

CoreMedia Digital Experience Platform 8  
//Version 7.5.45-10

COREMEDIA



# CoreMedia Analytics Connectors Manual

COREMEDIA



# CoreMedia Analytics Connectors Manual

**Copyright** CoreMedia AG © 2015

CoreMedia AG

Ludwig-Erhard-Straße 18

20459 Hamburg

## **International**

All rights reserved. No part of this manual or the corresponding program may be reproduced or copied in any form (print, photocopy or other process) without the written permission of CoreMedia AG.

## **Germany**

Alle Rechte vorbehalten. CoreMedia und weitere im Text erwähnte CoreMedia Produkte sowie die entsprechenden Logos sind Marken oder eingetragene Marken der CoreMedia AG in Deutschland. Alle anderen Namen von Produkten sind Marken der jeweiligen Firmen.

Das Handbuch bzw. Teile hiervon sowie die dazugehörigen Programme dürfen in keiner Weise (Druck, Fotokopie oder sonstige Verfahren) ohne schriftliche Genehmigung der CoreMedia AG reproduziert oder vervielfältigt werden. Unberührt hiervon bleiben die gesetzlich erlaubten Nutzungsarten nach dem UrhG.

## **Licenses and Trademarks**

All trademarks acknowledged.  
07.Mar 2017

1. Introduction .....	1
1.1. Audience .....	2
1.2. Typographic Conventions .....	3
1.3. CoreMedia Services .....	4
1.3.1. Registration .....	4
1.3.2. CoreMedia Releases .....	4
1.3.3. Documentation .....	5
1.3.4. CoreMedia Training .....	7
1.3.5. CoreMedia Support .....	8
1.4. Change Chapter .....	11
2. Overview .....	12
3. Tracking .....	13
3.1. Assembling Tracking Information .....	14
3.2. Views .....	15
3.3. JavaScript Code .....	16
3.3.1. Google Analytics .....	16
3.4. Studio Integration .....	17
3.4.1. Google Analytics .....	17
4. Retrieval .....	19
4.1. Google Analytics .....	22
4.2. Studio Extension .....	24
4.3. Analytics Feedback .....	25
5. Integrating an Analytics Service Provider .....	27
5.1. Studio Extension .....	28
5.2. CAE Extension .....	29
5.3. Elastic Worker Web Application Extension .....	30
Index .....	31

## List of Figures

4.1. Page Impression History for 30 days .....	26
--	----

## List of Tables

1.1. Typographic conventions .....	3
1.2. Pictographs .....	3
1.3. CoreMedia manuals .....	5
1.4. Log files check list .....	9
1.5. Changes .....	11
3.1. Google Analytics Tracking Configuration Options .....	18
4.1. Generic Retrieval Configuration Options .....	20
4.2. Google Analytics custom variables .....	22
4.3. Google Analytics Retrieval Configuration Options .....	22
4.4. Google Analytics Retrieval Configuration Options .....	25

# 1. Introduction

This manual describes the *CoreMedia Analytics Connectors*.

A general overview of analytics and how it is integrated into *CoreMedia Blueprint* is given in [Chapter 2, Overview \[12\]](#). Tracking is described in [Chapter 3, Tracking \[13\]](#). The description elaborates on the aspects configuration, assembling of tracking data and firing of tracking calls. [Chapter 4, Retrieval \[19\]](#) is dedicated to the topic of "top-n-lists", for example content lists that correspond to the top n entries of an analytics report. This manual closes with notes on how to integrate another third-party analytics solution in [Chapter 5, Integrating an Analytics Service Provider \[27\]](#).

## 1.1 Audience

This manual is intended for all readers who are interested to use one of the integrated third-party analytics solutions or to integrate a new one with *CoreMedia CMS*.

