains configuration properties for the available locales in *Studio*.

`available-locales.content-path`

Type	java.lang.String
Default	/Settings/Options/Settings/LocaleSettings
Description	Path of the content that defines the available locales. Defaults to '/Settings/Options/Settings/LocaleSettings'.

`available-locales.property-path`

Type	java.lang.String
Default	settings.availableLocales
Description	Property path to the Struct-StringListProperty containing the locales. Defaults to 'settings.availableLocales'.

Table 3.24. Available Locales Properties

3.4.3 Content Configuration

The following list contains configuration properties for content repository paths with special meaning in *Studio*.

`content.global-configuration-path`

Type	java.lang.String
Default	/Settings
Description	A global folder where additional settings are read from. Several Studio plugins lookup settings content from here. Defaults to '/Settings.

`content.site-configuration-path`

Type	java.lang.String
Default	Options/Settings

Description	A local folder where additional settings are read from. Several Studio plugins lookup settings content from here. Defaults to 'Options/Settings.

Table 3.25. Content Properties

3.4.4 Navigation Validator Configuration

The following list contains configuration properties for validating the navigation structure in *Studio*.

`validators.navigation.ignore-path`

Type	java.util.List<java.lang.String>
Default	[/Settings, /Home, /System]
Description	If a content path matches one of these values or is a subfolder of it, the corresponding validator won't continue it's validation. Wildcard annotations are allowed here, e.g. '/Home/*' matches '/Home/Adam' or '/Sites/*/*/Navigation/*' matches '/Sites/Chef Corp./English/Navigation/Content ABC'.

Table 3.26. Navigation Validators Properties

3.4.5 Preview URL Service Properties

The following list contains configuration properties related to the Multi Preview Menu in *Studio*.

`preview.urlservice.content-type`

Type	java.lang.String
Default	CMSettings
Description	Defines the content type that is used to configure previews.

`preview.urlservice.global-path`

Type	java.lang.String

Default	/Settings/Options/Settings/Multi Preview
Description	Defines the repository folder path with contents that configure previews for all sites and contents without a site. Subfolders are ignored.
`preview.urlservice.headless-preview-host`	
Type	java.lang.String
Default	
Description	Defined the host of the headless preview server.
`preview.urlservice.preview-url-allow-list`	
Type	java.util.List<java.lang.String>
Default	
Description	Defines the commercePreviewUrlAllowList. In order to limit the urls in the preview frame and prevent CSRF.
`preview.urlservice.property`	
Type	java.lang.String
Default	settings
Description	Defines the name of the struct property in the configured content type.
`preview.urlservice.site-path`	
Type	java.lang.String
Default	Options/Settings/Multi Preview
Description	Defines the repository folder path below a site root folder with contents that configure previews for one site. Subfolders are ignored.

Table 3.27. Preview URL Service Properties

3.4.6 Content Security Policy Configuration

The following list contains configuration properties related to Content Security Policy (CSP) in the *Studio*.

`studio.security.csp.child-src`

Type	java.util.List<java.lang.String>
Default	
Description	List of values for the 'child-src' policy directive. This directive is only applied if extended by plugins or extensions.

`studio.security.csp.connect-src`

Type	java.util.List<java.lang.String>
Default	
Description	List of values for the 'connect-src' policy directive. Defaults to 'self'.

`studio.security.csp.font-src`

Type	java.util.List<java.lang.String>
Default	
Description	List of values for the 'font-src' policy directive. Defaults to 'self'.

`studio.security.csp.frame-src`

Type	java.util.List<java.lang.String>
Default	
Description	List of values for the 'frame-src' policy directive. The hierarchy of default values for this directive is as follows 1. studio.previewUrlWhitelist values if specified 2. schema and authority of studio.previewUrlPrefix if specified

3. 'self'

To allow YouTube videos inside the external preview, add the Youtube URL: studio.security.csp.frameSrc=http://localhost:40980,*.coremedia.vm:40980, *.coremedia.vm,*.coremedia.com,*.coremedia.com:8000,*.coremedia.vm:8000, 'self',www.youtube.com

`studio.security.csp.img-src`

Type	java.util.List<java.lang.String>
Default	
Description	List of values for the 'img-src' policy directive. Defaults to 'self'.

`studio.security.csp.manifest-src`

Type	java.util.List<java.lang.String>
Default	
Description	List of values for the 'manifest-src' policy directive. Defaults to 'self'.

`studio.security.csp.media-src`

Type	java.util.List<java.lang.String>
Default	
Description	List of values for the 'media-src' policy directive. Defaults to 'self'.

`studio.security.csp.object-src`

Type	java.util.List<java.lang.String>
Default	
Description	List of values for the 'object-src' policy directive. Defaults to 'self'.

`studio.security.csp.report-uri`

Type	java.util.List<java.lang.String>

Default	
Description	List of values for the 'report-uri' policy directive. If no custom list is provided the directive is not included.
`studio.security.csp.script-src`	
Type	java.util.List<java.lang.String>
Default	
Description	List of values for the 'script-src' policy directive. Defaults to 'self','unsafe-eval'.
`studio.security.csp.style-src`	
Type	java.util.List<java.lang.String>
Default	
Description	List of values for the 'style-src' policy directive. Defaults to 'self','unsafe-inline'.
`studio.security.csp.frame-ancestors`	
Type	java.util.List<java.lang.String>
Default	
Description	List of values for the 'frame-ancestors' policy directive. Defaults to 'self'. @deprecated Configuring this setting does not have an effect anymore. Please configure this directive in deployment.

Table 3.28. Content Security Policy Related Studio Properties

3.4.7 Content Hub Configuration

The following table lists the configuration properties for the *CoreMedia Content Hub*.

`contenthub.studio.global-configuration-path`	
Type	java.lang.String

Default	/Settings/Options/Settings/Content Hub
Description	Defines the global configuration path. The Content Hub will look up 'Connection' settings document in this folders. The default value is /Settings/Options/Settings/Content Hub.
`contenthub.studio.site-configuration-path`	
Type	java.lang.String
Default	/Options/Settings/Content Hub
Description	Defines the site specific configuration path. If a Content Hub connection should only be available sites, the 'Connection' settings document can be but in this subfolder of a site. The default value is /Options/Settings.

Table 3.29. Content Hub Properties

3.4.8 Feedback Hub Configuration

The following table lists the configuration properties for the *CoreMedia Feedback Hub*.

`feedbackhub.bindings.content-type`	
Type	java.lang.String
Default	CMSettings
Description	Defines the content type that is used to configure Feedback Hub bindings.
`feedbackhub.bindings.global-path`	
Type	java.lang.String
Default	/Settings/Options/Settings/Feedback Hub
Description	Defines the repository folder path with contents that configure Feedback Hub bindings for all sites and contents without a site. Subfolders are ignored.
`feedbackhub.bindings.property`	

Type	java.lang.String
Default	settings
Description	Defines the name of the struct property in the configured content type with the Feedback Hub binding configuration.

`feedbackhub.bindings.site-path`

Type	java.lang.String
Default	Options/Settings/Feedback Hub
Description	Defines the repository folder path below a site root folder with contents that configure Feedback Hub bindings for one site. Subfolders are ignored.

Table 3.30. Feedback Hub Properties

3.4.9 Editorial Comments Configuration

The following table lists the configuration properties for the *CoreMedia Editorial Comments* feature, which establishes a connection to the relational database.

The most important property is `editorial.comments.datasource.url` which sets the URL to connect to.

In case another *schema/username/password* has to be configured use the properties `editorial.comments.db.schema/editorial.comments.db.username/editorial.comments.db.password`.

`editorial.comments.datasource.driver-class-name`

Type	java.lang.String
Default	
Description	Required to be set according to your database. See official spring documentation 'spring.datasource.driver-class-name' for detailed information

`editorial.comments.datasource.hikari.connection-timeout`

Type	java.lang.Integer

Default	20000ms
Description	Value must not be greater then the studio request timeout. See official spring documentation 'spring.datasource.hikari.connection-timeout' for more information

`editorial.comments.datasource.password`

Type	java.lang.String
Default	cm_editorial_comments
Description	See official spring documentation 'spring.datasource.password' for more information

`editorial.comments.datasource.url`

Type	java.lang.String
Default	
Description	Required to be set according to your database. See official spring documentation 'spring.datasource.url' for detailed information

`editorial.comments.datasource.username`

Type	java.lang.String
Default	cm_editorial_comments
Description	See official spring documentation 'spring.datasource.username' for more information

`editorial.comments.db.password`

Type	java.lang.String
Default	cm_editorial_comments
Description	Use to set the password for hibernate and liquibase.

`editorial.comments.db.schema`

Type	java.lang.String

Default	cm_editorial_comments
Description	Use to set the schema for hibernate and liquibase.

`editorial.comments.db.username`

Type	java.lang.String
Default	cm_editorial_comments
Description	Use to set the username for hibernate and liquibase.

`editorial.comments.jpa.database-platform`

Type	java.lang.String
Default	
Description	See official spring documentation 'spring.jpa.properties.hibernate.database-platform' for detailed information

`editorial.comments.jpa.properties.hibernate.default_schema`

Type	java.lang.String
Default	cm_editorial_comments
Description	See official spring documentation 'spring.jpa.properties.hibernate.default_schema' for more information

`editorial.comments.liquibase.change-log`

Type	java.lang.String
Default	classpath:db/changelog/db.changelog-editorial-comments.xml
Description	See official liquibase documentation 'liquibase.change-log' for more information

`editorial.comments.liquibase.default-schema`

Type	java.lang.String

Default	cm_editorial_comments
Description	See official liquibase documentation 'liquibase.default-schema' for more information

`editorial.comments.liquibase.enabled`

Type	java.lang.Boolean
Default	true
Description	Use this property to disable liquibase, however this means that you need to apply the changesets to the database manually. You can either do that by activating liquibase for at least one startup of a Studio-Server, or run liquibase manually (https://docs.liquibase.com/tools-integrations/cli/home.html) after an upgrade.

`editorial.comments.liquibase.password`

Type	java.lang.String
Default	cm_editorial_comments
Description	See official liquibase documentation 'liquibase.password' for more information

`editorial.comments.liquibase.user`

Type	java.lang.String
Default	cm_editorial_comments
Description	See official liquibase documentation 'liquibase.user' for more information

`editorial.comments.notification-strategies.created-comment-on-content`

Type	java.lang.Boolean
Default	true
Description	This property enables the CreatedCommentOnContentCollectUsersToNotifyStrategy which collects all users who wrote a comment at this content in the last 30 days. True if the strategy is enabled, false if the strategy is disabled.

`editorial.comments.notification-strategies.edited-content-in-last-thirty-days`	
Type	java.lang.Boolean
Default	true
Description	This property enables the EditedContentInLastThirtyDaysCollectUsersToNotifyStrategy which collects all users who edited the given content in last 30 days. True if the strategy is enabled, false if the strategy is disabled.
`editorial.comments.notification-strategies.my-edited-contents`	
Type	java.lang.Boolean
Default	true
Description	This property enables the MyEditedContentsCollectUsersToNotifyStrategy, which notifies every user that has the Content where an EditorialComment was created for in his "myEditedContent" List. true if the strategy is enabled, false if the strategy is disabled.

