contentwise

Reference Guide

version 6.0

moviri
www.moviri.com

1. Preface . . . 4
2. Reference Guide . . . 5
2.1 Introduction . . . 6
2.2 Terminology . . . 7
2.2.1 Business Rule . . . 8
2.2.2 Caller . . . 15
2.2.3 Caller Group . . . 16
2.2.4 ContentWise License . . . 17
2.2.5 Context . . . 21
2.2.6 Event . . . 22
2.2.7 Fallback strategy . . . 26
2.2.8 Item . . . 27
2.2.8.1 Item hierarchies . . . 27
2.2.8.2 Program item . . . 29
2.2.9 Layout . . . 31
2.2.10 Metadata . . . 32
2.2.11 Profile . . . 32
2.2.12 Provider and Service . . . 33
2.2.13 Recommendation . . . 34
2.2.14 Statistic . . . 36
2.2.15 Subdomain . . . 38
2.2.15.1 Subdomain Rule . . . 39
2.2.16 Testing and experiments . . . 40
2.2.17 User . . . 41
2.2.17.1 User Explicit Preference . . . 42
2.2.18 User Group . . . 42
2.2.19 Time diversity . . . 42
2.2.20 Dynamic stream . . . 43
2.3 Architecture and Integration . . . 45
2.3.1 ProcServer . . . 48
2.3.1.1 ETL . . . 48
2.3.1.2 Task . . . 52
2.3.1.2.1 Engine . . . 53
2.3.1.2.2 Engine Runner . . . 54
2.3.1.2.3 Stage Manager . . . 55
2.3.2 RecServer . . . 56
2.4 ContentWise Portal . . . 57
2.4.1 Service Model . . . 59
2.4.1.1 Services . . . 59
2.4.1.2 Subdomains . . . 62
2.4.1.2.1 Definitions . . . 62
2.4.1.2.2 Subdomain rules . . . 69
2.4.2 Data Management . . . 72
2.4.2.1 Data Import . . . 72
2.4.2.1.1 ETL Configuration . . . 72
2.4.2.1.2 ETL Execution . . . 77
2.4.2.2 Tasks . . . 79
2.4.2.2.1 Task Configuration . . . 79
2.4.2.2.2 Task Execution . . . 84
2.4.2.3 Metadata Enhancer . . . 86
2.4.2.4 Recommendation Models . . . 89
2.4.2.5 Dynamic Lists . . . 91
2.4.2.6 Item Hierarchy . . . 96
2.4.3 UX Design . . . 99
2.4.3.1 UX Builder . . . 99
2.4.3.1.1 Layouts . . . 102
2.4.3.1.2 Dynamic Layout . . . 108
2.4.3.1.3 AutoComplete . . . 113
2.4.3.1.4 Editorial Lists . . . 114
2.4.3.1.5 Search . . . 115
2.4.3.2 UX Integration . . . 115
2.4.3.2.1 UX Reference . . . 116
2.4.3.2.2 UX Reference Groups . . . 116
2.4.4 Profiles . . . 118
2.4.5 Business Rules . . . 124
2.4.5.1 Rules . . . 124
2.4.5.2 Rule Groups . . . 128
2.4.6 Analytics . . . 131
2.4.6.1 Dashboards . . . 131
2.4.6.2 Reporting . . . 132
2.4.6.2.1 Browse Reports . . . 132
2.4.6.2.2 Configure Reports . . . 133
2.4.6.3 Inspect Catalogue . . . 136
2.4.6.4 A B Testing . . . 140
2.4.6.4.1 Experiments . . . 140
2.4.6.4.2 Variations . . . 141
2.4.7 Administration . . . 142
2.4.7.1 Data Types . . . 143
2.4.7.1.1 Algorithm Types . . . 143
2.4.7.1.2 Item Types . . . 144

2.4.7.1.3 Metadata Types 151
2.4.7.1.4 User Group Types 153
2.4.7.1.5 User Types 154
2.4.7.2 Rating Types 156
2.4.7.3 Deployment 164
2.4.7.3.1 Components 164
2.4.7.3.2 Component Pools 168
2.4.7.4 Provider Settings 169
2.4.7.5 Statistics 170
2.4.7.6 System Settings 172
2.4.7.6.1 Configuration Properties 172
2.4.7.6.2 License 174
2.4.7.6.3 WS Accounts 175
2.5 Start and stop services 177
2.6 Metadata reference 179
2.7 Supported formats 190
2.8 ContentWise XML format 190
3. TreeNavigation 192
4. Other product versions 193
5. Caller Groups 194
6. Callers Definition 196

Preface

Legal Notices

Warranty

The only warranties for ContentWise products and services are set forth in the express warranty statements accompanying such products and services. The information provided by this document should not be interpreted as constituting an additional warranty.
ContentWise is not liable for technical or editorial errors or omissions contained in this document.
The information contained in this document is subject to change without notice.

Copyrights

Brands

- ContentWise is a trademark of ContentWise Srl, a subsidiary of Moviri S.p.A.;
- Java is a US trademark of Sun Microsystems, Inc;
- Linux is a U.S. registered trademark of Linus Torvalds;
- Microsoft, Windows, Windows XP and Windows 7 are U.S. registered trademarks of Microsoft Corporation;
- Oracle is a registered US trademark of Oracle Corporation, Redwood City, California;
- UNIX is a registered trademark of The Open Group;
- Other product names cited in this guide may be trademarks or registered trademarks of their respective owners or copyright holders.

License options

Some of the features of ContentWise described in this document are only available if additional modules or solutions are present. For information about ContentWise licences, please contact our sales department at sales@contentwise.tv

Support

As a ContentWise customer, you can contact the Product Support:

- via phone at number (+39) 02.49517019 - other EU and US numbers are available, check your support contract
- via e-mail at address support@contentwise.tv
- on the Support Site at address https://support.contentwise.tv using your customer account

On-line Resources

Information about product updates and release notes, available documentation and customer training can be found at https://support.contentwise.tv and http://www.contentwise.tv

ContentWise on the Web

ContentWise activities and news can be found at the address http://www.contentwise.tv.
For inquiries on ContentWise, please send an e-mail to info@contentwise.tv

ContentWise is a subsidiary of Moviri. Moviri activities and news can be found at http://www.moviri.com
For inquiries on Moviri, please send an e-mail to info@moviri.com

Reference Guide

Purpose of this guide

This document aims to describe the fundamental concepts and the administration activities of **ContentWise**^{TM} (from now on, in brief, ContentWise).

Please refer to:

- Installation Guide for installation requirements and how to install the product.
- API Guide for ContentWise APIs documentation.

Notations and Conventions

The following notations and conventions are used throughout this document:

> **Key Concept**
> It denotes a key concept, primarily needed to understand and correctly use ContentWise

> **Important Note**
> It is an important note

> **Warning**
> It is an alert about the provided information

External references

Third party documents referenced in this document:

Document	Source Reference
CableLabs VOD Metadata - VOD Content Specification 2.0	http://www.cablelabs.com Document ID: MD-SP-VOD-CONTENT2.0-I02-070105

Acronyms and abbreviations

List of acronyms and abbreviations used in this document:

Acronym	Definition
IP	Internet Protocol
TCP	Transmission Control Protocol
IPTV	Television over IP

VOD	Video On Demand
J2EE	Java 2 Enterprise Edition
EJB	Enterprise Java Bean
SOAP	Simple Object Access Protocol
API	Application Programming Interface
MPI	Message Passing Interface

Introduction

Product Overview

ContentWise is the Digital Media, OTT and IPTV content recommendation engine built for high-volume scalability. ContentWise produces personalized recommendations that increase content consumption and demand while its Business Rule engine empowers operators to use recommendations to maximize the return on their content catalogs. ContentWise is built for high-volume, high-performance digital TV systems and its integration-ready architecture and partner ecosystem enable operators to deploy fast and safely.

Recommend the right content for each viewer

ContentWise helps users easily **find relevant content**, enjoy a **personalized experience** and **discover new content** based on their community of interest and friends. Built on 10 years of research, the ContentWise recommendation engine analyzes **user profiles**, **preferences** and **activity**, ingests **VOD catalogs**, **EPG** and **Linear TV programming** metadata and suggests the best content for each individual viewer. Recommendations help users discover lesser-known content, increase time spent and reduce user frustration.

Lift revenue and customer retention

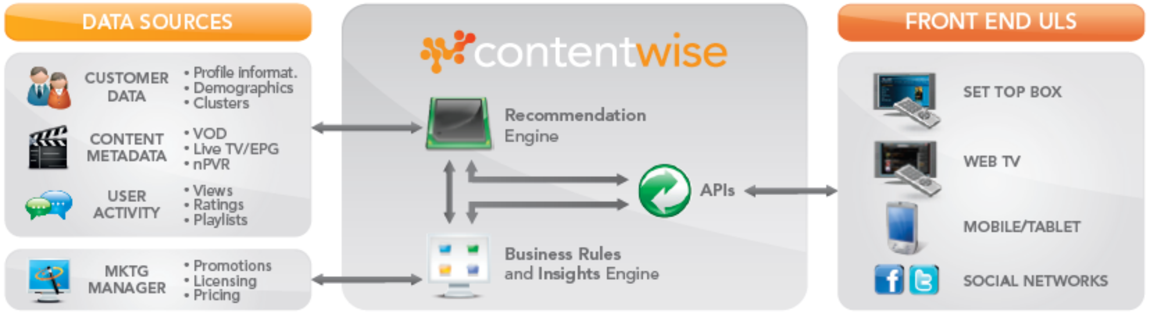
ContentWise helps operators manage **marketing KPIs**, by allowing to act on **revenue**, **profitability** and **retention** levers specific to each operator's business model. ContentWise recommendations have been proven to **increase VOD sales by 15%**. ContentWise Business Rules enable operators to control recommendation output to push selected content, **up-sell** revenue-generating content, **cross-sell** other media such as ringtones, music or games. Business Rules allow to target recommendations based on **demographic**, **pricing**, **behavioral** and **time-of-day** criteria.

Deploy to any service and scale safely

ContentWise is built for **high-volume**, **high-performance** digital TV deployments. Its integration-ready architecture and partner ecosystem enable operators to deploy fast and safely to millions of users. ContentWise works with any type of TV service such as **VOD**, **Linear TV**, **Web TV** and it delivers recommendations to **STBs**, **Connected TVs**, **mobile** and **web applications** and any media device. All services are exposed through standard APIs and ContentWise can be deployed as a cloud service (SaaS).

How it works

ContentWise is a backend engine that monitors user interactions across multiple channels and produces personalized recommendations on any application user interface or device. Operators can monitor, control and pilot recommendations and promotions with Business Rules and Insights to drive the KPIs that are specific to the operator business model.



ContentWise - How it works

Core features

Recommendation approaches

ContentWise recommendation styles are the result of a blend of optimized semantic and collaborative algorithms:

- **Item-to-item** recommendations: based on similarity of metadata such as genre, cast, director, release year or based on the semantic context.
- **Personalized** recommendations: based on the individual user profile information, such as demographics, and activity, such as viewing history and implicit and explicit rating data.
- **Social** recommendations: based on "People who watched this, also watched..." algorithms, on the activity of clusters of similar users and on the analysis of networks of followers or social networking friends.

Business rules engine

The Business Rule engine allows operators to run marketing campaigns and promotions. It can filter content in and out of recommendations results, push selected content and alter the order of recommended items, based on demographic, pricing, behavioral and time-of-day criteria.

Reporting

ContentWise enable operators to monitor the performance of “natural” recommendations and the effectiveness of Business Rules against KPIs and business targets. ContentWise provides out-of-the-box reports and the ability to create and generate custom reports.

Social network integration

ContentWise supports a social network integration framework that supports both follower-based and friend-based social networks.

Multi-service support

ContentWise monitors user interactions across multiple services such as VOD catalogs, Web OTT services, Live TV channels, DVRs and Media Centers.

Business benefits

Higher volume and profitability

ContentWise allows operators to increase customer purchase rate, attract new customers and uncover new revenue streams through up-/cross-selling opportunities. ContentWise business rules and marketing campaign management enables operators to set specific business goals and gives them the tools to push to users the most profitable, best-selling content.

Better user experience and higher customer retention

ContentWise improves the quality of the service delivered to viewers, improving and personalizing the end-user experience and making content discovery easy and comfortable. As a result, ContentWise reduces customer churn rate, while strengthening customer loyalty.

Fast return on investment

ContentWise ease of deployment and rapid integration guarantees a fast deployment and shortened payback period on the investment. ContentWise ships with native support of a large set of commercial data sources, both in the installed, enterprise version and the software-as-a-service (SaaS) version.

Strategic decision support

ContentWise provides marketers with key information regarding viewing and consumption trends, their promotion performance and their marketing campaign effectiveness, helping them to refine and adapt their business strategy.

Terminology

This chapter introduces some important concepts used in the ContentWise API and system.

- Business Rule
- Caller
- Caller Group
- ContentWise License
- Context
- Event
- Fallback strategy
- Item
  - Item hierarchies
  - Program item
- Layout
- Metadata
- Profile
- Provider and Service
- Recommendation
- Statistic
- Subdomain

    - Subdomain Rule
- Testing and experiments
- User
    - User Explicit Preference
- User Group
- Time diversity
- Dynamic stream

Business Rule

Definition

> **Key Concept**
> *Business Rule*: it is a statement applied during the system recommendation building phase in order to influence the content of a result list produced by the system, such as a recommendation.

> **Important Note**
> *Subdomain rule* and *Business Rule* are two different concepts. *Subdomain Rules* are used to define the characteristics of a subdomain, while *Business Rules* are used to customize the content of the result list.

There are four types of business rules:

- **Filter Rule**: allows to filter the items in the result list.
- **Push Rule**: pushes into the result list the items that satisfy a defined condition.
- **Update Rule**: changes the order of items in the result list, moving to the top or to the bottom the items that satisfy a defined condition.
- **Balance Rule**: allows to influence the homogeneity/heterogeneity of the items of a result list (e.g., a balance business rule can "avoids recommendations to have all items of the same genre").

A **Business Rule is always bound to a subdomain**. In addition, it can be set as active for all callers or only for a specific subset. The subset of callers can be identified by using caller groups.

> **Important Note**
> Business rules can be bound to SINGLEDOMAIN subdomains only.
> CROSSDOMAIN subdomains cannot be bound to any business rules; since they are a union of SINGLEDOMAIN subdomains, as a result business rules applied to CROSSDOMAIN are those configured to each SINGLEDOMAIN member.

> **Important Note**
> A subdomain must be configured to support business rules.
> In order to define business rules on a subdomain, its configuration must be edited and the following properties have to be configured:
>
> - List of metadata for business rules: the list of metadata that have to be available to business rules engine. These are the metadata that can be used to define business rules.
> - List of languages for business rules: the list of metadata languages that have to be available to business rules engine.

Business Rule Application

The set of business rules that can be defined depends on the API subcategory that is being used (see API Guide for the list of methods of each API subcategory.)

The following table shows which types of Business Rules are applied by the recommendation and search generation process.

API Subcategory	Filter Rule	Push Rule	Update Rule	Balance Rule	
Recommendation	X	X	X	X	
Item recommendations	X				*
Related items	X				*
Also viewed items	X				*
Top Rated	X				*

Top Viewed	X				*
Most recent	X				*
Programs HOT Live					
Programs NOW Live					
User group recommendation					
Advanced Search	X	X			**
Smart Search	X	X			**

* Default behavior can be modified by forcing the application of all business rule types. The behavior can be overridden by specifying *forceBR=true* as realtime caller parameter.
** Only on fields that have been configured to correctly manage lowercase values

Business Rules for Search APIs
Push rules prioritize the configured sets of items **if and only if they are part of the set of content matching user search criteria**

Business rules are applied on Advanced Search and Smart Search with the following limitation:

- Target evaluating user profile (counting actions performed on items having certain conditions) is not applied.
- Item context cannot be used to define rule actions
- Item positioning is not guaranteed
- Validation policy selection is limited to *Strict* (i.e. Lenient is not applicable)

Business rule - Target application

The following table shows which target conditions are supported by each ContentWise API subcategory

API Subcategory	Explicit preferences	User profile condition	User metadata
Recommendation	X	X	X
Item recommendations	X	X	X
Related items		X	X
Also viewed items		X	X
Static	X	X	X
Advanced Search	X		X
Smart Search	X		X

Rule Application Order

The business rules are applied in an order that depends on the their types. The application order is:

1. Filter rule
2. Push rule
3. Balance
4. Update

The application order of defined rules is important because it could affect the final result; i.e. the push rules are applied after the filtering ones: so it is not possible to push already filtered items.

Balance are applied after the push rules, so if the selected pushed items do not respect the balance rules, the final results could not contains the number of desired pushed items.

On Filter and Push rules, it is possible to enable the option "High priority".
A "High priority" filter rule:

- prevails over push rule

A "High priority" push rule:

- overcomes filter rules that are not high priority
- can push items already consumed by the user

Business Rule Format

Business Rules are defined in a XML format with the following structure:

```
<rules>
 <rule order="RULE_ORDER" type="RULE_TYPE" level="RULE_LEVEL" lang="RULE_LANG" strengthPolicy=
"RULE_STRENGTH">
 <!-- rule criteria goes here -->
 </rule>
</rules>
```

The element <*rule*> identifies a single rule definition.
Each rule definition requires the following attributes:

- *order* identifies the rule execution order. Currently not used.
- *type* identifies the type of the rule, according to defined business rule types. Allowed values are:
  - FILTER, for Filter Rules.
  - PUSH, for Push Rules.
  - UPDATE, for Update Rules.
  - BALANCE, for Balance Rules.
- *level* identifies the policy to adopt in validating a rule when metadata information is missing. Valid values are:
  - LENIENT: a lenient level rule is applied even if the metadata to check has no value on item to be checked.
    Example: let us consider a business rule that filters from recommendation movies with censure VM18 for users that have expressed a censure for VM18. With a lenient level, movies with no value for metadata Censure will be recommended to users with censure VM18.
  - STRICT: a strict level applies a strict validation for the rule. The rule is valid only if both itemValues and checkValues are valid.
    Example: let us consider a business rule that rule filters from recommendation movies with censure VM18 for users that have expressed a censure for VM18. With a strict level, movies with no value for metadata Censure are NOT recommended to users with censure VM18, because the system is not able to exclude the possibility that the movie with no censure information is a VM18 movie.
- *lang* identifies the language used during rule validation.
- *strengthPolicy* identifies, for filter and push rules, the priority of the rule. See *Rule Application Order*
  - STRONG: rule is high priority
  - WEAK: rule is not high priority

Each rule definition requires the element SCOPE, that defines the application scope of the rule. Possible values:

- RECOMMENDATION: the rule is applied to recommendations API.
- SEARCH: the rule is applied to search API. Update and Balance rules cannot be applied to search API.

The structure of rule criteria definition depends on the type of business rule and it is detailed in the following sections.

As a general indication, each rule may specify:

- A <*select*> element
- A <*condition*> element

<*select*> element

A <*select*> element specifies a rule validation criteria, according to one of the following types:

- Values: to specify a list of values that are used to validate the rule criteria.

```
<select type="VALUES">
 <value>MyValue1</value>
 <value>MyValue2</value>
</select>
```

- Item: to specify an item metadata whose values are used to validate the rule criteria.

```
<select type="ITEM">
 <metadata>MDNAME</metadata>
</select>
```

- Item context: to specify an item context metadata whose values are used to validate the rule criteria. Item context is provided at real time, as an API parameter.

```
<select type="ITEM_CONTEXT">
 <metadata>MDNAME</metadata>
</select>
```

- User: to specify a user metadata whose values are used to validate the rule criteria.

```
<select type="USER">
 <metadata>MDNAME</metadata>
</select>
```

- User Preference: to specify a user preference metadata whose values are used to validate the rule criteria.

```
<select type="USER_PREFERENCE">
 <metadata>MDNAME</metadata>
 <preferenceType>PREFERENCE_TYPE</preferenceType>
</select>
```

  Valid values for PREFERENCE_TYPE are:

  - LIKE: to consider positive preferences (e.g., preferred actors)
  - DISLIKE: to consider negative preferences (e.g., disliked actors)

- None: Used when the rule validation criteria is not related with specific values or metadata.

```
<select type="NONE">
</select>
```

<condition> element

A *<condition>* element specifies a precondition for the application of the rule. The rule is applied only if the condition is true for the target user.
For instance you may want to promote a specific item (by means of a Push Rule) only to users having a certain profile (e.g., only to kids or only to users that have watched more than a specified number of action movies). There are three types of conditions:

- *User*: analyzes a user metadata values to check the condition.

```
<condition type="USER" lang="COND_LANG">
 <metadata>MDNAME</metadata>
 <select type="VALUES">
  <value>VALUE1</value>
  ...
  <value>VALUEN</value>
 </select>
</condition>
```

- *User Preference* : analyzes the user preference metadata values to check the condition

```
<condition type="USER_PREFERENCE" lang="LANGUAGE">
 <metadata>MDNAME</metadata>
 <preferenceType>PREFERENCE_TYPE</preferenceType>
 <select type="VALUES">
  <value>VALUE1</value>
  <value>VALUE2</value>
  ...
  <value>VALUEN</value>
 </select>
</condition>
```

  The element *<metadata>* specifies the user Preference metadata MDNAME on which the condition is applied.

For a User Preference Condition the metadata must be a user preference metadata. See ContentWise Reference Guide for the list of metadata.
The <*select*> identifies the filter condition values to apply on MDNAME.
The element <*preferenceType*> specifies the type of preference. Valid values for *PREFERENCE_TYPE* are:

  - *LIKE*: to consider positive preferences (e.g. preferred actors)
  - *DISLIKE*: to consider negative preferences (e.g. disliked actors)
- *User Ratings*: analyzes the ratings of the user to check the condition.

```
<condition type="USERRATINGS" lang="COND_LANG">
 <metadata>MDNAME</metadata>
 <select type="SELECT_TYPE">
  <!-- select condition goes here -->
 </select>
 <quantity>COND_QTY</quantity>
 <rexoperator>COND_OPERATOR</rexoperator>
</condition>
```

The <*metadata*> element specifies the item metadata MDNAME on which the condition is applied.
The <*select*> identifies the filter condition values to apply on MDNAME.
The <*quantity*> and <*rexoperator*> specify the number of ratings of the users that must satisfy the condition. Valid values for <rexoperator> are GT (greater than), LT (lower than) and NO (never watched).

Filter Rule

A Filter Rule can be used to filter the items of the result list, including only items that satisfy a specified condition.

Editorial Filter Rule

An Editorial Filter Rule is used to include into the result list only the items that belong (or do not belong) to a given editorial list.

> **Editorial list**
> An editorial list is a list of content editorially defined.

It has the following XML format:

```
<rule order="1" type="FILTER" level="RULE_LEVEL" lang="RULE_LANG" strengthPolicy="RULE_STRENGTH"
>
 <scope>RULE_SCOPE</scope>
 <staticListName>LISTNAME</staticListName>
 <operator>RULE_OPERATOR</operator>
 <condition type="COND_TYPE" lang="COND_LANG">
  <!-- rule condition definition goes here -->
 </condition>
</rule>
```

- The <*staticListName*> element specifies the editorial list to be evaluated.
- The <*operator*> element specify how the filter rule works. Operator can be: IN, NOTIN.
  - IN: include into the result list only the items that belong to the specified editorial list.
  - NOTIN: include into the result list only the items that do not belong to the specified editorial list.
- The <*condition*> element allows to specify a precondition for the application of the rule. The rule is applied only if the condition is true for the target user.

By-metadata Filter Rule

A By-metadata filter Rule is used to include into the result list only the items that satisfy a specified condition defined on item metadata.

It has the following XML format:

```
<rule order="1" type="FILTER" level="RULE_LEVEL" lang="RULE_LANG" strengthPolicy="RULE_STRENGTH"
>
 <scope>RULE_SCOPE</scope>
 <metadata>MDNAME</metadata>
 <operator>RULE_OPERATOR</operator>
 <select type="SELECT_TYPE">
  <!-- select condition goes here -->
 </select>
 <condition type="COND_TYPE" lang="COND_LANG">
  <!-- rule condition definition goes here -->
 </condition>
</rule>
```

- The <*metadata*> element specifies the item metadata MDNAME on which the filter rule acts.
- The <*operator*> element defines the operator applied by the rule on MDNAME values to check if they are valid or not. Operator can be: IN, NOTIN.
- The <*select*> element identifies the filter to apply.
- The <*condition*> element allows to specify a precondition for the application of the rule. The rule is applied only if the condition is true for the *target user*.

Push Rule

A Push Rule allows to push in recommendation a set of items that satisfy a defined criteria. You can defined either an Editorial Push Rule or a By-metadata Push Rule.

Editorial Push Rule

An Editorial Push Rule allows to push in recommendation list a defined editorial list.

> **Editorial list**
> An editorial list is a list of content editorially defined.

It has the following XML format:

```
<rule order="1" type="PUSH" level="RULE_LEVEL" lang="RULE_LANG" strengthPolicy="RULE_STRENGTH">
 <scope>RULE_SCOPE</scope>
 <staticListName>LISTNAME</staticListName>
 <numItemToPush>NUM</numItemToPush>

 <pushSelectPolicy>PUSH_SELECT_POLICY</pushSelectPolicy>
 <pushInsertPolicy>PUSH_INSERT_POLICY</pushInsertPolicy>
 <pushProfileInfluence>PUSH_PROFILE_INFLUNCE</pushProfileInfluence>

 <condition type="COND_TYPE" lang="COND_LANG">
  <!-- rule condition definition goes here -->
 </condition>
</rule>
```

- The <*staticListName*> element specifies the editorial list from which items to recommend are extracted.
- The <*numItemToPush*> element specifies the number of items to push (the items ordering of the list is maintained).
- The <*pushSelectPolicy*> element specifies how items to be pushed are selected from editorial list. Valid values are:
    - TOP: select the first items of the editorial list.
    - RANDOM: randomly select the items from the editorial list.
    - USER: select the items of the editorial list according to user profile.
- The <*pushInsertPolicy*> element specifies how the items to be pushed are inserted into the result list. Valid values are:
    - TOP: push the items as the first items of the result list.
    - LIKELIHOOD: the positions of the items to be pushed are determined according to user profile.
- The <*pushProfileInfluence*> element, specified only if <*pushInsertPolicy*> is LIKELIHOOD, determines how much the user profile influences the pushing of the items. Lower the influence, greater the probability of pushing the content in the top of the list. Greater the influence, greater the probability that only items that fit the user profile are pushed.
- The <*condition*> element allows to specify a precondition for the application of the rule. The rule is applied only if the condition is true for the target user.

By-metadata Push Rule

A By-metadata Push Rule pushes in recommendation list a specified number of items that are extracted among all the items that satisfy a given set of conditions.

```
<rule order="1" type="PUSH" level="RULE_LEVEL" lang="RULE_LANG" strengthPolicy="RULE_STRENGTH">
 <scope>RULE_SCOPE</scope>
 <numItemToPush>NUM</numItemToPush>
 <pushSelectPolicy>PUSH_SELECT_POLICY</pushSelectPolicy>
 <pushInsertPolicy>PUSH_INSERT_POLICY</pushInsertPolicy>
 <pushProfileInfluence>PUSH_PROFILE_INFLUNCE</pushProfileInfluence>
 <metadata>MDNAME</metadata>
 <operator>RULE_OPERATOR</operator>
 <select type="SELECT_TYPE">
  <!-- select condition goes here -->
 </select>
 <condition type="COND_TYPE" lang="COND_LANG">
  <!-- rule condition definition goes here -->
 </condition>
</rule>
```

- The <*numItemToPush*> element specifies the number of items to push.
- The <*pushSelectPolicy*> element specifies how items to be pushed are selected from editorial list. Valid values are:
  - TOP: select the first items of the editorial list.
  - RANDOM: randomly select the items from the editorial list.
  - USER: select the items of the editorial list according to user profile.
- The <*pushInsertPolicy*> element specifies how the items to be pushed are inserted into the result list. Valid values are:
  - TOP: push the items as the first items of the result list.
  - LIKELIHOOD: the positions of the items to be pushed are determined according to user profile.
- The <*pushProfileInfluence*> element, specified only if <*pushInsertPolicy*> is LIKELIHOOD, determines how much the user profile influences the pushing of the items. Lower the influence, greater the probability of pushing the content in the top of the list. Greater the influence, greater the probability that only items that fit the user profile are pushed.
- The <*metadata*> element specifies the item metadata MDNAME on which the push rule acts.
- The <*operator*> element defines the operator applied by the rule on MDNAME values to check if they are valid or not. Operator can be: IN, NOTIN.
- The <*select*> element identifies the condition that metadata MDNAME must satisfy in order to add the item to the list of valid items for the push operation.
- The <*condition*> element allows to specify a precondition for the application of the rule. The rule is applied only if the condition is true for the *target user*.

Update Rule

An Update Rule is used to change the order of items within a recommendation, moving either to the top or to bottom of the recommendation the items that satisfy a defined criteria.
An Update Rule has the following XML format:

```
<rule order="1" type="UPDATE" level="RULE_LEVEL" lang="RULE_LANG">
 <scope>RULE_SCOPE</scope>
 <update>UPDATE_TYPE</update>
 <metadata>MDNAME</metadata>
 <operator>RULE_OPERATOR</operator>
 <select type="SELECT_TYPE">
  <!-- select condition goes here -->
 </select>
 <condition type="COND_TYPE" lang="COND_LANG">
  <!-- rule condition definition goes here -->
 </condition>
</rule>
```

- The <*update*> element specifies how the update rule acts. It can be:
  - UP, to put elements that satisfy the condition in the top of the recommendation results.
  - DOWN, to put elements that satisfy the condition in the bottom of the recommendation.
- The <*metadata*> specifies the item metadata MDNAME on which the update rule acts.
- The <*operator*> element defines the operator applied by the rule on MDNAME values to check if they are valid or not. Operator can be: IN, NOTIN.
- The <*select*> element identifies the condition that metadata MDNAME must satisfy in order to apply the required update operation on the item.
- The <*condition*> element allows to specify a precondition for the application of the rule. The rule is applied only if the condition is true for the *target user*.

Balance Rule

A Balance Rule allows to influence the homogeneity/heterogeneity of the items of a recommendation, by specifying maximum and minimum (in percentage) presence of items with specified metadata values. For instance, it is possible to define rules that can be read as the statement "Avoid recommendations having more than 30% of results made by comedy movies".

```
<rule order="1" type="BALANCE" level="RULE_LEVEL" lang="RULE_LANG">
 <scope>RULE_SCOPE</scope>
 <metadata>MDNAME</metadata>
 <select type="SELECT_TYPE">
  <!-- select condition goes here -->
 </select>
 <operator>BALANCE_OPERATOR</operator>
 <percentage>PERCENT</percentage>
 <condition type="COND_TYPE" lang="COND_LANG">
  <!-- rule condition definition goes here -->
 </condition>
</rule>
```

- The <*metadata*> element specifies the item metadata MDNAME on which the balance rule acts.
- The <*select*> element identifies the condition that metadata MDNAME must satisfy in order to apply the required balance operation on the item.
- The <*operator*> element defines the operator applied on MDNAME values. Valid values are:
  - GT, (greater than) to define a lower bound.
  - LT, (lower than) to define an upper bound.
- The <*percentage*> element specifies the bound-value, expressed in percentage of items of the recommendation.
- The <*condition*> element allows to specify a precondition for the application of the rule. The rule is applied only if the condition is true for the *target user*.

Caller

Definition

Key Concept
Caller: A caller is a client of the Recommendation Server (e.g., a VOD portal).

It is always true that:

- A caller is uniquely identified by a *CallerID*.
- A caller is mapped to a provider.

Caller types

ContentWise defines two types of callers:

- Plain callers
- Layout-based callers

Plain caller

Key Concept
A *Plain* caller works on single recommendation sources (e.g., editorial list, static, or personalized recommendations).

A Plain caller is configured by a set of properties. The most important are reported below:

Caller property	Description
Subdomain	The default subdomain of the caller. If a different subdomain is not specified with API parameters, this subdomain will be used.
Base service	The default service the caller operate with.
Base algorithm	The default algorithm used for generating a recommendation.
Recommendation length	The default number of items to return in a recommendation.
Similar item algorithm	The algorithm applied to retrieve the "Related Items" recommendation.

Also viewed algorithm	The algorithm used to retrieve the "Also Viewed" recommendation.
Fallback strategy	The recommendation algorithm that is used as fallback, i.e., when the requested recommendation cannot be obtained.

Layout-based caller

✓ **Key Concept**
A *layout-based* caller works on layouts (the layout currently *active* can change over time according to the layout scheduling), allowing to be associated to multiple recommendation sources (e.g., editorial list, static, or personalized recommendations).

A layout-based caller is configured with a set of properties. The most important are listed below:

Caller property	Description
Base layout	The default layout which is active when no layout schedule is running.
Base service	The default service the caller operate with.
Base algorithm	The default algorithm to use for generating *personalized* recommendations (for personalized layout items).
Similar item algorithm	The algorithm applied to retrieve the "Related Items" recommendation.
Also viewed algorithm	The algorithm used to retrieve the "Also Viewed" recommendation.
Fallback strategy	The recommendation algorithm that is used as fallback, i.e., when the requested recommendation cannot be obtained.

Layout-based callers and subdomains
Differently from plain caller, layout-based callers do not have a base *subdomain*. In fact, each item of a layout has its own subdomain, and a layout-based caller always return a layout-based recommendation.
However, a set of API calls are not compliant with the layout pattern - either (i) because they do not compose their output list according to a layout, such as the search API, or (ii) because they do not return a list of items, such as the getItem API.
For such API calls - when no subdomain has been explicitly specified at real-time - the system will use the default subdomain related to the layout currently active for such layout-based caller, according to the layout scheduling. For this reason, it is recommended to specify the subdomain when requesting for an API that is not layout compliant.

Dynamic streams

The caller can be optionally enabled to operate with dynamic streams. In such a case, for each subdomain the caller is bound to, it is possible to associate one stream configuration. When the caller is requested to provide the dynamic streams for a given user on a certain subdomain, it replies with the dynamic streams generated with the selected stream configuration.

Time diversity

The caller can be configured to apply a *time diversity* strategy, implemented to control the diversification of recommendations over time. Such mechanism is tuned by means of two parameters:

- *recommendation diversity*: a decimal number that specifies the level of time diversity to force, from 0 (no time diversity) to 1 (maximum time diversity)
- *recommendation diversity refresh*: an integer number that specifies the frequency the time diversity strategy forces recommendation list to change over time

See time diversity for further details.

Caller Group

Definition

✓ **Key Concept**
Caller Group: A caller group is a set of callers.

A caller group has a type which classifies the callers. Please note that there are not specific properties depending on the type; the type is only a classification tag.

Caller Group types

ContentWise defines ten types of caller groups:

- Generic
- Music
- Mobile
- Tablet
- Android
- PC
- Game
- Windows
- Apple
- TV

ContentWise License

License definition

ContentWise license applies on three parameters:

1. *Expiration time*: the time until you are granted to use the product
2. *Number of items*: the total number of items allowed
3. *Number of users*: the total number of users allowed

If one or more of the parameters above are not respected, we have a license violation. According to the parameter exceeded, we have different behaviors.

The following of this chapter describes how a license violation impacts:

- Data import
- APIs availability

Data import and license violation

Expiration time violation

With an expired license:

- New users are not imported
- New items are not imported

> ⚠ **Warning**
> With an expired license, Stage Manager process is no more executed. The system status is maintaned but new data are discarded.

Number of items or number of users violation

When the number of users exceeds the license limit, new users are imported with a "license violated" status. These users cannot receive personalized recommendations.

When the number of items exceeds the license limit, new items are imported into the system without any exception. Recommendation models are limited to the maximum number of items allowed.
Example: new items over the license limit are still imported, but the model of each subdomain is reduced by a proportional quote of items (hence, if you insert 1100 items on a license of 1000 items, ContentWise will discard the (1100-1000)/1000 = 10% of items from each subdomain).

> ℹ **Important Note**
> Number of items and number of users violations do NOT break the Data Import Interface and the Real Time API (see Architecture and Integration), ensuring that you do not loose data due to this kind of license violation.

ContentWise APIs and license violation

ContentWise defines three different license check types, that apply to ContentWise APIs:

- FREE: No check is made on license. APIs return results even if the license is expired.
- DATE VALIDATION: Only expiration date is checked. APIs return error code 272 only if the license is expired.
- FULL VALIDATION: Both expiration date and #user is checked. APIs return error code 272 if any of these license conditions are

violated.

The list of APIs with the related check type is reported below.

API	License check type	REST API
getAccess	FREE	/cuapi/userprofile/access/{callerid}/{userid}/{service}/{itemid}
getAccesses	FREE	/cuapi/userprofile/accesses/{callerid}/{userid}
getAccessSpecifiedMetadata	FREE	/cuapi/userprofile/accesses/md/{callerid}/{userid}
getBookmarks	FREE	/cuapi/userprofile/bookmarks/{callerid}/{userid}
getChannelList	FREE	/cuapi/item/channels/{callerid}
getChannelsEPG	FREE	/cuapi/item/epg/{callerid}/{timeoffset}/{startdate}/{enddate}
getCloud	FREE	/cuapi/search/cloud/{callerid}/{metadata}
getCrossBookmarks	FREE	/cuapi/userprofile/bookmarks/cross/{callerid}/{subdomains}/{userid}
getItem	FREE	/cuapi/item/item/{callerid}/{service}/{itemid}
getItemAlsoViewed	FREE	/cuapi/recommendation/item/alsoviewed/{callerid}/{userid}/{service}/{itemid}
getItemAlsoViewedAnonymous	FREE	/cuapi/recommendation/item/alsoviewed/anonymous/{callerid}/{service}/{itemid}
getItems	FREE	/cuapi/item/items/{callerid}
getItemsEvent	FREE	/cuapi/userprofile/event/items/{callerid}/{userid}
getItemSimilar	FREE	/cuapi/recommendation/item/similar/{callerid}/{userid}/{service}/{itemid}
getItemSimilarAnonymous	FREE	/cuapi/recommendation/item/similar/anonymous/{callerid}/{service}/{itemid}
getItemsRating	FREE	/cuapi/userprofile/rating/items/{callerid}/{userid}
getItemsSpecifiedMetadata	FREE	/cuapi/item/items/md/{callerid}
getItemsSpecifiedMetadataU	FREE	/cuapi/item/items/rating/md/{callerid}/{userid}
getItemsStems	FREE	/cuapi/item/items/stems/{callerid}
getItemStems	FREE	/cuapi/item/item/stems/{callerid}/{service}/{itemid}
getItemsU	FREE	/cuapi/item/items/rating/{callerid}/{userid}
getItemU	FREE	/cuapi/item/item/rating/{callerid}/{userid}/{service}/{itemid}
getMetadataValues	FREE	/cuapi/search/mdvalues/{callerid}/{metadata}
getMostRecent	FREE	/cuapi/recommendation/mostrecent/{callerid}/{userid}
getMostRecentAnonymous	FREE	/cuapi/recommendation/mostrecent/anonymous/{callerid}
getMostRecentF	FREE	/cuapi/recommendation/mostrecent/filter/{callerid}/{userid}
getMostRecentFAnonymous	FREE	/cuapi/recommendation/mostrecent/filter/anonymous/{callerid}
getProgramsHotLive	FREE	/cuapi/recommendation/hotlive/{callerid}
getProgramsNowLive	FREE	/cuapi/recommendation/nowlive/{callerid}
getTopRated	FREE	/cuapi/recommendation/toprated/{callerid}/{userid}
getTopRatedF	FREE	/cuapi/recommendation/toprated/filter/{callerid}/{userid}
getTopRatedAnonymous	FREE	/cuapi/recommendation/toprated/anonymous/{callerid}
getTopRatedFAnonymous	FREE	/cuapi/recommendation/toprated/filter/anonymous/{callerid}")
getTopViewed	FREE	/cuapi/recommendation/topviewed/{callerid}/{userid}
getTopViewedF	FREE	/cuapi/recommendation/topviewed/filter/{callerid}/{userid}
getTopViewedAnonymous	FREE	/cuapi/recommendation/topviewed/anonymous/{callerid}
getTopViewedFAnonymous	FREE	/cuapi/recommendation/topviewed/filter/anonymous/{callerid}
getUser	FREE	/cuapi/user/user/{callerid}/{userid}

getUserExplicitPreferences	FREE	/cuapi/userprofile/preferences/explicit/{callerid}/{userid}
getUserExplicitRatings	FREE	/cuapi/userprofile/ratings/explicit/{callerid}/{userid}
getUserImplicitRatings	FREE	/cuapi/userprofile/ratings/implicit/{callerid}/{userid}
getUserRatings	FREE	/cuapi/userprofile/ratings/{callerid}/{userid}
getUsers	FREE	/cuapi/user/users/{callerid}
getUsersSpecifiedMetadata	FREE	/cuapi/user/users/md/{callerid}
lookupItems	FREE	/cuapi/item/items/lookup/md/{callerid}
removeBookmark	FREE	/cuapi/userprofile/bookmark/{callerid}/{userid}/{service}/{itemid}
removeItem	FREE	/cuapi/item/{callerid}
removeUser	FREE	/cuapi/user/{callerid}/{userid}
advancedSearchContent	DATE VALIDATION	/cuapi/search/content/advanced/{callerid}/{userid}
advancedSearchContentAnonymous	DATE VALIDATION	/cuapi/search/content/advanced/{callerid}
enhanceUserProfile	DATE VALIDATION	/cuapi/userprofile/enhanceprofile/{callerid}
getRatingsIByGroupSpecifiedMetadata	DATE VALIDATION	/cuapi/usergroup/ratingsbygroup/md/{callerid}/{grouptype}/{groupid}/{service}/{it
getUserGroup	DATE VALIDATION	/cuapi/usergroup/{callerid}/{grouptype}/{groupid}
getUsersInUserGroup	DATE VALIDATION	/cuapi/usergroup/users/{callerid}/{grouptype}/{groupid}
joinChannel	DATE VALIDATION	/cuapi/channel/join/{callerid}
joinChannelAsyn	DATE VALIDATION	/cuapi/channel/leave/{callerid}
searchUserGroups	DATE VALIDATION	/cuapi/search/usergroup/{callerid}
searchUsers	DATE VALIDATION	/cuapi/search/user/{callerid}
setBookmark	DATE VALIDATION	/cuapi/userprofile/bookmark/{callerid}/{userid}
setBookmarks	DATE VALIDATION	/cuapi/userprofile/bookmarks/{callerid}/{userid}
setEPGSchedule	DATE VALIDATION	/cuapi/item/epg/{callerid}
setItem	DATE VALIDATION	/cuapi/item/item/{callerid}
setItems	DATE VALIDATION	/cuapi/item/items/{callerid}
setItemAccess	DATE VALIDATION	/cuapi/userprofile/access/{callerid}
setItemsAccess	DATE VALIDATION	/cuapi/userprofile/accesses/{callerid}
setItemAccessAsyn	DATE VALIDATION	/cuapi/userprofile/access/asyn/{callerid}
setItemsAccessAsyn	DATE VALIDATION	/cuapi/userprofile/accesses/asyn/{callerid}
setItemRating	DATE VALIDATION	/cuapi/userprofile/rating/{callerid}

setItemsRating	DATE VALIDATION	/cuapi/userprofile/ratings/{callerid}
setItemRatingAsyn	DATE VALIDATION	/cuapi/userprofile/rating/asyn/{callerid}
setItemsRatingAsyn	DATE VALIDATION	/cuapi/userprofile/ratings/asyn/{callerid}
setUser	DATE VALIDATION	/cuapi/user/{callerid}
setUserExplicitPreferences	DATE VALIDATION	/cuapi/userprofile/preferences/explicit/{callerid}/{userid}
setUserGroup	DATE VALIDATION	/cuapi/usergroup/{callerid}
getAutocomplete	FULL VALIDATION	/cuapi/search/autocomplete/{callerid}
getAutocompleteUsingProfile	FULL VALIDATION	/cuapi/search/autocomplete/{callerid}/{userid}
getItemBasedExplanation	FULL VALIDATION	/cuapi/item/explanation/item/{callerid}/{userid}/{algoname}
getItemBasedExplanationAnonymous	FULL VALIDATION	/cuapi/item/explanation/anonymous/item/{callerid}/{algoname}
getItemRecF	FULL VALIDATION	/cuapi/recommendation/item/filter/{callerid}/{userid}/{service}/{itemid}
getItemsToEnhanceProfile	FULL VALIDATION	/cuapi/userprofile/enhanceprofile/{callerid}/{userid}
getRec	FULL VALIDATION	/cuapi/recommendation/{callerid}/{userid}
getRecF	FULL VALIDATION	/cuapi/recommendation/filter/{callerid}/{userid}
getRecForDynamicStream	FULL VALIDATION	/cuapi/recommendation/streams/{callerid}/{userid}
getRecForFeature	FULL VALIDATION	/cuapi/recommendation/forfeature/{callerid}/{userid}
getRecM	FULL VALIDATION	/cuapi/recommendation/metadata/{callerid}/{userid}
getRecMF	FULL VALIDATION	/cuapi/recommendation/filter/metadata/{callerid}/{userid}
getRecWithMetadata	FULL VALIDATION	/cuapi/recommendation/populate/{callerid}/{userid}
getStemBasedExplanation	FULL VALIDATION	/cuapi/item/explanation/stem/{callerid}/{userid}/{algoname}
getStemBasedExplanationAnonymous	FULL VALIDATION	/cuapi/item/explanation/anonymous/stem/{callerid}/{algoname}
getUserDynamicStreams	FULL VALIDATION	/cuapi/userprofile/streams/{callerid}/{userid}
getUserFeatures	FULL VALIDATION	/cuapi/userprofile/features/{callerid}/{userid}
getUserGroupRec	FULL VALIDATION	/cuapi/usergroup/recommendation/{callerid}/{userid}/{grouptype}/{groupid}
getUserImplicitPreferencesByRatings	FULL VALIDATION	/cuapi/userprofile/preferences/implicitbyratings/{callerid}/{userid}
joinUserGroup	FULL VALIDATION	/cuapi/usergroup/join/{callerid}/{userid}/{grouptype}/{groupid}
leaveUserGroup	FULL VALIDATION	/cuapi/usergroup/leave/{callerid}/{userid}/{grouptype}/{groupid}

smartSearch	FULL VALIDATION	/cuapi/search/content/smart/{callerid}/{userid}
smartSearchAnonymous	FULL VALIDATION	/cuapi/search/content/smart/anonymous/{callerid}
smartSearchAnonymousF	FULL VALIDATION	/cuapi/search/content/smart/anonymous/filter/{callerid}
smartSearchF	FULL VALIDATION	/cuapi/search/content/smart/filter/{callerid}/{userid}
smartSearchM	FULL VALIDATION	/cuapi/search/content/smart/modifier/{callerid}/{userid}

Context

Definition

A **context** represents additional information about the user session; such information can influence API results, e.g., tailoring recommendations and searches to the context of the current user.
A context is typically characterized in representing data that are limited to the current user session and are not stored in the user profile (e.g., the location or the current item being displayed).

> A context is valid within the single API call. There is no persistence of contexts.
> A context is always optional.

A context is defined by:

- a context **type**
  - **USER_INFO**: a realtime user information. e.g. User rating, User lineup
  - **ITEM_INFO**: an item information.
- a **list of** context **information**

USER_INFO context

A USER_INFO context represents a user information that is passed in the API. Each USER_INFO context information is defined by:

- **name**: a user metadata name
- **valueArray**: a list of metadata values

> **Data overriding**
> USER_INFO context information REPLACE original data related user info.
> For instance, if a user has UserCountry 'Italy' but a USER_INFO context is defined (at real time) setting UserCountry equals to 'Switzerland', only context information is considered for what it concerns the metadata 'UserCountry'. This means that business rules with condition "apply to user with UserCountry 'Italy'" will not be applied, but the system will apply business rules with condition "apply to user with UserCountry 'Switzerland'".

ITEM_INFO context

A ITEM_INFO context represents a item information that is passed in the API.
An ITEM_INFO can represent either a metadata (e.g,. GenresArray) or an item (the current item if applicable, or a generic item to be specified).
Each ITEM_INFO context information is defined by:

- **name**:
  - an item metadata name
  - the strings: *itemId*, *serviceId*, *itemtype*. These three strings are used together (within the same context) to define an item identifier to be used as context.
  - the string: *RECOMMENDATION_ITEM*. Specifies that the item to be used as context is the item passed in the recommendation. Can be used only for getItemSimilar, getItemRec, .. APIs
- **valueArray**: a list of metadata values. Note that:
  - in the case ITEM_INFO refers to the current item (i.e., name is equals to *RECOMMENDATION_ITEM*), do not specify *valueArray*.
  - in the case ITEM_INFO refers to a generic item (i.e., name is equals to *itemId*, *serviceId*, or *itemtype*) only the first value is considered.

⚠ For performance optimization it is highly recommended to specify necessary metadata within contexts. If no metadata is available, getItem action is performed to retrieve metadata values necessary to evaluate context.

Event

This section presents the following concepts:

- Rating
- Access
- Item access handling policies
  - Default Handle
  - Channel Handle
  - Program Handle
- Event retention and history events

✓ **Key Concept**
Event: An event is an interaction of a user with an item.

Events are categorized according to the type of interaction:

- **Rating**: the degree of satisfaction of a user about an item.
- **Access**: an action of the user such as a purchase, a view, a channel switch. An access can generate a rating.

Events are stored into the system for different purposes:

- User profiling: events are a fundamental information for building user profiles. Without a history of interactions, a user cannot receive recommendations tailored to his/her tastes.
- Reporting: events can be analyzed for reporting and data analysis

Due to the large amount of data that events can generate, the events retention into ContentWise is limited among time. Retention period is configurable within the system and it is specific for each item type. When an event reaches its retention period, it is removed from the system, unless it satisfies a set of conditions that have been configured to preserve such events among time.
An event that is archived but not removed, remains available within the system for reporting and for a subset of the ContentWise APIs, but will not be part of the user profile when generating recommendation models.

Rating

✓ **Key Concept**
It is the degree of satisfaction of a user about an item.

We distinguish three types of ratings:

- *Explicit*: the degree of satisfaction about an item is explicitly expressed by the user (e.g. the user gives 4 stars out of five to an item).
- *Implicit*: the degree of satisfaction is inferred from item access information. Through the configuration of *Rating Types*, it is possible to configure the policies and the rules for implicit rating calculation.
- *Estimated*: the degree of satisfaction is estimated by the system according to user behavior and history. Estimated rating is also known as *appeal*.

An explicit rating is a decimal number between 1 and 5. The values from 1 to 5 can assume different meanings, generally going from "poor" to "excellent" user satisfaction.

Moreover, ContentWise provides users with the possibility to manage previous ratings, by means of rating actions.
The *rating action values* are integers between -3 and 0 that express specific rating actions but do not express any degree of satisfaction:

- The action value 0 says that the user wants to clear the explicit ratings previously assigned to the item. The implicit ratings are preserved.
- The action value -1 says that the user has explicitly requested to the recommendation system to remove the item from his recommended item list. The item will be blacklisted and the the action will overwrite the last explicit rating (if any).
- The action value -2 says that the user wants to clear the item implicit ratings history. The explicit ratings are preserved.
- The action value -3 says that the user wants to clear the item ratings history; both explicit and implicit ratings assigned are emptied.

Access

✓ **Key Concept**
It is an action of a user such as a view, a purchase, a channel switch (e.g. the user views an item, purchases an item, watches a movie,...).

An item access may generate an *implicit rating* within the system, that is the inferred degree of satisfaction of the user about the item of the interaction.

Item access handling policies

According to the type of user and the type of item, there is a specific *item access handling policy* that specify:

- which information is inferred by the user interaction
- how the access is managed within the system

The system defines three item access handling policies:

- *Default*: used for itemtypes without a predefined handle.
- *Channel*: used for channel itemtypes.
- *Program*: used for program itemtypes.

The following table shows which item access handle is applied for each type of item:

Itemtype	Applied Handle
VIDEO_CONTENT	Default
VIDEO_CHANNEL	Channel
VIDEO_PROGRAM	Program
VIDEO_PROGRAM_CLASS	Default
AUDIO_CONTENT	Default
AUDIO_CHANNEL	Channel
AUDIO_PROGRAM	Program
AUDIO_PROGRAM_CLASS	Default
WEBPAGE_CONTENT	Default
BOOK_CONTENT	Default
GENERIC_CONTENT	Default

Each item access handle defines:

- The information required for registering the *item access* and the eventual *rating*.

  ℹ **Important note**
  The information required may vary according to how item accesses are imported into the system. Batch import requirements differ from real-time import requirements.

- The additional information derived by the recommendation system.
- The filters that are applied to the item access and that define which accesses are actually to be registered.

Default Handle

Required information
The following information is required to define an ***item access***:

- *Identifier*: Item and user unique identifiers.
- *Time information*: StartTimestamp and StartTimeOffset for the access.
  - Real-time import: If not provided the current server Time and TimeOffset are used.
  - Batch import: An error occurs if information is not provided
- *Access metadata*:
  - Implicit: a set of metadata that will be used by the implicit rating estimator.
  - Explicit: explicit rating can be provided as value of RatingExpl metadata.

The following information is required to define a ***rating***:

- *Identifier*: Item and user unique identifiers.

- *Time information*: StartTimestamp and StartTimeOffset for the rating.
    - Real-time import: If not provided the current server Time and TimeOffset are used.
    - Batch import: An error occurs if information is not provided
- *Explicit rating*:
    - Real-time import: An explicit rating must be directly provided (RatingExpl metadata is not considered).
    - Batch import: RatingExpl metadata is mandatory

Derived information
No additional information is derived.

Filtering
No filtering is applied.

Channel Handle

In addition to *item access* and *rating* common to all item handles, the channel handle defines *Channel Join and Leave*, applicable only to channels.

> **Key Concept**
> *Channel Join/Leave*: Join and Leave are a particular pair of item accesses defined only for channels:
>
> - A Join occurs when a user starts watching a channel. It requires the related timestamp.
> - A Leave occurs when a user stops watching a channel. A Leave is linked to a Join. In addition to its timestamp, it requires also the timestamp of the related join.

Required information
The following information is required to define an ***item access***:

- *Identifier*:Item and user unique identifiers.
- *Time information*:
    - StartTimestamp and StartTimeOffset for the access.
        - Real-time import: If not provided the current server Time and TimeOffset are used.
        - Batch import: An error occurs if information is not provided
    - End of the item access:
        - Real-time import: EndTimestamp and EndTimeOffset for the access or PlayTime metadata. If not provided an error will be raised.
        - Batch import: EndTimestamp and EndTimeOffset and PlayTime metadata (must be calculated by the data extractor process)
- *Access metadata*:
    - Implicit: a set of metadata that will be used by the implicit rating estimator.
    - Explicit: explicit rating could be provided using RatingExpl metadata.

The following information is required to define a ***rating***:

- *Identifier*: Item and user unique identifiers.
- *Time information*: StartTimestamp and StartTimeOffset for the access.
    - Real-time import: If not provided the current server Time and TimeOffset are used.
    - Batch import: An error occurs if information is not provided
- *Explicit rating*:
    - Real-time import: An explicit rating must be directly provided (RatingExpl metadata is not considered).
    - Batch import: RatingExpl metadata is mandatory

The following information is required to define a ***join channel action***:

> ⚠ Join action is available only with real-time import. Batch import is responsible of providing already processed accesses.

- *Identifier*: Item and user unique identifiers.
- *Time information*: StartTimestamp and StartTimeOffset for the access. If not provided the current server Time and TimeOffset are used.

The following information is required to define a ***leave channel action***:

> ⚠ Leave action is available only with real-time import. Batch import is responsible of providing already processed accesses.

- *Identifier*:Item and user unique identifiers.
- *Time information*:
    - StartTimestamp and StartTimeOffset for the access. Must be the same provided in the related Join.
    - EndTimestamp and EndTimeOffset for the access or PlayTime metadata. If not provided an error will occur.

Derived information

> ⚠ Only real-time import derives access information. With batch import, the data extractor process is responsible of providing all item access information.

- Access. *Vision algo* step of Stage Manager task is executed to derive program implicit ratings from Channel Join/Leave.
- Metadata. The following metadata will be derived if not provided:
  - TimeStampEnd (from PlayTime).
  - TimeStampEndOffset (from TimeStampStartOffset).
  - TimeStampEndUTC (from the couple TimeStampEnd-TimeStampEndOffset).
  - PlayTime (from TimeStampEnd).

Filtering
No filtering is applied.

Program Handle

Required information
The following information is required to define an ***item access***:

- *Identifier*: Item and user unique identifiers. Item identifier must be a unique epg event identifier.
- *Time information*:
  - Real-time import: Three different modes are available. If none of them is satisfied an error will occur.
    - MODE 1: EndTimestamp and EndTimeOffset for the access. StartTimestamp, StartTimeOffset, PlayTime and VisionFactor are derived from program epg.
    - MODE 2: PlayTime of the access. StartTimestamp, StartTimeOffset, EndTimestamp, EndTimeOffset and VisionFactor are derived from program epg.
    - MODE 3: VisionFactor of the access. StartTimestamp, StartTimeOffset, EndTimestamp, EndTimeOffset and PlayTime are derived from program epg.
  - Batch import: to have well formed access information, the following information should be provided:
    - StartTimestamp
    - StartTimeOffset
    - EndTimestamp
    - EndTimeOffset
    - ChannelID
    - PlayTime
    - Viewed=1
    - VisionFactor
- *Access metadata*:
  - Implicit: A set of metadata that will be used by the implicit rating estimator.
  - Explicit: Explicit Rating can be provided using RatingExpl metadata.

The following information is required to define a ***rating***:

- *Identifier*: Item and user unique identifiers. Item identifier must be a unique epg event identifier.
- *Time information*: Item identifier is used to retrieve the epg event and the related time information.
- *Explicit rating*:
  - Real-time import: An explicit rating must be directly provided (RatingExpl metadata is not considered).
  - Batch import: RatingExpl metadata is mandatory

Derived information

> ⚠ Only real-time import derives access information. With batch import, the data extractor process is responsible of providing all item access information.

- Metadata. The following metadata will be derived if not provided:
  - TimeStampEnd (from PlayTime).
  - TimeStampEndOffset (from TimeStampStartOffset).
  - TimeStampEndUTC (from TimeStampEnd and TimeStampEndOffset).
  - PlayTime (from TimeStampEnd).
  - VisionFactor (from PlayTime)

Filtering

> ⚠ Only real-time import filters provided access information. With batch import, the data extractor process is responsible of filtering item accesses according to its custom criteria.

A program vision will be registered only if a matching epg entry is found.

Program events are filtered according to the following parameters:

- Minimum PlayTime: accesses shorter than 5 minutes will not be registered (The value is configurable).
- Minimum VisionFactor: accesses shorter than 30% of program length will not be registered (The value is configurable).

Event retention and history events

As described above, the events retention into ContentWise is limited among time. A retention period is required to guarantee the best performances without limiting the quality of the recommendations and the user representation (user profile) within the system. When an event reaches its retention period, it is removed from the system, unless it satisfies a set of conditions that have been configured to preserve such events among time. These conditions are defined by means of history events.

Important note
An event that is archived but not removed, remains available within the system for reporting and for a subset of the ContentWise APIs, but it does not belong to the user profile considered when generating recommendation models.

When a history event is defined, the following information is required:

- The rating types to be considered
- The condition that the access must satisfy to be evaluated by the event
- A set of metrics to be stored for the event
- An option specifying if the event has to be part of the user profile for generating recommendation models or not.

Example
Given the following information, two scenario can occur:

- A retention period of 6 months for the itemtype VIDEO_CONTENT
- An item access of the user U for the VIDEO_CONTENT item I, having "RatingExpl = 5", occurred 6 months and 1 day ago

1. No history event is configured: the item access is archived and removed from the system. The user profile of user U no more contain the item I (as if no access occurred 6 months and 1 day ago).
2. An history event is configured to keep accesses having "RatingExpl != null": the item access no longer belongs to the user profile of user U, but the information about the access is kept by the system and it is still available for a subset of ContentWise APIs. Asking ContentWise the rating given by user U on item I, the information "RatingExpl = 5" can be returned.

Important note
Event history can also be used to store access information that do not have to be used to build recommendation models and user profiles, but need to be available for other purposes, such as reporting.

Example
Given:

- An item access of the user U for the WEB_PAGE item I, having "Accessed = 1"
- An event history rule configured with "Only history = true" for the accesses on WEB_PAGE items characterized by "Accessed = 1"
  The access is not stored as part of the user profile used by recommendation algorithms, but it is stored by the system and remains available for a subset of ContentWise APIs. Asking ContentWise the access information given by user U on item I, the information "Accessed = 1" can be returned.

Fallback strategy

Key Concept
When a recommendation cannot be calculated, fallback strategy defines the behavior of the system to guarantee that a consistent and coherent management of the result.

Fallback strategy is applied in the following cases:

- The requested algorithm is not applicable for the requested user (e.g. Collaborative recommendation for user having empty profile).
- An error occurs while generating the result.
- The result is an empty list.

Important note
Fallback strategy is defined at caller level

The configuration of the fallback strategy consists of the selection of the policy to adopt when a fallback is required. Fallback can be managed with:

- a top rated recommendation.

- a top viewed recommendation.
- a most recent recommendation.
- a static list, such as an editorial list.
- an empty result.

> **Fallback and business rules**
> Business rules are applied by the fallback strategy.

Item

Definition

> **Key Concept**
> *Item*: an item represents a specific content within your platform (e.g., a movie, a book, a TV show).

> **Key Concept**
> *SERVICEITEMID*: the SERVICEITEMID is the identifier that unequivocally identifies an item in a service.

It is always true that:

- An item has a unique identifier for each service, referred to as **SERVICEITEMID**.
- An item has a type that specifies the nature of the item.
- An item can have one or more associated metadata.

The defined item types are:

- VIDEO_CONTENT: an item with Audio and Video (e.g., movies, TV shows, videoclipse, recorded programs).
- VIDEO_CHANNEL: a TV (broadcast) channel.
- VIDEO_PROGRAM: a TV show, TV event (specific content scheduled on a TV channel in a certain time slot, e.g., BBC News-Channel5-8pm:8.30pm) scheduled to be aired on a VIDEO_CHANNEL. VIDEO_PROGRAM items sharing defined characteristics are represented by the same VIDEO_PROGRAM_CLASS.
- AUDIO_CHANNEL: a channel with audio only (e.g., Radio channel).
- AUDIO_CONTENT: an audio content, e.g., a Music Release, CD or collection.
- AUDIO_PROGRAM: an audio content scheduled to be aired on an AUDIO_CHANNEL. AUDIO_PROGRAM items sharing defined characteristics are represented by the same AUDIO_PROGRAM_CLASS.
- WEBPAGE_CONTENT: a web page.
- BOOK_CONTENT: a book or ebook content.
- GENERIC_CONTENT: items that do not belong to other categories.

According to the item types, different metadata can be specified.

Item hierarchies

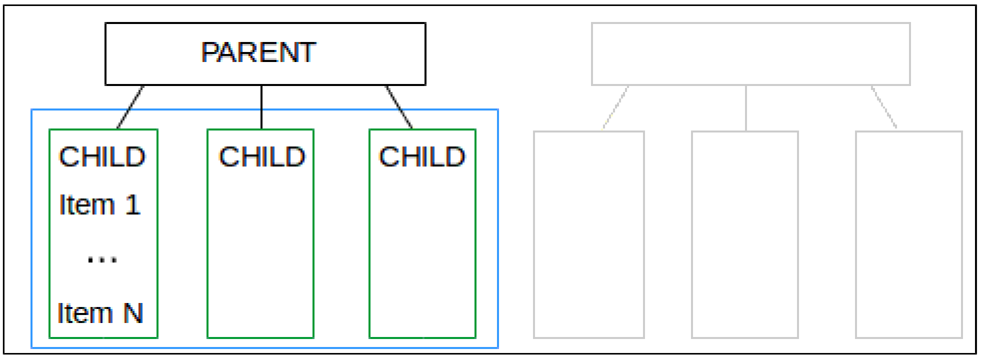
Definition

> **Key Concept**
> A hierarchy of items is a representation that models hierarchical relationships among items.

We identify two main types of relationship:

1. **Parent level**: the relationship that identifies that an item belongs to a wider entity.
   Example: an item of the catalog that is an episode of the tv series XY belongs to the wider entity "Tv series XY"
2. **Child level**: different items may represent the same entity. In this case the items belong to the same child.
   Example: both item A and item B represent the same movie. Item A is the english version, Item B is the french version. Both items can be grouped to the same child, that is the movie entity representation.

The following image represents structure and relationships of a hierarchy of items



Item hierarchy: model representation

An item hierarchy is mainly defined by the following:

- **The scope**: an item metadata and a value for which the item hierarchy is defined.
- **The parent grouping logic**: how items belong to the same parent (blue rectangle)
- **The filter hierarchy policy**: specify the filter policy of a series; you can filter out set with viewed items, recommend only set with viewed items or recommend everything
- **The first episode selection policy**: specify the policy to use in case of non-engaged user on a set. Available only if the child selection policy is **next** or **first**
- **The child grouping logic**: how items belong to the same child (green rectangle)
- **The child selection policy**: how ContentWise picks up the child to be recommended, once the parent to recommend has been identified
- **The item prioritization policy**: how ContentWise picks up the item to be recommended, once the child to recommend has been identified

The need of item hierarchies

Key Concept
Once imported into ContentWise, the items of a catalog may need to be managed considering logical relationships that connect each other.

The main needs of the item hierarchies are:

- Model item relationships (e.g. tv-series, season, episode relationship)
- Avoid recommending more than one item of the same parent within the same recommendation result (e.g. do not recommend two episodes of the same tv series within the same recommendation).
- Avoid recommending more than one item of the same child within the same recommendation result (e.g. do not recommend both english and french version of the same movie within the same recommendation).
- Specify the logic for picking up from the parent the child to be included in the recommendation result (e.g. recommend the next episode of a tv series).
- Specify the logic for picking up from the child the item to be included in the recommendation result (e.g. prioritize HD assets).

Item hierarchies application

Important note
It is possible to have different item hierarchy configurations available within the system. For each provider, it is possible to define different hierarchies of items. Moreover, a hierarchy of items is defined for a specific item type.

The application of the item hierarchy logic requires two steps:

- The generation of the item hierarchy model
- The real-time application of a specific item hierarchy

The generation of the item hierarchy model

Item hierarchy model needs to be computed before being available to recommendation algorithms. This computation is performed by ContentWise Scheduler, by means of specific tasks that can be included in the execution chain. These tasks are mandatory in order to apply item hierarchies logic.

The real-time application of a specific item hierarchy

At real-time, the recommendation is built according to the caller configuration. A caller specify:

- If item hierarchy has to be applied.
- The item hierarchy configuration to be applied when generating recommendation results. This is specific for each item type.

Example

TV Series and Episodes recommendations

In many cases, both SVOD and TVOD services include TV series in their catalog. Customer business logic often requires episodes (items within ContentWise) to be considered part of the same hierarchy (the tv series) for different reasons:

- TV Series are often recommended to users by suggesting episodes and, within a result set, it is usually required that no more than one episode of the same series is recommended.
- The picking up of an episode to be recommended may require ContentWise to be compliant with a specific business requirement, that, for instance, requires to recommend users the first not watched episode, or the best one among all episodes of the series.
- Once a version of an item (the episode 1 in HD format) has been recommended, do not include in the recommendation the same episode in another version.
- Once a version of an episode has been consumed by the user, do not recommend other version of the same episode.
- The user events performed on the episodes of the same tv series may need to be normalized, avoiding user profile polarization.

To setup the item hierarchy that support the above requirements, it is mandatory to identify:

- The scope: in this example "ShowType:Episode", because we are grouping together episodes of tv series (we want to recommend episodes).
- The parent grouping logic: a set of item metadata that allows ContentWise to identify that two episodes belong to the same tv series. It can be the SeriesTitleOriginal attribute.
- The child grouping logic: a set of item metadata that allows ContentWise to identify that two items are different versions of the same episode. It can be the combination of SeasonId and EpisodeId attributes.
- The child selection policy: once ContentWise has identified that a given tv series has to be recommended to the user, which episode should be recommended? Possible options are:
  - Recommend the best episode for the user
  - Recommend the first not watched episode
  - Recommend the next episode
- The item prioritization policy: once the episode to recommend has been identified, which version should be recommended? It is possible to select a set of item metadata and, for each of them, identify a sorting that defines the priority of the versions. E.g. HD version (Format item attribute) must have higher priority than SD version.

Program item

> ✓ **Key Concept**
> *Program item*: the ContentWise representation of a program, such as a live scheduled event. According to the domain of the event, we have two item types that represent programs:
>
> - VIDEO_PROGRAM: a live tv scheduled event. E.g. The episode X of the TV Series Y that goes on air at a scheduled timestamp of a specific tv channel.
> - AUDIO_PROGRAM: a live radio scheduled event.

This section defines the basic concept regarding the management of programs within ContentWise. The following refers to VIDEO_PROGRAM items. Every concept applies in the same way also to AUDIO_PROGRAM.

Program and program class

A live scheduled event, such as an episode of a tv series scheduled on a given channel in a specific date/time, is represented by a VIDEO_PROGRAM item. A VIDEO_PROGRAM is therefore usually characterized by:

- The program
- The channel on which the program is scheduled
- The scheduled air time

The same program, such as the same episode of a tv series, may go on air different times within the EPG. To avoid recommending same content multiple times and to properly manage user profiles, ContentWise introduces the concept of program class.

> ✓ **Key Concept**
> *Program class*: the ContentWise representation of a program regardless the schedule information (air time and channel). A program class groups together different occurrences of the same program.

A program class is usually never returned or received by ContentWise APIs, but it is used only internally for generating recommendation models and managing user profiles. In few cases, it may be necessary to build results (such as a search API result) with program classes identifiers instead of program identifiers. These cases are eventually defined during ContentWise integration projects, according to client requirements.

Define a program class

> **Warning**
> The definition of how ContentWise identifies programs that belong to the same program class is a critical task, due to its impacts on recommendation models and user profiles.

When integrating client data with ContentWise, the following aspects must be considered to define program classes:

- How are programs identified within the system?
- Which program attributes allow to identify that two programs belong to the same program class?
- How should program classes be identified?
- How should program attributes be propagated to program classes?

After requirements have been defined, ContentWise allows to configure program classes. See ContentWise Portal/Item types configuration for details.

Program set

> **Key concept**
> Program set, often referred to equality set or equset, is the set of rules that identify different programs belonging to the same set. Differently from program class, the equality set defines relations between different programs.

The most common example of equality set usage is the relation within all the episodes of the same tv series, where a metadata identifies the episode number within the series (e.g. S01E02 says Episode 2 of Series 1)

ContentWise allows to define how programs belonging to the same equality set should be managed by recommendation APIs. For instance, it is possibile to configure a caller in order to satisfy the use case "Recommend the next episode of a tv series that the user is watching".

Time windows for programs recommendation

Time information is very relevant for programs management and program recommendations delivery. Since a recommendation (or a search) is requested for a particular time period, only programs that are on air in the required period are candidates to be returned by the API.

Configuration of time windows (aka Freshness) is therefore required by ContentWise to properly delivery program recommendations that satisfy client requirements. For this reason, time windows must be defined during ContentWise integration activities.

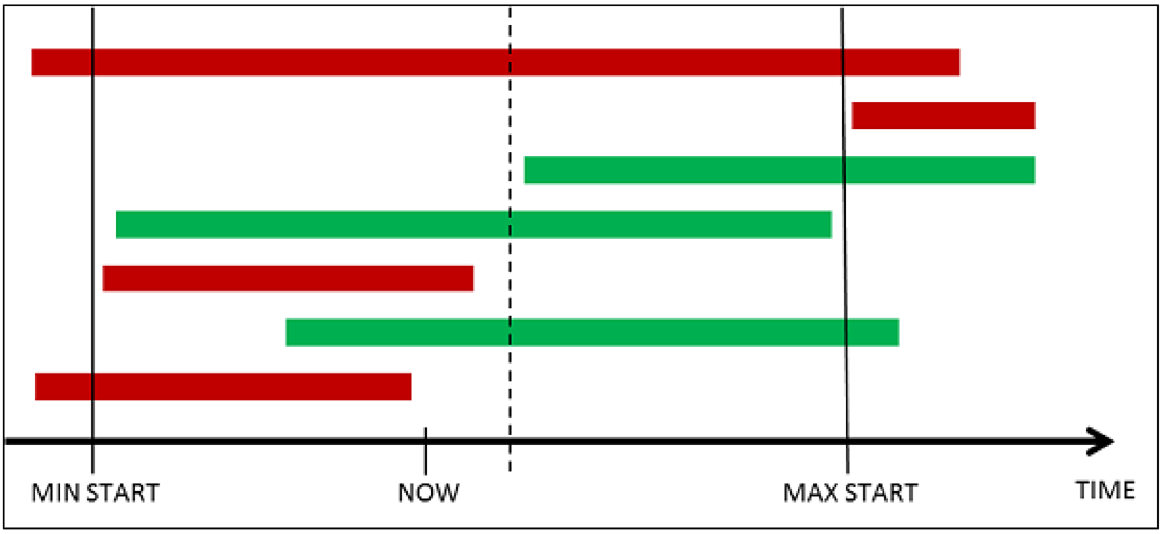
Each time window is characterized by the following information:

- Minimum start time (left bound)
- Maximum start time (right bound)
- Minimum remaining time

Given the current timestamp and a time window, the recommendation returns only programs that go on air between minimum start time and maximum start time. Additionally, ContentWise excludes from the recommendation results programs that end in less than the minimum remaining time.

> **Warning**
> The desired time window must be specified by the API. If the specified time window is not properly configured within the system, the recommendation cannot be provided. A fallback recommendation is provided.

The figure below reports some examples. Green programs can be recommended, red programs are excluded from recommendation result due to their start/end that do not comply with configured time window.



Time windows - Program recommendation examples

Layout

Definition

Key Concept
Layout: it is a position-based pattern used to compose recommendation lists. The layout is formed by a fixed number of positions. Each position (or element) of the layout is filled with an item associated to a specified source of recommendation (e.g., a personalized recommendation, an editorial list,...).

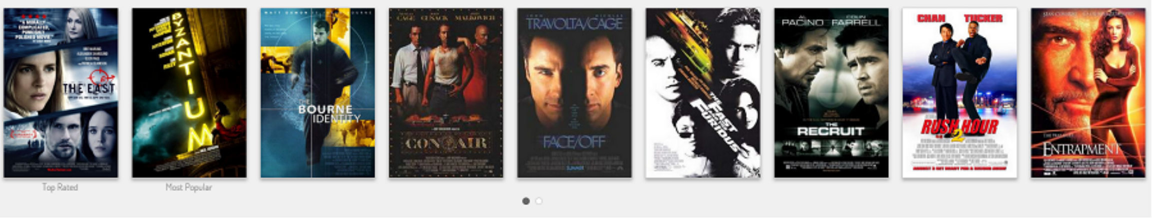
A layout is bound to a *subdomain*.

CROSSDOMAIN subdomain
In the case a layout is bound to a CROSSDOMAIN subdomain, each single position must be bound to one (SINGLEDOMAIN) member of such subdomain.

A layout is composed by a fixed number of items to recommend to the target user. For each position of the layout, you can configure the source of recommendation, choosing among:

- *Personalized recommendation algorithm*. Items are selected by a personalized algorithm that targets the items on the basis of the user profile. The algorithm applied is the one specified at real-time, or, when not specified, is the *caller* default algorithm. In addition, you can specify a set of filter business rules to apply.
- *Editorial list*. Items are selected from an editorial list. You can configure the selection policy, e.g., randomly or targeting the user preferences.
- *Static recommendations*, i.e., top rated, top viewed, and most recent. In addition, you can specify a set of filter business rules to apply.

You can specify a <key,value> *highlighter* for each position of the layout. As an example, the client requesting the recommendation can use this information to recognize the recommendation source of each item returned by the system.



An example of layout

Business rules
If rules can be configured (e.g. for personalized content) **only** filter rules are available.

Usage

This chapter describes how layouts are used within the system and what it is required to provide layout-based recommendations to the end-user.

An essential requirement for providing layout-based recommendation is the layout-caller association.

> **Key Concept**
> A layout is active within the system when:
>
> - It is the base layout of a layout-based caller and no layout schedules are active for the caller at the moment the recommendation is requested.
> - It is the default layout of a layout schedule and no other layouts are active for the schedule at the moment the recommendation is requested.
> - It is the currently designated layout of the currently active layout schedule for the caller at the moment the recommendation is requested.

Caller association

Only layout-based callers can operate with layouts. See Caller page for further details about callers.

Layout scheduling

> **Key Concept**
> Layout can be scheduled among time to plan different recommendations behavior and compositions according to time.

Each layout schedule definition is characterized by:

- A set of layout-based callers the schedule has to be active for.
- A validity interval of the schedule, in terms of start/end dates.
- A weekly calendar distribution of the schedule

Metadata

Definition

> **Key Concept**
> *Metadata*: a metadata is a structured information that represents the characteristics of entities such as: users, items, accesses, and explicit preferences.

A metadata is composed by:

- a *metadata name*: it indicates which feature the metadata refers to (e.g., the "genre" of a VIDEO_CONTENT item). The metadata name is also referred to simply as *metadata*.
- a set of *metadata values*: they indicate the values of the feature for the target item (e.g., "comedy").

Profile

Definition

> **Key Concept**
> *Profile*: a profile defines a subset of accesses, ratings, and users of a subdomain, according to a set of profile rules.

> **Important Note**
> ContentWise profiling system provides three ways for profiling a subdomain:
>
> - Access time profiling: only accesses and ratings registered in the specified time interval are considered.
> - Access caller groups profiling: only accesses and ratings registered for a set of specified caller groups are considered.
> - Access contexts profiling: only accesses and ratings registered for a set of specified contexts are considered.

Important Note
ContentWise can be configurated to generate a collaborative recommendation model for each defined subdomain profile. The default behavior is to not calculate separated models. If collaborative algorithms are configured to be profile aware, also the recommendation model related to the non-profiled subdomain (i.e., the subdomain formed by the full set of users, items, accesses, and ratings) is generated.

A profile can be associated to one or more subdomains.

The profile functionality is not addressed to profile users or items. User-based or item-based *profiling* can be done using different subdomains; see the Subdomain Rules section for more details.

Time slots

User accesses and ratings (e.g., views, purchases) are profiled according to a set of rules that identify the time slots of the week.

For example, the time slots can be *morning (7-12)*, *afternoon (12-18)*, *evening (18-23)*, *night (23-7)*. According to the current time slot:

- the user will get a different recommendation, because system accesses are divided by time slot.
- a user access will be considered for the current time slot, when the recommendation model will be generated.

Caller groups

User accesses and ratings (e.g., views, purchases) are profiled according to the caller. The rules are expressed by Caller Groups.

For example, the caller groups can be *mobile devices*, *home pages*, et cetera. According to the caller, which belongs to one o more caller groups, used in the real time request:

- the user will get a different recommendation, because system accesses are divided by caller groups.
- a user access will be considered for the current caller, when the recommendation model will be generated.

Contexts

User accesses and ratings (e.g., views, purchases) are profiled according to the input contexts.

Only metadata configured in the Rating type general properties section are considered building the profiles.

Custom strings can be used to create custom profiles like ContextsArray=[*home*, *work*, *running*] or MoodArray=[*happy*, *sad*].

Provider and Service

Provider

Key Concept
Provider: it is the global superset which contains the whole set of users and items that represent your platform and that can be evaluated to generate recommendations; a provider has a unique identifier referred to as **PROVIDERID**.

A Provider is characterized by:

- a set of subdomains (e.g., the VoD library and users enabled for VoD).
- a set of callers that are interested in showing the recommendation results within the provider (e.g., the VoD portal homepage, the special VoD portal section for kids).
- a set of services to which the provider is associated.

Service

Key Concept
Service: it represents a business service of the customer (e.g., Video-On-Demand, Broadcasting TV, Personal Video Recorder)

ContentWise defines three types of services:

- Standard Service
- Derived Service
- Alias Service

Standard Service

A standard service is the basic way to represent a business service provided to the customer.

> **Important Note**
> A standard service is associated to a set of ETL processes that import data sources in the system, identifying data for the service.

Derived Service

A derived service allows to identify items - previously imported from a standard service - by an identifier different from the one used during their import.

> **Warning**
> A derived service can only be used as an output service.

> **Important Note**
> A derived service **cannot** be associated to data sources and ETL to import data in the system.

A derived service is associated to a **parent** standard service that defines the set of items to be included.

A derived service derives items from a parent standard service by configuring one or more ***Derivation Fields***, one for each item type. In fact, a *derivation field* defines how items of a given type are derived from the standard service into a derived service.

Each derivation field is defined as a couple <*itemtype,field*>, where *field* is the derivation condition for the related *itemtype* that defines how the item identifiers have to be defined for the derived service.
The *field* refers to one of the metadata previously defined for that itemtype and that you would like to use as identifier in the derived service. The default value for *field* is "COPY_ID" that specifies to use the same identifier as defined for the parent standard service.

> **Important Note**
> Only cross language metadata can be used as derivation service metadata.

Alias Service

An alias service allows to identify items - previously imported from a service - by a different identifier, referred to as *alias*. Differently from a derived service, where the new identifier is derived from existing item metadata - the new identifier of an alias service (i.e., the alias) is imported with an ETL.

> **Alias service ETL**
> The ETL used to import aliases will be bound to a certain service (e.g., a standard service), that will be referred to as **parent** service.

An alias service can work either in *output* or in *input/output* mode:

- *input/output* mode: the alias service will be used together with the alias to identify an item (e.g., the pair <alias service, alias> identifies an item)
- *output* mode: the service alias will **not** be used to identify an item, but you have to use the parent service (e.g., the pair <parent alias service, alias> identifies an item)

Recommendation

Definition

> **Key Concept**
> *Recommendation*: it is an ordered list of items that are suitable to be recommended to a user.

It is important to note that:

- A recommendation is returned to a specific caller which is performing a call.
- The recommendation returned to the caller will respect its default setting, unless otherwise specified in the API call.

Recommendation types

The types of recommendations that ContentWise manages are:

- *Generic recommendations*: this is the simplest request. For a given user identifier, you receive the recommendation tailored to that specific user profile.
- *Filtered recommendations*: when requesting recommendations for a specific user, you can configure a filter in order to modify the recommendation.
- *Recommendations based on given item*: you can request recommendations related to a specific item. (e.g., movies related to "Kill Bill", or movies watched by people that like "Indiana Jones").
- *Recommendations based on statistics*: you can request recommendations based on statistics (e.g. top viewd or top rated items) or metadata sorting (e.g. most recent items).
- *Recommendations based on given metadata*: you can specify one or more values regarding item medatata in order to polarize the recommendation process (e.g., reflecting static user preferences and/or dynamic preferences gathered while the user is browsing the portal).

Recommendation strategies

ContentWise implements two different strategies for providing recommendations to users: Discovery and Prediction.

Discovery

✓ **Key Concept**
With the discovery strategy, the recommendations are generated on the basis of the user interests in given contexts. The algorithms discover and suggest items that do not necessarily correspond to the user watching habits, but that are likely to be appealing for the user since they match his/her preferences that have been implicitly learnt by the system.

Discovery strategy is available in all contexts, for every type of catalogue and userbase. ContentWise implements several discovery algorithms, to cover a wide range of use cases and requirements. The choice of the right algorithm depends on different variables, e.g. the set of available data and the expected result.

Discovery algorithms

✓ **Key Concept**
The recommendation algorithm is the technique used by discovery strategy to calculate the recommendation.

According to the algorithm, information such users, items, accesses, ratings, and metadata is used to compute a *recommendation model*, an efficient representation of users and items that can be used for generating a recommendation.

ContentWise defines several recommendation algorithms:

- *Collaborative*: it uses social affinity, i.e., it is based on the analysis of preferences and behaviors in order to find similar users.
- *Content*: it uses content-based affinity, i.e., it is based on the analysis of the similarity of the items, by inspecting their metadata.
- *Direct Collaborative*: another version of social affinity based algorithm.
- *Direct Content*: another version of content affinity based algorithm.
- *Collaborative KNN*: another version of social affinity based algorithm.
- *Content KNN*: another version of content affinity based algorithm.
- *Hybrid*: generates hybrid recommendations, it is based on a combination of social and content affinities.
- *Top Rated*: it recommends the highest-rated items, taking into consideration also their popularity (i.e., the number of ratings). The item statistics are computed with respect to a configured time period.
- *Top Viewed*: it recommends the most viewed items. The item statistics are computed with respect to a configured time period.
- *Most Recent*: it recommends the most recent items, i.e., the items most recently imported. The item statistics are computed with respect to a configured time period.

In addition to the algorithms described above, ContentWise can provide recommendations generated as a combination of algorithms. These recommendation algorithms are:

- *Interleaved SVD Shuffle*: it uses both content and collaborative algorithms, providing a recommendation formed by mixing the independent recommendations generated by the two algorithms. Results are shuffled.
- *Interleaved SVD Half*: it uses both content and collaborative algorithms. The first half of the recommendation is composed by items resulting from the content algorithm, while the second half contains the items resulting from the collaborative algorithm.
- *Interleaved DR Shuffle*: it uses both direct content and direct collaborative algorithms, providing a recommendation formed by mixing the recommendations generated by the two algorithms. Results are shuffled.
- *Interleaved DR Half*: it uses both direct content and direct collaborative algorithms.The first half of the recommendation is composed by items resulting from the direct content algorithm, while the second half contains the items resulting from the direct collaborative algorithm.
- *Mixed*: it specifies a list of algorithms to apply when generating the recommendation. You can specify the percentage of items in the recommendation that have to be generated by each algorithm specified. For instance, you can get a recommendation made by 25% of content algorithm, 25% of direct content algorithm, and 50% of collaborative algorithm.

Prediction

> ✓ **Key Concept**
> With the prediction strategy, the recommendation list for a given user is tailored according to his/her typical context-aware consumption patterns (e.g., user affiliation to specific channels or TV series). The algorithm recommends the available content (e.g., live TV programs) that matches the user watching habits in a certain context (e.g., the day of the week, the time of day, the device).

The prediction approach is applied to linear TV, to catch-up, and to Video-on-demand services where **recurrent patterns in given contexts** are inferred and exploited to recommend the users. The **user context** represents additional information that affects the current needs of the user, such as: the time (the day of week, the time of day, Christmas,...), the device (Smart TV, smartphone, tablet, PC,...), the place (at home, at work,...), the activity (running, working, reading,...), the mood, etc.
For instance, in linear TV domains users are typically strongly affiliated to their *preferred broadcast channels in specific time slots* - e.g., a given user watches sports on Monday night while he/she prefers TV shows on Wednesday evening.
Similarly, users have specific consumption patterns in using Video-on-demand and catch-up TV services - e.g., a given user usually watches an episode of a certain TV series the evening while he/she is travelling back home by train.

Recommendation algorithms based on a prediction approach are quite conservative as they learn the user consumption patterns in order to *predict* and *anticipate* his/her behaviour in the future. On the other hand, algorithms based on a discovery approach try to infer also novel topics that are believed to be interesting for the user, and take the risk to propose also content not strictly matching the past consumption patterns.

Statistic

Definition

> ✓ **Key Concept**
> *Statistic*: it is a definition of a set of measures of items, users or events.

It is calculated by applying SQL transformations to the values of ContentWise tables, that can be considered as set of data of the statistic. Statistics can be used in ContentWise algorithms (like the TopViewed or the TopRated), to provide exports, or to calculate new user/item properties. ContentWise contains a set of out-of-the-box statistics that should never be edited without the support team help.

Statistic composition

A statistic is defined by:

- type: the type of the statistic. One of:
  - GENERIC: a generic stats calculated with data taken from your ContentWise installation.
  - ITEM: a statistic about item-data
  - USER: a statistic about user-data
- last counter: starting date for next execution. It is empty if the statistic has to consider all available data
- statistic granularity: time granularity applied when generating the statistic. It is important to specify the right granularity. If I want a daily or weekly statistic, it is pointless to use a HOUR granularity
- algorithm binding requirement: some statistics are useful only if an algorithm is using them. If so the statistic is computed only if an algorithm is using it
- periods: the aggregation periods
- SQL query definition properties and table definition: you can define how many fields are present in the statistic and how to compute them

A statistic is stored in one of the three statistic tables:

- rs_cust_stat if the statistic type is GENERIC
- rs_item_cust_stat if the statistic type is ITEM
- rs_user_cust_stat if the statistic type is USER

Statistic usage in Top Viewed/Rated

Statistics that should be used by Top Viewed/Rated must be defined according to a particular schema. Each ContentWise installation comes out with three templates of stats for Top Algos:

- BYITEM_RATING_COUNT: Used to define a custom by period Top Viewed/Rated (Example: Top Viewed last 7 days).
- ITEM_STATS_BY_METADATA: Used to define a custom by metadata with NO period Top Viewed/Rated (Example: Top Viewed by lineup. Note that metadataName placeholder should be replaced in where condition and subkey list).
- DLY_ITEM_STATS_BY_METADATA: Used to define a custom by metadata by period Top Viewed/Rated (Example: Top Viewed last 7 days by lineup. Note that metadataName placeholder should be replaced in where condition and subkey list).

Statistic Table structure

rs_cust_stat

Field	Type	Description
STATID	number	The statistic ID
SUBDOMAINID	number	The subdomain ID
PROFILEID	number	The profile ID; if the statistic is not profiled, the profile ID is equal to -1
SUBKEY	string	It is populated with the metadata value if the statistic is by metadata, otherwise it has a default empty value
TS	datetime	The timestamp of the specific statistic. It is relative to the moment the statistic has been computed and it depends on the granularity. If the granularity is DAY you will have a row for each day (midnight) for each period. If the granularity is HOUR you will have a row for each hour (minute 0) for each period
PERIOD	string	The period key: e.g. DAY,1
NUM1	double	The field 1
NUM2	double	The field 2
NUM3	double	The field 3
NUM4	double	The field 4
NUM5	double	The field 5
NUM6	double	The field 6
NUM7	double	The field 7
NUM8	double	The field 8
NUM9	double	The field 9
NUM10	double	The field 10

rs_item_cust_stat

Field	Type	Description
STATID	number	The statistic ID
ITEMID	number	The item ID
SUBDOMAINID	number	The subdomain ID
PROFILEID	number	The profile ID; if the statistic is not profiled, the profile ID is equal to -1
SUBKEY	string	It is populated with the metadata value if the statistic is by metadata, otherwise it has a default empty value
TS	datetime	The timestamp of the specific statistic. It is relative to the moment the statistic has been computed and it depends on the granularity. If the granularity is DAY you will have a row for each day (midnight) for each period. If the granularity is HOUR you will have a row for each hour (minute 0) for each period
PERIOD	string	The period key: e.g. DAY,1
NUM1	double	The field 1
NUM2	double	The field 2
NUM3	double	The field 3
NUM4	double	The field 4
NUM5	double	The field 5
NUM6	double	The field 6
NUM7	double	The field 7
NUM8	double	The field 8
NUM9	double	The field 9
NUM10	double	The field 10

rs_user_cust_stat

Field	Type	Description
STATID	number	The statistic ID
USERID	number	The user ID
SUBDOMAINID	number	The subdomain ID
PROFILEID	number	The profile ID; if the statistic is not profiled, the profile ID is equal to -1
SUBKEY	string	It is populated with the metadata value if the statistic is by metadata, otherwise it has a default empty value
TS	datetime	The timestamp of the specific statistic. It is relative to the moment the statistic has been computed and it depends on the granularity. If the granularity is DAY you will have a row for each day (midnight) for each period. If the granularity is HOUR you will have a row for each hour (minute 0) for each period
PERIOD	string	The period key: e.g. DAY,1
NUM1	double	The field 1
NUM2	double	The field 2
NUM3	double	The field 3
NUM4	double	The field 4
NUM5	double	The field 5
NUM6	double	The field 6
NUM7	double	The field 7
NUM8	double	The field 8
NUM9	double	The field 9
NUM10	double	The field 10

Subdomain

Definition

> **Key Concept**
> *Subdomain*: it is a subset of users and items.

It is always true that:

- A subdomain has a unique identifier called SUBDOMAINID.
- Users and items may belong to multiple subdomains.
- The set of items and users forming a subdomain is defined by means of a set of *subdomain rules*
- A subdomain can be profiled in order to base recommendations only a subset of users and items (see profile).

Subdomain types

ContentWise defines three types of subdomain:

- SINGLEDOMAIN subdomains.
- MULTIDOMAIN subdomains.
- AGGREGATE subdomains.

SINGLEDOMAIN subdomain

> **Key Concept**
> A *SINGLEDOMAIN* subdomain contains items or users being homogeneous for some specific characteristics.

A SINGLEDOMAIN subdomain is defined by:

- Users of given *usertype*
- Items of given *itemtype*
- A set of SQL-like rules that define:
  - the condition that items must satisfy to belong to the subdomain. E.g., A subdomain that refers only to 'disney cartoons'

can be defined by filtering on "genre=cartoon and producer=disney".
- the condition that users must satisfy to belong to the subdomain.