## 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	Courier new	cm contentserver
Command line entries		start
Parameter and values		
Menu names and entries	Bold, linked with	Open the menu entry <b>Format Normal</b>
Field names	Italic	Enter in the field <i>Heading</i>
CoreMedia Components		The <i>CoreMedia Component</i>
Entries	In quotation marks	Enter "On"
(Simultaneously) pressed keys	Bracketed in "<>", linked with "+"	Press the keys <Ctrl>+<A>
Emphasis	Italic	It is <i>not</i> saved
Buttons	Bold, with square brackets	Click on the <b>[OK]</b> button
Code lines in code examples which continue in the next line	\	cm contentserver \ start

Table 1.1. Typographic conventions

In addition, these symbols can mark single paragraphs:





Pictograph	Description
	Tip: This denotes best practices or recommendations.
	Warning: Please pay special attention to the text.
	Danger: The violation of these rules causes severe damage.
	Summary: This symbol indicates a summary of the above text.

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” \[4\]](#) for details on how to register.

### 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” \[4\]](#) describes how to register for the usage of the services.
- [Section 1.3.2, “CoreMedia Releases” \[4\]](#) describes where to find the download of the software.
- [Section 1.3.3, “Documentation” \[5\]](#) 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” \[7\]](#) describes CoreMedia training. This includes the training calendar, the curriculum and certification information.
- [Section 1.3.5, “CoreMedia Support” \[8\]](#) 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” \[8\]](#)) 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:

<http://releases.coremedia.com/dxp8>

Refer to our [Blueprint Github mirror repository](#) for recommendations to upgrade the workspace either via Git or patch files.



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” \[4\]](#) for details about the registration process. If the problems persist, try clearing your browser cache and cookies.

## Maven artifacts

CoreMedia provides 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 CoreMedia Digital Experience Platform 8 Developer Manual.

## License files

You need license files to run the CoreMedia system. Contact the support (see [Section 1.3.5, “CoreMedia Support” \[8\]](#)) to get your licences.

## 1.3.3 Documentation

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

<http://documentation.coremedia.com/dxp8>

The manuals have the following content and use cases:

Manual	Audience	Content
CoreMedia Utilized Open-Source Software	Developers, architects, administrators	This manual lists the third-party software used by CoreMedia and lists, when required, the licence texts.
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.
Studio User Manual, English	Editors	This manual describes the usage of <i>CoreMedia Studio</i> for editorial and administrative work. It also describes the usage of the <i>Adaptive Personalization</i> and <i>Elastic Social</i> GUI that are integrated into <i>Studio</i> .

Table 1.3. CoreMedia manuals

Manual	Audience	Content
LiveContext for IBM WebSphere Manual	Developers, architects, administrators	<p>This manual gives an overview over the structure and features of CoreMedia LiveContext. It describes the integration with the IBM WebSphere Commerce system, the content type model, the <i>Studio</i> extensions, folder and user rights concept and many more details. It also describes administrative tasks for the features.</p> <p>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.</p>
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 or the usage of the watchdog component.
Adaptive Personalization Manual	Developers, architects, administrators	This manual describes the configuration of and development with <i>Adaptive Personalization</i> , the CoreMedia module for personalized websites. You will learn how to configure the GUI used in <i>CoreMedia Studio</i> , 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.
Content Application Developer Manual	Developers, architects	This manual describes concepts and development of the <i>Content Application Engine (CAE)</i> . You will learn how to write JSP or Freemarker templates that access the other CoreMedia modules and use the sophisticated caching mechanisms of the CAE.
Content Server Manual	Developers, architects, administrators	This manual describes the concepts and administration of the main CoreMedia component, the <i>Content Server</i> . 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.