Table 3.31. Editorial Comments Properties

NOTE

If the startup of a Studio-Server instance has been interrupted, it is possible that a lock is left by Liquibase on the database schema cm_editorial_comments. In this case it is necessary to remove the lock manually, as described at https://docs.liquibase.com/concepts/basic/databasechangeloglock-table.html. It is also possible to disable Liquibase with the configuration `editorial.comments.liquibase.enabled` (for further details, refer to Section 3.4, “Editorial Comments Database Configuration” in *Studio Developer Manual*).

3.4.10 Commerce Related Configuration

The following table lists the commerce related *CoreMedia Studio* properties.

`studio.commerce.preload-child-categories`	
Type	java.lang.String

Default	ALL
Description	The default behavior of the Studio library catalog tree is to load the next level of categories no matter if they are displayed. This is done to determine if a child category is virtual or not. All occurrences of a category that are not in the primary location in the catalog tree are considered as virtual. Set this property to *`ALL_EXCEPT_TOP_LEVEL`* if top level categories should be excluded from pre-loading. It can be useful if there is a huge number of top level categories and if you are sure they are not virtual. In commerce systems where no physical root category exists it must be this way. Moreover, if you are sure there is no virtual category at all you can use the value *`NONE`*. If a child category is not pre-loaded, its state is assumed to be non-virtual.

Table 3.32. Commerce Related Properties

3.5 User Changes Properties

The following list contains configuration properties related to the user changes app.

`userchanges.excluded-paths`

Type	java.util.List<java.lang.String>
Default	[/Home/*/EditorPreferences, /Home/*/My Preferences, /Home/*/My Dictionary, /System/Public Dictionary, /Home/*/EditorProfile]
Description	Paths to be excluded from tracking of user changes.

`userchanges.excluded-user-names`

Type	java.util.List<java.lang.String>
Default	[admin, publisher, workflow, importer, translation-workflow-robot]
Description	Users to be excluded from tracking of user changes.

`userchanges.listener.enabled`

Type	java.lang.Boolean
Default	false
Description	Set to 'true' to let the application register the user changes listener for content events.

`userchanges.max-length`

Type	java.lang.Integer
Default	1000
Description	The maximum number of contents in a user's "My Edited Content" list. The User Changes App will not add further contents to a list, if the number of contents has reached this maximum. It is recommended to configure a maximum that can still be handled by editors, and is a lot lower than the maximum number of contents that can be stored in a MongoDB document. The latter depends on the length of stored document IDs but can be estimated to something around 600.000 contents.

`userchanges.notifications.workflow.enabled`	
Type	java.lang.Boolean
Default	false
Description	Set to 'true' to let the application generate notifications for workflow events.

Table 3.33. User Changes App Properties

3.6 Workflow Server Properties

`workflow.blob-cache-size`	
Type	org.springframework.util.unit.DataSize
Default	32MB
Description	The size of the main disk cache of the UAPI
`workflow.blob-streaming-size-threshold`	
Type	org.springframework.util.unit.DataSize
Default	128KB
Description	The minimum size of streamed blobs
`workflow.blob-streaming-threads`	
Type	java.lang.Integer
Default	2
Description	The number of streaming threads
`workflow.domain`	
Type	java.lang.String
Default	
Description	The server's domain name
`workflow.heap-cache-size`	
Type	org.springframework.util.unit.DataSize
Default	20MB

Description	The size of the main memory cache of the UAPI

`workflow.ior-url`

Type	java.lang.String
Default	
Description	The *Content Server*'s IOR URL

`workflow.map-role`

Type	java.util.Map<java.lang.String,java.lang.String>
Default	
Description	Role mappings are used to replace group names in a process definition file by other group names without having to actually modify the textual definition

`workflow.max-cached-blob-size`

Type	org.springframework.util.unit.DataSize
Default	
Description	The maximum size of cached blobs

`workflow.navigatethrough`

Type	java.lang.Boolean
Default	true
Description	Whether to enable navigate through cache of the UAPI. This cache is used for computing read rights on folders in case no explicit read rights are defined and read rights on subfolders force read rights on folders on the path to these folders. Because these computations are expensive and sometimes unnecessary, this flag allows you to ignore navigate-through rights in the Workflow Server.

`workflow.password`

Type	java.lang.String

Default	workflow
Description	Define a password for the connection with the *Content Management Server*.
`workflow.pool.queue-size`	
Type	java.lang.Integer
Default	0
Description	Pool queue limit.
`workflow.pool.threads-max`	
Type	java.lang.Integer
Default	
Description	Pool thread max limit.
`workflow.pool.threads-min`	
Type	java.lang.Integer
Default	0
Description	Pool thread min limit.
`workflow.server.adapters.directory`	
Type	java.lang.String
Default	com.coremedia.workflow.impl.server.adapters.directory.UnifiedUserManagerAdapter
Description	The directory service adapter.
`workflow.server.adapters.persistence`	
Type	java.lang.String
Default	com.coremedia.workflow.impl.server.adapters.persistence.GenericRDBMSAdapter

Description	The persistence adapter.

`workflow.server.allow-live`

Type	java.lang.Boolean
Default	false
Description	Whether to allow connecting to a live *Content Server*

`workflow.server.archive.persistence`

Type	java.lang.String
Default	elastic-core
Description	Persistence option where archived workflow processes are stored. By default, Elastic Core is used for storage in MongoDB or in its in-memory replacement. Set this to "sql" to store archived processes in the Workflow Server's relational database.

`workflow.server.archive.retry-exceptions`

Type	java.util.List<java.lang.String>
Default	[java.io.IOException, com.mongodb.MongoTimeoutException]
Description	List of exception classes that indicate temporary problems when archiving a workflow process. If one of these exceptions is thrown directly or as cause of another exception in com.coremedia.workflow.common.actions.ArchiveProcessFinalAction, then the Workflow Server will retry archiving with some delay as configured in properties workflow.server.retry-final-actions.*.

`workflow.server.enable-workflow-converter`

Type	java.lang.Boolean
Default	false
Description	This flag allows you to enable the workflow converter while starting Workflow Server

`workflow.server.final-actions-max`

Type	java.lang.Integer
Default	10
Description	The maximum number of threads executing FinalActions of finished processes. This setting can be changed at runtime with the JMX attribute 'FinalActionsMax'.

`workflow.server.remote-action-handler`

Type	java.lang.String
Default	com.coremedia.cotopaxi.workflow.BuiltInRemoteActionHandler
Description	The remote action handler.

`workflow.server.retry-final-actions.delay.initial`

Type	java.time.Duration
Default	1s
Description	The initial time interval to wait before retrying the execution of a workflow final action after it failed for the first time. The value must be >= 1 millisecond. The initial interval is multiplied with the configured multiplier for each retry attempt (exponential back-off) until the configured maximum has been reached. This setting is ignored if the exception provides a different value in method RetryableActionException#getDelayBeforeRetry.

`workflow.server.retry-final-actions.delay.max`

Type	java.time.Duration
Default	1h
Description	The maximum time to wait before retrying a failed execution of a workflow final action. The duration must be larger or equal to the specified initial time to wait. This setting is ignored if the exception provides a different value in method RetryableActionException#getDelayBeforeRetry.

`workflow.server.retry-final-actions.delay.multiplier`

Type	java.math.BigDecimal

Default	2
Description	Multiplier for exponential back-off. The value must be >= 1.0. This setting is ignored if the exception provides a different value in method RetryableActionException#getDelay-BeforeRetry.

`workflow.server.retry-final-actions.delay.random-factor`

Type	java.math.BigDecimal
Default	0.1
Description	Factor to randomize the effective delay. The value must be >= 0.0 and < 1.0. Use 0.0 for no randomization. For example, a value of 0.1 creates effective delays in the range of plus/minus 10 percent of the computed delay. This also means that the actual delay can be up to 10 percent above the configured max value. This setting is ignored if the exception provides a different value in method RetryableActionException#getDelayBe-foreRetry.

`workflow.server.retry-final-actions.enabled`

Type	java.lang.Boolean
Default	true
Description	Whether the execution of workflow final actions will be retried if a com.core-media.cap.workflow.plugin.RetryableActionException was thrown. If set to false, retries after these exceptions are disabled completely and this cannot be overridden in the implementation of the RetryableActionException. This setting can be changed at runtime with the JMX attribute 'FinalActionsRetryEnabled'.

`workflow.server.retry-final-actions.max-retries`

Type	java.lang.Integer
Default	-1
Description	The maximum number of retry attempts, after a final action has failed with a com.coremedia.cap.workflow.plugin.RetryableActionException for a finished process instance. When the maximum number of retry attempts is reached, then the action will be skipped and the process instance will be deleted without successful execution of the failed final action. A value of -1 means unlimited retries. A value of 0 disables retries.

	This setting is ignored if the exception provides a different value in method RetryableActionException#getMaxRetries.

`workflow.server.session.reaper-timeout`

Type	java.lang.Integer
Default	60
Description	The interval of the session reaper checks.

`workflow.server.session.timeout`

Type	java.lang.Integer
Default	300
Description	The timeout until an inactive session is closed.

`workflow.server.standalone`

Type	java.lang.String
Default	
Description	Possible values: 'true'|'false'|'embedded'

`workflow.server.sweeper.concurrent`

Type	java.lang.Boolean
Default	false
Description	The concurrent sweep mode.

`workflow.server.sweeper.delay`

Type	java.lang.Long
Default	0

Description	The sweep delay.

`workflow.server.termination-timeout-seconds`

Type	java.lang.Integer
Default	30
Description	Timeout for waiting on running threads on shutdown

`workflow.server.tx.idletimeout`

Type	java.lang.Integer
Default	30
Description	Transaction handler idle timeout.

`workflow.server.tx.max`

Type	java.lang.Integer
Default	20
Description	Limits the maximum number of database connections.

`workflow.server.tx.retry-on-connection-loss`

Type	java.lang.Boolean
Default	false
Description	If 'true', will retry failed transactions due to a database connection failure. Before retrying, the DB connection will be renewed. This can also be useful in clustered databases to cope with failover switch.

`workflow.server.tx.retry-on-connection-loss-delay`

Type	java.time.Duration
Default	15s

Description	Time to wait before a new attempt is made after a transaction failed due to database connection failure. Must be larger than zero.

`workflow.server.tx.retry-on-read-only-db`

Type	java.lang.Boolean
Default	false
Description	If 'true', will retry failed transactions due to a read-only database. Before retrying, the DB connection will be renewed. This can be useful when running on a clustered database (e.g., Amazon Aurora) that dynamically assigns writer and reader instances to cope with failover switch. This is currently only supported on MySQL and MariaDB.

`workflow.server.tx.retry-on-read-only-db-delay`

Type	java.time.Duration
Default	5s
Description	Time to wait before a new attempt is made after a transaction failed due to a read-only DB. Must be larger than zero.

`workflow.server.uapi.connect-retry-delay-seconds`

Type	java.lang.Long
Default	10
Description	The delay in seconds the *Workflow Server* waits between retries to connect to the *Content Server* when starting.

`workflow.server.uapi.session-pool.max-idle-count`

Type	java.lang.Integer
Default	1000
Description	The maximum number of idle sessions to keep in the *Content Server* session pool.