A SINGLEDOMAIN subdomain is associated to an AGGREGATE subdomain, referred to as **parent** subdomain.

CROSSDOMAIN subdomain

> ✓ **Key Concept**
> A *CROSSDOMAIN* subdomain is defined by the union of a set of SINGLEDOMAIN subdomains, referred to as **members** of the CROSSDOMAIN subdomain.

A CROSSDOMAIN subdomain allows to define heterogeneous subdomains, that contain items and users of different types. For instance, it is possible to define a CROSSDOMAIN subdomain that contains both VoD and Live TV items.

> ℹ **Important Note**
> A subdomain composed by heterogenous items allows to build cross-domain recommendation models.
> This enables the possibility to recommend items of a domain, taking into account user preferences on a different domain. For instance, recommend VoD items according to Live TV user accesses and ratings.

Regardless a CROSSDOMAIN subdomain is associated to an AGGREGATE subdomain, a CROSSDOMAIN subdomain is not taken into account when generating recommendation models and recommendations (see Recommendation) for the AGGREGATE subdomain.

We recommend to take into consideration the following guidelines before creating a CROSSDOMAIN subdomain:

- An overlapping of a part of items or users from the different members should be verified, in order to allow the generation of a rich cross-domain recommendation model.
- To allow the generation of content-based cross-domain recommendation model, the compatibility among items metadata that belong to different members should be verified. In fact, if no common metadata is found, content-based cross-domain recommendation will not produce any meaningful result.
- In order to apply **profiling** to CROSSDOMAIN subdomains, all subdomains composing the CROSSDOMAIN subdomain must be compatible with the <itemtype,usertype> pairs associated with the profile. In fact, a profile is defined over a set of <itemtype,usertype> pairs; a CROSSDOMAIN subdomain can be bound to a profile only if all its members match with this set of user-item types (See Profile).

AGGREGATE subdomain

> ✓ **Key Concept**
> An *AGGREGATE* subdomain is defined by a set of SINGLEDOMAIN subdomains, referred to as **children** subdomains.

Recommendation algorithms are not executed on the AGGREGATE subdomain, but on the child (i.e., SINGLEDOMAIN) subdomains it is composed of.
The recommendation coming from any child subdomain are mixed together according to a set of rules to be configured.

E.g., The AGGREGATE subdomain 'cartoons' can be the aggregation of the 'disney cartoons' SINGLEDOMAIN subdomain and the 'other cartoons' SINGLEDOMAIN subdomain.

Subdomain Rule

Definition

> ✓ **Key Concept**
> *Subdomain Rule*: it is a statement that defines the characteristics of a subdomain.

> ℹ **Important Note**
> *Subdomain rule* and *Business Rule* are two different concepts. Subdomain *rules* are used to define the characteristics of a subdomain, while *Business Rules* are used to customize the result of the recommendations.

There are two types of subdomain rules that can be defined to characterize a subdomain:

- **Filter Rule**: a rule that allows, through the definition of filters, to specify which items and users have to be part of a SINGLEDOMAIN subdomain.
- **Mix Rule**: a rule that allows to configure how child subdomains participate in the generation of an AGGREGATE subdomain recommendation.

Filter Rule

The characteristics of a SINGLEDOMAIN subdomain (i.e., the set of users and items it is composed of) are defined by means of subdomain filter rules:

- filter rules can be defined only for SINGLEDOMAIN subdomains.
- a filter rule affects the size of a subdomain.

For instance, a filter rule might be used:

- to avoid adult content to be included into a video subdomain
- to define a SINGLEDOMAIN subdomain composed by items for kids, such as cartoons.

Filter rules are defined by filters.

> ✓ **Key Concept**
> *Filter*: it is a statement that specifies a subset of items or users according to the selected operator and the specified metadata values.

A filter consists of common logical and comparison operators to be applied on one or more metadata. Filtering expressions support standard operators:

- Logical operators: AND, OR, NOT
- Comparison operators: < , > , <> , = , LIKE, IS NULL, IS NOT NULL
- Wildcards: the special character '%' indicates any string (e.g.: '%horror%')

Examples of filters:

- Item.Genre <> 'Adult' AND Item.Censure <> 'VM18'
- Item.Genre = 'Animation' OR Item.Genre = 'Cartoon' OR Item.Audience = 'Kids'

Mix Rule

An AGGREGATE subdomain, as explained in subdomain definition section, results from the union of a set of SINGLEDOMAIN subdomains.
Mix rules allow to configure how the recommendations coming from the child subdomains are mixed into the recommendation for the AGGREGATE subdomain.
The mix rules defines the *appeal weight* assigned to any SINGLEDOMAIN recommendation. Such weights specify the importance of a subdomain in composing the final AGGREGATE subdomain recommendation. The higher the appeal weight for a child subdomain, the more items coming from that subdomain will be included in the final recommendation.

Testing and experiments

Introduction

A/B testing is a statistical *tool* to test the performance of alternative recommendation settings with respect to the current configuration.
For instance, A/B testing can be used to test the performance of a business rule with a limited amount of users, before applying it to all the customer base.

Definitions

Variation

A variation is a set of recommendation settings that we want to test.
The current configuration settings form the *baseline* variation.

> Currently, only *Business-rule* are supported as recommendation settings of a specific variation, allowing to define a set of business rules to activate to treatment users.

Treatment and control group

- *control group*: set of users assigned to the baseline variation. They will receive recommendations based on the current settings.
- *treatment groups*: sets of users assigned to a variation other than the baseline one. They will receive recommendations based on the specific settings of the variation they are assigned to.

Experiment

An experiment is composed by the set of variations we want to test.
A specified percentage of users will be assigned to the control group (i.e., baseline variation), the remaining will be randomly assigned to

one of the variations.
For the sake of testing correctness, the user-variation assignment is consistent over time, i.e., with respect to a given experiment, a user will be always assigned to the same variation.

The **A/B testing** tool allows to define an *experiment*, to monitor its progress, and to show and compare the results once the experiment is finished.
The performance of each treatment group (e.g., users assigned to *variation 1*) are compared against the performance of the control group (i.e., users assigned to *baseline variation*).

Performance are typically related to business KPIs (Key Performance Indicators), such as revenue per user or average number of purchases per user.

How long to run an experiment

An experiment is active for a certain range of dates to be specified in advance; however, **experiment significance** is a fundamental factor to consider to decide whether to stop or continue the experiment.
A/B testing significance strictly depends on the amount of data collected, i.e., the number of users involved in a certain variation.

An experiment allows to compare the KPI (e.g., revenue per user) of a treatment group with respect to the KPI of the control group. Thus, you can say that there is a (positive or negative) difference between control and treatment group only if a certain amount of data has been collected, otherwise nothing can be inferred. More data (i.e., more users) allows to appreciate finer differences among control and treatment KPIs.

> ✓ **Note**
> Configuring an experiment so that only a small percentage of users is exposed to treatment reduces the risk in the case the variations should perform poorly.
> However, a smaller percentage of users assigned to treatment group lengthens the experiment duration.

> ✓ **Note**
> We recommend to run an experiment for **at least** 7 days, so to include in the test both working and non-working days.

> ✓ **Note**
> If even after you have run a test for a while (e.g., several weeks) and collected many data you do not see a statistically significant result, then it is likely that your variations do not make a big impact on your measured KPIs.

> ℹ **Confidence**
> Experiments will be run with a *significance level* equals to 5% and a *statistical power* equals to 80%.

Experiment status

> ℹ You can run **one** experiment at a time.

An experiment can be in one of the following status:

- *RUNNING*: the experiment is running
- *END*: the experiment is terminated
- *SCHEDULED*: the experiment is scheduled for a future execution
- *WAITING*: the experiment is about to start within, at most, the next 5 minutes

User

Definition

> ✓ **Key Concept**
> *User*: a user represents an abstract customer of the IPTV platform. It can identify either a single person or, more generically, a set-top-box.

It is always true that:

- The user is identified by a unique user identifier, referred to as PROVUSERID.
- A user belongs to only one provider.
- The user has a type, referred to as usertype, that specifies the nature of the user. The possible user types are:

    - PERSON: a customer.
    - TERMINAL: a hardware terminal (e.g., set-top-box).
- A user can have one or more associated metadata.

A user can belong to a group of users: see User Group.

A user can have a set of explicit preferences: see User Explicit Preference.

User Explicit Preference

Definition

> **Key Concept**
> *User Explicit Preference*: it describes the preference of a user in terms of metadata (e.g., a preference about actors).

> **Important Note**
> *User Explicit Preferences* only refers to metadata. Preferences about items are referred to as *ratings* (see Access and rating).

ContentWise considers user explicit preferences in content-based recommendation algorithms.

An explicit preference is characterized by:

- a User Preference Metadata, for which the preference has to be stored. See Metadata Reference - User Preference Metadata for the list of available User Preference Metadata.
- a value, that represents the preference (e.g., "Tom Cruise", "Action").
- a weight, that describes if the preference is positive or negative. The weight is represented by a number in the range 1-5, where 1 stands for *Dislike* and 5 means *Like*.

For each User Preference Metadata, a user can have zero or more explicit preferences.

An example of positive preference, that represents the sentence "I like the actor Tom Cruise" is:

```
User Preference Metadata: PrefActorsLastNameFirstArray
Value: Tom Cruise
Weight: 5
```

An example of negative preference, that represents the sentence "I don't like the movie genre Western" is:

```
User Preference Metadata: PrefGenresArray
Value: Western
Weight: 1
```

User Group

Definition

> **Key Concept**
> *User Group*: a user group represents a group of users of the platform.

It is always true that:

- a user group has a unique identifier.
- a user group has a type that specifies the nature of the group. The possible user group types are:
    - FRIENDS: a user group that represents the list of friends of a given user (the owner of the group).
    - COMMUNITY: a user group that represents the set of users with some common characteristics (e.g., the fans of a certain actor).
- a user group has an owner.

Time diversity

✓ **Key Concept**
Time diversity is a mechanism implemented by the recommendation service that **regulates the diversification of the recommendation list over time**.

Time diversity consists of an advanced stragegy to select the items to recommend to the user so that his/her recommendations vary over time. In fact, as an example, assuming any other factor does not change (e.g., new ratings, new items, updated recommendation models,...), it would be preferrable the user not to the receive the same recommendation every day.

The time diversity configuration allows to regulate how recommendations are differentiated over time, controlling consistency (e.g., two sequential recommendation lists are to be identical) and assuring accuracy (i.e., the recommended items are tailored to the user interests). The parameters that can be configured are:

- **diversity**. This value indicates how diverse two recommendation lists should be. It is a decimal number that ranges from 0.0 to 1.0. Intuitively, 0.0 means no time diversity is forced, 1.0 means maximum time diversity, 0.5 is a "medium" time diversity (roughly speaking, it approximately means that, on average, about half the items changes among two recommendation lists).
- **refresh period**. How frequently the recommendation list can vary over time due to the time diversity strategy. This value is expressed in *minutes* (e.g., 1440 means that the time diversity strategy diversifies the recommendations every 1440 minutes, i.e., once a day). This value controls the consistency of recommendations. Note that, regardless the configured *refreshPeriod*, recommendations can change over time for reasons other than the time diversity, e.g., the fact that the user has updated his/her profile possibly leads to different recommendations.

Dynamic stream

Definition

✓ **Key Concept**
A **stream** is a **thematic list of items that share a mutual set of attributes or an affinity with an item or a series**.

As examples:

- the list of drama movies defines a stream whose common attribute is the genre 'drama'.
- the Because You Liked Titanic defines a stream whose common affinity is their similarity to Titanic.

A stream is mainly characterized by:

- the common set of attributes or an affinity defined by a rule on an item (item similarity) or an action.
- the list of items composing the stream.

There are two types of streams:

- **Dynamic Categories**: dynamic categories streams are thematic list of items within a specific set of attributes;
- **Because You X**: because you x streams are list of items with an affinity defined by the action X

Stream source

The common set of attributes used to define a stream can be:

- **computed**: a relevance algorithm computes the stream most interesting for the users.
- **editorial**: the common set of attributes is editorially defined. It is possible to define only **Dynamic Categories** editorial streams.

✓ **Key Concept**
A **dynamic stream** is a **particular stream tailored to the current user**, i.e., defined by taking into consideration his/her preferences.

In addition to the properties derived from the stream (i.e., common attributes and composing items), a dynamic stream is also characterized, among other properties, by:

- a **user** it is related to.
- a **score** that indicates the relevance of the dynamic stream for the user.

Both the editorial and the computed dynamic streams are assigned a relevance score that allows to sort them from the most interesting to the least interesting for the user and present the top dynamic streams. Finally, a recommendation service is used to customize the items composing the streams.

Stream title

A stream is associated a **title**, i.e., a textual description of the common set of attributes or the action that triggered the stream in natural language. For instance, the text "American drama movies" is the description of a Dynamic Categories stream characterized by two attributes: genre is drama and country is USA. The text "Because You Purchased Titanic" is the description of a Because You X stream characterized by item similar to Titanic.

Name generation uses the attributes of a stream to create a human-readable text. Such text is basically composed by a sequence of blocks, each one roughly corresponding to a word (or an entity, e.g., the name/surname of an actor or the event name). The way these blocks are composed together can be fully dynamic in the stream setup.

Quality management process

Dynamic streams can be composed by any combination of attributes. In particular in the case of *computed* streams, it can be the case that:

- some streams are generated using a combinations of attributes that would be preferable not to present to the user
- for some streams it was not possible to automatically generate a name
- some streams have been assigned a name that it might be more appropriate to replace by a most effective description

For such reasons, the process implements a validation mechanism to control the quality of the computed dynamic streams. Typically, streams are generated through the following steps:

- **setup**
- **generation**
- **validation**

Main configuration properties

A **stream configuration** defines the set of properties that regulate how streams are generated for a given subdomain (e.g., the items in the VOD subdomain). A *stream configuration* is characterized by unique name (i.e., the stream configuration identifier), a type (Dynamic Categories or Because You X), and is associated to one subdomain and one or more languages.

Properties can be configured at two different levels:

- **global properties** are defined per <subdomain, language> pair, i.e., they are valid for all the *stream configurations* of a given <subdomain, language> combination
- **stream properties** are valid only for a certain language of a given stream configuration

Global properties

Global properties are applied to all stream configurations of a certain combination of <subdomain, language>. The main properties that can be set at this level concern the optimization of the name generation. In fact, name generation can be optimized by configuring some transformations on the attribute values. For instance, it can be convenient for some values to be converted from noun to adjective - e.g., from USA to American - so that a fancier name can be generated. For a given attribute (e.g., the genre), there can be defined:

- *value replacements*, where a source value is replaced with a destination value.
- *replacement rules*, where rules - among which regular expressions - are applied to transform values.

Stream properties

Stream properties are applied, in addition to global properties, only to a specific language of a given *stream configuration*:

- **status**: sheduled/unsheduled, indicating whether the "stream genereration" task (configured in task management) has to generate the streams on the basis of the configured set of properties.
- **composition**: properties that define how the stream has to be composed, in particular:
    - *attributes*: used by **Dynamic Categories** streams, it is the list of attributes that can be used to form the stream. E.g., a stream can be configured to use the attributes genre and country.
    - *events*: used by **Because You X** streams, it is the list of users' events that can be used to generate the stream. E.g, a stream can be configured starting from a Watched event or a Purchased one.
    - *name generation*: the schema to generate the stream title. The name of a stream is composed by an ordered list of textual blocks. Each block roughly corresponds to either a single word (e.g., the genre "Comedy") or an entity (e.g., the actor "Tom Cruise" or the movie title "Titanic"). The properties of the name generation allows to define which blocks can be used and their order. The text of a block is the value of an attribute of the stream (e.g., the genre "comedy", the country "USA", the actor "Tom Cruise") or the stream event.
      In addition, for Dynamic Categories configurations, one of the block, denoted by **main word** can be forced to be always present in the stream title. The *main word* can be either one of the attribute or a static text (i.e., a user-defined text). In the case the main word is based on an attribute (e.g., a stream has necessarily to be composed at least by the attribute "genre" of a movie), the streams that do not have such attribute will be set to *conflict* status (see status in stream validation), waiting for a manual user validation.
      Note that cleaning and transformation of attribute values (e.g., from "USA" to "American") can be configured in the global properties, i.e., they are not specific of a specific stream generation but they are defined for specific combinations of subdomain and language. For Dynamic Categories configuration the name generation properties are common to all the generated streams, for Because You X configuration, the name generation are unique for each configured event.
- **generation settings**: these properties configure, among the others, the following properties:
    - the minimum number of items included in a dynamic stream. For each dynamic streams the process estimates a number

of items that will be included; dynamic streams with less than this number of items will be filtered out.
  - the miminum number of streams per user.
  - the minumum and maximum number of attributes to be used within a stream (only for Dynamic Categories configurations).
  - the popoularity threshold. Streams that are assigned to a number of user less than this threshold are filtered out (only for Dynamic Categories configurations).
  - diversity settings, to compose, for a user, diverse dynamic streams:
    - the maximum percentage of items that are shared by multiple dynamic streams of the same user
    - the attributes to guarantee not too be used in more than one dynamic stream of the same user with the same value (e.g., it grants that there will not be two dynamic streams for the same user with the same genre, if this attribute has been set, only for Dynamic Categories configurations)
    - the attributes to merge in a common stream (only for Dynamic Categories configurations)
- **filters**: a set of filters to limit the items of the selected subdomain to be included in the dynamic streams
- **default streams**: the streams displayed to the user can be configured so that a certain number of them is picked up from a set of selected streams (already existing and validated), referred to as **default streams**. For instance, it can be configured that users will be displayed 10 streams each, with 5 of them chosen from a limited set of existing streams manually selected.
- **fallback streams**: some existing and validated streams can be marked as **fallback streams**. They will be used as fallback to fill the list of streams displayed to the user only in the case it was not possible generating the required number of dynamic streams.

Dynamic stream generation

Two kinds of tasks are available in the task management:

- **Stream generation**: it generates the dynamic streams, i.e., it creates the streams and associate them to the users on the basis of their profile and preferences. Only streams related to *scheduled* configurations are generated (see status in stream properties).
- **Stream name generation**: for each existing stream (either a computed by the previous task or editorial), it generates the stream title (in the case the name does not exist yet).

Stream validation

The process allows to control which streams are valid and are allowed to be displayed to the users.

A generated stream can assume one of the following **status**:

- **to_evaluate**. When a new **Dynamic Categories computed stream** is generated together with its name, it is set to "*to_evaluate*" status, i.e., it **has to be manually validated**. Note that a stream is new only in the case the same set of attributes has never been generated in the past for a certain subdomain in a specific language. As a consequence, the number of new streams to validate is likely to be small after the first iteration. Typically, a *to_evaluate* stream will be either validated (i.e., it is mark as *valid*) or discarded (i.e., it is marked as *excluded*).
- **valid**. This is the status of a stream that have been validated and accepted to be displayed or a new **Because You X** stream that is not in conflict. **Because You X streams are automatically considered valid** because they are related to users' specific actions.
- **excluded**. Streams discarded are set to this status.
- **conflict**. New streams that were not possible generate a name for are set to *conflict* status. The typical actions on a *conflict* stream are either excluding it (i.e., set to *excluded* status) or manually defining a stream title and set the stream as valid. A stream could be in a conflict state because the generated name is longer than the limit set or the configuration has been changed before the stream name generation.

Architecture and Integration

ContentWise is a fully modular platform based on open standards that make its integration in the existing infrastructure easy and effortless.

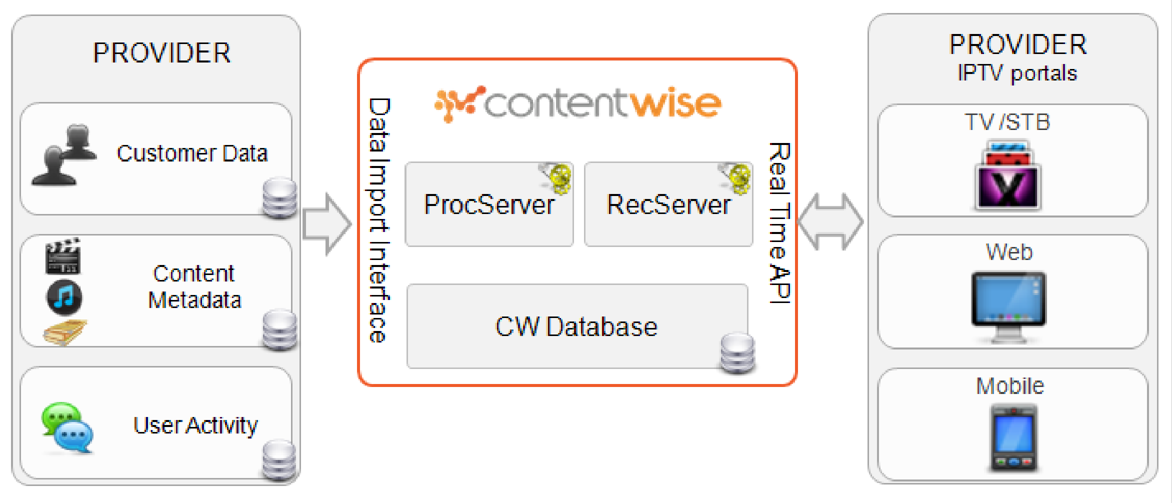
> ✓ **Key Concept**
> ContentWise provides a middleware that exposes services through standard interfaces.
> These services are available to the provider for retrieving information to be shown to its customers.

The main servers in a ContentWise installation are:

- *ProcServer*: server dedicated to offline processing. It is responsible of gathering and processing data, by providing the Data Import Interface.
- *RecServer*: server dedicated to online processing. It is responsible of providing the Real Time API interface.
- *CW Database*: A database schema containing all the information that ContentWise needs to operate correctly (ratings, items metadata, etc.).

Important Note
For high-availability (HA) purposes, ContentWise components can be deployed on different hardware machines. It is possible to deploy more than one ProcServer and more than one RecServer for offline and online processing.

The figure below represents the high level architecture of ContentWise and its integration with the IPTV infrastructure.



High level architecture and integration of ContentWise

ContentWise interacts with the provider infrastructure by means of two interfaces:

- *Data Import Interface*: the interface provided by ContentWise to import data.
- *Real Time API*: a passive listening API that is called by IPTV portals to retrieve/push data from/to ContentWise; it can be accessed via SOAP Web Service, REST Web Service or EJB calls.

Data Import Interface

Key Concept
The *Data Import Interface* is the interface provided by ContentWise to import data by means of ETL processes.

Important Note
The Data Import Interface is provided by the ProcServer component.

The Data Import Interface can deal with different types of data:

- *Customer data*: customer data such as profile information, geographics data, demographics, etc.
- *Content metadata*: information about content (e.g., title, director, duration, episode, ...).
- *User activity*: views, ratings, purchases, item accesses, ...

Data Import Interface allows to import data by means of ETL processes. See ETL for details about how ETL processes work.

See Metadata reference for details about metadata management in ContentWise.

Real Time API

Key Concept
The *Real Time API* is the ContentWise interface that provides APIs for interacting with the system at real time.

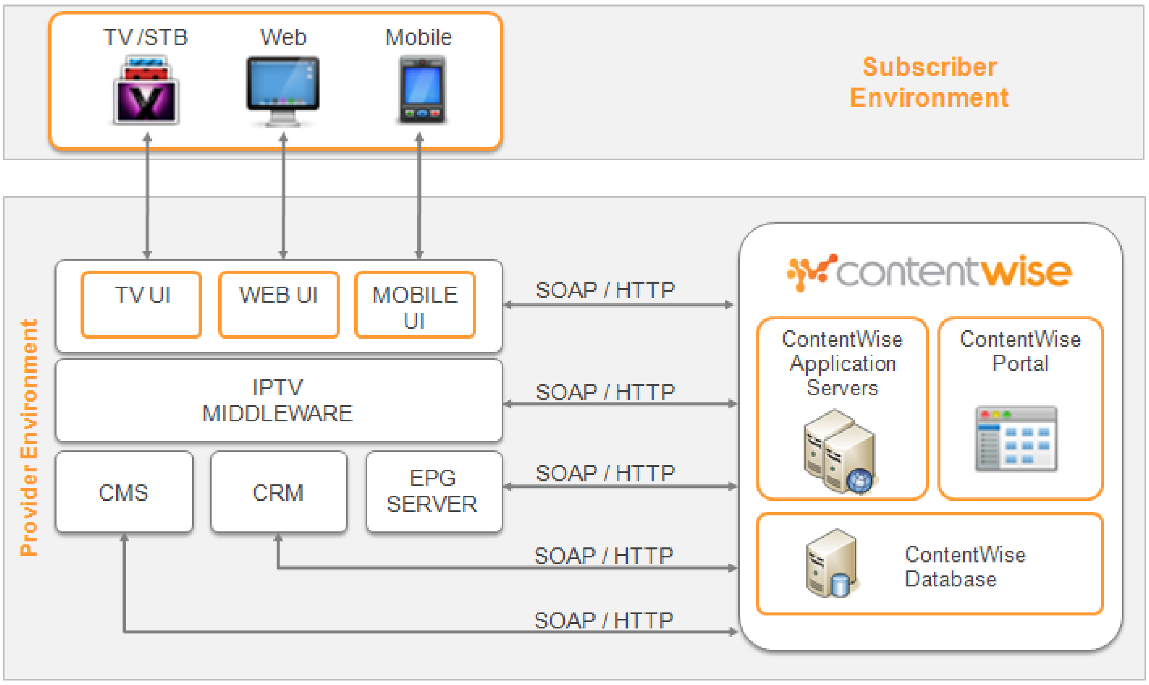
Important Note
The Real Time API interface is provided by the RecServer component.

The Real Time API consists of three APIs:

- *Frontend API* (FEAPI): provides the primitives to interact with the recommendation server. For instance 'get a recommendation', 'set a rating' and 'get the details of a content'.
- *Backend API* (BEAPI): provides the primitives that allow to import items and users into the system at realtime.

- *Management API* (MAPI): contains the primitives that allow to manage the system configuration. They are primarily used by the ContentWise Portal.

Frontend and *Backend* API are available as SOAP Web Service, REST Web Service and EJB calls.
Management API are available as SOAP Web Service and EJB calls.

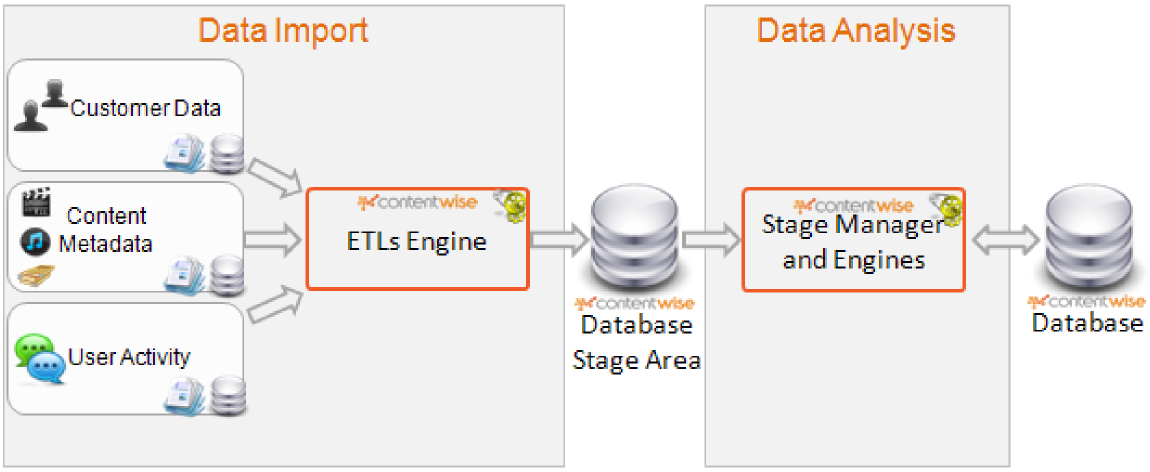


High level representation of ContentWise integration at real time

Data flow: from data import to recommendation

In order to provide recommendations, ContentWise needs data to be ingested and processed.

The diagram below shows the flow of information required in order to ingest and process data. At the end of the process, data will be available to the Real Time API interface.



Data import and analysis process

The data flow runs as follow:

1. *Data import*: ContentWise ingests data from data sources via the Data Import Interface. The Data Import Interface is implemented by ETLs, that are run by ContentWise ETL engine. The ETLs engine reads data from sources (database or files) and import them into the Stage Area of the ContentWise database.
2. *Data analysis*: gathered data are:
   a. processed by the Stage Manager task, that imports analyzed data into the ContentWise database and process data statistics.
   b. analyzed by the Engine tasks, that are the ContentWise proprietary data mining algorithms that produce recommendation models.

3. *Real-time recommendation*: At this point, recommendation models have been generated and, together with gathered data, are available to Real Time API, that is now able to answer the real time requests.

> **Important Note**
> *Data import* and *Data analysis* are offline processes, typically run once a day.

ProcServer

> **Key Concept**
> The *ProcServer* is the ContentWise server dedicated to offline processing. It is responsible of gathering and processing data.

The ProcServer manages:

- ETLs: batch processes that provide the Data Import Interface.
- Tasks: batch processes that are responsible of data processing.

The ProcServer is provided by the scheduler service.

> **Key Concept**
> The scheduler is the ContentWise service that takes care of executing ETLs and tasks at predefined intervals.

The scheduler should always be active in the system.

> **Warning**
> If the scheduler is inactive, no scheduled ETL or task will be executed. See Scheduler service to check and manage the status of the scheduler.

> **Important Note**
> ETLs and tasks need to be scheduled if a periodically execution is needed.

ETL

> An *ETL* (Extract, Transform and Load) is a batch process that extracts data from a source (e.g. a database or data files), transforms them in an appropriate format (if necessary), and then loads them into ContentWise.

> **Important Note**
> An ETL can be configured to import data for one and only one Service. Imported data will be available in the system for the Service for which the ETL has been configured.

An ETL has the following characteristics:

- it is *modular*, so that the whole process is defined by composing modules through configuration.
- it is *extensible*, so that the collection of a new data format is simple.
- it is *source independent*, i.e. all the transformation operations act on data that are represented through a standard format.
- it is *sequential*, so that only new data are loaded every time an ETL is run.

ETLs are composed of multiple modules that take care of each necessary operation:

- an *extractor* module, that connects to the source and extracts new data.
- one or more *transformer* modules, that apply in-memory transformations to data in order to convert it to the internal format.
- one or more *loader* modules, which load data into the destination.

ETLs can import:

- *Customer data* via
  - files in custom format.
  - custom SQL query over a generic JDBC datasource.
  - XML files in ContentWise XML format.
- *Content metadata* via
  - files in supported formats. See Supported formats for details about formats that are currently supported.

  - files in custom format.
  - XML files in ContentWise XML format.
  - custom SQL query over a generic JDBC datasource.
- *User activity* data via
  - files in custom format.
  - custom SQL query over a generic JDBC datasource.
  - XML files in ContentWise XML format.

> **Important Note**
> Custom ETLs can be developed to additionally import other types of data in any custom format.

ContentWise provides out-of-the-box ETLs that need to be configured according to your data sources, environment and system. See Data Import to configure and manage ETLs with the ContentWise Portal.

> **Warning**
> Editing ETL configuration without accurate knowledge of the system and of the change effects may result in failures of data import and in system malfunctions.

Datasets

> **Key Concept**
> A *dataset* is a logical structure used to define and organize in a standard format the data collected by an ETL.

An ETL populates one or more datasets, that are the output of the Extract-Transform-Load activity. The produced datasets are loaded into ContentWise.

The dataset structure is predefined:

- each dataset has a name and a numeric identifier (Dataset Id) that univocally refers to a specific structure.
- each structure details a type of data, e.g. all VoD data are defined in the *Item - Video content* (RCMITMVIDCONT) dataset.

Datasets have a horizontal structure: for each data there is one field and each row defines a new data sample. Thus, each row of the dataset will be compliant with the structure of the dataset.

> **Key Concept**
> A *field* is a piece of information specified for a dataset.

The table belows lists the datasets available:

Dataset name	Dataset Id	Description
Item - Video content	RCMITMVIDCONT_RAW	Contains a list of items of type VIDEO_CONTENT with corresponding metadata
Item - Video program	RCMITMVIDPROG_RAW	Contains a list of items of type VIDEO_PROGRAM with corresponding metadata
Item - Video channel	RCMITMVIDCHNL_RAW	Contains a list of items of type VIDEO_CHANNEL with corresponding metadata
Item - Audio content	RCMITMAUDCONT_RAW	Contains a list of items of type AUDIO_CONTENT with corresponding metadata
Item - Audio program	RCMITMAUDPROG_RAW	Contains a list of items of type AUDIO_PROGRAM with corresponding metadata
Item - Audio channel	RCMITMAUDCHNL_RAW	Contains a list of items of type AUDIO_CHANNEL with corresponding metadata
Item - Generic content	RCMITMGENCONT_RAW	Contains a list of items of type GENERIC_CONTENT with corresponding metadata
Item - Web page content	RCMITMWEBPAGECONT_RAW	Contains a list of items of type WEBPAGE_CONTENT with corresponding metadata
Item - Book content	RCMITMBOOKCONT_RAW	Contains a list of items of type BOOK_CONTENT with corresponding metadata
Item - People content	RCMITMPEOPLE_RAW	People dataset

Item - Catchup content	RCMITMCHUP_RAW	Catch up dataset
User - Person	RCMUSRPERS_RAW	Contains a list of users of type PERSON with corresponding metadata
User - Terminal	RCMUSRTERM_RAW	Contains a list of users of type TERMINAL with corresponding metadata
Item Access - Video and audio	RCMACCESS_RAW	Contains a list of video or audio item accesses with corresponding metadata
Item Access - Generic	RCMACCESSGEN_RAW	Contains a list of generic item accesses with corresponding metadata
Item Access - Web page	RCMACCESSWEB_RAW	Contains a list of web page accesses with corresponding metadata
Item Access - Book	RCMACCESSBOOK_RAW	Contains a list of book accesses with corresponding metadata
Item Access - People	RCMACCESSPEOP_RAW	People access dataset

Customer data

> **Important Note**
> Customer data needs to be converted into *User* datasets.

See User terminology for details related to how user is represented in the system.

When importing customer data, the following fields are mandatory:

- *provuserid*: unique identifier for the user within the service.
- *usertype*: the type of the user.
- *termid*: the identifier of the TERMINAL to which the user is associated, only if the usertype is PERSON.

See Metadata reference for the list of standard metadata that can be imported as customer information.

Non-standard metadata can be imported by the ETL:

- User metadata: set the following fields for each metadata:
  - *umd<A,B,C,D,E>name*: the name of the metadata.
  - *umd<A,B,C,D,E>*: the values of the metadata.
- User metadata specific for a provider: set the following fields for each metadata:
  - *upmd<A,B,C>name*: the name of the metadata.
  - *upmd<A,B,C>*: the values of the metadata.

Content metadata

> **Important Note**
> Content metadata needs to be converted into *Item* datasets.

> **Important Note**
> ETL should guarantee that metadata values for array metadata are unique (case insensitive).
> By default ContentWise does not check this condition on item creation, this behavior can be changed but this may lead to a loss of performance during content import phase.

See Item terminology for details related to how content is represented in the system.

When importing content metadata, the following fields are mandatory:

- *provitemid*: unique identifier for the item within the service.
- *itemtype*: the type of the item.
- *MdLanguage* : the metadata language (2 digits, format ISO 639)

See Metadata reference for the list of standard metadata that can be imported as customer information.

Non-standard metadata can be imported by the ETL:

- Item metadata: set the following fields for each metadata:
  - *imd<A,B,C,D,E>name*: the name of the metadata.
  - *imd<A,B,C,D,E>*: the values of the metadata.
- Item metadata specific for a provider: set the following fields for each metadata:
  - *ipmd<A,B,C>name*: the name of the metadata.
  - *ipmd<A,B,C>*: the values of the metadata.

User activity data

> **Important Note**
> User activity data needs to be converted into *Item Access* datasets.

See Event terminology for details related to how user activity is represented in the system.

When importing user activity data, the following fields are mandatory:

- *provuserid*: the user identifier.
- *usertype*: the type of the user.
- *provitemid*: the item identifier
- *itemtype*: the type of the item.
- *ts*: the item access timestamp.

In addition to mandatory fields, to have a well formed access it is suggested to populate:

- Metadata related to access type (Used to qualify the access and for implicit rating estimation): Accessed,Purchased,Viewed
- Metadata related to access duration (Used for statistical purposes and for implicit rating estimation): PlayTime, VisionFactor, TimestampEnd
- Metadata related to time zone (Used to correctly profile user activity): TimestampStartOffset,TimestampEndOffset

Refer to Event for further details on Ratings and Accesses.

See Metadata reference for the list of standard metadata that can be imported as item access.

Non-standard metadata can be imported by the ETL:

- User activity metadata: set the following fields for each metadata:
    - *md<A,B,C>name*: the name of the metadata.
    - *md<A,B,C>*: the values of the metadata.

Last counter

> **Key Concept**
> The *last counter* of an ETL process indicates which is the last data extracted from the data source.

When running ETLs, only new data must be loaded into the system; thus, a way to mark imported data is needed.

Each ETL has an associated parameter called last counter that keeps track of the last imported samples. This counter is updated after every run and, at the start of the next one, the extractor module reads it to distinguish between old and new data.

As ETL extractors deal with both databases and files, a different last counter logic is required. It can be a timestamp, as it is usually for ETLs that collect data from databases, or a regular expression, to prevent the ETL to parse files with a specific extension indicating that they have been already examined. In particular:

- Database extractors usually store the data's *timestamp* as last counter.
- File extractors generally mark the parsed file with a *done* suffix and move it to an archive directory, or recognize unparsed files through a file name pattern. The last method involves a file name pattern containing the creation date: the ETL looks for files created the day before the execution, and imports them.

In order to retrieve historical data or in case an ETL run has been missed, it is possible to manually edit the last counter.

> **Warning**
> It is strongly recommended to limit the amount of data to import if you need to recover historical data. Proceed by importing and processing chunks of data at a time (e.g. some days).

Configuration

An ETL is characterized by a set of configuration properties. The most common are listed below.

Name	Description
Log level	Logging detail: 1 low - 10 high
Log file name	The log filename
Behaviour on Empty dataset	Action todo on empty dataset (Default: ABORT)
Separator	The default field separator
Extractor type	The extractor module name

Datasets	The list of datasets to be generated, semicolon separated
Database driver	The database driver name
Database user	The user of the database to query
Database password	The password of the database to query
Database url	The url of the database to query
SQL query	The SQL query to use to extract
Lastcounter column	The column to use as lastcounter
Default lastcounter	The default value for the lastcounter
Directory	The directory to extract (and recursively its subdirs)
File list pattern	A regular expression. Only file names matching this expression will be extracted
Parsed files suffix	The suffix to append to parsed files
Recourse into subdirs	Used to recourse into subdirectories
Extractor type	The parser module name
Loader configuration	Specify the data destination

Task

> **Key Concept**
> A *task* is batch process that is responsible of data processing.

Tasks are executed within the ProcServer server.

> **Important Note**
> Tasks configuration, correct schedule and execution are required to avoid system malfunctions.

ContentWise provides out-of-the-box tasks that need to be configured according to your data and system installation. See Tasks to configure and manage tasks with the ContentWise Portal.

> **Warning**
> Editing task configuration without accurate knowledge of the system may result in a system malfunction.

Each task is characterized by:

- *name*, that univocally identifies the task in the system.
- *task type*, the type of the task.
- *configuration*, the task configuration.

> **Key Concept**
> A task type identifies the scope of a task and the type of data process for which the task is responsible.

You can define many tasks of the same type. This allow to have different configurations of a task type without the need to change existing configurations.

The table below lists the task types that are defined in ContentWise.

Task type	Description
Active monitoring	Monitors the system by executing a set of configured calls and checking the response of each one.
Autocomplete Index Generator	Updates auto complete index for autocomplete API
Aging (DBSpaceManager)	Manages data retention and Database partitioned tables.
Cache calculator	Generates caches used by Business Rules.

Cache reloader	Refreshes caches.
Chain controller	Allows to run in sequence a list of tasks and ETLs.
CSV Unloader	To download data retrieved by executing query on ContentWise database
Engine	Performs batch elaborations to generate recommendation models.
Engine runner	The task responsible for aggregating the set of engines that have to be run.
File system cleaner	Deletes old system logs.
Flat metadata generator	Generates a list of flat metadata used by the system to increase performances.
Index updater	Updates search index used by advanced search APIs.
Live Mask Generator and Live Mask matrices reloader	To support live recommendations.
Prediction Aging, Prediction Batch, Prediction Engine and Prediction Stage	Set of batch tasks to support and deliver prediction algorithm
Process Runner	To execute scripts
Query Executor	To execute queries on ContentWise database
Reloader	Refreshes real-time recommendation models.
Remove expired bookmarks	Removes expired bookmarks from the system.
Reporter	Generates the configured reports.
Repository cleaner	Deletes old recommendation models.
Stage manager	Imports and elaborates data and information needed for generating recommendations.
Statistics updater	Updates item statistics, such as number of raters, average rating, number of viewers
Statistics generator	Generates default statistics and a list of flat metadata used by the system to increase performances.
Stream Data Align CWtoDS, Stream Facet Generator, Stream Generator, Stream Name Generator, Stream Search Index Generator, Stream Stats Generator	To support and deliver dynamic streams
Warehouse Engine,Warehouse Loader, WarehouseStageManager, WarehouseStageTableGenerator, WarehouseValidatorTask	To integrate and ContentWise Analytics component
XML generator	Generates the XML format of the data stored in ContentWise, such as content metadata and customer data.

Engine

> **Key Concept**
> An Engine task provides an implementation of the ContentWise proprietary data mining algorithms that produce recommendation models.

Different types of engine, also called *algorithm types*, are defined. Each engine provides a specific algorithm implementation and generate a specific recommendation model.

The following types of engines (also called algorithm types) are defined:

- *Collaborative*: uses social affinity, i.e. it is based on the analysis of preferences and behaviors in order to find similar users.
- *Content*: uses content affinity, i.e. it is based on the analysis of the similarity of the items, by inspecting their metadata.
- *Direct Collaborative*: another version of social affinity based algorithm.
- *Direct Content*: another version of content affinity based algorithm.
- *Collaborative KNN*: another version of social affinity based algorithm.
- *Content KNN*: another version of content affinity based algorithm.
- *Hybrid*: generates hybrid recommendations, it is based on a combination of social and content affinities.
- *Top Rated*: generates the most popular items recommendations, based on the configured periods.
- *Top Viewed*: generates the most viewed items recommendation, based on the configured periods.
- *Most Recent*: generates the most recent items recommendations, on the base of its configuration.

Engine execution configuration

Algorithm types can be configured to run in different numeric modes. The available modes are:

EXTERN PROCESS: execute the numerical elaboration using an external process.

- *Pro*: Performance / Configuration effort ratio
- *Cons*: Need to install an external component.

EXTERN MULTI PROCESS: uses a full external calculation process, which works by mean of a parallel computation grid for the numerical elaboration. It uses MPI (Message Passing Interface), a standard interface that allows to run a process over a grid of servers that collaborate in a parallel computation. Hence, the EXTERN MULTI PROCESS configuration can scale-up to any complexity using a server grid to perform the elaboration.

- *Pro*: Higher performances.
- *Cons*: Need to install a complex set of components.

Important Note
Engine execution configuration availability depends on the system installation.

The following table lists the numeric modes available for each algorithm type.

Algorithm Type	EXTERN PROCESS	EXTERN MULTI PROCESS
Collaborative	X	X
Content	X	X
Direct Collaborative	X	
Direct Content	X	
Collaborative KNN	X	
Content KNN	X	
Hybrid	X	
Top Rated	X	
Top Viewed	X	
Most Recent	X	

Engine configuration properties

To configure an engine execution, both a global and a subdomain configuration are required:

Global Configuration

- *Algorithm*: The algorithm type.
- *Configure execution*: the engine execution configuration to apply.
- A set of custom properties according to the selected algorithm type.

Subdomain Configuration

- An engine task runs for a specified set of subdomains, that are the subdomains bound to the task.
- For each bound subdomain, a task subdomain configuration can be provided.

Important Note
When a task subdomain configuration is specified, it overrides the global configuration and the default configuration of the system.

Engine Runner

Key Concept
An Engine runner task executes a set of configured engine tasks, to set up recommendation models that have to be available within the system.

An engine runner task is characterized by:

- A list of *subdomains* for which the engines have to be executed

- A set of *shared data models*: an optional set of engines that generate statistic based recommendation models, such as top rated, top viewed and most recent. These shared data models are required to support layout-based recommendations.
- A set of *models*: the name of the engines that are required by the integration.

> **Warning**
> All the engines required by the engine runner configuration must be configured for the subdomains selected in the engine runner configuration

> **Important note**
> Once the engine runner completes successfully, recommendation models are stored but they are not yet available to real time. Relaoder task execution is required to apply the new generated models.

Stage Manager

> **Key Concept**
> The *Stage Manager* is the main batch task. It is mainly responsible of:
>
> - Importing data from the stage area into the system.
> - Processing item and user status transitions.
> - Processing data in order to extract statistic and information required by the system to generate updated recommendation models and content analysis

> **Important Note**
> A Stage Manager task can be configured to execute only some of the available processes. Being a task, its configuration can be managed by Tasks menu of Data Types section of the ContentWise Portal.

Equality Set configuration

- *Process item set information*: generates for each item a couple of values (*EquSet* and *EquObj*) used to assign the item to a specific set in a specified order. Only one item for each set is recommended. These values can be used during recommendation process in order to recommend only a single episode of a tv series.

Item status management configuration

- *Process item status transitions*: removes from the system deletable items.

Data processing configuration

Import data from stage area. Options:

- *Process data*: imports data from stage area. Valid values:
  - Full: import new items and update already imported data (if any)
  - Only new: import only new items
  - Skip: Skip data import
- What to process. The entities to import. Valid choices:
  - Items
  - User
  - Events (accesses, ratings, purchases, ...)

Aging configuration

- *Execute aging*: aging process for item accesses.

Derived service processing configuration

Generates item identifiers for derived services. It is required when a new derived service is configured to derive new identifiers for existing items. Options:

- *Start Service Generation Process*
- *List of services to be derived*

Profile management configuration

- *Update profiles*: evaluate the configuration of the user profiling (e.g. time-based profiles) and update user profiles.
- *Generate virtual users*: generate virtual users according to multi profile configuration.

Statistics generation configuration

- *Update statistics*: generates item and user statistics.

- *Generate real time queries*: generates a set of queries that are used by the aggregation process.
- *Generate Item Content Matrix*: processes item metadata to generate the stems needed by content-based algorithms.
- *Generate user groups recommendations*: updates user groups recommendations.
- *Process user cleaning*: dismisses users that have USER_PENDING_DISMISSED status.

User Groups management configuration

Update user group statistics (e.g. top rated by your friends)

User cleaning management configuration

Enable automatic user cleaning process

RecServer

✓ **Key Concept**
The *RecServer* is the ContentWise server dedicated to online processing. It is responsible of providing the Real Time API interface.

The RecServer is provided by the Application Server services of ContentWise. It should always be active in the system.

⚠ **Warning**
If RecServer is inactive, no Real Time API interface is available and the system is not able to satisfy real time requests. See Application Server services to check and manage the status of the application servers.

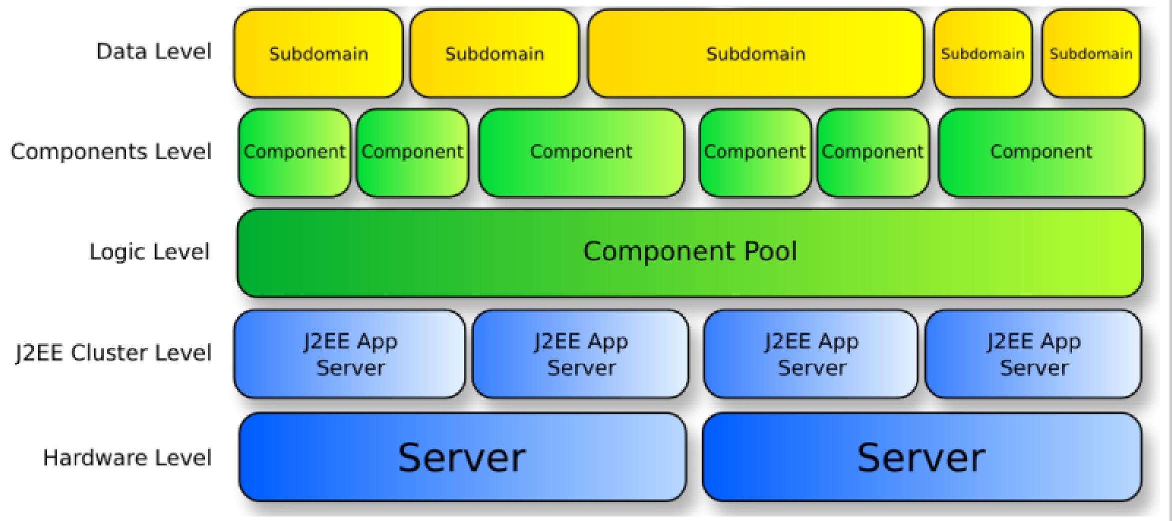
The RecServer is based on EJB services deployed in a J2EE Application Server. It has a two layer structure:

- *AlgoServer*: provides the core recommendation logic. An AlgoServer is strictly associated to an algorithm type and a set of served subdomains.
- *RecoServer*: provides the web service interfaces to access the system. RecoServer balances requests among AlgoServers configured for the same subdomain-algorithm pair.

ⓘ **Important Note**
An efficient configuration of component pools and components is required to configure the RecServer server. See API Interface - Deployment documentation of the ContentWise Portal to configure component pools and components.

The image below explains how components are mapped to hardware servers. Each application server logically serves a set of ContentWise components. The components belong to a component pool that provides the service (all subdomains, all algorithms) to the callers.

ⓘ **Important Note**
Two hardware servers can be used to obtain High Availability or for load balancing purposes. On a hardware server, one or more J2EE application servers can run (on different ports).



RecServer architecture

The following rules define AlgoServer and RecoServer basic concepts:

- an AlgoServer hosts one and only one algorithm type.
- an AlgoServer can be configured to manage many subdomains.
- a RecoServer receiving a call for a specific subdomain and algorithm will forward the request to one of the AlgoServers that matches all the following:
    - is in the same component pool of the RecoServer.
    - hosts the requested algorithm type.
    - is configured for the requested subdomain.

> **Important Note**
> RecoServer behavior applies a round-robin policy that balances load between AlgoServers.

> **Important Note**
> An AlgoServer that is not responding is marked as invalid and re-polled after five minutes.

ContentWise Portal

This chapter describes ContentWise Portal functionalities.

ContentWise Portal provides the UI for configuring, managing and monitoring ContentWise.

> **Key Concept**
> A *section* is an area of the ContentWise Portal that is dedicated to the configuration and management of a defined set of ContentWise functionalities. Each section is made up of pages.

> **Key Concept**
> A *page* is a container of interactive visual components, called *portlets*.

ContentWise Portal sections are listed below:

Section	Description
Service Model	Manages services and subdomains.
Data Management	Configures the batch processes of ContentWise such as ETL and tasks.
UX Design	Configures ContentWise callers and layouts used at real time.
Business Rules	Manages business rules.
Analytics	Reporting functionalities.
Knowledge Factory	Includes Knoledge Factory management console. If Knowledge Factory is not part of your installation, this section is empty.
Administration	Configures ContentWise data types, component deployments, license, settings, ...

Accessing the ContentWise Portal

To access the ContentWise Portal, point your web browser to the URL: `http://cw_server_address:cw_server_port/cwportal`, where:

- <cw_server_address> is the name of the server where ContentWise Portal is installed.
- <cw_server_port> is the port of the application server on which ContentWise Portal is deployed.

> **Supported web browsers**
> The ContentWise Portal works on Mozilla Firefox 3.5+, Google Chrome 10+, Windows Internet Explorer 9+.

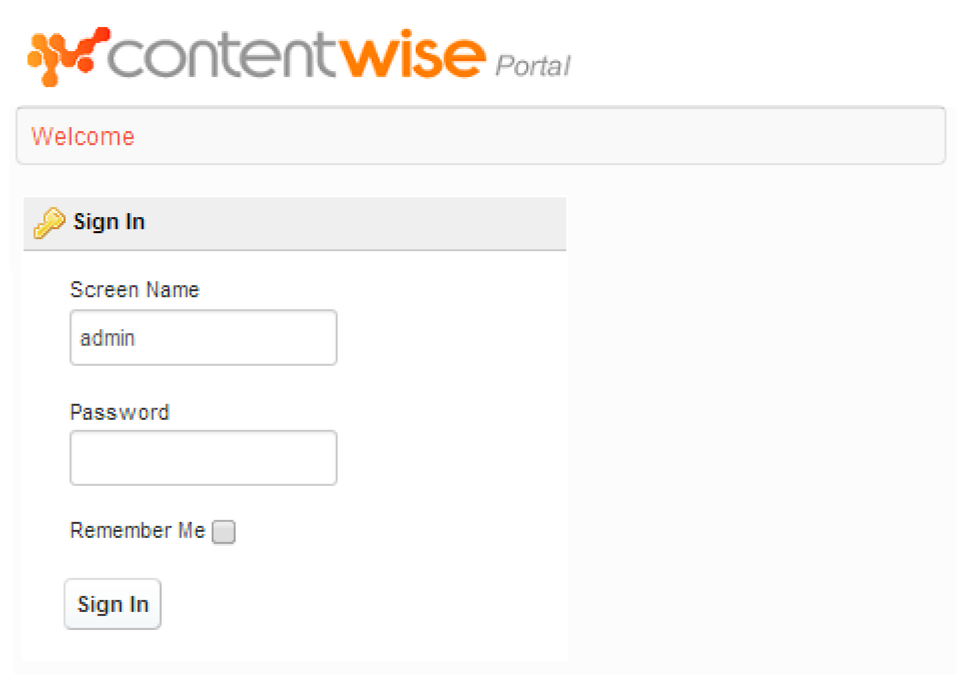
> **Important Note**
> In order to login to the ContentWise Portal, you need a valid username and a password.

Type your username and password and click the *Sign in* button.

User rights
User access to portal sections is controlled by user rights defined as administration level. Users may have no access to some sections of the portal.

Warning
Do not access ContentWise Portal from different browser tabs or windows concurrently. It may cause errors when updating configurations.

To define ContentWise Portal users, please refer to KB023 - Add a new portal user



Screen Name: admin

Password:

Remember Me □

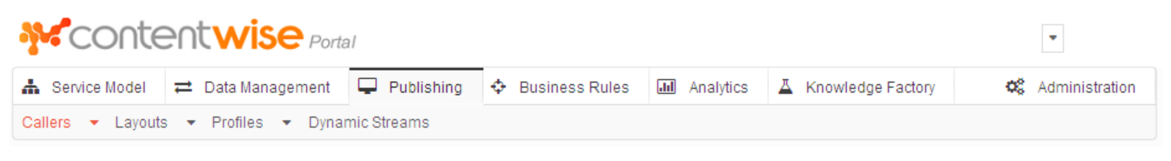
ContentWise Portal - Sign in

Interact with the ContentWise Portal

UI structure

After successful login, users are taken to the ContentWise Portal UI, which is composed of three main layers:

- *Section browser*: to navigate through the various sections.
- *Page browser*: to navigate through the pages of a section.
- *Portlets*: the building blocks of a page.



ContentWise Portal - Section and page browsers

Filters

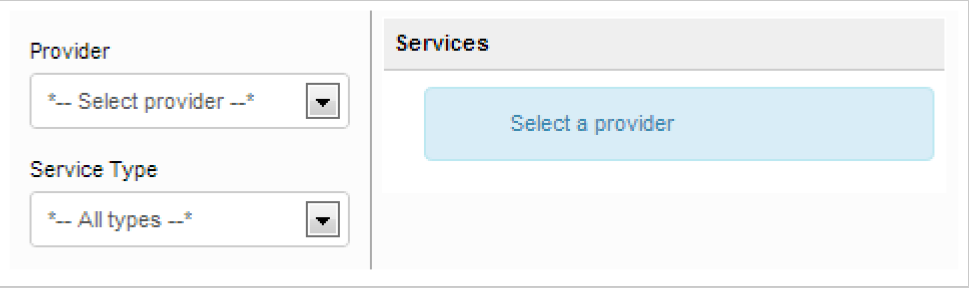
Key Concept
A filter is a particular type of portlet that is used to configure which data have to be shown. A filter may be:

- *required*: when no data can be shown until a filter is set by the user.
- *optional*: when a filter can be set to have a specific view on a subset of the available data.

The pages of the portal may require that one or more filters are configured. Portlets may not show data until required filters are set.

Portlets are interactive components that automatically refresh when registered filters change.

> **Important Note**
> If a portlet requires a specific filter to be set, it shows a message that prompts the user to set the filter.



Provider: *-- Select provider --*

Service Type: *-- All types --*

Services

Select a provider

ContentWise Portal - An example of portlets that require a filter to be set.

Service Model

This section describes functionalities provided by *Service Model* section of the ContentWise Portal.

The table below lists the pages of the section.

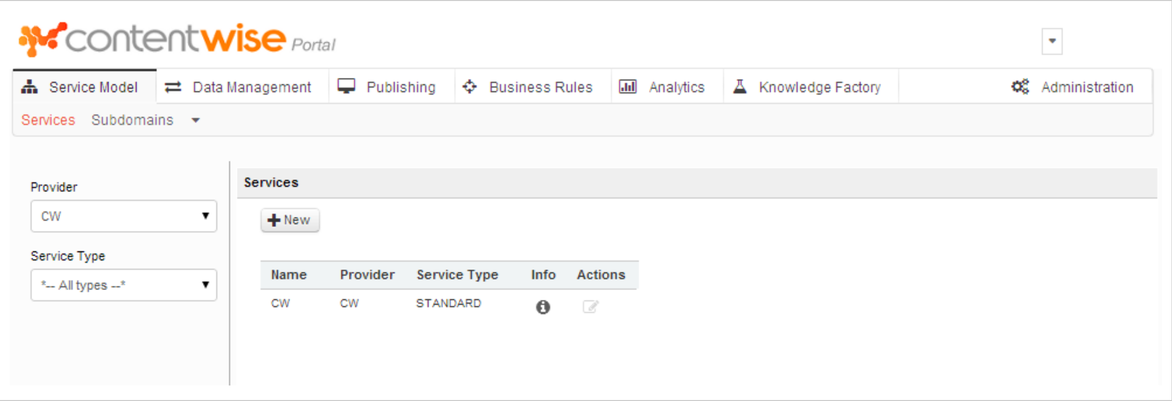
Page	Description
Services	Configures the services available in the system.
Subdomains	Configures the subdomains available in the system.

Services

This section describes functionalities provided by *Services* page. See Provider and Service for a definition of service.

In this page you can:

- have an overview of the services configured in the system.
- create new services and edit service configurations.



Provider: CW

Service Type: *-- All types --*

Services

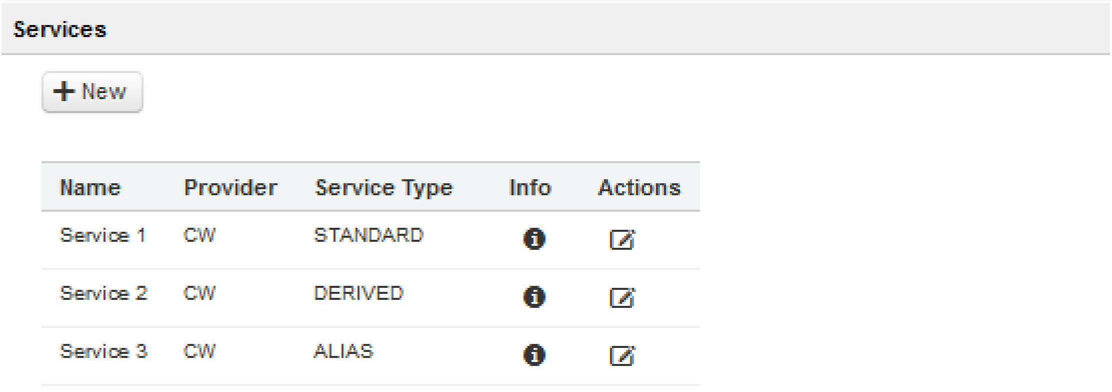
Name	Provider	Service Type	Info	Actions
CW	CW	STANDARD	ℹ	✎

Services: list of configured standard and derived services

Services

The *Services* portlet lists all available services associated to a selected provider.
For each service, the table shows:

- *Name*: unique service identifier.
- *Provider*: the provider associated to the service.
- *Service Type*: standard, derived, or alias service.
- *Info*: click the icon to show the service details.
- *Actions*: select the edit action to modify the service.



Services

Name	Provider	Service Type	Info	Actions
Service 1	CW	STANDARD	ℹ	✎
Service 2	CW	DERIVED	ℹ	✎
Service 3	CW	ALIAS	ℹ	✎

Services: Services portlet.

Create a new service

This section shows how to add a new service in the system.

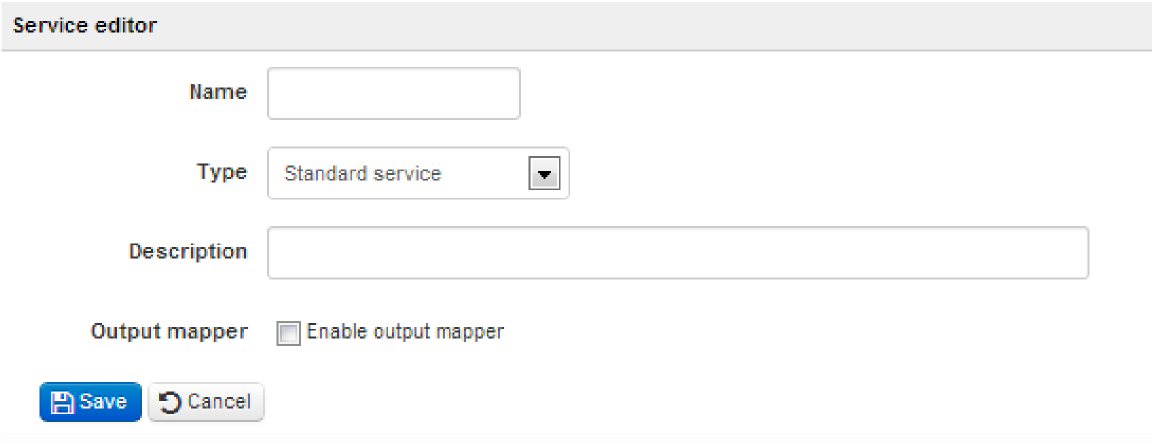
To create a new service click the *New* button in the Services portlet. Note that a service is created for a specific provider, that has to be selected in advance. If a provider is not selected, you will be prompted to select one.
You can create a standard service, a derived service, or an alias service.

Create a new standard service

To create a new standard service, select *Standard service* as value of the *Type* field.
All form information but description is mandatory. Below a description of the required data.

- *Name*: a string that univocally identifies the service in the system (e.g. VOD_SERVICE, LIVE_SERVICE). Valid characters are **[A-Za-z0-9]**, **-**, _, **.**
- *Output mapper*: see Output mapper

To save current service configuration click *Save* button.
Click *Cancel* to undo the operation.



Service editor

Name

Type: Standard service

Description

Output mapper: ☐ Enable output mapper

Services: create a new standard service.

Create a new derived service

To create a new derived service, select *Derived service* as value of the *Type* field.

All form information but description must be filled. Below a description of the required data.

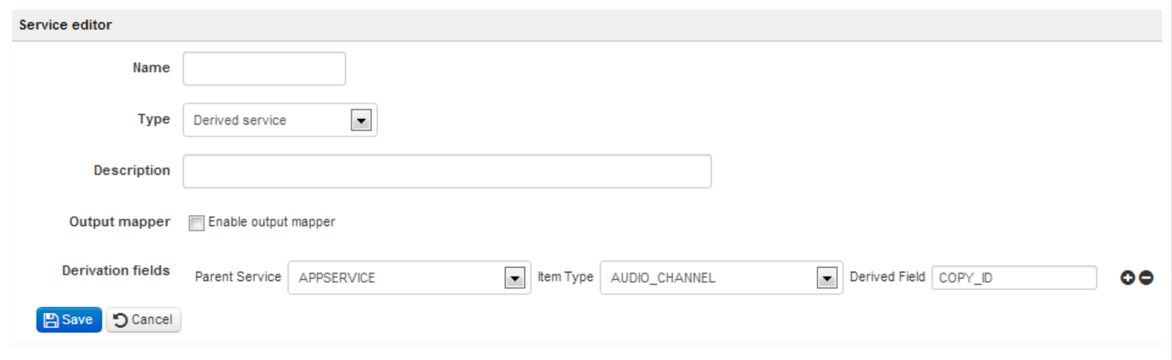
- *Name*: a string that univocally identifies the service in the system (e.g. VOD_SERVICE, LIVE_SERVICE).

> ⚠ **Warning**
> Valid characters are **[A-Za-z0-9]**, **-**, _, **.**

- *Output mapper*: see Output mapper
- *Derivation Fields*: a list of one or more derivation pairs. A derivation field specifies, for a given service (referred to as **parent** service) and a given itemtype, how item identifiers should be derived from the parent service.

> **Important Note**
> A derived field is a couple itemtype/derive field.
> Valid derive field values are:
>
> - COPY_ID: this string specifies that the item identifier for the derived service must be the same of the parent service.
> - An item metadata name: specifies that the item identifier for the derived service will correspond to the value of an item metadata. Must be a not language aware metadata.
> - A provider metadata name.

To save current service configuration click *Save* button.
Click *Cancel* to undo the operation.



Service editor

Name

Type: Derived service

Description

Output mapper: □ Enable output mapper

Derivation fields: Parent Service: APPSERVICE; Item Type: AUDIO_CHANNEL; Derived Field: COPY_ID

Services: create a new derived service.

Create a new alias service

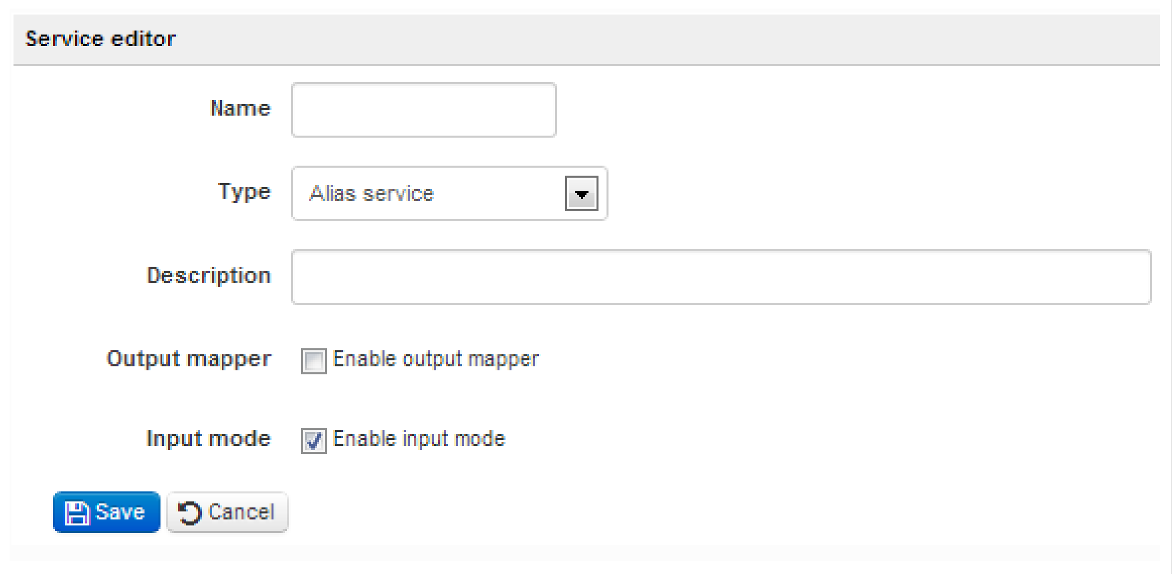
To create a new alias service, select *Alias service* as value of the *Type* field.

All form information but description must be filled. Below a description of the required data.

- *Name*: a string that univocally identifies the service in the system (e.g. VOD_SERVICE, LIVE_SERVICE).

> **Warning**
> Valid characters are **[A-Za-z0-9]**, **-**, _, **.**

- *Output mapper*: see Output mapper
- *Input mode*: if such option is selected, input mode will be enabled in the new alias service.



Service editor

Name

Type: Alias service

Description

Output mapper: □ Enable output mapper

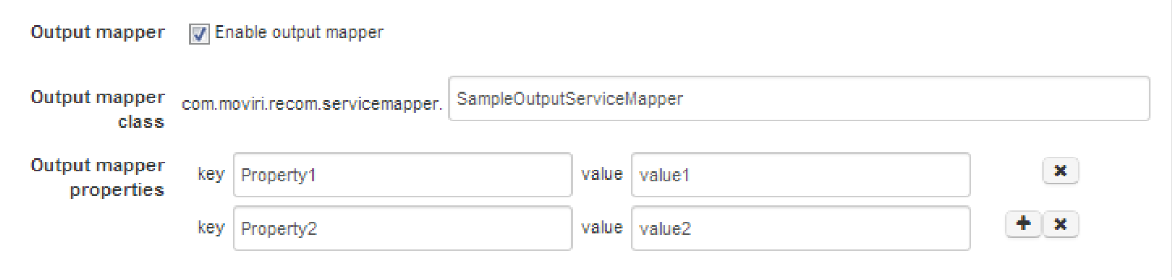
Input mode: ☑ Enable input mode

Services: create a new alias service.

Output mapper

When you create (or edit) a service you can optionally enable an *output mapper*. In such a case you have to specify:

- *Output mapper class*: a java class that implements the output mapper. See KB012 for further details.
- *Output mapper properties*: a set of <key, value> properties required by the output mapper class.



Output mapper ☑ Enable output mapper

Output mapper class: com.moviri.recom.servicemapper. SampleOutputServiceMapper

Output mapper properties:
key Property1 value value1
key Property2 value value2

Services: service output mapper.

Subdomains

This section describes functionalities provided by *Subdomains* menu.

Subdomains menu gives access to the following pages:

Page	Description
Definitions	It is the landing page of the section. It provides an overview of the subdomains in the system and allows to manage their configurations.
Rules	It is the page dedicated to the management of subdomain rules.

Definitions

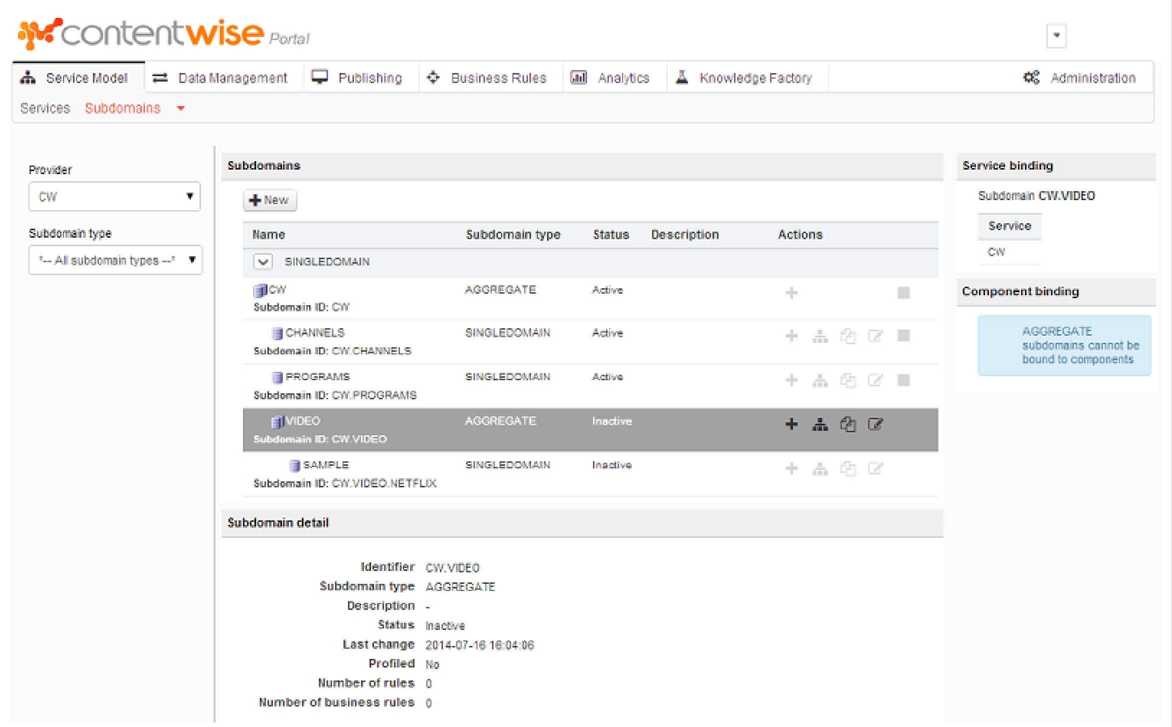
This page provides an overview about how subdomains are configured in the system and provides access to the pages that allow to create and edit subdomains.

The page contains the following portlets:

- Subdomains: lists the subdomains that are in the system and provides access to new/edit subdomain pages.
- Subdomain detail: provides details about the selected subdomain.
- Service bindings: manages the subdomain-service bindings.
- Component bindings: manages the subdomain-component bindings.

See create a new subdomain to create a new subdomain.

You can filter by subdomain type to view only subdomains of a specific type. If no subdomain type filter is selected, all subdomains are shown.



Provider: CW

Subdomain type: *-- All subdomain types --*

Subdomains

Name	Subdomain type	Status	Description	Actions
SINGLEDOMAIN				
CW Subdomain ID: CW	AGGREGATE	Active		
CHANNELS Subdomain ID: CW.CHANNELS	SINGLEDOMAIN	Active		
PROGRAMS Subdomain ID: CW.PROGRAMS	SINGLEDOMAIN	Active		
VIDEO Subdomain ID: CW.VIDEO	AGGREGATE	Inactive		
SAMPLE Subdomain ID: CW.VIDEO.NETFLIX	SINGLEDOMAIN	Inactive		

Subdomain detail

Identifier	CW.VIDEO
Subdomain type	AGGREGATE
Description	-
Status	Inactive
Last change	2014-07-16 16:04:06
Profiled	No
Number of rules	0
Number of business rules	0

Service binding

Subdomain CW.VIDEO

Service
CW

Component binding

AGGREGATE subdomains cannot be bound to components

Subdomain definitions: list of configured subdomains and related configurations.

Subdomains

This portlet provides an overview of subdomain configuration and provides access to subdomain configuration pages.

AGGREGATE and SINGLEDOMAIN subdomains are shown according to their hierarchical structure.

For each subdomain, the table shows:

- *Name*: the subdomain name and the subdomain ID (the subdomain ID is a unique identifier built by concatenating the subdomain name and the subdomain parent name)
- *Subdomain type*: the type of the subdomain.
- *Status*: subdomain status. If subdomain has at least one bound component the subdomain is ACTIVE (green arrow icon), otherwise it is INACTIVE (red square icon).
- *Description*: short subdomain description.
- *Actions*: see Subdomain actions

Subdomain actions

The available actions vary according to the subdomain type.

- AGGREGATE subdomain actions:
  - *add child*: to create a new subdomain as child of the selected one.
- SINGLEDOMAIN subdomain actions:
  - *add child*: to create a new subdomain as child of the selected one.

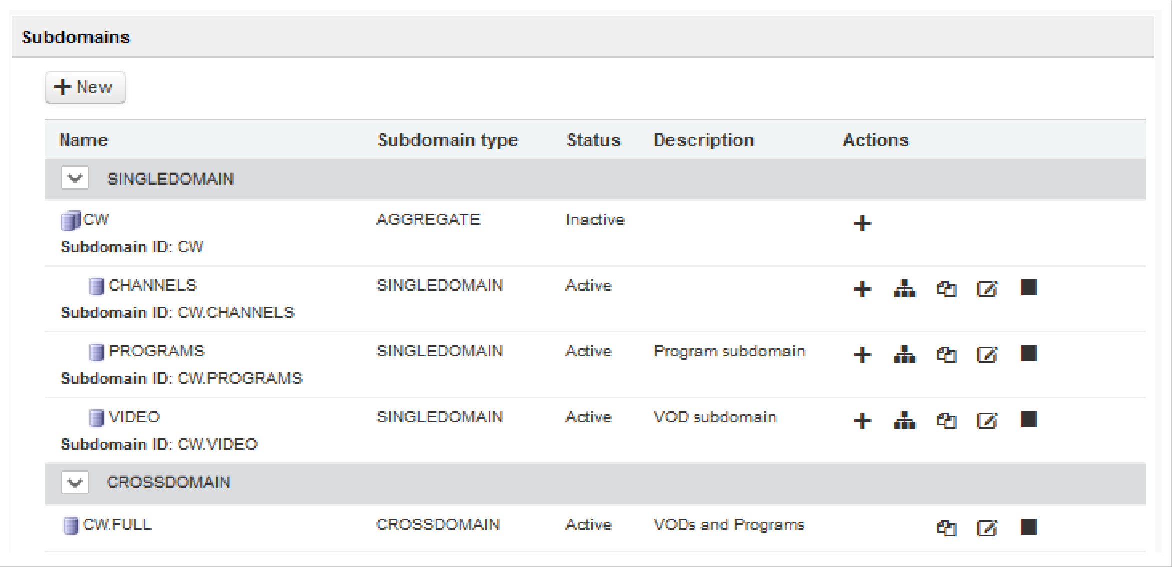
    Warning
    This operation turns SINGLEDOMAIN subdomain into an AGGREGATE subdomain.

  - *change parent*: to change current subdomain parent. It modifies the hierarchical structure of the subdomains.
  - *clone*: to copy the current subdomain and its configuration into a new subdomain.
  - *edit*: to change subdomain information (e.g. description).
  - *disable*: to disable a subdomain. It removes all subdomain-component bindings.

    Important Note
    This section is available only for active subdomains.

- CROSSDOMAIN subdomain actions:
  - *clone*: to copy the current subdomain and its configuration into a new subdomain.
  - *edit*: to change subdomain information (e.g. description).
  - *disable*: to disable a subdomain. It removes all subdomain-component bindings.

Important Note
This section is available only for active subdomains.



Subdomains

+ New

Name	Subdomain type	Status	Description	Actions
SINGLEDOMAIN				
CW Subdomain ID: CW	AGGREGATE	Inactive		+
CHANNELS Subdomain ID: CW.CHANNELS	SINGLEDOMAIN	Active		+ ⊥ ⧉ ✎ ■
PROGRAMS Subdomain ID: CW.PROGRAMS	SINGLEDOMAIN	Active	Program subdomain	+ ⊥ ⧉ ✎ ■
VIDEO Subdomain ID: CW.VIDEO	SINGLEDOMAIN	Active	VOD subdomain	+ ⊥ ⧉ ✎ ■
CROSSDOMAIN				
CW.FULL	CROSSDOMAIN	Active	VODs and Programs	⧉ ✎ ■

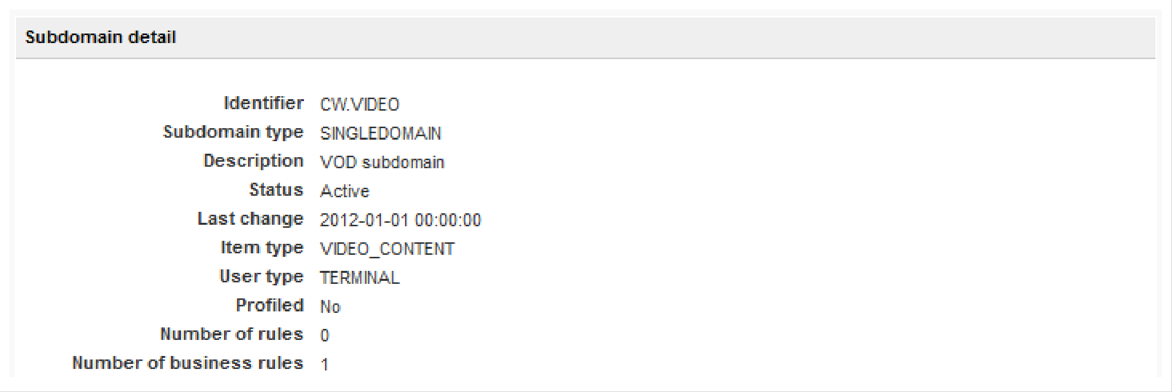
Subdomain definitions: subdomains portlet.

Subdomain detail

This portlet provides details about the subdomain that has been selected in the Subdomains portlet.

The provided information varies according to the type of the selected subdomain:

- *Name*: the subdomain identifier, as it must be used in the ContentWise API.
- *Type*: the type of the subdomain.
- *Description*: short subdomain description.
- *Status*: subdomain status. If subdomain has at least one bound component the subdomain is ACTIVE (green arrow icon), otherwise it is INACTIVE (red square icon).
- *Last change*: the timestamp of the last change occured on subdomain information.
- *Item type*: subdomain item type, only for SINGLEDOMAIN type.
- *User type*: subdomain user type, only for SINGLEDOMAIN type.
- *Profiled*: indicates if the subdomain is profiled or not, only for SINGLEDOMAIN and AGGREGATE types.
- *Number of rules*: number of configured subdomain rules, only for SINGLEDOMAIN and AGGREGATE types.
- *Number of business rules*: number of business rules bound to the subdomain, only for SINGLEDOMAIN and AGGREGATE types.
- *Members*: the list of subdomains that are part of the subdomain, only for CROSSDOMAIN type.



Subdomain detail

Identifier	CW.VIDEO
Subdomain type	SINGLEDOMAIN
Description	VOD subdomain
Status	Active
Last change	2012-01-01 00:00:00
Item type	VIDEO_CONTENT
User type	TERMINAL
Profiled	No
Number of rules	0
Number of business rules	1

Subdomain definitions: subdomain detail portlet.

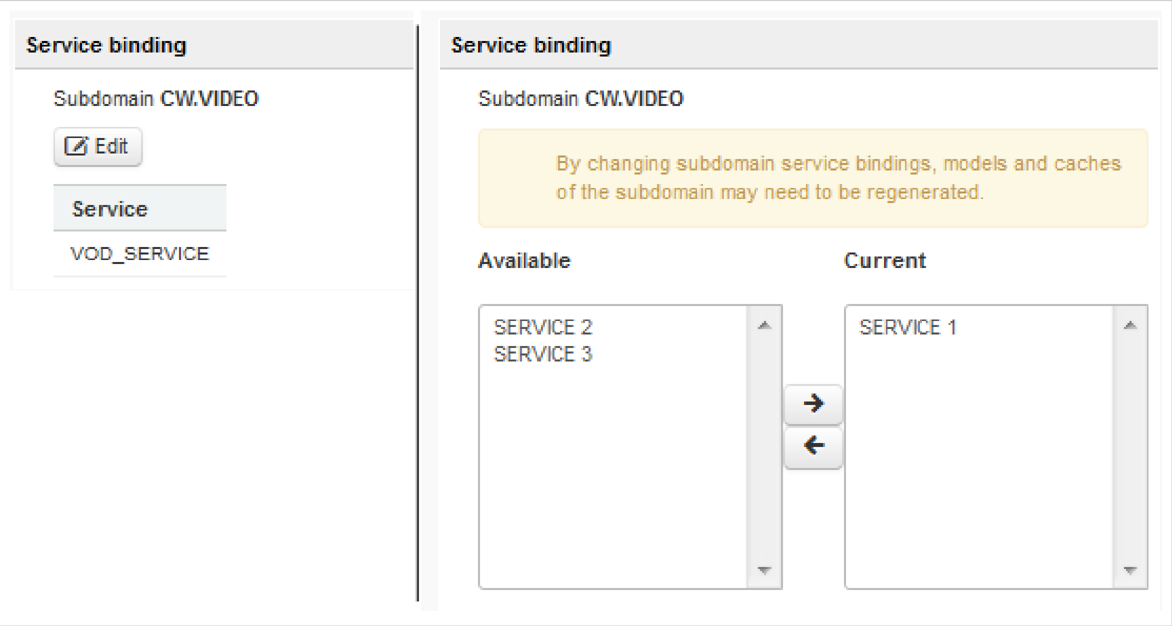
Service bindings

This portlet provides information about the bindings between the subdomain selected in Subdomains portlet and the services that are configured in the system.

The portlet lists the services that are currently bound to the subdomain.

To modify the current configuration:

1. Click *Edit* to change the configuration.
2. Select a service and click on left or right arrows to modify the bindings. This step can be repeated for more than a service.
3. Click *Save* to submit the new configuration, *Cancel* to undo the operation



By changing subdomain service bindings, models and caches of the subdomain may need to be regenerated.

Subdomain definitions: Service binding portlet (view and edit).

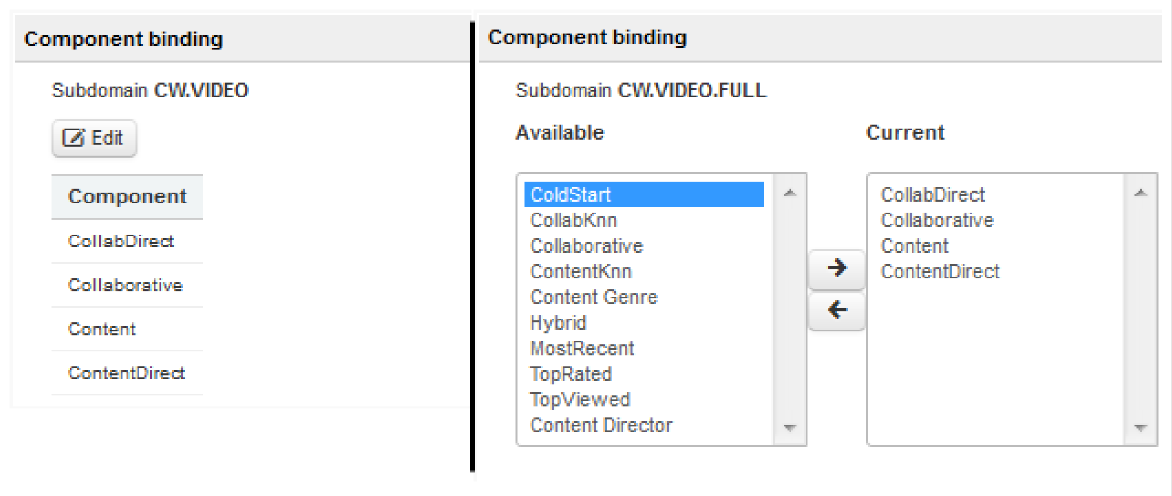
Component bindings

This portlet provides information about the subdomain selected in Subdomains portlet and the components that are configured in the system.

The portlet lists the components that are currently bound to the subdomain.

To modify the current configuration:

1. Click *Edit* to change the configuration.
2. Select a caller and click on left or right arrows to modify the bindings. This step can be repeated for more than a component.
3. Click *Save* to submit the new configuration, *Cancel* to undo the operation



Subdomain definitions: Component binding portlet (view and edit).

Create a new subdomain

This section shows how to add a new subdomain in the system.

To create a new subdomain click the *New* button in the Subdomains portlet.

You will be required to select:

- A provider for which the new subdomain will operate.
- A subdomain type for the new subdomain.

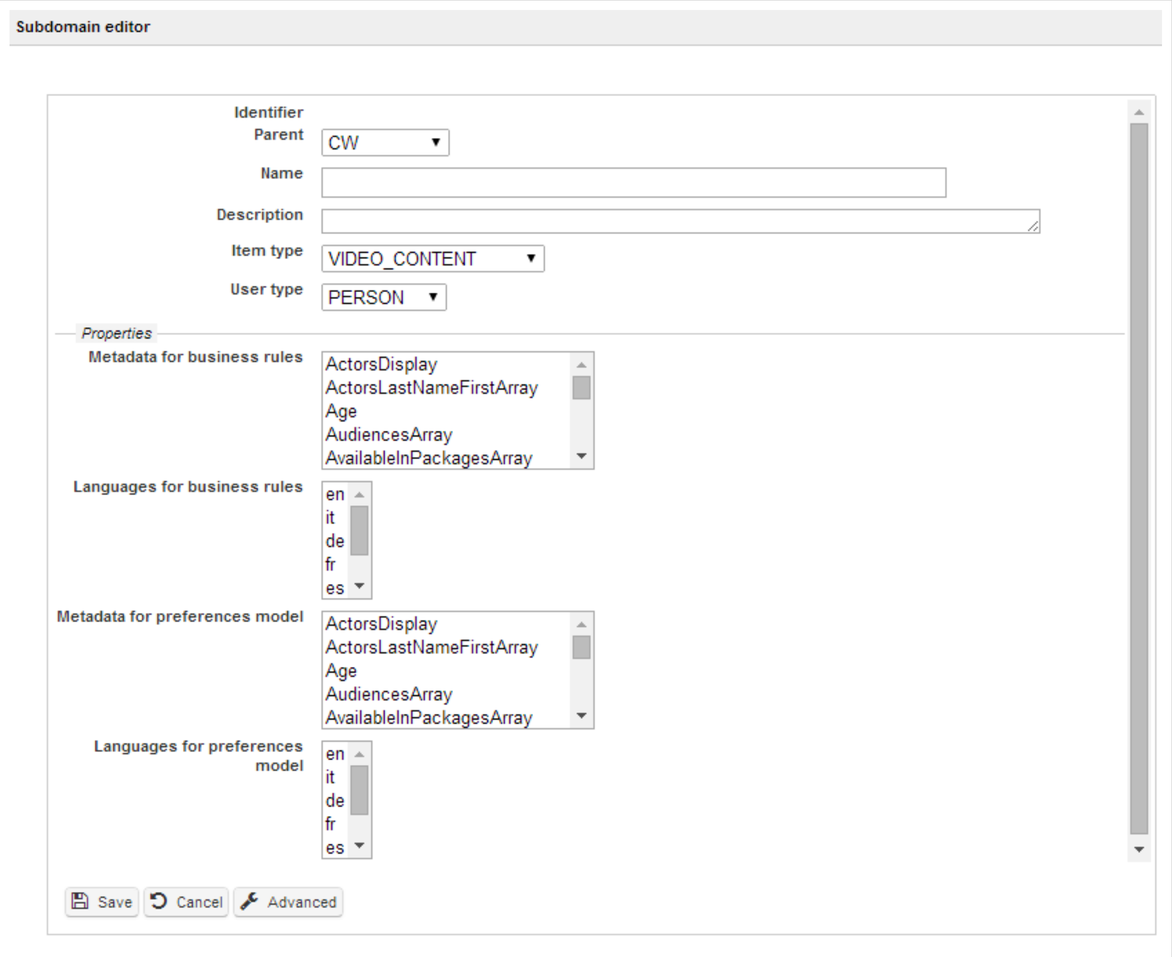
> **Important Note**
> AGGREGATE subdomains cannot be created directly by the user. AGGREGATE subdomains are generated by adding children to a SINGLEDOMAIN subdomain.

Create a new SINGLEDOMAIN subdomain

To create a new SINGLEDOMAIN subdomain, select SINGLEDOMAIN as value of the *Subdomain type* filter.

All information but description must be filled. The name of the new subdomain will be built by merging the identifier of the parent subdomain and the string that you will type in the appropriate field.
Note that additional field are available for VIDEO_CHANNEL and PROGRAM item types.

> **Warning**
> Subdomain name can not contain '.', '/' and '-' characters



Subdomain editor

Identifier

Parent: CW

Name:

Description:

Item type: VIDEO_CONTENT

User type: PERSON

Properties

Metadata for business rules: ActorsDisplay, ActorsLastNameFirstArray, Age, AudiencesArray, AvailableInPackagesArray

Languages for business rules: en, it, de, fr, es

Metadata for preferences model: ActorsDisplay, ActorsLastNameFirstArray, Age, AudiencesArray, AvailableInPackagesArray

Languages for preferences model: en, it, de, fr, es

Subdomain definitions: Create a new SINGLEDOMAIN subdomain.

The configuration of the subdomain properties is required to properly configure different functionalities that may be required by the integration with the subdomain:

- Business rules: only the item metadata and the languages configured in the subdomain properties "Metadata for business rules" and "Languages for business rules" will be available for the definition and application of business rules.