Manual	Audience	Content
Elastic Social Manual	Developers, architects, administrators	This manual describes the concepts and administration of the <i>Elastic Social</i> module and how you can integrate it into your websites.
Importer Manual	Developers, architects	This manual describes the structure of the internal CoreMedia XML format used for storing data, how you set up an <i>Importer</i> application and how you define the transformations that convert your content into CoreMedia content.
Search Manual	Developers, architects, administrators	This manual describes the configuration and customization of the <i>CoreMedia Search Engine</i> and the two feeder applications: the <i>Content Feeder</i> and the <i>CAE Feeder</i> .
Site Manager Developer Manual	Developers, architects, administrators	This manual describes the configuration and customization of <i>Site Manager</i> , the Java based stand-alone application for administrative tasks. You will learn how to configure the <i>Site Manager</i> with property files and XML files and how to develop your own extensions using the <i>Site Manager API</i> .
Studio Developer Manual	Developers, architects	This manual describes the concepts and extension of <i>CoreMedia Studio</i> . You will learn about the underlying concepts, how to use the development environment and how to customize <i>Studio</i> to your needs.
Unified API Developer Manual	Developers, architects	This manual describes the concepts and usage of the <i>CoreMedia Unified API</i> , which is the recommended API for most applications. This includes access to the content repository, the workflow repository and the user repository.
Workflow Manual	Developers, architects, administrators	This manual describes the <i>Workflow Server</i> . This includes the administration of the server, the development of workflows using the XML language and the development of extensions.

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

Email: [documentation@coremedia.com](mailto:documentation@coremedia.com)

### 1.3.4 CoreMedia Training

CoreMedia's training department provides you with the training for your CoreMedia projects either 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](mailto: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" \[4\]](#). The support email address is:

Email: [support@coremedia.com](mailto:support@coremedia.com)

#### Create a 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:

*Support request*

- 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.

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

*Support checklist*

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. The location where component log output is written is specified in its `<appName>-logback.xml` file.

*Log files*

### Which Log File?

Mostly at least two CoreMedia components are involved in errors. In most cases, the *Content Server* log files in `coremedia.log` files together with the log file from the client. If you are able locate the problem exactly, solving the problem becomes much easier.

### Where do I Find the Log Files?

By default, log files can be found in the CoreMedia component's installation directory in `/var/logs` or for web applications in the `logs/` directory of the servlet container. See the "Logging" chapter of the [Operations Basics Manual] for details.

*Table 1.4. Log files check list*

Component	Problem	Log files
CoreMedia Studio	general	CoreMedia-Studio.log coremedia.log
CoreMedia Editor	general	editor.log coremedia.log workflowserver.log capclient.properties
	check-in/check-out	editor.log coremedia.log workflowserver.log capclient.properties
	publication or pre-view	coremedia.log (Content Management Server) coremedia.log (Master Live Server)

Component	Problem	Log files
		workflowserver.log capclient.properties
	import	importer.log coremedia.log capclient.properties
	workflow	editor.log workflow.log coremedia.log capclient.properties
	spell check	editor.log MS Office version details coremedia.log
	licenses	coremedia.log (Content Management Server) coremedia.log (Master Live Server)
Server and client	communication errors	editor.log coremedia.log (Content Management Server) coremedia.log (Master Live Server) *.jpfif files
	preview not running	coremedia.log (content server) preview.log
	website not running	coremedia.log (Content Management Server) coremedia.log (Master Live Server) coremedia.log (Replication Live Server) Blueprint.log capclient.properties license.zip
Server	not starting	coremedia.log (Content Management Server) coremedia.log (Master Live Server) coremedia.log (Replication Live Server) capclient.properties license.zip

## 1.4 Change Chapter

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

Table 1.5. Changes

Section	Version	Description
<a href="#">Chapter 3, <i>Tracking</i> [13]</a> , <a href="#">Section 3.1, “Assembling Tracking Information” [14]</a>	24	Spring MVC interceptors exposing configurations for third-party service provider are replaced by CAE page aspects.
<a href="#">Section 3.2, “Views” [15]</a>	24	Associated third-party service provider JavaScript files are loaded from content repository and are no longer provided as static resources served from a JAR file.
<a href="#">Section 3.4, “Studio Integration” [17]</a>	24	Studio forms for tracking configuration have been removed. These settings are only configurable with the settings editor now.



## 2. Overview

*CoreMedia Analytics Connectors* demonstrates how to integrate third-party analytics services into *CoreMedia CMS*. The following third-party analytics services are integrated:

- Google Analytics

The integration extends the delivery side (CAE) with tracking of page impressions. To take advantage of tracked data, this contribution extends the document types to support "top-n-lists" based on tracking data. A "top-n-list" document aggregates a list of "n" top-performing documents - regarding page impressions -, where "n" is the number of resulting documents to be displayed. *CoreMedia Studio* is extended with document forms to set up top-n-lists and to configure the external analytics services.