`workflow.server.uapi.session-pool.max-idle-time-m-s`

Type	java.lang.Long
Default	60000
Description	The time in milliseconds after which an idle session of the *Content Server* session pool should be closed.

`workflow.server.uapi.session-pool.max-open-count`

Type	java.lang.Integer
Default	1000
Description	The maximum number of sessions to hold at any time (busy + idle) in the *Content Server* session pool.

`workflow.serviceindicator`

Type	java.lang.String
Default	
Description	The server's service indicator

`workflow.user`

Type	java.lang.String
Default	workflow
Description	Define a user for the connection with the *Content Management Server*.

`workflow.working-dir`

Type	java.lang.String
Default	var/tmp
Description	Working directory of the workflow-server used to handle custom classes uploaded to the server. Path can be absolute or relative to user.dir system-property.

`workflow.localization.auto-merge.legacy-list-merge`

Type	java.lang.Boolean
Default	false
Description	Configures whether an old but backwards-compatible merge algorithm for merging list properties should be used. This also applies to struct lists. If true, merge conflicts are reported if there are any changes in overlapping regions of list values. This also includes changes in nested property values, for example if there are changes in translatable properties and those properties aren't excluded with workflow.localization.auto-merge.translatable=false, or if the AutoMergeStructListMapKeyFactory is configured to merge a struct list as map, and some non-key properties have changed. An example for the latter would be a conflict caused by removing a link in a master content and a changed annotation property for that link in the derived content. If false, an improved merge algorithm is used for list values, which tries to avoid merge conflicts in more cases.

`workflow.localization.auto-merge.translatable`

Type	java.lang.Boolean
Default	true
Description	Configures if translatable properties are automatically merged in translation workflows. If true, changes of translatable properties in a master content are also merged to derived content items by translation workflows, if there are no conflicting changes and the properties haven't been excluded otherwise. Opposed to non-translatable properties, merge conflicts for translatable properties are typically resolved silently in favor of the derived content and do not generate a warning for the user. There are some exceptions, especially when using the legacy list merge algorithm (see property workflow.localization.auto-merge.legacy-list-merge). Even if set to true, translatable properties can still be excluded from auto-merge, for example with extensions:automerge="false" in the document type definition or with a custom implementation of com.coremedia.translate.workflow.AutoMergePredicateFactory.

`workflow.archive.endpoint-url`

Type	java.lang.String
Default	http://localhost:41080/studio
Description	Endpoint of the process archive in the in-memory setup.

`workflow.archive.password`

Type	java.lang.String
Default	
Description	Password to use for process archiving at the endpoint in the in-memory setup.
`workflow.archive.username`	
Type	java.lang.String
Default	
Description	Username to use for process archiving at the endpoint in the in-memory setup.

Table 3.34. Workflow Server Properties

3.7 Commerce Hub Properties

This section lists the Commerce Adapter client properties. For the configuration of the Commerce Adapter properties themselves, see the vendor specific Commerce Adapter manuals.

`commerce.hub.caching.enabled`

Type	java.lang.Boolean
Default	true
Description	De-/activates the commerce cache. If activated the caching is adjusted by the related property keys in com.coremedia.blueprint.base.livecontext.commerce-cache-defaults.properties.

`commerce.hub.data.custom-entity-params`

Type	java.util.Map<java.lang.String,java.lang.String>
Default	
Description	The custom entity parameters.

`commerce.hub.data.deadline`

Type	java.time.Duration
Default	20s
Description	Deadline value to be used when calling a grpc service method. After this time, the client no longer expects an answer and cancels the call. Default is 20 seconds. If no unit is given the value is taken as seconds.

`commerce.hub.data.endpoint-shutdown-timeout-millis`

Type	java.lang.Integer
Default	1000
Description	Timeout millis to be used when shutting down the data endpoints.

`commerce.hub.data.endpoints`	
Type	java.util.Map<java.lang.String,java.lang.String>
Default	
Description	The endpoints of the commerce adapter services to connect to. Each key-value pair represents one commerce adapter service to be connected. The keys can freely be defined, but they should easily identify the connected commerce system. Please refer to the java documentation of the method *com.coremedia.blueprint.base.livecontext.client.settings.CommerceSettings#getEndpoint()*. The value consists of the host name and the port of the connected commerce adapter service.

Table 3.35. Commerce Hub Properties

3.8 Elastic Social Properties

3.8.1 General Elastic Social Properties

`elastic.core.event-collection-ttl-sec`	
Type	java.lang.Integer
Default	300
Description	Time-To-Live index (TTL) for the collection event in seconds. Set it to 0 to disable automatic deletion.

`elastic.social.comments.auto-reject-interval-ms`	
Type	java.time.Duration
Default	
Description	Interval in milliseconds in which comments will be automatically rejected when the number of complaints reaches the limit {@code elastic.social.comments.auto-reject-limit}.

`elastic.social.comments.auto-reject-limit`	
Type	java.lang.Integer
Default	0
Description	Number of complaints after which a comment will automatically be rejected. Set to 0 to disable automatic rejection.

`elastic.social.mails.on-profile-changes`	
Type	java.lang.Boolean
Default	true

Description	Flag which states if a mail should be send on profile changes

`elastic.social.mails.on-registration`

Type	java.lang.Boolean
Default	true
Description	Flag which states if a mail should be send on registration

`elastic.social.password-hash-algorithm`

Type	java.lang.String
Default	
Description	The password hashing algorithm for elastic social user passwords. Allowed values are: {@code sha1} and {@code bcrypt:N}. The algorithm {@code sha1} is not recommended as it is vulnerable to brute-force attacks. The work factor N for bcrypt salt generation can be changed from the default value to an integer value between 4 and 30 (inclusive). This should be adapted to the available CPU resources.

`elastic.social.users.auto-block-interval-ms`

Type	java.time.Duration
Default	
Description	Interval in milliseconds in which users will be automatically blocked when the number of complaints reaches the limit {@code elastic.social.users.auto-block-limit}

`elastic.social.users.auto-block-limit`

Type	java.lang.Integer
Default	0
Description	A number of complaints after which a user will automatically be blocked. Set to 0 to disable automatic rejection.

`elastic.social.users.pre-moderation-properties`

Type	java.util.List<java.lang.String>
Default	
Description	Properties for the pre-moderation of users.

`elastic.social.users.token-expiration-time-ms`

Type	java.time.Duration
Default	
Description	Token expiration time in milliseconds, defines the lifetime of the user token used for user activation and password reset. After expiration, the token becomes invalid

Table 3.36. Elastic Social Properties

3.8.2 MongoDB Properties

`mongodb.client-uri`

Type	java.lang.String
Default	mongodb://localhost:27017
Description	The standard MongoDB connection string URI is used to configure your MongoDB connection, for example, it allows you to configure read preferences and write concerns. The format of a client URI is documented under the following link: http://docs.mongodb.org/manual/reference/connection-string/.

`mongodb.host`

Type	java.lang.String
Default	localhost
Description	Set the hostname of the mongoDb server. Only relevant for development purposes.

`mongodb.models.create-indexes`

Type	java.lang.Boolean
Default	true
Description	Set to false to skip index creation on startup. This can be useful to speed up initial data import. Make sure to set this true before production use.

`mongodb.prefix`

Type	java.lang.String
Default	elastic
Description	Prefix for MongoDB database names. Change this when sharing a MongoDB installation with other Elastic Core applications.

Table 3.37. MongoDb Properties

3.8.3 Counter Properties

`counters.aggregation-interval-milliseconds.daily`

Type	java.time.Duration
Default	
Description	Interval in milliseconds in which aggregated values for counter events of the last 24 hours will be updated. Defaults to 300000 (5 min).

`counters.aggregation-interval-milliseconds.for-all`

Type	java.time.Duration
Default	
Description	Interval in milliseconds in which aggregated values for all historic counter events will be updated. Defaults to 86400000 (24 h).

`counters.aggregation-interval-milliseconds.monthly`

Type	java.time.Duration
Default	
Description	Interval in milliseconds in which aggregated values for counter events of the last 30 days will be updated. Defaults to 86400000 (24 hours).

`counters.aggregation-interval-milliseconds.weekly`

Type	java.time.Duration
Default	
Description	Interval in milliseconds in which aggregated values for counter events of the last 7 days will be updated. Defaults to 10800000 (3 hours).

`counters.aggregation-interval-milliseconds.yearly`

Type	java.time.Duration
Default	
Description	Interval in milliseconds in which aggregated values for counter events of the last 365 days will be updated. Defaults to 86400000 (24 hours).

Table 3.38. Counters Properties

3.8.4 Task Queue Properties

`taskqueues.execution-timeout-milli-seconds`

Type	java.time.Duration
Default	
Description	Specifies the task queue execution timeout in milliseconds. Defaults to 600000 (10 m).

`taskqueues.execution-warning-timeout-milli-seconds`

Type	java.time.Duration

Default	
Description	Specifies duration in milliseconds after log warning for all long running tasks should be shown. Defaults to 300000 (5 m).

`taskqueues.number-of-retries`

Type	java.lang.Integer
Default	10
Description	Specifies the number of retries for a MongoDB task.

`taskqueues.polling-interval-milli-seconds`

Type	java.time.Duration
Default	
Description	Specifies the duration of the task queue polling interval in milliseconds. Used by the scheduler in the MongoDbTaskQueueService. Defaults to 500 (0.5 s).

`taskqueues.recovery-interval-milli-seconds`

Type	java.time.Duration
Default	
Description	Specifies the duration of the task queue recovery interval in milliseconds. Used by the scheduler in the MongoDbTaskQueueService. Defaults to 60000 (1 m).

`taskqueues.retry-interval-milli-seconds`

Type	java.time.Duration
Default	
Description	Specifies task queue retry interval in milliseconds. Defaults to 300000 (5 m).

`taskqueues.worker-node`

Type	java.lang.Boolean

Default	false
Description	Set to false to disable execution of background tasks. This setting can be used to differentiate applications nodes into worker nodes which only execute background tasks and rendering nodes which serve requests.

Table 3.39. Task-Queues Properties

3.8.5 Elastic Social Solr Properties

`elastic.solr.cloud`

Type	java.lang.Boolean
Default	
Description	Whether to connect to SolrCloud. Do not set this property to let Elastic Core use the standard CoreMedia setting. If true, connect to a SolrCloud cluster. SolrCloud connection details must be set either as ZooKeeper addresses (elastic.solr.zookeeper.addresses) or, if the former is unset or empty as HTTP URLs (elastic.solr.url). If false, connect to stand-alone Solr nodes via HTTP URLs (elastic.solr.url).

`elastic.solr.connection-timeout`

Type	java.lang.Integer
Default	
Description	Connection timeout in milliseconds, or 0 for no timeout, or a negative value to use SolrClient default. Do not set this property to let Elastic Core use the standard CoreMedia setting.

`elastic.solr.index-config`

Type	java.lang.String
Default	elastic
Description	Name of the Apache Solr config set for Elastic Core applications. This config set must exist in the Solr server, typically as subdirectory of "$SOLR_HOME/configsets".

`elastic.solr.index-data-directory`	
Type	java.lang.String
Default	data
Description	Value for the "dataDir" parameter of the Solr CoreAdmin API / Collection API request to create a Solr index.