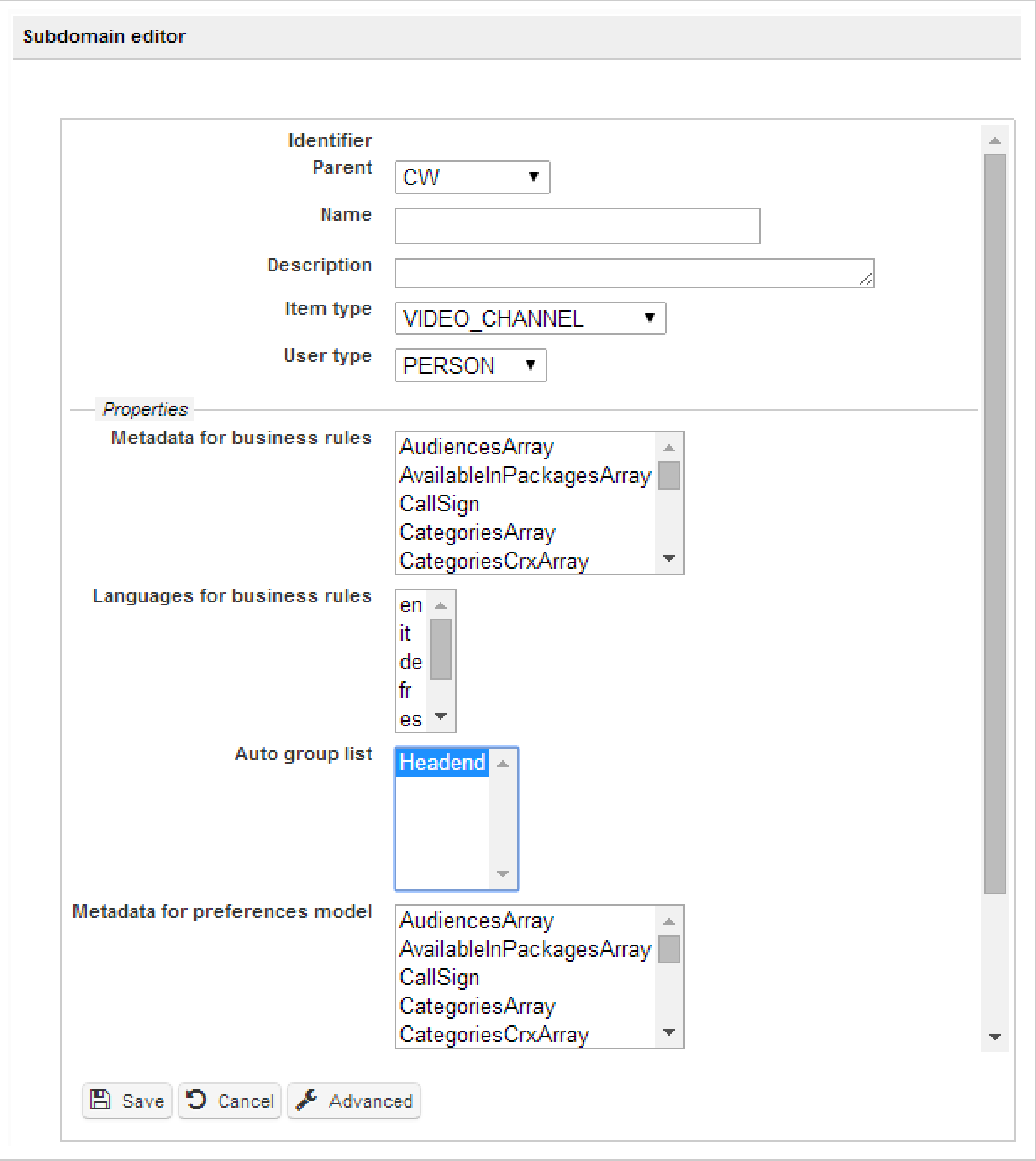
  > **Warning**
  > A business rule defined for a metadata that is not available in the subdomain as "Metadata for business rules" will generate errors if used.

- Preferences model: only the item metadata and the languages configured in the subdomain properties "Metadata for preference model" and "Language for preference model" will be considered by the engine to build preference models and recommendation explanations.

Create a new SINGLEDOMAIN subdomain of VIDEO_CHANNEL items

The following additional fields are available in the case of VIDEO_CHANNEL item type:

- *Auto group list*: it allows to enable one or more existing *auto groups* in order to group channels on the basis of a specific metadata (e.g., to group channels on the basis of the head-ends they are associated to).



Subdomain editor

Identifier

Parent: CW

Name

Description

Item type: VIDEO_CHANNEL

User type: PERSON

Properties

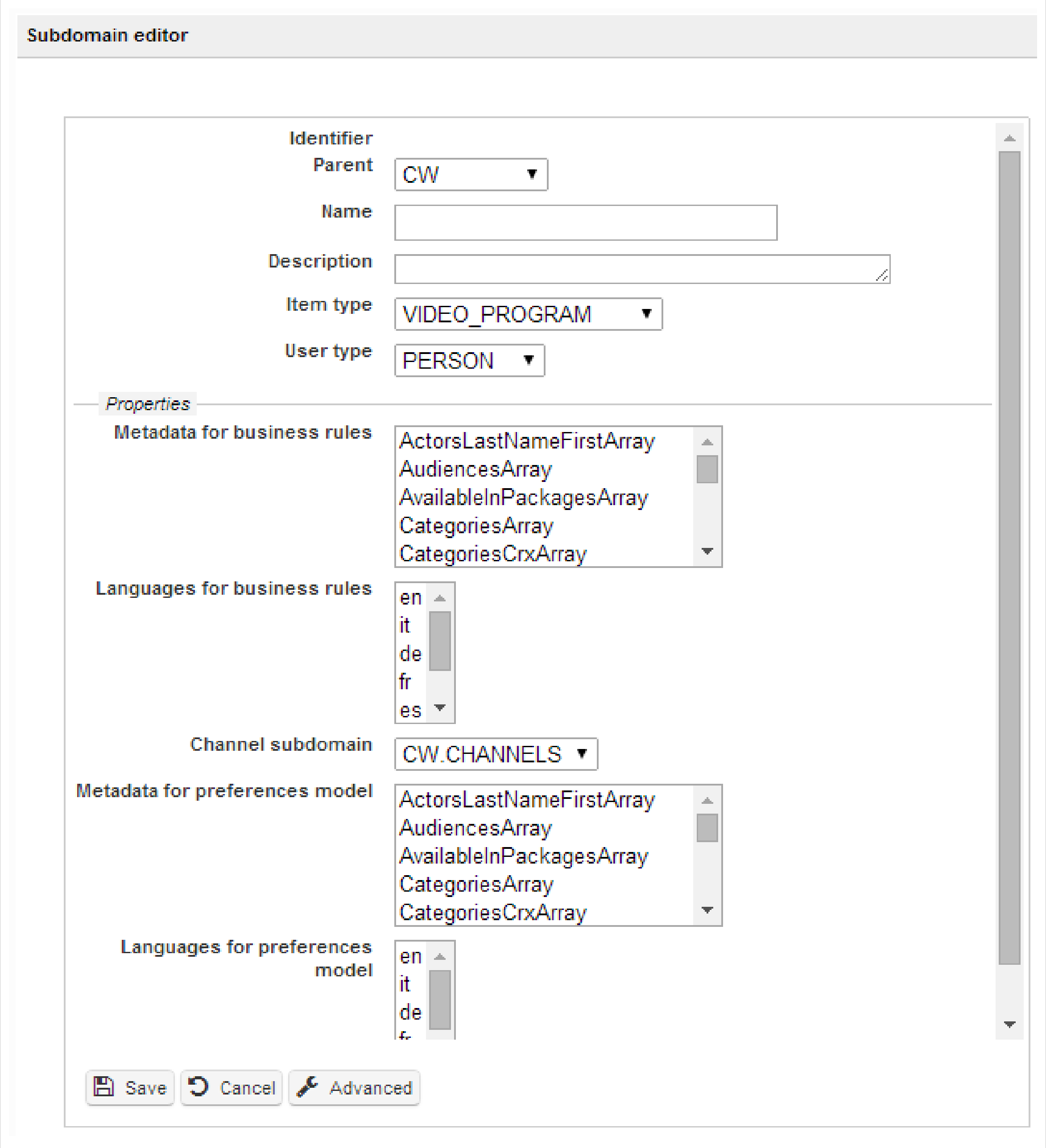
Auto group list: Headend

Subdomain definitions: Create a new SINGLEDOMAIN subdomain formed by items of type VIDEO_CHANNEL.

Create a new SINGLEDOMAIN subdomain of PROGRAM items

The following additional fields are available in the case of VIDEO_PROGRAM and AUDIO_PROGRAM item type:

- *Channel subdomain*: subdomain of channels related to the *VIDEO_PROGRAM* or *AUDIO_PROGRAM* items.



Subdomain editor

Identifier

Parent: CW

Name:

Description:

Item type: VIDEO_PROGRAM

User type: PERSON

Properties

Metadata for business rules: ActorsLastNameFirstArray, AudiencesArray, AvailableInPackagesArray, CategoriesArray, CategoriesCrxArray

Languages for business rules: en, it, de, fr, es

Channel subdomain: CW.CHANNELS

Metadata for preferences model: ActorsLastNameFirstArray, AudiencesArray, AvailableInPackagesArray, CategoriesArray, CategoriesCrxArray

Languages for preferences model: en, it, de

Subdomain definitions: Create a new SINGLEDOMAIN subdomain formed by items of type VIDEO_PROGRAM.

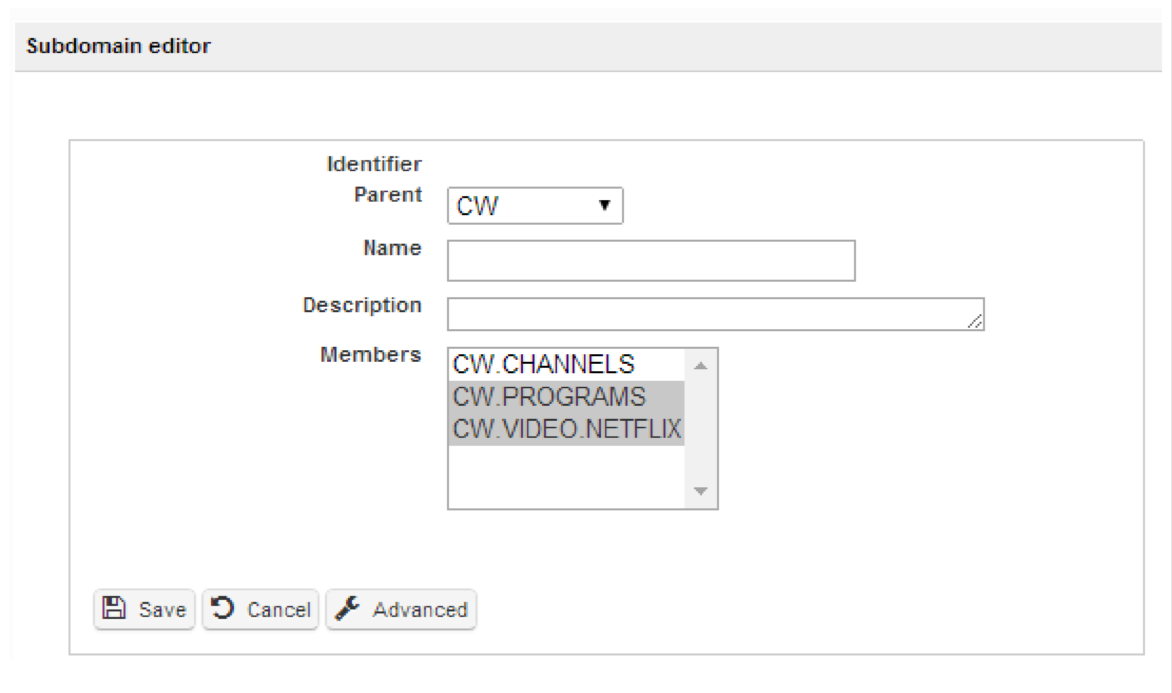
Create a new CROSSDOMAIN subdomain

To create a new CROSSDOMAIN subdomain, select CROSSDOMAIN as value of the *Subdomain type* filter.

All information but description must be filled. The name of the new subdomain will be built by merging the identifier of the parent subdomain and the string that you will type in the appropriate field.

Select all the subdomains that have to be members of the new subdomain.

⚠ **Warning**
Subdomain name can not contain '.', '/' and '-' characters



Subdomain editor

Identifier

Parent: CW

Name:

Description:

Members: CW.CHANNELS, CW.PROGRAMS, CW.VIDEO.NETFLIX

Subdomain definitions: Create a new CROSSDOMAIN subdomain.

Subdomain rules

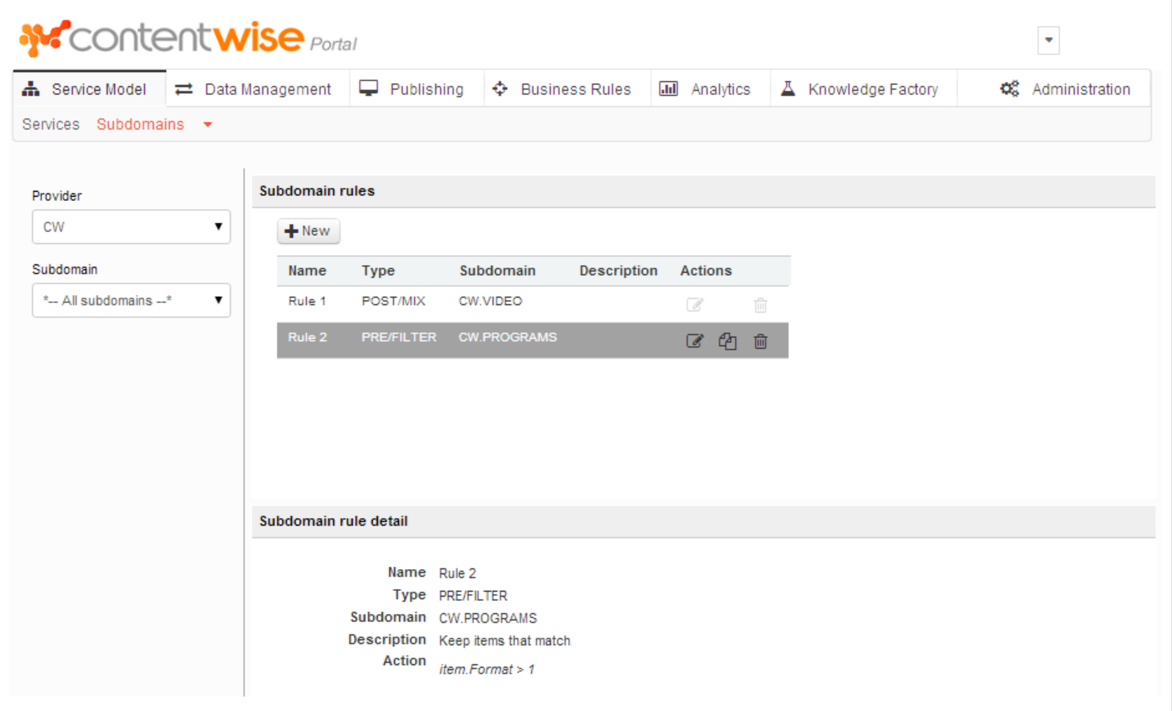
This page provides an overview about how subdomain rules are configured in the system and provides access to the pages that allow to create and edit subdomain rules.

The page contains the following portlets:

- Subdomain rules: lists the subdomain rules that are in the system and provides access to new/edit subdomain rule page.
- Subdomain rule detail: provides details about the selected subdomain rule.

See create a new rule to create a new subdomain rule.

If a provider filter is not selected, you will be prompted to select one.
You can filter by subdomain to view only subdomain rules of a specific subdomain. If no subdomain filter is selected, all subdomain rules are shown.



Provider: CW

Subdomain: *-- All subdomains --*

Subdomain rules

Name	Type	Subdomain	Description	Actions
Rule 1	POST/MIX	CW.VIDEO		
Rule 2	PRE/FILTER	CW.PROGRAMS		

Subdomain rule detail

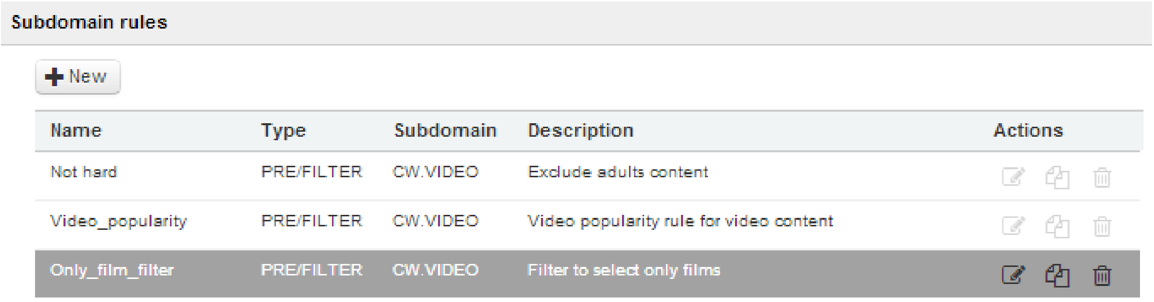
Name: Rule 2
Type: PRE/FILTER
Subdomain: CW.PROGRAMS
Description: Keep items that match
Action: *item.Format > 1*

Subdomain rules: list of configured subdomain rules and related configurations.

Subdomain rules

This portlet lists all available subdomain rules. For each subdomain rule, the table shows:

- *Name*: the rule unique identifier.
- *Type*: the rule type.
- *Subdomain*: the subdomain for which the rule has been defined.
- *Description*: a short rule description (optional).
- *Actions*: you can:
  - *edit*: edit the rule information and its configuration.
  - *clone*: it creates a copy of the rule.
  - *delete*: deletes the subdomain rule.



Subdomain rules

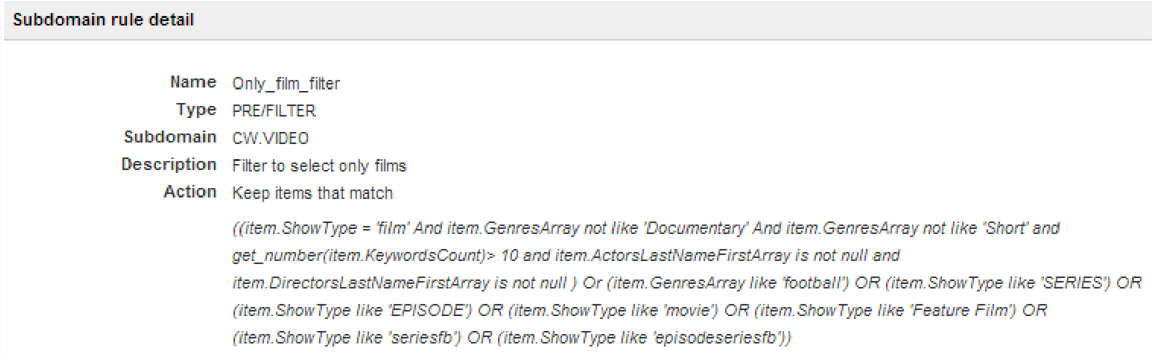
Name	Type	Subdomain	Description	Actions
Not hard	PRE/FILTER	CW.VIDEO	Exclude adults content	
Video_popularity	PRE/FILTER	CW.VIDEO	Video popularity rule for video content	
Only_film_filter	PRE/FILTER	CW.VIDEO	Filter to select only films	

Subdomain rules: Subdomain rules portlet, list of configured subdomain rules.

Subdomain rule detail

This portlet shows the information of the subdomain rule that has been selected in the Subdomain rules portlet.

- *Name*: the rule unique identifier.
- *Type*: the rule type.
- *Subdomain*: the subdomain for which the rule has been defined.
- *Description*: a short rule description (optional).
- *Action*: the rule definition or the appeal weights, depending on the rule type.



Subdomain rule detail

Name	Only_film_filter
Type	PRE/FILTER
Subdomain	CW.VIDEO
Description	Filter to select only films
Action	Keep items that match *((item.ShowType = 'film' And item.GenresArray not like 'Documentary' And item.GenresArray not like 'Short' and get_number(item.KeywordsCount)> 10 and item.ActorsLastNameFirstArray is not null and item.DirectorsLastNameFirstArray is not null) Or (item.GenresArray like 'football') OR (item.ShowType like 'SERIES') OR (item.ShowType like 'EPISODE') OR (item.ShowType like 'movie') OR (item.ShowType like 'Feature Film') OR (item.ShowType like 'seriesfb') OR (item.ShowType like 'episodeseriesfb'))*

Subdomain rules: Subdomain rule detail portlet.

Create New Rule

This section shows how to add a new subdomain rule in the system.

To create a new subdomain rule click the *New* button in the Subdomain rules portlet.

You will be prompted to select:

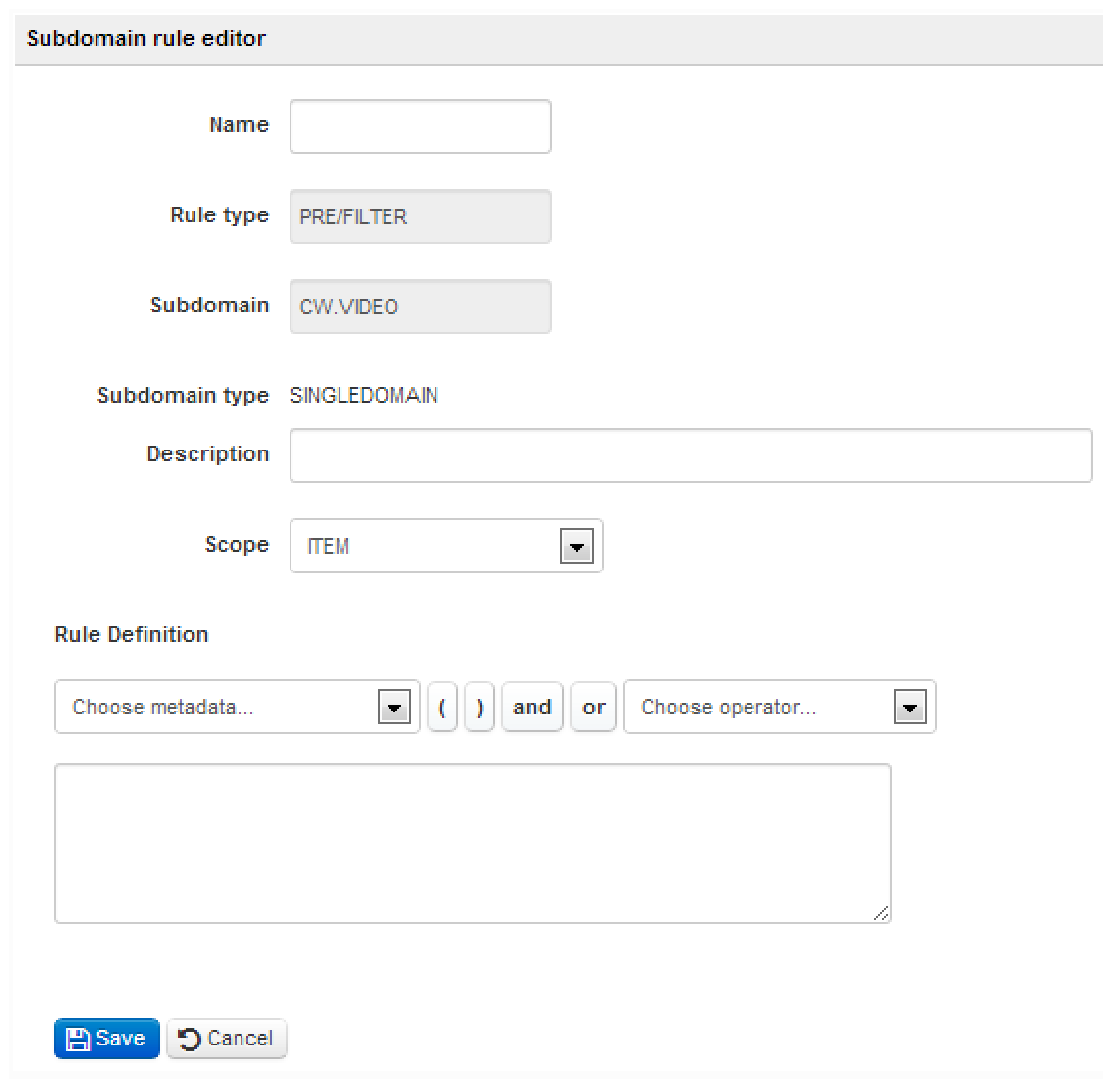
- A provider.
- A subdomain for the new rule.

According to the type of the selected subdomain, a different subdomain rule configuration is required.

Subdomain rule configuration for SINGLEDOMAIN subdomain

To create a new subdomain rule associated to a SINGLEDOMAIN subdomain, you are required to provided:

- *Name*: the subdomain rule name. Valid characters are [A-Za-z0-9], **-**, _
- *Description*: an optional description of the rule.
- *Scope*: specified if the rule will filter on items or users.
- *Rule definition*: a SQL-like condition to filter on item or user metadata, according to the scope of the rule.



Subdomain rule editor

Name	
Rule type	PRE/FILTER
Subdomain	CW.VIDEO
Subdomain type	SINGLEDOMAIN
Description	
Scope	ITEM

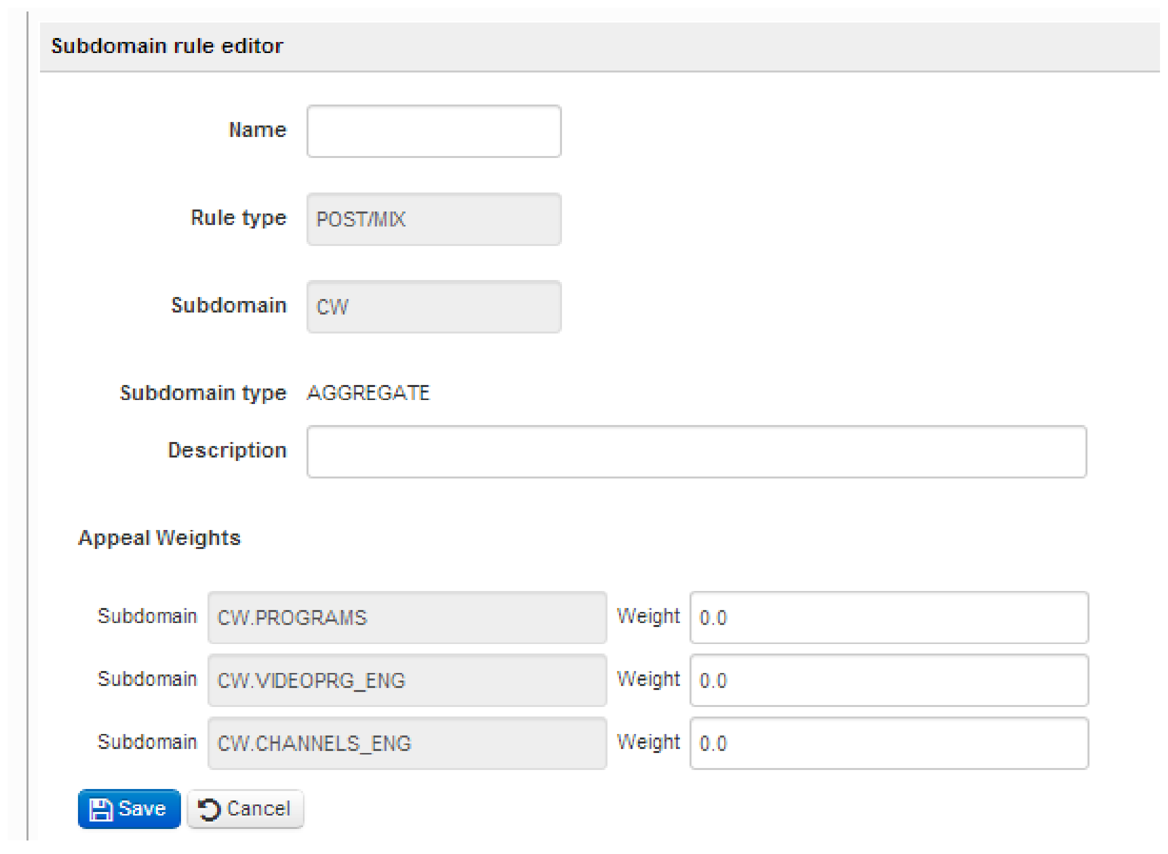
Rule Definition

Subdomain rules: Subdomain rule editor, SINGLEDOMAIN subdomain.

Subdomain rule configuration for AGGREGATE subdomain

- *Name*: the subdomain rule name. Valid characters are [A-Za-z0-9], **-**, _
- *Description*: an optional description of the rule.
- *Appeal weights*: specifies how recommendation results have to be mixed from children subdomains.

> **Important Note**
> The sum of the weights must be equal to 1.



Subdomain rule editor

Name

Rule type POST/MIX

Subdomain CW

Subdomain type AGGREGATE

Description

Appeal Weights

Subdomain CW.PROGRAMS Weight 0.0

Subdomain CW.VIDEOPRG_ENG Weight 0.0

Subdomain CW.CHANNELS_ENG Weight 0.0

Subdomain rules: Subdomain rule editor, AGGREGATE subdomain.

Data Management

This section describes functionalities provided by the *Data Management* section of the ContentWise Portal.

The table below lists the pages of the section.

Page	Description
Data Import	Configures ETL processes to import data in the system.
Tasks	Configures the batch processes of the system.
Metadata Enhancer	Configures the metadata enhancer process (MDE).
Recommendation Models	Manages the recommendation models generated by the batch process.
Dynamic Lists	Configure the dynamic lists
Item Hierarchy	Configure the item hierarchies. See Item hierarchies

Data Import

This section describes functionalities provided by *Data Import* menu.

See ETL to have an overview of ETL processes.

Data Import menu gives access to the following pages:

Page	Description
ETL Configuration	It is the landing page of the section. It provides an overview of the ETLs configured in the system and allows to manage their configurations.
ETL Execution	It is the page dedicated to the execution and the schedule of ETLs. In this page, it is possible to analyze ETL logs.

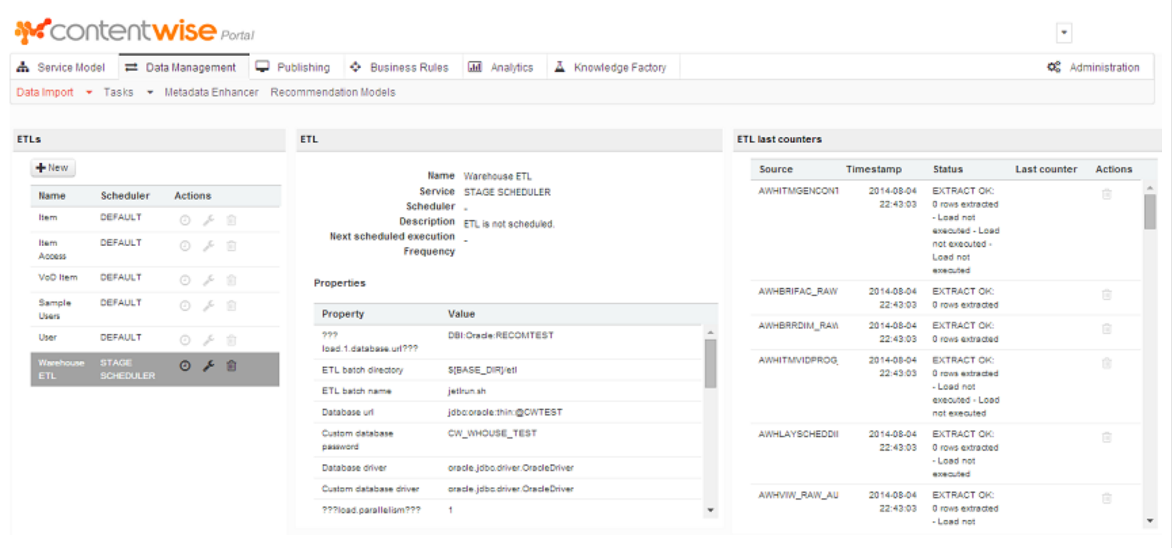
ETL Configuration

This page provides an overview about how ETLs are configured in the system and provides access to the pages that allow to create and edit ETLs.

The page contains the following portlets:

- ETLs: lists the ETLs that are in the system and provides access to new/edit/configure ETL pages.
- ETL: provides a detailed overview of the selected ETL.
- ETL last counters: lists and manages the last counters of the selected ETL.

See create a new ETL to create a new ETL.
See configure an ETL to configure an ETL.



ETLs

Name	Scheduler	Actions
Item	DEFAULT	
Item Access	DEFAULT	
VoD Item	DEFAULT	
Sample Users	DEFAULT	
User	DEFAULT	
Warehouse ETL	STAGE SCHEDULER	

ETL

Name	Warehouse ETL
Service	STAGE SCHEDULER
Scheduler	-
Description	ETL is not scheduled.
Next scheduled execution	-
Frequency	

Properties

Property	Value
??? load.1.database.url???	DBI:Oracle:RECOMTEST
ETL batch directory	$[BASE_DIR]/etl
ETL batch name	jetlrun.sh
Database url	jdbc:oracle:thin:@CWTEST
Custom database password	CW_WHOUSE_TEST
Database driver	oracle.jdbc.driver.OracleDriver
Custom database driver	oracle.jdbc.driver.OracleDriver
???load.parallelism???	1

ETL last counters

Source	Timestamp	Status	Last counter	Actions
AWHITMGENCONT	2014-08-04 22:43:03	EXTRACT OK: 0 rows extracted - Load not executed - Load not executed - Load not executed		
AWHBRIFAC_RAW	2014-08-04 22:43:03	EXTRACT OK: 0 rows extracted		
AWHBRRDIM_RAW	2014-08-04 22:43:03	EXTRACT OK: 0 rows extracted		
AWHITMVIDPROG	2014-08-04 22:43:03	EXTRACT OK: 0 rows extracted - Load not executed - Load not executed		
AWHLAYSCHEDDI	2014-08-04 22:43:03	EXTRACT OK: 0 rows extracted - Load not executed		
AWHVIW_RAW_AU	2014-08-04 22:43:03	EXTRACT OK: 0 rows extracted - Load not		

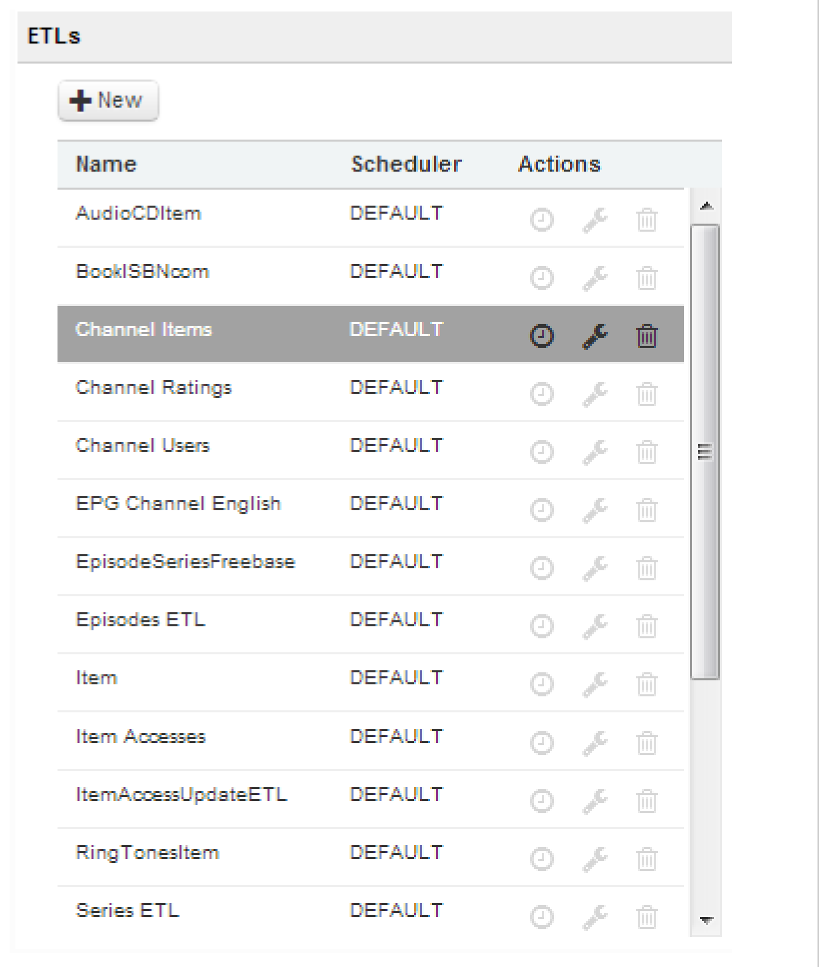
ETL Configuration: list of configured ETLs and related configurations.

ETLs

This portlet lists the ETLs that are available in the system and provides access to ETL configuration pages and to the new etl page.

For each ETL, the table shows:

- *Name*: the ETL name.
- *Actions*: you can:
    - *edit schedule*: edit the schedule configuration.
    - *edit properties*: edit the execution properties. See configure an ETL.



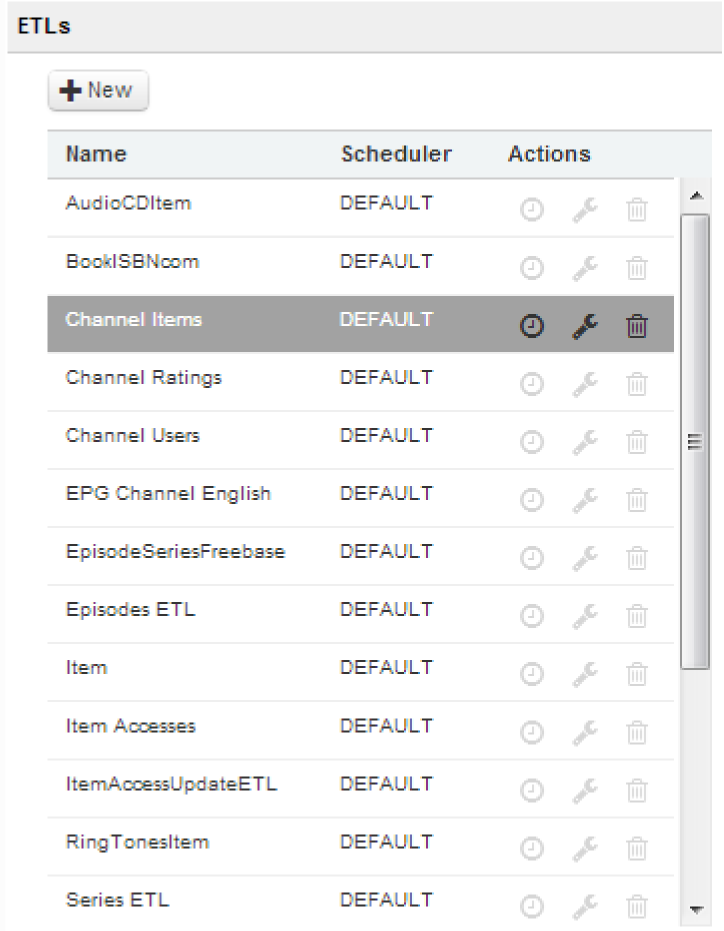
Name	Scheduler	Actions
AudioCDItem	DEFAULT	
BookISBNcom	DEFAULT	
Channel Items	DEFAULT	
Channel Ratings	DEFAULT	
Channel Users	DEFAULT	
EPG Channel English	DEFAULT	
EpisodeSeriesFreebase	DEFAULT	
Episodes ETL	DEFAULT	
Item	DEFAULT	
Item Accesses	DEFAULT	
ItemAccessUpdateETL	DEFAULT	
RingTonesItem	DEFAULT	
Series ETL	DEFAULT	

ETL Configuration: ETLs portlet.

ETL

This portlet provides details about the ETL that has been selected in the ETLs portlet.

- *Name*: the ETL name.
- *Type*: the ETL type.
- *Service*: the service for which the ETL has been defined.
- *Description*: a short description of the ETL.
- *Next scheduled execution*: the timestamp of the next scheduled execution, if any.
- *Properties*: a table that lists the execution properties of the ETL.



ETLs

Name	Scheduler	Actions
AudioCDItem	DEFAULT	
BookISBNcom	DEFAULT	
Channel Items	DEFAULT	
Channel Ratings	DEFAULT	
Channel Users	DEFAULT	
EPG Channel English	DEFAULT	
EpisodeSeriesFreebase	DEFAULT	
Episodes ETL	DEFAULT	
Item	DEFAULT	
Item Accesses	DEFAULT	
ItemAccessUpdateETL	DEFAULT	
RingTonesItem	DEFAULT	
Series ETL	DEFAULT	

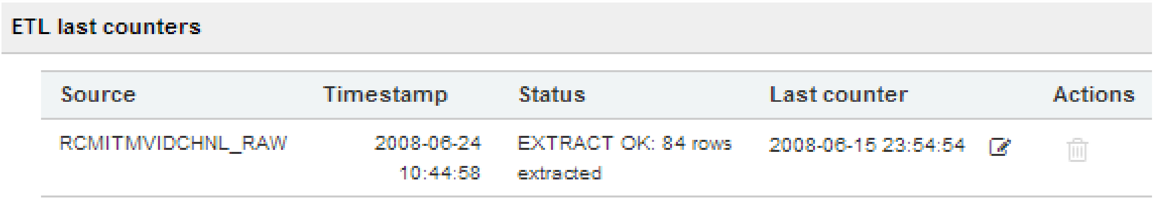
ETL Configuration: ETL portlet.

ETL last counters

This portlet list the last counters of the ETL that has been selected in the ETLs portlet.

For each last counter, the table shows:

- *Source*: the source associated to the dataset bound to the ETL. See ETL for a description about datasets.
- *Timestamp*: the timestamp of the last ETL execution.
- *Status*: a summary of the ETL execution associated to the timestamp.
- *Last counter*: the last counter associated to the source. See ETL#Lascounter for a description about last counters.
- *Actions*: you can:
  - *delete*: remove the last counter.



ETL last counters

Source	Timestamp	Status	Last counter	Actions
RCMITMVIDCHNL_RAW	2008-06-24 10:44:58	EXTRACT OK: 84 rows extracted	2008-06-15 23:54:54	

ETL Configuration: ETL last counters portlet.

Create a new ETL

To create a new ETL click the *New* button in the ETLs portlet.

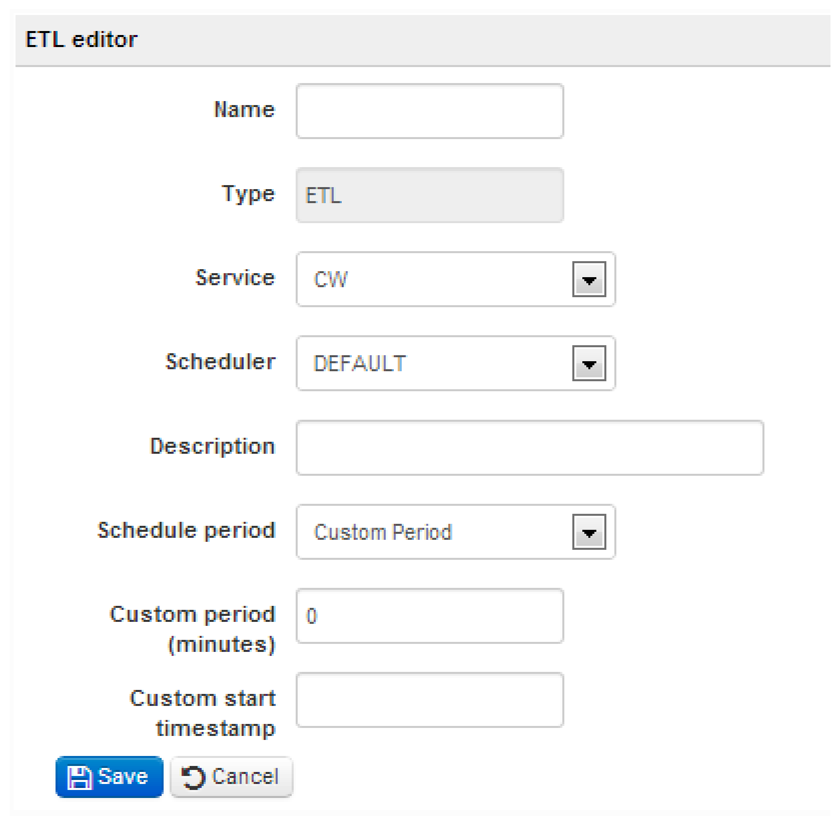
You will be prompted to select a provider.

To create a new ETL, you are required to provide:

- *Name*: the ETL identifier. Valid characters are [A-Za-z0-9], **-**, , (_*space*)
- *Service*: the service for which data will be imported by the ETL.

- *Scheduler* : the scheduler that runs the ETL.
- *Description*: a short ETL description (not mandatory).
- *Schedule period*: specifies the ETL schedule configuration (not mandatory).
    - *Custom period*: specifies the interval execution time of the ETL and a starting timestamp.
    - *Each day*: to execute the ETL once a day. Requires the specification of the schedule time.
    - *Each week*: to execute the ETL once a week. Requires the specification of the schedule day and time.
    - *Each month*: to execute the ETL once a month. Requires the specification of the schedule day and time.

> **Important Note**
> After an ETL has been created, it is necessary to configure its execution properties (see configure an etl).



ETL editor

Name	
Type	ETL
Service	CW
Scheduler	DEFAULT
Description	
Schedule period	Custom Period
Custom period (minutes)	0
Custom start timestamp	

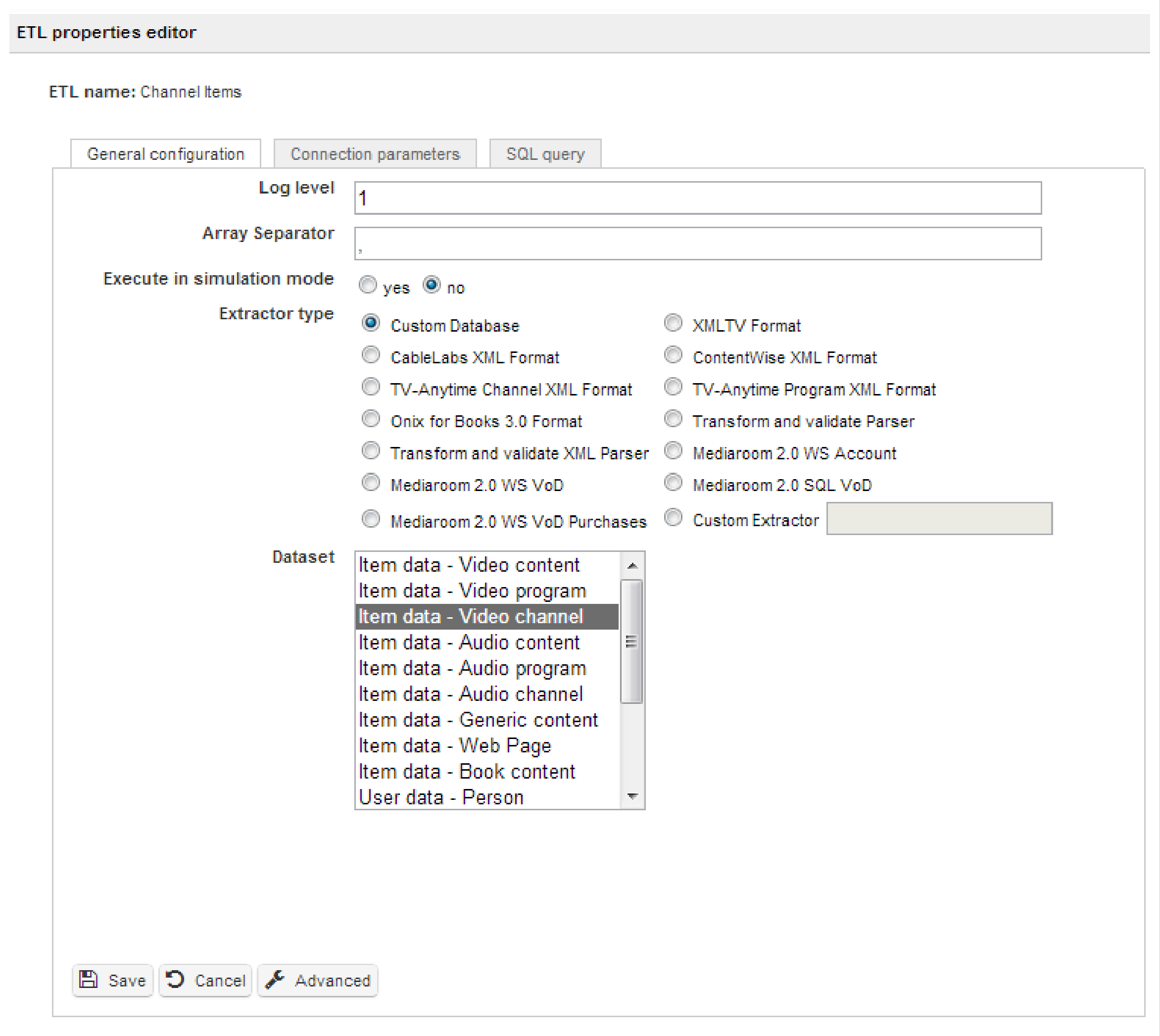
ETL Configuration: ETL editor portlet.

Configure an ETL

The execution properties of an ETL specify all the details required to import the desired data in the system.

To configure an ETL, click the *edit properties* button associated to ETL to be configured in the ETLs portlet.

See ETL for a description of the properties to configure.



ETL name: Channel Items

Log level: 1

Array Separator: ,

Execute in simulation mode: ○ yes ● no

Extractor type:

● Custom Database	○ XMLTV Format
○ CableLabs XML Format	○ ContentWise XML Format
○ TV-Anytime Channel XML Format	○ TV-Anytime Program XML Format
○ Onix for Books 3.0 Format	○ Transform and validate Parser
○ Transform and validate XML Parser	○ Mediaroom 2.0 WS Account
○ Mediaroom 2.0 WS VoD	○ Mediaroom 2.0 SQL VoD
○ Mediaroom 2.0 WS VoD Purchases	○ Custom Extractor

Dataset:

- Item data - Video content
- Item data - Video program
- Item data - Video channel
- Item data - Audio content
- Item data - Audio program
- Item data - Audio channel
- Item data - Generic content
- Item data - Web Page
- Item data - Book content
- User data - Person

ETL Configuration: ETL properties editor portlet.

ETL Execution

This page provides an overview of ETLs executions, allows to manage ETL executions, and to view logs.

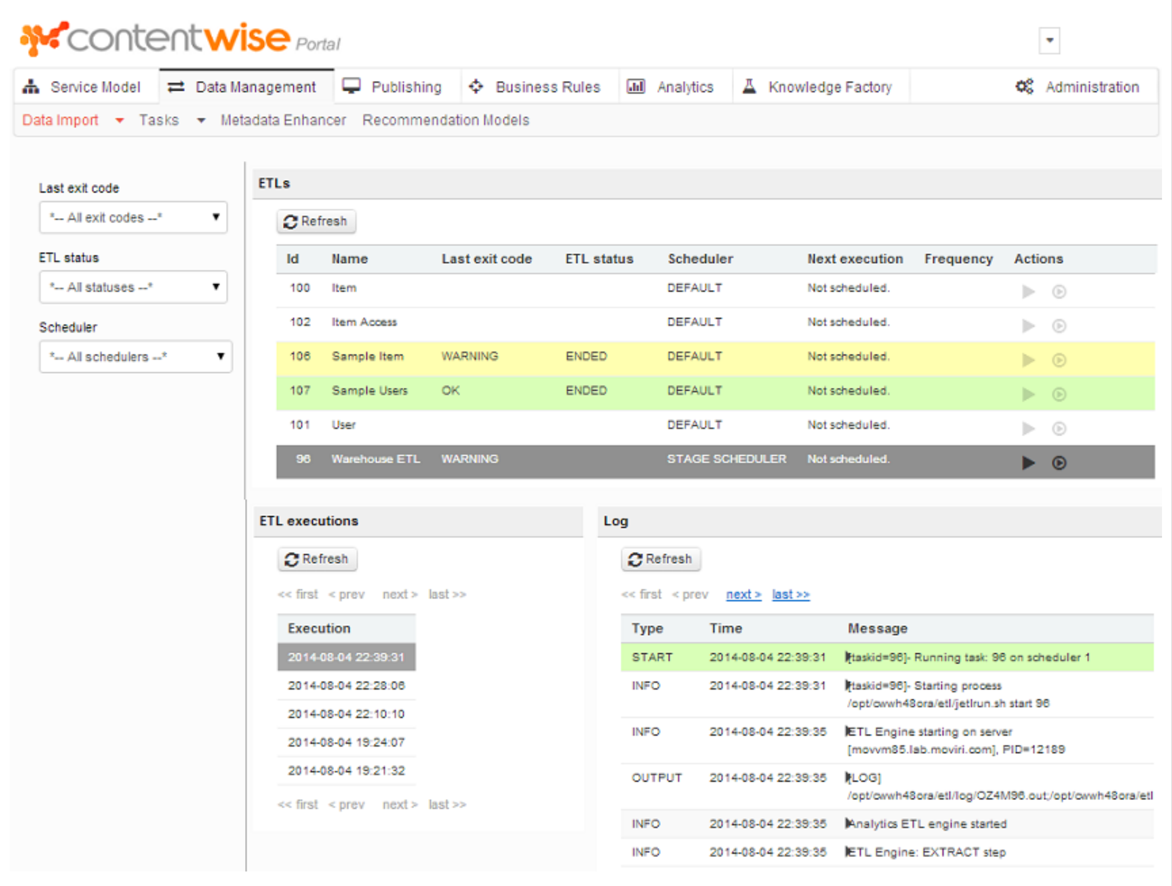
The page contains the following portlets:

- ETLs: lists the available ETLs and provides functionalities to manage ETL execution/schedule operations.
- ETL executions: lists the ETL executions.
- Log: shows the log of the ETL execution selected in the ETL executions portlet.

You can filter ETL executions by:

- Last exit code.
- ETL status.

If no filters are selected, all the ETLs are shown.



ETLs

Id	Name	Last exit code	ETL status	Scheduler	Next execution	Frequency	Actions
100	Item			DEFAULT	Not scheduled.		▶ ⦿
102	Item Access			DEFAULT	Not scheduled.		▶ ⦿
106	Sample Item	WARNING	ENDED	DEFAULT	Not scheduled.		▶ ⦿
107	Sample Users	OK	ENDED	DEFAULT	Not scheduled.		▶ ⦿
101	User			DEFAULT	Not scheduled.		▶ ⦿
96	Warehouse ETL	WARNING		STAGE SCHEDULER	Not scheduled.		▶ ⦿

ETL executions

Execution
2014-08-04 22:39:31
2014-08-04 22:28:06
2014-08-04 22:10:10
2014-08-04 19:24:07
2014-08-04 19:21:32

Log

Type	Time	Message
START	2014-08-04 22:39:31	[taskid=96]- Running task: 96 on scheduler 1
INFO	2014-08-04 22:39:31	[taskid=96]- Starting process /opt/cwwh48ora/etl/jetlrun.sh start 96
INFO	2014-08-04 22:39:35	ETL Engine starting on server [movvm85.lab.moviri.com], PID=12189
OUTPUT	2014-08-04 22:39:35	[LOG] /opt/cwwh48ora/etl/log/OZ4M96.out;/opt/cwwh48ora/etl
INFO	2014-08-04 22:39:35	Analytics ETL engine started
INFO	2014-08-04 22:39:35	ETL Engine: EXTRACT step

ETL Execution: list of ETL executions and related logs.

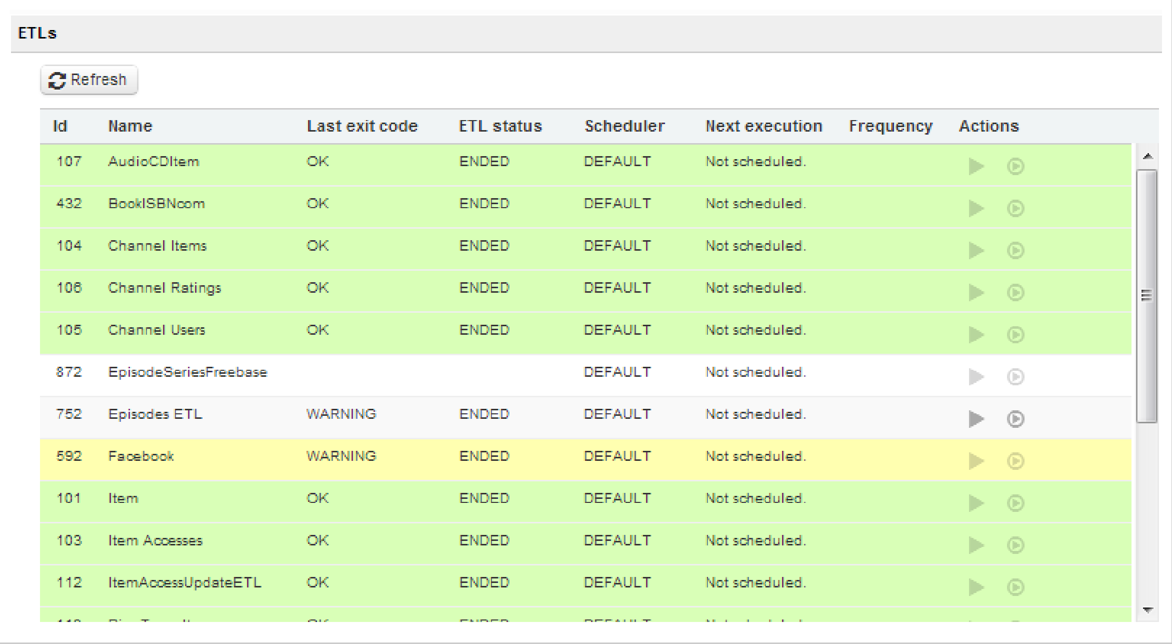
ETLs

This portlet lists the ETLs that are available in the system and provides functionalities to run/schedule/kill/stop ETLs.

> **Important Note**
> Table rows are coloured according to last ETL execution status:
>
> - *Green*: last ETL execution ended without warnings or errors.
> - *Yellow*: last ETL execution ended with warnings.
> - *Red*: last ETL execution failed.
> - *Blue*: ETL is currently running.
> - *White*: ETL has never been executed.

For each ETL, the table shows:

- *ETL id*: the ETL internal identifier.
- *Name*: the ETL name.
- *Last exit code*: the exit code of the last ETL execution (if any).
- *ETL status*: the current status of the ETL.
- *Next execution*: the next scheduled execution (if any).
- *Frequency*: the frequency in which the ETL is executed (if any).
- *Actions*: you can (depending on current ETL status):
  - *run*: immediately starts the execution of the ETL.
  - *kill*: terminates the current execution of the ETL.
  - *schedule*: schedules the ETL to be executed according to its schedule configuration.
  - *unschedule*: unschedules the ETL. ETL is removed from scheduled tasks and will not be automatically executed according to its schedule configuration.



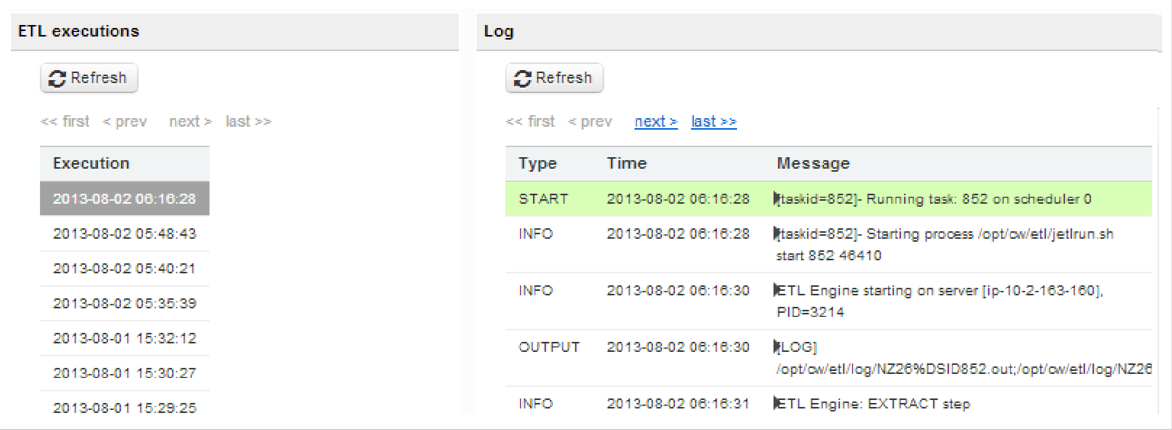
ETLs

Id	Name	Last exit code	ETL status	Scheduler	Next execution	Frequency	Actions
107	AudioCDItem	OK	ENDED	DEFAULT	Not scheduled.		
432	BookISBNcom	OK	ENDED	DEFAULT	Not scheduled.		
104	Channel Items	OK	ENDED	DEFAULT	Not scheduled.		
106	Channel Ratings	OK	ENDED	DEFAULT	Not scheduled.		
105	Channel Users	OK	ENDED	DEFAULT	Not scheduled.		
872	EpisodeSeriesFreebase			DEFAULT	Not scheduled.		
752	Episodes ETL	WARNING	ENDED	DEFAULT	Not scheduled.		
592	Facebook	WARNING	ENDED	DEFAULT	Not scheduled.		
101	Item	OK	ENDED	DEFAULT	Not scheduled.		
103	Item Accesses	OK	ENDED	DEFAULT	Not scheduled.		
112	ItemAccessUpdateETL	OK	ENDED	DEFAULT	Not scheduled.		

ETL Execution: ETLs portlet.

ETL executions and Logs

These portlets provide access to an ETL execution log. The *Refresh* button updates the portlet; use it when the ETL to monitor is currently running.



ETL executions

Refresh
<< first < prev next > last >>

Execution
2013-08-02 06:16:28
2013-08-02 05:48:43
2013-08-02 05:40:21
2013-08-02 05:35:39
2013-08-01 15:32:12
2013-08-01 15:30:27
2013-08-01 15:29:25

Log

Refresh
<< first < prev next > last >>

Type	Time	Message
START	2013-08-02 06:16:28	[taskid=852]- Running task: 852 on scheduler 0
INFO	2013-08-02 06:16:28	[taskid=852]- Starting process /opt/cw/etl/jetlrun.sh start 852 46410
INFO	2013-08-02 06:16:30	ETL Engine starting on server [ip-10-2-163-160], PID=3214
OUTPUT	2013-08-02 06:16:30	[LOG] /opt/cw/etl/log/NZ26%DSID852.out;/opt/cw/etl/log/NZ26
INFO	2013-08-02 06:16:31	ETL Engine: EXTRACT step

ETL Execution: ETL Executions and Log portlets.

Tasks

This section describes functionalities provided by *Tasks* menu.

See Task to have an overview of task processes.

Tasks menu gives access to the following pages:

Page	Description
Task Configuration	It is the landing page of the section. It provides an overview of the Tasks configured in the system and allows to manage their configurations.
Task Execution	It is the page dedicated to the execution and scheduling of Tasks. From this page, it is possible to analyze Task logs.

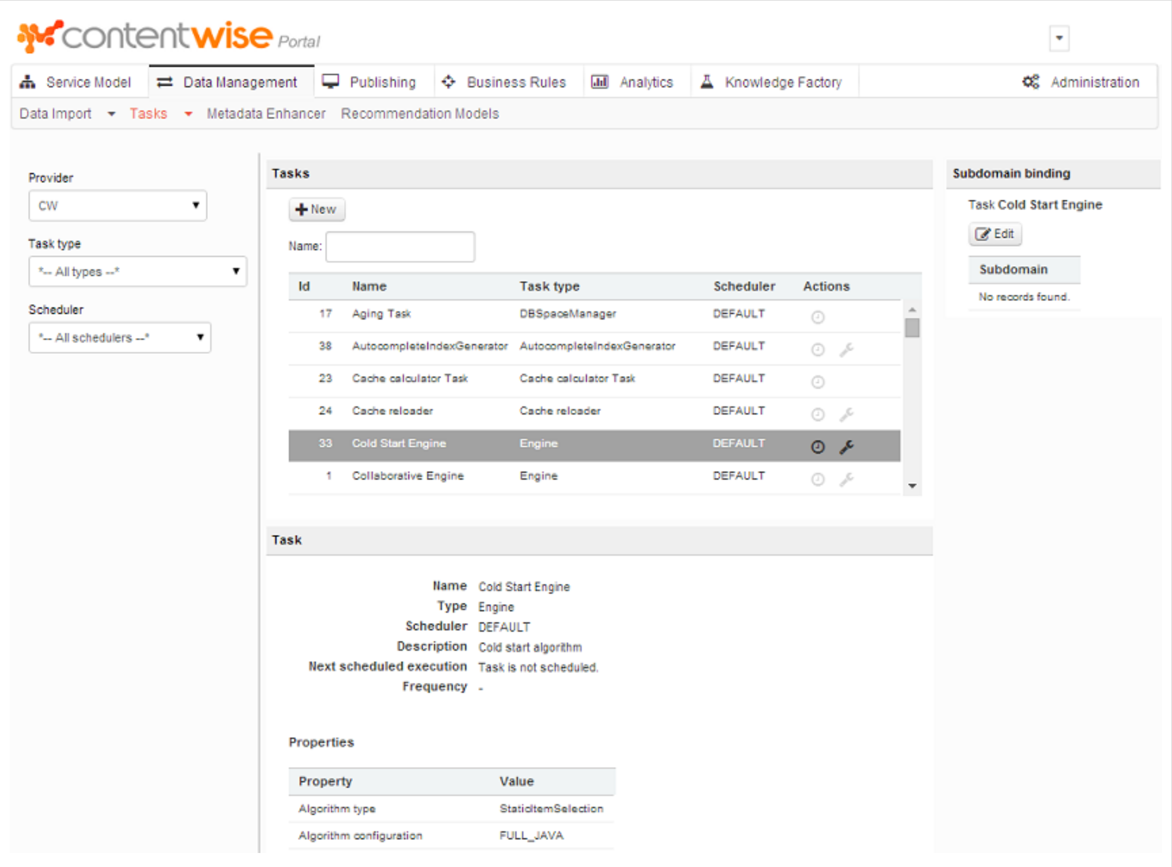
Task Configuration

This page provides an overview about how Tasks are configured in the system and provides access to the pages that allow to create and edit Tasks.

The page contains the following portlets:

- Tasks: lists the tasks that are in the system and provides access to new/edit/configure task pages.
- Task: provides a detailed overview of the selected task.
- Subdomain binding: lists and manages the subdomain-task bindings.

See create a new task to create a new task.
See configure a task to configure a task.



Provider: CW
Task type: *-- All types --*
Scheduler: *-- All schedulers --*

Tasks

Name:

Id	Name	Task type	Scheduler	Actions
17	Aging Task	DBSpaceManager	DEFAULT	
38	AutocompleteIndexGenerator	AutocompleteIndexGenerator	DEFAULT	
23	Cache calculator Task	Cache calculator Task	DEFAULT	
24	Cache reloader	Cache reloader	DEFAULT	
33	Cold Start Engine	Engine	DEFAULT	
1	Collaborative Engine	Engine	DEFAULT	

Task

Name: Cold Start Engine
Type: Engine
Scheduler: DEFAULT
Description: Cold start algorithm
Next scheduled execution: Task is not scheduled.
Frequency: -

Properties

Property	Value
Algorithm type	StaticItemSelection
Algorithm configuration	FULL_JAVA

Subdomain binding

Task Cold Start Engine

Subdomain
No records found.

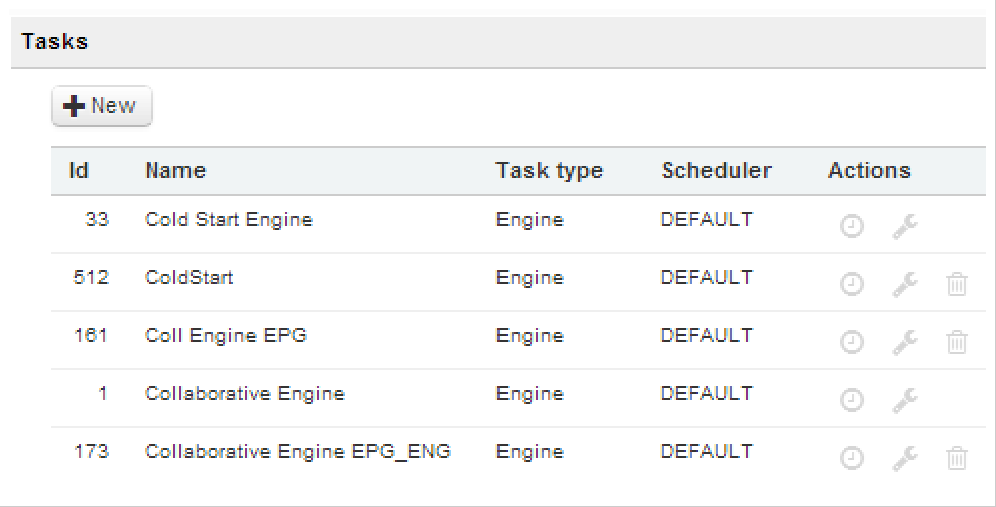
Task Configuration: list of configured tasks and related configurations.

Tasks

This portlet lists the tasks that are available in the system and provides access to task configuration pages and to the new task page.

For each task, the table shows:

- *ID*: the task internal identifier.
- *Name*: the task name.
- *Task type*: the type of the task.
- *Scheduler*: the scheduler on which the task runs
- *Actions*: you can:
  - *edit schedule*: edit the schedule configuration.
  - *edit properties*: edit the execution properties. Its availability depends on the task type: not all task types have an execution properties configuration.



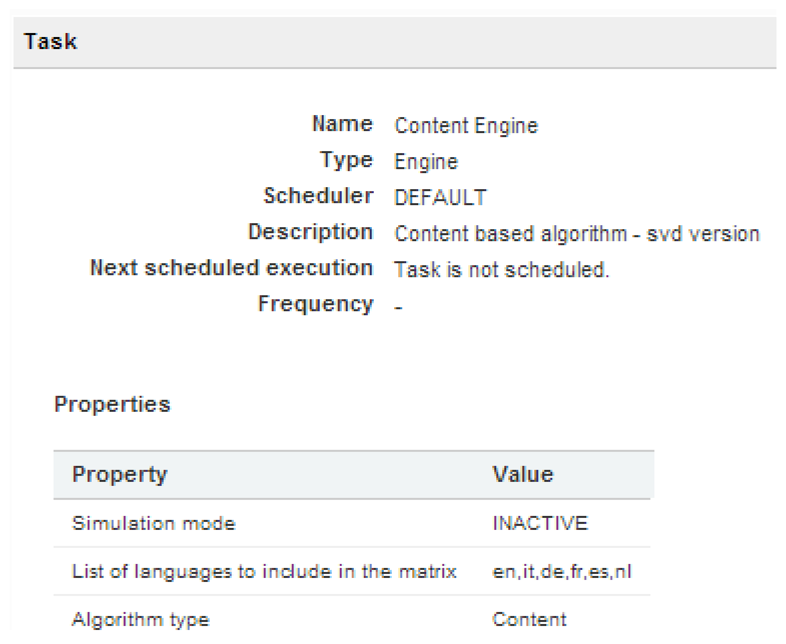
Id	Name	Task type	Scheduler	Actions
33	Cold Start Engine	Engine	DEFAULT	🕘 🔧
512	ColdStart	Engine	DEFAULT	🕘 🔧 🗑
161	Coll Engine EPG	Engine	DEFAULT	🕘 🔧 🗑
1	Collaborative Engine	Engine	DEFAULT	🕘 🔧
173	Collaborative Engine EPG_ENG	Engine	DEFAULT	🕘 🔧 🗑

Task Configuration: Tasks portlet.

Task

This portlet provides details about the task that has been selected in the Tasks portlet.

- *Name*: the task name.
- *Type*: the task type.
- *Scheduler*: the scheduler on which the task runs
- *Description*: a short description of the task.
- *Next scheduled execution*: the timestamp of the next scheduled execution (if any).
- *Properties*: a table that lists the execution properties of the task (if any).



Name Content Engine
Type Engine
Scheduler DEFAULT
Description Content based algorithm - svd version
Next scheduled execution Task is not scheduled.
Frequency -

Properties

Property	Value
Simulation mode	INACTIVE
List of languages to include in the matrix	en,it,de,fr,es,nl
Algorithm type	Content

Task Configuration: Task portlet.

Subdomain binding

This portlet provides information about the bindings between the task selected in Tasks portlet and the subdomains that are configured in the system.

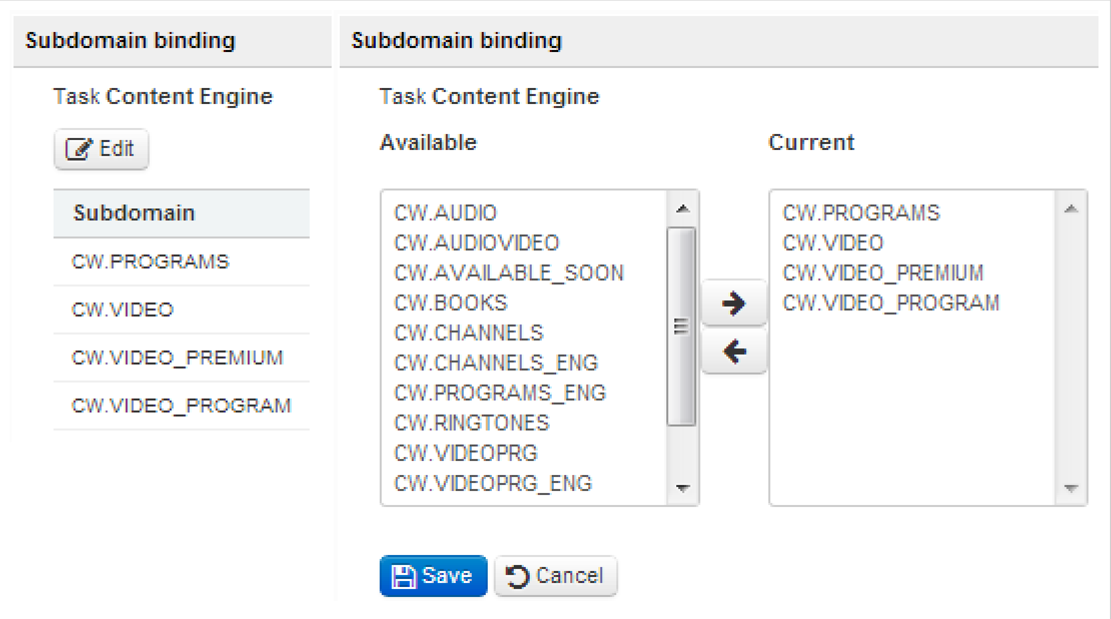
Subdomain binding configuration portlet availability depends on the type of the currently selected task.

The portlet lists the subdomains that are currently bound to the task.

To modify the current configuration:

1. Click *Edit*.
2. Select a subdomain and click on left or right arrows to modify the bindings. This step can be repeated for more than a subdomain.

3. Click *Save* to submit the new configuration, *Cancel* to undo the operation



Subdomain: CW.PROGRAMS, CW.VIDEO, CW.VIDEO_PREMIUM, CW.VIDEO_PROGRAM

Available: CW.AUDIO, CW.AUDIOVIDEO, CW.AVAILABLE_SOON, CW.BOOKS, CW.CHANNELS, CW.CHANNELS_ENG, CW.PROGRAMS_ENG, CW.RINGTONES, CW.VIDEOPRG, CW.VIDEOPRG_ENG

Current: CW.PROGRAMS, CW.VIDEO, CW.VIDEO_PREMIUM, CW.VIDEO_PROGRAM

Task Configuration: Subdomain binding portlet (view and edit).

Create a new task

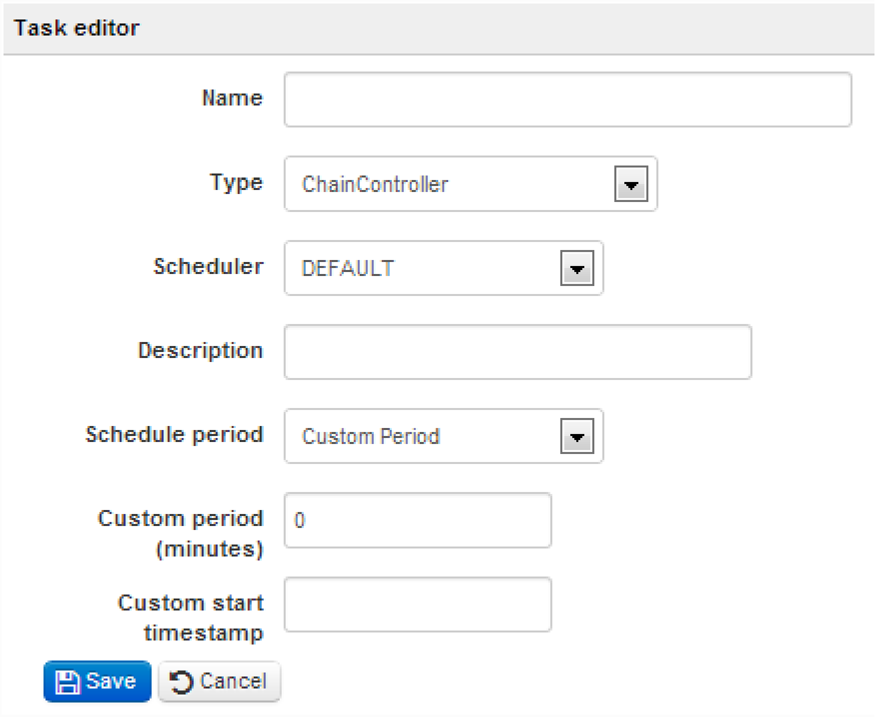
To create a new task click the *New* button in the Tasks portlet.

You will be prompted to select a provider.

To create a new task, you are required to provide:

- *Name*: the task identifier. Valid characters are [A-Za-z0-9], **-**, , (*_space*)
- *Type*: the task type.
- *Scheduler*: the scheduler on which the task runs
- *Description*: a short task description (not mandatory).
- *Schedule period*: specifies the task schedule configuration (not mandatory).
  - *Custom period*: specifies the interval execution time of the ETL and a starting timestamp.
  - *Each day*: to execute the task once a day. Requires the specification of the schedule time.
  - *Each week*: to execute the task once a week. Requires the specification of the schedule day and time.
  - *Each month*: to execute the task once a month. Requires the specification of the schedule day and time.

Important Note
After a task has been created, it may be necessary to configure its execution properties (see configure a task).



Task editor

Name

Type: ChainController

Scheduler: DEFAULT

Description

Schedule period: Custom Period

Custom period (minutes): 0

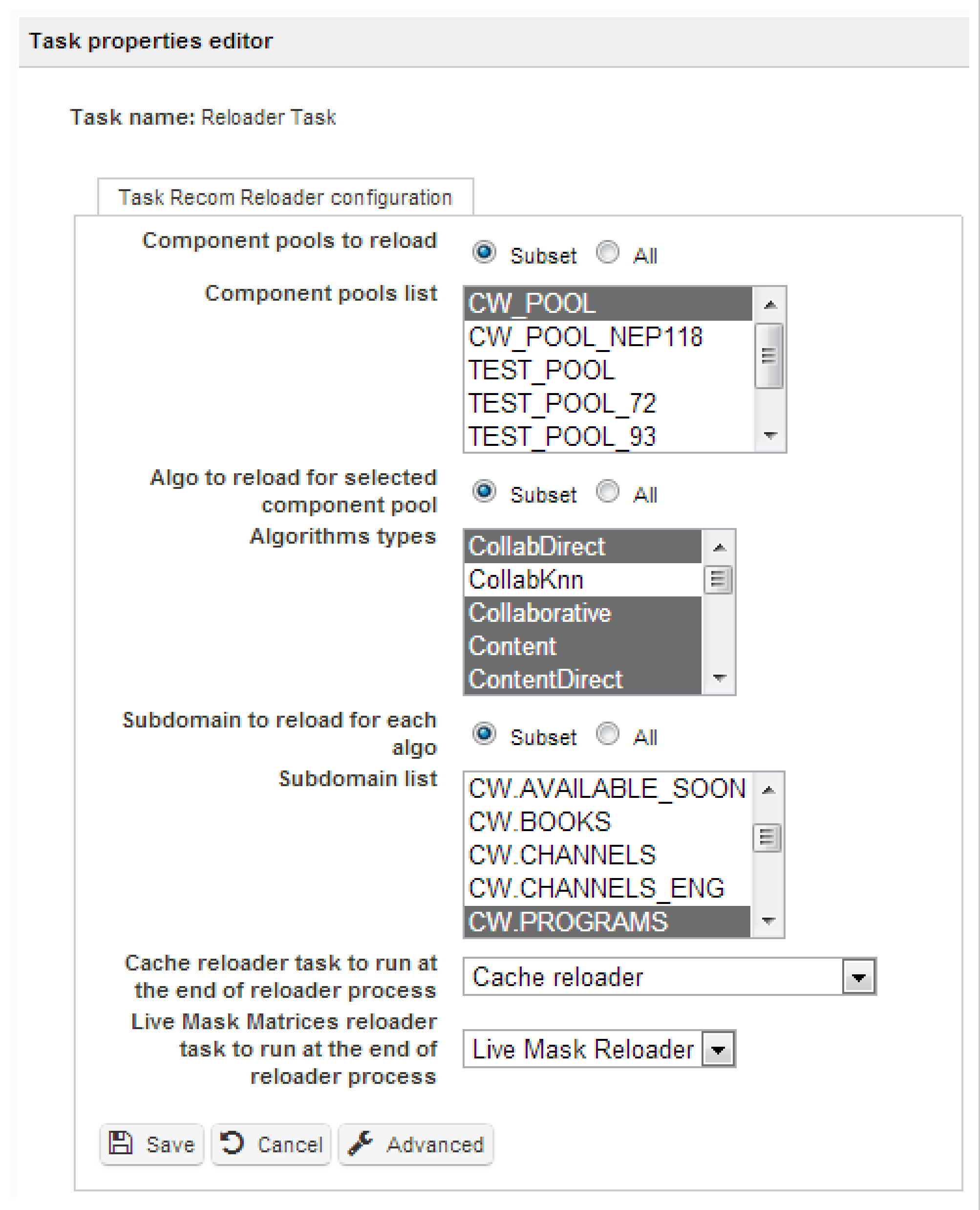
Custom start timestamp

Task Configuration: Task editor portlet.

Configure a task

To configure a task, click the *edit properties* button associated to task to be configured in the Tasks portlet.

The availability of the *edit properties* button depends on the task type. If the option is not available, no additional configuration is required.



Task properties editor

Task name: Reloader Task

Task Recom Reloader configuration

Component pools to reload ● Subset ○ All

Component pools list: CW_POOL, CW_POOL_NEP118, TEST_POOL, TEST_POOL_72, TEST_POOL_93

Algo to reload for selected component pool ● Subset ○ All

Algorithms types: CollabDirect, CollabKnn, Collaborative, Content, ContentDirect

Subdomain to reload for each algo ● Subset ○ All

Subdomain list: CW.AVAILABLE_SOON, CW.BOOKS, CW.CHANNELS, CW.CHANNELS_ENG, CW.PROGRAMS

Cache reloader task to run at the end of reloader process: Cache reloader

Live Mask Matrices reloader task to run at the end of reloader process: Live Mask Reloader

Task Configuration: Task properties editor portlet.

Task Execution

This page provides an overview of tasks executions, allows to manage task executions, and to view logs.

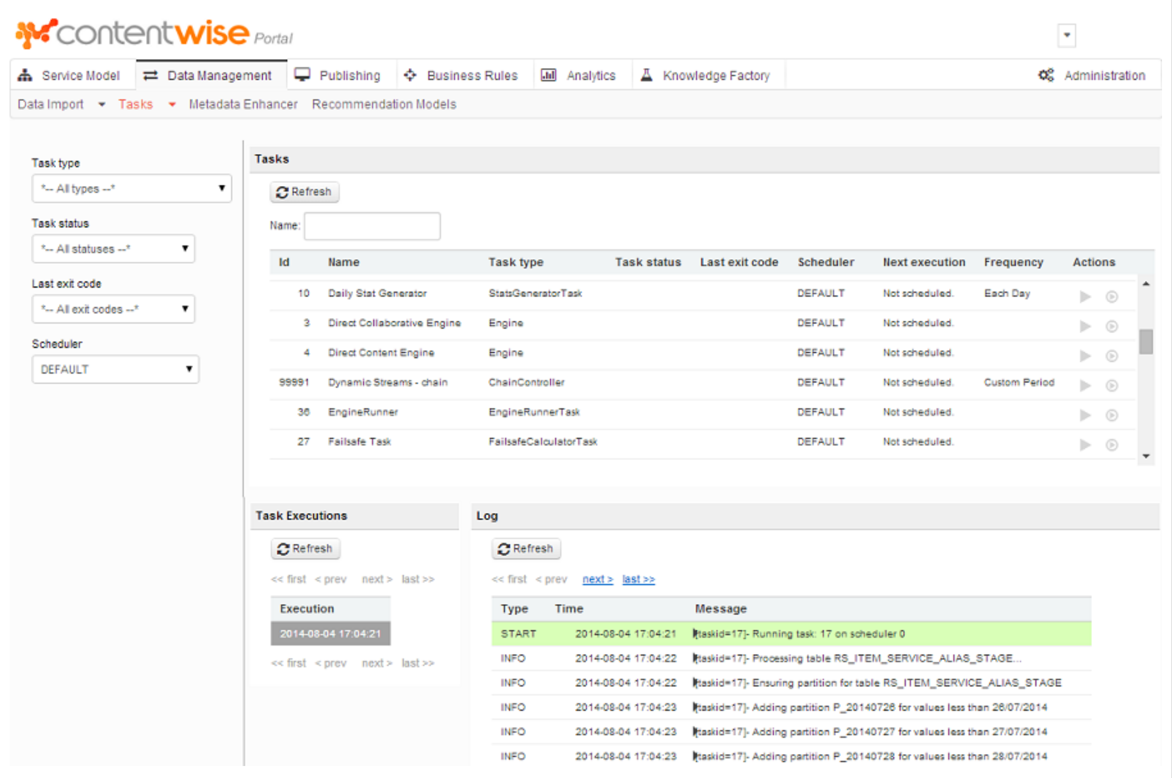
The page contains the following portlets:

- Task : lists the available tasks and provides functionalities to manage task execution/schedule operations.
- Task executions: lists the task executions.
- Log: shows the log of the task execution selected in the Task executions portlet.

You can filter task executions by:

- Task type.
- Task status.
- Last exit code.

If no filters are selected, all the tasks are shown.



Task type: *-- All types --*

Task status: *-- All statuses --*

Last exit code: *-- All exit codes --*

Scheduler: DEFAULT

Tasks

Name:

Id	Name	Task type	Task status	Last exit code	Scheduler	Next execution	Frequency	Actions
10	Daily Stat Generator	StatsGeneratorTask			DEFAULT	Not scheduled.	Each Day	▶ ⊙
3	Direct Collaborative Engine	Engine			DEFAULT	Not scheduled.		▶ ⊙
4	Direct Content Engine	Engine			DEFAULT	Not scheduled.		▶ ⊙
99991	Dynamic Streams - chain	ChainController			DEFAULT	Not scheduled.	Custom Period	▶ ⊙
36	EngineRunner	EngineRunnerTask			DEFAULT	Not scheduled.		▶ ⊙
27	Failsafe Task	FailsafeCalculatorTask			DEFAULT	Not scheduled.		▶ ⊙

Task Executions

Execution
2014-08-04 17:04:21

Log

Type	Time	Message
START	2014-08-04 17:04:21	[taskid=17]- Running task: 17 on scheduler 0
INFO	2014-08-04 17:04:22	[taskid=17]- Processing table RS_ITEM_SERVICE_ALIAS_STAGE...
INFO	2014-08-04 17:04:22	[taskid=17]- Ensuring partition for table RS_ITEM_SERVICE_ALIAS_STAGE
INFO	2014-08-04 17:04:23	[taskid=17]- Adding partition P_20140726 for values less than 26/07/2014
INFO	2014-08-04 17:04:23	[taskid=17]- Adding partition P_20140727 for values less than 27/07/2014
INFO	2014-08-04 17:04:23	[taskid=17]- Adding partition P_20140728 for values less than 28/07/2014

Task Execution: list of task executions and related logs.

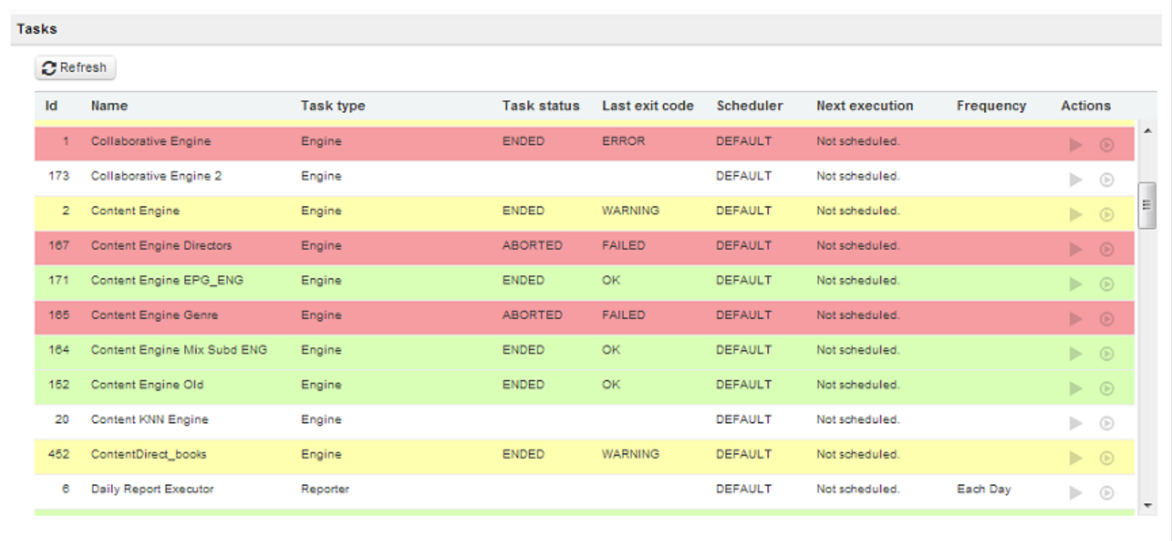
Tasks

This portlet lists the tasks that are available in the system and provides functionalities to run/schedule/kill/stop tasks.

> **Important Note**
> Table rows are coloured according to last task execution status:
>
> - *Green*: last task execution ended without warnings or errors.
> - *Yellow*: last task execution ended with warnings.
> - *Red*: last task execution failed.
> - *Blue*: task is currently running.
> - *White*: task has never been executed.

For each task, the table shows:

- *Task id*: the task internal identifier.
- *Name*: the task name.
- *Task type*: the task type.
- *Task status*: the current status of the task.
- *Last exit code*: the exit code of the last task execution (if any).
- *Next execution*: the next scheduled execution (if any).
- *Frequency*: the frequency in which the task is executed (if any).
- *Actions*: you can (depending on current task status):
  - *run*: immediately starts the execution of the task.
  - *kill*: terminates the current execution of the task.
  - *schedule*: schedules the task to be executed according to its schedule configuration.
  - *unschedule*: unschedules the task. task is removed from scheduled tasks and will not be automatically executed according to its schedule configuration.

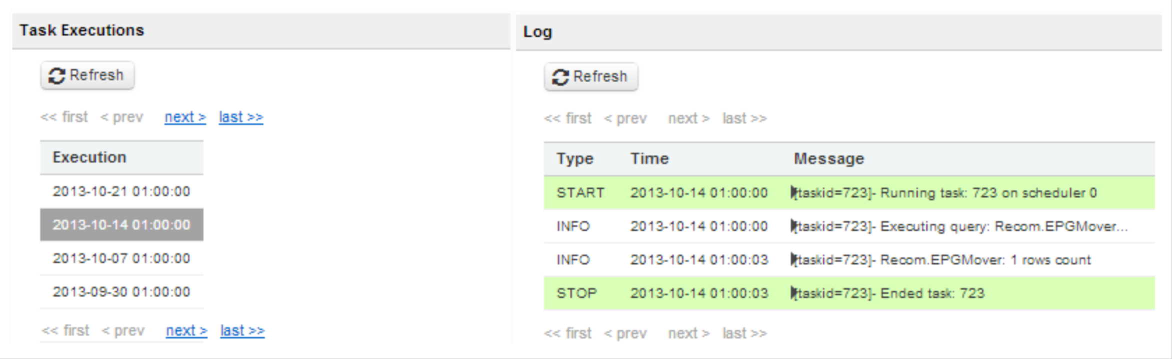


Id	Name	Task type	Task status	Last exit code	Scheduler	Next execution	Frequency	Actions
1	Collaborative Engine	Engine	ENDED	ERROR	DEFAULT	Not scheduled.		▶ ⊙
173	Collaborative Engine 2	Engine			DEFAULT	Not scheduled.		▶ ⊙
2	Content Engine	Engine	ENDED	WARNING	DEFAULT	Not scheduled.		▶ ⊙
167	Content Engine Directors	Engine	ABORTED	FAILED	DEFAULT	Not scheduled.		▶ ⊙
171	Content Engine EPG_ENG	Engine	ENDED	OK	DEFAULT	Not scheduled.		▶ ⊙
165	Content Engine Genre	Engine	ABORTED	FAILED	DEFAULT	Not scheduled.		▶ ⊙
164	Content Engine Mix Subd ENG	Engine	ENDED	OK	DEFAULT	Not scheduled.		▶ ⊙
152	Content Engine Old	Engine	ENDED	OK	DEFAULT	Not scheduled.		▶ ⊙
20	Content KNN Engine	Engine			DEFAULT	Not scheduled.		▶ ⊙
452	ContentDirect_books	Engine	ENDED	WARNING	DEFAULT	Not scheduled.		▶ ⊙
6	Daily Report Executor	Reporter			DEFAULT	Not scheduled.	Each Day	▶ ⊙

task Execution: tasks portlet.