If your project is based on *CoreMedia Blueprint*, you will be able to use the integration out of the box. Otherwise, it serves as an example of how to integrate external analytics services into your *CoreMedia* project.

*CoreMedia Analytics Connectors* combines the following major components:

- *CoreMedia Content Server* content types to define "top-n-lists"
- CAE content beans and service beans to render "top-n-lists"  
Services include retrieval of analytics data, generation of "top-n-lists" and access to analytics related settings.
- Solr external file field exporters
- a plugin for *CoreMedia Studio*  
The plugin allows editors to edit "top-n-list" documents and to configure the analytics service specific parameters to enable tracking and retrieval of tracked data.

## 3. Tracking

Tracking user actions on a website is typically implemented by adding calls to vendor specific JavaScript functions to the pages of the site. These functions populate a data structure which is eventually sent to the analytics service via a HTTP request for an invisible image, also known as tracking pixel.

The data being sent to the analytics service includes data about the following topics:

→ Content being displayed on the page

In addition, each analytics service requires some way of identifying the account the incoming data is to be written to.

The tracking configuration is stored in the content repository within the `settings` property of navigation content documents. Note that settings can be linked to a content and content objects inherit settings defined for their navigation contexts (channels). So settings defined at a navigation override settings defined at its parent navigation. For a Page content bean named `page`, the Google Analytics configuration, for example, is stored under the property path `page.settings.googleAnalytics`.

The configuration options for tracking are described further down in [Section 3.4, “Studio Integration” \[17\]](#).

## 3.1 Assembling Tracking Information

*CoreMedia Analytics Connectors* provides a `ViewHookEventListener` and an `AnalyticsProvider` implementation for each integrated third-party service provider. The `ViewHookEventListener` reacts on a `ViewHookEvent` of type `head` for content of type `Page`. If sufficient configuration is available, it renders the corresponding provider's JavaScript into the `head` section of the `Page`. The provider specific `AnalyticsProvider` implementation provides access to the basic configuration that is necessary to establish a tracking connection to that particular service. Most importantly, the `AnalyticsProvider` implementation can check if any required properties are missing and suppresses rendering of any output for that service if it is not properly configured.

Note that the integration of analytics extensions is controlled by the *CoreMedia* project extension Maven plugin. Per default, all analytics extensions are active but can be turned off by removing the corresponding BOM POM import from the *CoreMedia 7* root pom.

See the Javadoc for more details on which properties the analytics listeners and interceptors provide for tracking and [Section 3.4, "Studio Integration" \[17\]](#) for details on how to configure them.

Consult the [CAE Developer manual] for information on how to register interceptors and `ViewHookEventListeners`.

## 3.2 Views

As explained in the previous section, *CAE ViewHookEventListeners* are used to make data to be tracked (and the tracking configuration itself) accessible when rendering a view. This data is used in views to build the tracking calls. *CAE ViewHookEventListeners* react on rendering of `com.coremedia.blueprint.common.contentbeans.Page` beans to add code into the head of a page. The analytics integration uses the head view that serves the following purposes:

- `head`
  - Includes third-party tracking libraries,
  - Sets up vendor-specific JavaScript data structures required for tracking and
  - Includes the asynchronous calls to the JavaScript tracking code.

`GoogleAnalytics.head.ftl` includes JavaScript libraries specific for an external service and converts the tracking data into JavaScript objects used by the final tracking calls. The FTL checks if the service provider that it implements is enabled for the current page, that is, if it is properly configured and not explicitly disabled.

Page view tracking calls use the property "enabled" to check whether the third-party service is enabled. Setting this property explicitly to "false", disables the service provider, while setting it to "true" will only enable further processing of the provider's configuration (which might ultimately enable that service if the configuration is complete).

## 3.3 JavaScript Code

Tracking is performed by calling vendor-specific JavaScript functions. *CoreMedia Analytics Connectors* offers a thin layer around these.