`elastic.solr.index-prefix`	
Type	java.lang.String
Default	elastic
Description	Prefix for Apache Solr index names. Change this when sharing an Apache Solr installation with other Elastic Core applications.

`elastic.solr.lazy-index-creation`	
Type	java.lang.Boolean
Default	false
Description	Whether all indices are created lazily. The default is false, and all indices are created eagerly except indices for tenants configured with elastic.solr.tenants-with-lazy-index-creation.

`elastic.solr.password`	
Type	java.lang.String
Default	
Description	Password for HTTP basic authentication, used if a non-empty solr.username has been specified. The value may have been encrypted with the tool "cm encryptpasswordproperty". Do not set this property to let Elastic Core use the standard CoreMedia setting.

`elastic.solr.socket-timeout`	
Type	java.lang.Integer

Default	
Description	Socket timeout in milliseconds, or 0 for no timeout, or a negative value to use SolrClient default. Do not set this property to let Elastic Core use the standard CoreMedia setting.

`elastic.solr.tenants-with-lazy-index-creation`

Type	java.util.List<java.lang.String>
Default	
Description	List of tenants for which indices are not eagerly created when the application starts, but only upon first access.

`elastic.solr.url`

Type	java.lang.String
Default	
Description	The Solr URL to connect to, or null to let Elastic Core use the standard CoreMedia setting. In a Solr leader/follower setup, this must be the URL of the Solr leader. For SolrCloud, it's recommended to set elastic.solr.zookeeper.addresses instead.

`elastic.solr.username`

Type	java.lang.String
Default	
Description	Username for HTTP basic authentication, or empty string for no authentication. Do not set this property to let Elastic Core use the standard CoreMedia setting.

`elastic.solr.zookeeper.addresses`

Type	java.util.List<java.lang.String>
Default	
Description	ZooKeeper addresses for connecting to SolrCloud. Only used if elastic.solr.cloud=true. Do not set this property to let Elastic Core use the standard CoreMedia setting.

`elastic.solr.zookeeper.chroot`	
Type	java.lang.String
Default	
Description	Optional ZooKeeper chroot path for Solr. ZooKeeper chroot support makes it possible to isolate the SolrCloud tree in a ZooKeeper instance that is Only used if elastic.solr.cloud=true and elastic.solr.zookeeper.addresses is set to non-empty value. Do not set this property to let Elastic Core use the standard CoreMedia setting.
`elastic.solr.zookeeper.client-timeout`	
Type	java.lang.Integer
Default	
Description	Client-timeout for ZooKeeper in milliseconds, or a negative value to use SolrClient default. Only used if elastic.solr.cloud=true and elastic.solr.zookeeper.addresses is set to non-empty value. Do not set this property to let Elastic Core use the standard CoreMedia setting.
`elastic.solr.zookeeper.connect-timeout`	
Type	java.lang.Integer
Default	
Description	Connect-timeout for ZooKeeper in milliseconds, or a negative value to use SolrClient default. Only used if elastic.solr.cloud=true and elastic.solr.zookeeper.addresses is set to non-empty value. Do not set this property to let Elastic Core use the standard CoreMedia setting.

Table 3.40. Elastic Solr Properties

3.8.6 Renamed Properties

Deprecated Name	New Name

Table 3.41. Renamed Elastic Social Properties

3.9 Importer Properties

`import.user`	
Value	String
Default	importer
Description	The name of the CoreMedia user with which the importer logs on. Make sure that the user has the rights required to carry out operations triggered by the import process, for example, creating a new document, editing, approving, publishing. For this purpose, the standard CoreMedia installation offers a predefined user called importer (password also importer).

`import.password`	
Value	String
Default	importer
Description	The password of the user to log in with.

`import.autoLogoutSeconds`	
Value	int
Default	-1
Description	This property defines the time of inactivity in seconds after which the importer should log out. When the importer is active again, it will log in at the server automatically. A value of "-1" means that the importer will not log out.

`import.multiResultGeneratorFactory.property.sleepingSeconds`	
Value	int
Default	-1
Description	An importer remains logged in per default, whether data are imported or not. When configuring `SubDirGenerators`, the property defines the number of seconds for

the importer to be inactive after the completion of the import. If the number of seconds is very large, it is reasonable to log out the importer automatically. In this case, the released importer license can be used by another importer. Note that the special value "-1" will cause the importer to terminate after importing the contents of the inbox directories.

Table 3.42. Properties of the cm-xmlimport.properties file

`import.loginTimeoutSeconds`	
Value	long
Default	-1
Description	This property defines the timeout for login attempts after which the importer aborts. If `import.loginTimeoutSeconds=-1`, the importer tries to login forever without abortion.

`import.enforceCompleteVersion`	
Value	Boolean
Default	true
Description	This property handles the processing of XML importer files. See Section 4.1, "The CoreMedia XML Format" in *Importer Manual* for details on the CoreMedia XML format. • `import.enforceCompleteVersion=true` For each <version> element in the importer file a new version will be created in the CoreMedia repository. For all properties of a version the values must be given. It is not allowed to omit a property. • `import.enforceCompleteVersion=false` Now it is possible to omit even all property elements of a version. If there are only action elements and the document already exists on the server, then no new version is created and the corresponding actions are applied to the document (delete) or to the latest document version on the server (approve, delete). If there is at least one <property> element in the <version> element then for every property that is specified in the document type but missing in the XML importer file, the property value of the predecessor document version is taken. If there is no predecessor version, then a default value is inserted, that depends on the property type.

`import.validate-textproperty`

Value	Boolean
Default	false
Description	If "true" the importer validates all XML text properties against the associated DTD. If a validation fails, no document is created on the server. For big XML properties the validation may take some time.
`import.removeBrokenLinks`	
Value	Boolean
Default	false
Description	If "true" the importer removes broken content links in link list and markup properties. In markup properties only the link tag (`a` or `img`) is removed, not the containing link text. Be careful when enabling this option, as it may lead to invalid XML in markup properties.
`import.entityResolverClass`	
Value	class name
Default	see description
Description	Configures the name of a class of type `org.xml.sax.EntityResolver` used to resolve entities in markup properties during XML validation. The default value is `com.coremedia.xml.ClasspathURLEntityResolver`.

Table 3.43. Properties of the cm-xmlimport.properties file

3.10 Search Related Properties

3.10.1 Content Feeder Properties

Properties for the Content Feeder

`feeder.content.background-feed-delay`	
Type	java.time.Duration
Default	3s
Description	The minimum time after editorial changes were sent to the Search Engine and before background feeding takes place. This is used to prioritize feeding of editorial changes over background feeding, for example to process rights-rule changes or for periodic issue reindexing. It should not be necessary to change the default setting.

`feeder.content.index-deleted`	
Type	java.lang.Boolean
Default	true
Description	Whether contents in the trash should be indexed. If you do not need to find contents in the trash and want to keep your index smaller, you can change this to false.

`feeder.content.index-groups`	
Type	java.lang.Boolean
Default	true
Description	Whether the IDs of groups with potential rights to read the content are indexed in the field "groups". This set of groups is then used to narrow a user's search to the contents where he might have read rights to. This is an optimization to get smaller search results for some queries and content structures and to get more accurate search suggestion counts. The client has to check for read rights anyway. For details, see also the description of the field "groups" in Solr schema.xml. If set to false, then you must also configure

	Studio Server and Content Server to not add a query condition for the indexed groups. To this end, set the Studio property "studio.rest.search-service.use-groups-filter-query" and the Content Server "solr.use-groups-filter-query" to "false".

`feeder.content.index-name-in-textbody`

Type	java.lang.Boolean
Default	true
Description	Whether the content name should be indexed in field "textbody". It can make sense to disable this if lots of content names contain unique identifiers (from third-party systems, for example) to avoid problems with too many unique terms in field "textbody".

`feeder.content.index-referrers`

Type	java.lang.Boolean
Default	false
Description	Whether a content is reindexed after its referrers have changed.

`feeder.content.issues.index`

Type	java.lang.Boolean
Default	true
Description	Whether to index content issues.

`feeder.content.issues.initial-feeding`

Type	java.lang.Boolean
Default	false
Description	Whether content issues are already part of the initial feeding of an empty index. This property does not have any effect if feeder.content.issues.index is set to false. If true, initial feeding may take longer. If false, feeding of content issues starts after initial feeding has been completed.

`feeder.content.issues.reindex-after`

Type	java.time.Duration
Default	1d
Description	The duration after which indexed issues are considered outdated and become subject to periodic reindexing. This property does not have any effect if feeder.content.issues.index or feeder.content.issues.reindex-periodically are set to false.

`feeder.content.issues.reindex-periodically`

Type	java.lang.Boolean
Default	true
Description	Whether content issues are reindexed periodically. Note that issue reindexing is performed with low priority, and will not block feeding of editorial changes. Issue reindexing will be paused as long as editorial changes need to be processed. This property does not have any effect if feeder.content.issues.index is set to false.

`feeder.content.issues.reindex-time-max-percentage`

Type	java.lang.Integer
Default	100
Description	The maximum percentage of time used to trigger issue reindexing. If set to a value below 100, periodic issue reindexing will try to pause and stay inactive for some time, so that it does not use more than the configured percentage of a time window, even if issues are older than configured in feeder.content.issues.reindex-after. This only applies to issue reindexing and the Content Feeder may still perform other tasks. The configured value must be in the range of 1 to 100. Note that issue reindexing is always performed with low priority, and will be paused as long as editorial changes need to be processed, even if this property is set to 100. This property does not have any effect if feeder.content.issues.index or feeder.content.issues.reindex-periodically are set to false.

`feeder.content.issues.reindex-time-window`

Type	java.time.Duration
Default	10m
Description	The time window used with feeder.content.issues.reindex-time-max-percentage. Larger values for the time window lead to less but longer pauses. This property does not have

	any effect if feeder.content.issues.index or feeder.content.issues.reindex-periodically are set to false, or if feeder.content.issues.reindex-time-max-percentage is 100.

`feeder.content.management.password`

Type	java.lang.String
Default	feeder
Description	The password to be used in the HTTP authentication of the administration page of the Content Feeder.

`feeder.content.management.user`

Type	java.lang.String
Default	feeder
Description	The user name to be used in the HTTP authentication of the administration page of the Content Feeder. This is not an account from the user management of the Content Server.

`feeder.content.partial-update-aspects`

Type	java.util.List<java.lang.String>
Default	*
Description	Configures the aspects of index documents that can be updated with a partial update, provided that the connected Indexer supports partial updates (for example, feeder.solr.partial-updates.enabled=true for Solr). Multiple values are separated by comma. Use the special value "*" to use partial updates for all aspects, if possible. An empty value means that partial updates are not used. See the API documentation of Feedable.isPartialUpdate, FeedableAspect and ContentFeedableAspect in package com.coremedia.cap.feeder for more details.

`feeder.content.property-type.blob-max-size`

Type	org.springframework.util.unit.DataSize
Default	5MB

Description	Configure the maximum size of indexed blob properties. Larger blob values will be skipped. This configuration can be overridden for specific MIME-types by customizing Spring bean "feederContentBlobMaxSizePerMimeType".

`feeder.content.property-type.blob-mime-type.excludes`

Type	java.util.List<java.lang.String>
Default	
Description	List of MIME-types of "Blob" properties excluded from indexing. You can exclude a more specific type (e.g. text/xml) while including the corresponding primary type (e.g. text/*).