Task executions and Logs

These portlets provide access to a task execution log. The *Refresh* button updates the portlet; use it when the task to monitor is currently running.



Task Executions
Refresh
<< first < prev next > last >>
Log
Refresh
<< first < prev next > last >>

Execution
2013-10-21 01:00:00
2013-10-14 01:00:00
2013-10-07 01:00:00
2013-09-30 01:00:00

Type	Time	Message
START	2013-10-14 01:00:00	[taskid=723]- Running task: 723 on scheduler 0
INFO	2013-10-14 01:00:00	[taskid=723]- Executing query: Recom.EPGMover...
INFO	2013-10-14 01:00:03	[taskid=723]- Recom.EPGMover: 1 rows count
STOP	2013-10-14 01:00:03	[taskid=723]- Ended task: 723

Task Execution: Task Executions and Log portlets.

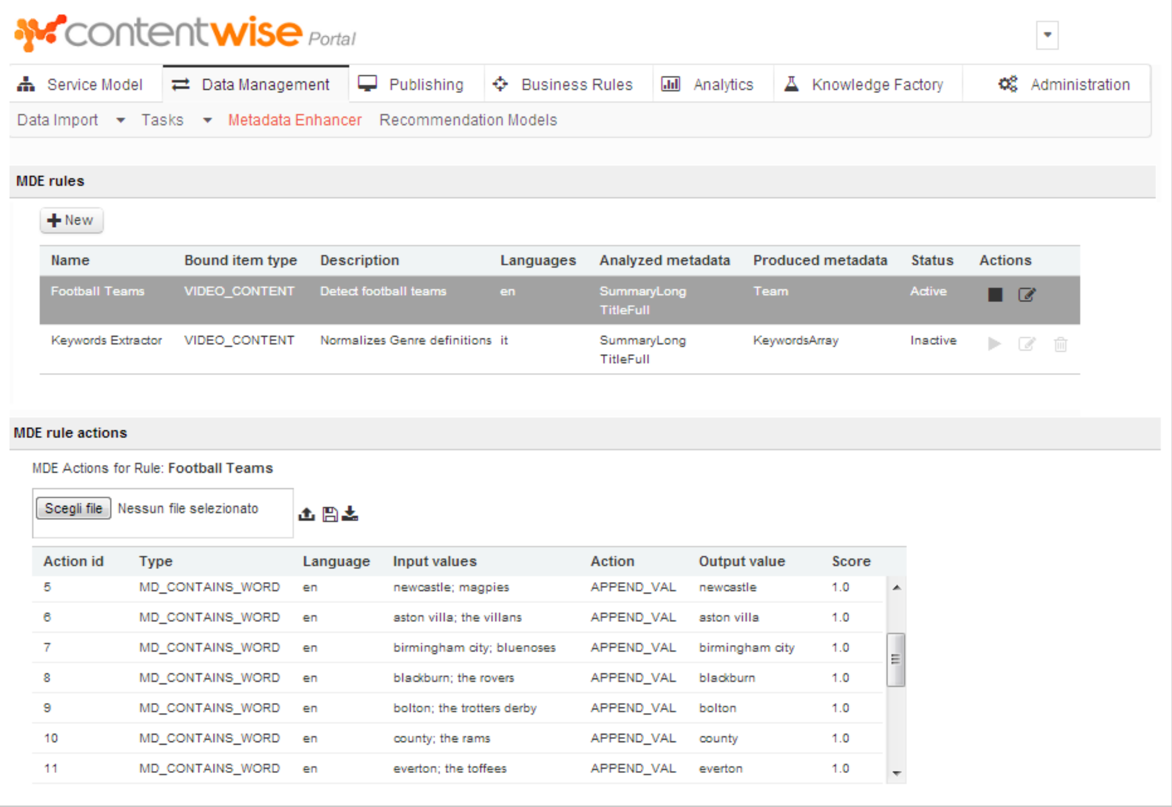
Metadata Enhancer

This page provides an overview of Metadata Enhancer (MDE) configuration and provides access to the page that allows to create MDE rules.

The page contains the following portlets:

- MDE rules: lists the MDE rules defined in the system and provides access to new rule page.
- MDE rule actions: lists the actions associated to the MDE rule currently selected in the MDE rules portlet.

See create a new rule to create a new MDE rule.



MDE rules

Name	Bound item type	Description	Languages	Analyzed metadata	Produced metadata	Status	Actions
Football Teams	VIDEO_CONTENT	Detect football teams	en	SummaryLong TitleFull	Team	Active	■ ✎
Keywords Extractor	VIDEO_CONTENT	Normalizes Genre definitions	it	SummaryLong TitleFull	KeywordsArray	Inactive	▶ ✎ 🗑

MDE rule actions

MDE Actions for Rule: **Football Teams**

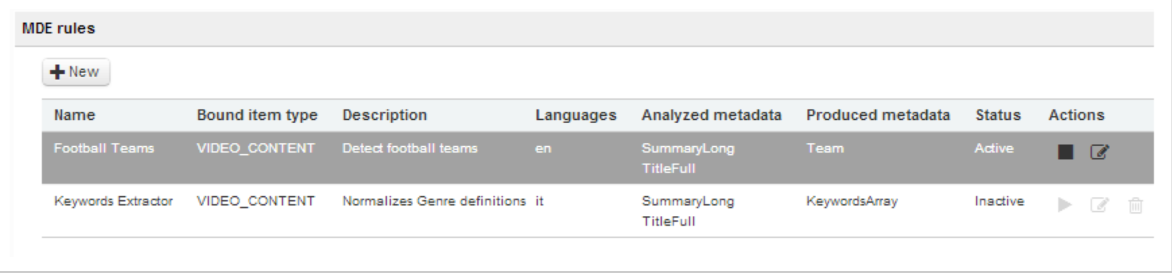
Action id	Type	Language	Input values	Action	Output value	Score
5	MD_CONTAINS_WORD	en	newcastle; magpies	APPEND_VAL	newcastle	1.0
6	MD_CONTAINS_WORD	en	aston villa; the villans	APPEND_VAL	aston villa	1.0
7	MD_CONTAINS_WORD	en	birmingham city; bluenoses	APPEND_VAL	birmingham city	1.0
8	MD_CONTAINS_WORD	en	blackburn; the rovers	APPEND_VAL	blackburn	1.0
9	MD_CONTAINS_WORD	en	bolton; the trotters derby	APPEND_VAL	bolton	1.0
10	MD_CONTAINS_WORD	en	county; the rams	APPEND_VAL	county	1.0
11	MD_CONTAINS_WORD	en	everton; the toffees	APPEND_VAL	everton	1.0

Metadata Enhancer: list of mde rules and related configuration.

MDE Rules

This portlet provides the list of MDE rules configured in the system. For each rule, the following information is provided:

- *Name*: the name of the rule.
- *Bound itemtype*: the itemtype for which the rule has been defined.
- *Description*: the description of the rule.
- *Languages*: the languages considered by the rule.
- *Analyzed Metadata*: the list of metadata analyzed by the rule.
- *Produced Metadata*: the metadata whose values are the output of the rule application.
- *Status*: the status of the rule. It can be ACTIVE or INACTIVE.
- *Actions*: You can:
  - *activate* and *deactivate* the rule.
  - *edit* the rule configuration.
  - *delete* the rule, available only for inactive rules.



MDE rules

Name	Bound item type	Description	Languages	Analyzed metadata	Produced metadata	Status	Actions
Football Teams	VIDEO_CONTENT	Detect football teams	en	SummaryLong TitleFull	Team	Active	■ ✎
Keywords Extractor	VIDEO_CONTENT	Normalizes Genre definitions	it	SummaryLong TitleFull	KeywordsArray	Inactive	▶ ✎ 🗑

Metadata Enhancer: list of mde rules.

MDE Rule Actions

This portlet allows to manage the actions associated to the MDE rule currently selected in the MDE rules portlet.

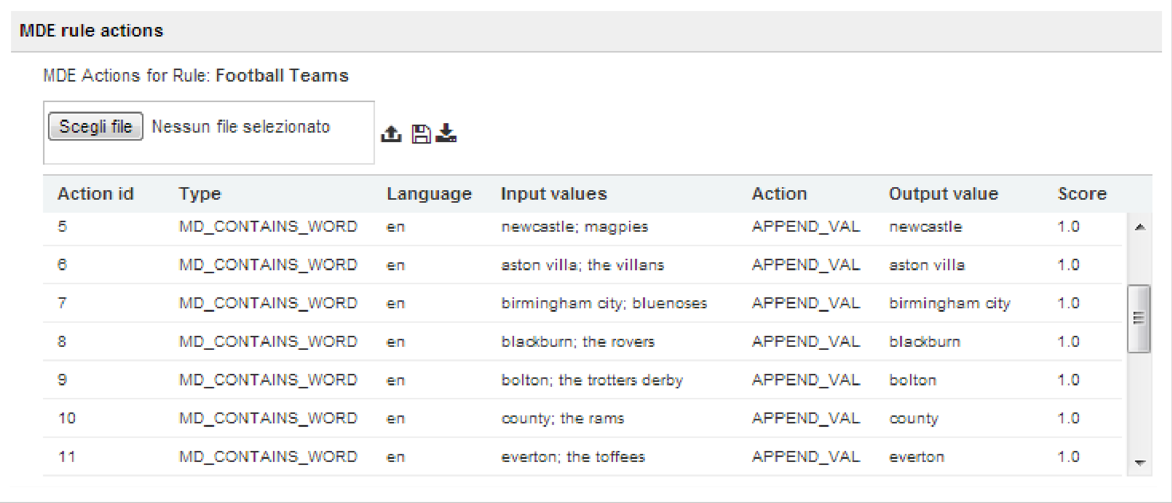
It lists the actions currently associated to the rule and it gives the possibility to:

- *Download* the action configuration file.
- *Upload* a new action configuration file and show a preview.

Warning
Uploading a new configuration file overwrites the previous action configuration.

Important note
To save the new configuration, click the save button.

- *Save* save the action configuration.



MDE rule actions

MDE Actions for Rule: **Football Teams**

Action id	Type	Language	Input values	Action	Output value	Score
5	MD_CONTAINS_WORD	en	newcastle; magpies	APPEND_VAL	newcastle	1.0
6	MD_CONTAINS_WORD	en	aston villa; the villans	APPEND_VAL	aston villa	1.0
7	MD_CONTAINS_WORD	en	birmingham city; bluenoses	APPEND_VAL	birmingham city	1.0
8	MD_CONTAINS_WORD	en	blackburn; the rovers	APPEND_VAL	blackburn	1.0
9	MD_CONTAINS_WORD	en	bolton; the trotters derby	APPEND_VAL	bolton	1.0
10	MD_CONTAINS_WORD	en	county; the rams	APPEND_VAL	county	1.0
11	MD_CONTAINS_WORD	en	everton; the toffees	APPEND_VAL	everton	1.0

Metadata Enhancer: list of MDE rule actions associated to the selected MDE rule.

MDE Rule Actions Configuration File

The actions to associate with a rule have to be edited in a text file.

Important note
The configuration file must be a *.txt* file, saved with UTF-8 LINUX format. The file cannot be uploaded if these requirements are not satisfied.

Each row of the file must be compliant with the following syntax:

```
ACTION_ID|LANGUAGE|TYPE|INPUT VALUES(; separated)|ACTION|OUTPUTVALUE|SCORE
```

- The fields are:
  - *ACTION_ID* is a unique identifier of the action. It must be an integer number.
  - *LANGUAGE* is the language applied by the action (2 digits, format ISO 639).
  - *TYPE* is the type of the action. It can be one of:
    - *MD_CONTAINS_WORD*: search for words, that are specified in the input values field.
    - *MD_CONTAINS_PERSON*: search for persons name, specified in the input values field.
    - *MD_MATCH_REGEX*: search matches with a given regular expression, specified in the input values field.
  - *INPUT VALUES* is the list of values to search for in the input metadata of the associated rule. Values must be **;** separated.
  - *ACTION* is the operation executed by the action if the threshold is reached.
    Valid values for TYPE *MD_CONTAINS_WORD* and *MD_CONTAINS_PERSON* are:
    - *APPEND_VAL*: append the output value to the values of the produced metadata.
    - *REPLACE_VAL*: substitute the value of produced metadata with the specified output value.
      Valid values for TYPE *MD_MATCH_REGEX* are:
    - *APPEND_MATCHES*: appends to produced metadata the matching words from analyzed metadata.
    - *REPLACE_MATCHES*: replaces into produced metadata the matches. Is valid only if produced metadata = analyzed metadata.
    - *DROP_MATCHES*: removes from produced metadata the matches. Is valid only if produced metadata = analyzed metadata.
  - *OUTPUTVALUE* is the output value that the action produces if the input values match in the analyzed metadata with a sufficient score.
  - *SCORE* is the score of the action. An action, to apply its effect, must produce a score greater than the defined threshold.
- Each field is **|** separated.
- Comment can be inserted by placing a **#** as first character of the row.

An example of action configuration file is reported below:

```
# This is a comment
# ACTION_ID|LANGUAGE|TYPE|INPUT VALUES(; separated)|ACTION|OUTPUTVALUE|SCORE
1|en|MD_CONTAINS_WORD|soap;love story|APPEND_VAL|romance|0.5
2|en|MD_CONTAINS_WORD|police;investigation;detective|APPEND_VAL|crime|0.7
3|en|MD_CONTAINS_WORD|dance;musical|APPEND_VAL|musical|0.4
```

MDE rule editor

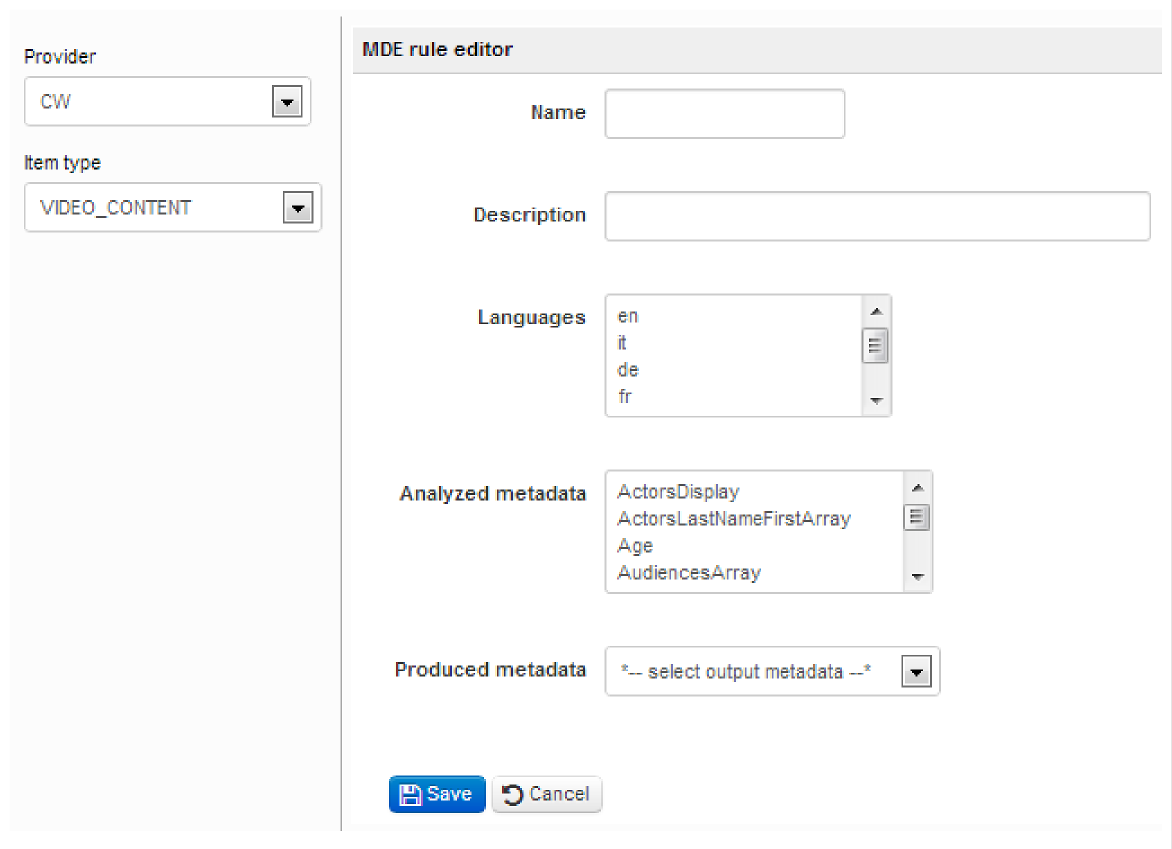
This section shows how to add a new MDE rule in the system.

To create a new MDE rule click the *New* button in the MDE rules portlet.

You will be prompted to select:

- A provider.
- An itemtype for which the rule will be created.

> ⚠ **Warning**
> Once a rule has been created, the bound item type cannot be modified.



Provider: CW

Item type: VIDEO_CONTENT

MDE rule editor

Name

Description

Languages: en, it, de, fr

Analyzed metadata: ActorsDisplay, ActorsLastNameFirstArray, Age, AudiencesArray

Produced metadata: *-- select output metadata --*

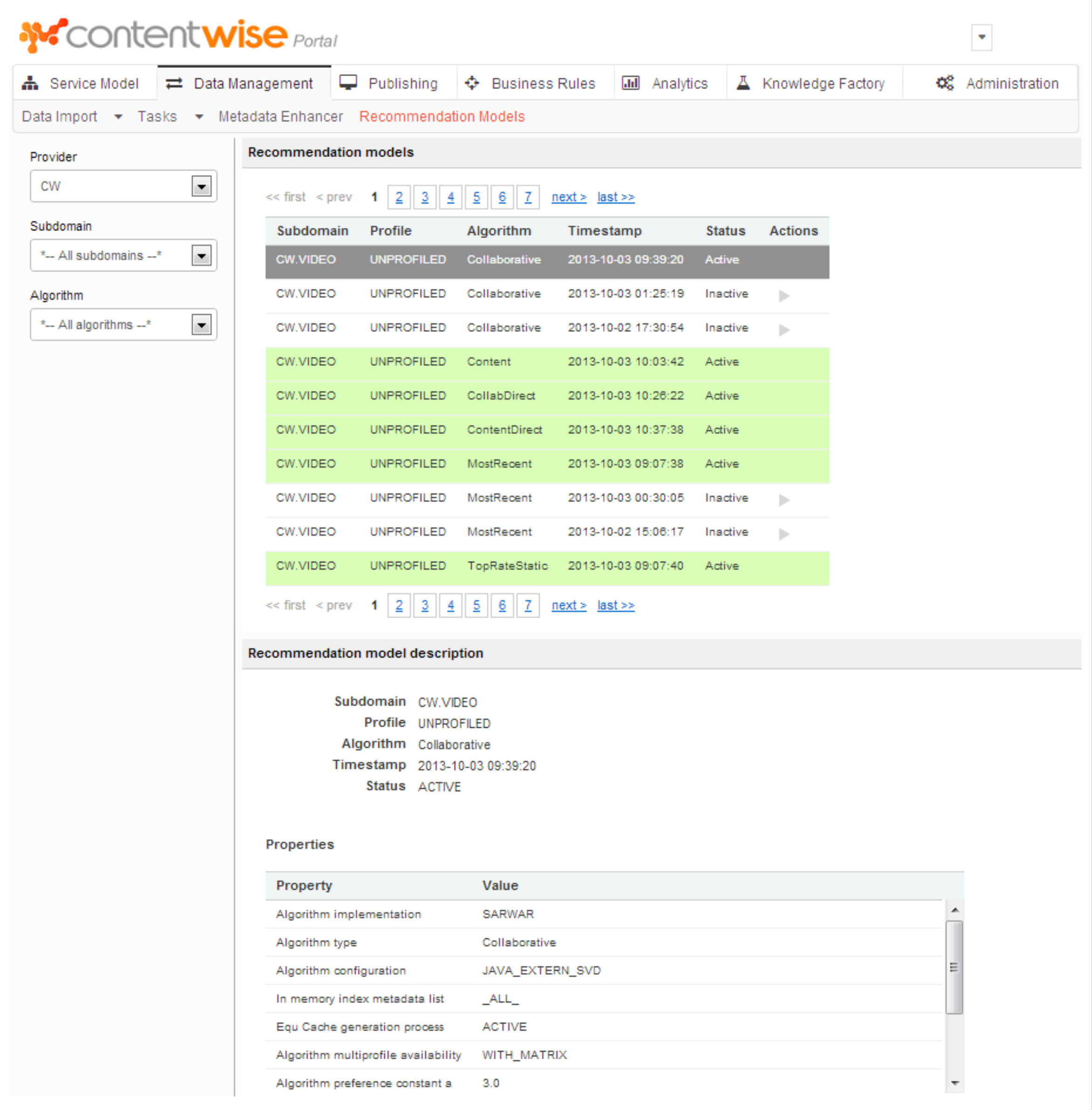
Metadata Enhancer: MDE rule editor.

Recommendation Models

This page lists the recommendation models that have been generated by ContentWise and allows to switch a model currently in use with a previous recommendation model.

It is possible to filter results by:

- *Subdomain*: if a subdomain filter is selected, only results of the specified subdomain are shown.
- *Algorithm*: if an algorithm filter is selected, only results of the specified algorithm are shown.



Recommendation models

Subdomain	Profile	Algorithm	Timestamp	Status	Actions
CW.VIDEO	UNPROFILED	Collaborative	2013-10-03 09:39:20	Active	
CW.VIDEO	UNPROFILED	Collaborative	2013-10-03 01:25:19	Inactive	▶
CW.VIDEO	UNPROFILED	Collaborative	2013-10-02 17:30:54	Inactive	▶
CW.VIDEO	UNPROFILED	Content	2013-10-03 10:03:42	Active	
CW.VIDEO	UNPROFILED	CollabDirect	2013-10-03 10:26:22	Active	
CW.VIDEO	UNPROFILED	ContentDirect	2013-10-03 10:37:38	Active	
CW.VIDEO	UNPROFILED	MostRecent	2013-10-03 09:07:38	Active	
CW.VIDEO	UNPROFILED	MostRecent	2013-10-03 00:30:05	Inactive	▶
CW.VIDEO	UNPROFILED	MostRecent	2013-10-02 15:06:17	Inactive	▶
CW.VIDEO	UNPROFILED	TopRateStatic	2013-10-03 09:07:40	Active	

Recommendation model description

Subdomain CW.VIDEO
Profile UNPROFILED
Algorithm Collaborative
Timestamp 2013-10-03 09:39:20
Status ACTIVE

Properties

Property	Value
Algorithm implementation	SARWAR
Algorithm type	Collaborative
Algorithm configuration	JAVA_EXTERN_SVD
In memory index metadata list	_ALL_
Equ Cache generation process	ACTIVE
Algorithm multiprofile availability	WITH_MATRIX
Algorithm preference constant a	3.0

Recommendation Models: list of recommendation models generated by ContentWise

Recommendation Models

This portlet lists the recommendation models generated by ContentWise.

Important Note
The models that are currently active in the system are highlighted in green. These are the models that are providing real time recommendations in the system.

For each recommendation model, the table provides:

- *Subdomain*: The subdomain for which the model has been generated.
- *Profile*: The profile for which the model has been generated.
- *Algorithm*: The algorithm used to generate the model.
- *Timestamp*: The timestamp of the model generation.
- *Status*:
    - ACTIVE, if the model is currently used by the system to provide recommendations.
    - INACTIVE, if the model is not in use.
- *Actions*: you can:
    - *activate*: to activate a model that is not currently in use.

> **Warning**
> The activation of a model causes the de-activation of the current active model for the same subdomain-profile-algorithm.
> Moreover, changing the active model for a given subdomain-profile-algorithm, causes the AlgoServer to reload.

Recommendation model description

This portlet shows the configuration of the recommendation model selected.

Dynamic Lists

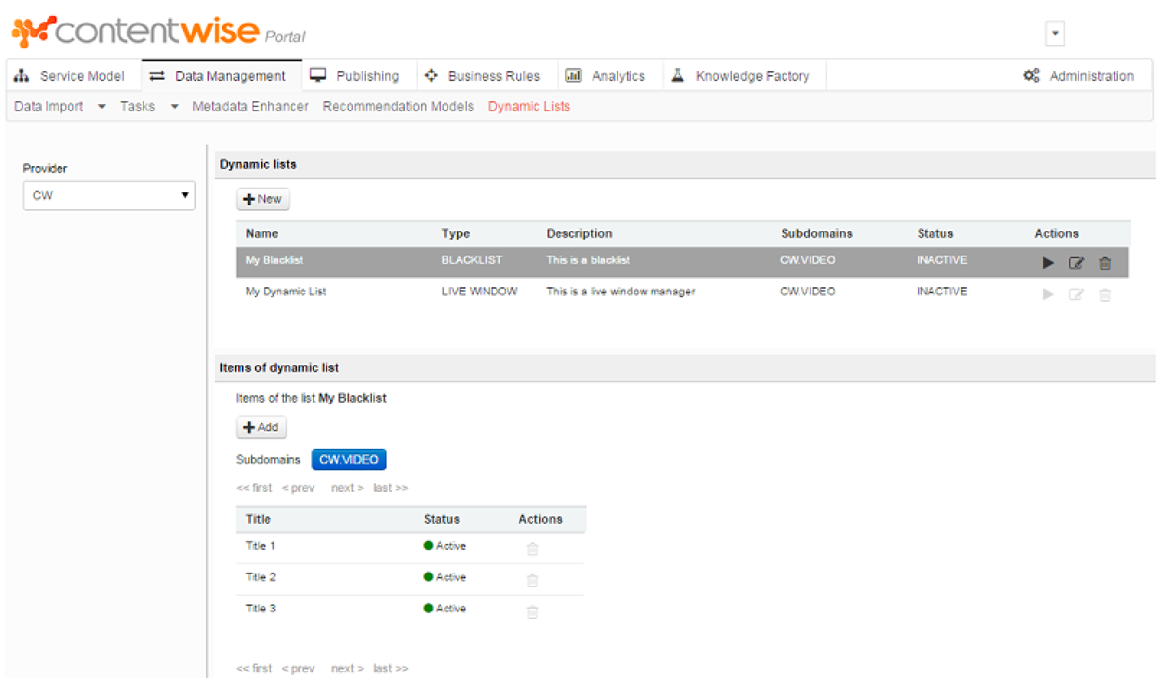
This page provides an overview about Dynamic Lists and their configuration in the system.

> **Key Concept**
> A dynamic list represents a set of items that match a given condition. A dynamic list affects the recommendation results according to its configuration:
>
> - A **Blacklist dynamic list** removes items from recommendation results
> - A **Live Window dynamic list** filters out from recommendation results the items whose live window (defined through item attributes) does not satisfy the conditions of the dynamic list.

Dynamic lists are continuously evaluated by the system. This allow to include/exclude items from the list dynamically, upon changes of the item attributes that are part of the list conditions.

The following of the page describes how to define and configure each of the available dynamic list types.

The configuration of the dynamic list is available at * Access *Data Management* > *Dynamic Lists* page of the ContentWise Portal



Name	Type	Description	Subdomains	Status	Actions
My Blacklist	BLACKLIST	This is a blacklist	CW.VIDEO	INACTIVE	
My Dynamic List	LIVE WINDOW	This is a live window manager	CW.VIDEO	INACTIVE	

Items of the list **My Blacklist**

Title	Status	Actions
Title 1	● Active	
Title 2	● Active	
Title 3	● Active	

Dynamic Lists

Blacklist

> **Key Concept**
> Dynamic lists of type *Blacklist* allow to exclude items from the recommendation results.

This section describes how to create and configure a Blacklist.

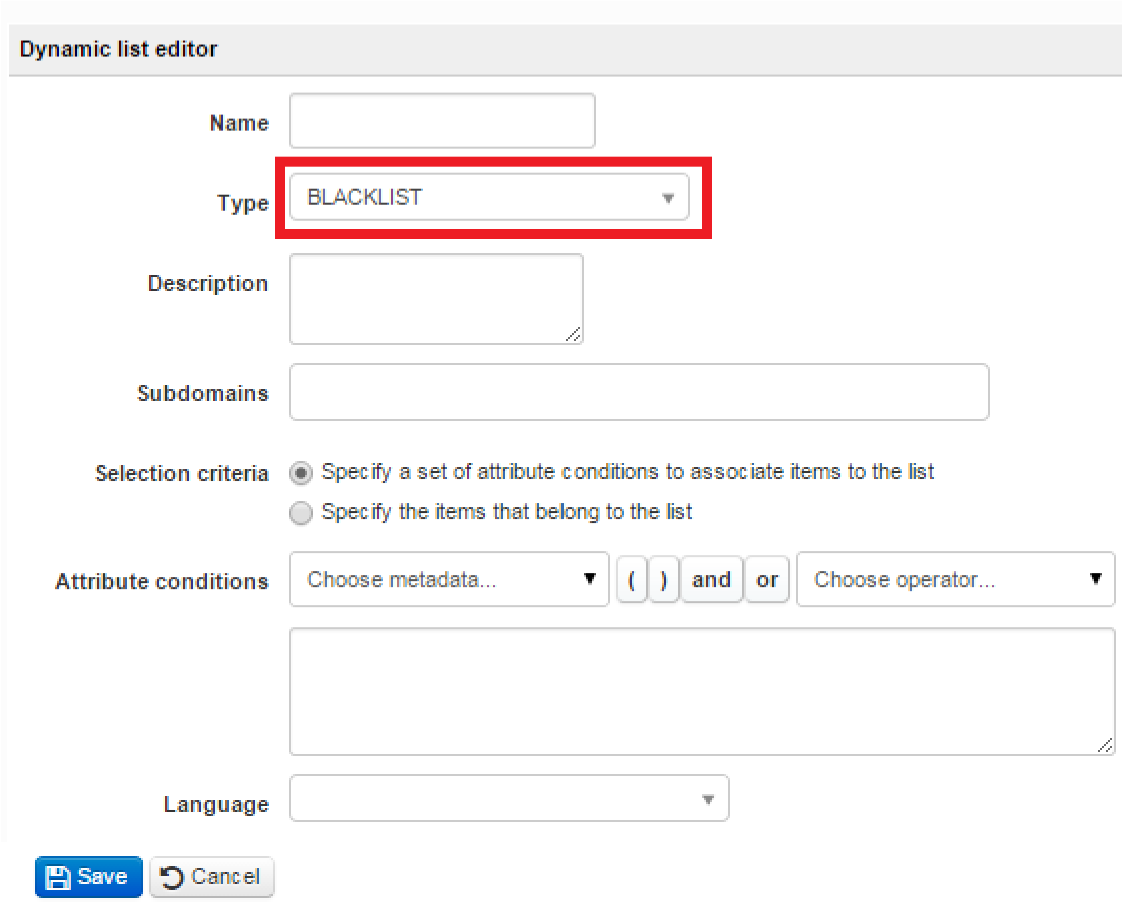
ContentWise provides two types of blacklists:

- **By attribute**: only items that satisfy a specified attribute condition will be part of the blacklist.
- **By item**: items that have to belong to the list are explicitly chosen by the user, through a search box.

Once the blacklist has been defined and activated, it is processed by the core engine. Once processed, it is included in the recommendation generation process and the items that belong to the blacklist are excluded from recommendation results.

This section describes the steps to define and configure a blacklist.

1. Create a new Dynamic List and choose the BLACKLIST type



Dynamic list editor

Name

Type: BLACKLIST

Description

Subdomains

Selection criteria
● Specify a set of attribute conditions to associate items to the list
○ Specify the items that belong to the list

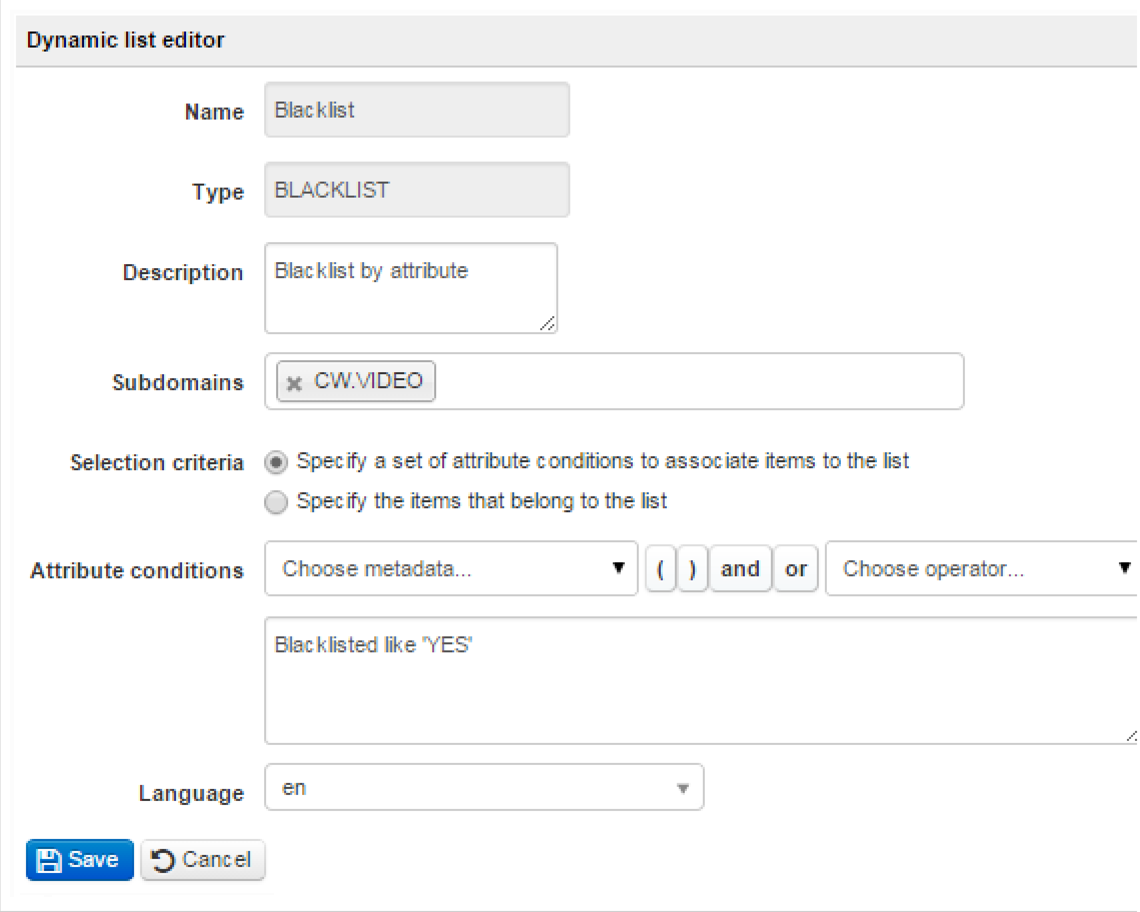
Attribute conditions: Choose metadata... ( ) and or Choose operator...

Language

Dynamic Lists: Create a new blacklist

2. Select the subdomain(s) for which the blacklist has to operate.
3. Specify the desired selection criteria. It is possible to choose between:
   - *Specify a set of attribute conditions to associate items to the list*: user is prompted to define a mandatory attribute condition to identify the criteria for which an item has to belong to the list.

> **Important note**
> - Attribute condition is specified using SQL syntax
> - Item attributes always contain strings.
> - Include string values between single quotes. E.g. GenresArray like '%#Action#%' is a valid condition
> - Remember that ContentWise metadata adopts # (sharp) as metadata value separator, in case of multi-value metadata.

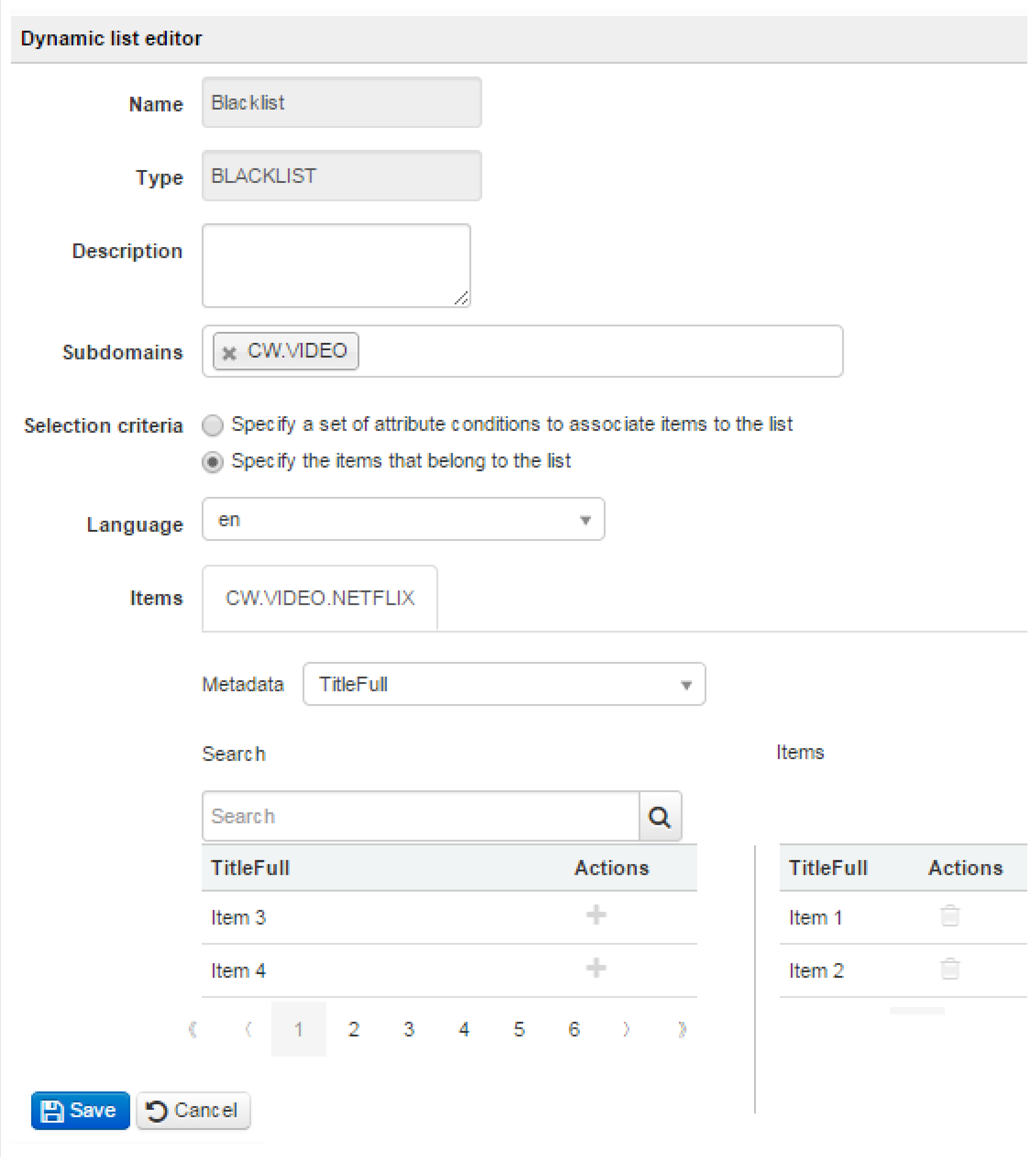


Dynamic list editor

Name	Blacklist
Type	BLACKLIST
Description	Blacklist by attribute
Subdomains	CW.VIDEO
Selection criteria	● Specify a set of attribute conditions to associate items to the list ○ Specify the items that belong to the list
Attribute conditions	Choose metadata... ( ) and or Choose operator... Blacklisted like 'YES'
Language	en

Dynamic Lists: "Specify a set of attribute conditions to associate items to the list" option

- *Specify the items that belong to the list*: you can search contents by any item attribute (e.g. Title). Once a search has returned result, it is possible to add items to the list by clicking the *add* (plus) icon. It is possible to remove an already in item by clicking the related *trash* icon.



Dynamic list editor

Name: Blacklist

Type: BLACKLIST

Description:

Subdomains: ✖ CW.VIDEO

Selection criteria:
○ Specify a set of attribute conditions to associate items to the list
● Specify the items that belong to the list

Language: en

Items: CW.VIDEO.NETFLIX

Metadata: TitleFull

Search

TitleFull	Actions
Item 3	+
Item 4	+

Items

TitleFull	Actions
Item 1	🗑
Item 2	🗑

Dynamic Lists: "Specify the items that belong to the list" option

4. Once configuration is completed, click Save to commit the changes.

Important note
Once a blacklist is created or modified, all the changes are propagated into the system. They will be available once all the components have been notified and refreshed. This process is automatic.

Live Windows dynamic list

Items of the catalogue may have live windows, that define and regulate the availability of the item to the end-users. An item is recommendable only when its live window is active (if any). *Live Window* dynamic lists allow to support this requirement.

Key Concept
By defining a dynamic list of type *Live Window*, it is possible to configure a set of conditions that allow to filter out from recommendation results all the items that do not have an active live window.

A Live Window dynamic list is characterized by:

- One or more subdomains for which the list is defined
- An attribute condition that identify the subset of the catalog that the list has to manage
- The item attributes that store the information about the live window of the item. Other than item attributes, it is possible to provide

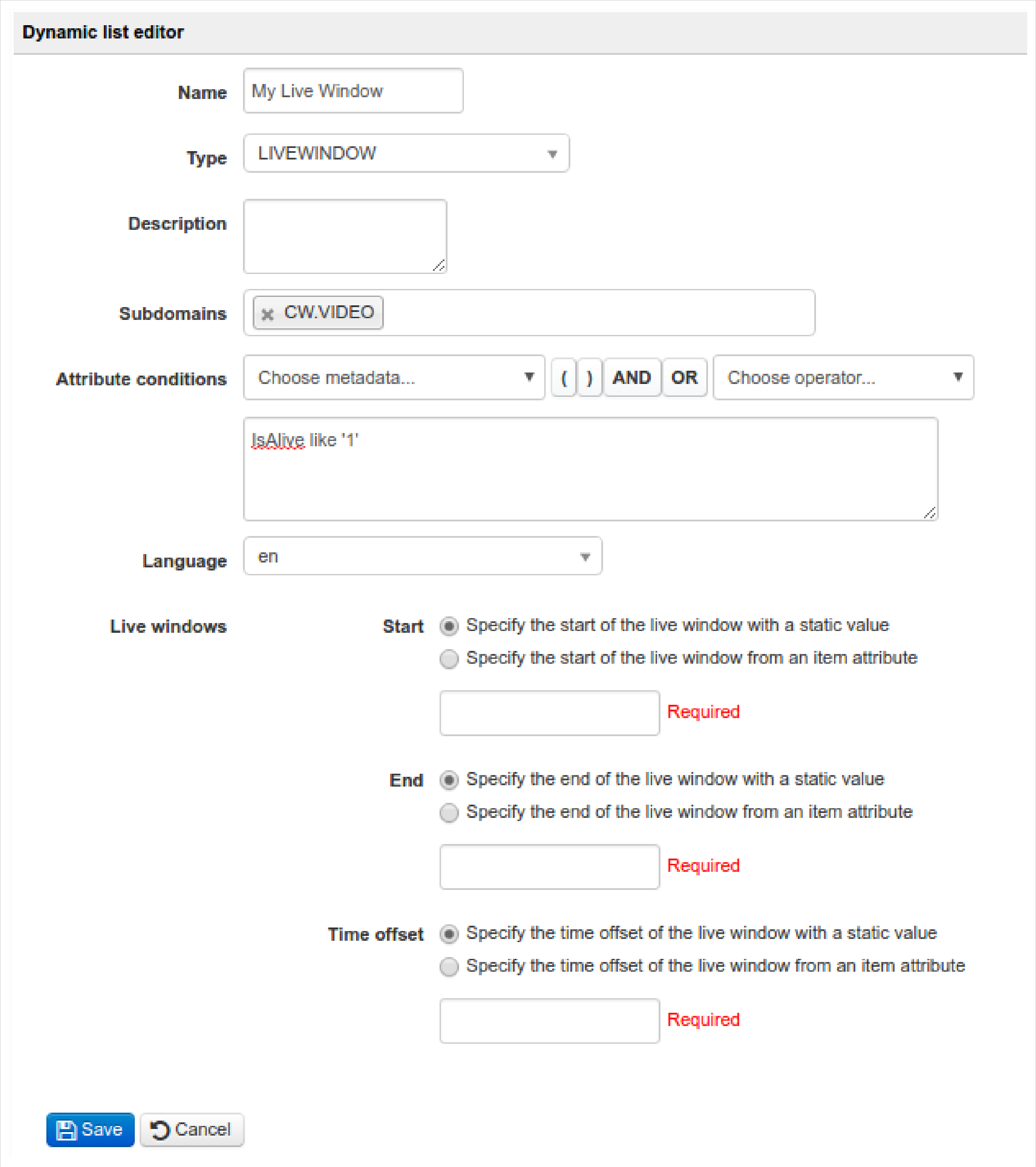
specific values. For each item, three data are required:

- A start of the live window. It is a date and it must be compliant with the format *yyyy-mm-dd HH24:MI:SS*
- An end of the live window. It is a date and it must be compliant with the format *yyyy-mm-dd HH24:MI:SS*
- The time zone used to represent start and end specified above. It must be in the format *+/-NN:NN*

This section describes the steps to define and configure a live window dynamic list.

1. Create a new Dynamic List and choose the LIVEWINDOW type
2. Select the subdomain(s) for which the live window has to be enabled.
3. Specify an optional attribute condition to identify the subset of the items (within the selected subdomains) for which the live window has to be calculated.

> **Important note**
> - Attribute condition is specified using SQL syntax
> - Item attributes always contain strings.
> - Include string values between single quotes. E.g. GenresArray like '%#Action#%' is a valid condition
> - Remember that ContentWise metadata adopts # (sharp) as metadata value separator, in case of multi-value metadata.



Dynamic list editor

Name: My Live Window

Type: LIVEWINDOW

Description:

Subdomains: CW.VIDEO

Attribute conditions: Choose metadata... ( ) AND OR Choose operator...

IsAlive like '1'

Language: en

Live windows

Start
● Specify the start of the live window with a static value
○ Specify the start of the live window from an item attribute
Required

End
● Specify the end of the live window with a static value
○ Specify the end of the live window from an item attribute
Required

Time offset
● Specify the time offset of the live window with a static value
○ Specify the time offset of the live window from an item attribute
Required

Dynamic Lists: Create a new live window dynamic list

4. Specify the item attributes or the fixed values to be used by the dynamic list manager to calculate the live window of the items.

User is prompted to select an attribute or a value for each of the following:

- A start date for the live window of the item
- An end date for the live window of the item
- A time offset used to represent start and end date.

5. Once configuration is completed, click Save to commit the changes.

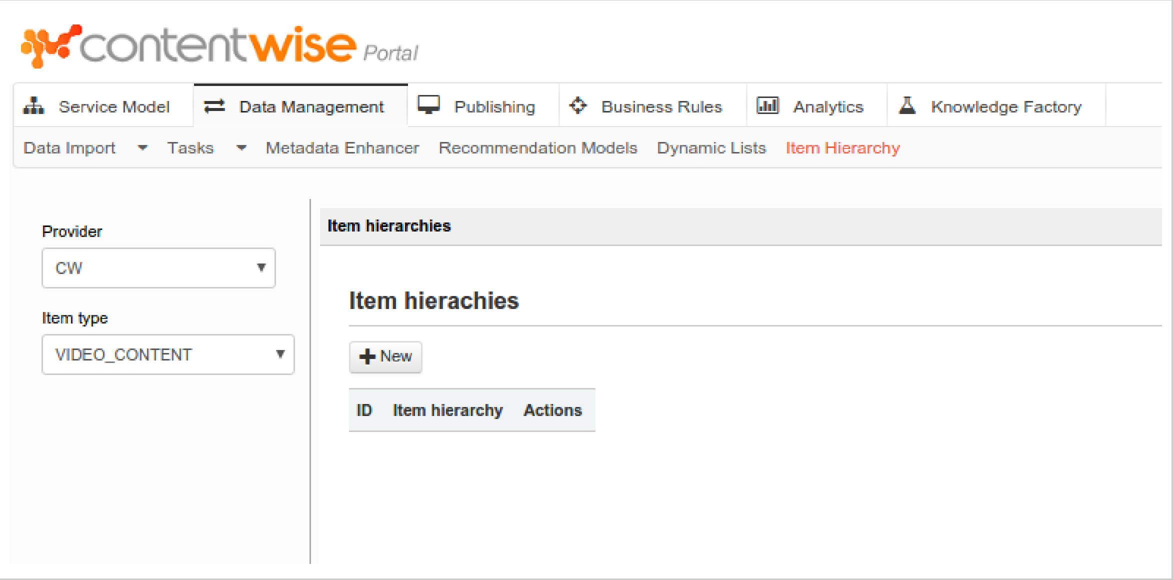
> **Important note**
> Once a live window dynamic list is created or modified, all the changes are propagated into the system. They will be available once all the components have been notified and refreshed. This process is automatic and may require some time to complete.

Item Hierarchy

This page provides an overview about Item Hierarchies configuration in the system. Please take a look to the Item hierarchies page for their definition and terminology.

The following of the page describes how to define and configure an Item Hierarchy.

The configuration of the item hierarchy is available at *Data Management* > *Item Hierarchy* page of the ContentWise Portal



Provider: CW

Item type: VIDEO_CONTENT

Item hierachies

ID	Item hierarchy	Actions

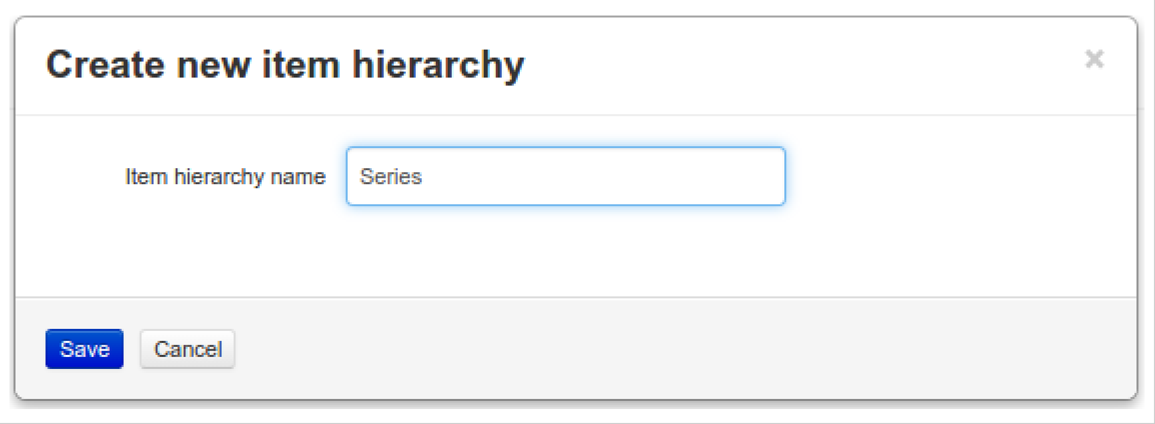
Item Hierarchy

Create a new Item Hierarchy

ContentWise can support multiple item hierarchies. You can select which hierarchy you would like to use in the caller editor page.

> ⚠ The item hierarchies models are computed via batch. Please verify if the hierarchy you would like to use in the caller has already been computed before select it in the caller editor page.

To create a new hierarchy click in the new button and select the hierarchy name. This name will appear in the caller editor page as caller configuration.



Create new item hierarchy

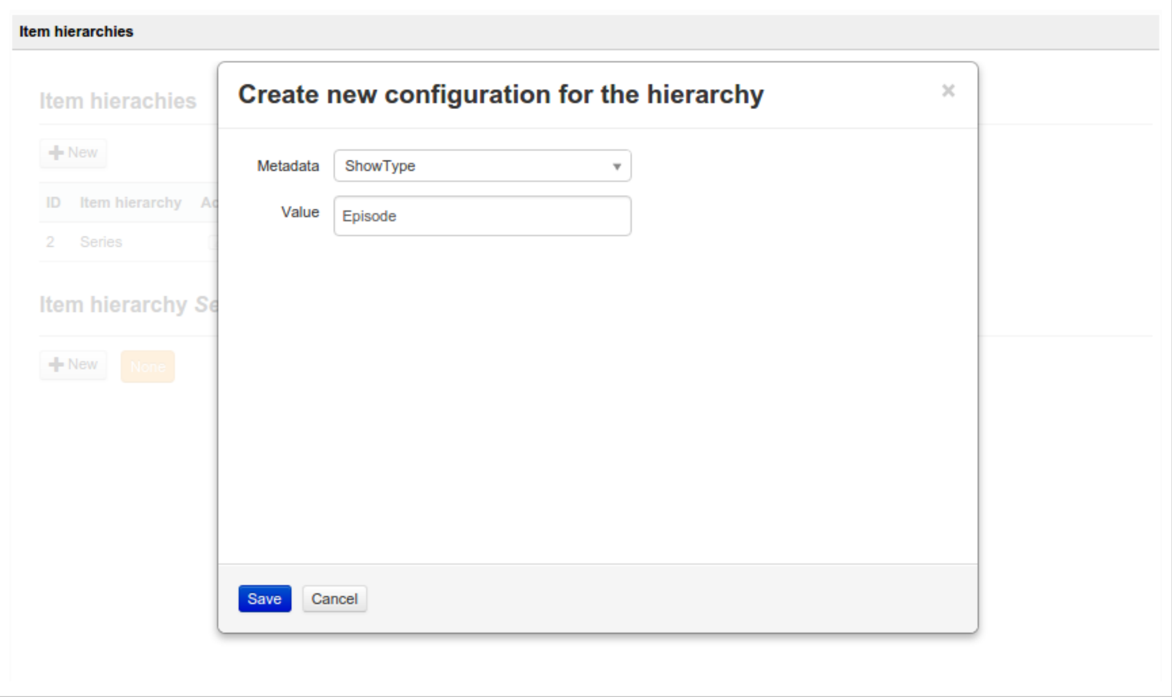
Item hierarchy name: Series

Item Hierarchy Creation

Create a new Item Hierarchy Configuration

An Item Hierarchy can be made by multiple configuration. You can select a metadata (e.g. the ShowType) to define different behaviors for different types of items. For example in the same hierarchy you can define a behavior for **movies** and an other one for **episodes**. In this guide an example for **episodes** is given.

> ⚠ Once you select the first configuration it cannot change the metadata for the other configurations. So if you select *ShowType* as metadata in the first configuration you have to use it also for the others.



Item hierarchies

Item hierachies

ID	Item hierarchy	Ac
2	Series	

Item hierarchy Se

Create new configuration for the hierarchy

Metadata: ShowType

Value: Episode

Item Hierarchy Configuration Creation

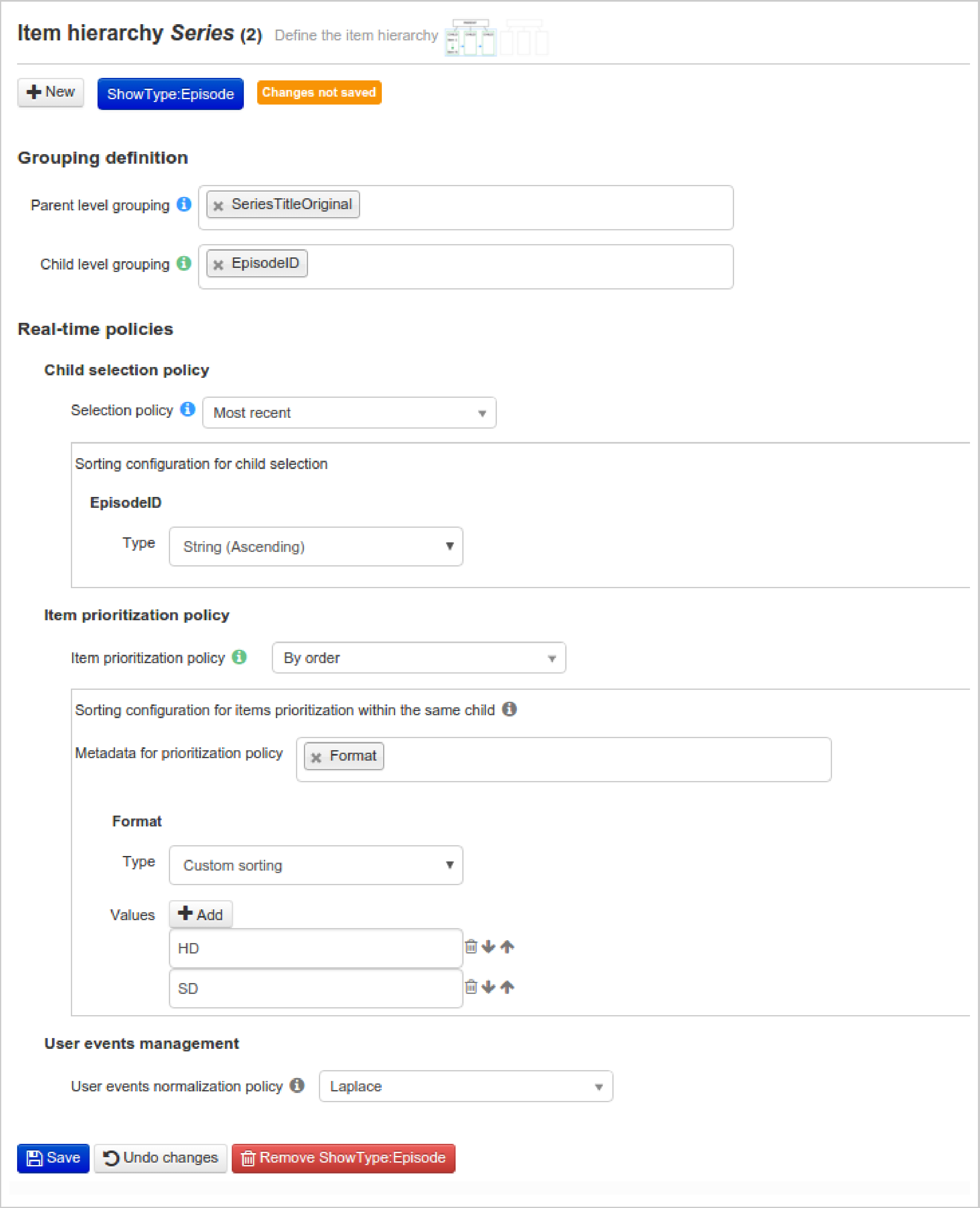
Edit an Item Hierarchy Configuration

Once you create a new configuration you have to define*

- Grouping definition: this part describe how you would like to group items. Please see the Item hierarchies page to understand the differences between *Parent level grouping* and *Child level grouping*.
- Real-time policies: once you have defined how group items, you have to define:
  - Child selection policy: possible values are
    - First missed: ContentWise will select the first item not seen by the user in the group. E.g. a user watch the first and the third episodes of a series, ContentWise will select the second episode
    - Best matching: to use for not "serial" content, ContentWise will select the item with the highest appeal with the respect to the used algorithm
    - Next one: ContentWise will select the next item not seen by the user in the group. E.g. a user watch the first and the third episodes of a series, ContentWise will select the forth episode
    - According to item metadata: each item contains a metadata with one of the previous policies, ContentWise will

    select the item in the group using the policy described in the selected metadata

- Item prioritization policy: possible values are
    - By order: ContentWise will select the first child available. E.g. if the user can watch the HD and the SD version of a movie and you select the HD as higher value, ContentWise will select the HD child.
    - Best Matching: ContentWise will select the child with the highest appeal.
    - According to item metadata: each child contains a metadata with one of the previous policies, ContentWise will select the child using the policy described in the selected metadata
- User events management: to be used in case of series. If a user watched several episodes of a series his/her profile could be strongly affected by the series episodes. To mitigate this effect you can enable the event normalization. Possible values are
    - Do not normalize user events: ContentWise will not do anything. To be used in case of movies.
    - By sum: ContentWise will normalize events of a group (e.g. a series) by the number of the items in the group. E.g. if a group is a representation of The Walking Dead and there are 48 episodes available in the system, ContentWise will normalize the users' events of Walking Dead episodes dividing by 48
    - Laplace: ContentWise will normalize events of a group (e.g. series) by the number of items the user watched. E.g. if a group is a representation of The Walking Dead and there are 48 episodes available in the system and the user watched 12 episodes, ContentWise will normalize the users' events of Walking Dead episodes dividing by 12



Item hierarchy *Series* (2) Define the item hierarchy

Grouping definition

Parent level grouping: SeriesTitleOriginal

Child level grouping: EpisodeID

Real-time policies

Child selection policy

Selection policy: Most recent

Sorting configuration for child selection

EpisodeID

Type: String (Ascending)

Item prioritization policy

Item prioritization policy: By order

Sorting configuration for items prioritization within the same child

Metadata for prioritization policy: Format

Format

Type: Custom sorting

Values: + Add

HD

SD

User events management

User events normalization policy: Laplace

Item Hierarchy Configuration

UX Design

The UX Design feature of Contentwise 6.0 allows the creation, management and publishing of a page that can be requested via API from the front end app.

The main concepts to know in order to use the UX Design are:

1) Use case: A use case represent every single content section of the front end UI. Sample of use case are: "Top Viewed", "Most Recent" "Recommended for You", a dynamic stream or an editorial section.

2) Page: A page is a set of use cases.

3) Target: A target represents the audience of a specific page. In this way you can create a generic page that can be shown only to the specified target. A target can be a set of users or a set of devices.

From the *UX Design* portal section you can access to:

Section	Description
UX Builder	Manage all the elements that can be used to personalise the user experience.
UX Integration	Manage the way a front end app can be integrated with the UX Builder
Profiles	It is the page dedicated to the management of the profiles.

UX Builder

The section *UX Builder* of the portal allows the definition and management of all the objects the you need to build a page.

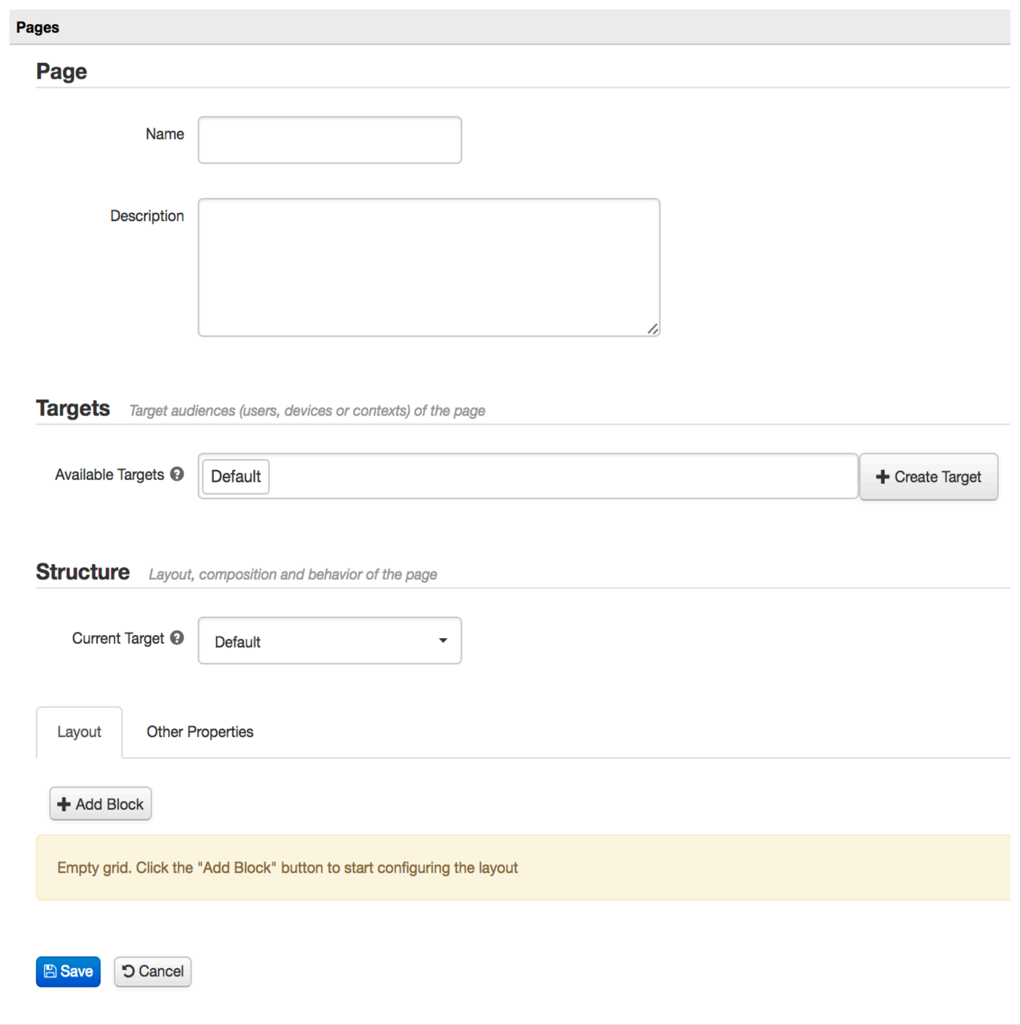
From the UX Builder menu it is possible to define and manage pages, use cases and targets.

Page

In Contentwise 6.0 a Page is an ordered set of use cases. To access the Pages management section follow the path: UX Design -> UX Builder -> Pages. From here it is possible to create a new page, delete o modify an existing page.

Create a new page.

To create a new page click the New button in the page management section of the portal. The page editor will open:



Pages

Page

Name

Description

Targets Target audiences (users, devices or contexts) of the page

Available Targets: Default

Structure Layout, composition and behavior of the page

Current Target: Default

Empty grid. Click the "Add Block" button to start configuring the layout

Layout configuration: list of configured layouts.

The page editor is divided in three sections: Page, Targets and Structures.

The Page section allows to specify a page name and a page description.
The Target section allows to assign the page to a specific audience.
The Structure section allows to specify the page layout in terms of number of page section, widget and use case for each section.

Use Case: Layout

A Layout represent a set of item that can be displayed on the UI. For each layout it is possible to specify the number of items to display and the criteria to use to chose the content for each item in the layout. As an example it is possible to set a layout of ten elements and configure to meet the following criteria:
#In the first position use always an item from the list of the Top Viewed
#In position two and three use items from a personalised recommendation
#In position for, five and six use items from an editorial list
#Use a personalised recommendation for all the other position in the layout.

Use Case: Dynamic Layout

Use Case: Search

A search use case is the search configuration to use to create a layout use case showing a set of item from a search query.

Use Case: Autocomplete

Use Case Groups

A use case group can be used to group use cases sharing the same schedule

Editorial List

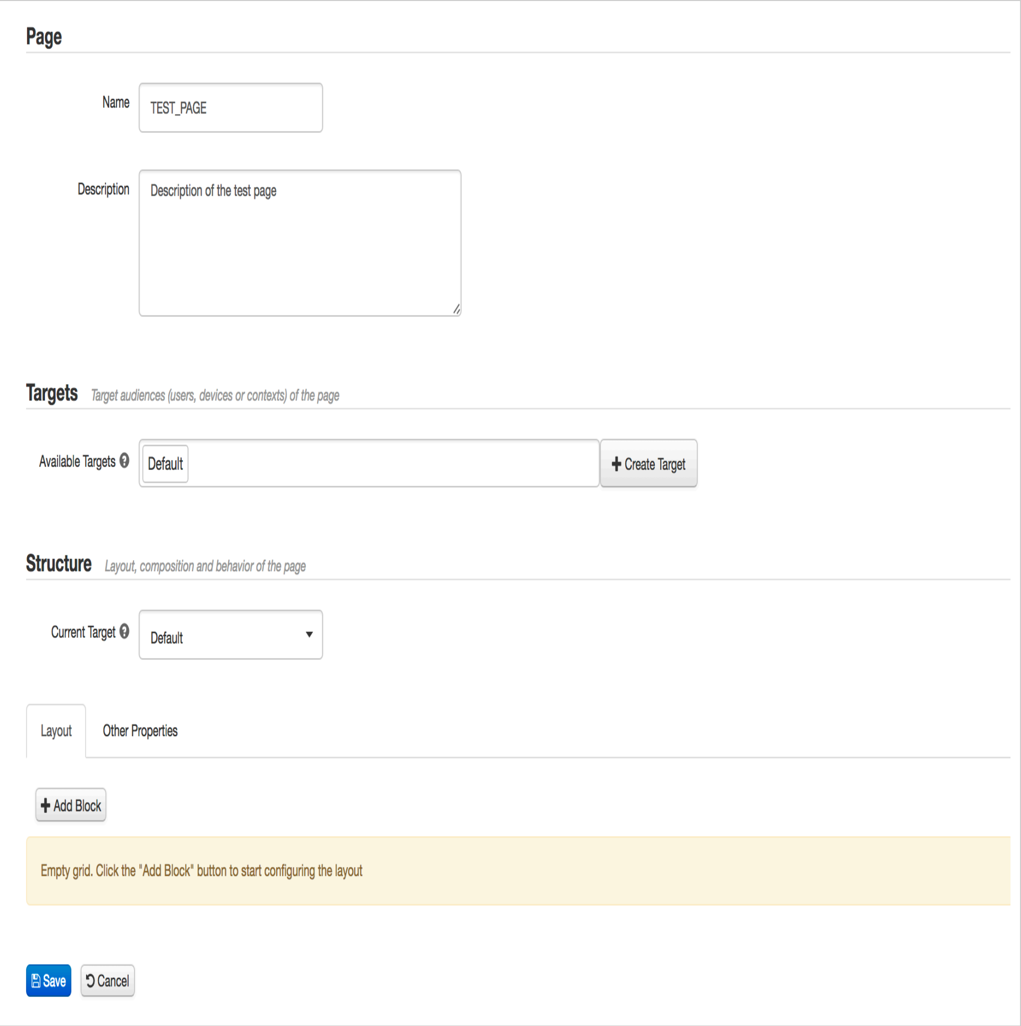
An editorial list is a set of item that are manually selected by an editor. The editorial list can be used to populate a whole layout or only some specified position.

Targets

A target represent the audience of a page. A target can be a set of users or a set of items.

How To: Create, edit and delete a page
At the end of this how-to you will be able to create a test page with the UX Builder tool.
Before to start this tutorial check that you have at

Enter a name and a description. For the purpose of this guide the page will be named TEST_PAGE and the description will be: "Description of TEST_PAGE"



Page

Name: TEST_PAGE

Description: Description of the test page

Targets *Target audiences (users, devices or contexts) of the page*

Available Targets: Default

Structure *Layout, composition and behavior of the page*

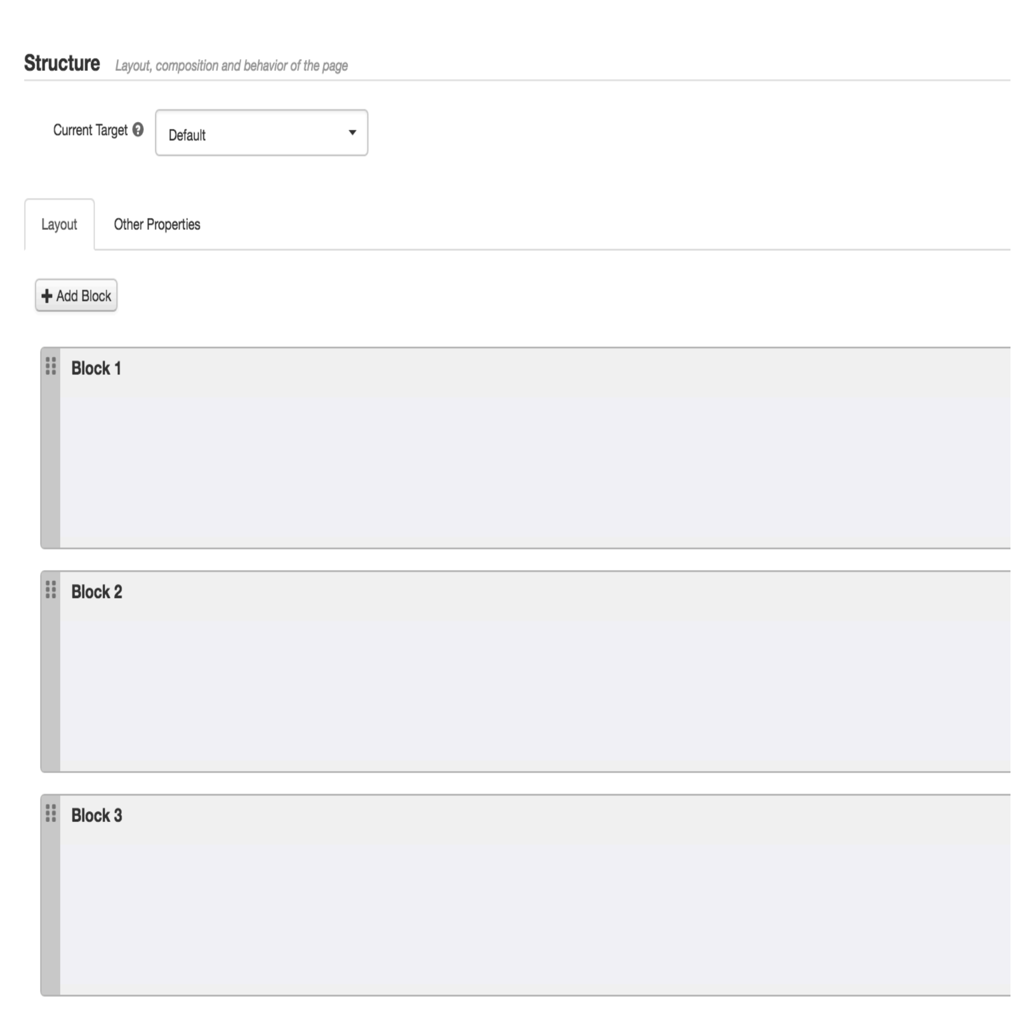
Current Target: Default

Empty grid. Click the "Add Block" button to start configuring the layout

Layout configuration: list of configured layouts.

Add a target. For the purpose of this document leave "Default".

and then add as many block as many use case you want to display on the page. For the purpose of this document we will add 3 use cases then we have to add 3 blocks to the page.
The Structure section of the page editor should be like this:



Structure Layout, composition and behavior of the page

Current Target: Default

Layout configuration: list of configured layouts.

Use Case
The next step to finish the creation of a page is to link a use case to every block on the page. You are not be able to save a page with a block not linked to a use case. Note that all the use cases you need must be already defined.

For the purpose of this document we will add:

Layouts

This section describes functionalities provided by *Layouts* menu.

Layouts menu gives access to the following pages:

Page	Description
Layout Configuration	It is the landing page of the section. It provides an overview of the layouts in the system and allows to manage their configurations.
Layout Scheduling	It is the page where it is possible to manage layout schedules, defining which layouts are active on which callers among time.

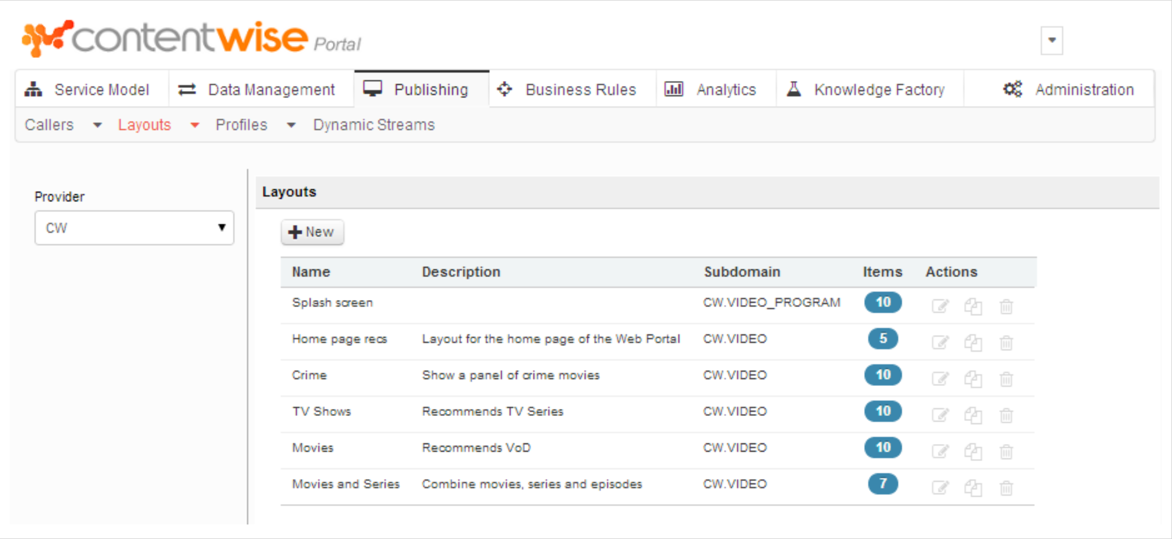
Layout Configuration

This page provides an overview about how layouts are configured in the system and provides access to the pages that allow to create and edit layouts.
For a definition of layout within ContentWise, see Layout.

The page contains the Layouts portlet that lists the layouts defined in the system and provides access to the layout editor.

- See Configuring a layout to create or edit a layout.
- See Scheduling a new layout for information about immediate schedule of a new layout.

If a provider filter is not selected, you will be prompted to select one.



Name	Description	Subdomain	Items	Actions
Splash screen		CW.VIDEO_PROGRAM	10	
Home page recs	Layout for the home page of the Web Portal	CW.VIDEO	5	
Crime	Show a panel of crime movies	CW.VIDEO	10	
TV Shows	Recommends TV Series	CW.VIDEO	10	
Movies	Recommends VoD	CW.VIDEO	10	
Movies and Series	Combine movies, series and episodes	CW.VIDEO	7	

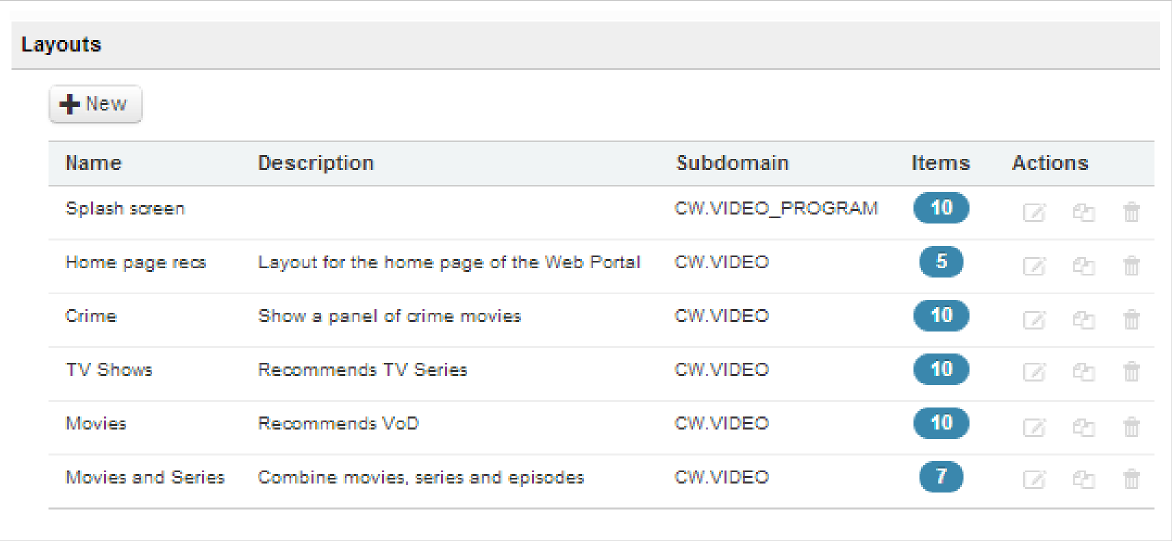
Layout configuration: list of configured layouts.

Layouts

This portlet provides an overview of layouts configuration and provides access to layout editor.

For each layout, the table shows:

- *Name*: the layout name
- *Description*: short layout description.
- *Subdomain*: the subdomain for which the layout has been defined.
- *Items*: a badge that shows the size of the layout.
- *Actions*: For each layout the following actions are available:
  - *Edit*: edit the layout to manage its configuration
  - *Clone*: clone the layout
  - *Remove*: remove a layout from the system. Take care about alert messages.



Name	Description	Subdomain	Items	Actions
Splash screen		CW.VIDEO_PROGRAM	10	
Home page recs	Layout for the home page of the Web Portal	CW.VIDEO	5	
Crime	Show a panel of crime movies	CW.VIDEO	10	
TV Shows	Recommends TV Series	CW.VIDEO	10	
Movies	Recommends VoD	CW.VIDEO	10	
Movies and Series	Combine movies, series and episodes	CW.VIDEO	7	

Layout Configuration: layouts portlet.

Configuring a layout

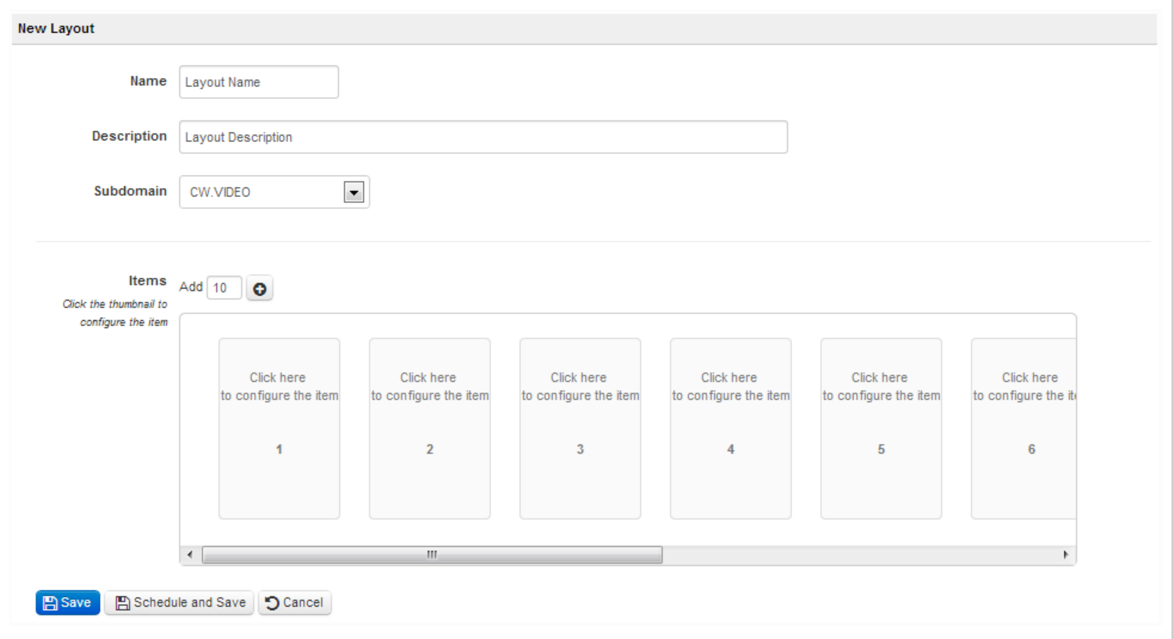
This section describes how to add a new layout in the system.

To create a new layout click the *New* button in the Layouts portlet.

To define a layout, the following information is required:

- *Name*: the identifier of the layout. Once a layout is defined, the name cannot be modified.
- *Description*: an optional description for the layout.
- *Subdomain*: the subdomain for which the layout is defined. Once a layout is defined, the subdomain cannot be modified.
- *Items*: the layout items configuration that define how the layout is built. It can be modified at any time.

Layout configuration changes application
When the configuration of an existent layout is modified by changing the layout items composition, the effects of the changes are immediate. It means that if the layout is active in the system at the moment in which the changes are saved, end-user that are receiving recommendations based on the layout will immediately see the effect of the changes.



New Layout

Name: Layout Name

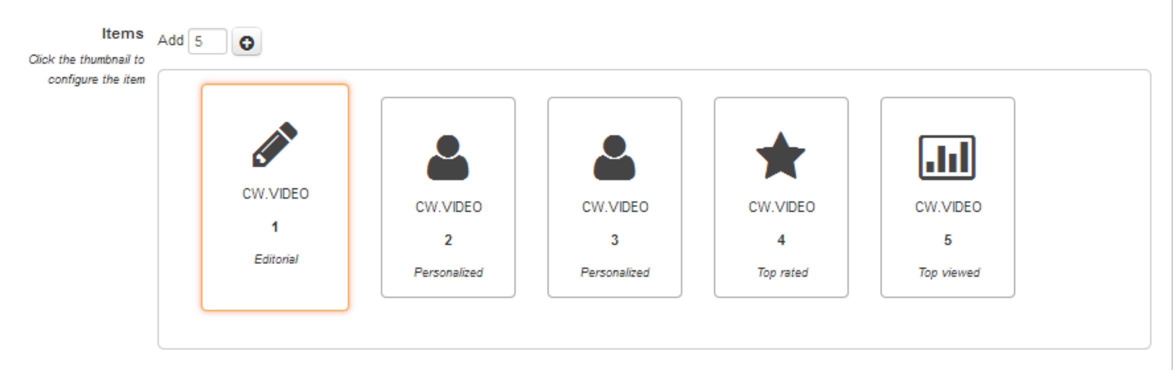
Description: Layout Description

Subdomain: CW.VIDEO

Items — *Click the thumbnail to configure the item* — Add 10

Layout editor

The core configuration is the definition of the items that compose the layout.



Items — *Click the thumbnail to configure the item* — Add 5

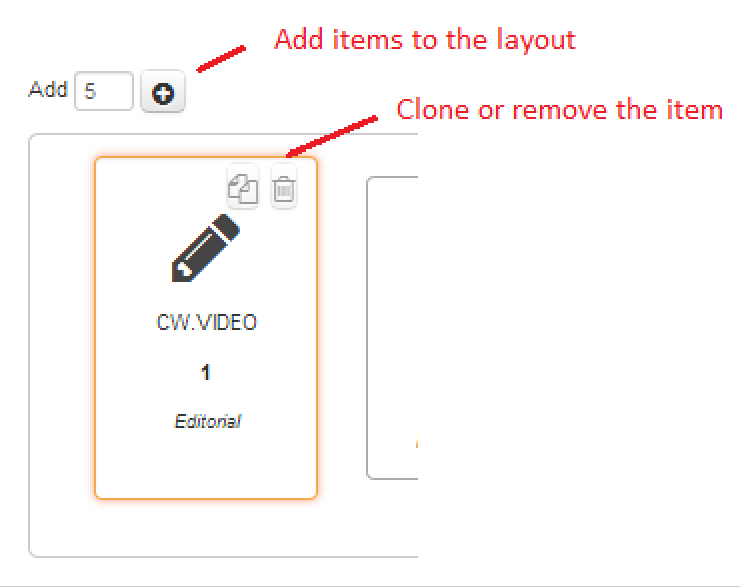
An example of layout composition

An item of the layout represents a single slot of the recommendation.

For each layout item, it is possible to:

- *Set* or change its configuration: select the desired item in order to open the item layout configuration dialog.
- *Clone* it: click the clone icon that is shown by hovering the mouse on the item preview
- *Remove* it: click the trash icon that is shown by hovering the mouse on the item preview
- *Move* it to a new position: drag&drop the item preview from the starting to the desired position

To add new items to the layout, type the desired number of items and click the add button. Items are enqueued to alredy existent items. Move them if necessary by using drag&drop.



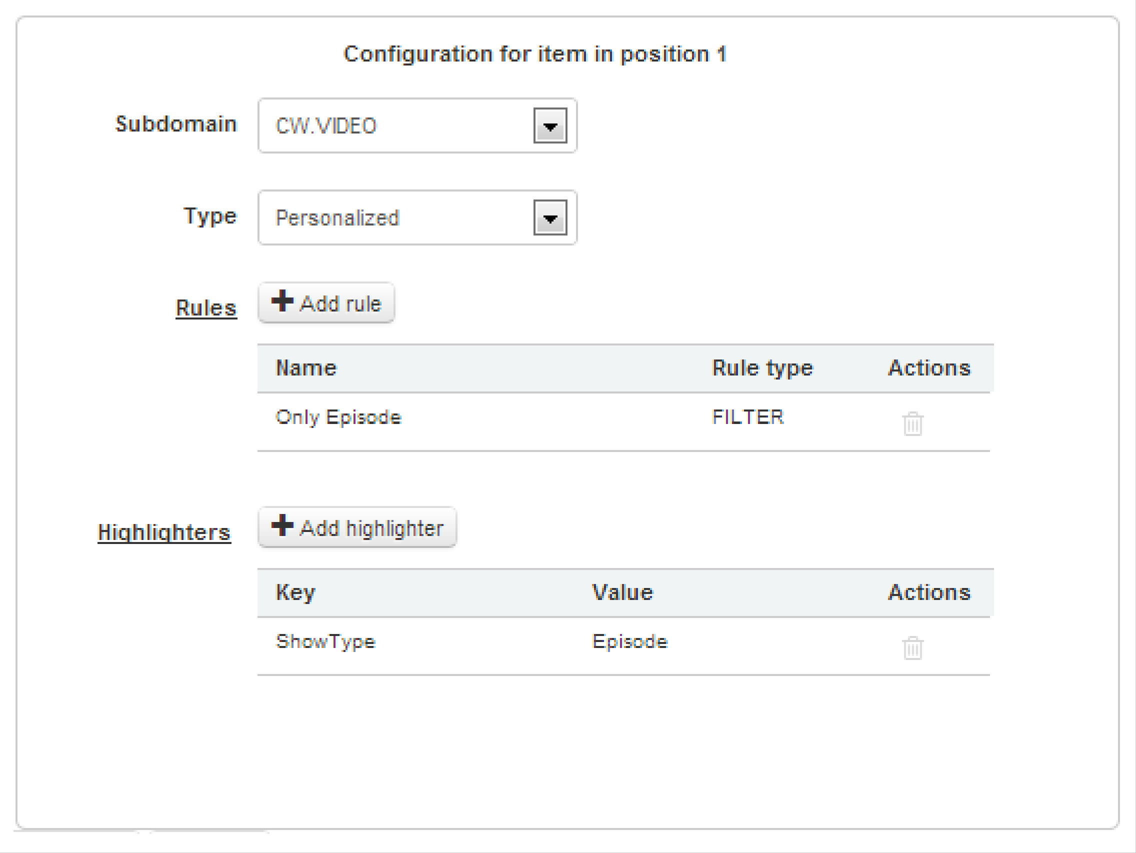
Add items to the layout

Clone or remove the item

Actions on a layout item

Each item of the layout can be configured and customized by choosing:

- *The subdomain*. When a layout is defined on a cross-domain, by choosing different subdomains
- *The type*. A layout element can be:
  - An item retrieved from a personalized recommendation algorithm
  - An item retrieved from an editorial list. You will be prompted to select the editorial list (must be one defined for the selected subdomain) and the selection policy to adopt.
  - An item retrieved from a static recommendation algorithm, such as top rated, top viewed and most recent algorithms.
- *Filter business rules* to be applied to the item, only if the layout element is a personalized or a static recommendation item.
- A pre-calculated statistic: Only for top rated and top viewed types, allows to choose an optional custom statistic to be used for generating the result.
- A set of *highlighters*. An highlighter is key-value pair of data that will be associated by the system to the layout element when returning it to the client.



Configuration for item in position 1

Subdomain: CW.VIDEO

Type: Personalized

Rules

Name	Rule type	Actions
Only Episode	FILTER	🗑

Highlighters

Key	Value	Actions
ShowType	Episode	🗑

An example of personalized layout item configuration

Scheduling a new layout

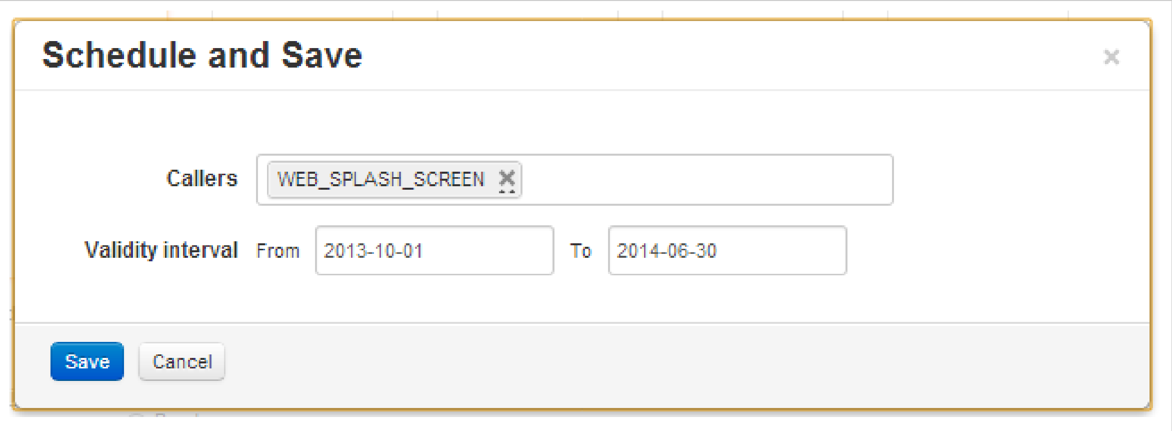
When defining a new layout, it is possible to schedule it directly from the layout editor page.
This is a shortcut that can be useful when layout schedules require the existence of only one layout at a time.
For complex layout scheduling configuration, see Layout Scheduling page.