Page view tracking calls are fired when a page is loaded. Corresponding calls are either included by the vendor specific JavaScript files to include, or have to be included in the implementation's `asHead` FTL of its analytics implementation.

The following subsections describe the JavaScript variables that are set in the `head` views of the FTLs described in the previous section.

### 3.3.1 Google Analytics

The Google Analytics integration provides an abstraction layer implemented in a JavaScript file `alx-integration-googleanalytics.js` which contains constructors for objects holding common data and functions to track page views and events:

- `GaAccountData` for the web property id and domain names,
- `GaPageviewData` for data related to a page view, such as the URL and the content id, and
- `GaEventData` for data related to an observed event, such as event category and label.

These data objects are supplied to the implemented functions

- `gaTrackPageview` for tracking page views, and
- `gaTrackEvent` for tracking events.

See JavaScript inline documentation for details.

## 3.4 Studio Integration

*CoreMedia Analytics Connectors* can be configured per site and per page. The settings for this can be configured using the struct editor for the property field `localSettings`. Each tracking provider is configured in a separate `StructProperty`. An example of the configuration is shown below.

```
<StructProperty Name="googleAnalytics">
<Struct>
<IntProperty Name="pageViewHistoryInterval">60</IntProperty>
<StringProperty Name="pageReport">content-pages</StringProperty>
<IntProperty Name="accountId">12345678</IntProperty>
<IntProperty Name="wpid">12345678</IntProperty>
<IntProperty Name="pid">12345678</IntProperty>
<StringProperty Name="webPropertyId">UA-XXXXXXX-1</StringProperty>
<StringProperty Name="domainName">auto</StringProperty>
<LinkProperty Name="pl2File"
LinkType="coremedia:///cap/contenttype/CMDownload"
  xlink:href="Options/Settings/googleAnalytics%2Ep12.xml"
  cmexport:path="Settings/googleAnalytics%2Ep12"/>
<StringProperty
Name="serviceAccountEmail">ID@developer.gserviceaccount.com</StringProperty>
<StringProperty Name="applicationName">appName</StringProperty>
<IntProperty Name="interval">120</IntProperty>
<IntProperty Name="limit">9999</IntProperty>
</Struct>
</StructProperty>
```

Note that tracking can be temporarily disabled for any service provider (even for a particular page) by adding a Boolean property `disabled` with value `true` to the provider's struct property.



The `AnalyticsSettingsProvider` implementations can expose settings for Studio components. These can, for example, be used to provide deep links for service provider reports. See Javadoc for details.

In the following section describes the property names to be used in the generic struct editor of the *CoreMedia Studio*.



### 3.4.1 Google Analytics

The integration of Google Analytics not only allows you to configure tracking but also to configure the preview toolbar button to open a Google Analytics report drilled down to the active content object. Both aspects are covered by the options presented in the following table.

Table 3.1. Google Analytics Tracking Configuration Options

Technical Variable Name	Description/Value	Required
webPropertyId	The Google Analytics web property ID to track to. The Web Property ID has the format UA-12345678-1	true
domainName	The domain name to be sent to Google Analytics, if you are running a multi-site environment.	false
pageReport	The name of the report to create the drill down URL for. Defaults to <code>content-pages</code>	false
accountId	Your numeric Google Analytics account ID to create the drill down URL for. See the tip below.	false
wpid	Your numeric Google Analytics web property ID to create the drill down URL for. See the paragraph below.	false
pid	Your numeric Google Analytics profile ID to create the drill down URL for. See the paragraph below.	false

Given a Google Analytics drill down URL, you may notice the URL a hash parameter fragment of the form `/aXXXXXXXXwYYYYYYYpZZZZZZZZ/`. In this case, `XXXXXXXX` is your 8 digit account ID, `YYYYYYY` is your 8 digit web property ID and `ZZZZZZZZ` is your 8 digit profile ID. Each of these values is required to build a valid report drill down URL.



If the *CoreMedia Elastic Social* Extension is enabled and social tracking is configured, `webPropertyId` must be set to the same value as `socialTrackingId` in the *Elastic Social* configuration.