`feeder.content.property-type.blob-mime-type.includes`

Type	java.util.List<java.lang.String>
Default	[text/*, application/pdf, application/msword, application/vnd.openxmlformats-office-document.wordprocessingml.document]
Description	List of MIME-types of indexed "Blob" properties. If you don't configure any MIME-types in the includes property, no blob properties will be indexed.

`feeder.content.property-type.date`

Type	java.lang.Boolean
Default	false
Description	Whether properties of type "Date" are indexed.

`feeder.content.property-type.integer`

Type	java.lang.Boolean
Default	false
Description	Whether properties of type "Integer" are indexed.

`feeder.content.property-type.link-list`

Type	java.lang.Boolean

Default	false
Description	Whether properties of type "LinkList" are indexed.

`feeder.content.property-type.string`

Type	java.lang.Boolean
Default	true
Description	Whether properties of type "String" are indexed.

`feeder.content.property-type.struct`

Type	java.lang.Boolean
Default	false
Description	Whether properties of type "Struct" are indexed.

`feeder.content.property-type.xml-grammars`

Type	java.util.List<java.lang.String>
Default	coremedia-richtext-1.0
Description	The list of grammars of indexed "Markup" properties (as used in the document type definition as attribute "Name" of element "XmlGrammar").

`feeder.content.retry-connect-to-index-delay`

Type	java.time.Duration
Default	10s
Description	The time to wait between retries to connect to the search engine on startup.

`feeder.content.type.excludes`

Type	java.util.List<java.lang.String>

Default	[Preferences, EditorPreferences, Dictionary, Query]
Description	List of abstract or concrete content types excluded from feeding. With the configuration of some type, all of its subtypes are excluded implicitly, if not configured otherwise. Note that it is an error to configure the same content type in this property and in feeder.content.type.includes. Rules for more specific types override rules for less specific types. Regular expressions are not supported.

`feeder.content.type.includes`

Type	java.util.List<java.lang.String>
Default	Content_
Description	List of abstract or concrete content types included for feeding. With the configuration of some type, all of its subtypes are included implicitly, if not configured otherwise. Note that it is an error to configure the same content type in this property and in feeder.content.type.excludes. Rules for more specific types override rules for less specific types. Regular expressions are not supported.

`feeder.content.update-groups-immediately`

Type	java.lang.Boolean
Default	false
Description	If feeder.content.index-groups is true, configures whether the field "groups" is updated immediately after a change of a folder's right rule. It is recommended to keep this set to false, and let the Content Feeder update the index field in the background with lower priority than updates for editorial changes. It is quite expensive to set this to true because all contents below the folder would be reindexed.

Table 3.44. Content Feeder Configuration Properties

Solr specific properties for Content Feeder

`feeder.solr.nested-documents.enabled`

Type	java.lang.Boolean
Default	true

Description	Whether storing nested feedables as nested documents is supported in Solr. This requires that the Solr schema contains a _root_ field. Note that if you add that field to the schema, you have to recreate the index from scratch.

`feeder.solr.nested-documents.skip-index-check`

Type	java.lang.Boolean
Default	false
Description	If feeder.solr.nested-documents.enabled is true, the Solr index schema is checked whether it contains the _root_ field. The Feeder will log a warning and not use nested documents, if feeding of nested documents is attempted but the index does not support it. You can set this property to true to skip checking the index schema.

`feeder.solr.partial-updates.enabled`

Type	java.lang.Boolean
Default	true
Description	Whether partial updates are supported for updating content metadata in Solr. This requires that all fields in the Solr index are configured as stored="true" or docValues="true" except fields that are copyField destinations, which must be configured as stored="false". This is because partial updates are applied to the index document reconstructed from the existing stored field values.

`feeder.solr.partial-updates.skip-index-check`

Type	java.lang.Boolean
Default	false
Description	If feeder.solr.partial-updates.enabled is true, the Solr index schema is analyzed whether fields are stored as required for partial updates. The Feeder will log a warning and not use partial update functionality if the index seems to not support it. You can set this property to true to skip the check.

`feeder.solr.send-retry-delay`

Type	java.time.Duration

Default	30s
Description	The delay to wait before the Feeder retries to send data after failures from Solr.

`feeder.content.issues.solr.fetch-size`

Type	java.lang.Integer
Default	1000
Description	The maximum number of results to fetch with a single paginated Solr query when retrieving content items with outdated issues. If more results are available, multiple queries with Solr cursor pagination will be used, and each one will be restricted to this configured maximum number of results.

`feeder.content.issues.solr.filter`

Type	java.lang.String
Default	types:Document_
Description	Solr filter query to restrict the content items for which outdated issues are reindexed.

`feeder.content.issues.solr.query-min-delay`

Type	java.time.Duration
Default	10s
Description	The minimum time to wait before Solr is queried again for content items with outdated issues after the last query. This delay is not used for paginated queries that just retrieve the next page for a previous query.

`solr.cloud`

Type	java.lang.Boolean
Default	false
Description	Whether to connect to SolrCloud. If true, connect to a SolrCloud cluster. SolrCloud connection details must be set either as ZooKeeper addresses (solr.zookeeper.addresses)

	or, if the former is unset or empty as HTTP URLs (solr.url). If false, connect to stand-alone Solr nodes via HTTP URLs (solr.url).

`solr.connection-timeout`

Type	java.lang.Integer
Default	0
Description	Connection timeout in milliseconds, or 0 for no timeout, or a negative value to use SolrClient default.

`solr.content.collection`

Type	java.lang.String
Default	studio
Description	The name of the Solr collection for editorial search.

`solr.content.config-set`

Type	java.lang.String
Default	content
Description	The name of the Solr config set to use when creating the collection for editorial search. This property is used by the Content Feeder.

`solr.index-data-directory`

Type	java.lang.String
Default	data
Description	Value for the "dataDir" parameter of the Solr CoreAdmin API / Collection API request to create a Solr index.

`solr.password`

Type	java.lang.String

Default	
Description	Password for HTTP basic authentication, used if a non-empty solr.username has been specified. The value may have been encrypted with the tool "cm encryptpasswordproperty".

`solr.proxy-host`

Type	java.lang.String
Default	
Description	Proxy host for Solr communication that needs to be set if a proxy should be used.

`solr.proxy-is-secure`

Type	java.lang.Boolean
Default	false
Description	Secure flag for Solr proxy.

`solr.proxy-is-socks4`

Type	java.lang.Boolean
Default	false
Description	SOCKS 4 flag for Solr proxy.

`solr.proxy-port`

Type	java.lang.Integer
Default	0
Description	Proxy port for Solr communication that needs to be set if a proxy should be used.

`solr.socket-timeout`

Type	java.lang.Integer

Default	600000
Description	Socket timeout in milliseconds, or 0 for no timeout, or a negative value to use SolrClient default.

`solr.url`

Type	java.util.List<java.lang.String>
Default	http://localhost:40080/solr
Description	The list of Solr URLs to connect to. These URLs are ignored if connecting to SolrCloud (solr.cloud=true) and non-empty ZooKeeper addresses (solr.zookeeper.addresses) have been set. For a Feeder application that is not connected to a SolrCloud cluster, a single URL to the Solr leader must be configured.

`solr.use-xml-response-writer`

Type	java.lang.Boolean
Default	false
Description	Whether SolrJ should use XML response format instead of Javabin format.

`solr.username`

Type	java.lang.String
Default	
Description	Username for HTTP basic authentication, or empty string for no authentication.

`solr.zookeeper.addresses`

Type	java.util.List<java.lang.String>
Default	
Description	ZooKeeper addresses for connecting to SolrCloud. Only used if solr.cloud=true.

`solr.zookeeper.chroot`

Type	java.lang.String
Default	
Description	Optional ZooKeeper chroot path for Solr. ZooKeeper chroot support makes it possible to isolate the SolrCloud tree in a ZooKeeper instance that is Only used if solr.cloud=true and solr.zookeeper.addresses is set to non-empty value.

`solr.zookeeper.client-timeout`

Type	java.lang.Integer
Default	10000
Description	Client-timeout for ZooKeeper in milliseconds, or a negative value to use SolrClient default. Only used if solr.cloud=true and solr.zookeeper.addresses is set to non-empty value.

`solr.zookeeper.connect-timeout`

Type	java.lang.Integer
Default	10000
Description	Connect-timeout for ZooKeeper in milliseconds, or a negative value to use SolrClient default. Only used if solr.cloud=true and solr.zookeeper.addresses is set to non-empty value.

`solr.use-http1`

Type	java.lang.Boolean
Default	false
Description	Whether HTTP/1 (true) or HTTP/2 (false) shall be used by Solr clients.

Table 3.45. Content Feeder Solr Configuration Properties

Login properties for Content Feeder

The following properties are used to define the login data for the *Content Server*

`repository.user`

Value	user name
Default	feeder
Description	The user account the *Content Feeder* uses to read content.
`repository.password`	
Value	password
Default	feeder
Description	The password for the user account of the *Content Feeder*.

Table 3.46. Properties for login

Batch configuration properties for Content Feeder

With these properties you can configure the processing of batches.

`feeder.batch.max-bytes`	
Type	org.springframework.util.unit.DataSize
Default	5MB
Description	The maximum batch size in bytes. The Feeder sends a batch to the search engine if its maximum size would be exceeded when adding more entries. Note, that byte computation is a rough estimate only. A smaller batch may be sent if the maximum number of index documents is reached before, or if configured delays are reached.
`feeder.batch.max-open`	
Type	java.lang.Integer
Default	5
Description	The maximum number of batches indexed in parallel. This setting is not used with the default integration of Apache Solr but only with custom implementations of the com.coremedia.cap.feeder.index.async.AsyncIndexer interface. The Feeder does not call the index method of the AsyncIndexer interface to index another batch if the max-

	imum number of parallel batches has been reached. The method will not be called until a callback about the persistence of one of these batches has been received.

`feeder.batch.max-processed`

Type	java.lang.Integer
Default	1
Description	The maximum number of batches processed by the Indexer in parallel. This setting is not used with the default integration of Apache Solr but only with custom implementations of the com.coremedia.cap.feeder.index.async.AsyncIndexer interface. The Feeder does not call the index method of the AsyncIndexer interface to index another batch if the configured number of currently processed batches has been reached. The method will not be called until a callback about completed processing or persistence of one of these batches has been received.

`feeder.batch.max-size`

Type	java.lang.Integer
Default	500
Description	The maximum number of index documents in a batch. If the maximum number is reached, the Feeder sends the batch to the search engine. A smaller batch may be sent if the maximum byte size is reached before, or if configured delays are reached.

`feeder.batch.retry-send-idle-delay`

Type	java.time.Duration
Default	1m
Description	The time to wait before retrying to send index documents to the search engine after failures. This delay is used if the feeder is idle.