To directly schedule a new layout, click the *Schedule and Save* button of the layout editor page, once the layout has been configured and it is ready to be activated in the system.

You will be prompted to specify:

- *A list of callers* for which the layout has to be activated
- The *duration* of the schedule in terms of start - end interval

The system will check for schedule conflicts that may exist in case one or more of the selected callers already belong to a layout schedule that impacts the selected start-end interval. If no conflicts will be found, both the layout configuration and the schedule will be saved.



Schedule and Save

Callers: WEB_SPLASH_SCREEN

Validity interval From: 2013-10-01 To: 2014-06-30

Save and schedule a new layout

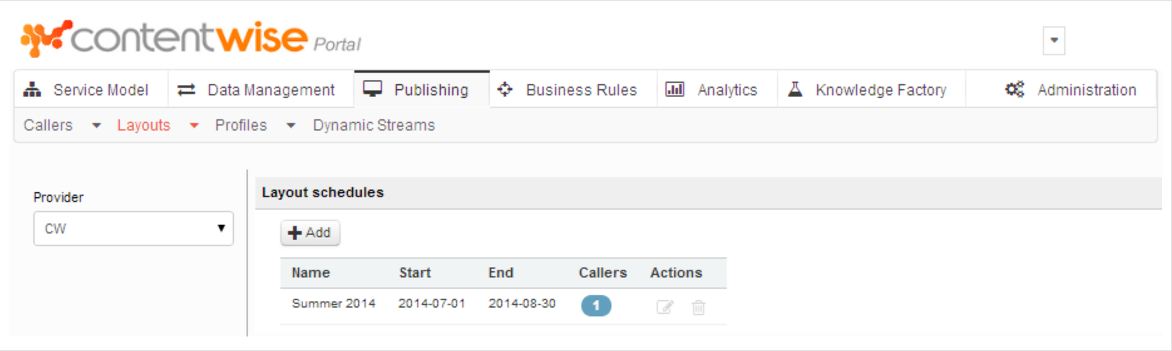
Layout Scheduling

This page provides an overview about how layouts can be scheduled in the system among the different callers.

For a definition of layout within ContentWise, see Layout.

The page contains the Layouts portlet that lists the layout schedules defined in the system and provides access to the layout schedule editor.

If a provider filter is not selected, you will be prompted to select one.



Provider: CW

Layout schedules

Name	Start	End	Callers	Actions
Summer 2014	2014-07-01	2014-08-30	1	

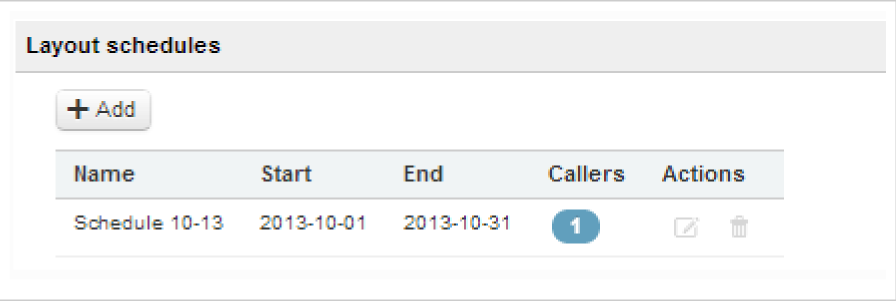
List of configured layout schedules.

Layout schedules

This portlet provides an overview of layouts scheduling configuration and provides access to layout schedule editor.

For each schedule, the table shows:

- *Name*: the layout schedule name
- *Start*: the begin of the validity interval
- *End*: the end of the validity interval
- *Callers* : list the callers that have been associated to the schedule
- *Actions*: For each schedule the following actions are available:
    - *Edit*: edit the schedule to manage its configuration
    - *Remove*: remove a schedule from the system.



Layout schedules

Name	Start	End	Callers	Actions
Schedule 10-13	2013-10-01	2013-10-31	1	

Layout scheduling portlet.

Configuring a schedule

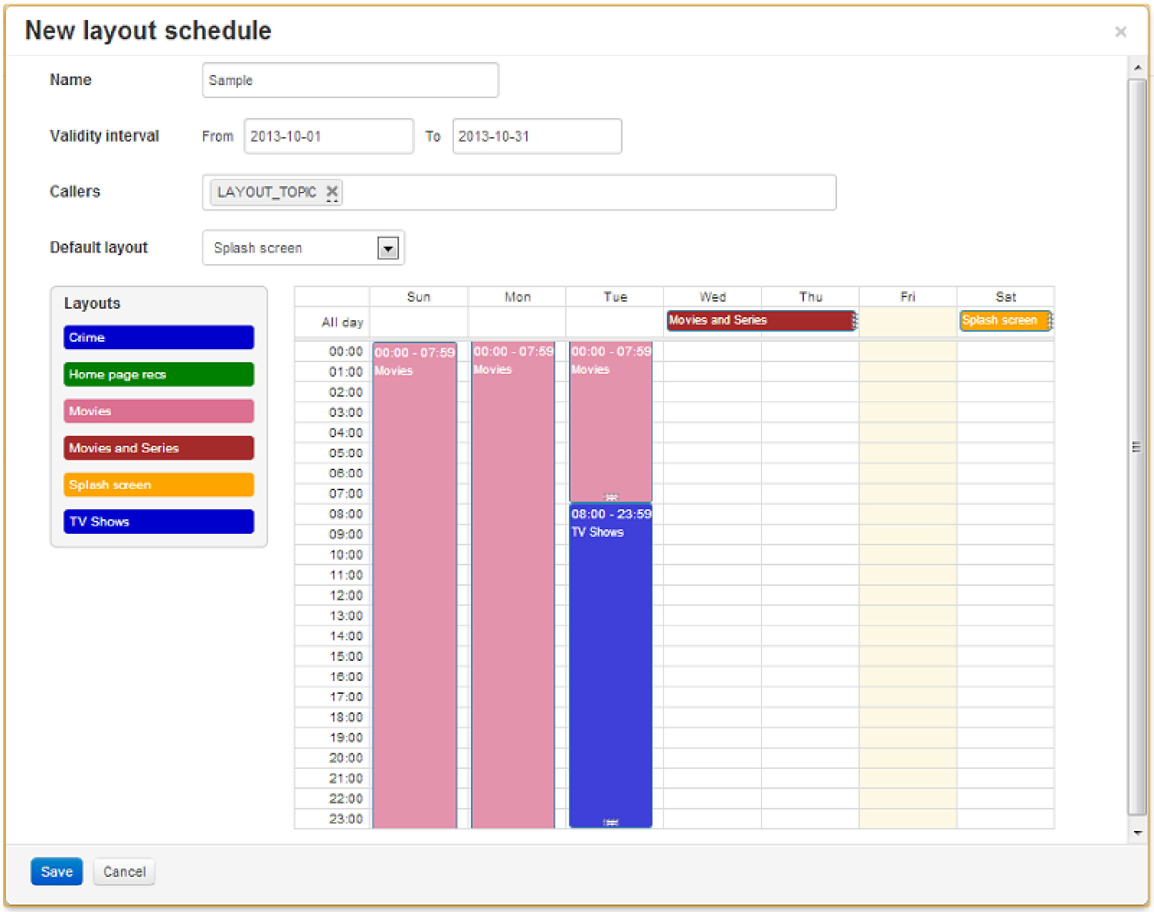
This section describes how it is possible to define and edit a layout schedule.

To define a new schedule, click the *new* button in the layout scheduling portlet
To edit an existent schedule, click the *edit* icon of the desired scheduling.

> ⚠ **Important note**
> When saving a layout scheduling configuration, the system checks for conflicts on scheduling.
>
> A conflict occurs if another layout schedule has been already defined for an overlapping validity interval, for one or more of the callers associated to the new schedule.

The scheduling editor dialogue requires the following information:

- *Name*: the name of the schedule. It is useful to retrieve the schedule from the layout scheduling portlet. Once a schedule has been defined, the name cannot be changed.
- *Validity interval*: the start day and the end day of the schedule
- *Callers*: the list of callers for which the schedule has to be defined.
- *Default layout*: the layout that is used when no scheduling is defined within the validity interval
- *Schedule configuration*: the configuration of the schedule among time, defined in a weekly calendar.



New layout schedule

Name: Sample

Validity interval: From 2013-10-01 To 2013-10-31

Callers: LAYOUT_TOPIC

Default layout: Splash screen

Schedule editor.

Weekly configuration

Schedule configuration editor provides a weekly calendar that can be filled by positioning the layouts in the desired time slots. This allow to configure layout scheduling with a per hour granularity within a schedule week.

> This configuration is not mandatory. If a slot is left empty, default layout of the schedule is applied to the requests that occurr during empty slots.

To configure a week of schedule, drag&drop the desired layouts from the list of available (on the left) to the calendar. Then resize the box related to the layout to fit desired time slot.

> Layout schedule cannot overlap each other.

Dynamic Layout

This page provides an overview of the personalized streams configuration. See also the page personalized streams for further details.

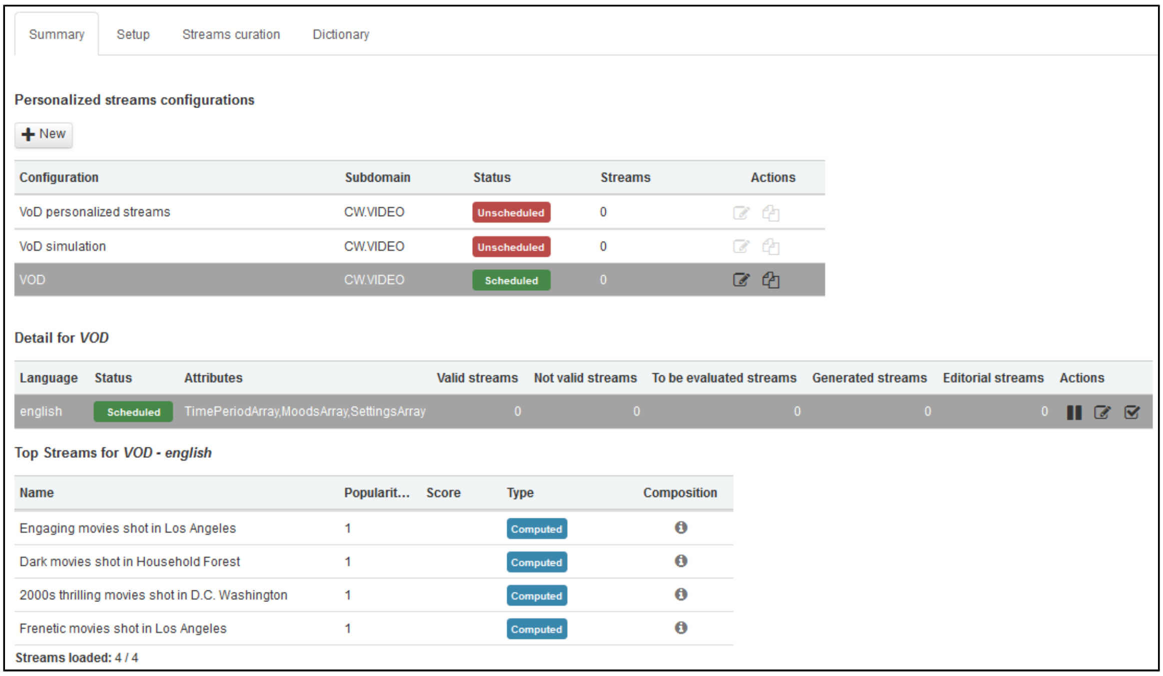
The configuration page is divided into four sections:

- Summary. It allows creating new configurations and seeing all the defined configurations.
- Setup. It allows configuring the core properties to manage the generation of personalized streams.
- Streams curation. It provides the tools to validate the generated personalized streams.
- Dictionary. It allows setting the advanced properties to clean and transform attributes.

Dynamic Layout - Summary

The overview section allows to:

- **create a new configuration**, i.e., a set of *properties* that control the generation of personalized streams. The new configuration requires to specify: the name, the subdomain, and one or more languages.
- **shows all the configurations** that have been defined.



Personalized streams configurations

Configuration	Subdomain	Status	Streams	Actions
VoD personalized streams	CW.VIDEO	Unscheduled	0	
VoD simulation	CW.VIDEO	Unscheduled	0	
VOD	CW.VIDEO	Scheduled	0	

Detail for *VOD*

Language	Status	Attributes	Valid streams	Not valid streams	To be evaluated streams	Generated streams	Editorial streams	Actions
english	Scheduled	TimePeriodArray,MoodsArray,SettingsArray	0	0	0	0	0	

Top Streams for *VOD - english*

Name	Popularit...	Score	Type	Composition
Engaging movies shot in Los Angeles	1		Computed	ℹ
Dark movies shot in Household Forest	1		Computed	ℹ
2000s thrilling movies shot in D.C. Washington	1		Computed	ℹ
Frenetic movies shot in Los Angeles	1		Computed	ℹ

Streams loaded: 4 / 4

In turn, for each configuration, the summarizing table reports:

- **subdomain**. The subdomain streams are to be generated for.
- **global status**. The global status can be either *scheduled* (when the configuration is scheduled for at least a language) or *unscheduled* (if the configuration is unscheduled for all languages).
- **number** of generated **streams**.

Two actions are available on an existing configuration:

- it can be **cloned**.

- adding additional **languages** to the configuration.

By clicking on a configuration the interface shows the details of each language defined for the configuration: the status (scheduled or unscheduled), the list of defined attributes, and the number of generated streams (separated in valid, not valid, to be evaluated, generated, and editorial). For each language of a given configuration it is possible:

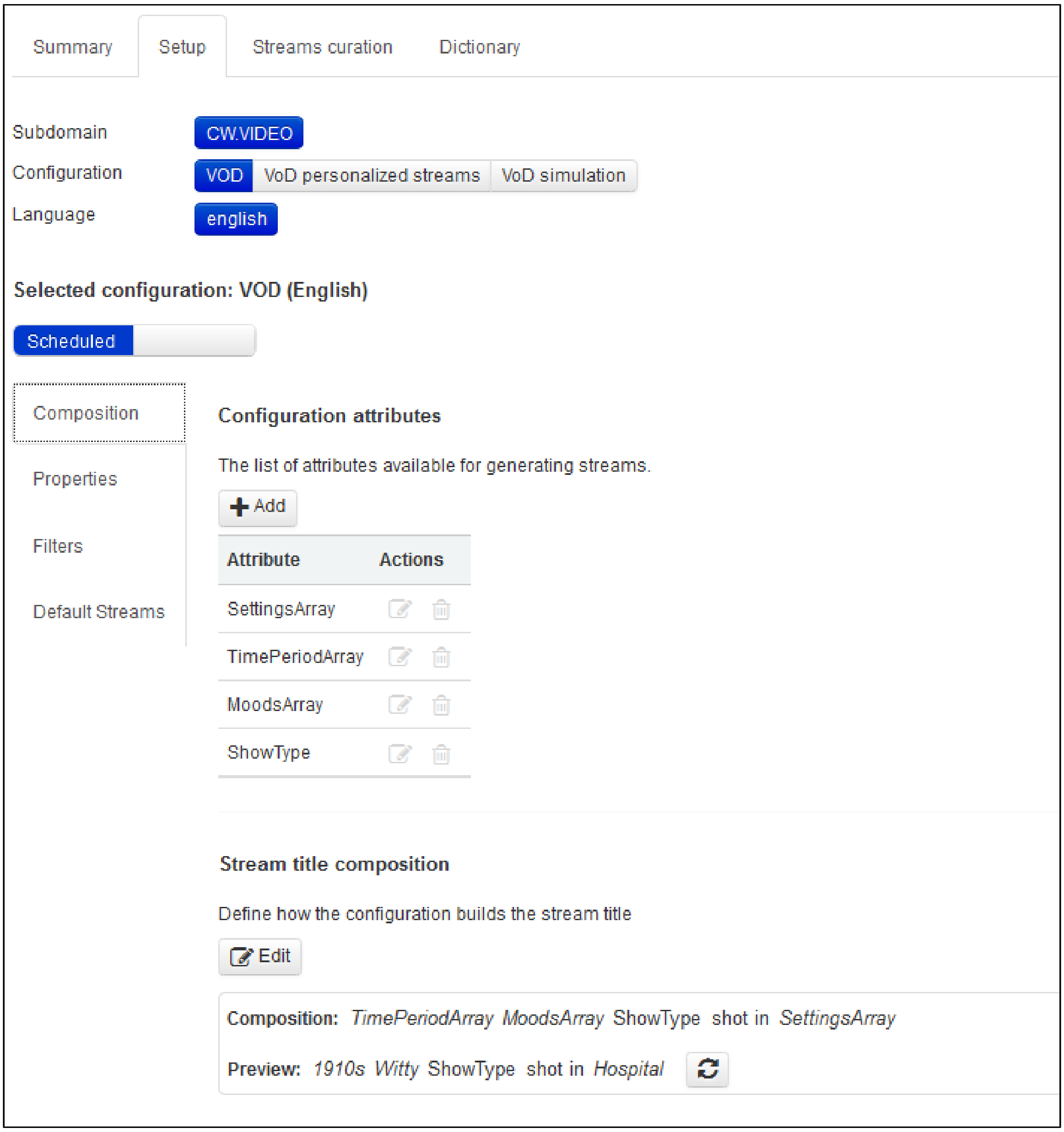
- **schedule/unschedule**. Scheduling a language of a certain stream configuration means enabling the *stream generator task* to process it next time it will be run.

> Unscheduling a language of a given stream configuration does not mean that the related streams that have been generated will be disabled, but it only implies that they will not be updated next time the *stream generator task* will be run.

- **edit** its properties. This action leads to the stream setup section.
- **manage** the generated streams. This action leads to the stream curation section.

Dynamic Layout - Setup

This section is used to configure the properties of a certain language of a given configuration (associated to a given subdomain). You can select a specific <configuration, language> pair by means of the *subdomain*, *configuration*, and *language* filters in the top of the section. Alternatively, you can directly point to the desired properties by selecting the "edit" action in the by-language configuration details available in the summary section.



Selected configuration: VOD (English)

Configuration attributes

The list of attributes available for generating streams.

Attribute	Actions
SettingsArray	
TimePeriodArray	
MoodsArray	
ShowType	

Stream title composition

Define how the configuration builds the stream title

Composition: *TimePeriodArray* *MoodsArray* ShowType shot in *SettingsArray*

Preview: *1910s* *Witty* ShowType shot in *Hospital*

Once a <configuration,language> pair has been selected, it is possible to toggle its **status** (schedule, unschedule) or edit the **properties** explained in the following.

Composition

Properties that define how the stream has to be composed, in particular:

- **attributes**: the list of attributes (e.g., genre and country) that can be used to form the stream. E.g., a stream can be configured to use the attributes .
- **title composition**: the schema to generate the stream title. The name of a stream is composed by an ordered list of textual blocks. Each block roughly corresponds to either a single word (e.g., the genre "Comedy") or an entity (e.g., the actor "Tom Cruise"). The properties of the name generation allows to define which blocks can be used and their order. The text of a block is the value of an attribute of the stream (e.g., the genre "comedy", the country "USA", the actor "Tom Cruise"). In addition, one of the block, denoted as **subject** can be forced to be always present in the stream title. The subject can be either one of the attribute or a static text (i.e., a user-defined text).

  > In the case the subject is based on an attribute (e.g., a stream has necessarily to be composed at least by the attribute "genre" of a movie), the streams that do not have such attribute will be set to conflict status (see status in stream validation), waiting for a manual user validation

  .

  > Cleaning and transformation of attribute values (e.g., from "USA" to "American") can be configured in the global properties (see dictionary section), i.e., they are not specific of a specific stream generation but they are defined for specific combinations of subdomain and language

  .

Properties

These settings control the core of stream generation process.

Generic settings

The main **generic settings** to tune are:

- **minimum number of items included in a personalized stream**. The process will estimates the number of items that are candidate to be included in the stream; personalized streams with less than this number of items will be filtered out.

  > The estimate of the number of items canditate to form the stream can be finer tuned by means of two advanced options: *exclude items already consumed* by the user and to *enable equality set*. See advanced options for further details.

- **maximum number of streams per user**. The process will generate at most this number of personalized streams for each user in the subdomain.
- **minumum** and **maximum number of attributes** to be used to compose a stream. These properties constrain the cardinality of the attributes composed to generate personalized streams.
- **popoularity threshold**. Streams that are assigned to a number of users lower than this threshold are filtered out.

Stream titles settings

The configuration editor allows also to set some properties related to the generation of the **stream titles**:

- **max length of stream title**. It limits the number of characters to be used in the stream title. Streams that have been generated names longer than this threshold are set in *conflict* status (see personalized streams validation for details about the stream status).
- **item heterogeneity**. This option aims at increasing the diversity among the personalized streams of the same users. You can configure a maximum percentage of item overlap among the streams. If two personalized streams have a percentage of common items greater than this value, only the most relevant one is maintained while the other is discarded.
- **attribute heterogenity**. This option aims at increasing the diversity among the personalized streams of the same users. You can select one or more attributes (e.g., genre, actors) whose values will be forced to appear at most once among the personalized streams of a certain user. For instance, if the "genre" attribute is selected, there can be at most a personalized stream composed - among the other attributes - by the genre "comedy".
- **stream merging**. This option enables the process to compact multiple personalized streams in a single personalized stream. You can select one or more attributes (e.g., genre, actors) that two personalized streams must have in common in order to be merged. As an example, if the "actor" attribute is selected and there exist two separate streams - for a given user - sharing the same actor but one has genre "comedy" and the other genre "drama", the merging option will compact them in a single stream whose attributes are the common actor and the two genres (comedy and drama). For a finer tuning, you can set the *maximum number of merges*, i.e., the maximum merges of a stream (e.g., a value equals to 2 means that at most a stream can be merged with other two streams, chosen among the most relevant).

Advanced settings

Finally, the **advanced** options allow to configure particular options:

- **exclude items already consumed**. This option affects how the process estimates the number of items candiate to compose a stream. If this option is enabled, for instance, the estimate will exclude movies already watched by the user.

  > ℹ This option only affects the estimate, not the content of personalized streams that will be presented to the user.

- **enable equality set**. This option affects how the process estimates the number of items candiate to compose a stream. If this option is enabled, for instance, the estimate will count all episodes of the same TV series as one item.

  > ℹ This option only affects the estimate, not the content of personalized streams that will be presented to the user.

- **stream status to reprocess** on next execution. Streams already generated can be forced to be reprocessed on next execution of the *stream generator task*. This option allows to set which streams are to be reprocessed, according to their current status. You can opt to reprocess streams in *to evaluate* or in *conflict* status.

Filters

Filters are boolean conditions applied to the item attributes whose objective is to limit the items of the subdomain that will be included in the personalized streams.
For each attribute you can set which values are to be present (e.g., to select only items that are "TV series") or are NOT to be present (e.g., to select only items that are NOT "TV series").

Default streams

The streams displayed to the user can be configured so that a certain number of them is picked up from a set of selected streams (already existing and validated), referred to as default streams.

> ℹ Note that the default streams can be indifferently selected among the editorial streams and the computed streams. (see Stream source)

Properties

- **Number of default streams per user**. Defines the number of personalized streams presented to the user that are picked up from the set of default streams. This number is limited by the *maximum number of streams per user* configured among the generic settings. For instance, if *maximum number of streams per user* is equals to 10 and *Number of default streams per user* is equals to 3, it means that the user will be displayed at most 10 personalized streams, 3 choosen among the default streams and 8 among the personalized stream automatically *computed* by the stream generator.
- **Default streams choice method**. It defines the policy to select which default streams show to the user.
  - *default* selects the default streams in the same order as they have been defined
  - *by stream score* selects the default streams the most relevant for the user
- **Default streams positioning criteria**. It defines the policy to mix the default streams together with the other personalized streams.
  - *by stream score* sorts the streams according to the relevance for the user
  - *by layout* allows to manually defined in which using the default streams and in which position using the computed personalized streams. For instance, I can configure that 2 default streams that have been set are to be put in 3rd and 10th position; the other positions will be filled with the generated personalized streams.

Configuration default streams

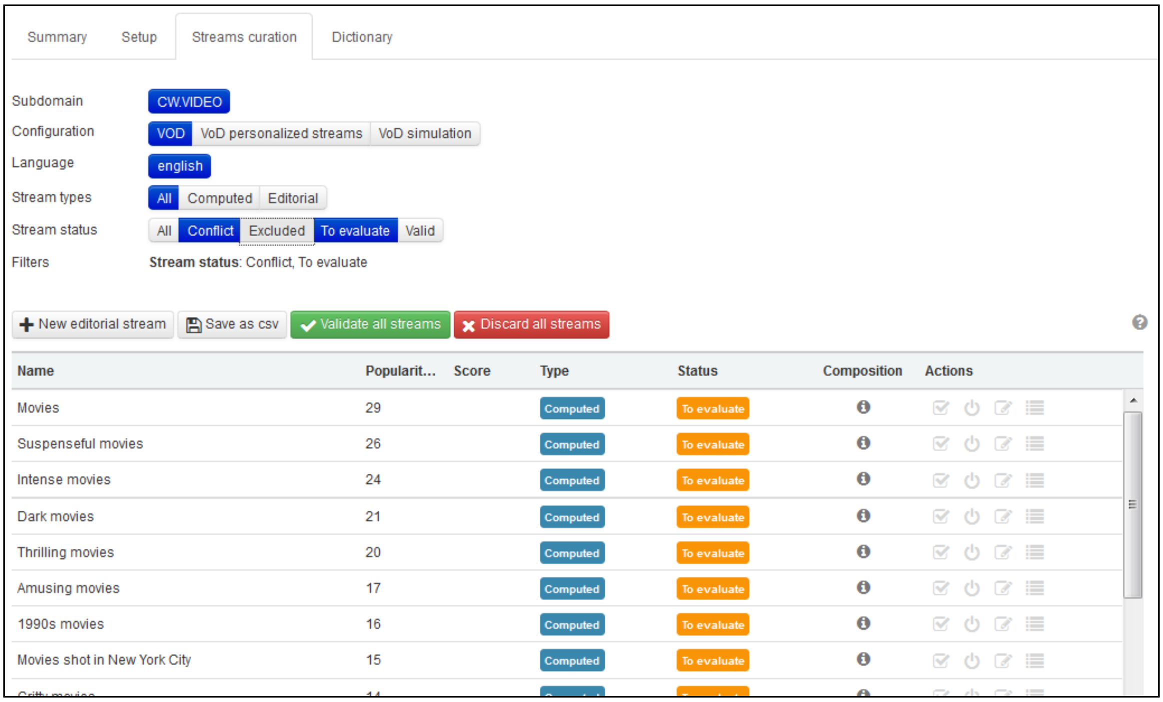
This editor allows to define the list of default streams. You can select any existing streams both among the list of the ones computed by the generator and among the list of editorial streams manually defined in the stream validation section.

Fallback streams

Some defaults streams can be marked as **fallback streams**. Such streams will be used as fallback by the stream generator to fill the list of streams displayed to the user only in the case - for any reason - it was not possible generating the requested number of personalized streams.

Dynamic Layout - Curation

This section is used to control the personalized streams that are meant to be displayed to the user.
Select a subdomain, a configuration, and a language to show the available streams. For a certain <configuration, language> pair, you can further filter the subset of available streams to visualize by filtering the streams on the basis of their source (i.e., *computed* or *editorial*) and their status (i.e., *to_evaluate*, *valid*, *excluded*, *conflict*).



Subdomain: CW.VIDEO
Configuration: VOD | VoD personalized streams | VoD simulation
Language: english
Stream types: All | Computed | Editorial
Stream status: All | Conflict | Excluded | To evaluate | Valid
Filters: **Stream status**: Conflict, To evaluate

Name	Popularit...	Score	Type	Status	Composition	Actions
Movies	29		Computed	To evaluate	ℹ	
Suspenseful movies	26		Computed	To evaluate	ℹ	
Intense movies	24		Computed	To evaluate	ℹ	
Dark movies	21		Computed	To evaluate	ℹ	
Thrilling movies	20		Computed	To evaluate	ℹ	
Amusing movies	17		Computed	To evaluate	ℹ	
1990s movies	16		Computed	To evaluate	ℹ	
Movies shot in New York City	15		Computed	To evaluate	ℹ	

The main actions available are:

- **Defining new editorial streams** that can be configured as *default streams* from the stream configuration section.
- **Editing stream name**, changing the name that was automatically generated.
- **Managing stream status**, either accepting or discarding streams in *conflict* or *to_evaluate* status, or even discarding existing valid streams

New streams

This function allows creating a new **editorial stream** by defining:

- **Name**. The *stream name* assigned to the stream.
- **Composition**. The set of conditions to compose the stream; each condition specifies the value that an attribute must either have or not have. For example, a condition can define that the attribute "genre" must have (or must not have) the value "comedy".

Editorial streams can be used as *default streams* within the list of streams presented to the user. See stream configuration section for further configuration details.

Editing stream name

Each existing stream can be edited by modifying its name. In particular in the case of *computed* streams it can be convenient in some cases manually changing the name the has been automatically generated by the system with a fancier text. Note that anytime the name automatically generated by the system can be reset.

Managing stream status

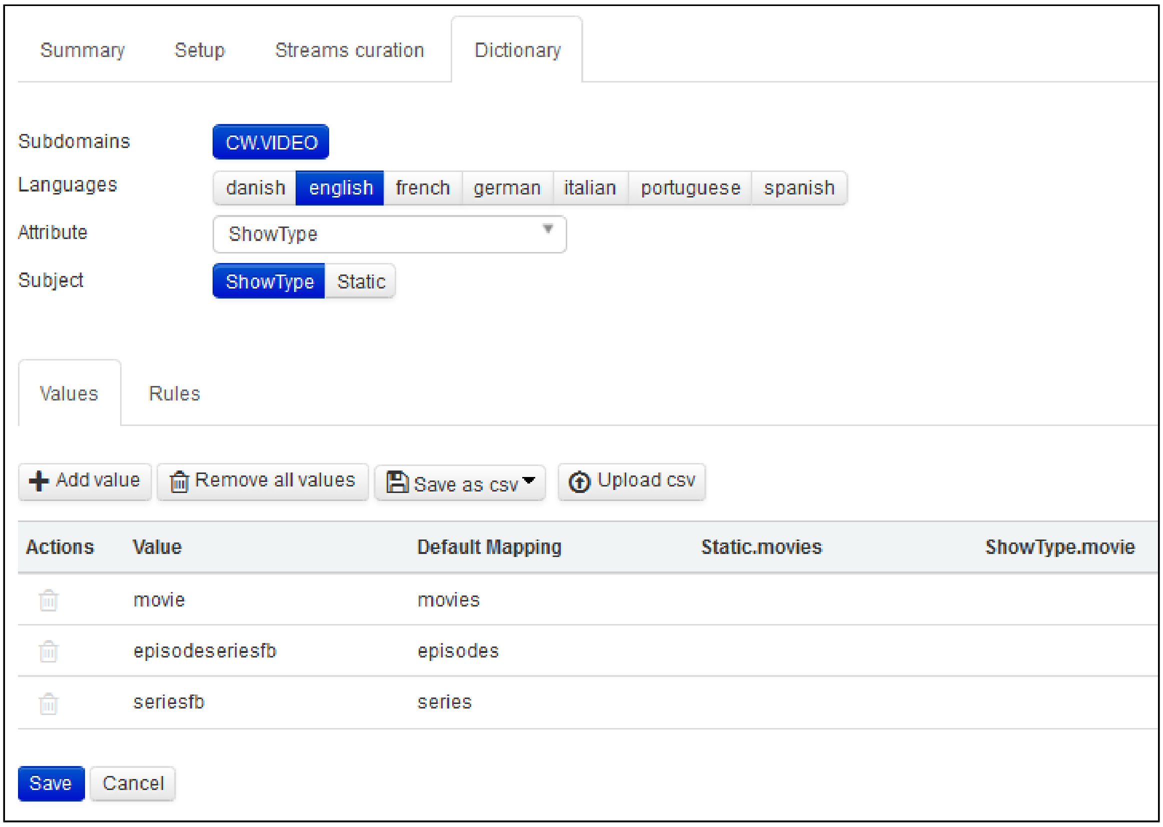
The status of existing streams can be changed, in particular:

- **conflict** streams can be either accepted or discarded
- **to_evaluate** streams can be either accepted or discarded
- **valid** streams can be discarded
- **excluded** streams can be accepted

> **Discarding** a stream means changing its status to *excluded*. **Accepting** a stream means changing its status to *valid*

Dynamic Layout - Dictionary

This section provides some functionalities for optimizing the generation of stream names. For instance, it can be convenient for some values to be converted from noun to adjective - e.g., from USA to American - so that a fancier name can be generated.



Subdomains: CW.VIDEO

Languages: danish, english (selected), french, german, italian, portuguese, spanish

Attribute: ShowType

Subject: ShowType (selected), Static

Actions	Value	Default Mapping	Static.movies	ShowType.movie
	movie	movies		
	episodeseriesfb	episodes		
	seriesfb	series		

> These settings apply to all configurations defined for the selected language of the given subdomain.

For each language of a subdomain, any attribute configured to be used to compose the stream can be manipulate by means of

- **values replacements** where a source value is replaced with a destination value.
- **replacement rules** where rules - among which regular expressions - are applied to transform values.

Values replacements

This tool consists of a set of **substitutions** to replace existing values. For a given attribute (e.g., the "genre"), the specified **value** will be replaced with the defined **replacement**. For example, the genre "series" might be replaced by "TV series", or the country "United States of America" might be replaced by "USA".

The defined replacement values can be exported as *csv* to be later imported.

Replacement rules

A replacement rule consists of a *pattern* and a *replacement*. The pattern describes the text to search for within the value of an attribute, the replacement how this text is to be replaced. In addition, the rule pattern and replacement can be set to be treated as *regular expressions*.
As as example of replacement rule, it can be created a rule that applies to the attribute "genre" with pattern equals to "/" and replacement equals to "~~-" that simply replaces all the occurrences of the character "/" within the values of the genre with the character "-~~" (e.g., the genre "sci/fi" would be transformed into "sci-fi").

Default replacement and per-subject replacement

Each entry (either a *value replacement* or a *replacement rule*) specifies a *default replacement* and a set of optional *per-subject replacement*. The **per-subject word** replacement indicates, for each value of the configured subjects, the replacement to apply. In the case there is not a replacement configured for a certain value of the current subject, the **default** replacement value will be used.

For instance, you can configure a *value replacement* for the **value** "subitem" (of a certain attribute) to be transformed into:

- "episode" in the case the *subject* is "TV series"
- "part" in the case the *subject* is "movie"
- "chapter" in any else case (*default* replacement)

AutoComplete

In this section it is possible to specify the basic configuration for the search autocomplete capability.

The options that can be specified in this sections are:

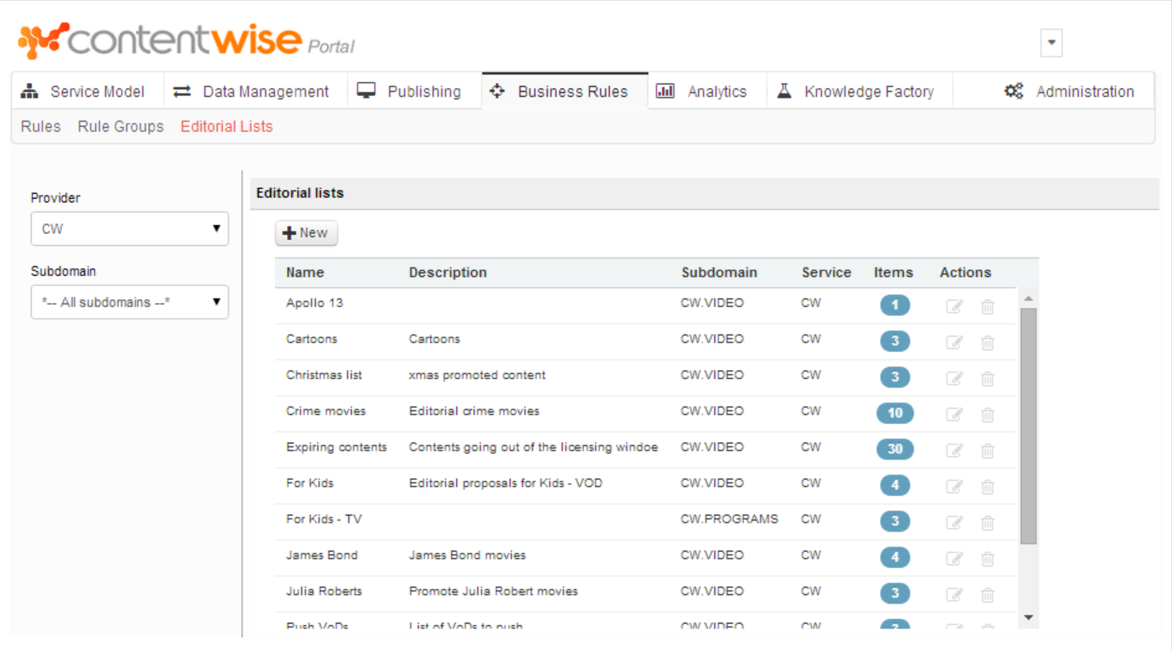
Name: The name to identify the use case
Description: A short description of the use case
Sub Domain: The sub-domain where to apply the use case
Language: The language to use when the use case is applied
Number of Suggested Results: Number of autocomplete options
_Matching Criteria: Specify if the autocomplete option must start or contain the typed search string

Editorial Lists

This page provides an overview of the editorial lists configured in the system and provides access to the pages that allow to create and edit editorial lists.

See create/edit a new editorial list to create or edit an editorial list.

The *Editorial lists* portlet requires *provider* to be selected.



Provider: CW

Subdomain: *-- All subdomains --*

Editorial lists

Name	Description	Subdomain	Service	Items	Actions
Apollo 13		CW.VIDEO	CW	1	
Cartoons	Cartoons	CW.VIDEO	CW	3	
Christmas list	xmas promoted content	CW.VIDEO	CW	3	
Crime movies	Editorial crime movies	CW.VIDEO	CW	10	
Expiring contents	Contents going out of the licensing windoe	CW.VIDEO	CW	30	
For Kids	Editorial proposals for Kids - VOD	CW.VIDEO	CW	4	
For Kids - TV		CW.PROGRAMS	CW	3	
James Bond	James Bond movies	CW.VIDEO	CW	4	
Julia Roberts	Promote Julia Robert movies	CW.VIDEO	CW	3	
Push VoDs	List of VoDs to push	CW.VIDEO	CW	[illegible]	

Editorial lists: list of configured editorial lists.

Editorial lists

This portlet lists all available editorial lists according to the selected provider, subdomain, and caller.

For each editorial list, the table shows:

- *Name*: the editorial list identifier.
- *Description*: a short description of the editorial list (optional).
- *Subdomain*: the subdomain for which the rule has been defined.
- *Service*: the service associated to the editorial list.
- *Number of items*: the number of items included in the editorial list (click on the button to display the list of items).
- *Actions*: the possible actions are:
  - *edit*: edits the editorial list information.
  - *delete*: removes the editorial list from the system.

Create or edit an editorial list

To create a new editorial list, click the *New* button in the Editorial lists portlet.
To edit, click on the *edit* button close to an existing editorial list in the Editorial lists portlet.

It is required to select a provider for which the editorial list has to be defined.

The following information has to be provided:

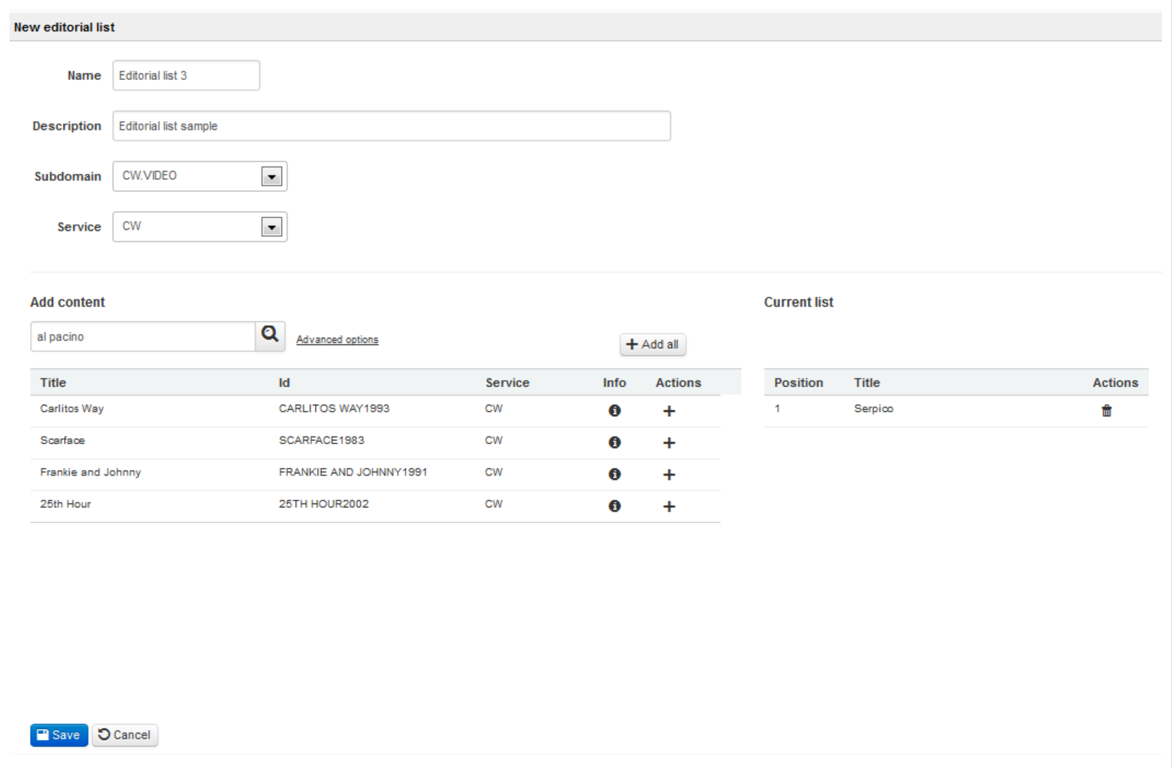
- *Name*: the editorial list identifier.

Warning
Valid characters are [A-Z a-z 0-9], -, , ., *space*

- *Description*: a short description of the editorial list (optional).
- *Subdomain*: the subdomain for which the editorial list has to be defined.
- *Service*: the service for which the editorial list has to be defined.

Finally, content can be added to the editorial list by first searching for content using the *search* field and then adding the selected items to the *current list*.

Warning
Once an editorial list has been created, the associated subdomain and service cannot be changed.



New editorial list

Name	Editorial list 3
Description	Editorial list sample
Subdomain	CW.VIDEO
Service	CW

Add content

al pacino

Title	Id	Service	Info	Actions
Carlitos Way	CARLITOS WAY1993	CW	ⓘ	+
Scarface	SCARFACE1983	CW	ⓘ	+
Frankie and Johnny	FRANKIE AND JOHNNY1991	CW	ⓘ	+
25th Hour	25TH HOUR2002	CW	ⓘ	+

Current list

Position	Title	Actions
1	Serpico	🗑

Editorial lists: Editorial list editor portlet.

Search

A search use case is the search configuration to use to create a layout use case showing a set of item from a search query.

To define a search use case it is possible to specify the following parameters:

Name: the name of the use case
Description: a short description of the use case
SubDomain: The sub-domain where the use case is applyied
Return all the fields: A flag to ask all the fields of the items in the search result. If not flagged it is possible to specify which fields the search service must return
Search On: Specify the list of fields to use for the search
Search Operator: The Logic to use for the search query
Item per page:The number of item included in the search query result
Language: The language to use to perform the search query
Return all facets: a flag to ask all the facets

UX Integration

Every page or use case created with the UX Builder can be returned to an external application via REST API. Each entity, page or use case, to be returned by the API service must be linked to an identifier called UX Reference.

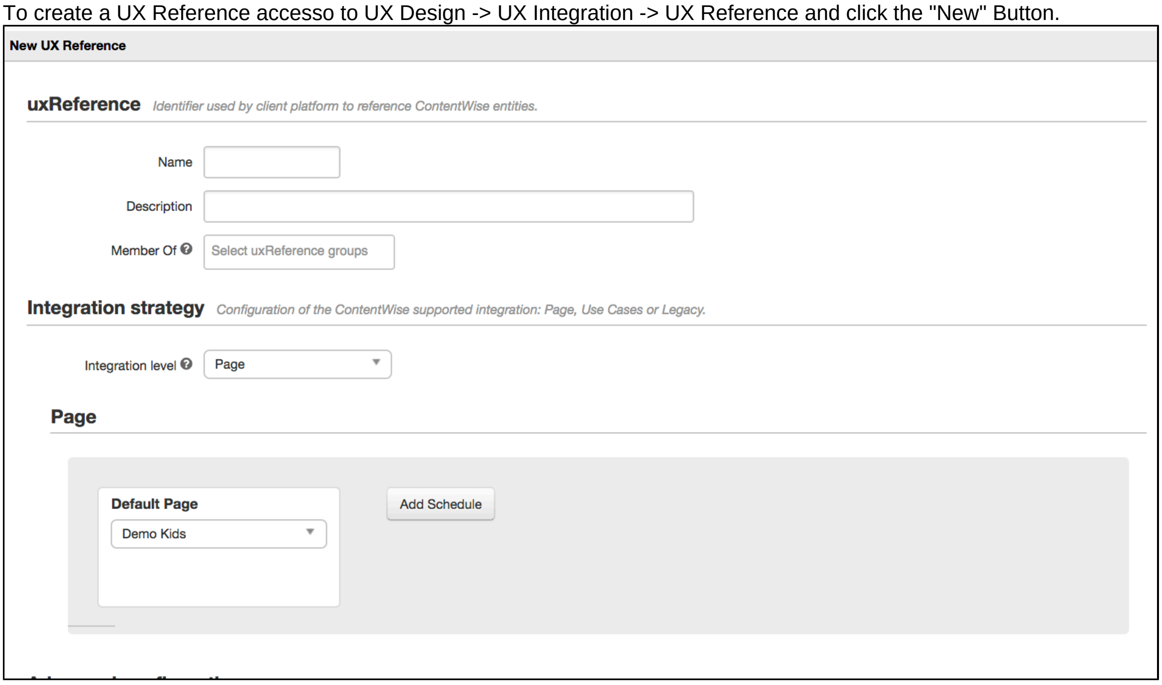
The UX Integration section of the portal can be used to manage the relationship between page and use case.

The UX Integration section provide access to UX Refence and UX Reference Groups.

UX Reference

A UX Reference is a unique identifier for any page or use case created with the UX Builder.

To create a UX Reference accesso to UX Design -> UX Integration -> UX Reference and click the "New" Button.



New UX Reference

uxReference *Identifier used by client platform to reference ContentWise entities.*

Name

Description

Member Of: Select uxReference groups

Integration strategy *Configuration of the ContentWise supported integration: Page, Use Cases or Legacy.*

Integration level: Page

Page

Default Page: Demo Kids

Fill the form with the required parameters:

1) UX Reference Name
2) Description
3) UX Reference Group Optional

4) Specify the integration Strategy. If Contentwise 6 is the first Contentwise implementation and you are not migrating from a previous version select Page.

5) Select the page (or use case accordingly to the selected integration strategy) to link to the UX Reference.

UX Reference Groups

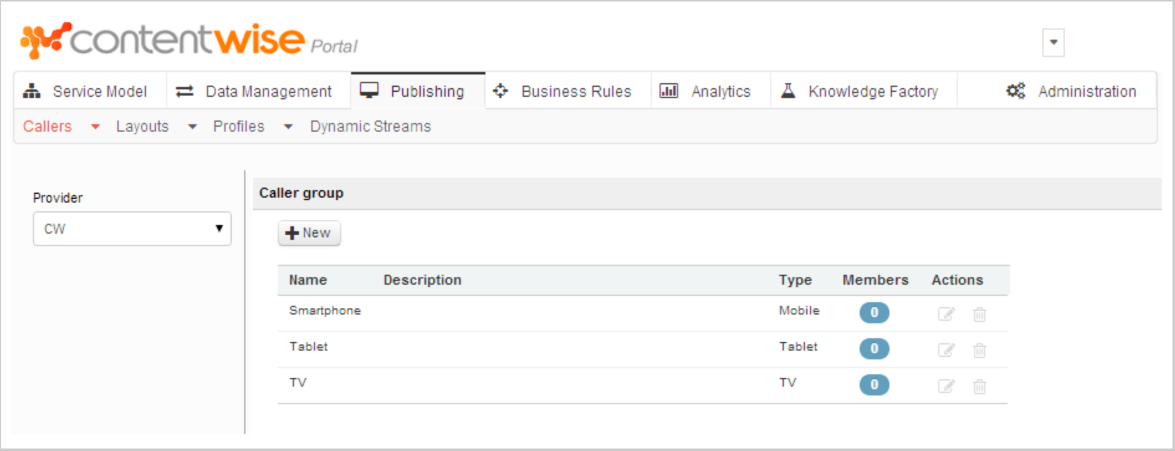
This page provides an overview of caller configurations and provides access to the pages that allow to create and manage caller groups.

The page contains the following portlets:

- UX Reference Groups\: lists the UX Reference Groups that have been defined and provides access to new/edit ux reference groups modals.

See create a new UX Reference group\ to create a new UX Reference group.

If a provider filter is not selected, you will be prompted to select one.



Provider: CW

Caller group

Name	Description	Type	Members	Actions
Smartphone		Mobile	0	
Tablet		Tablet	0	
TV		TV	0	

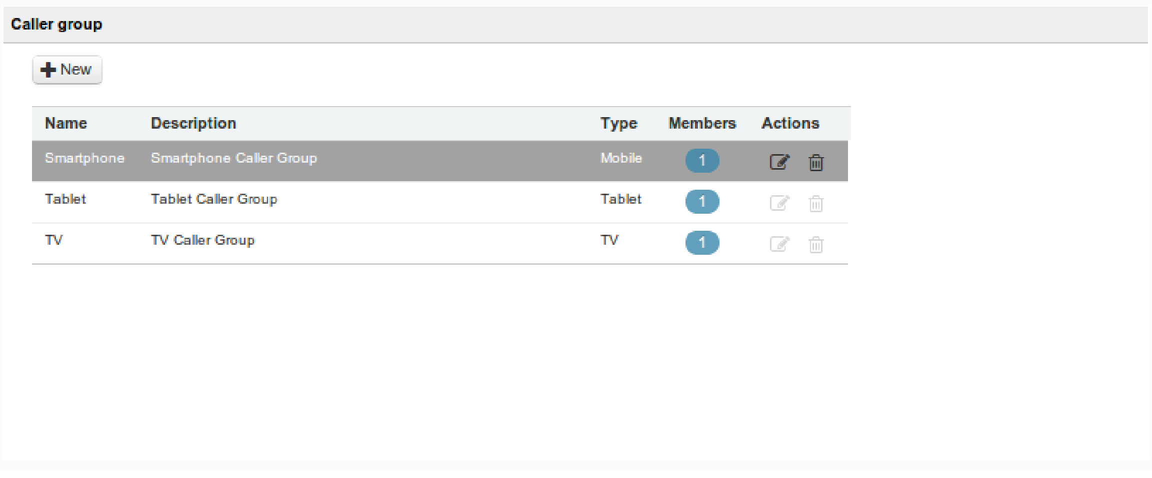
UX Reference Groups: list of configured UX Reference groups and related configurations.

UX Reference Groups

This portlet provides an overview of UX Reference group configuration and provides access to UX Reference Group configuration modals.

For each UX Reference groups, the table shows:
For each UX Reference Group, the table shows:

- *Name*: unique ux reference group identifier.
- *Description*: a short ux reference group description.
- *Type*: the ux reference group type.
- *Members*: the ux reference group members. A member is a ux reference which belongs to the UX Reference group.
- *Actions*: you can:
    - *edit*: modifies the UX Reference group configuration.
    - *delete*: remove a UX Reference group.



Name	Description	Type	Members	Actions
Smartphone	Smartphone Caller Group	Mobile	1	✎ 🗑
Tablet	Tablet Caller Group	Tablet	1	✎ 🗑
TV	TV Caller Group	TV	1	✎ 🗑

Caller Groups: Caller groups portlet.

Create a new UX Reference Group

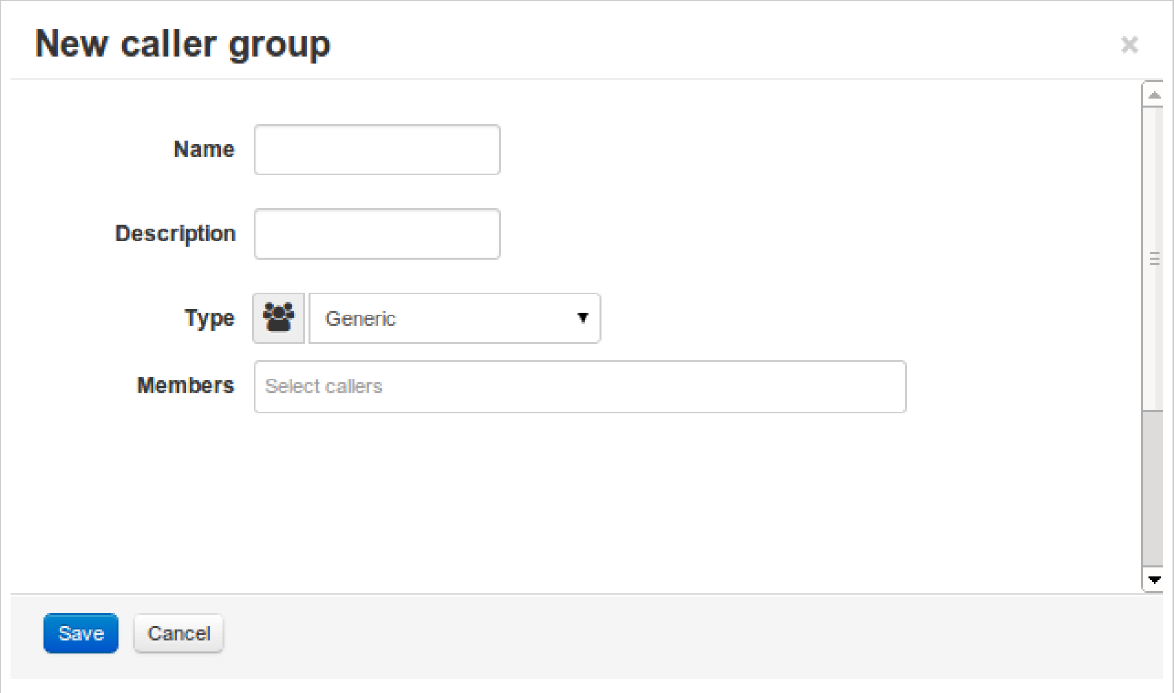
This section describes how to add a new UX Reference Group in the system.

To create a new UX Reference Group click the *New* button in the Caller Groups portlet\.

Definition

Define the general settings of the UX Reference

- *Name*: unique UX Reference group identifier.
- *Description*: a short UX Reference group description (optional).
- *Type*: the type of the UX Reference group. Once a UX Reference group has been defined, its type can be changed. Please note that there are not specific properties depending on the type; the type is only a classification tag.
- *Members*: the UX Reference that belong to the UX Reference group.



New caller group

Type: Generic

Caller Groups: New caller group modal.

Profiles

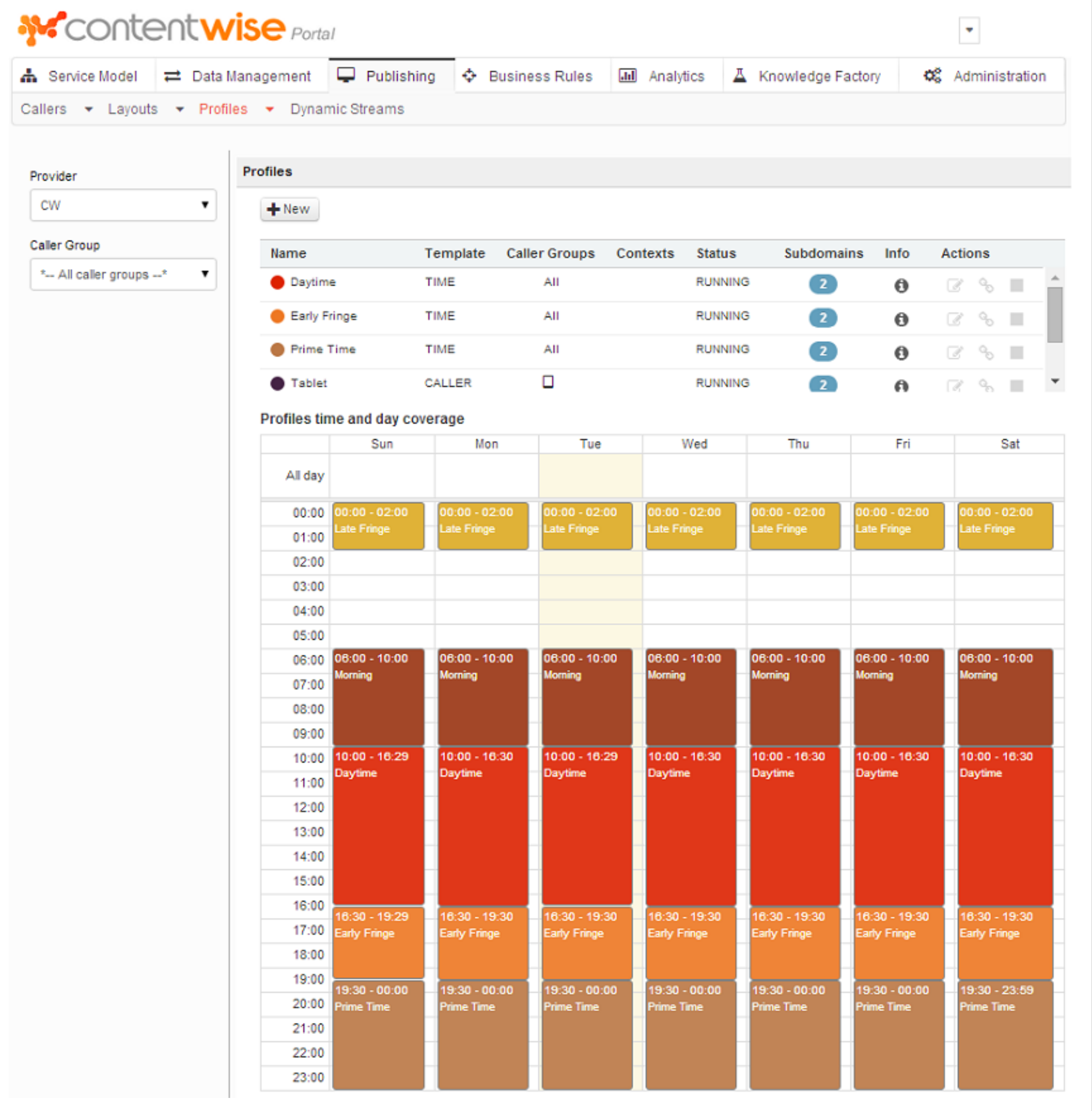
This page provides an overview of profile configuration and provides access to the pages that allow to manage profiles.

The page contains the following sections:

- Profiles: lists the profiles defined in the system and provides access to new profile page.
- Profiles time and day coverage: shows the profiles time and day distribution.

See create a new profile to create a new profile.

If a provider filter is not selected, you will be prompted to select one.
You can filter by caller groups to view only profiles defined for a particular caller group. If no caller groups filter is selected, all profiles are shown.



Provider: CW

Caller Group: *-- All caller groups --*

Profiles

Name	Template	Caller Groups	Contexts	Status	Subdomains	Info	Actions
● Daytime	TIME	All		RUNNING	2		
● Early Fringe	TIME	All		RUNNING	2		
● Prime Time	TIME	All		RUNNING	2		
● Tablet	CALLER	□		RUNNING	2		

Profiles time and day coverage

	Sun	Mon	Tue	Wed	Thu	Fri	Sat
All day							
00:00	00:00 - 02:00 Late Fringe	00:00 - 02:00 Late Fringe	00:00 - 02:00 Late Fringe	00:00 - 02:00 Late Fringe	00:00 - 02:00 Late Fringe	00:00 - 02:00 Late Fringe	00:00 - 02:00 Late Fringe
01:00							
02:00							
03:00							
04:00							
05:00							
06:00	06:00 - 10:00 Morning	06:00 - 10:00 Morning	06:00 - 10:00 Morning	06:00 - 10:00 Morning	06:00 - 10:00 Morning	06:00 - 10:00 Morning	06:00 - 10:00 Morning
07:00							
08:00							
09:00							
10:00	10:00 - 16:29 Daytime	10:00 - 16:30 Daytime	10:00 - 16:29 Daytime	10:00 - 16:30 Daytime	10:00 - 16:30 Daytime	10:00 - 16:30 Daytime	10:00 - 16:30 Daytime
11:00							
12:00							
13:00							
14:00							
15:00							
16:00	16:30 - 19:29 Early Fringe	16:30 - 19:30 Early Fringe	16:30 - 19:30 Early Fringe	16:30 - 19:30 Early Fringe	16:30 - 19:30 Early Fringe	16:30 - 19:30 Early Fringe	16:30 - 19:30 Early Fringe
17:00							
18:00							
19:00	19:30 - 00:00 Prime Time	19:30 - 00:00 Prime Time	19:30 - 00:00 Prime Time	19:30 - 00:00 Prime Time	19:30 - 00:00 Prime Time	19:30 - 00:00 Prime Time	19:30 - 23:59 Prime Time
20:00							
21:00							
22:00							
23:00							

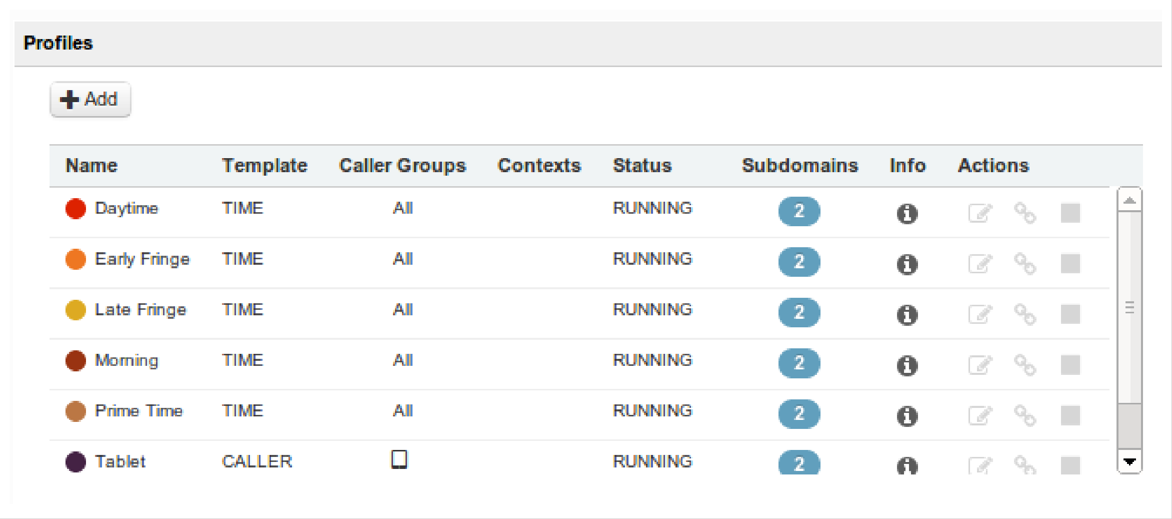
Profiles: list of profiles and related configurations.

Profiles

This section lists the profiles that are configured in the system and provides access to the wizard that allows to create a new profile.

For each profile, the table shows:

- *Name*: the name of the profile.
- _Template: the template shows which components are active for the profile.
- *Caller Groups*: the list of caller groups associated to the profile.
- *Contexts*: the list of contexts keywords associated to the profile.
- *Status*: the status of the profile. It can be one of the following:
  - RUNNING: the profile is valid and actually in use.
  - NEW: the profile has been defined but it has not been generated yet.
  - OBSOLETE: the profile is going to be deleted/substituted.
- *Subdomains*: the number of subdomains bound to the profile. A tool-tip with the list of subdomains will pop-up pointing the cursor over the number.
- *Info*: a description tool-tip. It contains more details about a profile:
  - *Types*: a list of couples item type/user type for which the profile is defined.
- *Actions*: you can:
  - *edit*: open the editor wizard.
  - *bind*: open the subdomain bindings modal.
  - *disable*: set the profile in status OBSOLETE. The profile will be deleted by batch process.



Profiles

Name	Template	Caller Groups	Contexts	Status	Subdomains	Info	Actions
● Daytime	TIME	All		RUNNING	2	ℹ	
● Early Fringe	TIME	All		RUNNING	2	ℹ	
● Late Fringe	TIME	All		RUNNING	2	ℹ	
● Morning	TIME	All		RUNNING	2	ℹ	
● Prime Time	TIME	All		RUNNING	2	ℹ	
● Tablet	CALLER	□		RUNNING	2	ℹ	

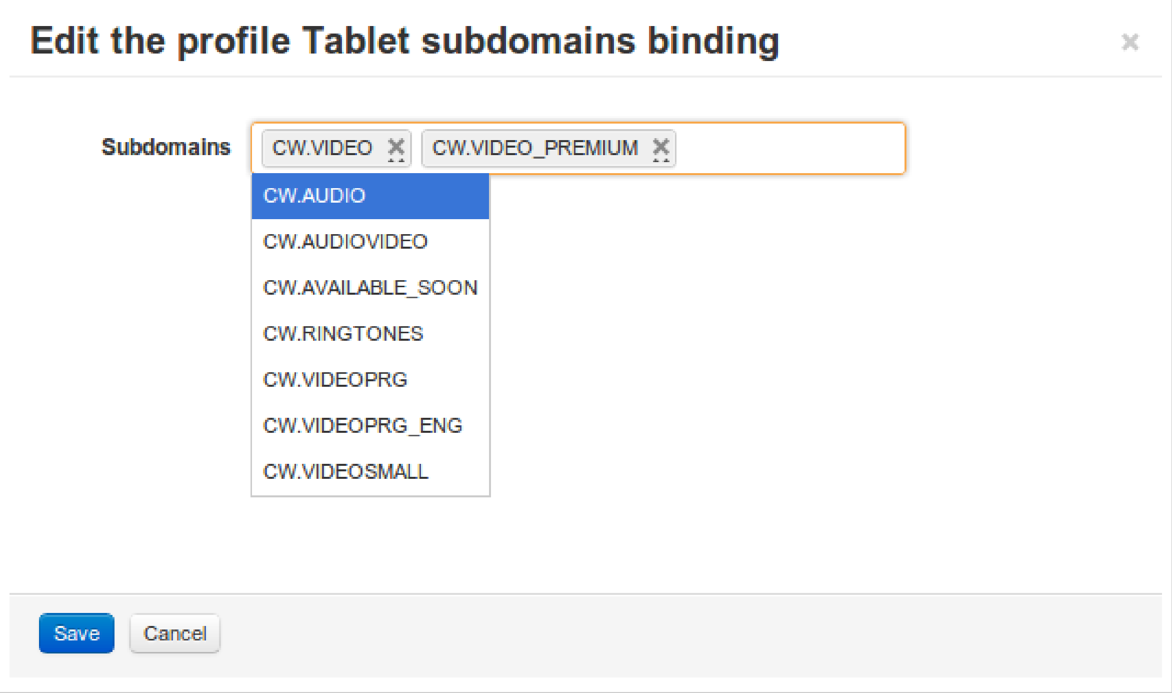
Profiles: list of profiles configured in the system.

Subdomain bindings modal

This modal provides allows you to edit the bindings between the profile selected in Profiles portlet and the subdomains that are configured in the system.

To modify the current configuration:

1. Add one or more subdomain in the multi selection input. You can also remove one or more subdomains from the list
2. Click *Save* to submit the new configuration, *Cancel* to undo the operation



Edit the profile Tablet subdomains binding

Subdomains: CW.VIDEO ✕ CW.VIDEO_PREMIUM ✕

Profiles: subdomain binding portlet (view and edit).

Create a Profile

This section shows how to add a new profile in the system.

To create a new profile click the *New* button in the Profiles portlet.

The new profile will be defined for the provider selected in the provider filter portlet.

> **Important Note**
> The profile will be created with status RUNNING. So it will be immediately available in the system.

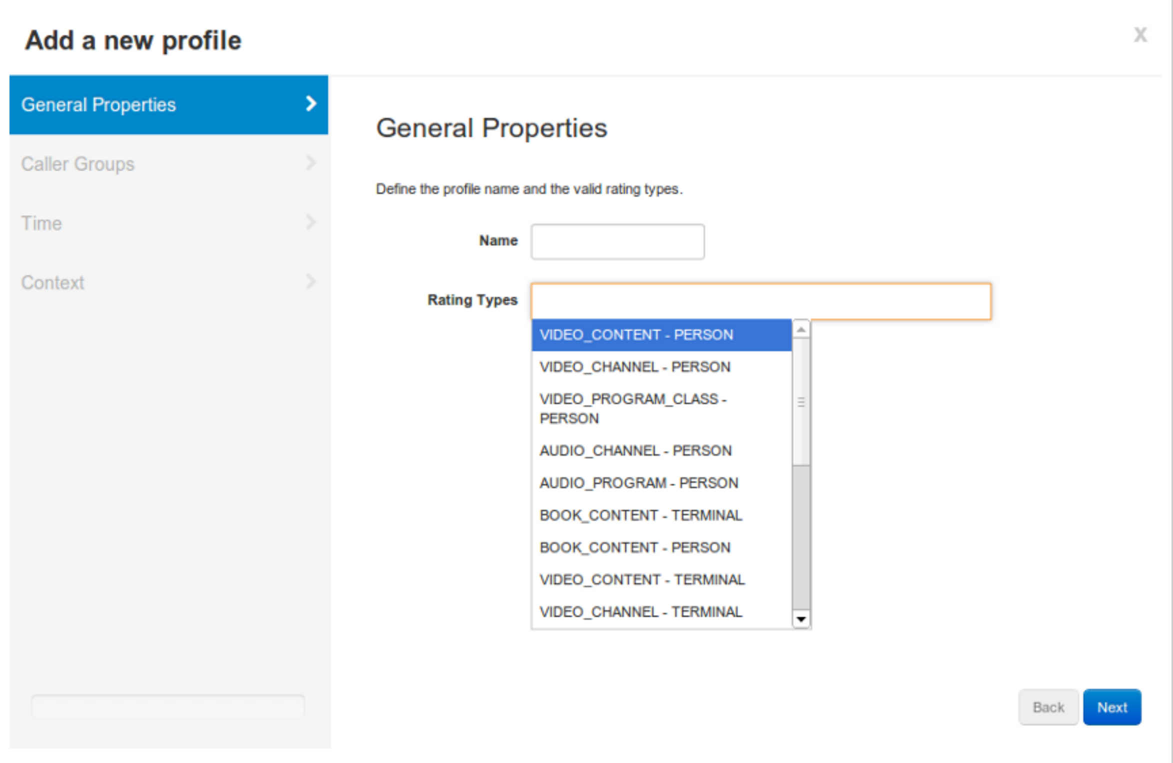
For each profile you can define a set of conditions that must be satisfied by ratings or accesses in order to be considered by the profile:

- Caller groups: user accesses and ratings (e.g., views, purchases) are profiled according to the caller. The rules are expressed by Caller Groups.
- Time: user accesses and ratings (e.g., views, purchases) are profiled according to a set of rules that identify the time slots of the week.
- Context: user accesses and ratings (e.g., views, purchases) are profiled according to the input contexts.

The creation wizard will guide you in the new profile definition.

General Properties

In the general properties step you can define the unique name of the profile and select the rating types you want to profile.



Add a new profile

General Properties

Define the profile name and the valid rating types.

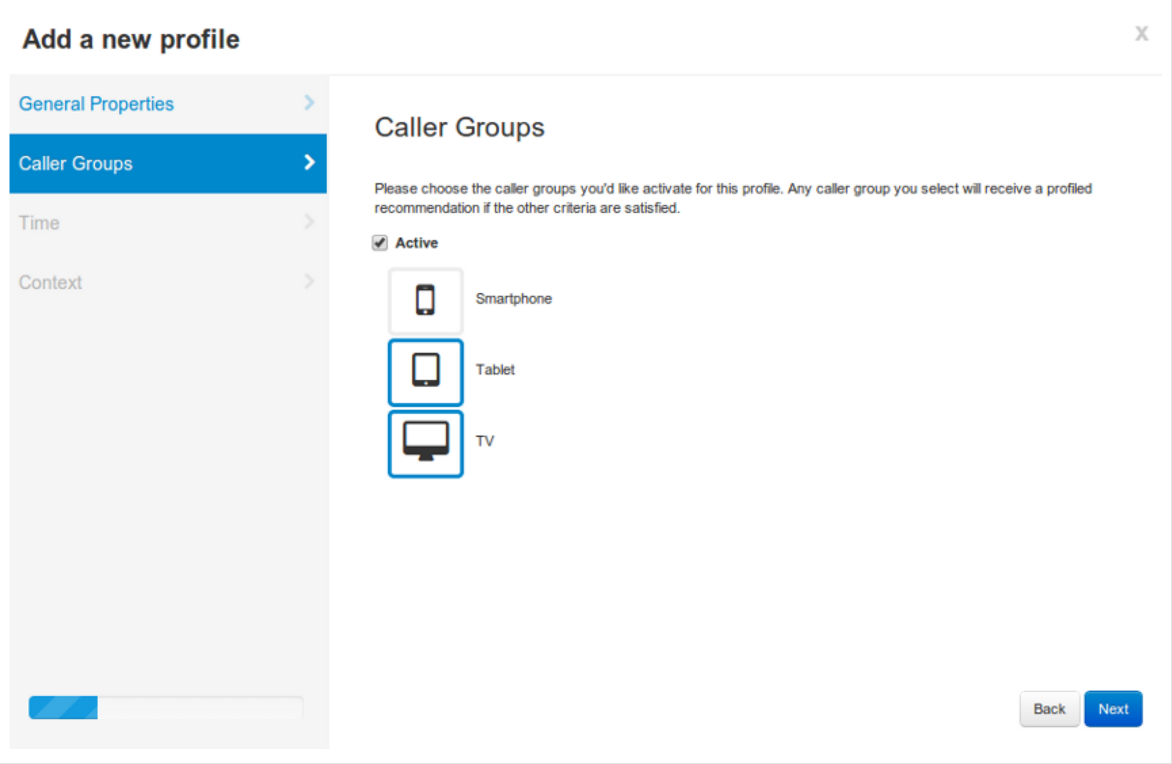
Name

Rating Types

Profiles: profile editor portlet.

Caller Groups

In the caller groups step you can activate the profiling by callers. Once the profiling is active, you can choose the caller groups to profile.



Add a new profile

Caller Groups

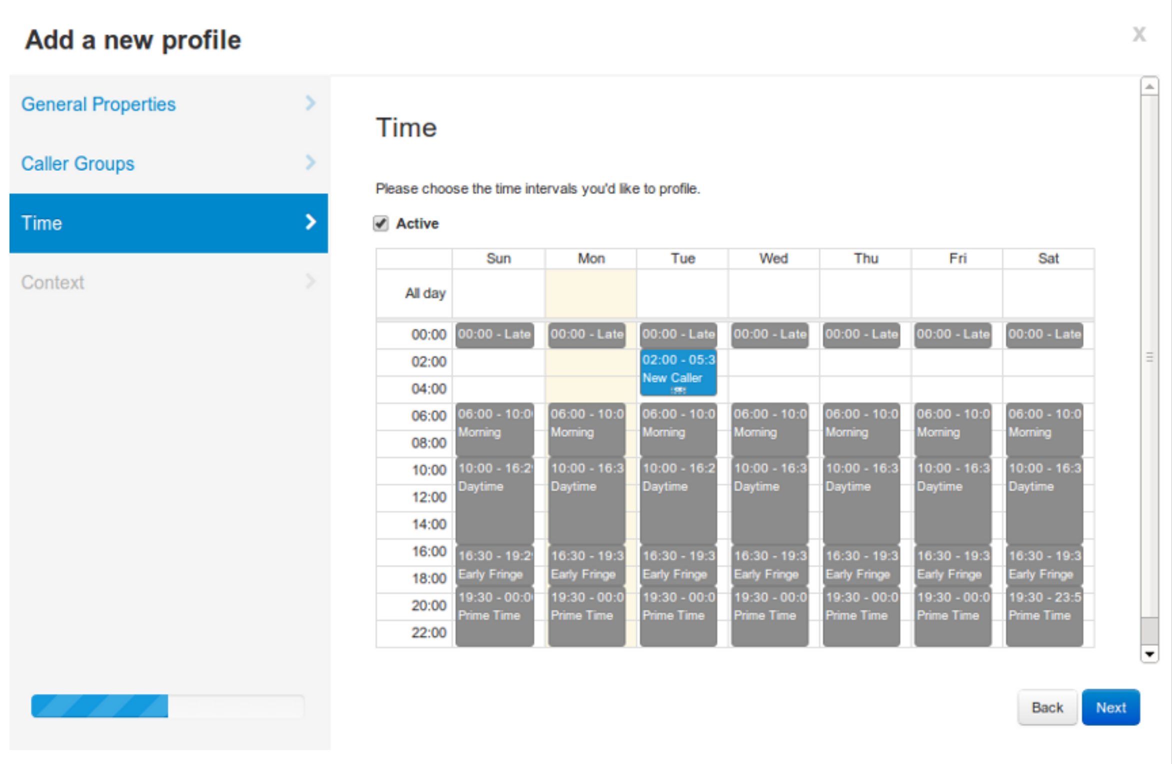
Please choose the caller groups you'd like activate for this profile. Any caller group you select will receive a profiled recommendation if the other criteria are satisfied.

☑ Active

Profiles: profile editor portlet.

Time

In the time step you can activate the time interval profiling. Once the profiling is active, you can create several time slot in the week calendar view.



Add a new profile

Time

Please choose the time intervals you'd like to profile.

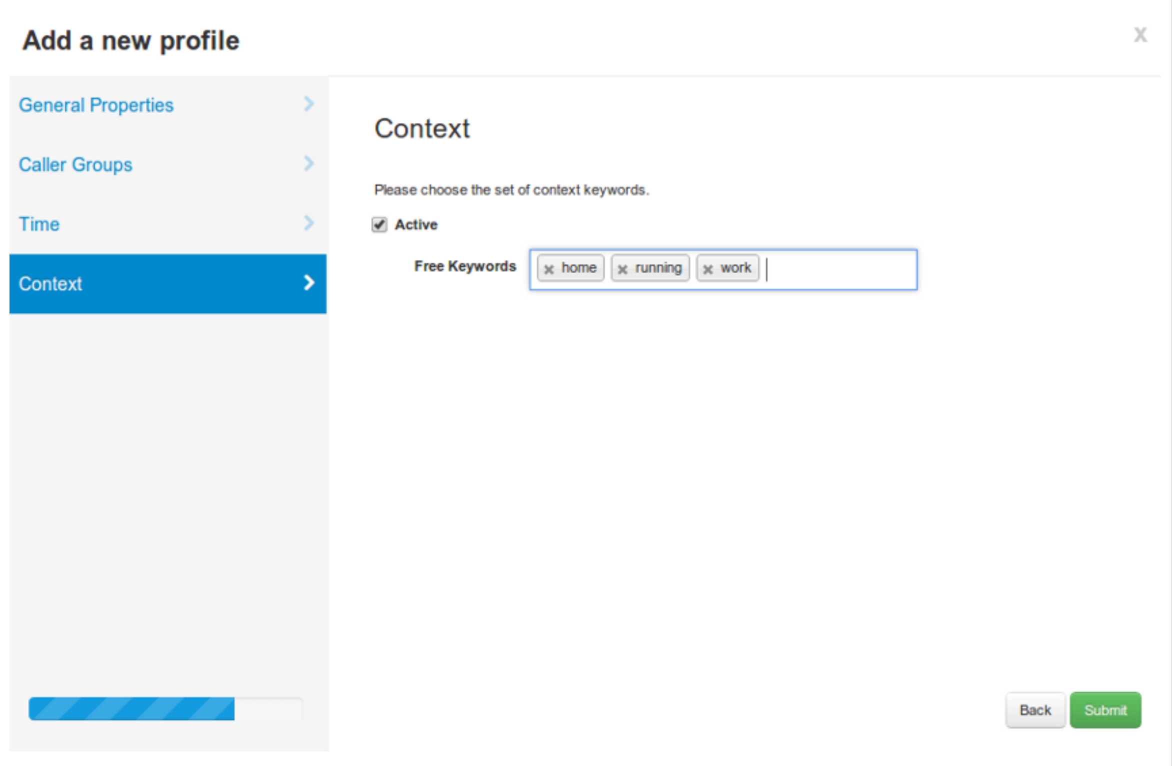
☑ Active

	Sun	Mon	Tue	Wed	Thu	Fri	Sat
All day							
00:00	00:00 - Late	00:00 - Late	00:00 - Late	00:00 - Late	00:00 - Late	00:00 - Late	00:00 - Late
02:00			02:00 - 05:3 New Caller				
04:00							
06:00	06:00 - 10:0 Morning	06:00 - 10:0 Morning	06:00 - 10:0 Morning	06:00 - 10:0 Morning	06:00 - 10:0 Morning	06:00 - 10:0 Morning	06:00 - 10:0 Morning
08:00							
10:00	10:00 - 16:2 Daytime	10:00 - 16:3 Daytime	10:00 - 16:2 Daytime	10:00 - 16:3 Daytime	10:00 - 16:3 Daytime	10:00 - 16:3 Daytime	10:00 - 16:3 Daytime
12:00							
14:00							
16:00	16:30 - 19:2 Early Fringe	16:30 - 19:3 Early Fringe	16:30 - 19:3 Early Fringe	16:30 - 19:3 Early Fringe	16:30 - 19:3 Early Fringe	16:30 - 19:3 Early Fringe	16:30 - 19:3 Early Fringe
18:00	19:30 - 00:0 Prime Time	19:30 - 00:0 Prime Time	19:30 - 00:0 Prime Time	19:30 - 00:0 Prime Time	19:30 - 00:0 Prime Time	19:30 - 00:0 Prime Time	19:30 - 23:5 Prime Time
20:00							
22:00							

Profiles: profile editor portlet.

Context

In the time step you can activate the profiling by context keywords. Once the profiling is active, you can define new keywords or select already used ones.



Add a new profile

Context

Please choose the set of context keywords.

☑ Active

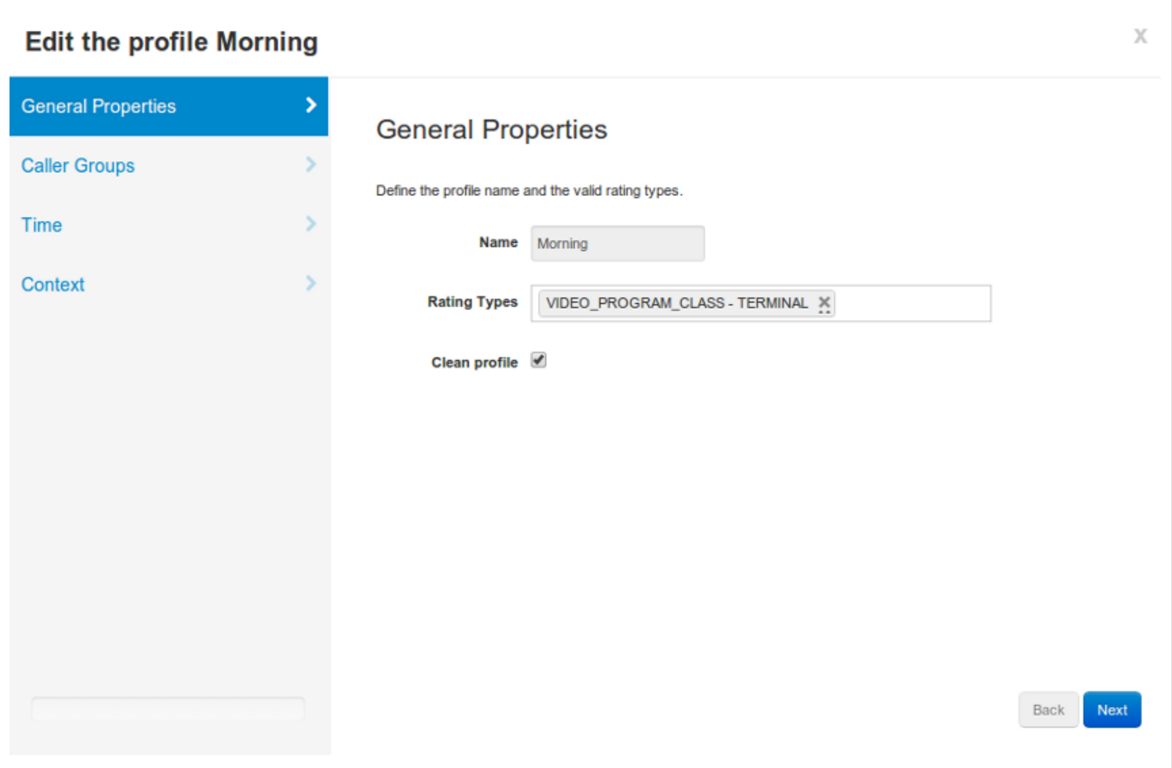
Free Keywords: home, running, work

Profiles: profile editor portlet.

Configure profile

Once a profile has been created, it can be configured by the same wizard. The only difference is that you can choose to clean the profile ratings in the general properties step.

> **Warning**
> If the clean option is checked, the current profile is cloned into a new one.
> The existent profile will switch to OBSOLETE status, while NEW status is assigned to the newly created profile.



Edit the profile Morning

General Properties

Define the profile name and the valid rating types.

Name: Morning

Rating Types: VIDEO_PROGRAM_CLASS - TERMINAL

Clean profile ☑

Profiles: profile editor portlet.

Business Rules

This section describes functionalities provided by the *Business Rules* section of the ContentWise Portal.

The table below lists the pages of the section.

Page	Description
Rules	It is the landing page of the section. It provides an overview of the business rules configured in the system and allows to manage their configurations.
Rule Groups	It provides an overview of the rule groups configured in the system and allows to manage them.
Editorial Lists	It provides an overview of the editorial lists configured in the system and allows to manage their configurations.

Rules

This page provides an overview of the defined business rules and provides functionalities to create and configure business rules.

The page contains the following portlets:

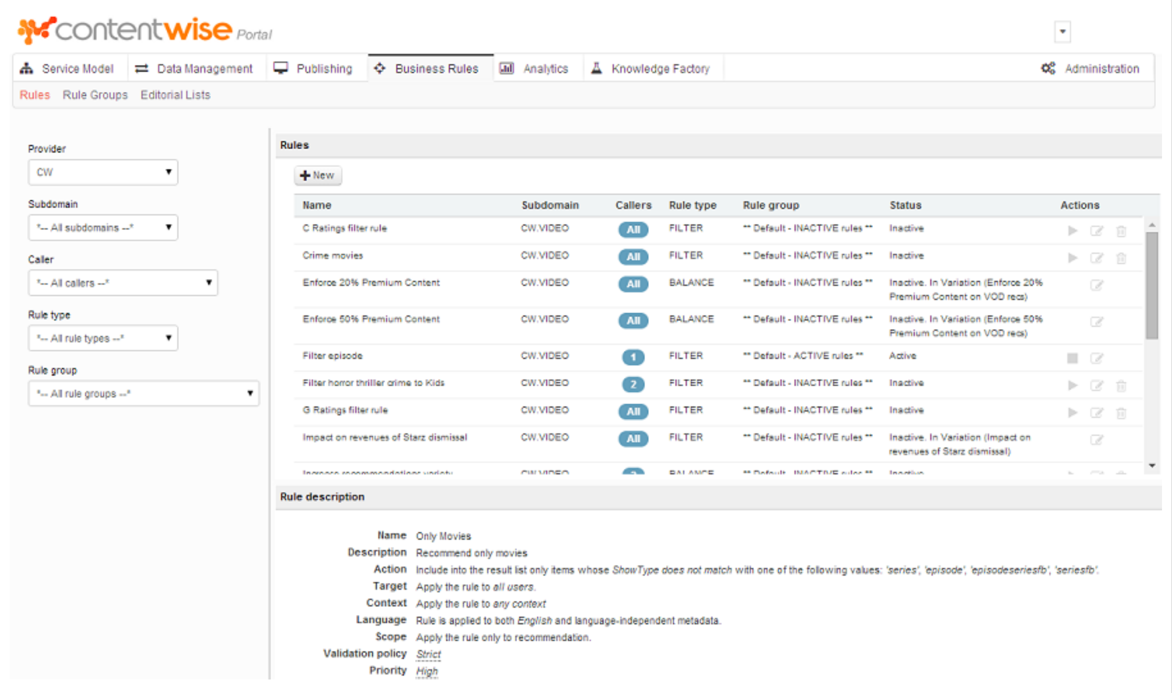
- Rules: lists the business rules defined in the system and provides access to new/edit rule and related configuration.
- Rule description: provides details about the selected rule.

See create a new rule to create a new business rule.

If a provider filter is not selected, you will be prompted to select one.

You can filter results by:

- Subdomain: shows the business rules bound to a specific subdomain.
- Caller: shows the business rules bound to a specific caller.
- Caller group: shows the business rules bound to a specific caller group.
- Rule type: shows business rules of a specific type.
- Rule group: shows the business rules associated to a specific caller.



Provider: CW

Subdomain: *-- All subdomains --*

Caller: *-- All callers --*

Rule type: *-- All rule types --*

Rule group: *-- All rule groups --*

Rules

Name	Subdomain	Callers	Rule type	Rule group	Status	Actions
C Ratings filter rule	CW.VIDEO	All	FILTER	** Default - INACTIVE rules **	Inactive	
Crime movies	CW.VIDEO	All	FILTER	** Default - INACTIVE rules **	Inactive	
Enforce 20% Premium Content	CW.VIDEO	All	BALANCE	** Default - INACTIVE rules **	Inactive. In Variation (Enforce 20% Premium Content on VOD recs)	
Enforce 50% Premium Content	CW.VIDEO	All	BALANCE	** Default - INACTIVE rules **	Inactive. In Variation (Enforce 50% Premium Content on VOD recs)	
Filter episode	CW.VIDEO	1	FILTER	** Default - ACTIVE rules **	Active	
Filter horror thriller crime to Kids	CW.VIDEO	2	FILTER	** Default - INACTIVE rules **	Inactive	
G Ratings filter rule	CW.VIDEO	All	FILTER	** Default - INACTIVE rules **	Inactive	
Impact on revenues of Starz dismissal	CW.VIDEO	All	FILTER	** Default - INACTIVE rules **	Inactive. In Variation (Impact on revenues of Starz dismissal)	
Increase recommendations variety	CW.VIDEO	[illegible]	BALANCE	** Default - INACTIVE rules **	Inactive	

Rule description

Name	Only Movies
Description	Recommend only movies
Action	Include into the result list only items whose *ShowType does not match* with one of the following values: *'series', 'episode', 'episodeseriesfb', 'seriesfb'.*
Target	Apply the rule to *all users.*
Context	Apply the rule to *any context*
Language	Rule is applied to both *English* and language-independent metadata.
Scope	Apply the rule only to recommendation.
Validation policy	*Strict*
Priority	*High*

Rules: list of business rules and related configurations.

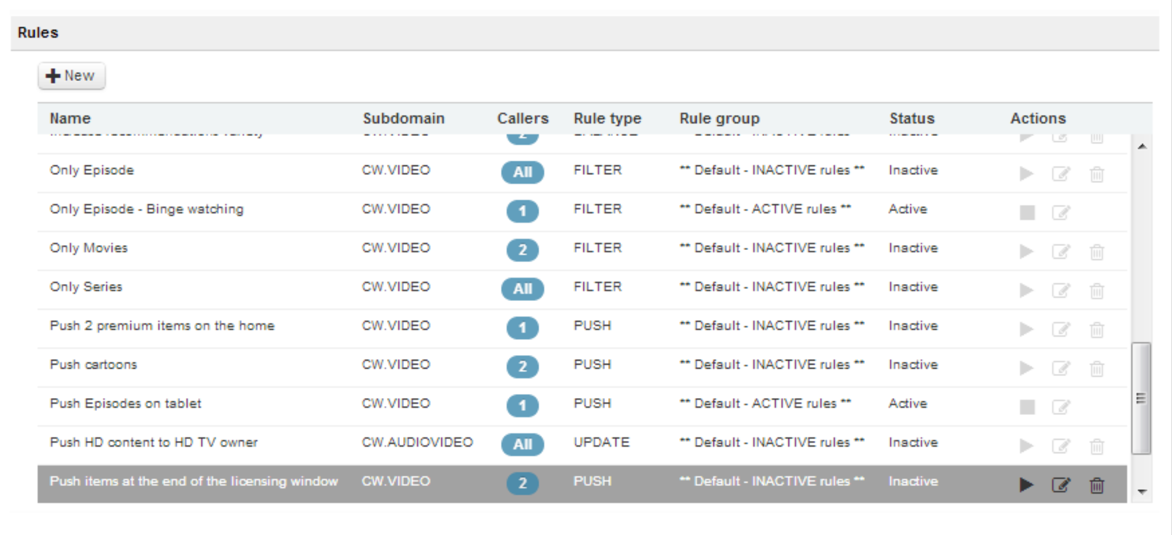
Rules

This portlet provides an overview of the business rules that are configured in the system. It provides access to the new rule page and allows to manage the configuration of existent rules.