## 4. Retrieval

Data aggregated by analytics service providers can not only be used to generate tables and diagrams but also to generate "top-n-lists" for use on the delivery side of a content management environment. To generate "top-n-lists" of contents based on their rank within an analytics report, report data is gathered and cached using *CoreMedia Elastic Core* infrastructure. The following components play key roles in this setting:

- `CMALXBaseList`  
content objects

Instances of this content type serve as configuration objects for a retrieval task that fetches the corresponding data. The content beans are also used at rendering time to retrieve the content objects corresponding to the tracking data cached using *CoreMedia Elastic Core*.

- `AnalyticsServiceProvider`

Implementations actually access the third-party analytics service provider and gather data. Data is persisted using the `CMALXBaseListModelService`. This model service retrieves and stores objects of type `ReportModel` which hold the current configuration of the report, the preprocessed report data and a timestamp to represent the reports freshness.

- `FetchReportsTask`

An elastic worker task iterates over the "top-n-list" documents of a tenant and uses all `AnalyticsServiceProvider` implementations available in the current Spring context to retrieve data for them. First the task gets all root navigation items and "top-n-list" items for a tenant and executes for "top-n-list" documents and the root documents of the same site. Then task checks, if the corresponding `ReportModel` is too old or differs in its configuration. This ensures that changes in the configuration trigger an almost immediate new retrieval of data. If retrieval is due, data is fetched and passed to the `CMALXBaseList` instance to preprocess the result list. Then the report model is saved.

There are different types of time intervals involved, which can be confusing:

- Interval of the `FetchReportsTask` - the task is executed quite often, for example every minute, but only fetches data if necessary.



- ➔ Interval for retrieving data from a specific analytics provider - the effective retrieval interval in which data is actually retrieved if the configuration has not changed, for example every 180 minutes. It is configured per top-n-list instance using the `interval` property (see Table 4.1, “Generic Retrieval Configuration Options” [20]).
- ➔ Time range of the fetched data - usually you only retrieve data for a certain time range, for example you are interested in the report data for the last week.

`FetchReportTask` assumes that data is fetched synchronously. If the analytics service provider provides asynchronous access only, you will have to set up additional tasks that fetch the report data. An implementation of `ElasticAnalyticsServiceProvider` should then store information used by the additional tasks (for subsequent calls to the analytics service provider) in the report model and return an empty list themselves.



Top-n-lists (`CMALXBaseList` instances) provide an `analyticsProvider` property that determines the analytics service provider to use.

Retrieval configuration is stored in the settings of the content (or one of its channels). Note that settings defined at a content override settings defined by its channel. For a Page content bean named `page`, the Google Analytics configuration, for example, is stored under the property path `page.settings.googleAnalytics`.

The following configuration properties apply to all analytics service provider implementations:

Technical Variable Name	Description/Value	Required
<code>analyticsProvider</code>	The service key of the analytics provider to use. If not set, your list will be empty, even if data is cached.	false
<code>maxLength</code>	The 'n' of a top-n-list (its maximum length). Default is 10.	false
<code>interval</code>	The interval to fetch report data. Default is 24 hours.	false
<code>limit</code>	The maximum number of records of a fetched report. Default is 1000.	false

Table 4.1. Generic Retrieval Configuration Options

Note that retrieval can be temporarily disabled (even for a particular page) by setting the service's `interval` property, for example `googleAnalytics.interval`, to 0.



Subtypes of `CMALXBaseList` define report types to fetch. The following examples are provided by *CoreMedia Analytics Connectors*:

→ `CMALXPageList`

Instances of this content type refer to a report containing page views. Generic properties used at retrieval time are `documentType` and `baseChannel` limiting the items to display at rendering time. Hence, the "top-n-list" will be made up of content objects of type `documentType` below channel `baseChannel`. The property `defaultContent` defines a list of content objects to be displayed if no report data is available.

→ `CMALXEventList`

Instances of this content type refer to a report containing tracked events. Generic event properties are `category` and `action`. The category is the name you supply for the group of objects to track. The action is a string identifying the type of user interaction to be tracked. The pair of category and action should uniquely identify the event.