`feeder.batch.retry-send-max-delay`

Type	java.time.Duration
Default	10m

Description	The maximum time to wait before retrying to send index documents to the search engine after failures. This delay is used if the feeder is not idle. The setting is typically larger than retry-send-idle-delay.
`feeder.batch.send-idle-delay`	
Type	java.time.Duration
Default	3s
Description	The time between adding an index document to a batch and sending that batch to the search engine, if the batch is not yet full according to the max-size and max-bytes configuration properties, and if the feeder is idle. If a change needs to be sent to the search engine, and no further changes were made within the specified time, then an index document for the change will be sent after that time to the search engine. A small delay ensures low latency for changes to become visible in the search engine, as long as the system is not too busy.
`feeder.batch.send-max-delay`	
Type	java.time.Duration
Default	20s
Description	The maximum time between adding an index document to a batch and sending that batch to the search engine. This setting is typically larger than send-idle-delay to allow batches to grow and increase throughput, for example when large amounts of content are created by an import process. The configured value may still be exceeded under high load, or if there are problems connecting to the search engine.

Table 3.47. Feeder Batch Configuration Properties

Properties to configure Apache Tika

You can customize text extraction with Apache Tika using the following properties:

`feeder.tika.append-metadata`	
Type	java.lang.String
Default	

Description	Comma-separated list of metadata identifiers returned by Apache Tika to append to the extracted body text.

`feeder.tika.config`

Type	org.springframework.core.io.Resource
Default	
Description	The location of a custom Tika Config XML, for example to customize the default Tika parsers. See Apache Tika documentation for details on configuring Tika. The value of this property must be a Spring Resource location (e.g. file:/path/to/local/file) or empty for defaults.

`feeder.tika.copy-metadata`

Type	java.lang.String
Default	
Description	Comma-separated list of metadata identifiers returned by Apache Tika and names of Feedable elements to copy the metadata to. Entries in the comma separated list have the following format: "metadata identifier"="element name". With Apache Solr, target index fields must be defined as multiValued="true" to avoid indexing errors if there are multiple metadata values with the same identifier.

`feeder.tika.timeout`

Type	java.time.Duration
Default	2m
Description	The maximum time after which text extraction from binary data with Apache Tika fails. If extraction fails, the binary data will be skipped for the index document. Lower values will avoid that the Feeder is blocked for a long time in text extraction.

`feeder.tika.warn-time-threshold`

Type	java.time.Duration
Default	15s

Description	The time after which a warning is logged when text extraction from binary data with Apache Tika takes some time.

`feeder.tika.zip-bomb-prevention.enabled`

Type	java.lang.Boolean
Default	true
Description	Sets whether Apache Tika's "Zip bomb" prevention is enabled. When a "Zip bomb" is detected, no text will be extracted from the Blob, but a warning will be logged. Note that "Zip bombs" are not restricted to ZIP files but also apply to PDFs or other formats. Disabled "Zip bomb" prevention bears the risk of OutOfMemoryError-s. Note that false positives are possible.

`feeder.tika.zip-bomb-prevention.maximum-compression-ratio`

Type	java.lang.Long
Default	-1
Description	Sets the ratio between output characters and input bytes for the Apache Tika "Zip bomb" prevention. If this ratio is exceeded (after the output threshold has been reached) then no text will be extracted and a warning will be logged. Set to -1 to use the default of Apache Tika.

`feeder.tika.zip-bomb-prevention.maximum-depth`

Type	java.lang.Integer
Default	-1
Description	Sets the maximum XML element nesting level for the Apache Tika "Zip bomb" prevention. If this depth level is exceeded then no text will be extracted, and a warning will be logged. Set to -1 to use the default of Apache Tika.

`feeder.tika.zip-bomb-prevention.maximum-package-entry-depth`

Type	java.lang.Integer
Default	-1

Description	Sets the maximum package entry nesting level for the Apache Tika "Zip bomb" prevention. If this depth level is exceeded then no text will be extracted, and a warning will be logged. Set to -1 to use the default of Apache Tika.

Table 3.48. Feeder Tika Configuration Properties

Feeder Core Properties

You can use the following properties to customize some internal settings of the *Content Feeder*.

`feeder.core.executor-queue-capacity`	
Type	java.lang.Integer
Default	100
Description	Maximum capacity of the Feeder's executor queue, which is internally used to transfer evaluated values.

`feeder.core.executor-retry-delay`	
Type	java.time.Duration
Default	1m
Description	The delay to wait before the Feeder retries to access the source data after failures.

Table 3.49. Feeder Core Configuration Properties

3.10.2 CAE Feeder Properties

Properties for general configuration

`repository.user`	
Value	user name
Default	none

Description	The name of the user to connect to the *CoreMedia Content Server*.

`repository.password`

Value	password
Default	none
Description	The password of the user to connect to the *CoreMedia Content Server*.

`repository.domain`

Value	domain
Default	none
Description	The domain of the user to connect to the *CoreMedia Content Server*. Empty String for a built-in user.

`repository.url`

Value	URL
Default	none
Description	The URL to the IOR of the *CoreMedia Content Server*.

`jdbc.driver`

Value	driver class
Default	none
Description	The class of the database driver. For example: `oracle.jdbc.driver.OracleDriver`

`jdbc.url`

Value	URL
Default	none

Description	The URL to connect to the database.

`jdbc.user`

Value	user name
Default	none
Description	The name of the user to connect to the database.

`jdbc.login-user-name`

Value	the user name for the database login
Default	value of jdbc.user
Description	The user name for a database login. If not set, the value of "jdbc.user" will be used to log in to the database. In some cases the login username differs from the actual user, e.g. with PostgreSQL on Azure a postfix on the user name is necessary to log in. Set this property additionally to jdbc.user. (e.g. jdbc.login-user-name=username@domain jdbc.user=username).

`jdbc.password`

Value	password
Default	none
Description	The password of the user to connect to the database.

`feeder.contentSelector.basePath`

Value	String
Default	`/Sites`
Description	A comma-separated list of base folders for which content beans are indexed. Changing this property will not trigger any re-indexing of already indexed content. See Section 5.3.2, “Resetting” in *Search Manual* for details on re-indexing.

`feeder.contentSelector.contentTypes`

Value	String
Default	`Document_`
Description	A comma-separated list of content types for which content beans are indexed. Changing this property will not trigger any re-indexing of already indexed content. See Section 5.3.2, "Resetting" in *Search Manual* for details on re-indexing.

`feeder.contentSelector.includeSubTypes`

Value	Boolean
Default	`true`
Description	Specifies whether the sub types of the content types configured with property `feeder.contentSelector.contentTypes` are selected as well. Changing this property will not trigger any re-indexing of already indexed content. See Section 5.3.2, "Resetting" in *Search Manual* for details on re-indexing.

`feeder.core.executor-queue-capacity`

Value	int
Default	2000
Description	Capacity of the *CAE Feeder*'s executor queue, which is internally used to transfer evaluated values

`feeder.core.executor-retry-delay`

Value	milliseconds
Default	60000
Description	The delay in milliseconds to wait before the *CAE Feeder* retries to access the source data after failures to do so.

`feeder.batch.max-bytes`

Value	bytes
Default	20971520 (20 MB)

Description	The maximum size of a batch in bytes. The *CAE Feeder* sends a batch to the *Search Engine* if its maximum size would be exceeded when adding more entries. Note, that byte computation is a rough estimate only.

`feeder.batch.max-size`

Value	int
Default	500
Description	The maximum number of entries in a batch. If the maximum number is reached, the *CAE Feeder* sends the batch to the *Search Engine*.

`feeder.batch.max-open`

Value	int
Default	5
Description	The maximum number of batches indexed in parallel. This setting is not used with the default integration of Apache Solr but only with custom implementations of the `com.coremedia.cap.feeder.index.async.AsyncIndexer` interface. The *CAE Feeder* does not call the index method of the AsyncIndexer interface to index another batch if the maximum number of parallel batches has been reached. The method will not be called until a callback about the persistence of one of these batches has been received.

`feeder.batch.max-processed`

Value	int
Default	1
Description	The maximum number of batches processed by the Indexer in parallel. This setting is not used with the default integration of Apache Solr but only with custom implementations of the `com.coremedia.cap.feeder.index.async.AsyncIndexer` interface. The *CAE Feeder* does not call the index method of the AsyncIndexer interface to index another batch if the configured number of currently processed batches has been reached. The method will not be called until a callback about completed processing or persistence of one of these batches has been received.

`feeder.batch.retry-send-idle-delay`

Value	milliseconds
Default	60000
Description	The *CAE Feeder* sends a batch which only contains retried entries and is not full with regard to the `feeder.batch.max-size` and `feeder.batch.max-bytes` properties after the *CAE Feeder* was idle for the time configured in this property. A retried entry is an entry which was sent to the *Search Engine* before but could not be indexed successfully. If the batch contains entries which are not retried, the value of property `feeder.batch.send-idle-delay` is used instead.

`feeder.batch.retry-send-max-delay`

Value	milliseconds
Default	600000
Description	The maximum time in milliseconds between the time the *CAE Feeder* received an error from the *Search Engine* and the time, the *CAE Feeder* tries to send the failed entry as part of a batch to the *Search Engine* again. The time is exceeded if an error occurs while contacting the *Search Engine*. If the batch contains entries which are not retried, the value of property `feeder.batch.send-max-delay` is used instead.

`feeder.beanPropertyMaxBytes`

Value	number of bytes
Default	5242880 (5 MB)
Description	The maximum size in bytes for the value of a bean property or -1 for no limitation. Larger values are ignored and will not be sent to the Search Engine.

`feeder.beanMapping.mimeType.includes`

Value	comma-separated list of included MIME types
Default	*/*
Description	List of included MIME types for blob properties configured for indexing at the BeanMappingFeedablePopulator. For details, see the API documentation of method `setMimeTypeIncludes` of `com.coremedia.cap.feeder.bean.BeanMappingFeedablePopulator`

	Example `feeder.beanMapping.mimeType.includes=text/*` Only indexes blobs of MIME type `text/*`.

`feeder.beanMapping.mimeType.excludes`	
Value	comma-separated list of excluded MIME types
Default	
Description	List of excluded MIME types for blob properties configured for indexing at the BeanMappingFeedablePopulator. For details, see the API documentation of method `setMimeTypeExcludes` of `com.coremedia.cap.feeder.bean.BeanMappingFeedablePopulator` Example `feeder.beanMapping.mimeType.excludes=text/xml` Indexes all blobs except blobs of MIME type `text/xml`.

`feeder.batch.send-idle-delay`	
Value	milliseconds
Default	10000
Description	The *CAE Feeder* sends a batch which is not full with regard to the `feeder.batch.max-size` and `feeder.batch.max-bytes` properties after the *CAE Feeder* was idle for the configured time in milliseconds.

`feeder.batch.send-max-delay`	
Value	milliseconds
Default	120000
Description	The maximum time in milliseconds after which the *CAE Feeder* sends a batch which is not full with regard to the `feeder.batch.max-size` and `feeder.batch.max-bytes` properties. The time may be exceeded if an error occurs while contacting the *Search Engine* or if the *CAE Feeder* is under high load.

`proactiveengine.log.progress.interval.seconds`

Value	seconds
Default	600
Description	Set the time interval to log some statistics about the progress, including the number of keys that are currently invalid and still need to be computed.

`proactiveengine.senders.evaluators`

Value	number of threads
Default	50
Description	Number of evaluator threads in the *CAE Feeder*. The number of threads influences performance not only because evaluations can execute concurrently but also because higher values increase the probability that the *CAE Feeder* writes the state of multiple evaluations to the database in one database transaction.