For each rule, the table shows:

- *Name*: the rule identifier.
- *Subdomain*: the subdomain for which the rule has been defined.
- *Callers*: the list of callers and caller groups for which the rule has been defined.
  - ALL if the rule is valid for all callers.
  - A plus sign is shown if the rule is bound to one or more caller groups.
- *Rule type*: the type of the rule.
- *Rule group*: the rule group to which the rule has been associated.
- *Status*: the status of the rule. It can be ACTIVE or INACTIVE, according to the status of the group it is associated to. If a rule belongs to an experiment variation, it is listed in the status.
- *Actions*: the possible actions are:
  - *activate* / *stop* : if the rule is associated to a default group, it is possible to manually activate or stop it. Please note that rules bound to experiment variation cannot be manually activated.
  - *edit*: edits the rule configuration.
  - *delete*: removes the rule from the system.

> **Important Note**
> Active rules and rules belonging to an experiment variation cannot be deleted.



Rules

Name	Subdomain	Callers	Rule type	Rule group	Status	Actions
Only Episode	CW.VIDEO	All	FILTER	** Default - INACTIVE rules **	Inactive	▶ ✎ 🗑
Only Episode - Binge watching	CW.VIDEO	1	FILTER	** Default - ACTIVE rules **	Active	■ ✎
Only Movies	CW.VIDEO	2	FILTER	** Default - INACTIVE rules **	Inactive	▶ ✎ 🗑
Only Series	CW.VIDEO	All	FILTER	** Default - INACTIVE rules **	Inactive	▶ ✎ 🗑
Push 2 premium items on the home	CW.VIDEO	1	PUSH	** Default - INACTIVE rules **	Inactive	▶ ✎ 🗑
Push cartoons	CW.VIDEO	2	PUSH	** Default - INACTIVE rules **	Inactive	▶ ✎ 🗑
Push Episodes on tablet	CW.VIDEO	1	PUSH	** Default - ACTIVE rules **	Active	■ ✎
Push HD content to HD TV owner	CW.AUDIOVIDEO	All	UPDATE	** Default - INACTIVE rules **	Inactive	▶ ✎ 🗑
Push items at the end of the licensing window	CW.VIDEO	2	PUSH	** Default - INACTIVE rules **	Inactive	▶ ✎ 🗑

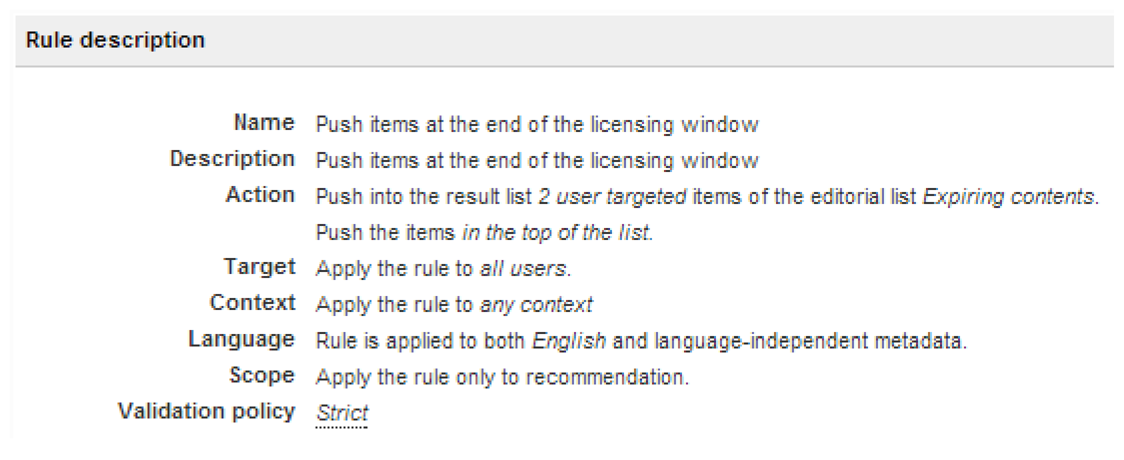
Rules: Rules portlet.

Rule description

This portlet shows the description of the rule selected in the Rules portlet.

The following information is shown:

- *Name*: the name of the rule.
- *Description*: a short description of the rule.
- *Action*: a sentence that describes the action of the rule.
- *Target*: the user condition that must be verified for the application of the rule.
- *Context* : the context for which the rule is valid
- *Language*: the metadata language considered by the rule.
- *Scope*: the scope of the business rule.
- *Validation policy*: the validation policy of the rule.



Rule description

Name	Push items at the end of the licensing window
Description	Push items at the end of the licensing window
Action	Push into the result list 2 *user targeted* items of the editorial list *Expiring contents*. Push the items *in the top of the list*.
Target	Apply the rule to *all users*.
Context	Apply the rule to *any context*
Language	Rule is applied to both *English* and language-independent metadata.
Scope	Apply the rule only to recommendation.
Validation policy	*Strict*

Rules: Rule description portlet.

Rule editor

This section shows how to add a new rule in the system and how to edit an existent rule configuration.

Create a new rule

To create a new rule click the *New* button in the Rules portlet.

It is required to select a provider for which the rule has to be defined.

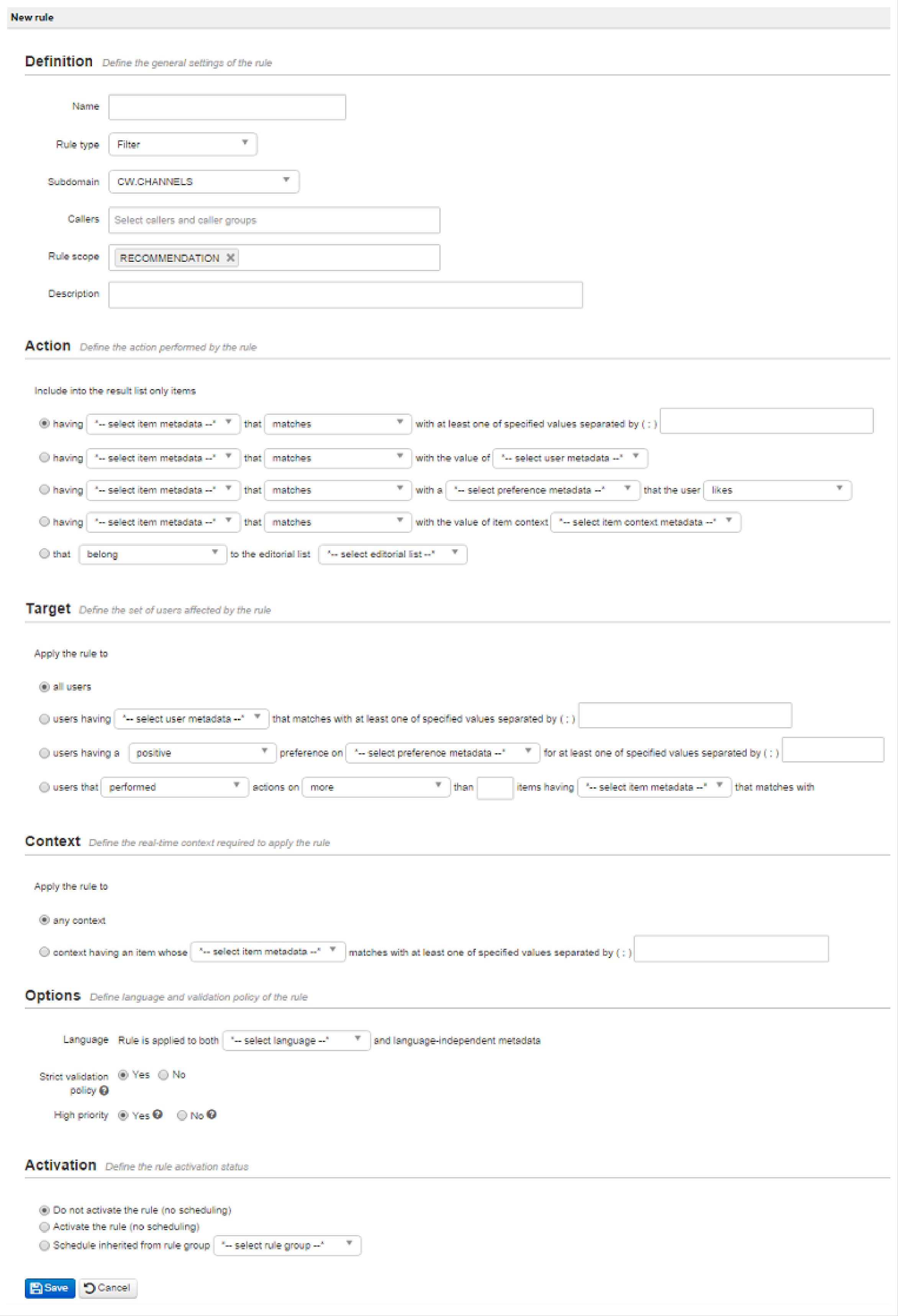
Fill the form to configure the business rule.

The following information is required:

- *Name*: the rule identifier.

> **Warning**
> Valid characters are [A-Z a-z 0-9], -, , ., *space*

- *Rule type*: one of:
  - Filter: to create a rule that filters items before returning a result to the user.
  - Push: to create a rule that promote items.
  - Balance: to create a rule that balance items within a result, according to item characteristics.
  - Update: to smoothly modify recommendation result, by moving up or down items having certain characteristics.
- *Subdomain*: the subdomain affected by the rule.
- *Callers*: the callers for which the rule have to be defined.
- *Rule Scope*: defines the scope of the rule (recommendation / search). According to the type of the rule, search scope may be not available.
- *Description*: an optional description of the rule.
- *Action*: the action performed by the rule. Action conditions may vary according to rule type and rule scope.
- *Target*: the set of users affected by the rule. Target conditions may vary according to rule type and rule scope.
- *Context*: the real-time context required to apply the rule. Context availability may vary according to rule type and rule scope.
- *Options*: configuration options such as language, validation policy and priority.
- *Activation*: the rule activation status.



New rule

Definition *Define the general settings of the rule*

Name:

Rule type: Filter

Subdomain: CW.CHANNELS

Callers: Select callers and caller groups

Rule scope: RECOMMENDATION ✕

Description:

Action *Define the action performed by the rule*

Include into the result list only items

● having '-- select item metadata --' that matches with at least one of specified values separated by (;)

○ having '-- select item metadata --' that matches with the value of '-- select user metadata --'

○ having '-- select item metadata --' that matches with a '-- select preference metadata --' that the user likes

○ having '-- select item metadata --' that matches with the value of item context '-- select item context metadata --'

○ that belong to the editorial list '-- select editorial list --'

Target *Define the set of users affected by the rule*

Apply the rule to

● all users

○ users having '-- select user metadata --' that matches with at least one of specified values separated by (;)

○ users having a positive preference on '-- select preference metadata --' for at least one of specified values separated by (;)

○ users that performed actions on more than items having '-- select item metadata --' that matches with

Context *Define the real-time context required to apply the rule*

Apply the rule to

● any context

○ context having an item whose '-- select item metadata --' matches with at least one of specified values separated by (;)

Options *Define language and validation policy of the rule*

Language: Rule is applied to both '-- select language --' and language-independent metadata

Strict validation policy ?: ● Yes ○ No

High priority: ● Yes ? ○ No ?

Activation *Define the rule activation status*

● Do not activate the rule (no scheduling)
○ Activate the rule (no scheduling)
○ Schedule inherited from rule group '-- select rule group --'

Rules: Rule editor portlet, create a new rule.

Edit a rule

> **Warning**
> Once a business rule has been created, it is not possible to change:
>
> - Rule name
> - Rule type
> - Subdomain

When editing an existent rule you can modify:

- *Callers*: the callers and caller groups for which the rule has to be defined.
- *Scope*: the scope of the rule.
- *Description*: an optional description of the rule.
- *Action*: what the rule does.
- *Target*: the user target of the rule.
- *Context*: the real-time context required to apply the rule.
- *Configuration options* such as language, validation policy and priority.
- *Activation*: the rule activation status.

Rule Groups

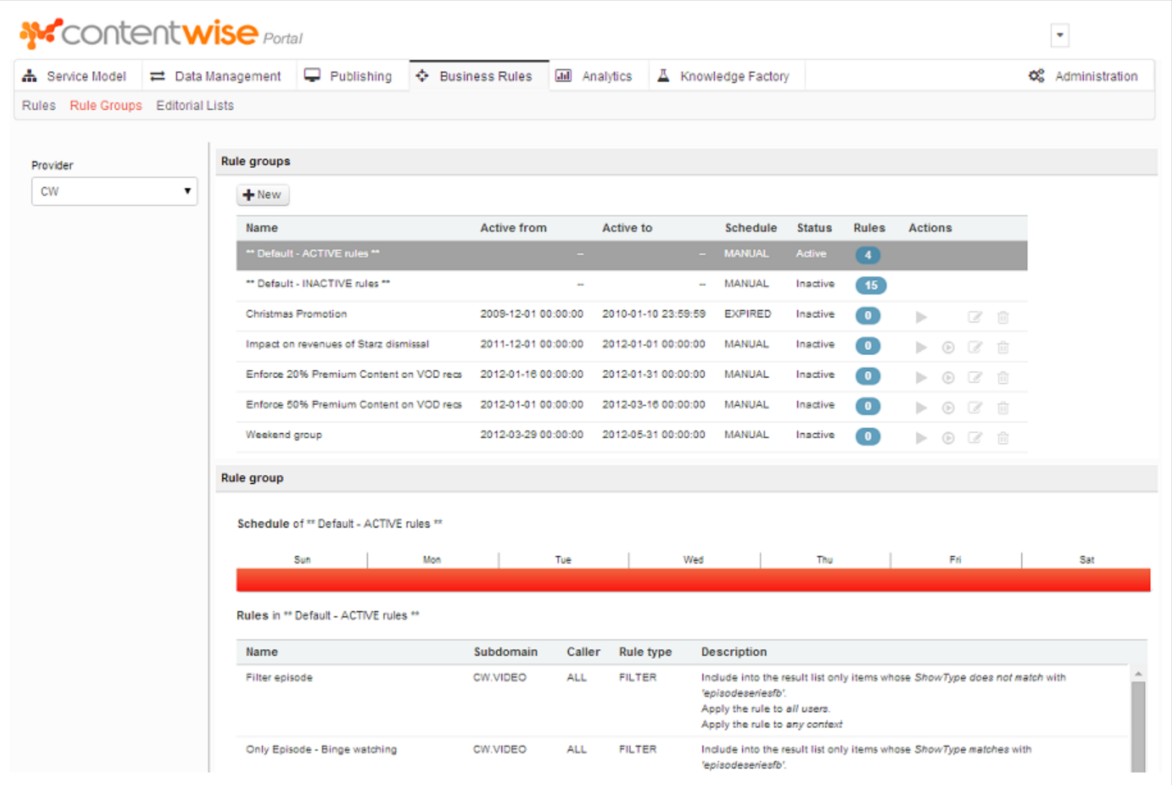
This page provides an overview of the defined rule groups and provides functionalities to schedule and configure rule groups.

The page contains the following portlets:

- Rule groups: lists the rule groups that are in the system and provides access to new/edit rule group and related configuration.
- Rule group: provides details about the selected rule group.

See create a new rule group to create a new rule group.

If a provider filter is not selected, you will be prompted to select one.



Name	Active from	Active to	Schedule	Status	Rules	Actions
** Default - ACTIVE rules **	--	--	MANUAL	Active	4	
** Default - INACTIVE rules **	--	--	MANUAL	Inactive	15	
Christmas Promotion	2009-12-01 00:00:00	2010-01-10 23:59:59	EXPIRED	Inactive	0	
Impact on revenues of Starz dismissal	2011-12-01 00:00:00	2012-01-01 00:00:00	MANUAL	Inactive	0	
Enforce 20% Premium Content on VOD recs	2012-01-16 00:00:00	2012-01-31 00:00:00	MANUAL	Inactive	0	
Enforce 50% Premium Content on VOD recs	2012-01-01 00:00:00	2012-03-16 00:00:00	MANUAL	Inactive	0	
Weekend group	2012-03-29 00:00:00	2012-05-31 00:00:00	MANUAL	Inactive	0	

Schedule of ** Default - ACTIVE rules **

Rules in ** Default - ACTIVE rules **

Name	Subdomain	Caller	Rule type	Description
Filter episode	CW.VIDEO	ALL	FILTER	Include into the result list only items whose *ShowType does not match* with *'episodeseriesfb'*. Apply the rule to *all users*. Apply the rule to *any context*
Only Episode - Binge watching	CW.VIDEO	ALL	FILTER	Include into the result list only items whose *ShowType matches* with *'episodeseriesfb'*.

Rule Groups: list of rule groups and related configurations.

Rule groups

This portlet provides an overview of the rule groups that are configured in the system. It provides access to the new rule group page and allows to manage the scheduling and the configuration of existent groups.

For each rule group, the table shows:

- *Name*: the rule group identifier.
- *Active from*: the start validity date of the group.
- *Active to*: the end validity date of the group.
- *Schedule*: the schedule status of the group. Possible values are:
  - MANUAL: the group activation or deactivation has been manually set.
  - SCHEDULED: the group is currently monitored by the scheduler, that activate and deactivate it according to its configuration.
  - UNSCHEDULED: the group is not scheduled, so it is not monitored by the scheduler.
  - EXPIRED: the end validity date of the group has been reached.
- *Status*: the current status of the group. Possible values are:
  - ACTIVE: the group is active.

    > **Important Note**
    > The business rules associated to an active group are currently active in the system.

  - INACTIVE: the group is not active.

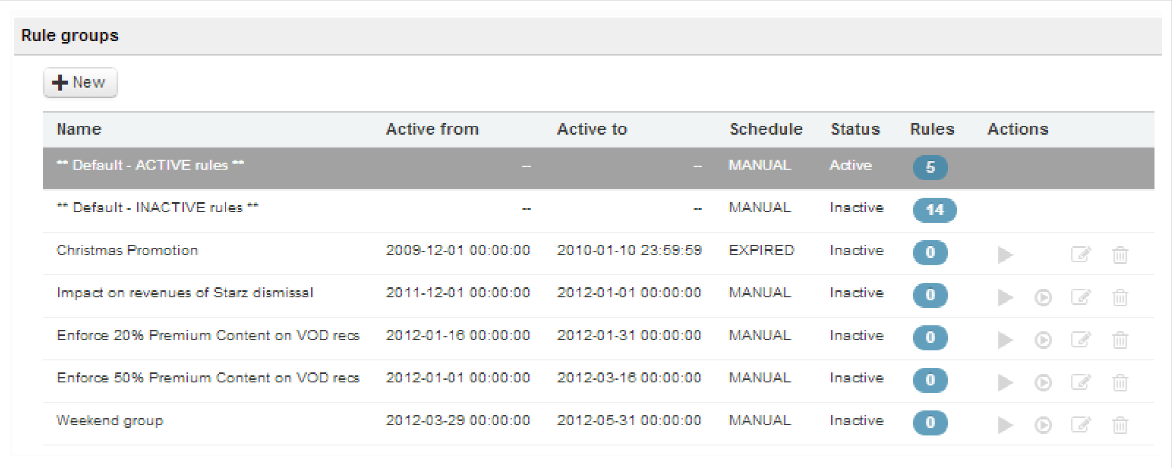
    > **Important Note**
    > The business rules associated to an inactive group are not active in the system.

- *Rules*: the list of rules associated to the group.
- *Actions*: Action availability varies according to the rule group status. Possible actions are:
  - *start*: manually activates a rule group. The schedule configuration of the group is ignored and the group remains active until its status is manually changed.
  - *stop*: manually deactivates a rule group. The group is removed from the list of groups monitored by the scheduler and it remains inactive until its status is manually changed.
  - *schedule*: schedules the rule group. A group that is scheduled is controlled by the scheduler process, that establishes when the group is active or inactive according to its configuration.
  - *unschedule*: unschedules the rule group. The rule group is removed from the list of scheduled groups, avoiding future activations.

    > **Important Note**
    > If the group is currently ACTIVE, it is not forced to INACTIVE but it will stay ACTIVE till its planned end.

  - *edit*: edits the rule group configuration.
  - *delete*: removes the rule group from the system. Active rule groups cannot be deleted.

    > **Warning**
    > If a rule group is removed from the system, all associated business rules are set as inactive.



Rule groups

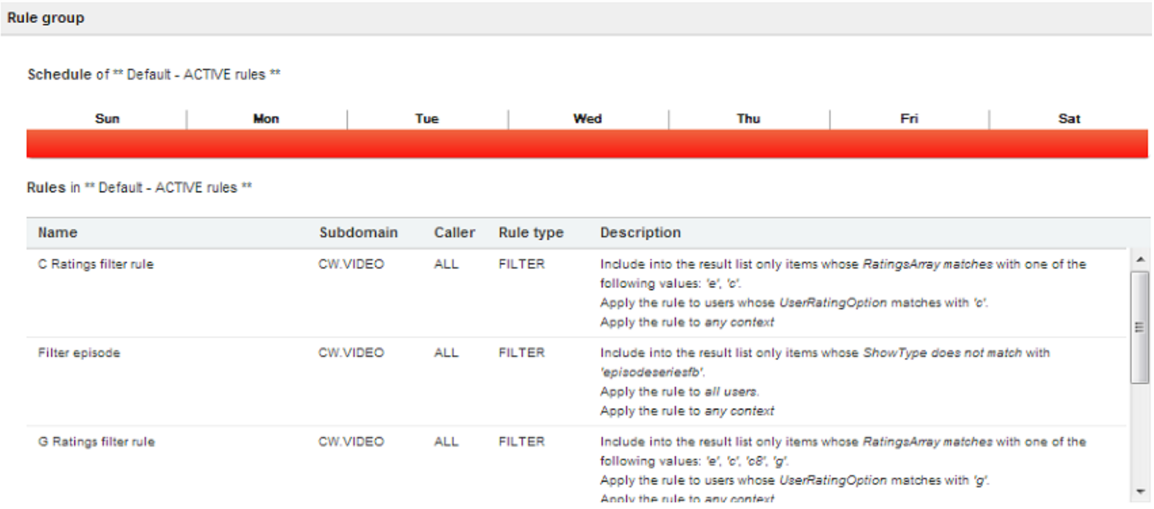
Name	Active from	Active to	Schedule	Status	Rules	Actions
** Default - ACTIVE rules **	--	--	MANUAL	Active	5	
** Default - INACTIVE rules **	--	--	MANUAL	Inactive	14	
Christmas Promotion	2009-12-01 00:00:00	2010-01-10 23:59:59	EXPIRED	Inactive	0	▶ ✎ 🗑
Impact on revenues of Starz dismissal	2011-12-01 00:00:00	2012-01-01 00:00:00	MANUAL	Inactive	0	▶ ⊙ ✎ 🗑
Enforce 20% Premium Content on VOD recs	2012-01-16 00:00:00	2012-01-31 00:00:00	MANUAL	Inactive	0	▶ ⊙ ✎ 🗑
Enforce 50% Premium Content on VOD recs	2012-01-01 00:00:00	2012-03-16 00:00:00	MANUAL	Inactive	0	▶ ⊙ ✎ 🗑
Weekend group	2012-03-29 00:00:00	2012-05-31 00:00:00	MANUAL	Inactive	0	▶ ⊙ ✎ 🗑

Rule Groups: Rule groups portlet.

Rule group

This portlet shows:

- the scheduling configuration of the rule group selected in the Rule groups portlet.
- the list of rules belonging to the group.



Rule group

Schedule of ** Default - ACTIVE rules **

Rules in ** Default - ACTIVE rules **

Name	Subdomain	Caller	Rule type	Description
C Ratings filter rule	CW.VIDEO	ALL	FILTER	Include into the result list only items whose *RatingsArray matches* with one of the following values: 'e', 'c'. Apply the rule to users whose *UserRatingOption* matches with 'c'. Apply the rule to *any context*
Filter episode	CW.VIDEO	ALL	FILTER	Include into the result list only items whose *ShowType does not match* with *'episodeseriesfb'*. Apply the rule to *all users*. Apply the rule to *any context*
G Ratings filter rule	CW.VIDEO	ALL	FILTER	Include into the result list only items whose *RatingsArray matches* with one of the following values: 'e', 'c', 'c8', 'g'. Apply the rule to users whose *UserRatingOption* matches with 'g'. Apply the rule to *any context*

Rule Groups: Rule group description portlet.

Create a new rule group

This section shows how to add a new rule group in the system.

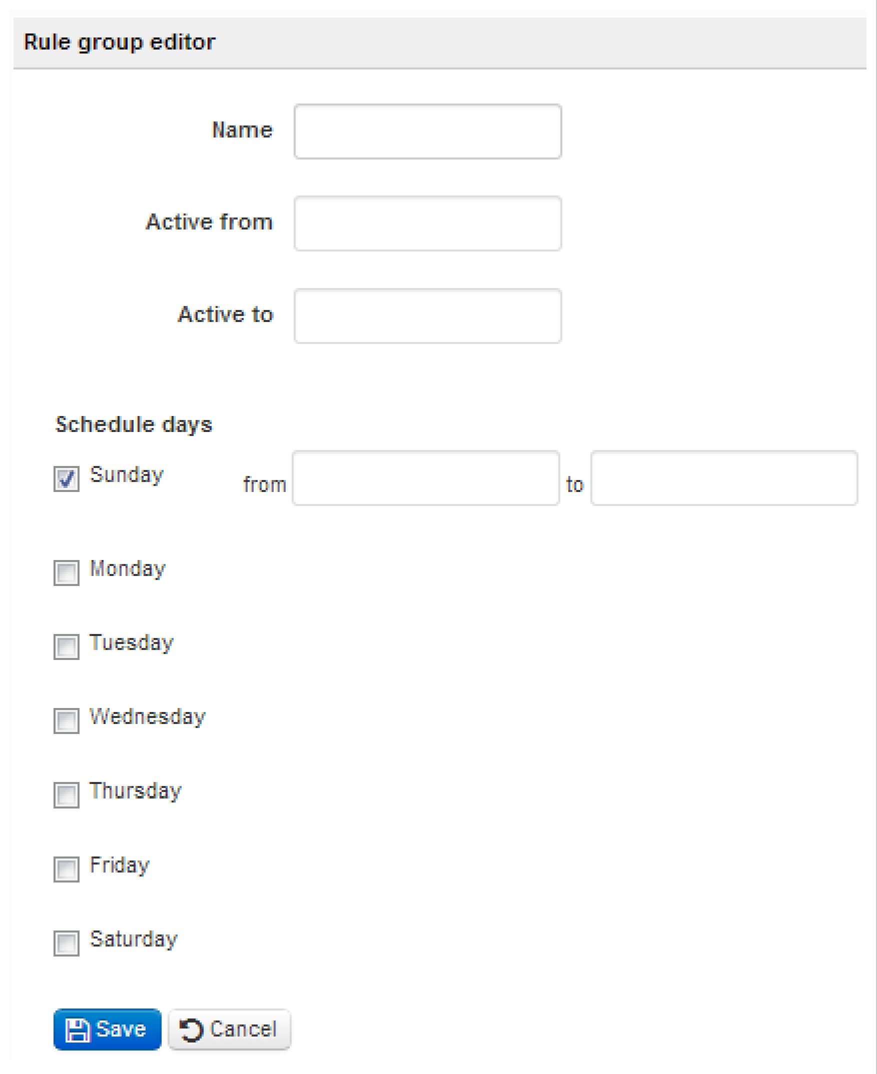
To create a new rule group click the *New* button in the Rule groups portlet.

The following information is required:

- *Name*: the rule group identifier.

> **Warning**
> Valid characters are [A-Z a-z 0-9], -, , ., *space*

- *Active from*: the start validity date, from which the group has to be considered by the system.
- *Active to*: the end validity date.
- *Schedule*: a weekly based schedule configuration. For each day, it is possible to specify a start and an end time.



Rule group editor

Name

Active from

Active to

Schedule days

☑ Sunday from to

□ Monday

□ Tuesday

□ Wednesday

□ Thursday

□ Friday

□ Saturday

Rule Groups: Rule group editor portlet.

Analytics

This section describes functionalities provided by the *Insight* section of the ContentWise Portal.

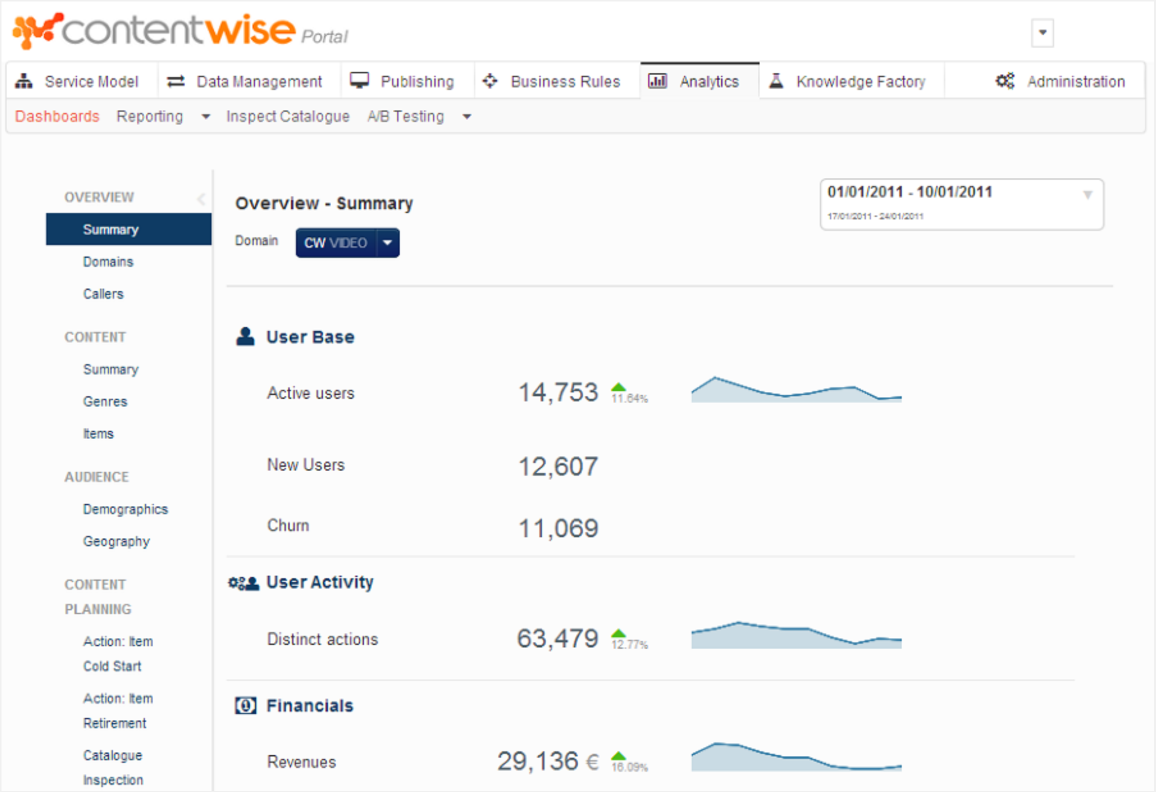
The table below lists the pages of the section.

Page	Description
Dashboards	If available in your installation, this section contains analytics data calculated by ContentWise warehouse component
Reporting	It allows to manage report configuration and to browse report results.
Inspect Catalogue	It allows to search content in the catalogue and to analyze item details.
A/B Testing	It allows to manage a/b testing and experiments

Dashboards

This section contains a set of dashboards showing data retrieved from ContentWise Analytics component.

Important note
Dashboards availability may vary according to the installation and the integration with your system and ContentWise.



Overview - Summary

Domain: CW VIDEO

01/01/2011 - 10/01/2011
17/01/2011 - 24/01/2011

User Base

Active users 14,753 11.64%

New Users 12,607

Churn 11,069

User Activity

Distinct actions 63,479 12.77%

Financials

Revenues 29,136 € 16.09%

An example of dashboard

Reporting

This section describes functionalities provided by *Reporting* menu.

Reporting menu gives access to the following pages:

Page	Description
Browse Reports	It is the landing page of the section. It provides a report viewer that allows to browse reports results previously configured.
Configure Reports	It allows to create new reports and edit report configurations.

Browse Reports

This page allows to browse the report results and to execute configured reports.

The page contains the following portlets:

- Reports: lists the reports configured in the system and allow to run reports.
- Report result: lists the results available for the selected report.
- Report viewer: shows a report result.

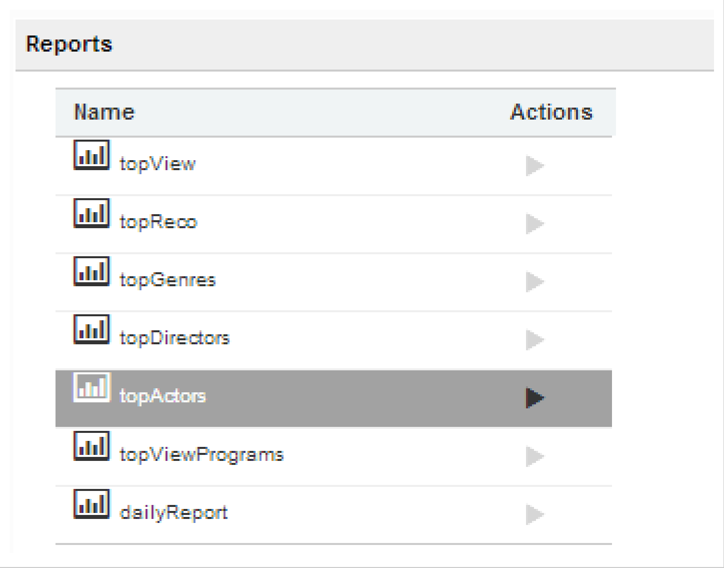
You will be prompted to select a provider.

Reports

This portlet lists the reports that are available in the system.

For each report, the table shows:

- *Name*: the report name.
- *Actions*: you can:
  - *run*: runs the report. The result will be added to report results.



Reports

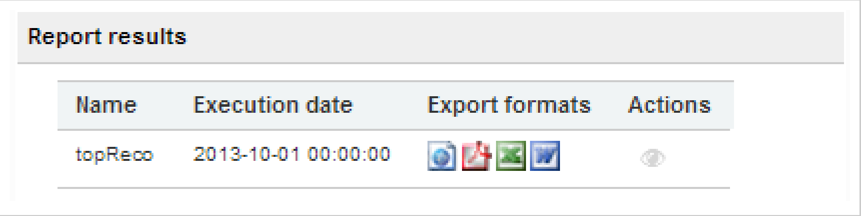
Name	Actions
topView	▶
topReco	▶
topGenres	▶
topDirectors	▶
topActors	▶
topViewPrograms	▶
dailyReport	▶

Browse Reports: Reports portlet.

Report results

This portlet lists the results available for the report selected in the Reports portlet.

It is possible to download the report in one of the available export format.
To view the report in the embedded viewer, select the report or click on the *View* button.



Report results

Name	Execution date	Export formats	Actions
topReco	2013-10-01 00:00:00		

Browse Reports: Reports result portlet.

Report viewer

This portlet allows to browse the report result selected in the Report results portlet.

Configure Reports

This page allows to configure reports.

The page contains the following portlets:

- Reports: lists the reports configured in the system and allow to run reports.
- Report description: lists the results available for the selected report.

See create a new report to create a new report and configure it.

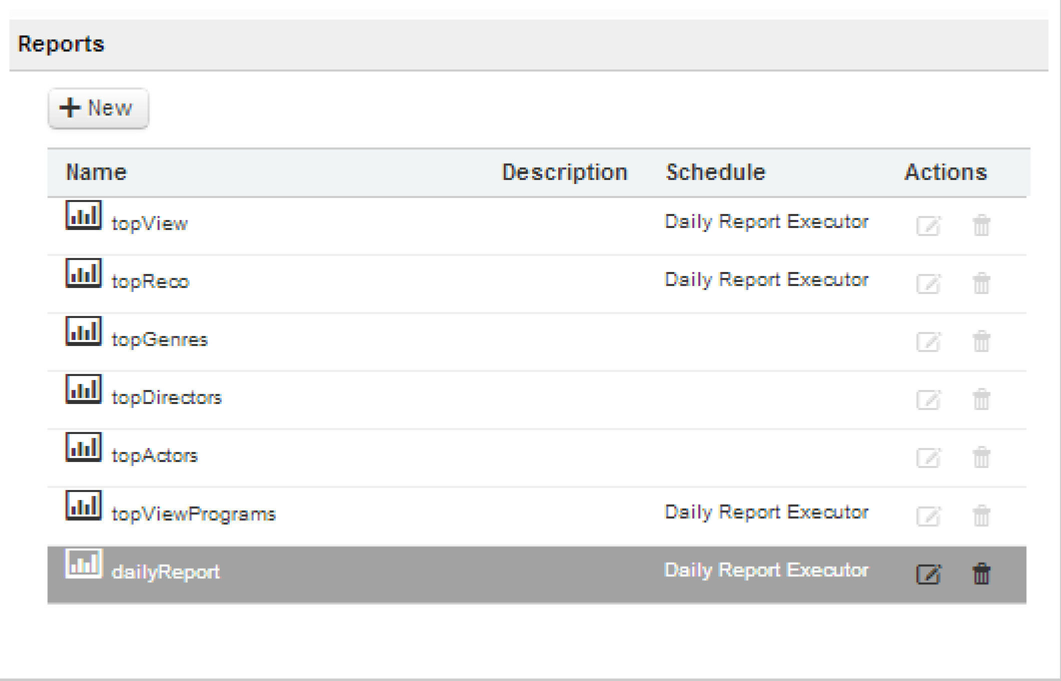
You will be prompted to select a provider.

Reports

This portlet lists the reports that are available in the system.

For each report, the table shows:

- *Name*: the report name.
- *Description*: the report description.
- *Schedule*: the report scheduler associated, if any.
- *Actions*: possible actions are:
    - *edit*: edits the report configuration.
    - *delete*: removes the report from the system.



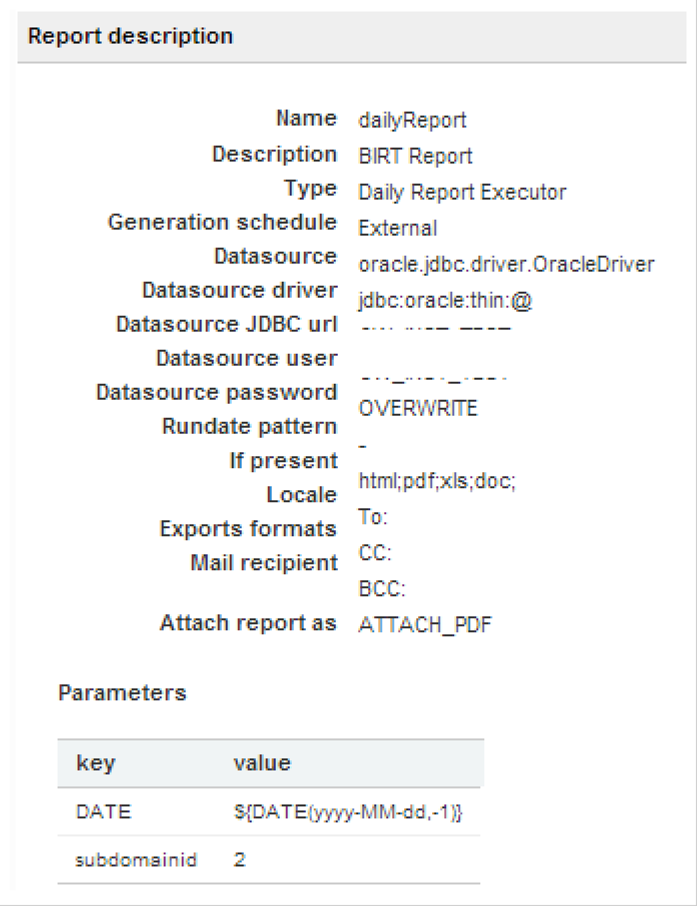
Reports

Name	Description	Schedule	Actions
topView		Daily Report Executor	
topReco		Daily Report Executor	
topGenres			
topDirectors			
topActors			
topViewPrograms		Daily Report Executor	
dailyReport		Daily Report Executor	

Configure Reports: Reports portlet.

Report description

This portlet shows details of the report selected in the Reports portlet.



Report description

Name	dailyReport
Description	BIRT Report
Type	Daily Report Executor
Generation schedule	External
Datasource	oracle.jdbc.driver.OracleDriver
Datasource driver	jdbc:oracle:thin:@
Datasource JDBC url	[illegible]
Datasource user	[illegible]
Datasource password	OVERWRITE
Rundate pattern	-
If present	html;pdf;xls;doc;
Locale	To:
Exports formats	CC:
Mail recipient	BCC:
Attach report as	ATTACH_PDF

Parameters

key	value
DATE	${DATE(yyyy-MM-dd,-1)}
subdomainid	2

Configure Reports: Report description portlet.

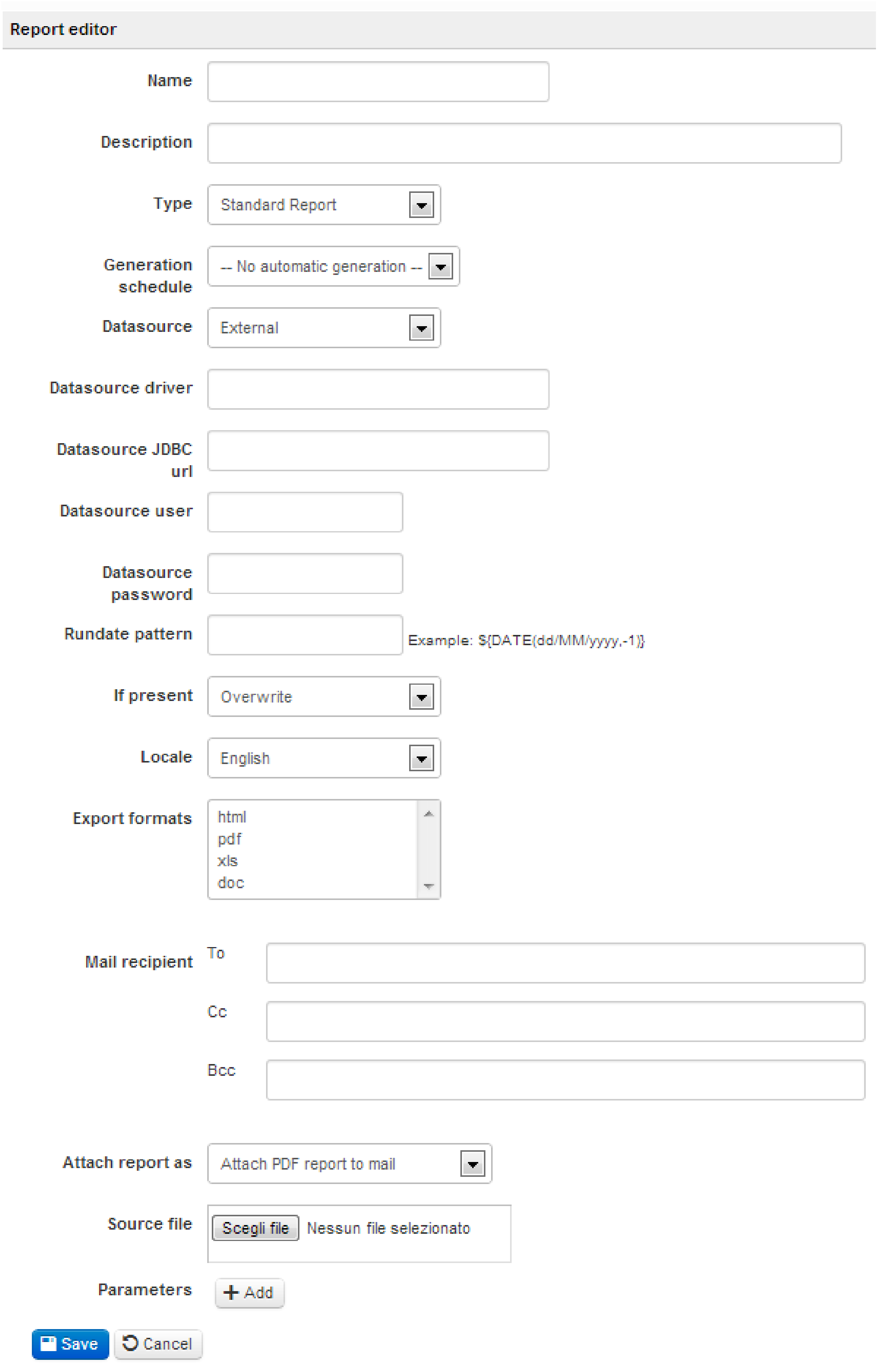
Create a new report

To create a new report click the *New* button in the Reports portlet.

You will be prompted to select a provider.

To configure a report, you are required to provide:

- *Name*: the report identifier.
- *Description*: an optional report description.
- *Type*: the type of the report.
- *Generation schedule*: the scheduler to associate to the report (optional).
- *Datasource*: the datasource associated to the report. Can be:
    - *Internal*
    - *External*: requires the definition of the datasource.
- *Rundate pattern*
- *If present*: the behavior to adopt in case a report result already exist for the same date.
- *Locale*: the locale applied by the report.
- *Export formats*: the formats in which the report has to be generated.
- *Mail recipient*: the list of the receivers of the report.
- *Attach report as*: specifies how the report has to be sent to mail recipient.
- *Source file*: the report template file.
- *Parameters*: optional runtime parameters expressed in terms of pairs <key,value>



Report editor

Name	
Description	
Type	Standard Report
Generation schedule	-- No automatic generation --
Datasource	External
Datasource driver	
Datasource JDBC url	
Datasource user	
Datasource password	
Rundate pattern	Example: ${DATE(dd/MM/yyyy,-1)}
If present	Overwrite
Locale	English
Export formats	html pdf xls doc
Mail recipient	To
	Cc
	Bcc
Attach report as	Attach PDF report to mail
Source file	Scegli file Nessun file selezionato
Parameters	+ Add

Configure Reports: Report editor portlet.

Inspect Catalogue

This page provides search functionalities that allow to inspect the catalogue.

The page contains the following portlets:

- Search query: configure the search parameters.
- Search results: lists the search results.
- Content detail: show metadata of the selected item.
- Content recommendation: show recommendations of items related to the selected.
- Content stems: show stems of the selected item.

Search query

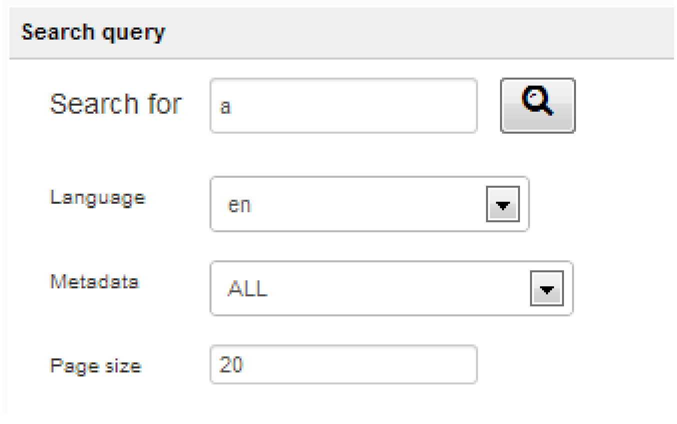
This portlet allows to configure search parameters.

The *search query* parameter is mandatory to perform a search.

The following parameters can be optionally defined:

- *Language*: the language of the items to search.
- *Metadata*: the metadata in which search is done. It is possible to search in ALL metadata.
- *Page size*: the number of items returned per page.
- *Start* and *end* interval: only for program subdomains, specify the search date interval.

Click the *search* button to search items according the configured parameters.



Search query

Search for: a

Language: en

Metadata: ALL

Page size: 20

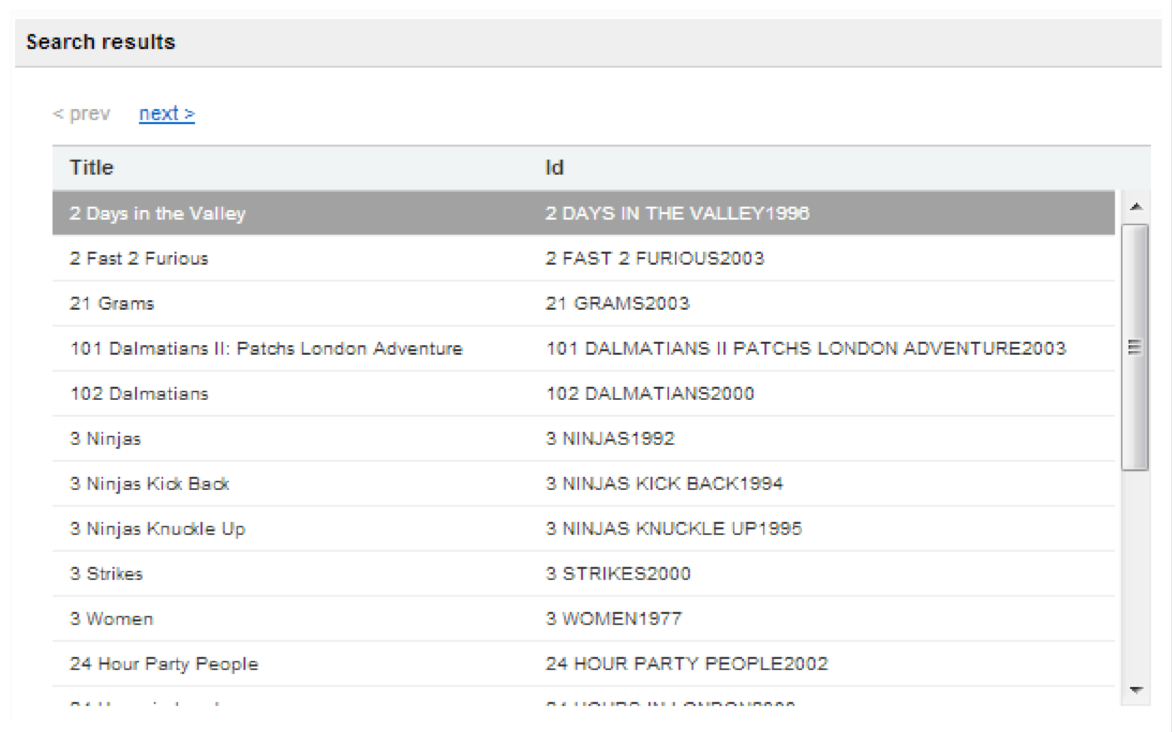
Inspect Catalogue: Search query portlet.

Search results

This portlet lists the search results. Navigate pages by using the next and previous links.

For each item, the table shows:

- *Title*: the item title.
- *Id*: the item identifier.
- *Service*: the service for which the item is identified with the provided Id.



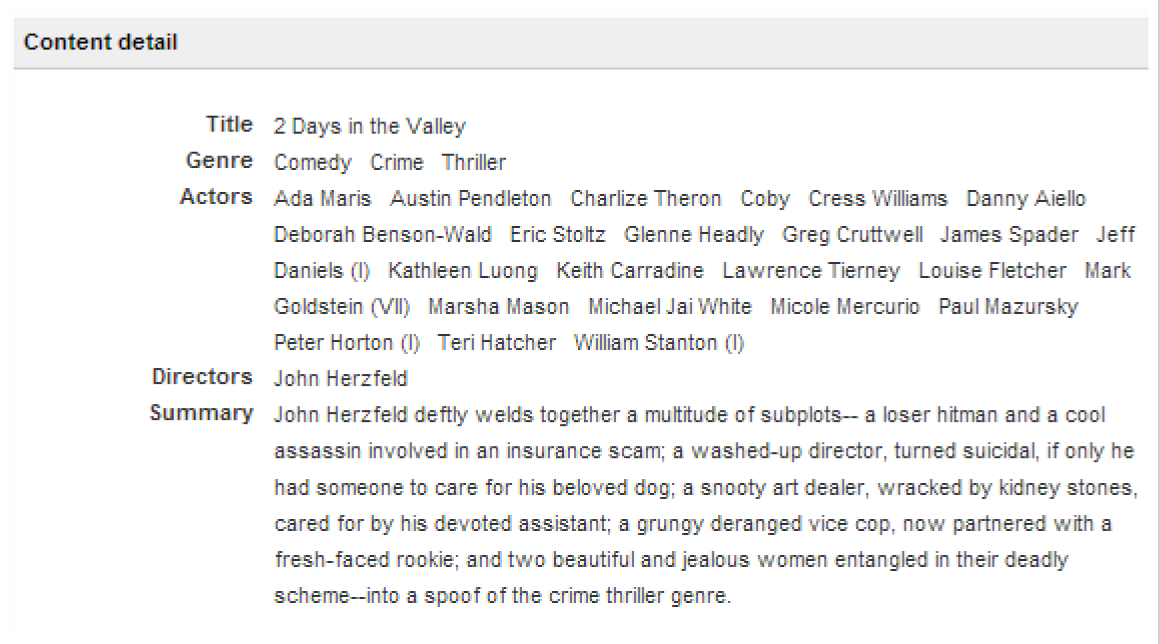
Search results

Title	Id
2 Days in the Valley	2 DAYS IN THE VALLEY1996
2 Fast 2 Furious	2 FAST 2 FURIOUS2003
21 Grams	21 GRAMS2003
101 Dalmatians II: Patchs London Adventure	101 DALMATIANS II PATCHS LONDON ADVENTURE2003
102 Dalmatians	102 DALMATIANS2000
3 Ninjas	3 NINJAS1992
3 Ninjas Kick Back	3 NINJAS KICK BACK1994
3 Ninjas Knuckle Up	3 NINJAS KNUCKLE UP1995
3 Strikes	3 STRIKES2000
3 Women	3 WOMEN1977
24 Hour Party People	24 HOUR PARTY PEOPLE2002

Inspect Catalogue: Search results portlet.

Content detail

This portlet shows the metadata of a the item selected in the Search results portlet.



Content detail

Title 2 Days in the Valley

Genre Comedy Crime Thriller

Actors Ada Maris Austin Pendleton Charlize Theron Coby Cress Williams Danny Aiello Deborah Benson-Wald Eric Stoltz Glenne Headly Greg Cruttwell James Spader Jeff Daniels (I) Kathleen Luong Keith Carradine Lawrence Tierney Louise Fletcher Mark Goldstein (VII) Marsha Mason Michael Jai White Micole Mercurio Paul Mazursky Peter Horton (I) Teri Hatcher William Stanton (I)

Directors John Herzfeld

Summary John Herzfeld deftly welds together a multitude of subplots-- a loser hitman and a cool assassin involved in an insurance scam; a washed-up director, turned suicidal, if only he had someone to care for his beloved dog; a snooty art dealer, wracked by kidney stones, cared for by his devoted assistant; a grungy deranged vice cop, now partnered with a fresh-faced rookie; and two beautiful and jealous women entangled in their deadly scheme--into a spoof of the crime thriller genre.

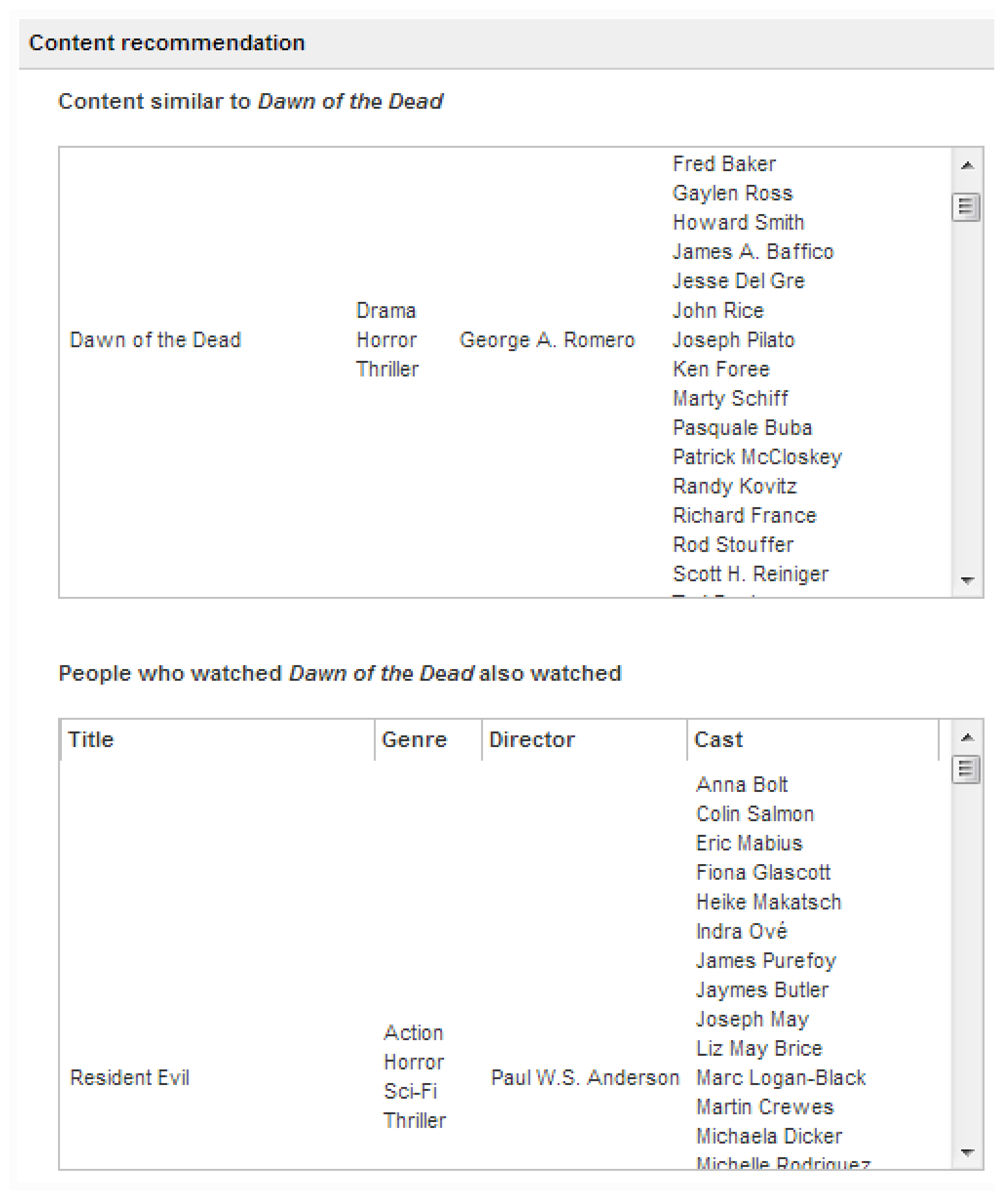
Inspect Catalogue: Search detail portlet.

Content recommendation

This portlet shows two recommendation related to the item selected in the Search results portlet.

It shows:

- Content similar to the selected item.
- Content watched by people who watched the selected item.



Content recommendation

Content similar to *Dawn of the Dead*

Dawn of the Dead	Drama Horror Thriller	George A. Romero	Fred Baker Gaylen Ross Howard Smith James A. Baffico Jesse Del Gre John Rice Joseph Pilato Ken Foree Marty Schiff Pasquale Buba Patrick McCloskey Randy Kovitz Richard France Rod Stouffer Scott H. Reiniger

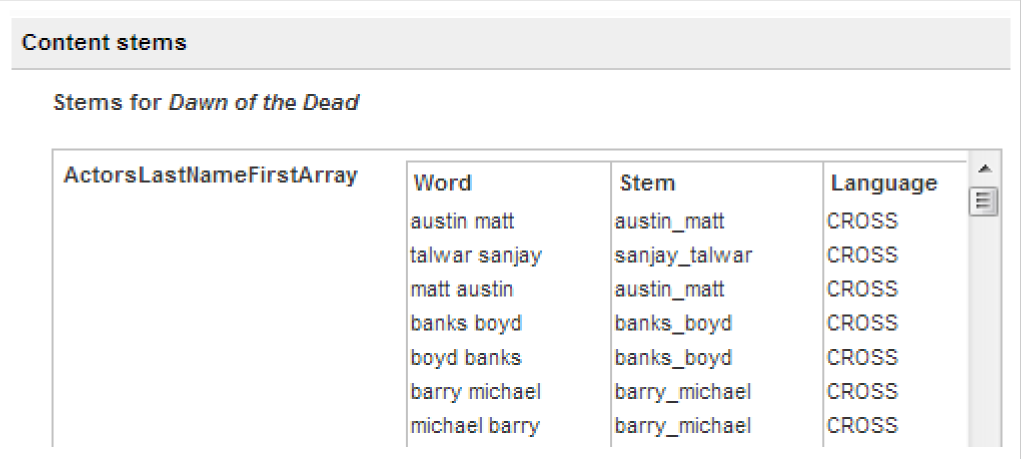
People who watched *Dawn of the Dead* also watched

Title	Genre	Director	Cast
Resident Evil	Action Horror Sci-Fi Thriller	Paul W.S. Anderson	Anna Bolt Colin Salmon Eric Mabius Fiona Glascott Heike Makatsch Indra Ové James Purefoy Jaymes Butler Joseph May Liz May Brice Marc Logan-Black Martin Crewes Michaela Dicker Michelle Rodriguez

Inspect Catalogue: Content recommendation portlet.

Content stems

This portlet lists the stems of the item selected in the Search results portlet.



Content stems

Stems for *Dawn of the Dead*

ActorsLastNameFirstArray

Word	Stem	Language
austin matt	austin_matt	CROSS
talwar sanjay	sanjay_talwar	CROSS
matt austin	austin_matt	CROSS
banks boyd	banks_boyd	CROSS
boyd banks	banks_boyd	CROSS
barry michael	barry_michael	CROSS
michael barry	barry_michael	CROSS

Inspect Catalogue: Content stems portlet.

A B Testing

This section describes functionalities provided by the *A/B Testing* section of the ContentWise Portal.

The table below lists the pages of the section.

Page	Description
Experiments	It allows to configure the testing experiments.
Variations	It allows to define the recommendation settings that can be object of an experiment.

Experiments

This section describes functionalities provided by *Experiments* page. See Testing and experiments.

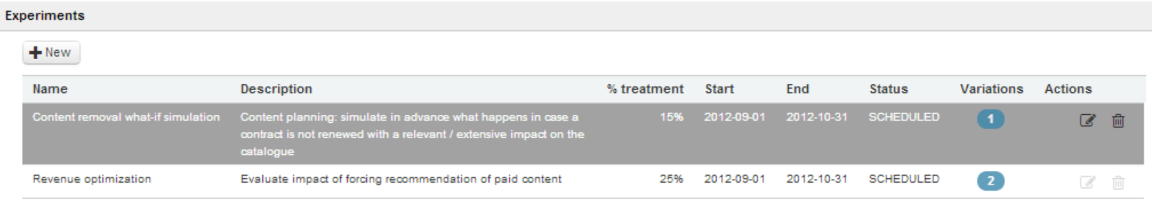
In this page you can:

- have an overview of the experiments configured in the system.
- create new experiments and edit experiment configurations.

Experiments

For each experiment, the table shows:

- *Name*: unique experiment identifier.
- *Description*: a description of the experiment.
- *Treatment percentage*: percentage of users assigned to one of the configured variations (other than the baseline variation). The remaining percentage of users will be assigned to the baseline variation and will form the control group.
  For example, if treatment percentage is set equals to 30% and there are 3 variations (in addition to the baseline one), 30% of users (i.e., the treatment group) will be recommended using one of the 3 variations (10% to each single variation), the remaining 70% of users (i.e., the control group) will be recommended using the baseline variation.
- *Start*: scheduled experiment start time.
- *End*: scheduled experiment end time.
- *Status*: experiment status (RUNNING, SCHEDULED, WAITING, BLOCKED, END).
- *Variations*: variations included in the experiment (in addition to the baseline variation that is not listed).
- *Actions*: the edit action allows to modify the experiment settings (available only in the case of scheduled or blocked experiments).



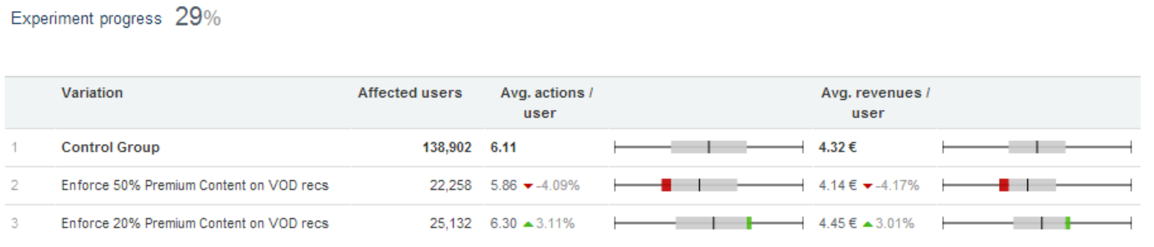
Name	Description	% treatment	Start	End	Status	Variations	Actions
Content removal what-if simulation	Content planning: simulate in advance what happens in case a contract is not renewed with a relevant / extensive impact on the catalogue	15%	2012-09-01	2012-10-31	SCHEDULED	1	
Revenue optimization	Evaluate impact of forcing recommendation of paid content	25%	2012-09-01	2012-10-31	SCHEDULED	2	

List of experiments.

Experiment status and results

If available, you can access experiment status and results by selecting the experiment from the experiments table.

> **Important note**
> Experiment status and results information is available from ContentWise Warehouse. This information is calculated by the warehouse engine with a batch process. This means that experiment result may no be updated instantly.



	Variation	Affected users	Avg. actions / user		Avg. revenues / user	
1	**Control Group**	**138,902**	**6.11**		**4.32 €**	
2	Enforce 50% Premium Content on VOD recs	22,258	5.86 ▼ -4.09%		4.14 € ▼ -4.17%	
3	Enforce 20% Premium Content on VOD recs	25,132	6.30 ▲ 3.11%		4.45 € ▲ 3.01%	

Status of an experiment.

Create a new experiment

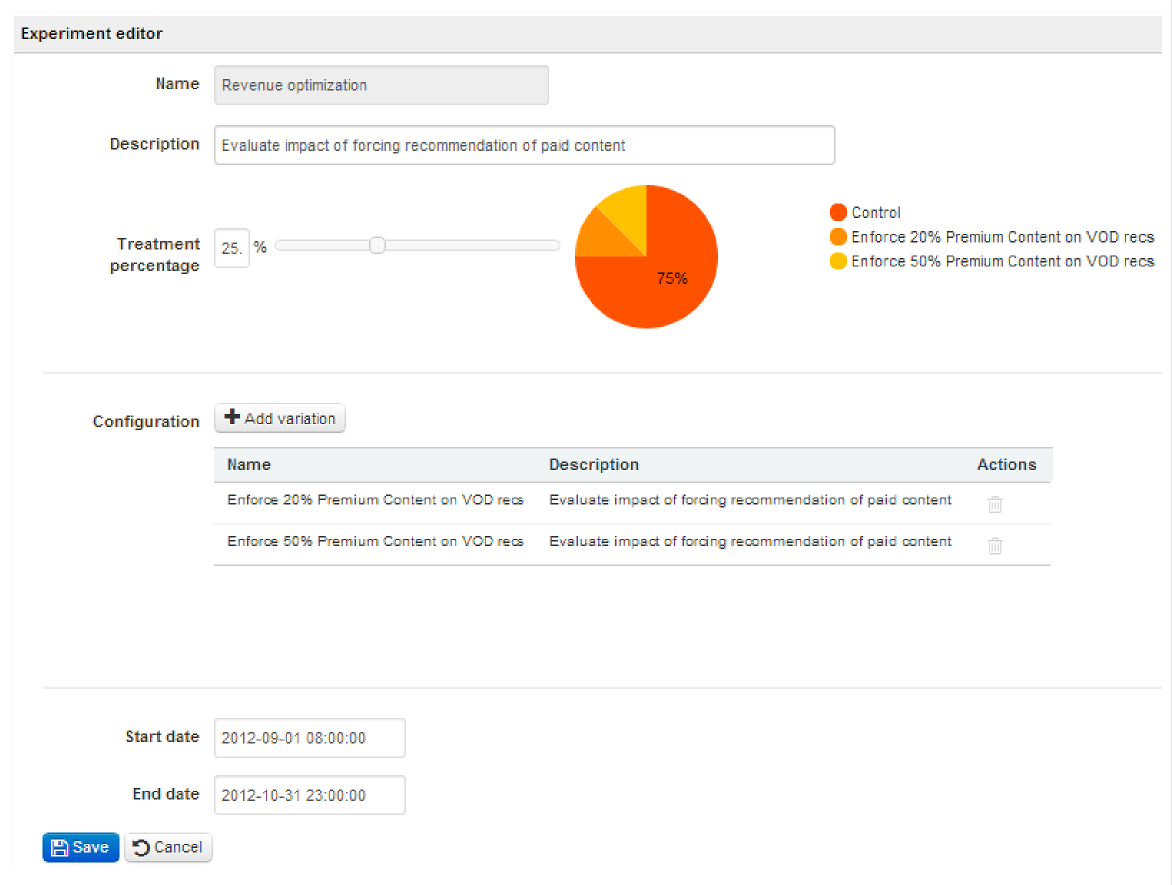
This section shows how to add a new experiment in the system.

To create a new experiment click the *New* button.

All form information but description is mandatory. Below a description of the required data.

- *Name*: a string that univocally identifies the experiment in the system. Valid characters are **[A-Za-z0-9]**, **-**, _, **.**
- *Description*: a description of the experiment
- *Treatment percentage*: the percentage of users assigned to the treatment groups, i.e., recommended using one of the configured variations (other than the baseline one)
- *Configuration*: this section allows to manage the variations to be included in the current experiment. The *Add variation* button allows to add one of the variations previously configured in the Variations page. Use the trash icon to remove from the experiment variations previously added.
- *Start date*: scheduled experiment start time. Users of the treatment group will start receiving recommendation based on one of the configured variations starting from such time (the only exception occurs in the case there are conflicts with other concurrent experiments and experiment status is changed to BLOCKED).
- *End date*: scheduled experiment end time.

To save current experiment configuration click *Save* button.
Click *Cancel* to undo the operation.



Experiment editor

Name	Revenue optimization
Description	Evaluate impact of forcing recommendation of paid content
Treatment percentage	25. %

75%
● Control
● Enforce 20% Premium Content on VOD recs
● Enforce 50% Premium Content on VOD recs

Configuration

Name	Description	Actions
Enforce 20% Premium Content on VOD recs	Evaluate impact of forcing recommendation of paid content	🗑
Enforce 50% Premium Content on VOD recs	Evaluate impact of forcing recommendation of paid content	🗑

Start date	2012-09-01 08:00:00
End date	2012-10-31 23:00:00

Experiments: create a new experiment.

Variations

This section describes functionalities provided by *Variations* page. See Testing and experiments.

In this page you can:

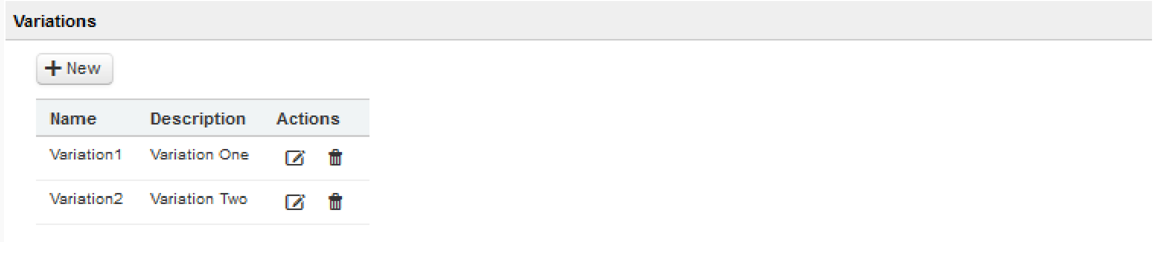
- have an overview of the variations configured in the system.
- create new variations and edit variation configurations.

Variations

For each variation, the table shows:

- *Name*: unique variation identifier.

- *Description*: a description of the variation.
- *Actions*: you can edit and remove an existing variation.



Variations

Name	Description	Actions
Variation1	Variation One	✎ 🗑
Variation2	Variation Two	✎ 🗑

List of variations.

Create a new variation

This section shows how to add a new variation in the system.

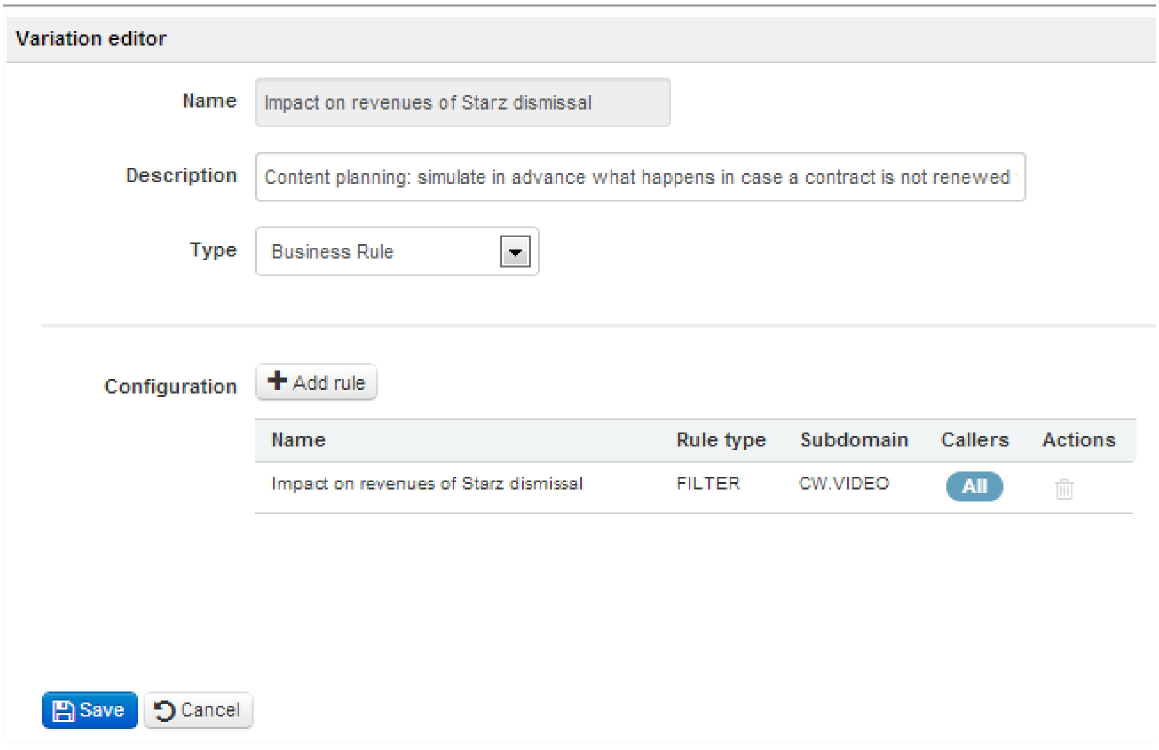
To create a new variation click the *New* button.
Currently, the only available type of variation is *Business Rule*, that allows to configure a variation to test a set of business rules.

Creating a *business-rule* variation

All form information but description is mandatory. Below a description of the required data.

- *Name*: a string that univocally identifies the variation in the system. Valid characters are **[A-Za-z0-9]**, **-**, _, **.**
- *Description*: a description of the variation
- *Type*: the type of variation, in this case *Business Rule*.
- *Configuration*: this section allows to manage the business rules to be included in the current variation. The *Add rule* button allows to add one of the business rules previously configured in the Business Rules page. Use the trash icon to remove from a rule previously added.

To save current variation configuration click *Save* button.
Click *Cancel* to undo the operation.



Variation editor

Name: Impact on revenues of Starz dismissal

Description: Content planning: simulate in advance what happens in case a contract is not renewed

Type: Business Rule

Configuration

Name	Rule type	Subdomain	Callers	Actions
Impact on revenues of Starz dismissal	FILTER	CW.VIDEO	All	🗑

Variations: create a new variation.

Administration

This section describes functionalities provided by the *Administration* section of the ContentWise Portal.

The table below lists the pages of the section.

Page	Description
Data Types	Configures the types of data available in the system.
Rating Types	It provides an overview of the rating types configured in the system and allows to manage their configurations.
Deployment	Configures the deployment of the system components.
Statistics	Configures the statistics calculated and used by the system.
System Settings	Configures system settings.
License	Gives information about ContentWise license of the installation and allows to change it.
WS Accounts	Configures the web services accounts, used by the clients to interact with the system.

Data Types

This section describes functionalities provided by *Data Types* menu.

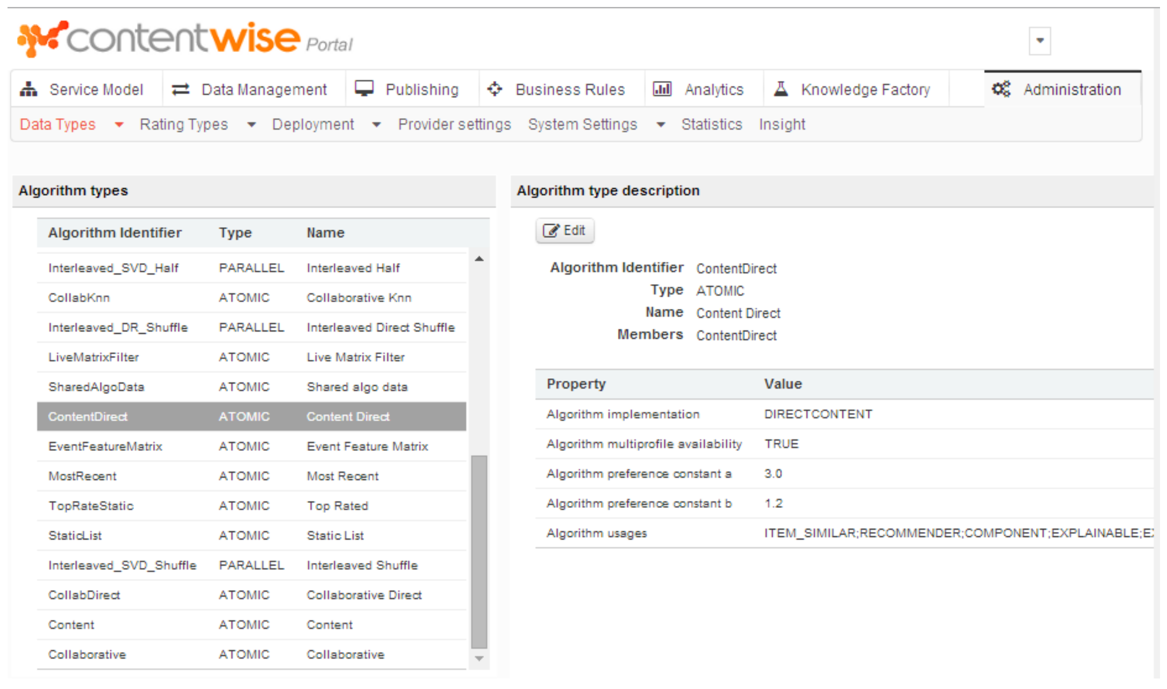
Data Types menu gives access to the following pages:

Page	Description
Algorithm Types	It provides an overview of the algorithm types configured in the system.
Item Types	It provides an overview of the item types configured in the system and allows to manage their configurations.
Metadata Types	It provides an overview of the metadata types configured in the system and allows to manage their configurations.
User Group Types	It provides an overview of the user group types configured in the system and allows to manage their configurations.
User Types	It provides an overview of the user types configured in the system and allows to manage their configurations.

Algorithm Types

This section describes functionalities provided by *Algorithm Types* page.

The page gives an overview of the algorithm types that are configured in the system.



Algorithm types

Algorithm Identifier	Type	Name
Interleaved_SVD_Half	PARALLEL	Interleaved Half
CollabKnn	ATOMIC	Collaborative Knn
Interleaved_DR_Shuffle	PARALLEL	Interleaved Direct Shuffle
LiveMatrixFilter	ATOMIC	Live Matrix Filter
SharedAlgoData	ATOMIC	Shared algo data
ContentDirect	ATOMIC	Content Direct
EventFeatureMatrix	ATOMIC	Event Feature Matrix
MostRecent	ATOMIC	Most Recent
TopRateStatic	ATOMIC	Top Rated
StaticList	ATOMIC	Static List
Interleaved_SVD_Shuffle	PARALLEL	Interleaved Shuffle
CollabDirect	ATOMIC	Collaborative Direct
Content	ATOMIC	Content
Collaborative	ATOMIC	Collaborative

Algorithm type description

Algorithm Identifier ContentDirect
Type ATOMIC
Name Content Direct
Members ContentDirect

Property	Value
Algorithm implementation	DIRECTCONTENT
Algorithm multiprofile availability	TRUE
Algorithm preference constant a	3.0
Algorithm preference constant b	1.2
Algorithm usages	ITEM_SIMILAR;RECOMMENDER;COMPONENT;EXPLAINABLE;E

Algorithm Types: algorithm types configuration.

Algorithm types

The *Algorithm types* portlet lists all algorithm types configured in the system.

For each algorithm type, the table shows:

- *Algorithm identifier*: the algorithm identifier. It is the *algorithmName* to be used in recommendation APIs.
- *Type*: the algorithm type structure.
- *Name*: a friendly name for the algorithm type.

Algorithm type description

The *Algorithm type description* portlet shows details about the algorithm type selected in the Algorithm types portlet

- *Algorithm identifier*: the algorithm identifier. It is the *algorithmName* to be used in recommendation APIs.
- *Type*: the algorithm type structure.
- *Name*: a friendly name for the algorithm type.
- *Members*: the list of algorithms that are part of the algorithm type.
- *Properties*: a set of algorithm type configuration properties.

Item Types

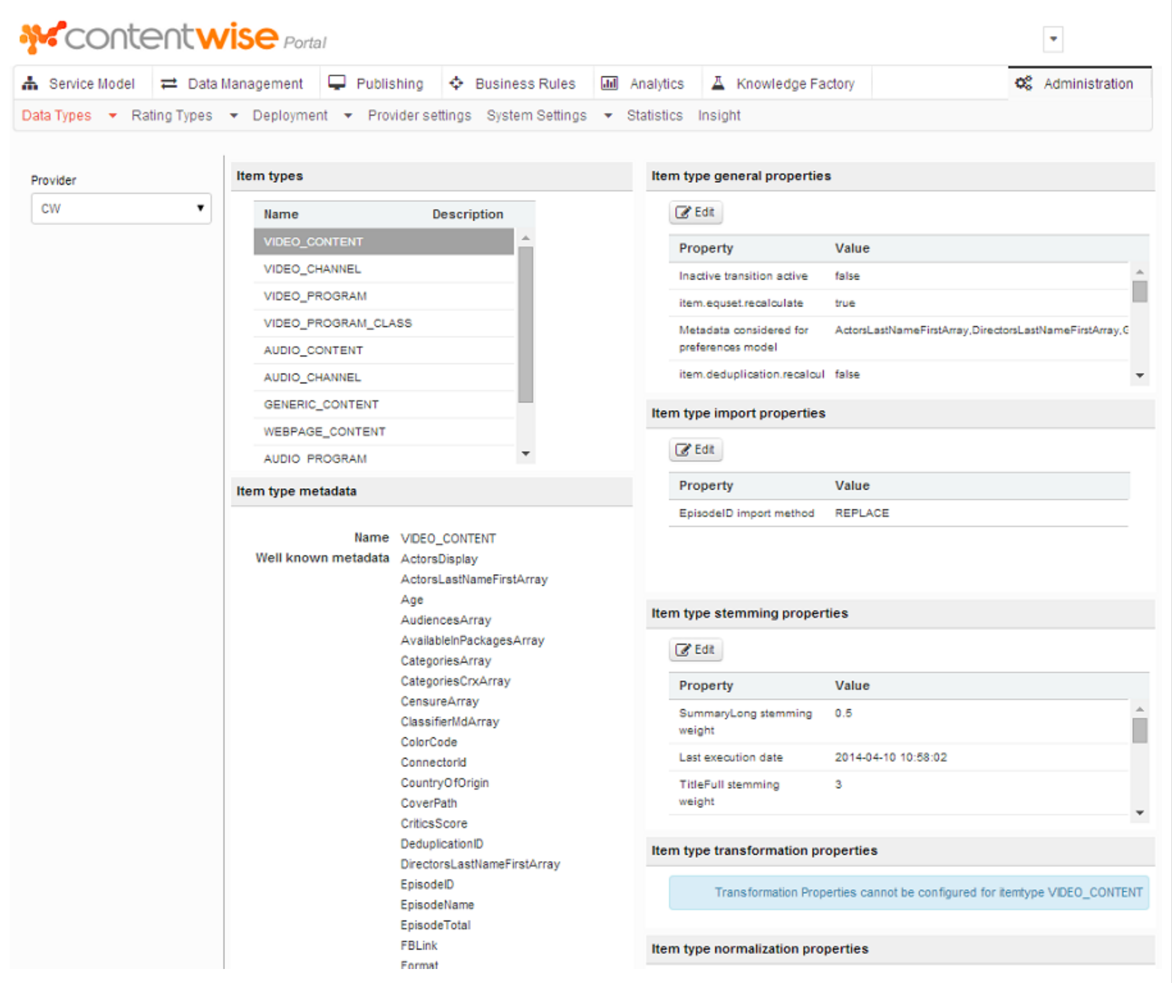
This section describes functionalities provided by *Item Types* page.

The page gives an overview of the item types that are in the system and allows to manage their configuration.

The page contains the following portlets:

- Item types: lists the item types that are in the system.
- Metadata: lists the metadata defined for the item type.
- General properties: configures the global properties of the item type.
- Import properties: configures how data are imported for the item type.
- Stemming properties: configures stemming properties for the item type.
- Transformation properties: configures transformation properties for VIDEO_PROGRAM and AUDIO_PROGRAM item types.
- Normalization properties: configures normalization properties for VIDEO_PROGRAM and AUDIO_PROGRAM item types.

If a provider filter is not selected, you will be prompted to select one.



Provider: CW

Item types

Name	Description
VIDEO_CONTENT	
VIDEO_CHANNEL	
VIDEO_PROGRAM	
VIDEO_PROGRAM_CLASS	
AUDIO_CONTENT	
AUDIO_CHANNEL	
GENERIC_CONTENT	
WEBPAGE_CONTENT	
AUDIO_PROGRAM	

Item type metadata

Name	VIDEO_CONTENT
Well known metadata	ActorsDisplay ActorsLastNameFirstArray Age AudiencesArray AvailableInPackagesArray CategoriesArray CategoriesCrxArray CensureArray ClassifierMdArray ColorCode ConnectorId CountryOfOrigin CoverPath CriticsScore DeduplicationID DirectorsLastNameFirstArray EpisodeID EpisodeName EpisodeTotal FBLink Format

Item type general properties

Property	Value
Inactive transition active	false
item.equset.recalculate	true
Metadata considered for preferences model	ActorsLastNameFirstArray,DirectorsLastNameFirstArray,C
item.deduplication.recalcul	false

Item type import properties

Property	Value
EpisodeID import method	REPLACE

Item type stemming properties

Property	Value
SummaryLong stemming weight	0.5
Last execution date	2014-04-10 10:58:02
TitleFull stemming weight	3

Item type transformation properties

Transformation Properties cannot be configured for itemtype VIDEO_CONTENT

Item type normalization properties

Item Types: item types configuration.

Item types

The *Item types* portlet lists all the item types configured in the system.

For each item type, the table shows:

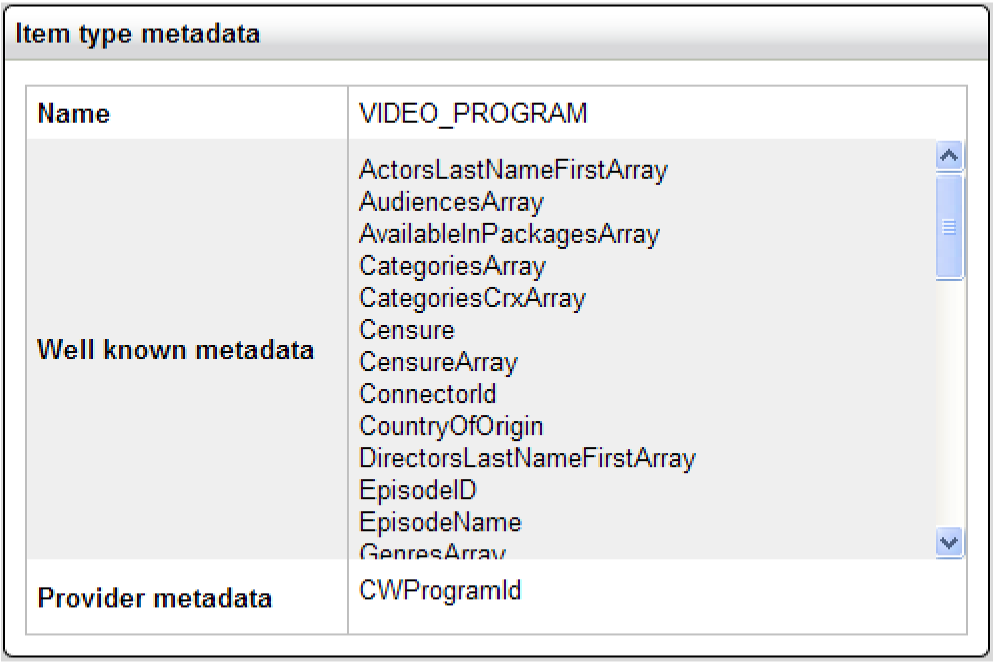
- *Name*: the item type identifier.
- *Description*: a short description of the item type (if any).

Metadata

The *Item type metadata* portlet lists all the metadata configured for the item type selected in Item types portlet.

There are two types of metadata:

- Well known metadata: standard metadata for the item type and custom defined metadata valid for all providers.
- Provider metadata: metadata defined for the selected provider.



Item type metadata

Name	VIDEO_PROGRAM
Well known metadata	ActorsLastNameFirstArray AudiencesArray AvailableInPackagesArray CategoriesArray CategoriesCrxArray Censure CensureArray ConnectorId CountryOfOrigin DirectorsLastNameFirstArray EpisodeID EpisodeName GenresArray
Provider metadata	CWProgramId

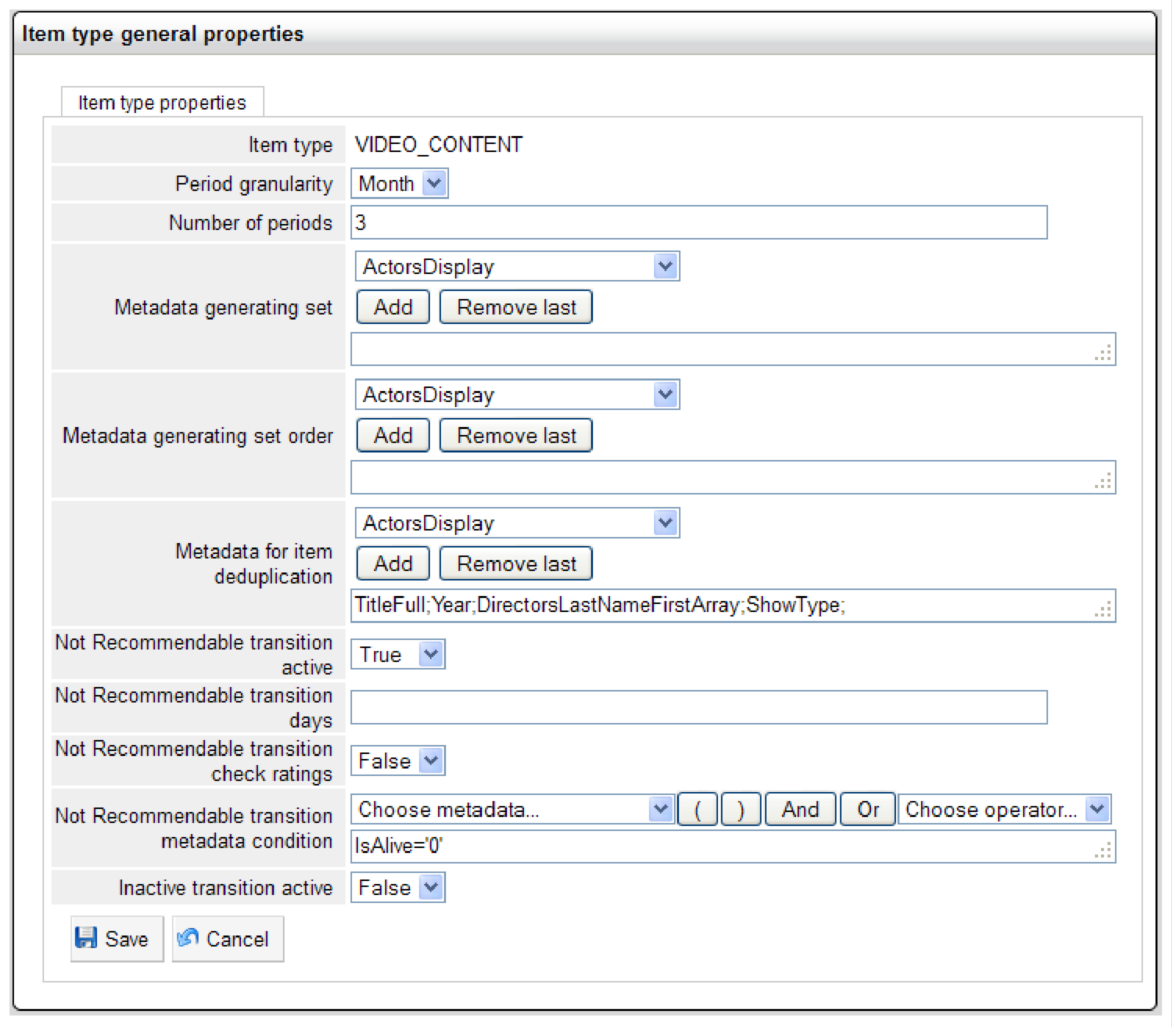
Item Types: Item type metadata portlet.