On the retrieval side, only implementations of `AnalyticsServiceProvider` are specific to an analytics service provider. In the next sections, the existing service provider implementations are presented.

Cached report models are cleared, if the data has not been updated in the last 30 days.

The tables in the following sections use the technical names of configuration options. Look them up in the resource bundles of the corresponding Studio extension modules to find out the localized names of the properties.

The analytics service providers restrict usage in different ways (with respect to request frequency, request count per time unit or response size in terms of record count). Ensure that your configuration matches those limitations.



## 4.1 Google Analytics

Google Analytics is accessed using the `Google Analytics Universal API`. To access Google Analytics Data, you need to create a service account (Google Developer Console). The service account email and p12 Key File (PKCS Standard) are used for OAuth 2.0 authentication.

Remember to grant access to the analytics reports for the service account.



Custom variables must be defined as `dimension1...n` in the Google Analytics Web interface.

Table 4.2. Google Analytics custom variables

Custom Variable	Use
<code>dimension1</code>	Content Id
<code>dimension2</code>	Content Type
<code>dimension3</code>	Navigation Path

Both page views and tracked events are considered at retrieval time. The following configuration options are available:

Table 4.3. Google Analytics Retrieval Configuration Options

Technical Variable Name	Description/Value	Required
<code>applicationName</code>	Name of the app accessing Google Analytics data, used in the UserAgent header of each request.	true
<code>pid</code>	A numeric Google Analytics profile ID providing the data to fetch (compare with property <code>pid</code> described in <a href="#">Table 3.1, "Google Analytics Tracking Configuration Options" [18]</a> ).	true
<code>serviceAccountEmail</code>	Email address of the service account.	true
<code>p12File</code>	Create a CMDownload Document and upload the p12 Key File as blob. Link this document in the Analytics Settings P12 Key file field.	true

Please take care of security protection of the CMDownload Document containing your Google Analytics p12 Key File:

→ Restrict read access to the CMDownload Document



- Ensure that no link is generated to the CMDownload document, for example by linking to it in another document
- Exclude the document from website search by checking the corresponding option.

Please refer to the Google Analytics Developers Documentation for further information on request quotas.



## 4.2 Studio Extension

*CoreMedia Studio* plugins provide editing forms for the configuration of the "top-n-list" documents. They are available for the analytics document types `CMALXPageList` and `CMALXEventList`.

## 4.3 Analytics Feedback

To get direct feedback on the performance of an article, a page impression history is displayed in *CoreMedia Studio* on the metadata tab of each article. Furthermore, a *Site Performance Widget* can be configured on the *Dashboard* to show the total number of page impressions and publication events for a site.

The page impression history can be displayed for the last 7 or 30 days.

As the page impressions are retrieved asynchronously from a third-party analytics provider, the timestamp of the last retrieval is displayed.

Data is currently retrieved from Google Analytics only. Corresponding configuration must be provided.

The configuration option `pageViewHistoryInterval` is used exclusively for the page impression history. The setting defines the interval for requests to the analytics service provider. The configuration option `interval` has no effects for this functionality.

Table 4.4. Google Analytics Retrieval Configuration Options

Technical Variable Name	Description/Value	Required	Default Value
<code>pageViewHistoryInterval</code>	The update interval for requests to the analytics provider in minutes.	false	1440 (1 day)
<code>publicationHistoryInterval</code>	The update interval for calculating the site publication history in minutes.	false	180 (3h)
<code>publicationHistoryDocumentType</code>	The document type used for calculating the site publication history. It may contain a comma separated list.	false	CMLinkable



The jobs for data retrieval is triggered each minute. The analytics retrieval task will only retrieve data from the analytics provider if the configuration of the analytics provider has changed, on the first run or after the default interval (24 h) or the customized interval `pageViewHistoryInterval` has expired.

Page impressions equal to zero represent either no page impressions or no data available. If no data is available for the selected time span, no chart is displayed.