`proactiveengine.senders.delay`

Value	milliseconds
Default	0
Description	Minimum delay in milliseconds between notifications of the Feeder by the internal *Proactive Engine* sub component. Higher values lead to reduced throughput.

`proactiveengine.senders.idledelay`

Value	milliseconds
Default	10000
Description	Delay in milliseconds between notifications of the Feeder by the internal *Proactive Engine* sub component if the application is idle. Smaller values can be configured to reduce the latency of the *CAE Feeder* but may lead to increased load on the database.

`dependencyStore.maxTransactionWeight`

Value	maximum number of changed keys per database transaction
Default	2500

Description	The maximum weight of a database transaction to change stored dependencies. The weight is interpreted as the number of changed keys, that is, a transaction with one deleted key has weight 1. Multiple transactions will be used to process an event that causes the invalidation of more keys.

Table 3.50. Configuration of general properties independent from the type of the search engine

Properties to configure Apache Tika

You can customize text extraction with Apache Tika using the following properties:

`feeder.tika.append-metadata`

Type	java.lang.String
Default	
Description	Comma-separated list of metadata identifiers returned by Apache Tika to append to the extracted body text.

`feeder.tika.config`

Type	org.springframework.core.io.Resource
Default	
Description	The location of a custom Tika Config XML, for example to customize the default Tika parsers. See Apache Tika documentation for details on configuring Tika. The value of this property must be a Spring Resource location (e.g. file:/path/to/local/file) or empty for defaults.

`feeder.tika.copy-metadata`

Type	java.lang.String
Default	
Description	Comma-separated list of metadata identifiers returned by Apache Tika and names of Feedable elements to copy the metadata to. Entries in the comma separated list have the following format: "metadata identifier"="element name". With Apache Solr, target index fields must be defined as multiValued="true" to avoid indexing errors if there are multiple metadata values with the same identifier.

`feeder.tika.timeout`	
Type	java.time.Duration
Default	2m
Description	The maximum time after which text extraction from binary data with Apache Tika fails. If extraction fails, the binary data will be skipped for the index document. Lower values will avoid that the Feeder is blocked for a long time in text extraction.

`feeder.tika.warn-time-threshold`	
Type	java.time.Duration
Default	15s
Description	The time after which a warning is logged when text extraction from binary data with Apache Tika takes some time.

`feeder.tika.zip-bomb-prevention.enabled`	
Type	java.lang.Boolean
Default	true
Description	Sets whether Apache Tika's "Zip bomb" prevention is enabled. When a "Zip bomb" is detected, no text will be extracted from the Blob, but a warning will be logged. Note that "Zip bombs" are not restricted to ZIP files but also apply to PDFs or other formats. Disabled "Zip bomb" prevention bears the risk of OutOfMemoryError-s. Note that false positives are possible.

`feeder.tika.zip-bomb-prevention.maximum-compression-ratio`	
Type	java.lang.Long
Default	-1
Description	Sets the ratio between output characters and input bytes for the Apache Tika "Zip bomb" prevention. If this ratio is exceeded (after the output threshold has been reached) then no text will be extracted and a warning will be logged. Set to -1 to use the default of Apache Tika.

`feeder.tika.zip-bomb-prevention.maximum-depth`	
Type	java.lang.Integer
Default	-1
Description	Sets the maximum XML element nesting level for the Apache Tika "Zip bomb" prevention. If this depth level is exceeded then no text will be extracted, and a warning will be logged. Set to -1 to use the default of Apache Tika.
`feeder.tika.zip-bomb-prevention.maximum-package-entry-depth`	
Type	java.lang.Integer
Default	-1
Description	Sets the maximum package entry nesting level for the Apache Tika "Zip bomb" prevention. If this depth level is exceeded then no text will be extracted, and a warning will be logged. Set to -1 to use the default of Apache Tika.

Table 3.51. Feeder Tika Configuration Properties

Properties for Solr configuration

The following properties are only used for a *CoreMedia Search Engine* based on Apache Solr:

`feeder.solr.nested-documents.enabled`	
Type	java.lang.Boolean
Default	true
Description	Whether storing nested feedables as nested documents is supported in Solr. This requires that the Solr schema contains a _root_ field. Note that if you add that field to the schema, you have to recreate the index from scratch.
`feeder.solr.nested-documents.skip-index-check`	
Type	java.lang.Boolean
Default	false

Description	If feeder.solr.nested-documents.enabled is true, the Solr index schema is checked whether it contains the _root_ field. The Feeder will log a warning and not use nested documents, if feeding of nested documents is attempted but the index does not support it. You can set this property to true to skip checking the index schema.

`feeder.solr.send-retry-delay`

Type	java.time.Duration
Default	30s
Description	The delay to wait before the Feeder retries to send data after failures from Solr.

`solr.cae.collection`

Type	java.lang.String
Default	
Description	The name of the Solr collection for web site search. This property does not have a default. It's typically set to 'preview' or 'live'.

`solr.cae.config-set`

Type	java.lang.String
Default	cae
Description	The name of the Solr config set to use when creating the CAE collection. This property is used by the CAE Feeder.

`solr.cloud`

Type	java.lang.Boolean
Default	false
Description	Whether to connect to SolrCloud. If true, connect to a SolrCloud cluster. SolrCloud connection details must be set either as ZooKeeper addresses (solr.zookeeper.addresses) or, if the former is unset or empty as HTTP URLs (solr.url). If false, connect to standalone Solr nodes via HTTP URLs (solr.url).

`solr.connection-timeout`	
Type	java.lang.Integer
Default	0
Description	Connection timeout in milliseconds, or 0 for no timeout, or a negative value to use SolrClient default.

`solr.index-data-directory`	
Type	java.lang.String
Default	data
Description	Value for the "dataDir" parameter of the Solr CoreAdmin API / Collection API request to create a Solr index.

`solr.password`	
Type	java.lang.String
Default	
Description	Password for HTTP basic authentication, used if a non-empty solr.username has been specified. The value may have been encrypted with the tool "cm encryptpasswordproperty".

`solr.proxy-host`	
Type	java.lang.String
Default	
Description	Proxy host for Solr communication that needs to be set if a proxy should be used.

`solr.proxy-is-secure`	
Type	java.lang.Boolean
Default	false

Description	Secure flag for Solr proxy.

`solr.proxy-is-socks4`

Type	java.lang.Boolean
Default	false
Description	SOCKS 4 flag for Solr proxy.

`solr.proxy-port`

Type	java.lang.Integer
Default	0
Description	Proxy port for Solr communication that needs to be set if a proxy should be used.

`solr.socket-timeout`

Type	java.lang.Integer
Default	600000
Description	Socket timeout in milliseconds, or 0 for no timeout, or a negative value to use SolrClient default.

`solr.url`

Type	java.util.List<java.lang.String>
Default	http://localhost:40080/solr
Description	The list of Solr URLs to connect to. These URLs are ignored if connecting to SolrCloud (solr.cloud=true) and non-empty ZooKeeper addresses (solr.zookeeper.addresses) have been set. For a Feeder application that is not connected to a SolrCloud cluster, a single URL to the Solr leader must be configured.

`solr.use-xml-response-writer`

Type	java.lang.Boolean

Default	false
Description	Whether SolrJ should use XML response format instead of Javabin format.

`solr.username`

Type	java.lang.String
Default	
Description	Username for HTTP basic authentication, or empty string for no authentication.

`solr.zookeeper.addresses`

Type	java.util.List<java.lang.String>
Default	
Description	ZooKeeper addresses for connecting to SolrCloud. Only used if solr.cloud=true.

`solr.zookeeper.chroot`

Type	java.lang.String
Default	
Description	Optional ZooKeeper chroot path for Solr. ZooKeeper chroot support makes it possible to isolate the SolrCloud tree in a ZooKeeper instance that is Only used if solr.cloud=true and solr.zookeeper.addresses is set to non-empty value.

`solr.zookeeper.client-timeout`

Type	java.lang.Integer
Default	10000
Description	Client-timeout for ZooKeeper in milliseconds, or a negative value to use SolrClient default. Only used if solr.cloud=true and solr.zookeeper.addresses is set to non-empty value.

`solr.zookeeper.connect-timeout`

Type	java.lang.Integer

Default	10000
Description	Connect-timeout for ZooKeeper in milliseconds, or a negative value to use SolrClient default. Only used if solr.cloud=true and solr.zookeeper.addresses is set to non-empty value.
`solr.use-http1`	
Type	java.lang.Boolean
Default	false
Description	Whether HTTP/1 (true) or HTTP/2 (false) shall be used by Solr clients.

Table 3.52. CAE Feeder Solr Configuration Properties

3.11 UAPI Client Properties

3.11.1 Unified API Spring Boot Client Properties

`repository.blob-cache-path`

Type	java.lang.String
Default	
Description	The directory in which cached blobs are stored. Make sure that the file system for this directory is large enough. Note that forced shutdowns of component's web application may result in leftover files in this directory, which should be cleared while the components are down. The configured directory is shared among components, because the actual cache content is placed in dynamically allocated subdirectories.

`repository.blob-cache-size`

Type	org.springframework.util.unit.DataSize
Default	-1B
Description	The maximum allowed size that the *transformed image blob cache* can occupy on the disk. This is a separate cache where results of blob transformations are stored persistently. Note that the file system overhead for storing the files does not count towards this value. So the physical space that has to be reserved on the disk for the cache has to be slightly higher than value of this configuration property. The value of "-1" means, that the default value "32000000" defined in *CacheFactory.java* for disk caches is used. If several concurrent threads write large blobs at the same time, the deletion of the folder with the old unused files can be postponed for later, thus this is the second reason why the maximum allowed cache size can grow slightly higher than this configuration property. The size of such deviation depend on the blobs size as well as the amount of parallel threads.

`repository.blob-streaming-size-threshold`

Type	org.springframework.util.unit.DataSize

Default	-1B
Description	The minimum size of streamed blobs. blobs less than or equal to this size will be downloaded completely to disk before the first byte can be read. Larger blobs will be downloaded in the background. Negative values trigger a fallback to the default as defined in *ConnectStrategy* (128 KB).

`repository.blob-streaming-threads`

Type	java.lang.Integer
Default	-1
Description	The number of threads reserved for streaming blob. The value of "-1" means, that the default value "2" defined in *ConnectStrategy.java* is used.

`repository.blob-upload.connect-timeout`

Type	java.time.Duration
Default	60s
Description	The timeout used for establishing a connection to the server for blob uploads.

`repository.blob-upload.request-timeout`

Type	java.time.Duration
Default	1h
Description	The timeout used for blob uploads. When uploading a blob, the data of the response must become available for reading before this timeout is exceeded.

`repository.caplist.connect`

Type	java.lang.Boolean
Default	
Description	Whether to connect the Cap list repository. Disabled by default.

`repository.caplist.mongo-db-client-uri`

Type	java.lang.String
Default	mongodb://localhost:27017
Description	The Cap List MongoDB connection string URI is used to configure your MongoDB connection. Property 'readpreference' must be 'primary'. See also mongoDb documentation.

`repository.caplist.mongo-db-prefix`

Type	java.lang.String
Default	elastic
Description	Prefix for Cap List MongoDB database names.

`repository.connect-retry-delay-seconds`

Type	java.lang.Long
Default	10
Description	The delay between UAPI connect retries.