General properties

This portlet allows to manage the general properties of the item type.

The portlet shows the current configuration. To modify it:

1. Click the *Edit* button.
2. Update the configuration.
3. Click *Save* to submit the changes, *Cancel* to undo the operation.



Item type	VIDEO_CONTENT
Period granularity	Month
Number of periods	3
Metadata generating set	ActorsDisplay
Metadata generating set order	ActorsDisplay
Metadata for item deduplication	ActorsDisplay TitleFull;Year;DirectorsLastNameFirstArray;ShowType;
Not Recommendable transition active	True
Not Recommendable transition days	
Not Recommendable transition check ratings	False
Not Recommendable transition metadata condition	Choose metadata... IsAlive='0'
Inactive transition active	False

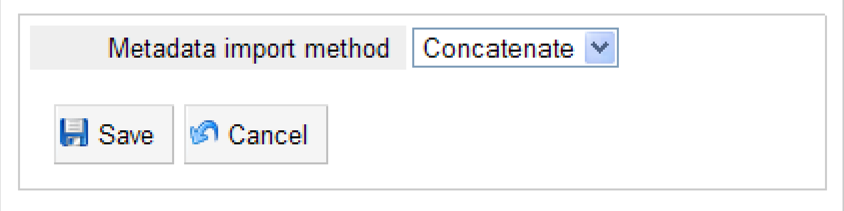
Item Types: Item type general properties portlet (edit mode).

Import properties

This portlet allows to manage the import properties of the item type. It allows to define how metadata are processed during the import step.

The portlet shows the current configuration. To modify it:

1. Click the *Edit* button.
2. Update the configuration. You can:
   - Remove a metadata configuration: click the *remove* button of the metadata to remove.
   - Edit a metadata configuration: click the *edit* button of the metadata and modify its configuration.

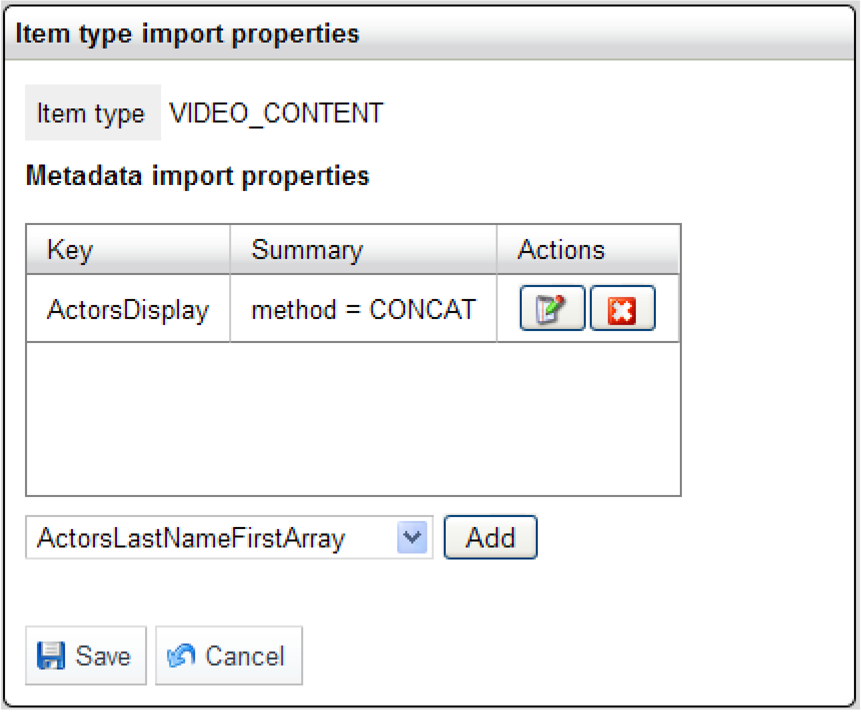


Metadata import method: Concatenate

Item Types: Item type import properties portlet, edit a metadata configuration.

> **Import policy**
> Recall that, by default, *Array* metadata are updated using the CONCAT policy, while all other metadata using the REPLACE policy. See Metadata reference for further details.

   - Add a configuration for a metadata: select the metadata from the list of the availables, click *add* and then edit its configuration.
3. Click *Save* to submit the changes, *Cancel* to undo the operation.



Item type import properties

Item type VIDEO_CONTENT

Metadata import properties

Key	Summary	Actions
ActorsDisplay	method = CONCAT	

Item Types: Item type import properties portlet (edit mode).

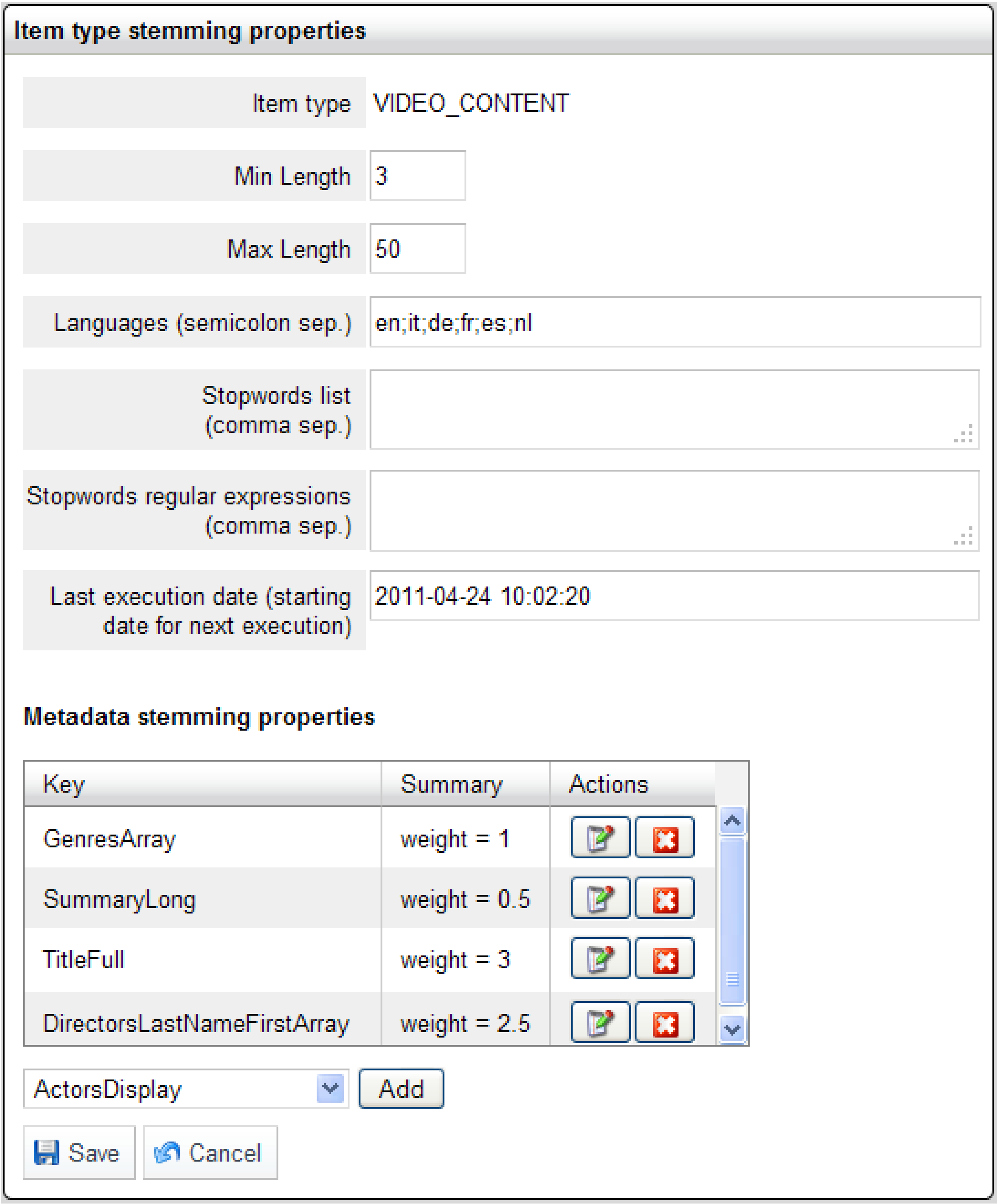
Stemming properties

The *Item type stemming properties* portlet allows the manage the stemming configuration for the item type selected in Item types portlet.

The following information are required:

- *Min Length*: Defines the minimum stem length that a stem must have to be considered.
- *Max Length*: Defines the maximum stem length that a stem must have to be considered.
- *Languages* : Defines the language used to process metadata content.
- *Stopwords list*: Defines a list (; separated) of stopwords that must used to remove matching stems, and their relative words, during content extraction.
- *Stopwords regular expressions* : Defines a list (; separated) of regexp that must be used to remove matching stems, and their relative words, during content extraction.
- *Last execution date* : Reports the last execution date of the item content matrix generation process. It is the starting date considered for the next execution. Can be modified to customize the process execution.
- *Metadata stemming properties*: See Metadata stemming properties

> **Important Note**
> ContentWise provides, for each supported language, a set of stopwords that are applied by default during the content extraction process.



Item type stemming properties

Item type	VIDEO_CONTENT
Min Length	3
Max Length	50
Languages (semicolon sep.)	en;it;de;fr;es;nl
Stopwords list (comma sep.)	
Stopwords regular expressions (comma sep.)	
Last execution date (starting date for next execution)	2011-04-24 10:02:20

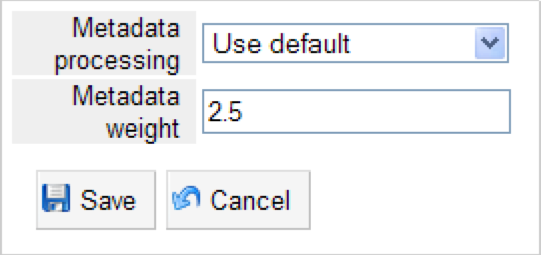
Metadata stemming properties

Key	Summary	Actions
GenresArray	weight = 1	✎ ✖
SummaryLong	weight = 0.5	✎ ✖
TitleFull	weight = 3	✎ ✖
DirectorsLastNameFirstArray	weight = 2.5	✎ ✖

ActorsDisplay

Item Types: Item type stemming properties portlet (edit mode).

Metadata stemming properties



Metadata processing	Use default
Metadata weight	2.5

Item Types: Item type stemming properties portlet, edit a metadata configuration.

For each metadata of the itemtype, you can override default metadata configuration by specifying the following properties:

- **Metadata weight**: Defines the weight of metadata stems used by recommendation system.
- **Metadata processing**: Specifies the processing mode for the current metadata. Possible values are:
    - Use default: Do not override metadata processing method.

- Skip (SKIP): Skip metadata (default behavior if metadata is not specified)
- Separate word (SEPARATE WORD): The list of stems and words generated processing metadata value is stored.

  > **Example**
  >
  > Metadata value : "Pirates of the Caribbean: At World's End"
  > Metadata weight : 3
  > Stems generated (weight) : pirat (3), end (3), caribbean (3), world (3)
  > Words generated : pirates, end, caribbean, world's

- One word (ONE WORD): A single stem and word are stored as a contatenation of the stems and words generated.

  > **Example**
  >
  > Metadata value : "Pirates of the Caribbean: At World's End"
  > Metadata weight : 3
  > Stems generated (weight) : caribbean end pirat world (3)
  > Words generated : caribbean end pirates world's

- Separate and one word (SEPANDONE WORD): Save both the separete list and the concatenation.

  > **Example**
  >
  > Metadata value : "Pirates of the Caribbean: At World's End"
  > Metadata weight : 3
  > Stems generated : pirat (3), end (3), caribbean (3), world (3), caribbean and pirat world (3)
  > Words generated : pirates, end, caribbean, world's, caribbean end pirates world's

- Concatenate weight (CONCATENATE WEIGHT): Same as "One word" but the weight associated to the stem is = "n° of word concatenated-1" + weight.

  > **Example**
  >
  > Metadata value : "Pirates of the Caribbean: At World's End"
  > Metadata weight : 3
  > Stems generated (weight) : caribbean end pirat world (6)
  > Words generated : caribbean end pirates world's

- Separate and concatenate weight (SEPANDCONC WEIGHT): Same as "Sepandone word" but the weight associated to cancatenated stem is = "n° of word concatenated-1" + weight.

  > **Example**
  >
  > Metadata value : "Pirates of the Caribbean: At World's End"
  > Metadata weight : 3
  > Stems generated : pirat (3), end (3), caribbean (3), world (3), caribbean and pirat world (6)
  > Words generated : pirates, end, caribbean, world's, caribbean end pirates world's

- People (PEOPLE): Generate a single stem with first and last name ordered alphabetically and a couple of words first+last name and last+first name (this words are used to extract correctly suggested words on people).

  > **Example**
  >
  > Metadata value : "John Brown"
  > Metadata weight : 3
  > Stems generated : brown john
  > Words generated : john brown, brown john

- As Is (ASIS): Leave metadata as is

  > **Example**
  >
  > Metadata value : "Pirates of the Caribbean: At World's End"
  > Metadata weight : 3
  > Stems generated : pirates of the caribbean: at world's end (3)
  > Words generated : pirates of the caribbean: at world's end

Transformation properties

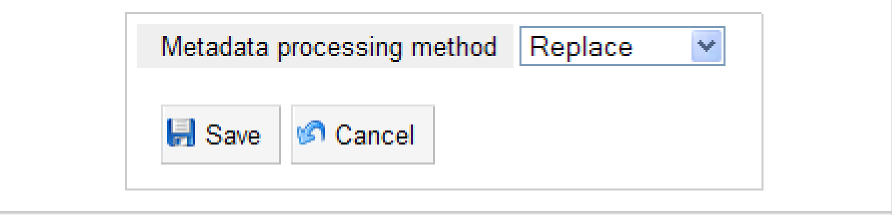
This portlet allows to configure transformation properties.

Transformation properties define how metadata of a program item type are transformed into metadata of the associated program class item type.

Important Note
Transformation properties can be defined only for VIDEO_PROGRAM and AUDIO_PROGRAM itemtype.

The portlet shows the current configuration. To modify it:

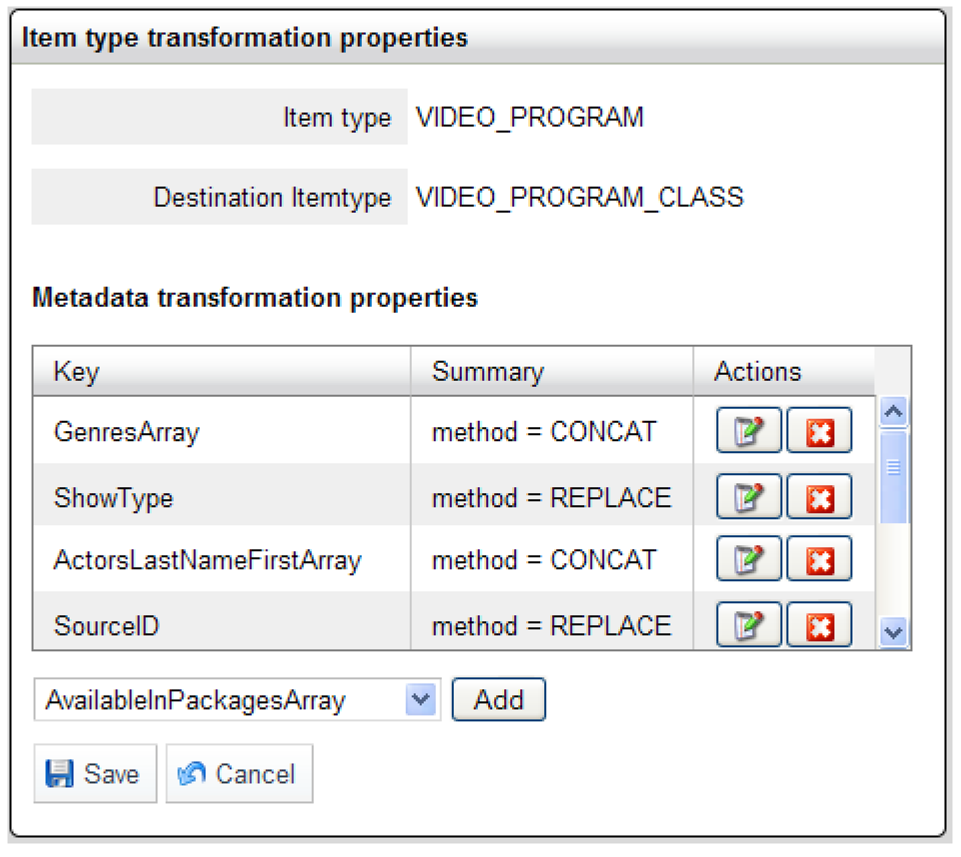
1. Click the *Edit* button.
2. Update the configuration. You can:
   - Remove a metadata configuration: click the *remove* button of the metadata to remove.
   - Edit a metadata configuration: click the *edit* button of the metadata and modify its configuration. You can modify the metadata processing method, that specifies how class item type metadata are generated from program item type metadata. Possible values are:
     - *Replace*: Program metadata value replaces the actual item metadata value.
     - *Concatenate*: Program metadata value is concatenated with actual item metadata value.



Metadata processing method: Replace

Item Types: Item type transformation properties portlet, edit a metadata configuration.

   - Add a configuration for a metadata: select the metadata from the list of the availables, click *add* and then edit its configuration.
3. Click *Save* to submit the changes, *Cancel* to undo the operation.



Item type transformation properties

Item type: VIDEO_PROGRAM

Destination Itemtype: VIDEO_PROGRAM_CLASS

Metadata transformation properties

Key	Summary	Actions
GenresArray	method = CONCAT	
ShowType	method = REPLACE	
ActorsLastNameFirstArray	method = CONCAT	
SourceID	method = REPLACE	

AvailableInPackagesArray

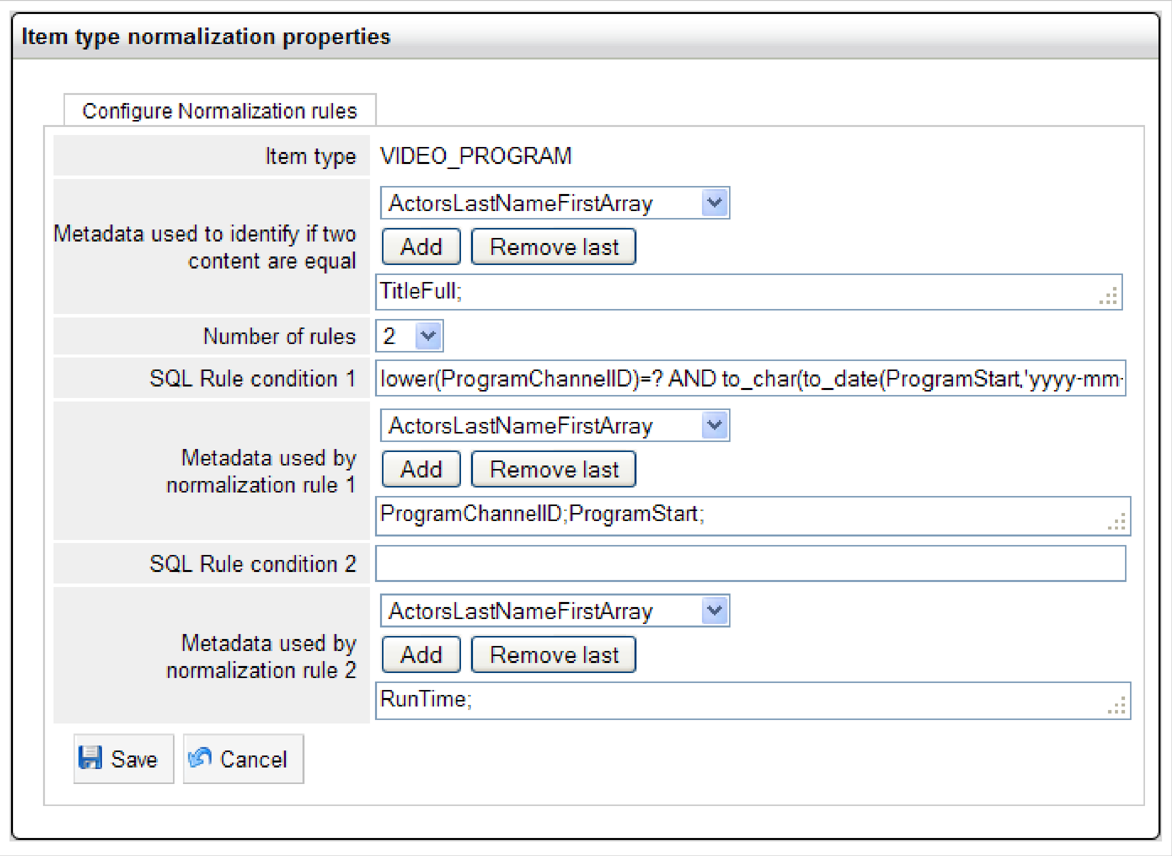
Item Types: Item type transformation properties portlet (edit mode).

Normalization properties

This portlet allows to configure normalization properties.

Important Note
Normalization properties can be defined only for VIDEO_PROGRAM and AUDIO_PROGRAM itemtype.

- *Metadata used to identify if two content are equal*: the list of metadata on which items are compared to determine if they are equals.
- *Number of rules*: the number of defined normalization rules. For each rule you have to specify:
    - *Metadata used by normalization rule X*: Specifies the list of metadata used by normalization rule X.
    - *SQL Rule condition X*: Specifies a SQL condition to be used by the normalization rule X (instead of default condition).



Item type normalization properties

Configure Normalization rules

Item type	VIDEO_PROGRAM
Metadata used to identify if two content are equal	ActorsLastNameFirstArray TitleFull;
Number of rules	2
SQL Rule condition 1	lower(ProgramChannelID)=? AND to_char(to_date(ProgramStart,'yyyy-mm-
Metadata used by normalization rule 1	ActorsLastNameFirstArray ProgramChannelID;ProgramStart;
SQL Rule condition 2	
Metadata used by normalization rule 2	ActorsLastNameFirstArray RunTime;

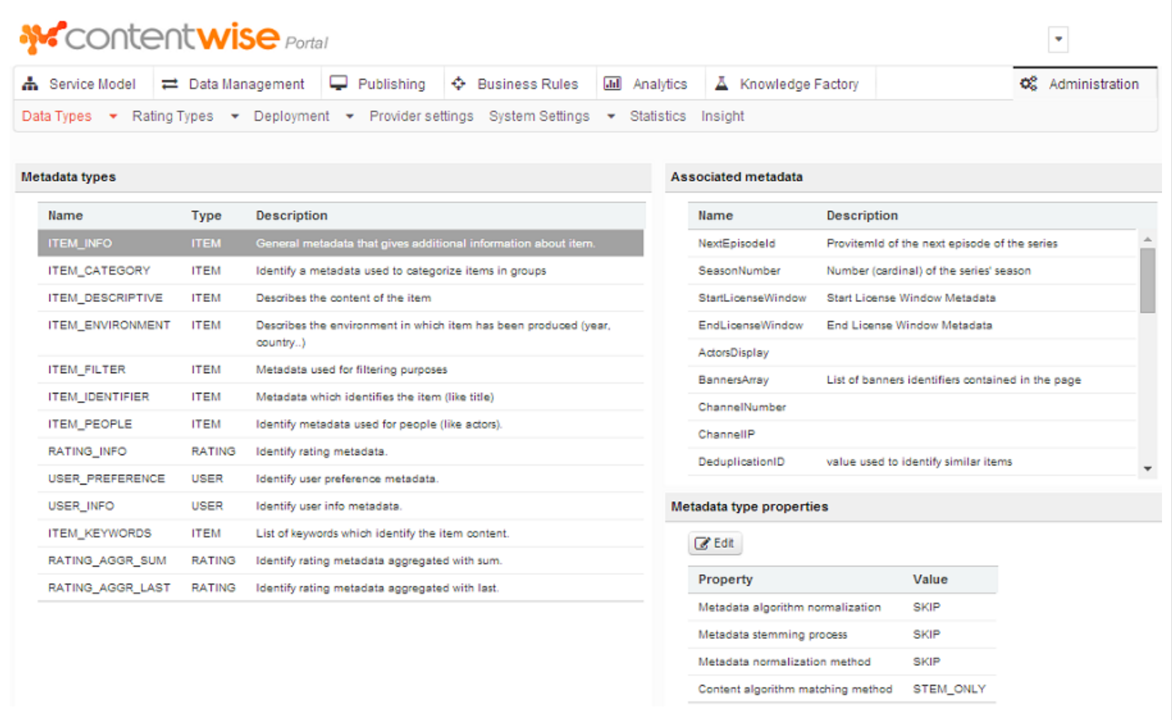
Item Types: Item type normalization properties portlet (edit mode).

Metadata Types

This section describes functionalities provided by *Metadata Types* page.

The page gives an overview of the metadata types that are configured in the system and allows to edit metadata type settings. It contains the following portlets:

- Metadata Types: lists the available metadata types.
- Associated metadata: lists the metadata associated to the selected metadata type.
- Metadata type properties: allows to configure the selected metadata type.



Metadata types

Name	Type	Description
ITEM_INFO	ITEM	General metadata that gives additional information about item.
ITEM_CATEGORY	ITEM	Identify a metadata used to categorize items in groups
ITEM_DESCRIPTIVE	ITEM	Describes the content of the item
ITEM_ENVIRONMENT	ITEM	Describes the environment in which item has been produced (year, country..)
ITEM_FILTER	ITEM	Metadata used for filtering purposes
ITEM_IDENTIFIER	ITEM	Metadata which identifies the item (like title)
ITEM_PEOPLE	ITEM	Identify metadata used for people (like actors).
RATING_INFO	RATING	Identify rating metadata.
USER_PREFERENCE	USER	Identify user preference metadata.
USER_INFO	USER	Identify user info metadata.
ITEM_KEYWORDS	ITEM	List of keywords which identify the item content.
RATING_AGGR_SUM	RATING	Identify rating metadata aggregated with sum.
RATING_AGGR_LAST	RATING	Identify rating metadata aggregated with last.

Associated metadata

Name	Description
NextEpisodeId	ProvitemId of the next episode of the series
SeasonNumber	Number (cardinal) of the series' season
StartLicenseWindow	Start License Window Metadata
EndLicenseWindow	End License Window Metadata
ActorsDisplay	
BannersArray	List of banners identifiers contained in the page
ChannelNumber	
ChannelIP	
DeduplicationID	value used to identify similar items

Metadata type properties

Property	Value
Metadata algorithm normalization	SKIP
Metadata stemming process	SKIP
Metadata normalization method	SKIP
Content algorithm matching method	STEM_ONLY

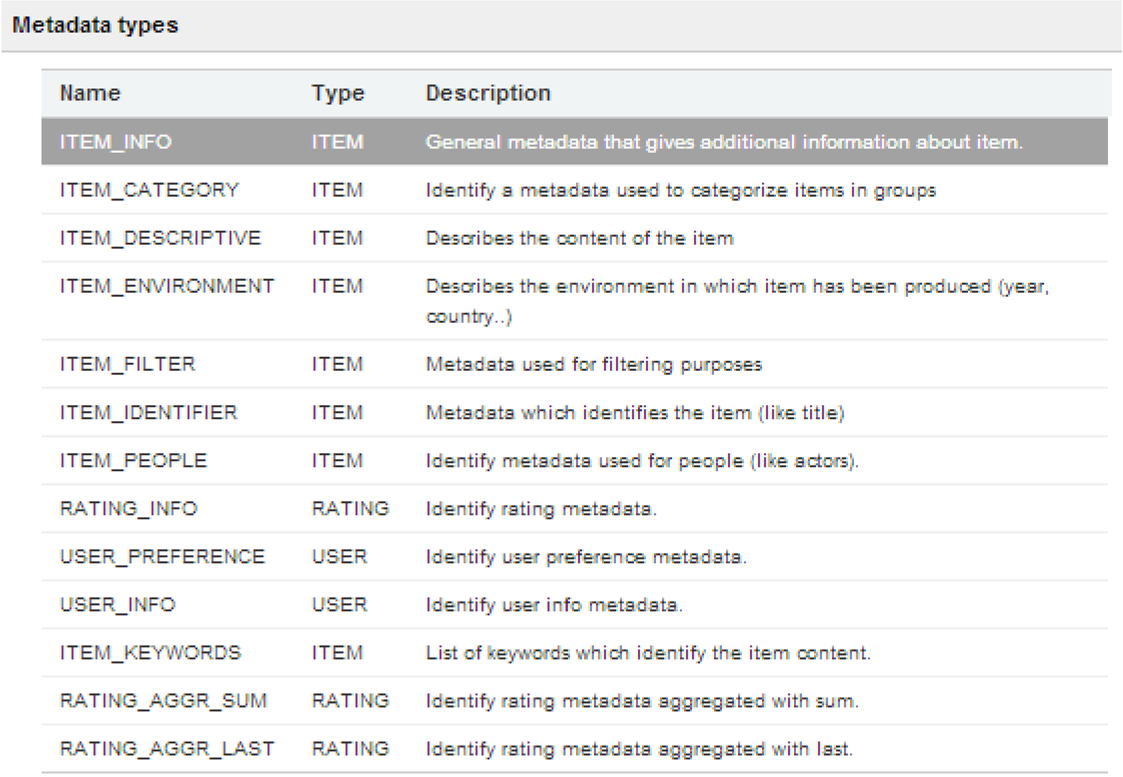
Metadata Types: metadata types configuration.

Metadata types

The *Metadata types* portlet lists all metadata types configured in the system.

For each metadata type, the table shows:

- *Name*: the name of the metadata type.
- *Type*: the type of the metadata type (one of USER, RATING, ITEM).
- *Description*: a short description of the metadata type.



Metadata types

Name	Type	Description
ITEM_INFO	ITEM	General metadata that gives additional information about item.
ITEM_CATEGORY	ITEM	Identify a metadata used to categorize items in groups
ITEM_DESCRIPTIVE	ITEM	Describes the content of the item
ITEM_ENVIRONMENT	ITEM	Describes the environment in which item has been produced (year, country..)
ITEM_FILTER	ITEM	Metadata used for filtering purposes
ITEM_IDENTIFIER	ITEM	Metadata which identifies the item (like title)
ITEM_PEOPLE	ITEM	Identify metadata used for people (like actors).
RATING_INFO	RATING	Identify rating metadata.
USER_PREFERENCE	USER	Identify user preference metadata.
USER_INFO	USER	Identify user info metadata.
ITEM_KEYWORDS	ITEM	List of keywords which identify the item content.
RATING_AGGR_SUM	RATING	Identify rating metadata aggregated with sum.
RATING_AGGR_LAST	RATING	Identify rating metadata aggregated with last.

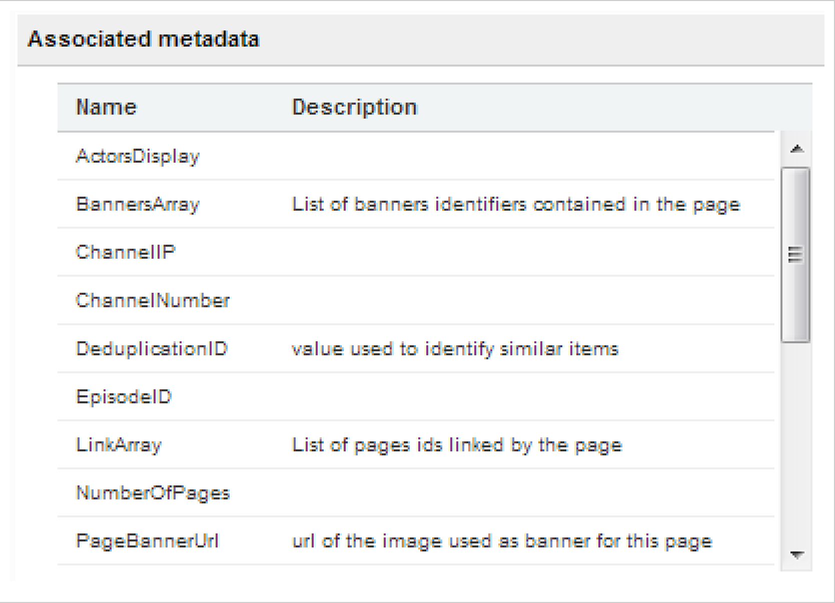
Metadata Types: Metadata types portlet.

Associated metadata

The *Associated metadata* portlet shows the list of metadata associated to the metadata type selected in the Metadata types portlet.

For each metadata, the table shows:

- *Name*: the metadata name.
- *Description*: a short description of the metadata (if any).



Associated metadata

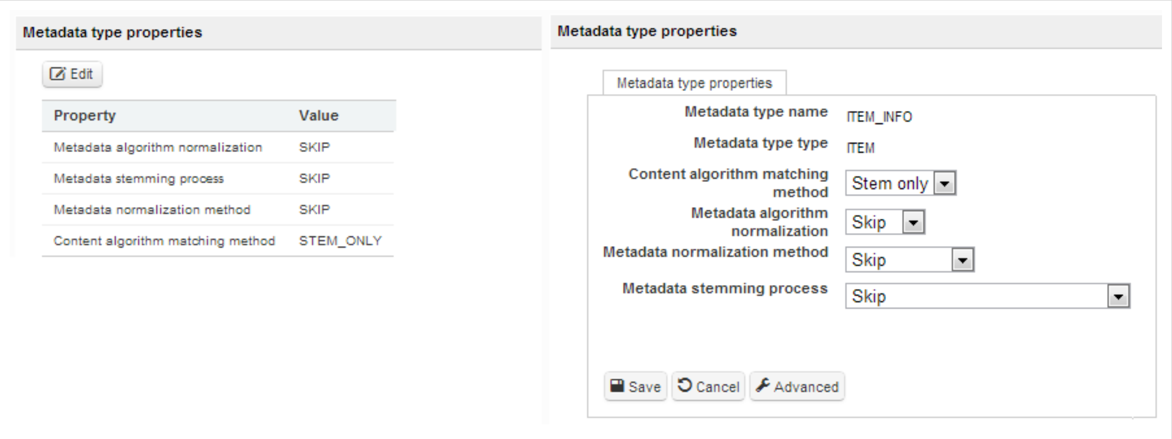
Name	Description
ActorsDisplay	
BannersArray	List of banners identifiers contained in the page
ChannelIP	
ChannelNumber	
DeduplicationID	value used to identify similar items
EpisodeID	
LinkArray	List of pages ids linked by the page
NumberOfPages	
PageBannerUrl	url of the image used as banner for this page

Metadata Types: Associated metadata portlet.

Metadata type properties

The *Metadata type properties* portlet gives an overview of the configuration related to the metadata type selected in the Metadata types portlet.

The portlet allows to modify the configuration. Click the *Edit* button to access the configuration page.



Metadata type properties

Property	Value
Metadata algorithm normalization	SKIP
Metadata stemming process	SKIP
Metadata normalization method	SKIP
Content algorithm matching method	STEM_ONLY

Metadata type properties

Metadata type properties

Metadata type name: ITEM_INFO
Metadata type type: ITEM
Content algorithm matching method: Stem only
Metadata algorithm normalization: Skip
Metadata normalization method: Skip
Metadata stemming process: Skip

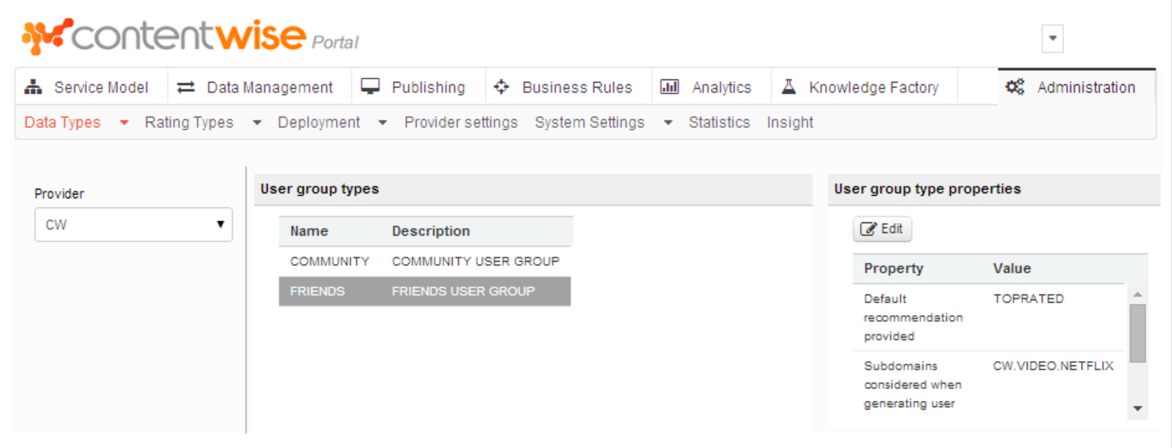
Metadata Types: Metadata type properties portlet (view and editor).

User Group Types

This section describes functionalities provided by *User Group Types* page.

The page gives an overview of the user group types that are configured in the system and allows to edit group type settings. It contains the following portlets:

- User group types: lists the available user group types.
- User group types properties: shows user group type settings and allows to modify its configuration.



Provider: CW

User group types

Name	Description
COMMUNITY	COMMUNITY USER GROUP
FRIENDS	FRIENDS USER GROUP

User group type properties

Property	Value
Default recommendation provided	TOPRATED
Subdomains considered when generating user	CW.VIDEO.NETFLIX

User Group Types: user group types configuration.

User group types

The *User group types* portlet lists all user group types configured in the system.

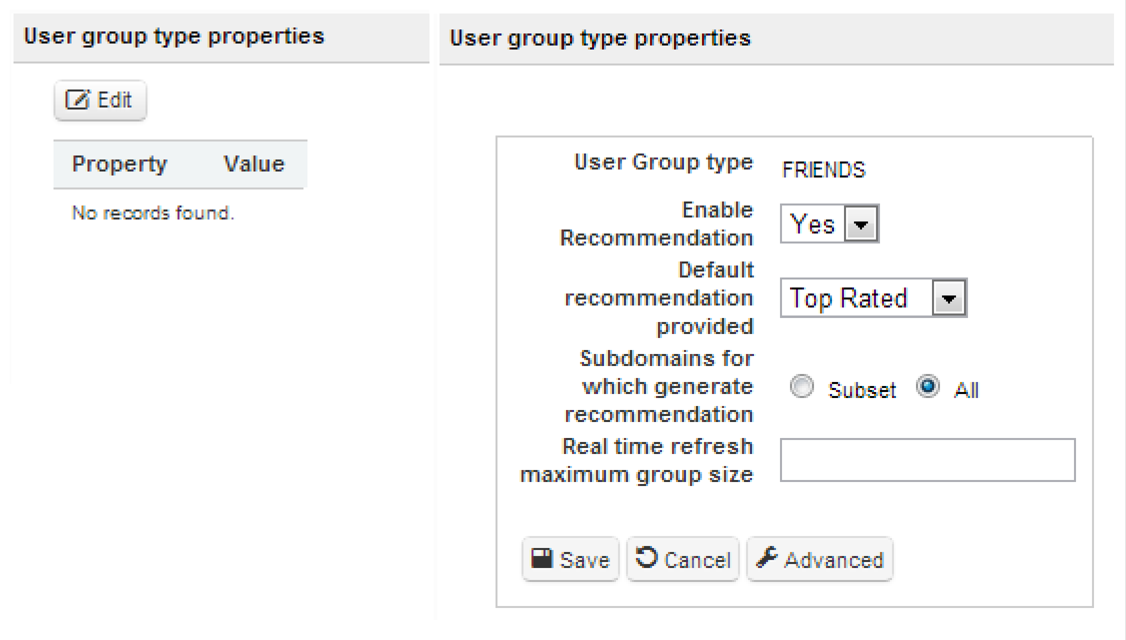
For each user group type, the table shows:

- *Name*: the name of the user group type.
- *Description*: a short description of the user group type.

User group type properties

The *User group type properties* portlet gives an overview of the configuration related to the user group type selected in the User group types portlet.

Click the *Edit* button to modify the group type configuration.



User group type properties

Property	Value
No records found.	

User group type properties

User Group type	FRIENDS
Enable Recommendation	Yes
Default recommendation provided	Top Rated
Subdomains for which generate recommendation	○ Subset ● All
Real time refresh maximum group size	

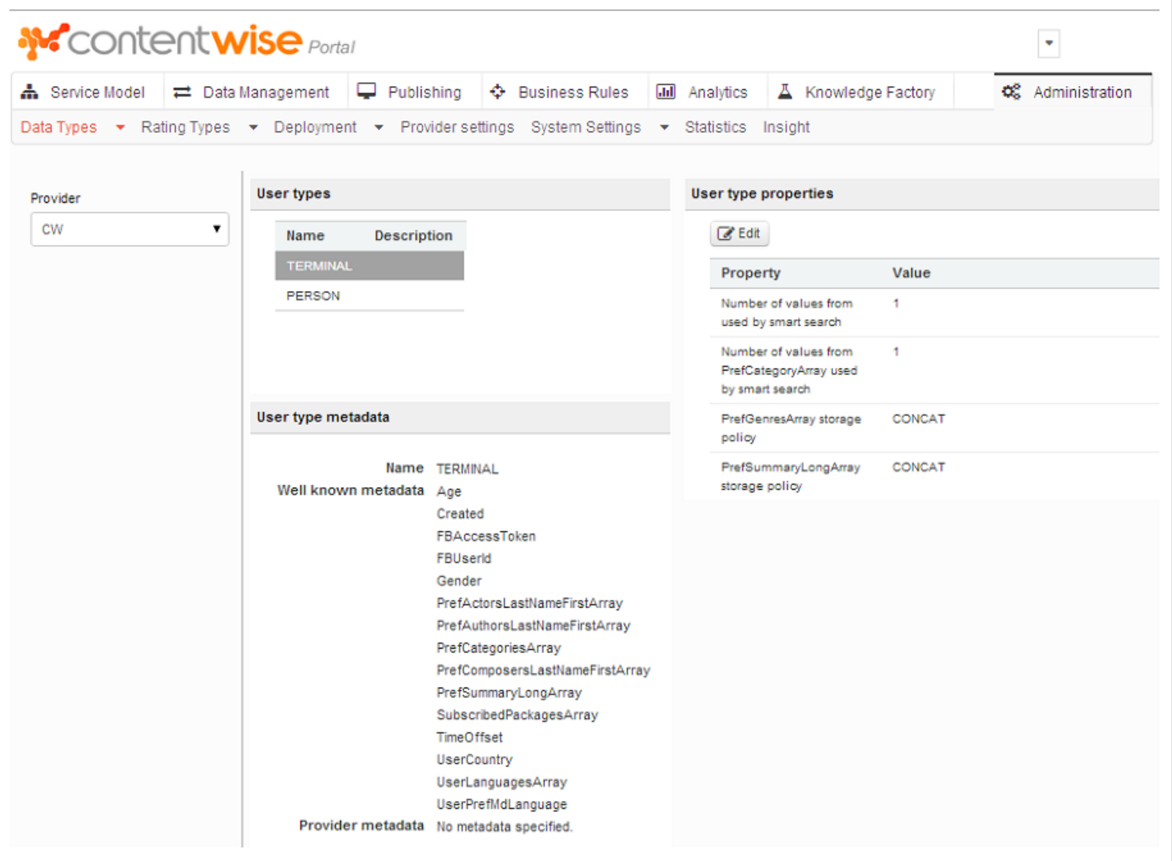
User Group Types: User group type properties portlet (view and editor).

User Types

This section describes functionalities provided by *User Types* page.

The page gives an overview of the user types that are configured in the system and allows to edit type settings. It contains the following portlets:

- User types: lists the available user types.
- User types metadata: lists the metadata that are defined for the selected user type.
- User types properties: shows user type settings and allows to modify its configuration.



Provider: CW

User types

Name	Description
TERMINAL	
PERSON	

User type properties

Property	Value
Number of values from used by smart search	1
Number of values from PrefCategoryArray used by smart search	1
PrefGenresArray storage policy	CONCAT
PrefSummaryLongArray storage policy	CONCAT

User type metadata

Name	TERMINAL
Well known metadata	Age Created FBAccessToken FBUserId Gender PrefActorsLastNameFirstArray PrefAuthorsLastNameFirstArray PrefCategoriesArray PrefComposersLastNameFirstArray PrefSummaryLongArray SubscribedPackagesArray TimeOffset UserCountry UserLanguagesArray UserPrefMdLanguage
Provider metadata	No metadata specified.

User Types: user types configuration.

User types

The *User types* portlet lists all user types configured in the system.

For each user type, the table shows:

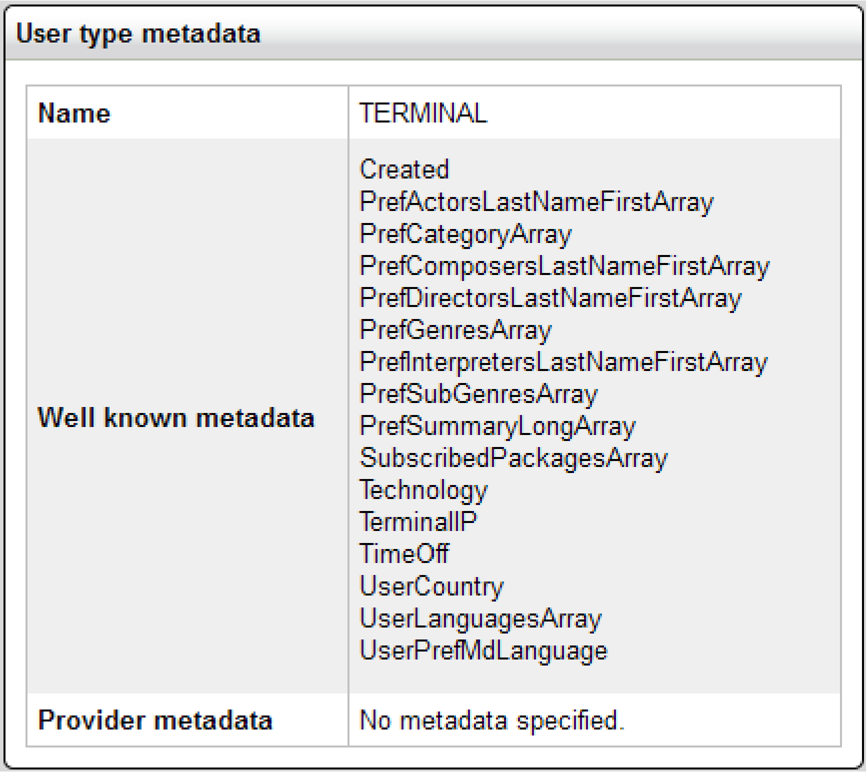
- *Name*: the name of the user type.
- *Description*: a short description of the user type (if any).

Metadata

The *User type metadata* portlet lists all the metadata configured for the user type selected in User types portlet.

There are two types of metadata:

- Well known metadata: standard metadata for the user type and custom defined metadata valid for all providers.
- Provider metadata: metadata defined for the selected provider.



User type metadata

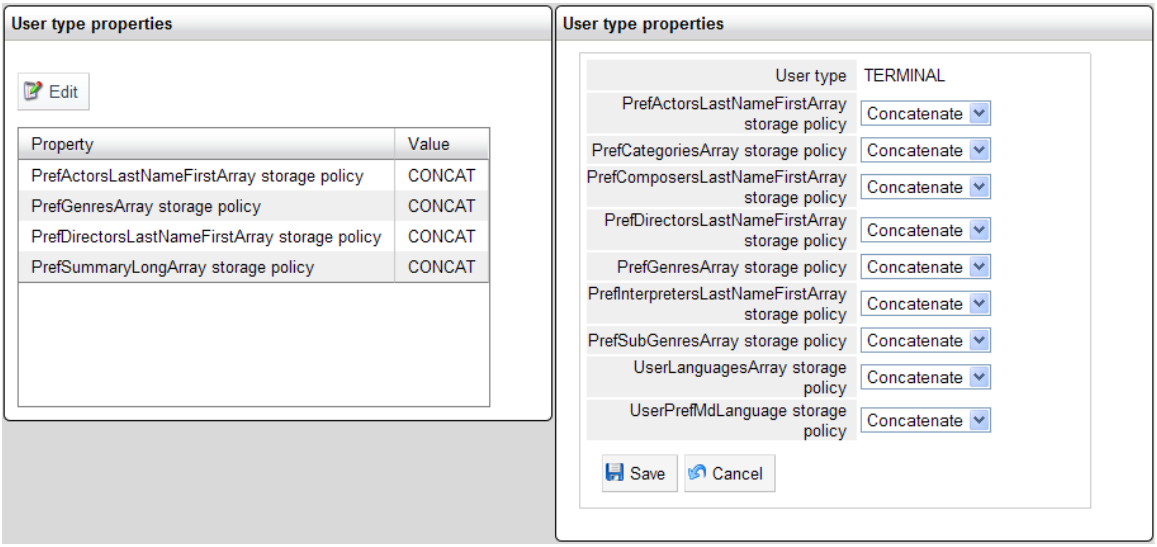
Name	TERMINAL
Well known metadata	Created PrefActorsLastNameFirstArray PrefCategoryArray PrefComposersLastNameFirstArray PrefDirectorsLastNameFirstArray PrefGenresArray PrefInterpretersLastNameFirstArray PrefSubGenresArray PrefSummaryLongArray SubscribedPackagesArray Technology TerminalIP TimeOff UserCountry UserLanguagesArray UserPrefMdLanguage
Provider metadata	No metadata specified.

User Types: User type metadata portlet.

User type properties

The *User type properties* portlet gives an overview of the configuration related to the user type selected in the User types portlet.

Click the *Edit* button to modify the type configuration.



User type properties

Property	Value
PrefActorsLastNameFirstArray storage policy	CONCAT
PrefGenresArray storage policy	CONCAT
PrefDirectorsLastNameFirstArray storage policy	CONCAT
PrefSummaryLongArray storage policy	CONCAT

User type properties

User type	TERMINAL
PrefActorsLastNameFirstArray storage policy	Concatenate
PrefCategoriesArray storage policy	Concatenate
PrefComposersLastNameFirstArray storage policy	Concatenate
PrefDirectorsLastNameFirstArray storage policy	Concatenate
PrefGenresArray storage policy	Concatenate
PrefInterpretersLastNameFirstArray storage policy	Concatenate
PrefSubGenresArray storage policy	Concatenate
UserLanguagesArray storage policy	Concatenate
UserPrefMdLanguage storage policy	Concatenate

User Types: User type properties portlet (view and editor).

Rating Types

This section describes functionalities provided by *Rating Types* section.

The section gives an overview of the rating types that are in the system and allows to manage their configuration.

The section contains the following pages:

- General properties: configure the global properties of the rating type.
- Aggregation: configure how ratings and accesses are aggregated.

- Implicit rating: configure how implicit ratings are determined for the rating type.
- History events: configure the events that must be retained among time, despite of system retention configuration.
- Vision algo: for rating types having VIDEO_CHANNEL or AUDIO_CHANNEL as item type, configure vision algo properties.
- Propagation: configure propagations and their properties.

If a provider filter is not selected, you will be prompted to select one.

Each page shows:

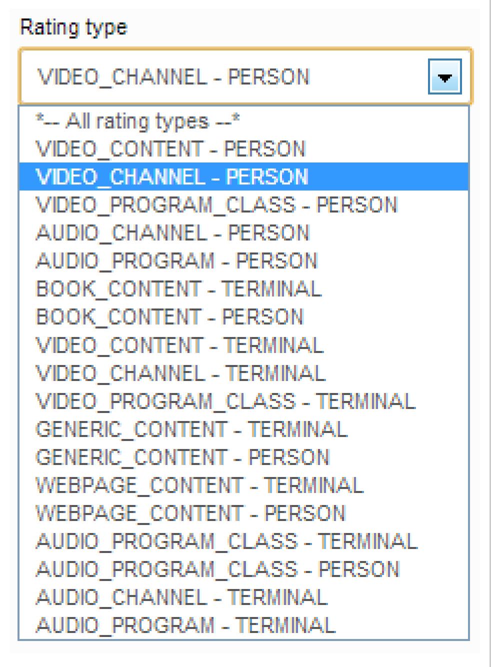
- Rating types filter: shows the list of available rating types.
- Metadata portlet: lists the metadata defined for the selected rating type.

Rating types filter

The *Rating types* filter lists all the rating types configured in the system.

For each rating type, the filter shows:

- *Item type*: the item type associated to the rating type.
- *User type*: the user type associated to the rating type.



Rating type

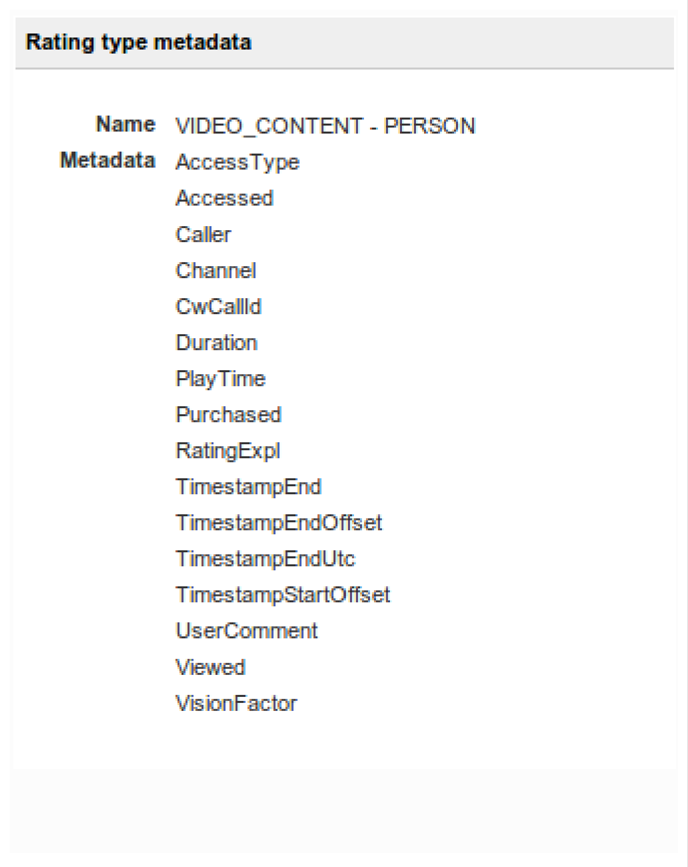
VIDEO_CHANNEL - PERSON

-- All rating types --
VIDEO_CONTENT - PERSON
VIDEO_CHANNEL - PERSON
VIDEO_PROGRAM_CLASS - PERSON
AUDIO_CHANNEL - PERSON
AUDIO_PROGRAM - PERSON
BOOK_CONTENT - TERMINAL
BOOK_CONTENT - PERSON
VIDEO_CONTENT - TERMINAL
VIDEO_CHANNEL - TERMINAL
VIDEO_PROGRAM_CLASS - TERMINAL
GENERIC_CONTENT - TERMINAL
GENERIC_CONTENT - PERSON
WEBPAGE_CONTENT - TERMINAL
WEBPAGE_CONTENT - PERSON
AUDIO_PROGRAM_CLASS - TERMINAL
AUDIO_PROGRAM_CLASS - PERSON
AUDIO_CHANNEL - TERMINAL
AUDIO_PROGRAM - TERMINAL

Rating Types: Rating type filter.

Metadata

The *Rating type metadata* portlet lists all the metadata configured for the rating type selected in Rating types filter.



Rating type metadata

Name VIDEO_CONTENT - PERSON

Metadata AccessType
Accessed
Caller
Channel
CwCallId
Duration
PlayTime
Purchased
RatingExpl
TimestampEnd
TimestampEndOffset
TimestampEndUtc
TimestampStartOffset
UserComment
Viewed
VisionFactor

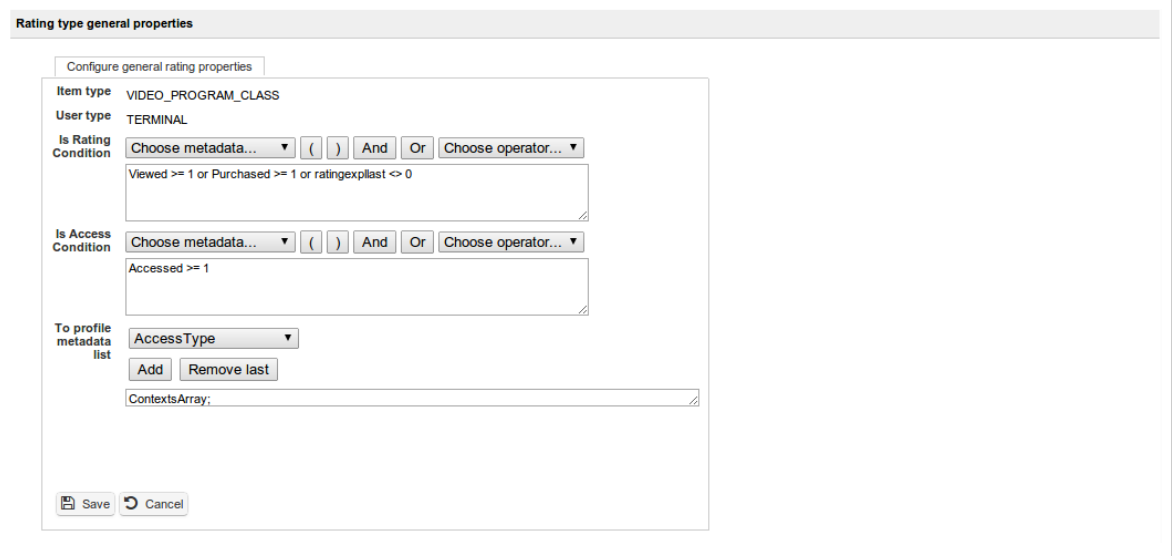
Rating Types: Rating type metadata portlet.

General properties

This page allows to manage the general properties of the rating type.

The page shows the current configuration. To modify it:

1. Update the configuration.
2. Click *Save* to submit the changes, *Cancel* to undo the operation.



Rating type general properties

Configure general rating properties

Item type VIDEO_PROGRAM_CLASS

User type TERMINAL

Is Rating Condition: Viewed >= 1 or Purchased >= 1 or ratingexpllast <> 0

Is Access Condition: Accessed >= 1

To profile metadata list: AccessType

ContextsArray;

Rating Types: Rating type general properties portlet (edit mode).

- **Is Rating Condition**: it is the condition used to identify if a user event should be considered as a rating, this means that, depending on the item type configuration, the content should be removed from user recommendation
- **Is Access Condition**: it is the condition used to identify if a user event should be considered as an access, this means that, depending on the item type configuration, the content should not be removed from user recommendation
- **To profile metadata list**: it is the list of metadata that can be used to configure profile's context options

Aggregation properties

This page allows to manage the aggregation properties of the rating type.

The page shows the current configuration. To modify it:

1. Update the configuration.
2. Click *Save* to submit the changes, *Cancel* to undo the operation.

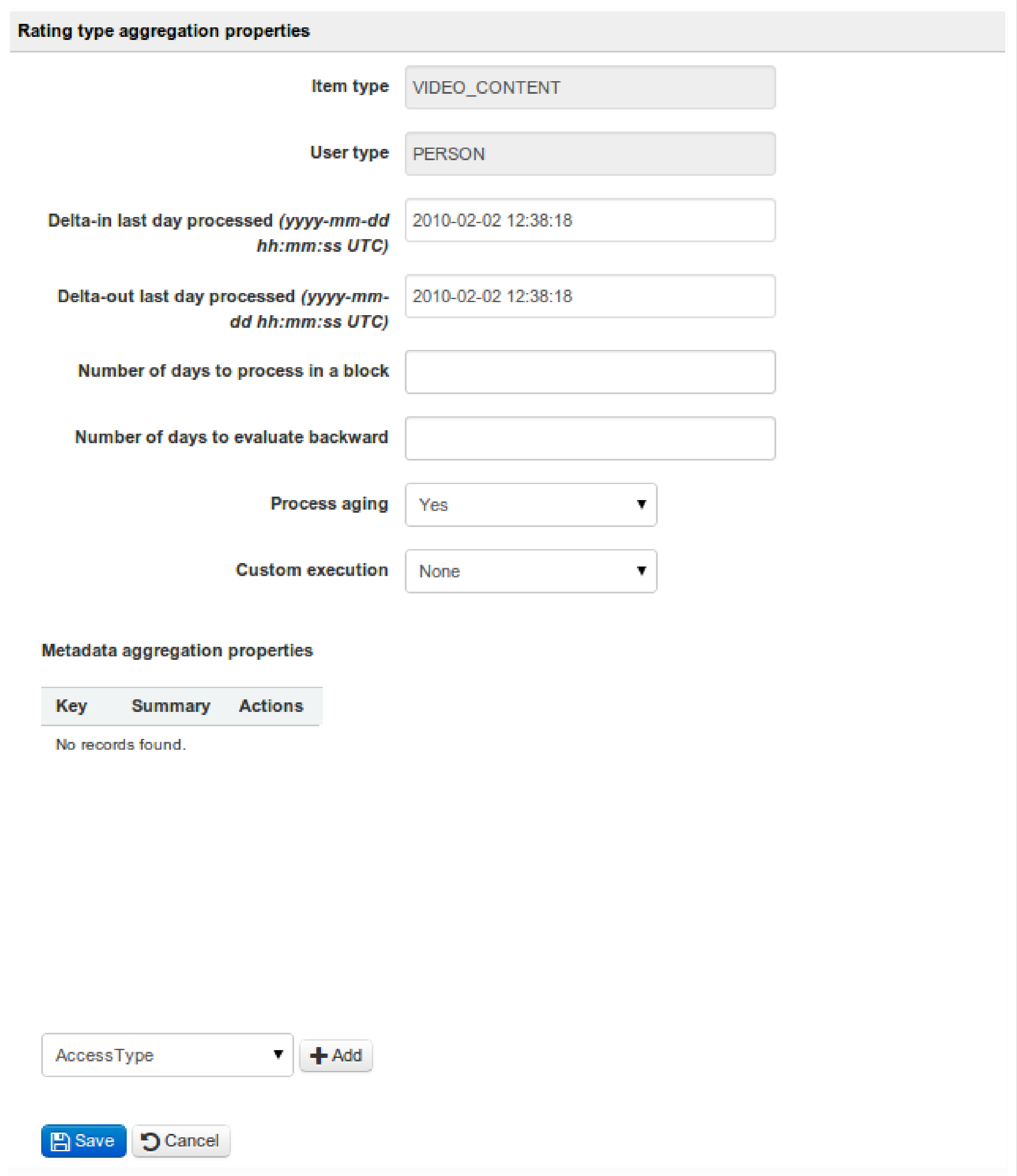
It is possible to customize the execution of the aggregation process. The following custom executions are available:

- Define a number of days to be considered by the process, starting from the last day processed.
- Define a date interval to be considered by the process.

By modifying metadata aggregation properties, you can specify how metadata values are aggregated. You can:

- Remove a metadata configuration: click the remove button of the metadata to remove.
- Edit a metadata configuration: click the edit button of the metadata and modify its configuration.
- Add a configuration for a metadata: select the metadata from the list of the availables, click add and then edit its configuration.

> ⚠ **Warning**
> ContentWise is provided with a list of default aggregation rules. Changes of the aggregation properties may cause errors during the item accesses aggregation process.



Rating type aggregation properties

Item type	VIDEO_CONTENT
User type	PERSON
Delta-in last day processed *(yyyy-mm-dd hh:mm:ss UTC)*	2010-02-02 12:38:18
Delta-out last day processed *(yyyy-mm-dd hh:mm:ss UTC)*	2010-02-02 12:38:18
Number of days to process in a block	
Number of days to evaluate backward	
Process aging	Yes
Custom execution	None

Metadata aggregation properties

Key	Summary	Actions
No records found.		

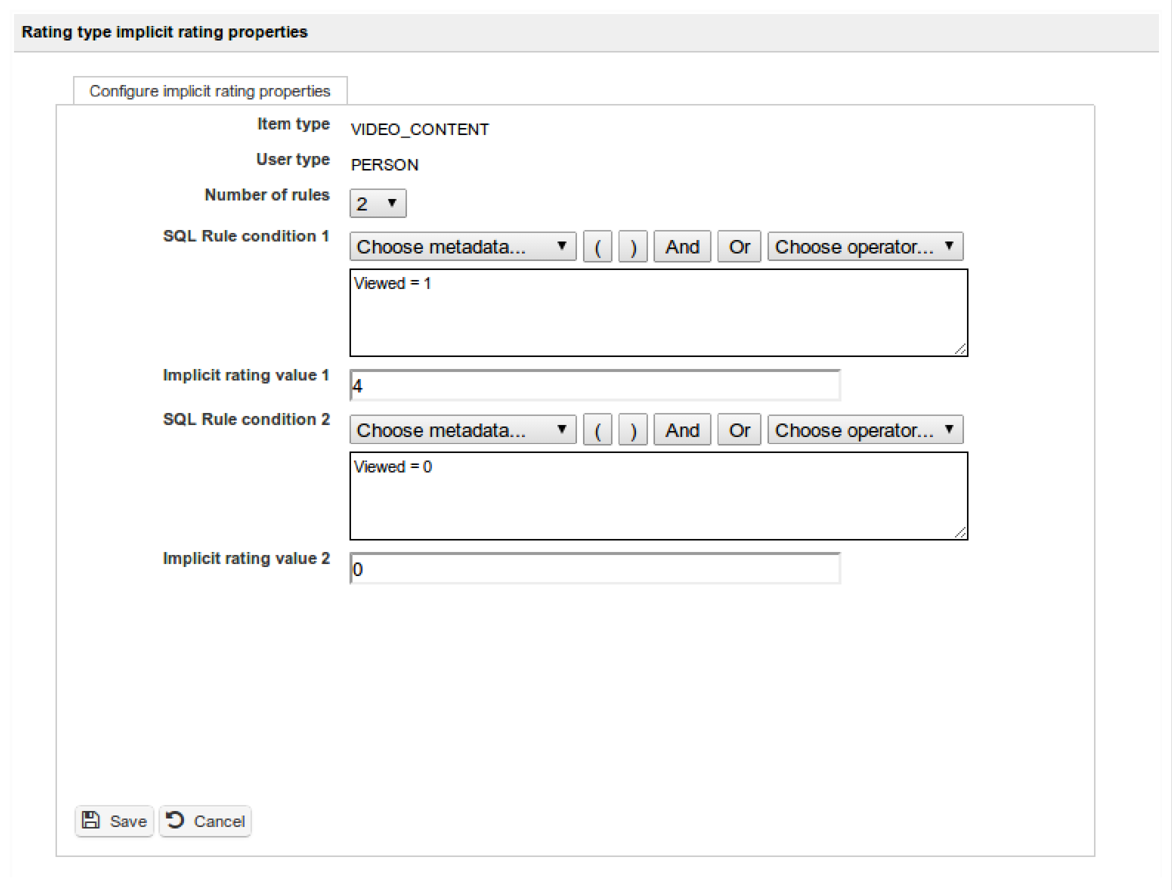
Rating Types: Rating type aggregation properties editor.

Implicit rating properties

This page allows to configure the implicit rating calculation rules. They define how implicit rating is determined according to ratings metadata values.

The page shows the current configuration. To modify it:

1. Update the configuration.
2. Click *Save* to submit the changes, *Cancel* to undo the operation.



Rating type implicit rating properties

Configure implicit rating properties

Item type	VIDEO_CONTENT
User type	PERSON
Number of rules	2
SQL Rule condition 1	Viewed = 1
Implicit rating value 1	4
SQL Rule condition 2	Viewed = 0
Implicit rating value 2	0

Rating Types: Rating type implicit properties.

History events

This page allows to configure the events that must be retained among time, despite of system retention configuration.

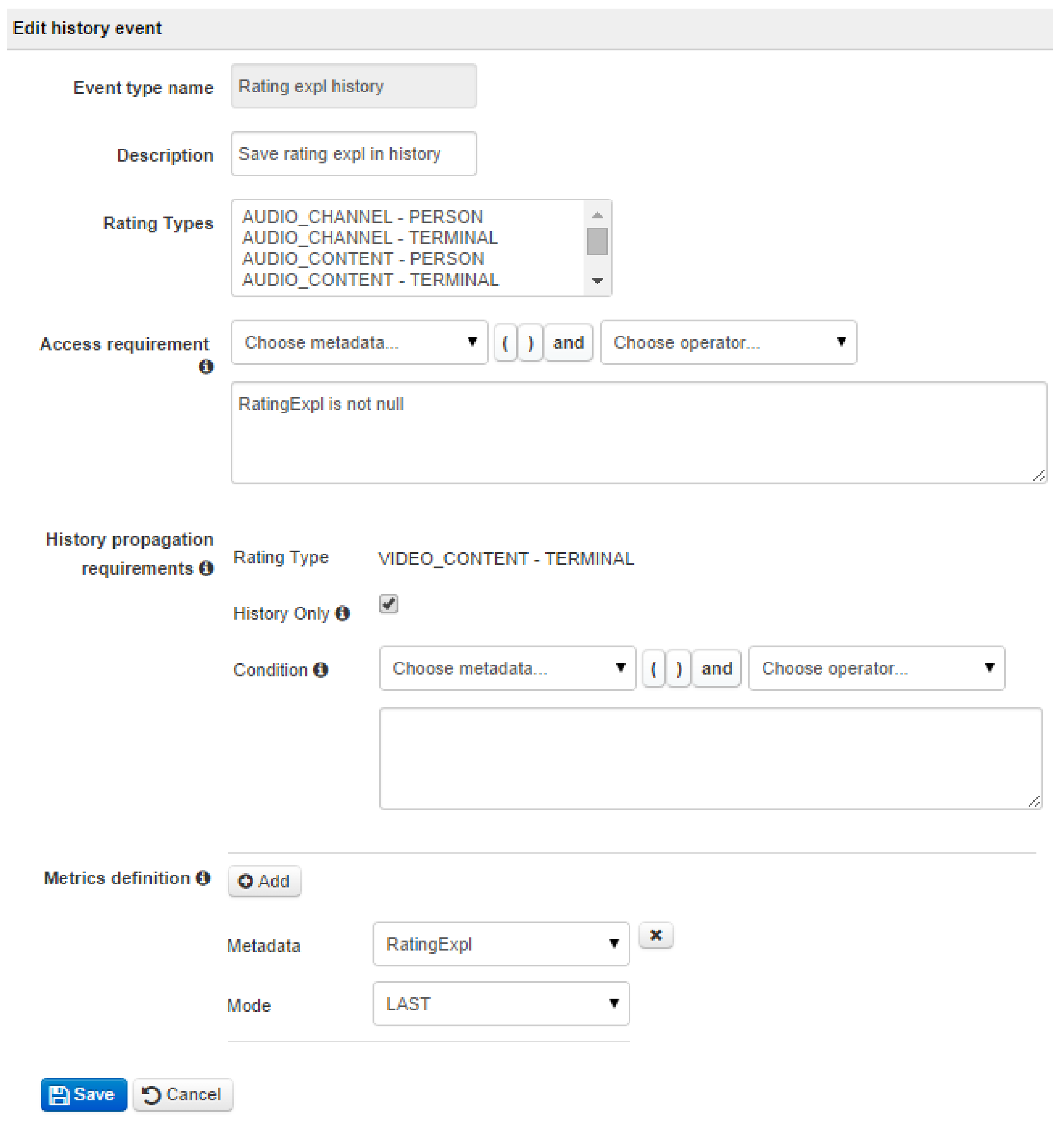
The most common event that is required to retain among time is the explicit rating action. This is the case in which a given information (the explicit rating) must be available each time it is requested.

History events is the ContentWise entry point for configuring such events. It is based on two portlets:

- The list of configured history events
- The history event configurator

Each history event is characterized by:

- A name: It is the identifier that the client must know to request the event back from the system.
- A set of rating types for which the history event is configured.
- An access requirement: it is the condition that an item access must satisfy to be considered an event that belongs to the current configuration
- A set of optional history propagation requirements. Each rating type can define an additional condition to be applied together with the access requirement.
- A set of metrics that have to be stored together with the event.



Edit history event

Event type name: Rating expl history

Description: Save rating expl in history

Rating Types:
AUDIO_CHANNEL - PERSON
AUDIO_CHANNEL - TERMINAL
AUDIO_CONTENT - PERSON
AUDIO_CONTENT - TERMINAL

Access requirement: Choose metadata... ▾ ( ) and Choose operator... ▾

RatingExpl is not null

History propagation requirements

Rating Type: VIDEO_CONTENT - TERMINAL

History Only: ☑

Condition: Choose metadata... ▾ ( ) and Choose operator... ▾

Metrics definition: Add

Metadata: RatingExpl ▾

Mode: LAST ▾

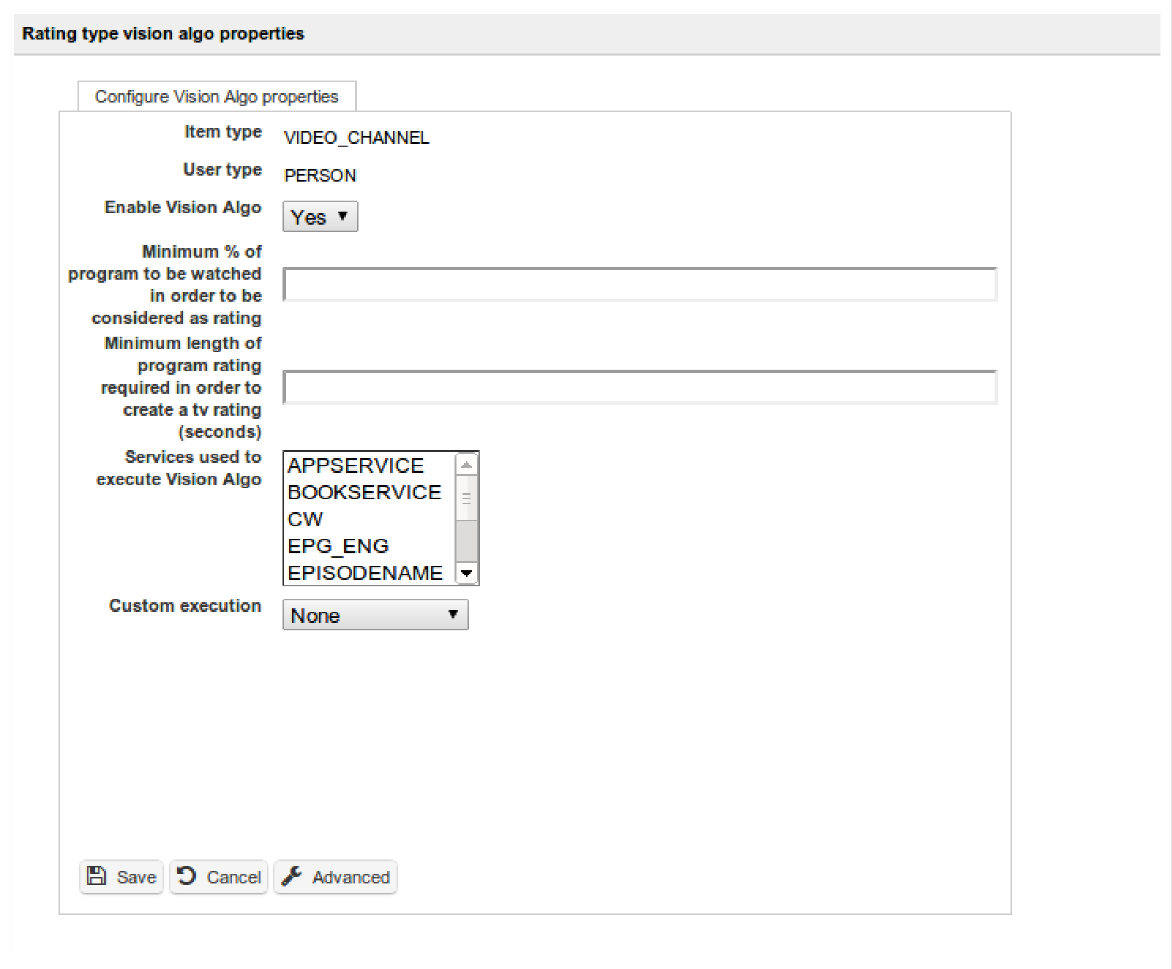
Save Cancel

History events: configuration

Vision algo properties

This page allows to configure the vision algo process. Vision algo process infers accesses and ratings of programs starting from channel accesses information (join-leave).

> **Important Note**
> Vision algo configuration is available only for rating types having VIDEO_CHANNEL or AUDIO_CHANNEL item type.



Rating type vision algo properties

Configure Vision Algo properties

Item type	VIDEO_CHANNEL
User type	PERSON
Enable Vision Algo	Yes
Minimum % of program to be watched in order to be considered as rating	
Minimum length of program rating required in order to create a tv rating (seconds)	
Services used to execute Vision Algo	APPSERVICE BOOKSERVICE CW EPG_ENG EPISODENAME
Custom execution	None

Rating Types: Rating type vision algo properties.

Propagation properties

This page allows to configure the propagation rules. They allow to propagate an access (or a rating) from a rating type to another.

For each rating type it is possible to specify a set of propagation rules.

A rating type propagation is characterized by a set of rules. Each rule:

- defines a valid condition for propagating an item access.
- is characterized by:
  - a sql like condition: the condition that must be true to propagate the item access
  - a couple <destination metadata, value>: the metadata and related value to be set on the item access propagated to the destination rating type.

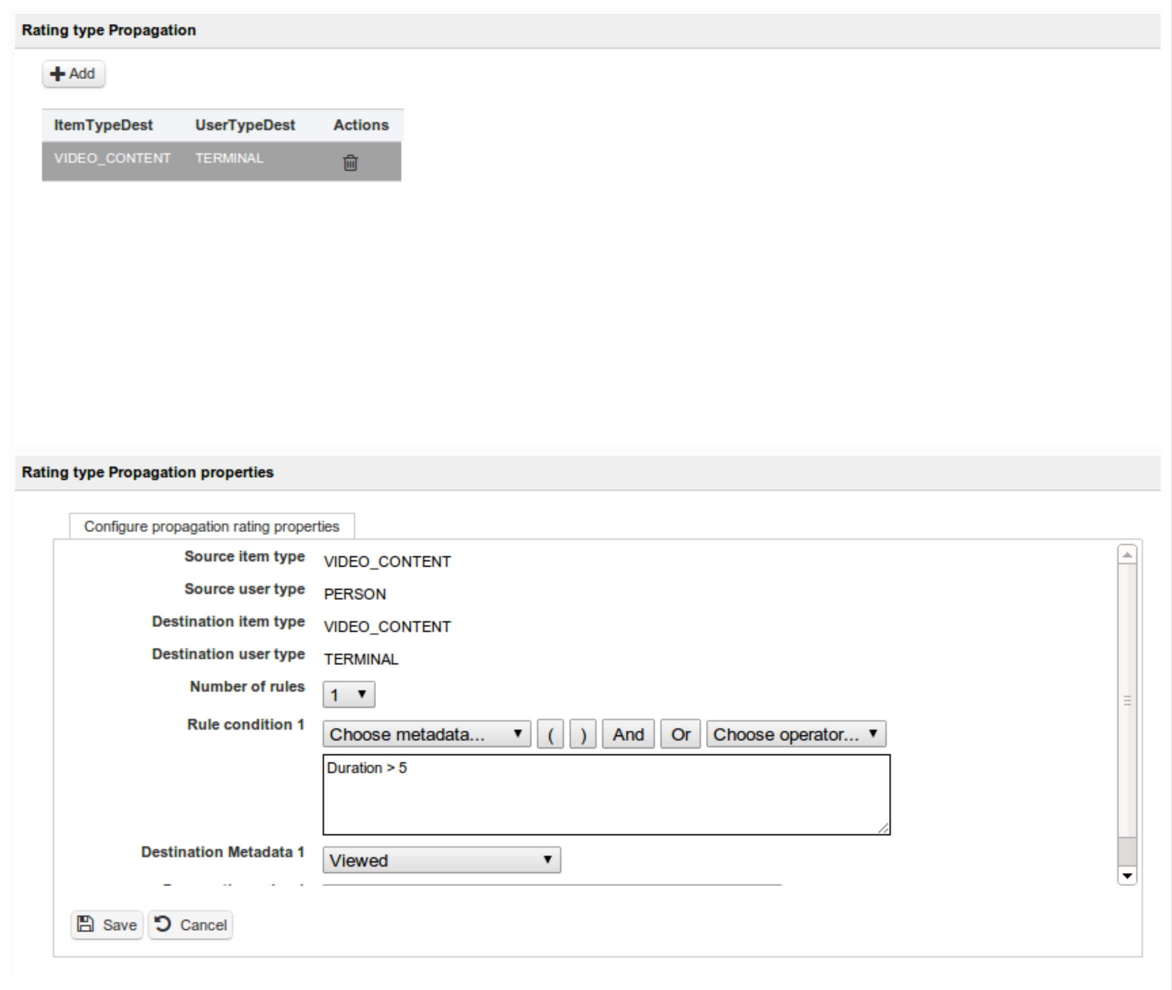
Rules of the same rating type propagation are applied in or.

E.g.: If a provided VIDEO_CONTENT/TERMINAL access has "Viewed=1" metadata, then assign "Rating=5" to the access defined for the rating type VIDEO_CONTENT/PERSON.

⚠ **Important note**
Propagation rules can be defined only on numeric metadata.

By default, propagation rules are not defined. To create a new propagation rule, select the source rating type on the left column and click the Add button to select the destination rating type. To define rules, click the Edit button of the Rating type propagation properties portlet.

Please note that the list of available destination rating types depends on the selected source rating type.



Rating type Propagation
+ Add

ItemTypeDest	UserTypeDest	Actions
VIDEO_CONTENT	TERMINAL	🗑

Rating type Propagation properties
Configure propagation rating properties

Source item type VIDEO_CONTENT

Source user type PERSON

Destination item type VIDEO_CONTENT

Destination user type TERMINAL

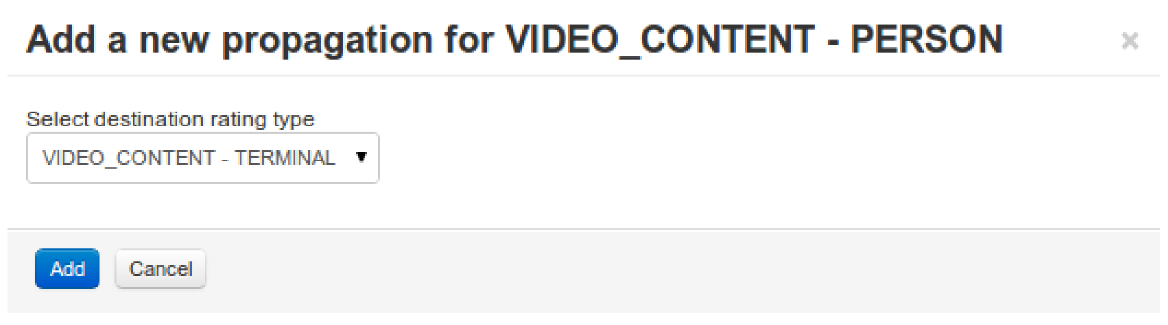
Number of rules 1

Rule condition 1 Choose metadata... ( ) And Or Choose operator...

Duration > 5

Destination Metadata 1 Viewed

Rating Types: Rating type propagations.



Add a new propagation for VIDEO_CONTENT - PERSON

Select destination rating type VIDEO_CONTENT - TERMINAL

Rating Types: Rating type propagations add modal.

Deployment

This section describes functionalities provided by *Deployment* menu.

Deployment menu gives access to the following pages:

Page	Description
Components	It is the landing page of the section. It provides an overview of the components configured in the system and allows to manage their configurations.
Component Pools	It provides an overview of the component pools configured in the system and allows to manage their configurations.

Components

This page provides an overview of component configurations and provides access to the pages that allow to create and manage components.

The page contains the following portlets:

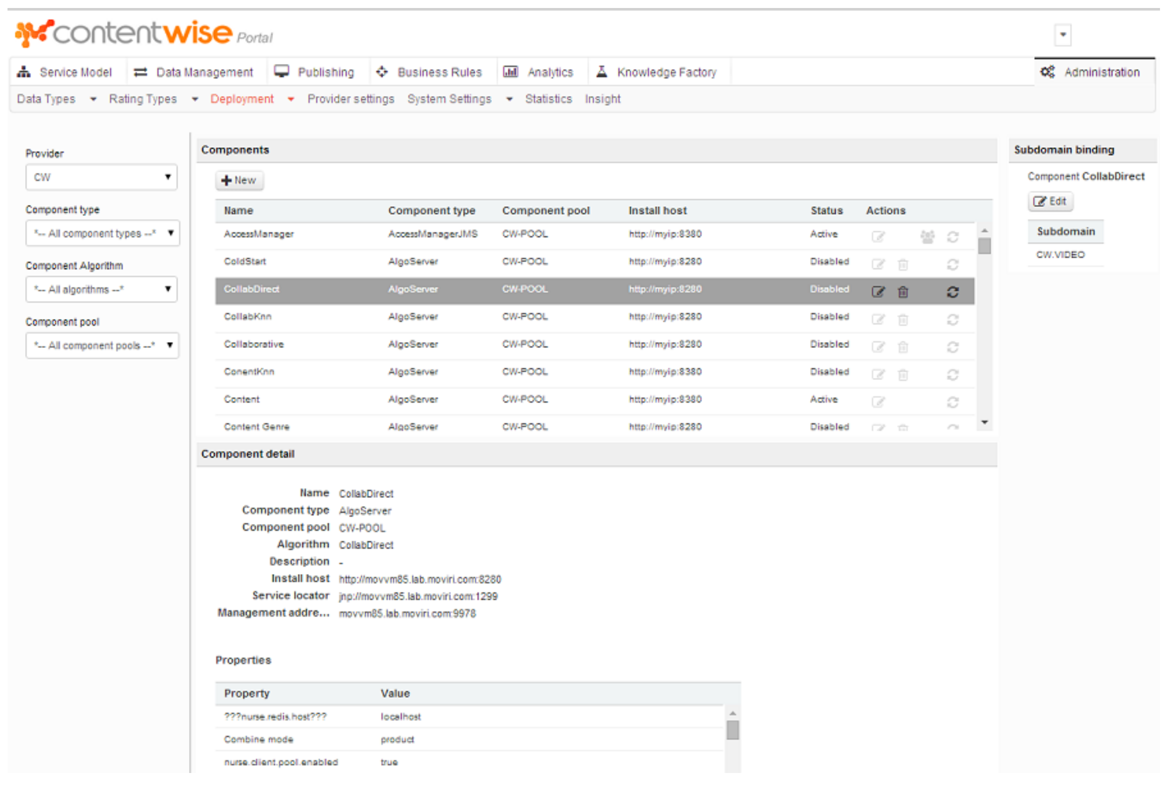
- Components: lists the components that have been defined and provides access to new/edit components pages.
- Components detail: provides details about the selected component.
- Subdomain bindings: manages the component-subdomain bindings.

See create a new component to create a new component.

You can filter by:

- component type.
- component pool.

If no filter is applied, all available components are shown.



Provider: CW

Component type: *-- All component types --*

Component Algorithm: *-- All algorithms --*

Component pool: *-- All component pools --*

Components

Name	Component type	Component pool	Install host	Status	Actions
AccessManager	AccessManagerJMS	CW-POOL	http://myip:8380	Active	
ColdStart	AlgoServer	CW-POOL	http://myip:8280	Disabled	
CollabDirect	AlgoServer	CW-POOL	http://myip:8280	Disabled	
CollabKnn	AlgoServer	CW-POOL	http://myip:8280	Disabled	
Collaborative	AlgoServer	CW-POOL	http://myip:8280	Disabled	
ConentKnn	AlgoServer	CW-POOL	http://myip:8380	Disabled	
Content	AlgoServer	CW-POOL	http://myip:8380	Active	
Content Genre	AlgoServer	CW-POOL	http://myip:8280	Disabled	

Component detail

Name: CollabDirect
Component type: AlgoServer
Component pool: CW-POOL
Algorithm: CollabDirect
Description: -
Install host: http://movvm85.lab.moviri.com:8280
Service locator: jnp://movvm85.lab.moviri.com:1299
Management addre...: movvm85.lab.moviri.com:9978

Properties

Property	Value
???nurse.redis.host???	localhost
Combine mode	product
nurse.client.pool.enabled	true

Subdomain binding

Component CollabDirect

Subdomain
CW.VIDEO

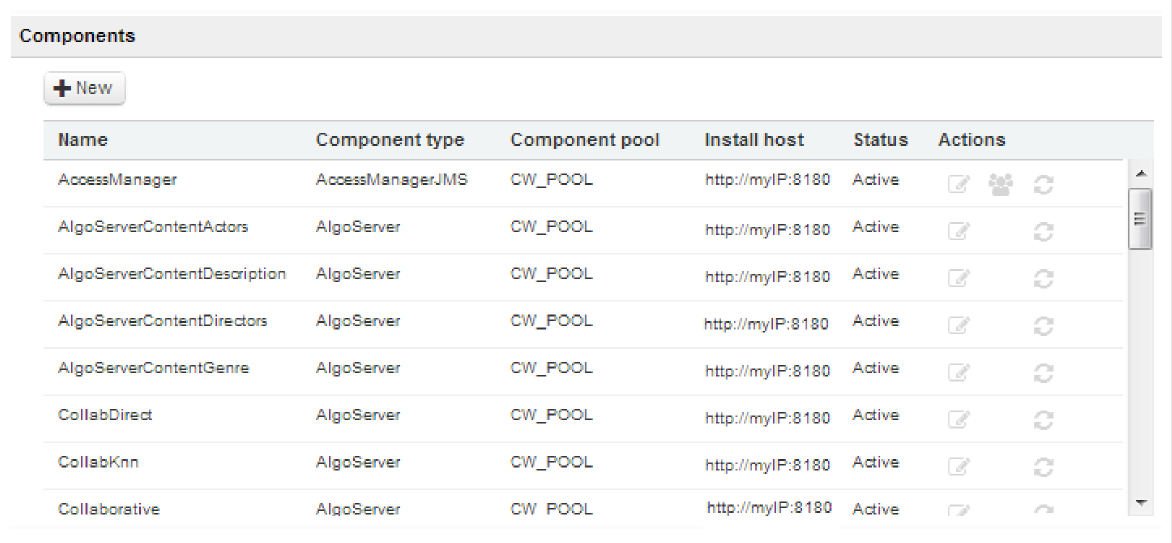
Components: list of configured components and related configurations.

Components

This portlet provides an overview of component configuration and provides access to component configuration pages.

For each component, the table shows:

- *Name*: the name of the component.
- *Component type*: the type of the component.
- *Component pool*: the component pool for which the component has been defined.
- *Install host*: the url of the machine the component is associated to.
- *Status*: Indicates if the component is active or not.
- *Actions*: you can:
  - *edit*: modifies the component configuration.
  - *delete*: removes the component from the system. Active components cannot be deleted.
  - *clear user cache*: only for RecoServer and AccessManager components, forces the clear of the user cache.