The page view history data is cleared for a document, if the data has not been updated in the last 30 days.

The publication history will be calculated, when the document type configuration has changed or when the data is older than configured in the interval configuration.

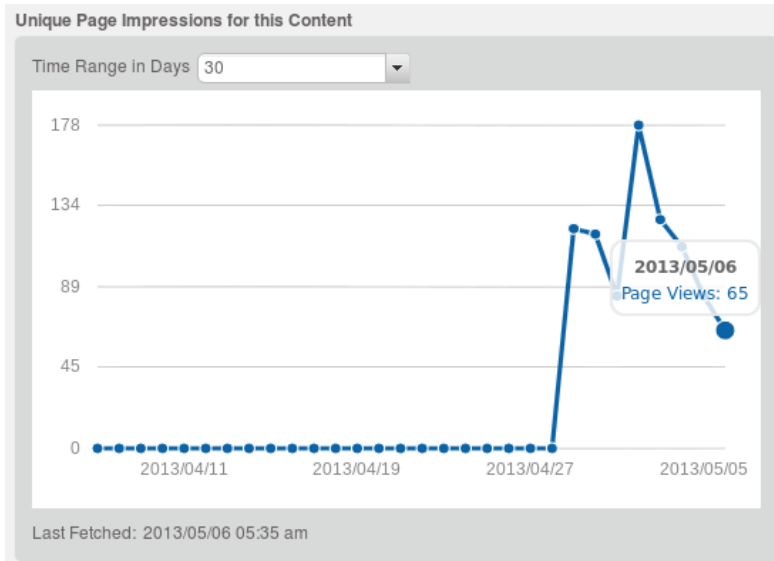


Figure 4.1. Page Impression History for 30 days

Publication events are integrated into the page impression history for articles. When you hover over the publication history, the corresponding date is marked in the page impression history.

Publication events are also aggregated per site and displayed in the *Site Performance Widget*. Aggregation is performed asynchronously, a timestamp indicates the date of the last retrieval. The default interval for aggregation is three hours.

## 5. Integrating an Analytics Service Provider

To integrate another analytics service provider you have to consider the following aspects:

- Configure the tracking and retrieval information of the new provider on the root channel of a site using the generic struct editor.
- Create a tracking extension that hooks into *CoreMedia CAE*.
- Create a retrieval extension that hooks into `es-alex-retrieval-component`. The extension requires a single managed bean of type `AnalyticsServiceProvider`.



## 5.1 Studio Extension

Forms for configuring tracking can be hooked into the `CMALXBaseList` editing forms using a *CoreMedia Studio* extension. It is good practice to further extend the existing extension points:

→ `CMALXBaseList` form

The form contains a tab for each analytics provider, so add a new tab. The tab contains only the retrieval configuration fields.

For more information have a look at the module `alx-studio-plugin` as it provides the base components to be used to implement the Studio plugin for analytics service providers.

## 5.2 CAE Extension

The purpose of the CAE integration is to add the tracking aspect to page impressions and user interactions with your website.

The recommended way is to use a `ViewHookEventListener` that adds tracking to `com.coremedia.blueprint.common.contentbeans.Page` instances. Add an implementation of `AnalyticsProvider` and a template that should prepare tracking calls in its `head` view. These views are automatically included when rendering a page and your `ViewHookEventListener` is present.

## 5.3 Elastic Worker Web Application Extension

Set up an extension for the `es-alx-retrieval-component` providing a managed bean of type `AnalyticsServiceProvider`. Have a look at the implementation of `ElasticGoogleAnalyticsServiceProvider` as example for how to implement synchronous access.

# Index

## V

views, 15

## A

Analytics feedback, 25  
assembling tracking information, 14

## C

components, 12

## G

Google Analytics  
retrieval, 22  
tracking configuration, 17

## I

integrate other service provider, 27  
integrated services, 12

## J

JavaScript code, 16

## R

retrieval, 19  
temporarily disabled, 20

## S

Studio integration, 17

## T

top-n-lists, 19  
Tracking, 13  
tracking  
configuration, 13  
identifying, 13  
sent data, 13