`repository.domain`

Type	java.lang.String
Default	
Description	The domain of the *Content Server*

`repository.enable-blob-download-urls`

Type	java.lang.Boolean
Default	true
Description	A flag which allows to disable serving blob download URLs. The server side download URL computation is expensive, esp. if a client iterates over many contents, so that the download URLs are evicted from the cache. Clients may reduce server load by setting this flag to false, if they don't need the download URLs. Default is true.

`repository.force-immediate-login`	
Type	java.lang.Boolean
Default	false
Description	Configuration option that forces the UAPI connection to connect immediately instead of waiting for the *Content Server* to come up. This is useful for command line tools like the workflow converter that should provide quick feedback.

`repository.heap-cache-size`	
Type	org.springframework.util.unit.DataSize
Default	-1B
Description	The total number of bytes used by the main memory cache. For 32 bit JVMs this value is exact, for 64 bit JVMs, the actual memory consumption may be up to 2 times the configured value.

`repository.max-cached-blob-size`	
Type	org.springframework.util.unit.DataSize
Default	-1B
Description	The maximum size of blobs that are cached on the local disk. Larger blobs are downloaded from the Content Server on every request. The value of "-1" means, that the default value "Integer.MAX_VALUE" defined in *ConnectStrategy.java* is used.

`repository.password`	
Type	java.lang.String
Default	
Description	The password of the user.

`repository.url`	
Type	java.lang.String

Default	
Description	The URL of the *Content Server*. This property determines where to get the IOR of the *Content Server* (format: http://<server>:<port>/ior). • <server> must be the name of the *Content Server* host. • <port> must be the server's web server HTTP port.

`repository.user`

Type	java.lang.String
Default	
Description	Define the user which connects to the *Content Server*.

`repository.workflow.connect`

Type	java.lang.Boolean
Default	
Description	Whether to connect the workflow repository. Workflow is disabled by default.

`repository.workflow.url`

Type	java.lang.String
Default	
Description	The workflow server IOR URL.

Table 3.53. UAPI Spring Boot Client Properties

3.11.2 Renamed Properties

Deprecated Name	New Name

Table 3.54. Renamed UAPI Spring Boot Client Properties

3.12 Cache Properties

`cache.capacities`	
Type	java.util.Map<java.lang.String,java.lang.Long>
Default	
Description	Number of cache entries per cache class until cache eviction takes place. The keys must match the cache classes as defined by the cache keys. Please refer to javadoc of com.coremedia.cache.CacheKey.
`cache.timeout-seconds`	
Type	java.util.Map<java.lang.String,java.lang.Long>
Default	
Description	TTL in seconds until certain cache entries are invalidated.

Table 3.55. Cache Properties

3.13 Plugin Manager Properties

`plugins.directories`	
Type	java.util.List<java.lang.String>
Default	
Description	The directories from where plugins are loaded.
`plugins.required-plugins`	
Type	java.util.List<java.lang.String>
Default	
Description	A list of ids for plugins that are required. If one of these plugins is missing or could not be started, the application startup will be aborted. This property is only effective when 'plugins.directories' is set.

Table 3.56. Plugin Manager Properties

3.14 Blob Transformer Properties

`blobtransformer.enable-builtin-transformation`

Type	java.lang.Boolean
Default	true
Description	If 'false', the Spring beans for the built-in Java-based image transformation are not instantiated. The setting is for example automatically set to 'false' in a CoreMedia Content Cloud Service instance that is using the cloud-only image transformation service with WebP support. In this case, customizations of the 'BlobTransformer' are not possible and as such it helps preventing illegal configurations. The setting may only be set to 'false', if there is a plugin which provides a 'BlobTransformer'. Otherwise, the application will fail to start with a bean creation error.

Table 3.57. Blob Transformer Properties

3.15 Image Transformation Properties

`imagetransformation.default-avif-quality`	
Type	java.lang.Float
Default	0
Description	The default avif quality used for re-encoding. Initially, this is set to zero which suppresses adding the corresponding parameter to the transformation string. Choice of default quality is thus left to the image transformation component.

`imagetransformation.default-jpeg-quality`	
Type	java.lang.Float
Default	0.8
Description	The default jpeg quality used for re-encoding.

`imagetransformation.default-web-p-quality`	
Type	java.lang.Float
Default	0
Description	The default webp quality used for re-encoding. Initially, this is set to zero which suppresses adding the corresponding parameter to the transformation string. Choice of default quality is thus left to the image transformation component.

`imagetransformation.dynamic-variants`	
Type	java.lang.Boolean
Default	true
Description	If true, resolve transformations from content, otherwise only programmatically configured transformations are used.

`imagetransformation.remove-metadata`	
Type	java.lang.Boolean
Default	true
Description	The fallback value for removeMetadata. Will be effective for transformations where neither the given Breakpoint nor the given Transformation have a removeMetadata value. If true, the "rm" image operation will be applied to remove the image metadata
`imagetransformation.sharpen`	
Type	java.lang.Boolean
Default	true
Description	The fallback value for sharpen. Will be effective for transformations where neither the given Breakpoint nor the given Transformation have a sharpen value. If true, the "usm" image operation will be applied and the image will be sharpened.

Table 3.58. Image Transformation Properties

4. Encryption Service Setup

CoreMedia Content Cloud supports an encryption service that can be used to encrypt secret strings. This chapter covers the configuration of the mentioned service.
<section>
<title>Plugin Based Encryption Service Setup</title>
<para> The following table lists the properties that can be used to configure the plugin based encryption service. Plugin specific configuration properties are defined in the plugin documentation. </para>

`encryption.plugin-support.enabled`

Type	java.lang.Boolean
Default	false
Description	Whether loading of encryption service bean from plugins is enabled. Note that encryption service plugins must be 'independent'.

Table 4.1. Plugin Based Encryption Service Properties

</section>

4.1 Java Keystore based encryption service

This encryption service utilizes a single pair of public and private keys to encrypt and decrypt values. The keys are retrieved from a Java keystore located in the file system.

4.1.1 Prerequisites

Before you can use the keystore based encryption service, you have to create a keystore file using the Java `keytool` command. The keystore will contain the keys used for encryption and decryption. On the command prompt type:

```
keytool -genkeypair -keyalg RSA
    -keystore <KEYSTORE_FILENAME>
    -storepass <KEYSTORE_PASSWORD>
    -alias <KEY_ALIAS>
    -keypass <KEY_PASSWORD>
```

For secure usage at command line, it is not recommended to provide passwords directly on the command line. Use the secure input by not presenting a value for the password options. The `keytool` will prompt for secure password input. For details, see the documentation of `keytool`. Certain keystore types require identical passwords for the keystore and key. You will get an appropriate warning when generating the key in this case.

NOTE

For the above-mentioned command, the generated PKCS12 KeyStore needs to have the same `KEY_PASSWORD` and `KEYSTORE_PASSWORD`.

The `keytool` will prompt you for your username, organizational unit, organization, city, state/province and country code. This information (which goes into your self-signed certificate) is not relevant for the keystore to work properly. The resulting key/certificate will be valid for 90 days unless you specify another validity duration using the `validity` option. Note that you have to update the stored encrypted values when your keypair changes. Chose your keystore validity carefully.

The next step is to configure the keystore encryption service. This is described in the following sections. The recommended way is to use the Spring Boot configuration properties.

4.1.2 Configuration Properties Setup

The following table lists the properties that can be used to configure the keystore encryption service. They are all mandatory.

`encryption.keystore.cipher-transformation-name`	
Type	java.lang.String
Default	RSA
Description	The name of the cipher transformation which should be used for de- & encryption. It is strongly recommended to overwrite the default by setting this property to a more secure transformation with padding.

`encryption.keystore.key-alias`	
Type	java.lang.String
Default	
Description	Alias of the key inside the key store specified by encryption.keystore.keystore-file-location, that should be used for de- & encryption. The property value must not be blank.

`encryption.keystore.key-password`	
Type	java.lang.String
Default	
Description	The password of the key, specified by encryption.keystore.key-alias. The property value must not be blank.

`encryption.keystore.keystore-file-location`	
Type	java.lang.String
Default	${user.home}/.cmservices/.keystore
Description	Absolute path of the Java key store. The property value must not be blank.

`encryption.keystore.keystore-password`	
Type	java.lang.String
Default	
Description	Password of the Java key store specified by encryption.keystore.keystore-file-location. The property value must not be blank.

Table 4.2. Key Store Encryption Service Properties

NOTE

The encryption service Spring bean takes precedence over the static encryption service instance configured using the `CM_KEYSTORE_LOCATION` and `CM_KEYSTORE_PASSWORD_FILE_LOCATION` properties.

4.1.3 Properties File Setup

After creating the keystore, servers and clients need to have access to the keystore credentials. For this, they have to be stored in a Java properties file. Then the servers and clients will be able to access the keystore without prompting for passwords. The password file has to contain the following entries:

```
CM_KEYSTORE_PASSWORD=<KEYSTORE_PASSWORD>
CM_KEY_ALIAS=<KEY_ALIAS>
CM_KEY_PASSWORD=<KEY_PASSWORD>
```

As the password file contains the clear text passwords for your keystore, the file has to be protected from unauthorized access. This could be done by setting reasonable access rights for the file, or by putting it on a removable device.

Cipher transformation: By default the service uses less secure RSA-transformation, which is known to be available on all systems. For enhanced security, it is recommended to switch to an RSA algorithm with padding. You may do so by providing an additional property `CM_CIPHER_TRANSFORMATION` in the password file mentioned above. For available `Cipher` transformations, have a look at your installed security providers. By default, your Java platform should support the following `Cipher` transformations:

- `RSA` (default and fallback; see below)
- `RSA/ECB/PKCS1Padding`
- `RSA/ECB/OAEPWithSHA-1AndMGF1Padding`

- `RSA/ECB/OAEPWithSHA-256AndMGF1Padding`

Example configuration for different cipher transformation:

```
CM_KEYSTORE_PASSWORD=<KEYSTORE_PASSWORD>
CM_KEY_ALIAS=<KEY_ALIAS>
CM_KEY_PASSWORD=<KEY_PASSWORD>
CM_CIPHER_TRANSFORMATION=RSA/ECB/OAEPWithSHA-256AndMGF1Padding
```

Cipher transformation migration: To ease migrating from the default `RSA` transformation to a more secure transformation with padding, `RSA` is always used as a fallback. In case decrypting a password failed for the configured cipher, `RSA` will be tried as well.

Configuring the location of keystore files: In order to use the keystore with the encryption service, the service needs to know the location of the keystore and keystore password files. For this, you have the following two options:

- By default, the service expects
  - the keystore file under the path `${user.home}/.cmservices/.keystore`
  - and the password file under `${user.home}/.cmservices/.keystore.properties`
- If you want to store the files under different paths, you have to provide the following two system properties:
  - `CM_KEYSTORE_LOCATION`: location of the keystore file
  - `CM_KEYSTORE_PASSWORD_FILE_LOCATION`: location of the password file

NOTE

The encryption service Spring bean takes precedence over the static encryption service instance configured using the `CM_KEYSTORE_LOCATION` and `CM_KEYSTORE_PASSWORD_FILE_LOCATION` properties.

Index