Components

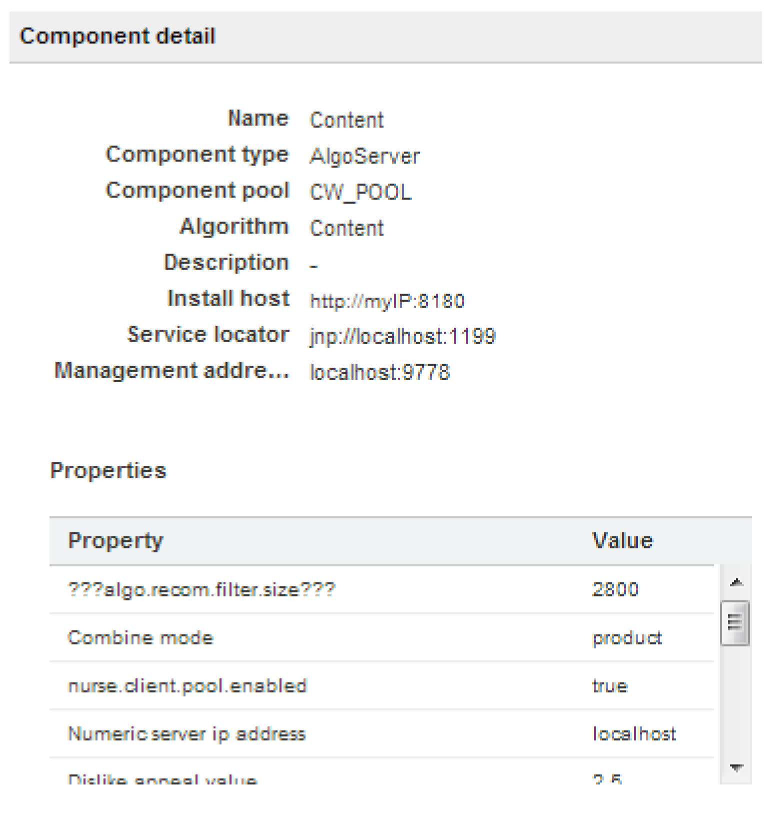
Name	Component type	Component pool	Install host	Status	Actions
AccessManager	AccessManagerJMS	CW_POOL	http://myIP:8180	Active	
AlgoServerContentActors	AlgoServer	CW_POOL	http://myIP:8180	Active	
AlgoServerContentDescription	AlgoServer	CW_POOL	http://myIP:8180	Active	
AlgoServerContentDirectors	AlgoServer	CW_POOL	http://myIP:8180	Active	
AlgoServerContentGenre	AlgoServer	CW_POOL	http://myIP:8180	Active	
CollabDirect	AlgoServer	CW_POOL	http://myIP:8180	Active	
CollabKnn	AlgoServer	CW_POOL	http://myIP:8180	Active	
Collaborative	AlgoServer	CW_POOL	http://myIP:8180	Active	

Components: Components portlet.

Component detail

This portlet provides details about the component that has been selected in the Components portlet.

- *Name*: unique component identifier.
- *Component type*: the type of the component.
- *Component pool*: the component pool.
- *Algorithm*: for AlgoServer components, specifies the algorithm managed by the component.
- *Description*: a short description.
- *Install host*, *Service locator* and *Management address*.
- *Status*: the status of the component.
- *Properties*: component configuration properties.



Component detail

Name	Content
Component type	AlgoServer
Component pool	CW_POOL
Algorithm	Content
Description	-
Install host	http://myIP:8180
Service locator	jnp://localhost:1199
Management addre...	localhost:9778

Properties

Property	Value
???algo.recom.filter.size???	2800
Combine mode	product
nurse.client.pool.enabled	true
Numeric server ip address	localhost
Dislike appeal value	2.5

Components: Component detail portlet.

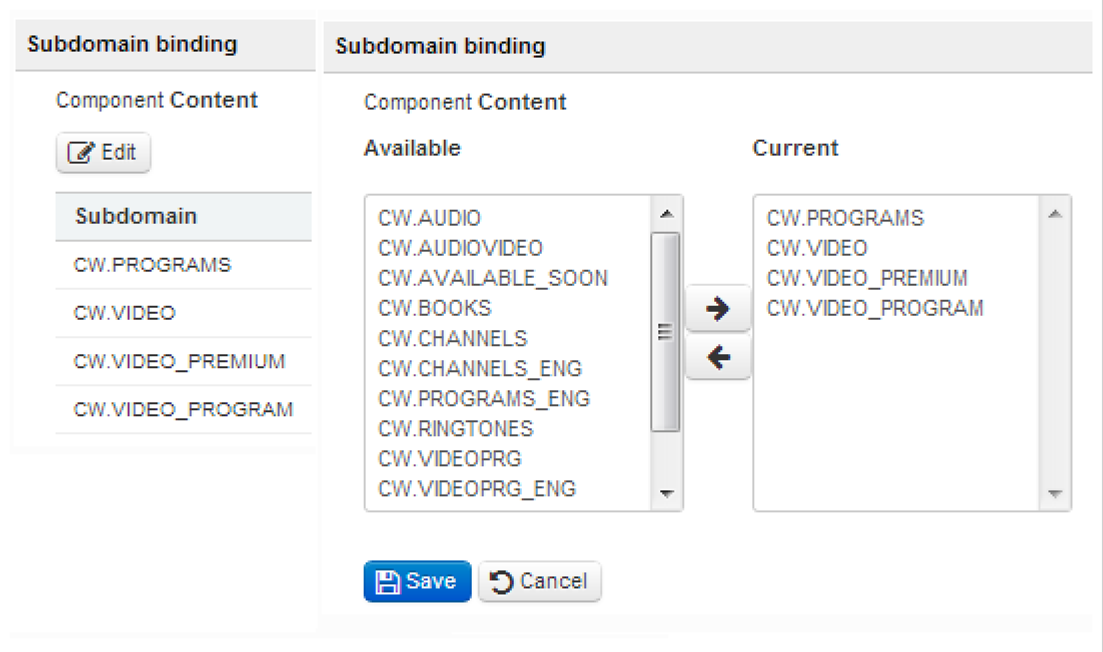
Subdomain binding

This portlet provides information about the bindings between the component selected in Components portlet and the subdomains that are configured in the system.

The portlet lists the subdomains that are currently bound to the component.

To modify the current configuration:

1. Click *Edit* to change the configuration.
2. Select a subdomain and click on left or right arrows to modify the bindings. This step can be repeated for more than a subdomain.
3. Click *Save* to submit the new configuration, *Cancel* to undo the operation



Components: Subdomain binding portlet (view and edit).

Create a new component

This section shows how to add a new component in the system.

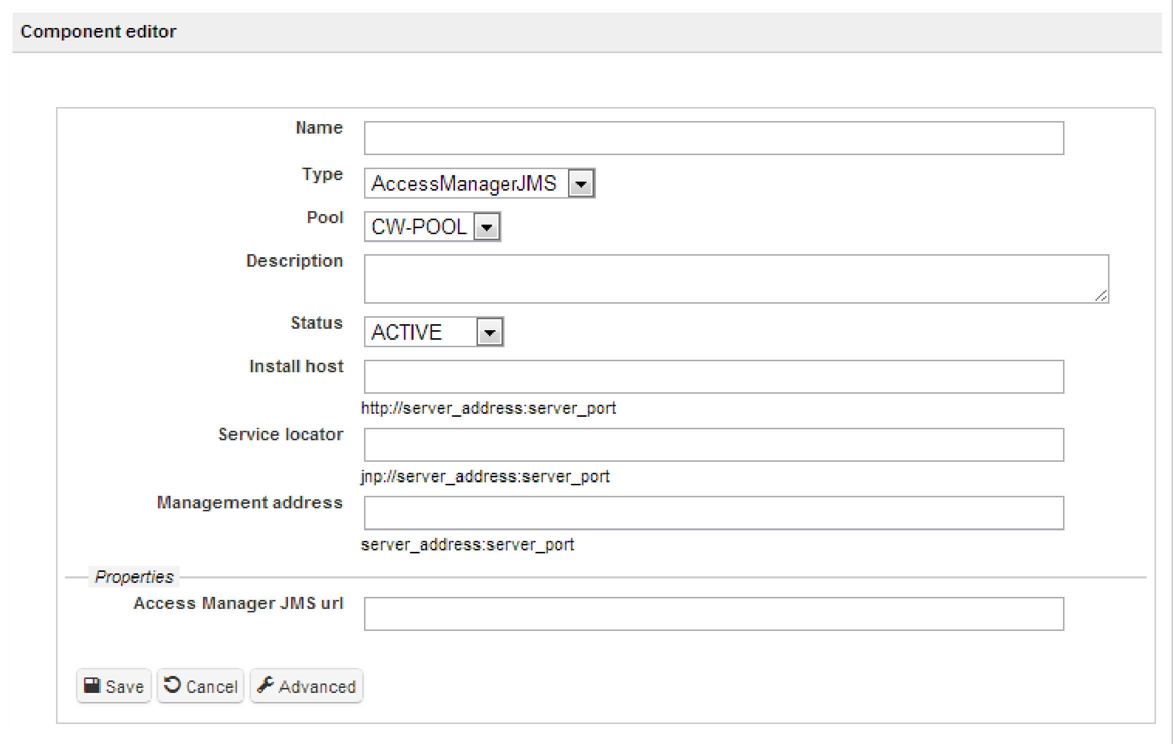
To create a new component click the *New* button in the Components portlet.

The following information must be provided:

- *Name*: unique component identifier.

> **Warning**
> Valid characters are **[A-Za-z0-9]**, **-**, _, **.**

- *Type*: the type of the component to create.
- *Pool*: the component pool.
- *Description*: a short description (optional).
- *Status*: the status of the component:
  - *Active*
  - *Disabled*
- *Install host*
- *Service locator*
- *Management address*
- *Properties*: a set of properties of the component, according to the component type.



Component editor

Name	
Type	AccessManagerJMS
Pool	CW-POOL
Description	
Status	ACTIVE
Install host	http://server_address:server_port
Service locator	jnp://server_address:server_port
Management address	server_address:server_port

Properties

Access Manager JMS url	

Components: Component editor portlet.

Component Pools

This page provides an overview of component pools configurations and provides access to the pages that allow to create and manage component pools.

In the *Component Pools* page you can:

- have an overview of the component pools configured in the system.
- create new component pools and edit their configurations.

See create a new component pool to create a new component pool.

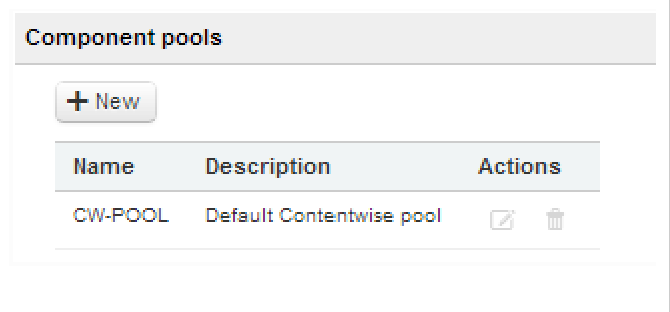
Component Pools

This portlet provides an overview of component pools and provides access to their configuration pages.

For each component pool, the table shows:

- *Name*: unique pool identifier.
- *Description*: a short description.
- *Actions*: you can:
  - *edit*: modifies the description of the component pool.
  - *delete*: removes the component pool from the system.

> ⚠ **Warning**
> If a component pool is deleted, all associated components will be removed from the system.



Component pools

Name	Description	Actions
CW-POOL	Default Contentwise pool	

Component Pools: Component pools portlet.

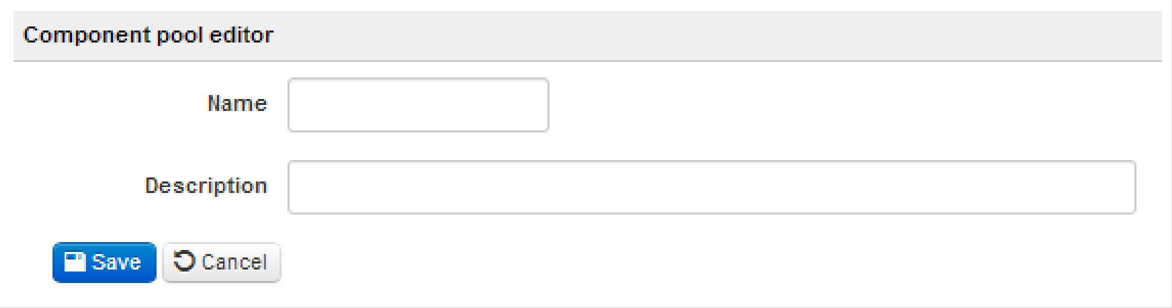
Create a new component pool

This section shows how to add a new component pool in the system.

To create a new component pool click the *New* button in the Component Pools portlet.

The following information must be provided:

- *Name*: unique component pool identifier. Valid characters are **[A-Za-z0-9]**, **-**, _, **.**.
- *Description*: a short description (optional).



Component Pools: Component pool editor portlet.

Provider Settings

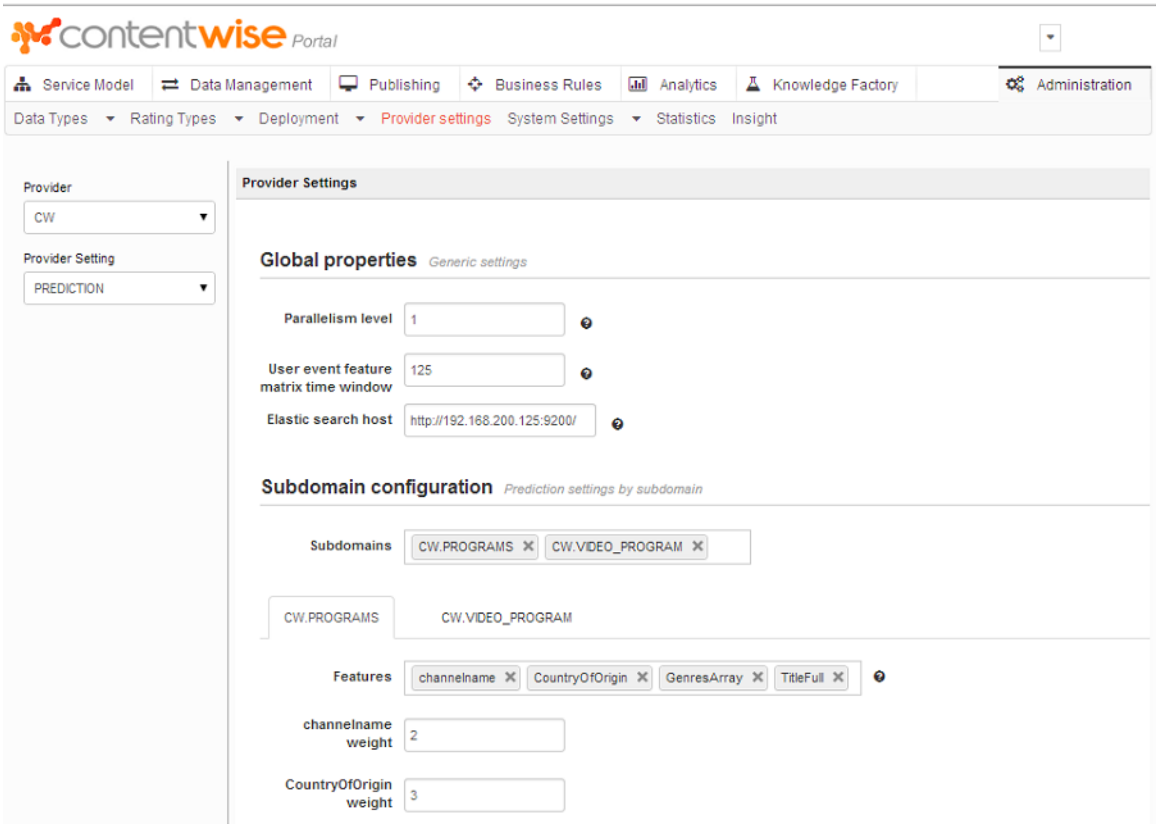
This page allows to configure different settings for the providers that are available in the system.

To define a new provider, please contact ContentWise Support.

> ⚠ **Warning**
> Provider settings changes may impact system both on functional and performance aspects. Please pay attention in updating a provider configuration.
> Provider settings should be configured during ContentWise integration project. Any further change, should be carefully evaluated.

There are different settings that can be applied to a provider, according to the impact they have on the available functionalities.

- Freshness: required to provide live events recommendations, freshness configuration defines the live windows to be supported by the system
- Prediction: allows to configure the settings required to provide the prediction algorithms recommendations
- Profile: allows to configure some provider-based options regarding the management of user profile
- Utility: provider specific utility settings.



Provider: CW

Provider Setting: PREDICTION

Provider Settings

Global properties *Generic settings*

Parallelism level: 1

User event feature matrix time window: 125

Elastic search host: http://192.168.200.125:9200/

Subdomain configuration *Prediction settings by subdomain*

Subdomains: CW.PROGRAMS ✕ CW.VIDEO_PROGRAM ✕

CW.PROGRAMS | CW.VIDEO_PROGRAM

Features: channelname ✕ CountryOfOrigin ✕ GenresArray ✕ TitleFull ✕

channelname weight: 2

CountryOfOrigin weight: 3

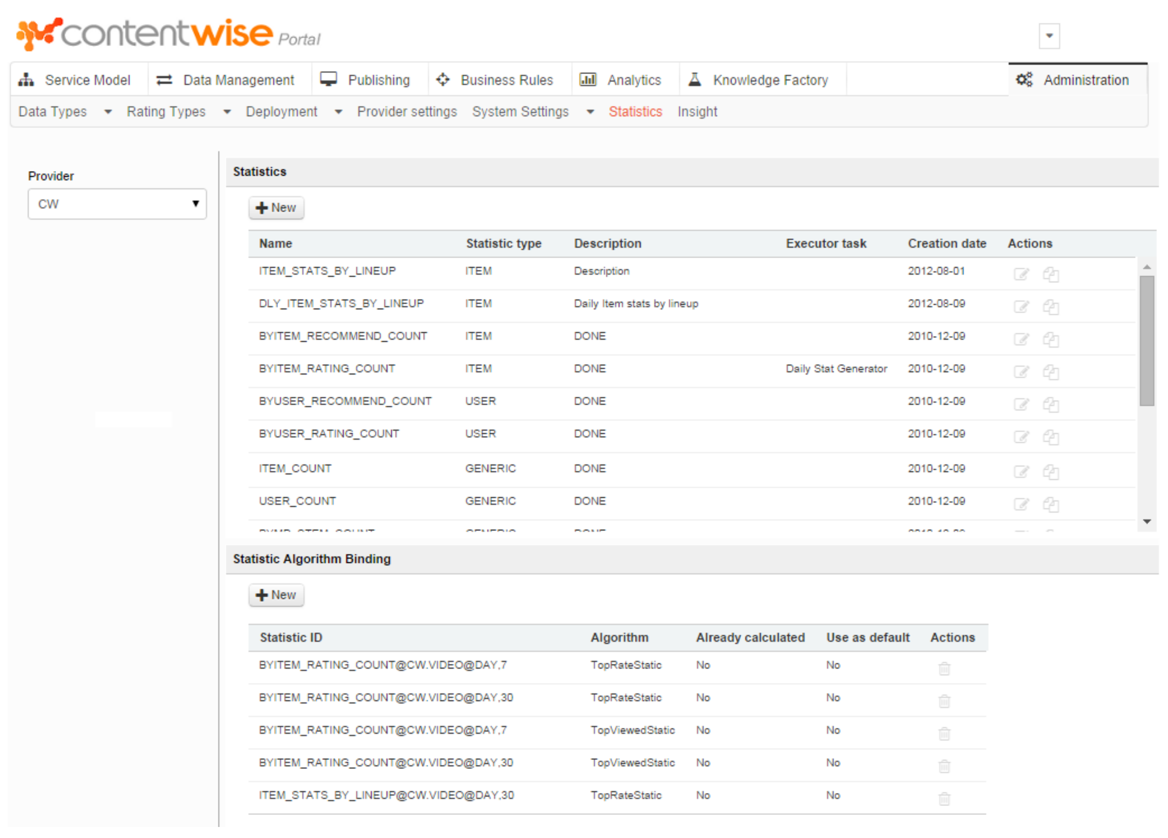
Provider settings page

Statistics

This page provides an overview of statistics configurations and provides access to the pages that allow to create and manage statistics.

Warning
Statistics configuration is very important to guarantee the correct behavior of the system.

- Do not modify existing statistics without sharing the changes with ContentWise Support.
- To avoid performance issues, please always contact ContentWise Support before creating new statistics.



Statistics

Name	Statistic type	Description	Executor task	Creation date	Actions
ITEM_STATS_BY_LINEUP	ITEM	Description		2012-08-01	
DLY_ITEM_STATS_BY_LINEUP	ITEM	Daily Item stats by lineup		2012-08-09	
BYITEM_RECOMMEND_COUNT	ITEM	DONE		2010-12-09	
BYITEM_RATING_COUNT	ITEM	DONE	Daily Stat Generator	2010-12-09	
BYUSER_RECOMMEND_COUNT	USER	DONE		2010-12-09	
BYUSER_RATING_COUNT	USER	DONE		2010-12-09	
ITEM_COUNT	GENERIC	DONE		2010-12-09	
USER_COUNT	GENERIC	DONE		2010-12-09	

Statistic Algorithm Binding

Statistic ID	Algorithm	Already calculated	Use as default	Actions
BYITEM_RATING_COUNT@CW.VIDEO@DAY,7	TopRateStatic	No	No	
BYITEM_RATING_COUNT@CW.VIDEO@DAY,30	TopRateStatic	No	No	
BYITEM_RATING_COUNT@CW.VIDEO@DAY,7	TopViewedStatic	No	No	
BYITEM_RATING_COUNT@CW.VIDEO@DAY,30	TopViewedStatic	No	No	
ITEM_STATS_BY_LINEUP@CW.VIDEO@DAY,30	TopRateStatic	No	No	

List of statistics configured in the system

Statistics

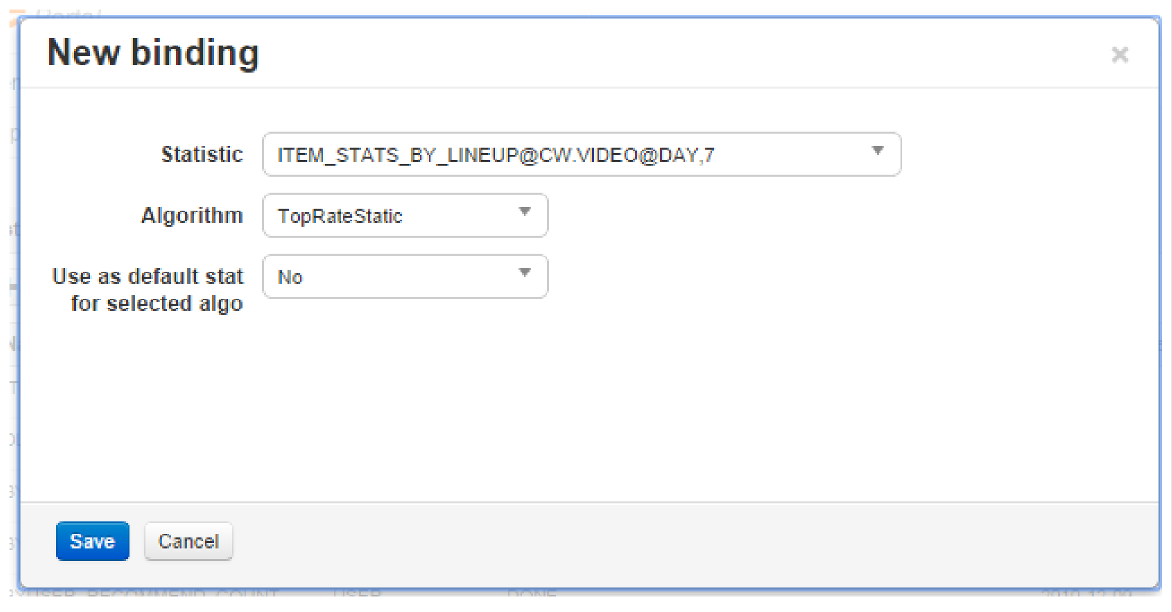
This portlet provides an overview of statistics configuration and provides access to statistic configuration page.

For each statistic, the table shows:

- *Name*: the name of the statistic.
- *Statistic type*: the type of the statistic. One of:
  - GENERIC: a generic stats calculated with data taken from your ContentWise installation.
  - ITEM: a statistic about item-data
  - USER: a statistic about user-data
- *Description*: brief description of the statistic (optional)
- *Executor task*: the Statistic generator task that will calculate the statistic during its execution
- *Actions*: you can:
  - *edit*: modifies the statistic configuration.

Bind a statistic to a recommendation algorithm.

Statistics can be defined to be used by recommendation algorithms to generate recommendation models.
Please note that only a subset of recommendation algorithms is available.



New binding

Statistic: ITEM_STATS_BY_LINEUP@CW.VIDEO@DAY,7

Algorithm: TopRateStatic

Use as default stat for selected algo: No

Bind a statistic to a recommendation algorithm

System Settings

This section describes functionalities provided by *System Settings* menu.

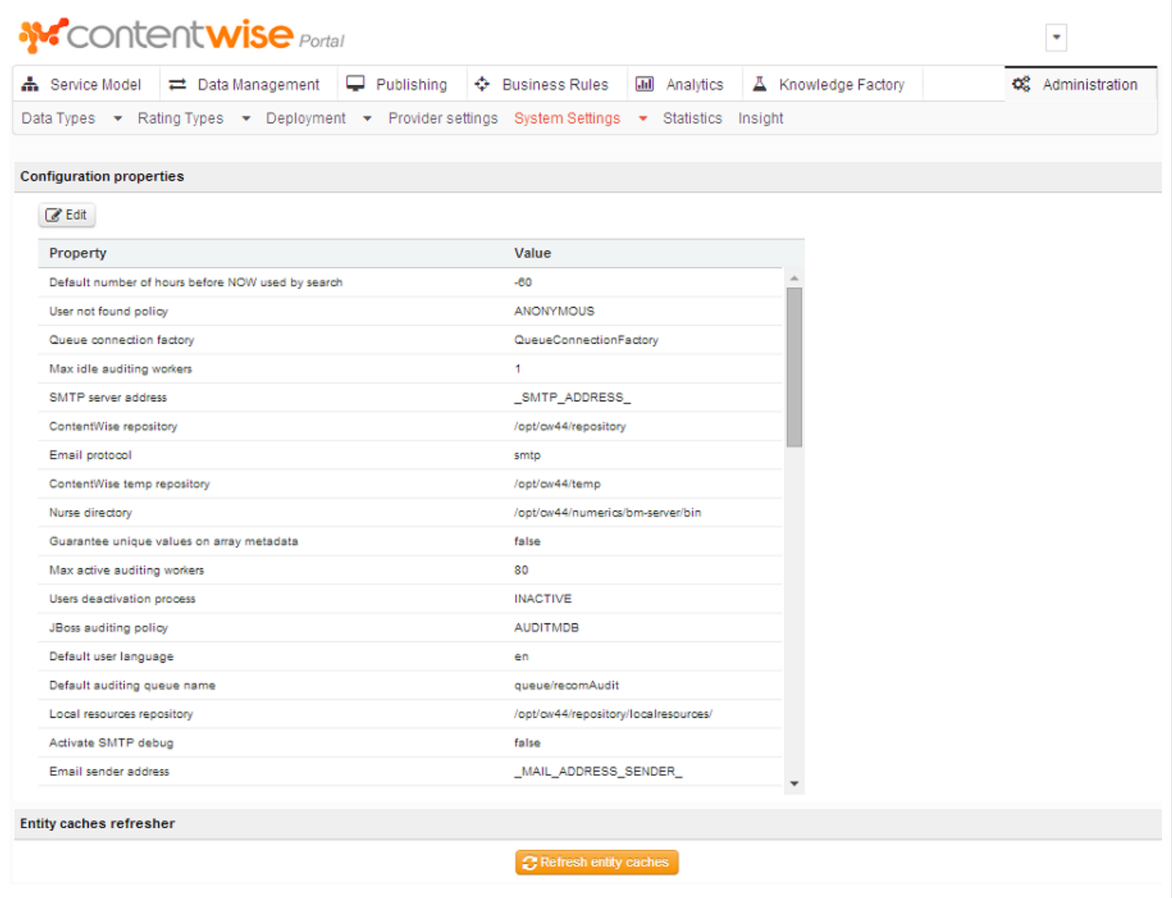
System Settings menu gives access to the following pages:

Page	Description
Configuration Properties	It is the landing page of the section. It allows to manage configuration properties defined at system level.
License	View and manage your ContentWise license key
WS Accounts	It allows to manage the web service accounts that are able to access ContentWise web services.

Configuration Properties

This section describes functionalities provided by *Configuration Properties* page.

In this page you can manage properties defined at system level. These properties are valid for all the providers available in the system.



Configuration properties

Property	Value
Default number of hours before NOW used by search	-60
User not found policy	ANONYMOUS
Queue connection factory	QueueConnectionFactory
Max idle auditing workers	1
SMTP server address	_SMTP_ADDRESS_
ContentWise repository	/opt/cw44/repository
Email protocol	smtp
ContentWise temp repository	/opt/cw44/temp
Nurse directory	/opt/cw44/numerics/bm-server/bin
Guarantee unique values on array metadata	false
Max active auditing workers	80
Users deactivation process	INACTIVE
JBoss auditing policy	AUDITMDB
Default user language	en
Default auditing queue name	queue/recomAudit
Local resources repository	/opt/cw44/repository/localresources/
Activate SMTP debug	false
Email sender address	_MAIL_ADDRESS_SENDER_

Entity caches refresher

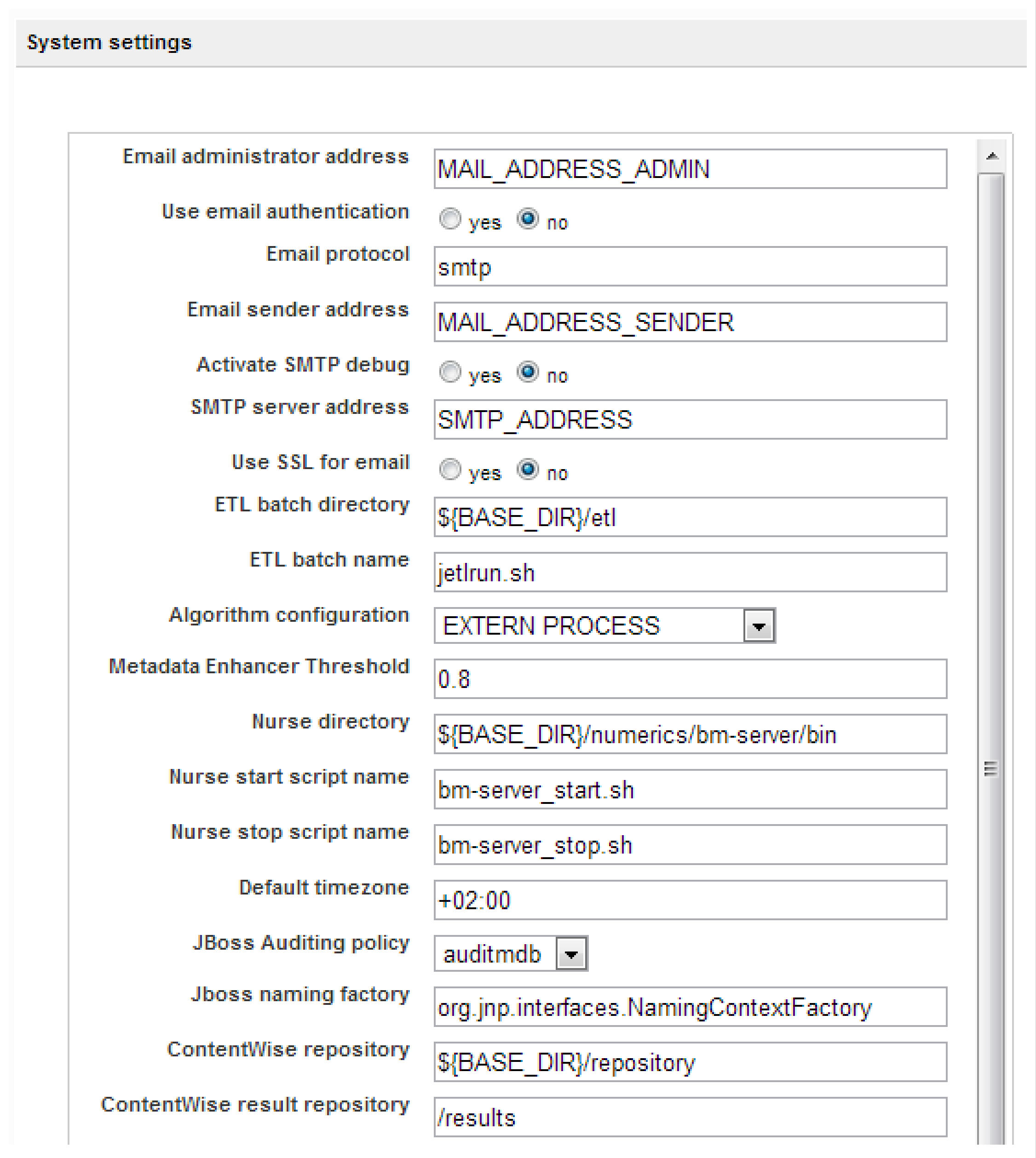
Configuration Properties page.

Configuration Properties

The *Configuration Properties* portlet lists all system settings and allow to change their configuration.

To change the configuration:

1. Click the *Edit* button.
2. Modify the settings.
3. Click *Save* button to update the configuration or click *Cancel* to undo the operation.



System settings

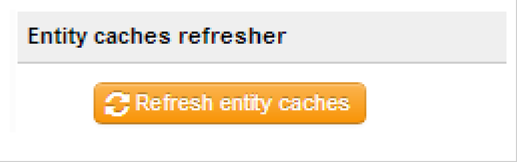
Email administrator address	MAIL_ADDRESS_ADMIN
Use email authentication	○ yes ● no
Email protocol	smtp
Email sender address	MAIL_ADDRESS_SENDER
Activate SMTP debug	○ yes ● no
SMTP server address	SMTP_ADDRESS
Use SSL for email	○ yes ● no
ETL batch directory	${BASE_DIR}/etl
ETL batch name	jetlrun.sh
Algorithm configuration	EXTERN PROCESS
Metadata Enhancer Threshold	0.8
Nurse directory	${BASE_DIR}/numerics/bm-server/bin
Nurse start script name	bm-server_start.sh
Nurse stop script name	bm-server_stop.sh
Default timezone	+02:00
JBoss Auditing policy	auditmdb
Jboss naming factory	org.jnp.interfaces.NamingContextFactory
ContentWise repository	${BASE_DIR}/repository
ContentWise result repository	/results

Configuration Properties: System settings portlet, edit mode.

Entity caches refresher

The *Entity caches refresher* portlet is a utility portlet that allows to refresh some internal caches of the system.

> **Warning**
> The *Refresh entity caches* function should never be invoked unless ContentWise support asked to do it.



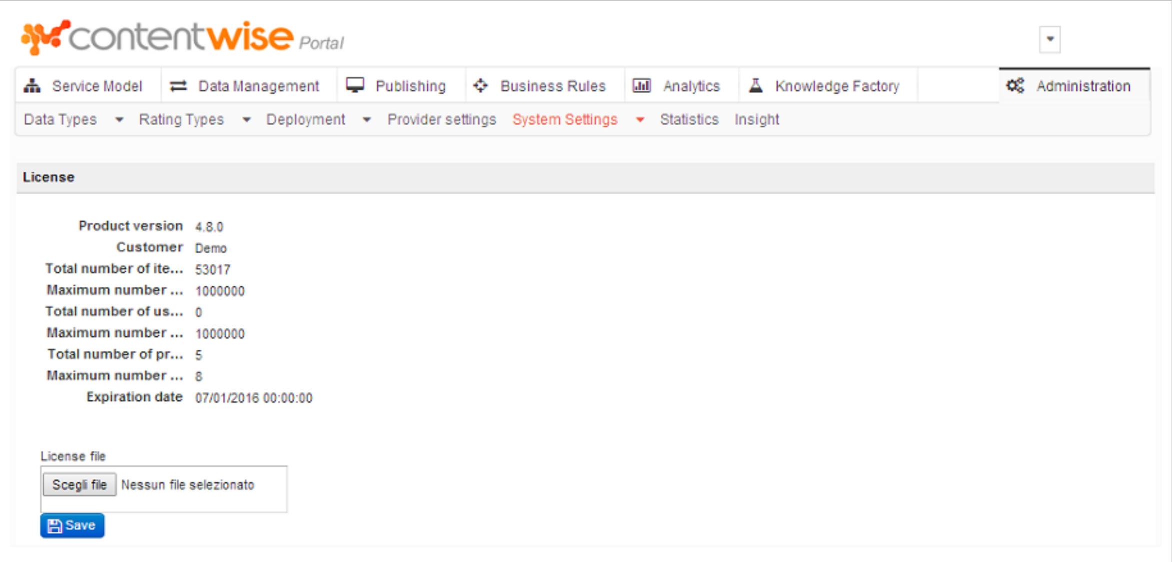
Entity caches refresher

Configuration Properties: Entity caches refresh.

License

This section describes functionalities provided by *License* page.

In this page you can manage the license of your ContentWise installation.



Product version 4.8.0
Customer Demo
Total number of ite... 53017
Maximum number ... 1000000
Total number of us... 0
Maximum number ... 1000000
Total number of pr... 5
Maximum number ... 8
Expiration date 07/01/2016 00:00:00

License file
Scegli file Nessun file selezionato
Save

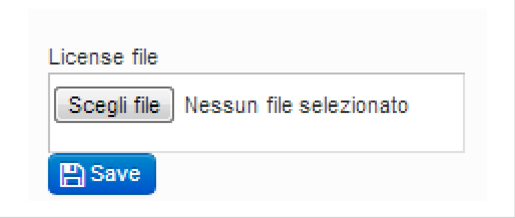
License: ContentWise license detail and management.

License

The *License* portlet shows details of your current ContentWise license and allows to update it with a new one.

To update the license:

1. Click the *Browse* button associated to *License file* field.
2. Select your ContentWise license file.
3. Click *Save* button to update the license.



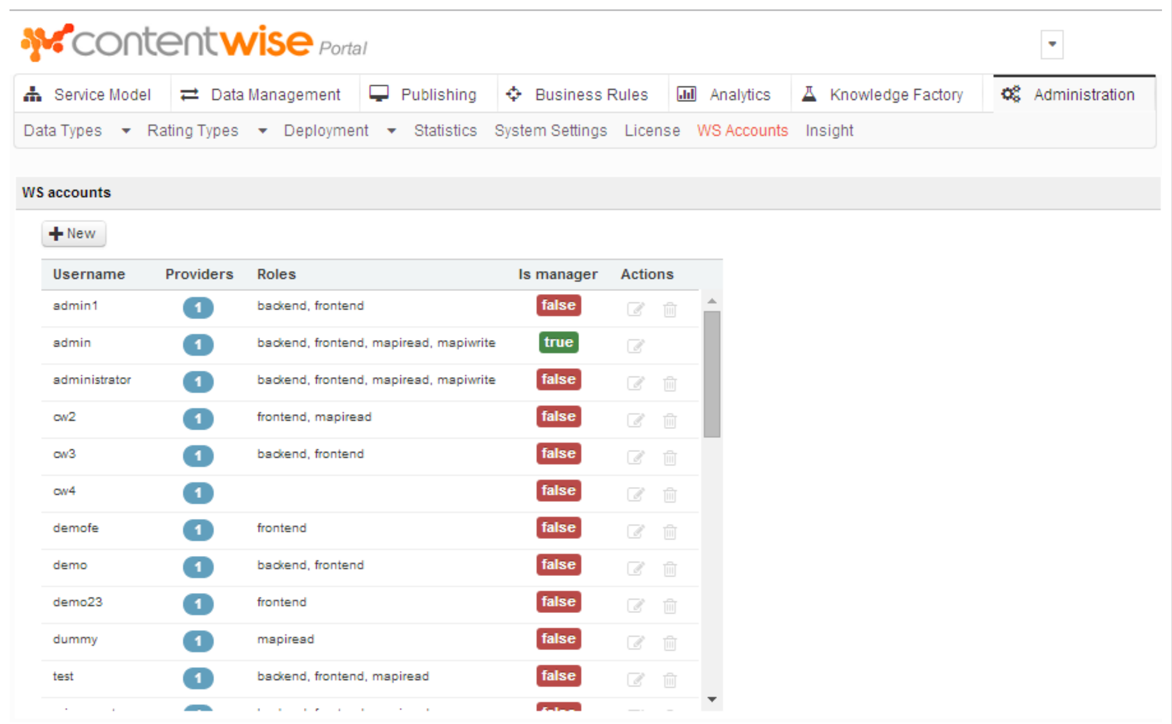
License: License portlet, update new license file.

WS Accounts

This section describes the functionalities provided by WS Accounts submenu.

In the *WS Accounts* page you can:

- have an overview of the web service accounts configured in the system.
- create new accounts and edit account configurations.



WS accounts

+ New

Username	Providers	Roles	Is manager	Actions
admin1	1	backend, frontend	false	✎ 🗑
admin	1	backend, frontend, mapiread, mapiwrite	true	✎
administrator	1	backend, frontend, mapiread, mapiwrite	false	✎ 🗑
cw2	1	frontend, mapiread	false	✎ 🗑
cw3	1	backend, frontend	false	✎ 🗑
cw4	1		false	✎ 🗑
demofe	1	frontend	false	✎ 🗑
demo	1	backend, frontend	false	✎ 🗑
demo23	1	frontend	false	✎ 🗑
dummy	1	mapiread	false	✎ 🗑
test	1	backend, frontend, mapiread	false	✎ 🗑

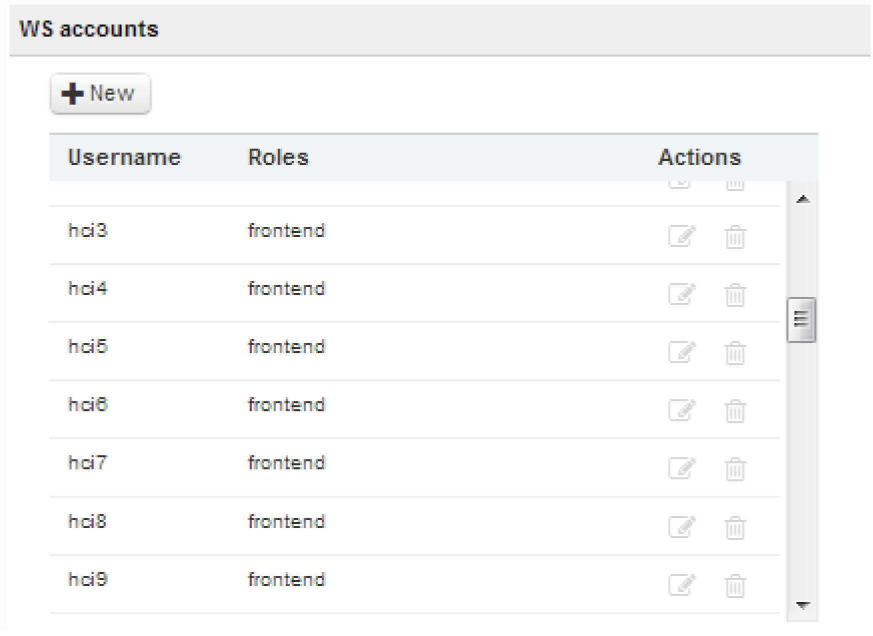
WS Accounts: list of configured standard and derived services.

WS accounts

The *WS accounts* portlet lists all available ws accounts.

For each account, the table shows:

- *Username*: the username.
- *Roles*: the list of roles enabled for the account.
- *Actions*: you can:
  - *edit*: changes username, password and associated roles.
  - *delete*: removes the account.


WS accounts

+ New

Username	Roles	Actions
hci3	frontend	✎ 🗑
hci4	frontend	✎ 🗑
hci5	frontend	✎ 🗑
hci6	frontend	✎ 🗑
hci7	frontend	✎ 🗑
hci8	frontend	✎ 🗑
hci9	frontend	✎ 🗑

WS Accounts: WS accounts portlet.

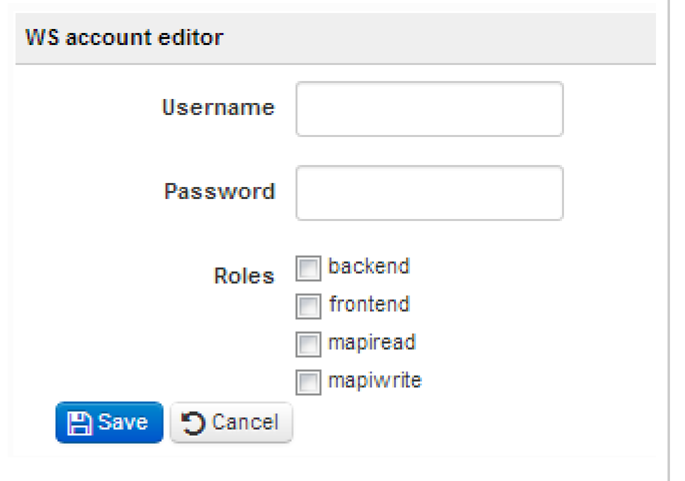
Create a ws account

To create a new ws account, click the *New* button in the WS accounts portlet.

All form information must be filled:

- *Username*: the username.
- *Password*: the password.
- *Roles*: the list of access roles of the account.

To save the account click *Save* button.
Click *Cancel* to undo the operation.



WS account editor

Username	
Password	
Roles	□ backend □ frontend □ mapiread □ mapiwrite

WS Accounts: WS account editor portlet.

Start and stop services

This section describes how to start, stop and monitor the status of ContentWise services.

Key Concept
A ContentWise service is responsible for providing a specific set of functionalities.

ContentWise services are:

- *Application Server services*: supply the *ContentWise RecServer*
- *Scheduler service*: supply the *ContentWise ProcServer*
- *Portal service*: provides the ContentWise Portal.
- *Supervisor service*: integrates the *ContentWise ProcServer* with the management of Warehouse processes.
- *Prediction scheduler service*: integrates the *ContentWise ProcServer* with the management of prediction batch processes.
- *Elastic search service*: required to supply the prediction recommendation feature.
- *Redis service*: required to supply the live events (e.g. Live TV) management

Default ContentWise services

This section describes how to start, stop and monitor the default services of a ContentWise installation.

Key Concept
The default services is a list that contains all ContentWise services that have been configured to be active in a ContentWise installation.

Configure default services

By configuring the default services of a ContentWise installation, it is easier to start, stop and monitor all of them.

The available services are:

Service	Description
jboss1	The read application server
jboss2	The write application server

scheduler	Required to execute backend activities (e.g. data processing)
portal	The administration portal service
supervisor	Required if the installation is in charge of managing the warehouse scheduler services
redis	Required if the installation handles with live events (tv or audio)
prediction scheduler	Required if the installation has to provide prediction recommendations
elasticsearch	Required if the installation has to provide prediction recommendations

To configure default services:

1. Access the base directory with the *recom* user
2. Execute the script

```
sh configure_services.sh
```

3. You will be prompted to answer y (Yes) / n (No) for each of the available services.

Start, stop and monitor default services

Operation	Command	Description
start	sh cw.sh start all	Configured services are started. The script checks if the services are already running.
status	sh cw.sh status all	Checks if configured services are running.
stop	sh cw.sh stop all	Stops configured services and the related watch-dog services.
restart	sh cw.sh restart all	Stops and starts configured services.

All ContentWise services

This section describes how to start, stop and monitor the single services of a ContentWise installation.

For instance you can stop or restart the ContentWise Portal without stopping the application server.

Application Server services

Operation	Command	Description
start	sh cw.sh start jboss	ContentWise recommendation service is started. The script checks if the service is already running.
status	sh cw.sh status jboss	Checks if the recommendation service is running.
stop	sh cw.sh stop jboss	Stops the recommendation service and the related watch-dog service.
restart	sh cw.sh restart jboss	Stops and starts the recommendation service.

Important note
It is possible to start, stop and monitor a single Application Server. Use the following parameters

- **jboss1** or **jbossread**, to manage the read application server
- **jboss2** or **jbosswrite**, to manage the write application server

Scheduler service

Operation	Command	Description
start	sh cw.sh start scheduler	Scheduler service is started. The script checks if the service is already running.
status	sh cw.sh status scheduler	Checks if the scheduler service is running.
stop	sh cw.sh stop scheduler	Stops the scheduler service and the related watch-dog service.
restart	sh cw.sh restart scheduler	Stops and starts the scheduler service.

Portal service

Operation	Command	Description
start	sh cw.sh start portal	Portal service is started. The script checks if the service is already running.
status	sh cw.sh status portal	Checks if the portal service is running.
stop	sh cw.sh stop portal	Stops the portal service and the related watch-dog service.
restart	sh cw.sh restart portal	Stops and starts the portal service.

Supervisor service

Operation	Command	Description
start	sh cw.sh start supervisor	Supervisor service is started. The script checks if the service is already running.
status	sh cw.sh status supervisor	Checks if the supervisor service is running.
stop	sh cw.sh stop supervisor	Stops the supervisor service and the related watch-dog service.
restart	sh cw.sh restart supervisor	Stops and starts the supervisor service.

Prediction scheduler service

Operation	Command	Description
start	sh cw.sh start pscheduler	Prediction scheduler service is started. The script checks if the service is already running.
status	sh cw.sh status pscheduler	Checks if the prediction scheduler service is running.
stop	sh cw.sh stop pscheduler	Stops the prediction scheduler service and the related watch-dog service.
restart	sh cw.sh restart pscheduler	Stops and starts the prediction scheduler service.

Elastic-search service

Operation	Command	Description
start	sh cw.sh start elasticsearch	Elastic-search service is started. The script checks if the service is already running.
status	sh cw.sh status elasticsearch	Checks if the elastic-search service is running.
stop	sh cw.sh stop elasticsearch	Stops the elastic-search service and the related watch-dog service.
restart	sh cw.sh restart elasticsearch	Stops and starts the elastic-search service.

Redis service

Operation	Command	Description
start	sh cw.sh start redis	Redis service is started. The script checks if the service is already running.
status	sh cw.sh status redis	Checks if the redis service is running.
stop	sh cw.sh stop redis	Stops the redis service and the related watch-dog service.
restart	sh cw.sh restart redis	Stops and starts the redis service.

Watchdog

Watchdog service monitors the following events:

- `java.lang.OutOfMemoryError` is thrown and logged into monitored log files.
- service is killed or dies unexpectedly.

Metadata reference

Important Note
This is a reference of standard metadata. ContentWise can additionally import any custom metadata.

Important Note
Metadata with the *Array* suffix are interpreted as a list of elements (separated by the character specified in the ETL configuration).

Default import policy
In the case you import a metadata that already exists for the associated item, two possible policies can be applied:

- **CONCAT**: the new value is added to the value previously stored
- **REPLACE**: the new value overwrites the value previously stored

This option can be configured for each metadata of each itemtype.
By default, CONCAT policy is applied in the case of *Array* metadata, REPLACE policy otherwise.

- Identifiers
- Item metadata
  - Metadata for type VIDEO_CONTENT
  - Metadata for type VIDEO_PROGRAM
  - Metadata for type VIDEO_CHANNEL
  - Metadata for type VIDEO_CATCHUP
  - Metadata for type AUDIO_CONTENT
  - Metadata for type AUDIO_PROGRAM
  - Metadata for type AUDIO_CHANNEL
  - Metadata for type GENERIC_CONTENT
  - Metadata for type WEBPAGE_CONTENT
  - Metadata for type BOOK_CONTENT
  - Metadata for type PEOPLE
- User metadata
  - Metadata for type PERSON
  - Metadata for type TERMINAL
- User Preference metadata
- Access metadata

Identifiers

Identifiers are special metadata used to univocally identify an imported object.
We will here use the terms defined in chapter Terminology to refer to all the basic objects in ContentWise.

As explained in chapter Terminology, items and users are linked to an external identification used by a provider. For example, an ETL is importing items and it says that a specific item is called "20002" into the
datasource. Hence, you have:

- provitemid: is the unique identifier of an item within a provider (PROVITEMID)
- provuserid: is the unique identifier of a user within a provider (PROVUSERID)

An important information in this scenario is also the *type* of the item or user. Hence, you have:

- itemtype: is the type of an item
- usertype: is the type of a user

An item is unique if identified by the triple (providerid, itemtype, provitemid); i.e., the provitemid must be unique within a certain provider and itemtype.
The same is for users. Ratings are identified by the couple of identifiers of the user that is rating and of the rated item, hence the unique identifier is (providerid, itemtype, provitemid, usertype, provuserid).

Item metadata

ContentWise manages item metadata, that are used for different purposes:

1. Generate recommendations
2. Apply pre/post and dynamic filtering by defining specific criteria on metadata
3. Generate reporting information

The set of metadata that is interpreted to generate recommendations is called *well-known* metadata set, that is a subset of all available metadata.

ContentWise can import any custom metadata for the purposes 2 and 3, while the well-known metadata are the only used for the purpose 1. However, a specific installation can miss some well known metadata without being compromised; the more well-known metadata you give to the recommendation engine, the more accurate is the recommendation.

> ⚠ Metadata should not contain duplicated values. If a content is created with duplicated metadata values, they will not be merged; however if a content is updated providing duplicated values they are removed.
>
> This behavior will be fixed in future versions.

> ℹ **Array metadata values and separator**
> Metadata that contain multiple values (e.g. the cast of a movie) are characterized by the **Array** suffix (e.g. ActorsLastNameFirstArray)
> Array metadata values must be separated each other. Default separator is # (sharp) character.
> Array metadata values must start and end with the separator. Example: #Cruise Tom#Roberts Julia#

This section lists the standard metadata for each itemtype.

- Metadata for type VIDEO_CONTENT
- Metadata for type VIDEO_PROGRAM
- Metadata for type VIDEO_CHANNEL
- Metadata for type VIDEO_CATCHUP
- Metadata for type AUDIO_CONTENT
- Metadata for type AUDIO_PROGRAM
- Metadata for type AUDIO_CHANNEL
- Metadata for type GENERIC_CONTENT
- Metadata for type WEBPAGE_CONTENT
- Metadata for type BOOK_CONTENT

Metadata for type VIDEO_CONTENT

Metadata	Description
ActorsLastNameFirstArray	A list of actors of the item
AudiencesArray	Indicates the target audiences (e.g.: "Mature", "Adult", etc.)
AvailableInPackagesArray	The list of packages in which the item is available
CategoriesArray	A list of categories of the item for a specific language
CategoriesCrxArray	A list of categories of the item
CensureArray	Item censure identificators
CountryOfOrigin	The country of origin of the item
DirectorsLastNameFirstArray	A list of directors of the item
EpisodeID	A unique identifier for an episode of a series
EpisodeName	The name of an individual episode of a series
Format	Item format.
GenresArray	List of genres of the item for a specific language
GenresCrxArray	List of genres of the item
IsAlive	Indicates if the item is currently available
KeywordsArray	A list of keywords
LanguageOriginal	The original language for the event (2 digits, format ISO 639)
LanguagesArray	The languages of the event (each language in 2 digits, format ISO 639)
MdLanguage	The language of item metadata (2 digits, format ISO 639)
PriceCategory	Item price category identificator
ProducersLastNameFirstArray	A list of producers of the item
RunTime	The duration of the event (in minutes)
SeriesTitle	The series title

SeriesTitleOriginal	The series title in its original language
ShowType	Indicates the type of the event
StudioNamesArray	The name of the production studios
SubGenresArray	List of sub-genres of the item for a specific language
SubGenresCrxArray	List of sub-genres of the item
SummaryLong	The item description or plot
TitleFull	The title of the item
TitleFullOriginal	The item title in its original language
Year	The year of release

Metadata for type VIDEO_PROGRAM

Metadata	Description
ActorsLastNameFirstArray	A list of actors of the item
AudiencesArray	Indicates the target audiences (e.g.: "Mature", "Adult", etc.)
AvailableInPackagesArray	The list of packages in which the item is available
CategoriesArray	A list of categories of the item for a specific language
CategoriesCrxArray	A list of categories of the item
CensureArray	Item censure identificators
CountryOfOrigin	The country of origin of the item
DirectorsLastNameFirstArray	A list of directors of the item
EpisodeID	A unique identifier for an episode of a series
EpisodeName	The name of an individual episode of a series
Format	Item format.
GenresArray	List of genres of the item for a specific language
GenresCrxArray	List of genres of the item
KeywordsArray	A list of keywords for a specific language
LanguagesArray	The languages of the item (each language in 2 digits, format ISO 639)
MdLanguage	The language of item metadata (2 digits, format ISO 639)
PriceCategory	Item price category identificator
ProducersLastNameFirstArray	A list of the producers of the item
ProgramChannelID	A unique identifier for program's channel
ProgramEnd	The end timestamp of the program (ISO format)
ProgramEndOffset	The end offset of the program in the following format [+|-]HH:MM
ProgramStart	The start timestamp of the program (ISO format)
ProgramStartOffset	The start offset of the program in the following format [+|-]HH:MM
RunTime	The duration of the item (minutes)
SeriesTitle	The series title
SeriesTitleOriginal	The series title in its original language
ShowType	Indicates the type of the event
StudioNamesArray	The name of the production studios
SubGenresArray	List of sub-genres of the item for a specific language
SubGenresCrxArray	List of sub-genres of the item

SummaryLong	The item description or plot
TitleFull	The title of the item
TitleFullOriginal	The title of the item in its original language
Year	The year of release

Metadata for type VIDEO_CHANNEL

Metadata	Description
AudiencesArray	Indicates the target audiences (e.g.: "Mature", "Adult", etc.)
AvailableInPackagesArray	The list of packages in which the item is available
CategoriesArray	List of categories of the item for a specific language
CategoriesCrxArray	List of categories of the item
CensureArray	Item censure identificators
ChannelNumber	Number of the channel
ChannelIP	IP of channel
ChannelName	The name of the channel
CountryOfOrigin	The country of origin of the item
Format	Item format.
GenresArray	List of genres of the item for a specific language
GenresCrxArray	List of genres of the item
HeadendsArray	List of headends of the channel represented by the item
IsAlive	Indicates if the item is currently available
KeywordsArray	A list of keywords
LanguageOriginal	The original language of the item (2 digits, format ISO 639)
LanguagesArray	The languages of the item (each language in 2 digits, format ISO 639)
MdLanguage	The language of item metadata (2 digits, format ISO 639)
PriceCategory	Item price category identificator
ShowType	Indicates the type of the event
SubGenresArray	List of sub-genres of the item for a specific language
SubGenresCrxArray	List of sub-genres of the item
SummaryLong	The item description or plot
TitleFull	The title of the item
TitleFullOriginal	The title of the item in its original language

Metadata for type VIDEO_CATCHUP

Metadata	Description
ActorsLastNameFirstArray	
AudiencesArray	
AvailableInPackagesArray	
CategoriesArray	
CategoriesCrxArray	
CensureArray	
CountryOfOrigin	

DeduplicationID	
DirectorsLastNameFirstArray	
EndLicenseWindow	
EpisodeID	
EpisodeName	
Format	
GenresArray	
GenresCrxArray	
KeywordsArray	
LanguageOriginal	
LanguagesArray	
LicenseWindowTimeOffset	
MdLanguage	
PriceCategory	
ProducersLastNameFirstArray	
ProgramChannelID	
RunTime	
SeriesSummaryLong	
SeriesTitle	
SeriesTitleOriginal	
ShowType	
StartLicenseWindow	
StudioNamesArray	
SubGenresArray	
SubGenresCrxArray	
SummaryLong	
TitleFull	
TitleFullOriginal	
Year	

Metadata for type AUDIO_CONTENT

Metadata	Description
ArtistLastNameFirstArray	A list of artists of the content
AudiencesArray	Indicates the target audiences (e.g.: "Mature", "Adult", etc.)
AvailableInPackagesArray	The list of packages in which the item is available
CategoriesArray	List of categories of the item for a specific language
CategoriesCrxArray	List of categories of the item
CensureArry	Item censure identificators
ComposersLastNameFirstArray	A list of composers of the item
CountryOfOrigin	The country of origin of the item
Format	Item format.
GenresArray	List of genres of the item for a specific language

GenresCrxArray	List of genres of the item
InterpretersLastNameFirstArray	A list of interpreters of the item
IsAlive	Indicates if the item is currently available
KeywordsArray	A list of keywords
LanguageOriginal	The original language of the item (2 digits, format ISO 639)
LanguagesArray	The languages of the event (each language in 2 digits, format ISO 639)
MdLanguage	The language of item metadata (2 digits, format ISO 639)
PriceCategory	Item price category identificator
Publisher	The Publisher of the item
RunTime	The duration of the item
ShowType	Indicates the type of the event
SubGenresArray	List of sub-genres of the item for a specific language
SubGenresCrxArray	List of sub-genres of the item
SummaryLong	The item description or plot
TitleFull	The title of the item
TitleFullOriginal	The title of the item in its original language
Year	The year of release

Metadata for type AUDIO_PROGRAM

Metadata	Description
ArtistLastNameFirstArray	A list of artists of the content
AudiencesArray	Indicates the target audiences (e.g.: "Mature", "Adult", etc.)
AvailableInPackagesArray	The list of packages in which the item is available
CategoriesArray	List of categories of the item for a specific language
CategoriesCrxArray	List of categories of the item
CensureArray	Item censure identificators
ComposersLastNameFirstArray	A list of composers of the item
CountryOfOrigin	The country of origin of the item
Format	Item format.
GenresArray	List of genres of the item for a specific language
GenresCrxArray	List of genres of the item
InterpretersLastNameFirstArray	A list of interpreters of the item
KeywordsArray	A list of keywords
LanguageOriginal	The original language of the item (2 digits, format ISO 639)
LanguagesArray	The languages of the event (each language in 2 digits, format ISO 639)
MdLanguage	The language of item metadata (2 digits, format ISO 639)
PriceCategory	Item price category identificator
ProgramChannelID	A unique identifier for program's channel
ProgramEnd	The end timestamp of the program (ISO format)
ProgramEndOffset	The end offset of the program in the following format [+|-]HH:MM
ProgramStart	The start timestamp of the program (ISO format)
ProgramStartOffset	The start offset of the program in the following format [+|-]HH:MM

Publisher	The Publisher of the item
RunTime	The duration of the item (in minutes)
ShowType	Indicates the type of the event
SubGenresArray	List of sub-genres of the item for a specific language
SubGenresCrxArray	List of sub-genres of the item
SummaryLong	The item description or plot
TitleFull	The title of the item
TitleFullOriginal	The title of the item in its original language
Year	The year of release

Metadata for type AUDIO_CHANNEL

Metadata	Description
AudiencesArray	Indicates the target audiences (e.g.: "Mature", "Adult", etc.)
AvailableInPackagesArray	The list of packages in which the item is available
CategoriesArray	List of categories of the item for a specific language
CategoriesCrxArray	List of categories of the item
CensureArray	Item censure identificatora
ChannelIP	IP of channel
ChannelName	The name of the channel
CountryOfOrigin	The country of origin of the item
Format	Item format.
GenresArray	List of genres of the item for a specific language
GenresCrxArray	List of genres of the item
IsAlive	Indicates if the item is currently available
KeywordsArray	A list of keywords
LanguageOriginal	The original language of the event (2 digits, format ISO 639)
LanguagesArray	The languages of the event (each language in 2 digits, format ISO 639)
MdLanguage	The language of metadata content (2 digits, format ISO 639)
PriceCategory	Item price category identificator
ShowType	Indicates the type of the event
SubGenresArray	List of sub-genres of the item for a specific language
SubGenresCrxArray	List of sub-genres of the item
SummaryLong	The item description or plot
TitleFull	The title of the item
TitleFullOriginal	The title of the item in its original language
Year	The year of release

Metadata for type GENERIC_CONTENT

Metadata	Description
AudiencesArray	Indicates the target audiences (e.g.: "Mature", "Adult", etc.)
AvailableInPackagesArray	The list of packages in which the item is available
CategoriesArray	List of categories of the item for a specific language

CategoriesCrxArray	List of categories of the item
CensureArray	Item censure identificators
DeduplicationID	
Format	Item format.
IsAlive	Indicates if the item is currently available
KeywordsArray	A list of keywords
LanguagesArray	The languages of the event (each language in 2 digits, format ISO 639)
MdLanguage	The language of metadata content (2 digits, format ISO 639)
PriceCategory	Item price category identificator
SubCategoriesArray	List of sub-categories of the item for a specific language
SubCategoriesCrxArray	List of sub-categories of the item
SummaryLong	The item description or plot
TitleFullOriginal	The title of the item in its original language

Metadata for type WEBPAGE_CONTENT

Metadata	Description
AudiencesArray	Indicates the target audiences (e.g.: "Mature", "Adult", etc.)
BannersArray	
ContentCategoriesArray	List of categories of the item for a specific language
ContentSubCategoriesArray	List of sub-categories of the item for a specific language
ContentCategoriesCrxArray	List of categories of the item
ContentSubCategoriesCrxArray	List of sub-categories of the item
Format	Item format.
IsAlive	Indicates if the item is currently available
KeywordsArray	A list of keywords
LanguagesArray	The languages of the event (each language in 2 digits, format ISO 639)
LinkArray	
MdLanguage	The language of metadata content (2 digits, format ISO 639)
PageBannerUrl	
PageBody	
PageTreeArray	
PageTechnology	
PageUrl	
Section	
TitleFull	The title of the item

Metadata for type BOOK_CONTENT

Metadata	Description
AuthorsLastNameFirstArray	List of book's authors
AudiencesArray	Indicates the target audiences (e.g.: "Mature", "Adult", etc.)
AvailableInPackagesArray	The list of packages in which the item is available
CategoriesArray	A list of categories of the item for a specific language

CategoriesCrxArray	A list of categories of the item
CensureArray	Item censure identificators
CollectionTitle	The collection title
CollectionTitleOriginal	The collection title in its original language
ContributorsLastNameFirstArray	List of book's contributors
CountryOfOrigin	The country of origin of the item
Format	Item format.
GenresArray	List of genres of the item for a specific language
GenresCrxArray	List of genres of the item
IsIllustrated	Indicates if book has illustrations
KeywordsArray	A list of keywords
LanguageOriginal	The original language for the event (2 digits, format ISO 639)
LanguagesArray	The languages of the event (each language in 2 digits, format ISO 639)
MdLanguage	The language of metadata content (2 digits, format ISO 639)
NumberOfPages	Number of pages of the book
PublishersArray	List of publishers of the book
SubCategoriesArray	List of sub-categories of the item for a specific language
SubCategoriesCrxArray	List of sub-categories of the item
SubGenresArray	List of sub-genres of the item for a specific language
SubGenresCrxArray	List of sub-genres of the item
SubjectsArray	List of subjects (keywords) of the book
SuppliersArray	List of suppliers of the book
SummaryLong	The item description or plot
TitleFull	The title of the book
TitleFullOriginal	The item title in its original language
Year	The year of release

Metadata for type PEOPLE

AliasArray	List of alias of the person
BirthDate	Birth date
CategoriesCrxArray	List of categories
CountryOfOrigin	Country of origin
FirstName	Name
LastNameFirst	Name and surname
LastName	Surname
IsAlive	Indicates if the item is currently available
KeywordsArray	List of keywords
MdLanguage	The language of metadata content (2 digits, format ISO 639)
Role	Role of the person (Actor, Writer, ...)
SummaryLong	The item description

User metadata

- Metadata for type PERSON
- Metadata for type TERMINAL

Metadata for type PERSON

Metadata	Description
Age	Age
BirthYear	Birth year
Gender	Gender
HeadendAutoGroupArray	List of headends available to the user
SubscribedPackagesArray	List of packages the user is subscribed to
UserCountry	Coutry or region of the user
UserLanguagesArray	List of user languages
UserPrefMdLanguage	User preferred metadata language

Metadata for type TERMINAL

Metadata	Description
HeadendAutoGroupArray	List of headends available to the user
SubscribedPackagesArray	List of packages the user is subscribed to
Technology	Technology used by user
TerminalIP	IP of user's terminal
TimeOffset	Time offset of the terminal in the following format [+|-]HH:MM
UserCountry	Coutry or region of the user
UserLanguagesArray	List of user languages
UserPrefMdLanguage	User preferred metadata language

User Preference metadata

> **Important Note**
> User Preference metadata are defined for both TERMINAL and PERSON usertypes.

Metadata	Description
PrefActorsLastNameFirstArray	Preferred actors list
PrefCategoriesArray	Preferred categories list
PrefComposersLastNameFirstArray	Preferred composers list
PrefDirectorsLastNameFirstArray	Preferred directors list
PrefFormatArray	Preferred format of the items (e.g. HD)
PrefGenresArray	Preferred genres list
PrefSummaryLongArray	Preferred description words
PrefInterpretersLastNameFirstArray	Preferred interpreters list
PrefSubGenresArray	Preferred sub-genres list
PrefAuthorsLastNameFirstArray	Preferred authors list

Access metadata

Metadata	Description

Accessed	Specifies if the item has been accessed ('0'=no , '1'=yes)
Caller	Specifies the caller from which the access has been done
Channel	Identifies the channel of the access
CwCallId	The call identifier. It is returned by a recommendation API and must be set as access metadata to calculate direct effectiveness
Duration	Item access duration (seconds)
PlayTime	Total time of content play (seconds)
Purchased	Specifies if the item has been purchased ('0'=no , '1'=yes)
RatingExpl	User explicit rating
TimestampEnd	Item access finish timestamp (ISO Format)
TimestampEndOffset	Item access finish offset in the following format [+|-]HH:MM
TimestampStartOffset	Item access start offset in the following format [+|-]HH:MM
Viewed	Specifies if the item has been viewed ('0'=no , '1'=yes)
VisionFactor	Percentage of Runtime item viewed
UserComment	User comment about the item

Supported formats

This chapter lists the data formats that are supported by ContentWise for importing data in the system.

> **Important Note**
> This is a list of already supported data formats. Custom ContentWise ETLs can be developed to additionally import any custom format.

ContentWise supports the following standard formats:

Format	Reference
CableLabs	CableLabs VOD Metadata - VOD Content Specification 2.0 http://www.cablelabs.com Document ID: MD-SP-VOD-CONTENT2.0-I02-070105
TV-Anytime	http://www.tv-anytime.org/
Tribune	http://www.tribunemediaservices.com - Television and Movies
XMLTV	http://wiki.xmltv.org/index.php/XMLTVFormat - XMLTV Format
Onix 3.0	http://www.editeur.org/12/About-Release-3.0/ - Onix Books format

ContentWise XML format

This chapter describes the XML format used by ContentWise (in addition to other standard or custom formats) in the Data Import Interface to import data from external systems.

The ContentWise XML format can be used for items, users and item accesses.
To distinguish the three types, the tag "item", "user", "itemaccess" is used. The most important attribute of the three types is the unique identifier "id" of the object. For the itemaccess document, two attributes have to be specified: "itemid" and "userid".

The structure of the document for an item is the following:

```
<contentwisexml version="1.0">
<item ds_provitemid="12345" type="VIDEO_CONTENT">
....item elements....
</item>
</contentwisexml>
```

For a user, we have:

```
<contentwisexml version="1.0">
<user ds_provuserid="12345" type="TERMINAL">
....user elements....
</user>
</contentwisexml>
```

For an itemaccess, we have:

```
<contentwisexml version="1.0">
<itemaccess ds_provuserid="12345" ds_provitemid="1100"
                 usertype="TERMINAL" itemtype="VIDEO_CONTENT">
....itemaccess elements....
</itemaccess>
</contentwisexml>
```

Item XML Elements

The elements that can be used in the item description are exactly the same as the metadata reported in Reference Guide, Appendix A. Hence, for example, you can have:

```
<contentwisexml version="1.0">
<item ds_provitemid="12345" type="VIDEO_CONTENT">
  <TitleFull>The Godfather</TitleFull>
</item>
</contentwisexml>
```

If the metadata is of Array type, the XML must contain a sequence of $<value>$ elements:

```
<contentwisexml version="1.0">
<item ds_provitemid="12345" type="VIDEO_CONTENT">
 <TitleFull>The Godfather</TitleFull>
 <GenresArray><value>Drama</value><value>Crime</value></GenresArray>
</item>
</contentwisexml>
```

The XML can contain custom metadata to describe extra item features, using the following format:

- Item metadata specific for a provider (up to three): in this case you have to use the elements ipmdA, ipmdB, ipmdC
- Item metadata that are common for all providers (up to five): in this case you have to use the elements imdA, imdB, imdC, imdD, imdE

For example, you can have:

```
<contentwisexml version="1.0">
<item ds_provitemid="12345" type="VIDEO_CONTENT">
  <ipmdA name="mycustp">value1</ipmdA>
  <imdA name="mycust">value2</imdA>
</item>
</contentwisexml>
```

These extra elements will not be used for recommendation purposes, but only for reporting.

User XML Elements

The elements that can be used in the user description are exactly the same as the metadata reported in Reference Guide. Hence, for example, you can have:

```
<contentwisexml version="1.0">
<user ds_provuserid="12345" type="TERMINAL">
 <TerminalIP>223.212.12.2</TerminalIP>
</user>
</contentwisexml>
```

If the metadata is of Array type, the XML must contain a sequence of `<value>` elements, as described above.
The XML can contain custom metadata to describe extra user features, using the following format:

- User metadata specific for a provider (up to three): in this case you have to use the elements upmdA, upmdB, upmdC
- User metadata that are common for all providers (up to five): in this case you have to use the elements umdA, umdB, umdC, umdD, umdE

For example, you can have:

```
<contentwisexml version="1.0">
<user ds_provuserid="12345" type="TERMINAL">
 <upmdA name="mycustp">value1</upmdA>
 <umdA name="mycust">value2</umdA>
</user>
</contentwisexml>
```

These extra elements will not be used for recommendation purposes, but only for reporting.

Item Access XML Elements

The elements that can be used in the Item Access description are exactly the same as the metadata reported in Reference Guide.
Hence, for example, you can have:

```
<contentwisexml version="1.0">
<itemaccess ds_provuserid="12345" usertype="TERMINAL"
 ds_provitemid="4455" itemtype="VIDEO_CONTENT">
 <Viewed>1</Viewed>
</itemaccess>
</contentwisexml>
```

If the metadata is of Array type, the XML must contain a sequence of `<value>` elements, as described above.

Filename format

For the integration of XML files, a specific file name and internal format is adopted.
The file name must respect the following rule
`<type><id><updatedate>.xml`
Where:

- `<type>` is one of ITEM, USER, ITEMACCESS
- `<id>` is the id of the item if type=ITEM, the id of the user if type=USER, the string "userid:itemid" if type=ITEMACCESS
- `<updatedate>` is the date of insert/update of the item, the user or the item access; the format for the date is YYYYMMDDHHMISS

For example:
`ITEM_AAA01123_20070101140030.xml`
Is a file containing information for the update of the item AAA01123 occurred at 2007-01-01 14:00:30

For the file content, the format can be:

- For ITEM files: CableLabs XML or ContentWise XML
- For USER: ContentWise XML
- For ITEMACCESS: ContentWise XML

TreeNavigation

Other product versions

Home

Other product versions

ContentWise v.5.5 guides

- Reference Guide
- Installation Guide
- API Guide

ContentWise v.5.4 guides

- Reference Guide
- Installation Guide
- API Guide

ContentWise v.5.3 guides

- Reference Guide
- Installation Guide
- API Guide

ContentWise v.5.2 guides

- Reference Guide
- Installation Guide
- API Guide

ContentWise v.5.1 guides

- Reference Guide
- Installation Guide
- API Guide

ContentWise v.5.0 guides

- Reference Guide
- Installation Guide
- API Guide

ContentWise v.4.8 guides

- Reference Guide
- Installation Guide
- API Guide

ContentWise v.4.6 guides

- Reference Guide
- Installation Guide
- API Guide

ContentWise v.4.4 guides

- Reference Guide

- Installation Guide
- API Guide

ContentWise v.4.3 guides

- Reference Guide
- Installation Guide
- API Guide

ContentWise v.4.2 guides

- Reference Guide
- Installation Guide
- API Guide

ContentWise v.4.1 guides

- Reference Guide
- Installation Guide
- API Guide

ContentWise v.4.0 guides

- Reference Guide
- Installation Guide
- API Guide

ContentWise v.3.6 guides

- Reference Guide
- Installation Guide
- API Guide

ContentWise v.3.5 guides

- Reference Guide
- Installation Guide
- API Guide

ContentWise v.3.1 guides

- Reference Guide
- Installation Guide
- API Guide

ContentWise v.2.X guides

- Reference Guide
- Installation Guide
- Administration Guide
- API Guide

Caller Groups

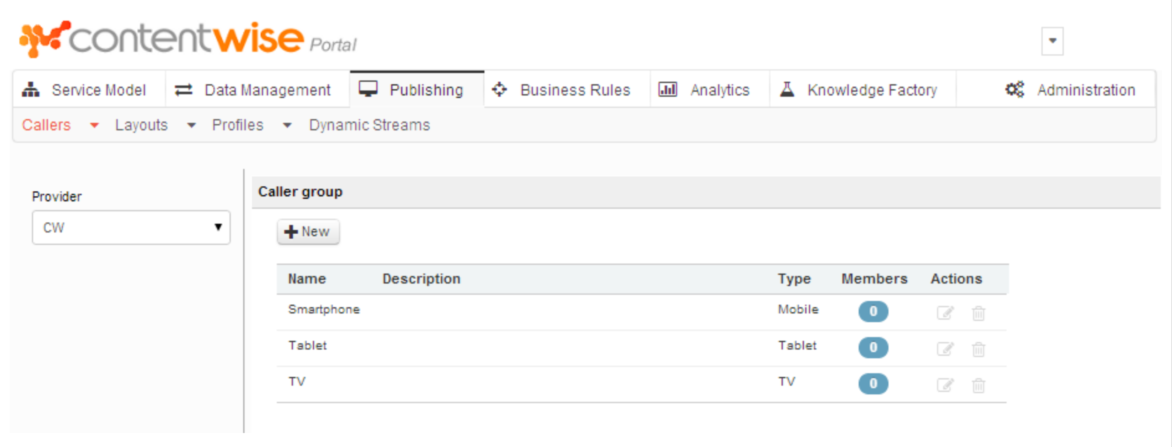
This page provides an overview of caller configurations and provides access to the pages that allow to create and manage caller groups.

The page contains the following portlets:

- Caller Groups: lists the caller groups that have been defined and provides access to new/edit caller groups modals.

See create a new caller group to create a new caller group.

If a provider filter is not selected, you will be prompted to select one.



Name	Description	Type	Members	Actions
Smartphone		Mobile	0	
Tablet		Tablet	0	
TV		TV	0	

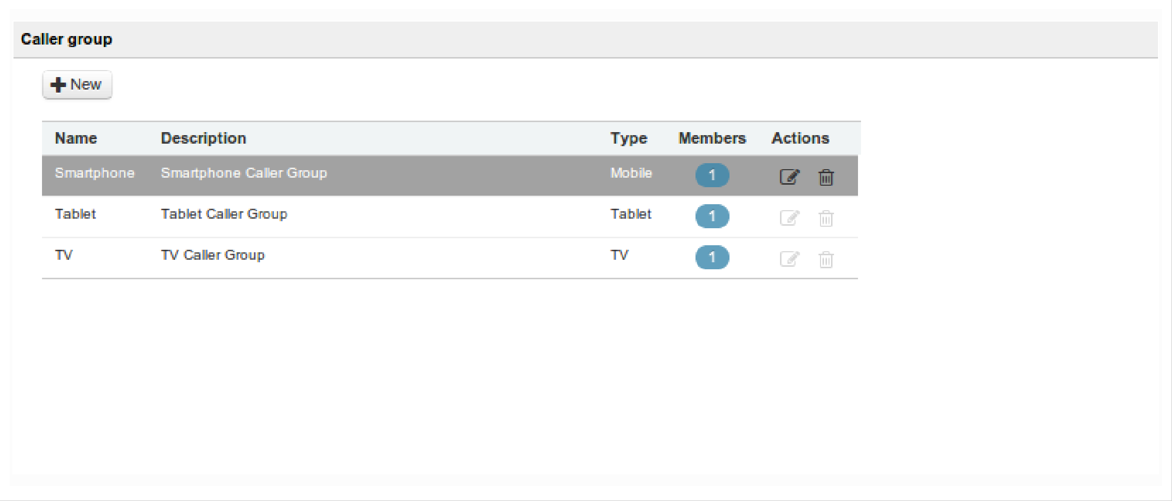
Caller Groups: list of configured caller groups and related configurations.

Caller Groups

This portlet provides an overview of caller group configuration and provides access to caller configuration modals.

For each caller groups, the table shows:
For each caller, the table shows:

- *Name*: unique caller group identifier.
- *Description*: a short caller group description.
- *Type*: the caller group type.
- *Members*: the caller group members. A member is a caller which belongs to the caller group.
- *Actions*: you can:
    - *edit*: modifies the caller group configuration.
    - *delete*: remove a caller group.



Name	Description	Type	Members	Actions
Smartphone	Smartphone Caller Group	Mobile	1	
Tablet	Tablet Caller Group	Tablet	1	
TV	TV Caller Group	TV	1	

Caller Groups: Caller groups portlet.

Create a new caller group

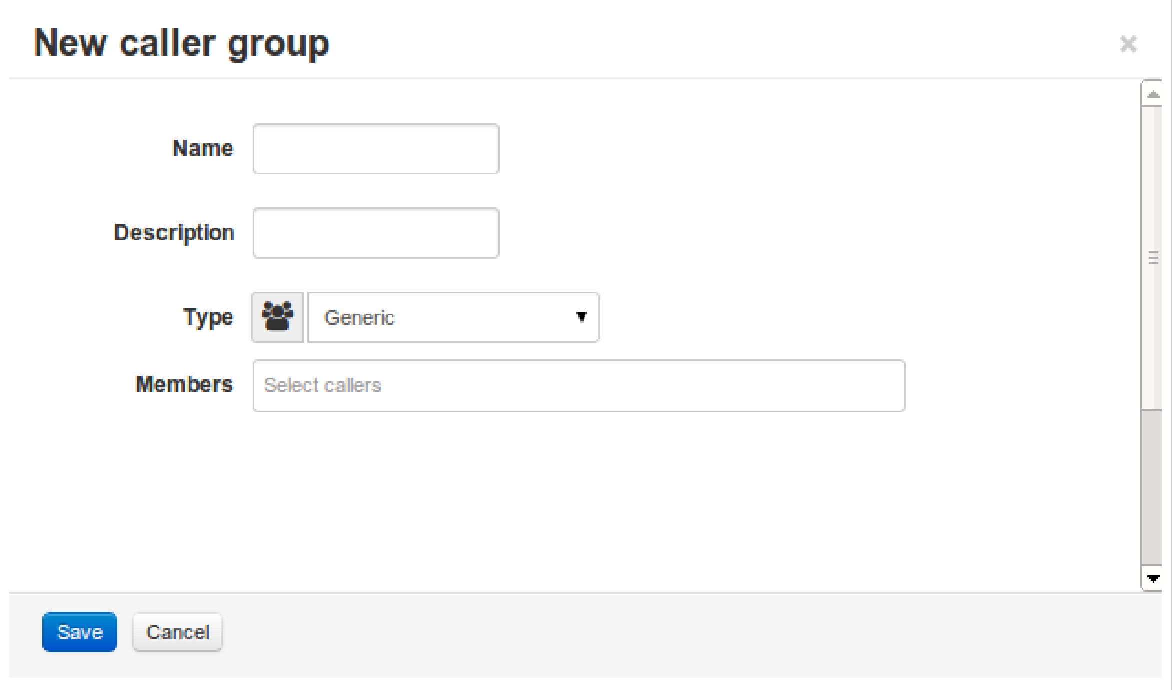
This section describes how to add a new caller group in the system.

To create a new caller group click the *New* button in the Caller Groups portlet.

Definition

Define the general settings of the caller

- *Name*: unique caller group identifier.
- *Description*: a short caller group description (optional).
- *Type*: the type of the caller group. Once a caller group has been defined, its type can be changed. Please note that there are not specific properties depending on the type; the type is only a classification tag.
- *Members*: the callers that belong to the caller group.



New caller group

Name:

Description:

Type: Generic

Members: Select callers

Caller Groups: New caller group modal.

Callers Definition

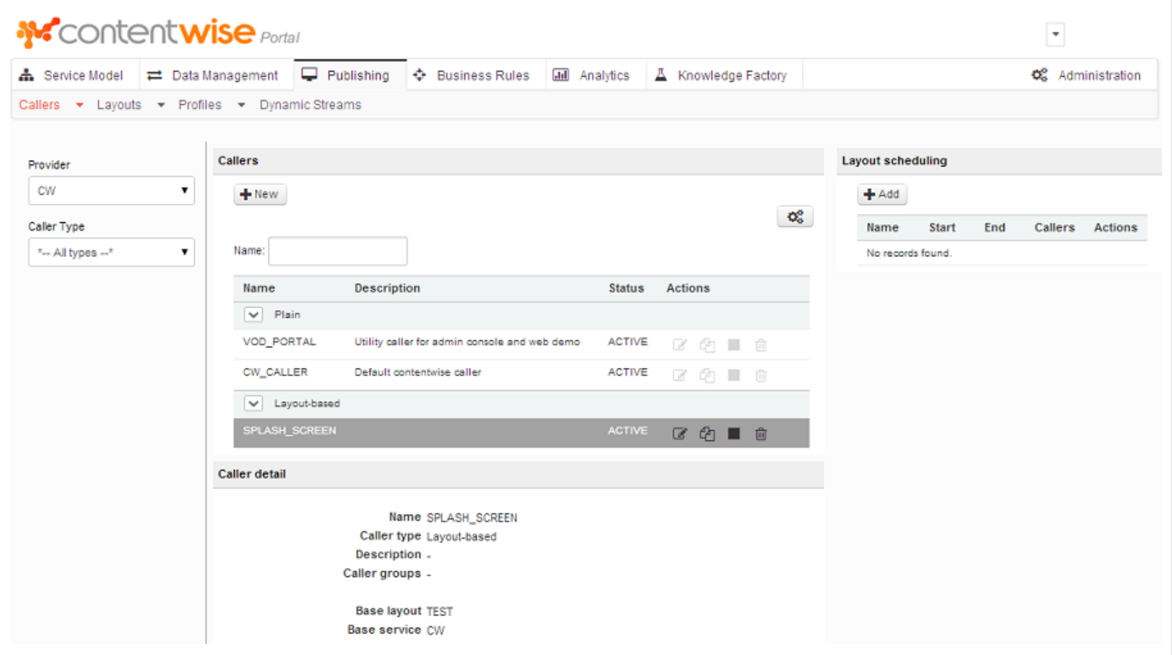
This page provides an overview of caller configurations and provides access to the pages that allow to create and manage callers.

The page contains the following portlets:

- Callers: lists the callers that have been defined and provides access to new/edit caller pages.
- Caller detail: provides details about the selected caller.
- Layout scheduling: Available only for layout-callers, allows to manage layout schedules.

See create a new caller to create a new caller.

If a provider filter is not selected, you will be prompted to select one.



Provider: CW

Caller Type: *-- All types --*

Callers

Name:

Name	Description	Status	Actions
Plain			
VOD_PORTAL	Utility caller for admin console and web demo	ACTIVE	
CW_CALLER	Default contentwise caller	ACTIVE	
Layout-based			
SPLASH_SCREEN		ACTIVE	

Caller detail

Name SPLASH_SCREEN
Caller type Layout-based
Description -
Caller groups -

Base layout TEST
Base service CW

Layout scheduling

Name	Start	End	Callers	Actions
No records found.				

Callers: list of configured callers and related configurations.

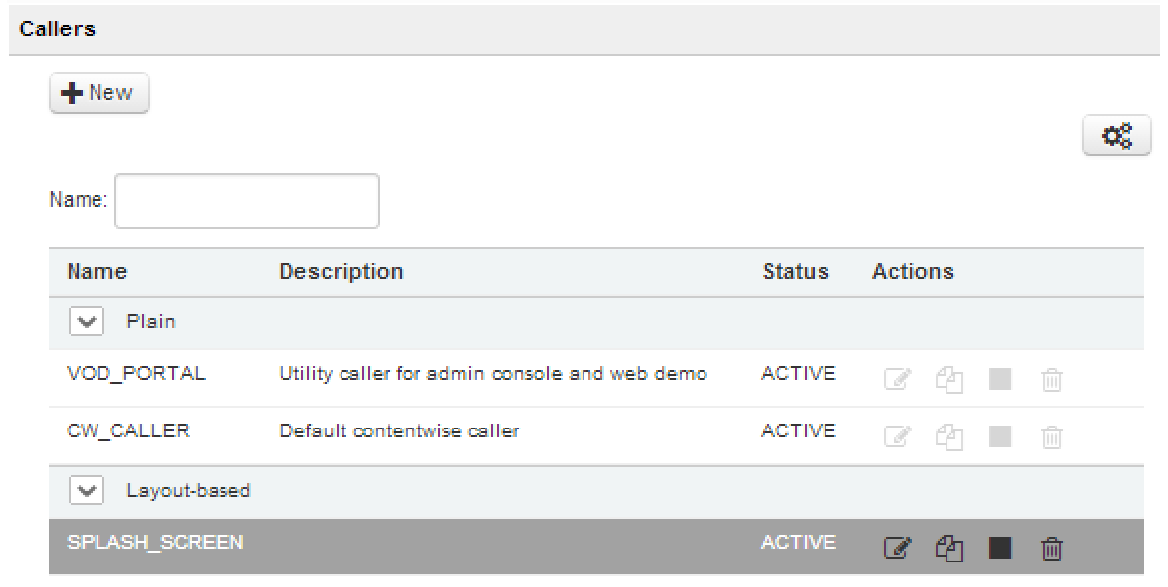
Callers

This portlet provides an overview of caller configuration and provides access to caller configuration pages. Callers are grouped by type.

For each caller, the table shows:

- *Name*: unique caller identifier.
- *Description*: a short caller description.
- *Status*: the caller status. A caller is ACTIVE if it has one or more bound subdomains, otherwise it is INACTIVE
- *Actions*: you can:
    - *edit*: modifies the caller configuration.
    - *clone*: copies the caller configuration into a new caller.
    - *disable*: removes all caller-subdomain bindings.
    - *delete*: remove a caller

It is possible to filter the caller list by caller names. Start typing into the text box above the list. The list of callers will be upadated by showing only callers that match the query string.



Callers

Name:

Name	Description	Status	Actions
Plain			
VOD_PORTAL	Utility caller for admin console and web demo	ACTIVE	
CW_CALLER	Default contentwise caller	ACTIVE	
Layout-based			
SPLASH_SCREEN		ACTIVE	

Callers: Callers portlet.

Caller detail

This portlet provides an overview of the caller that has been selected in the Callers portlet.

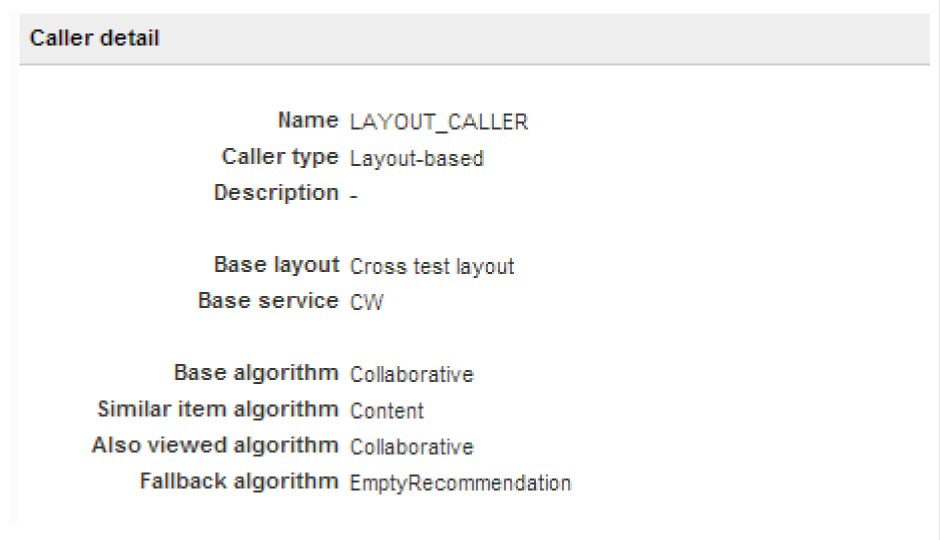
- *Name*: unique caller identifier.
- *Caller type*: the type of the caller (Plain or Layout-based)
- *Description*: a short caller description.
- *Caller groups*: the list of caller groups the caller belongs to (if any).
- *Base service*: the caller default service.
- *Base algorithm*: the caller default algorithm.
- *Similar Item Algorithm*: the algorithm applied by getSimilarItem APIs.
- *Also Viewed Algorithm*: the algorithm applied by getItemAlsoViewed APIs.
- *Fallback strategy*: the recommendation algorithm applied as fallback policy.

Only for plain callers:

- *Subdomain*: the caller default subdomain.
- *Recommendation length*: the caller default length of a recommendation.

Only for layout callers:

- *Base layout*: the default layout of the caller, applied if no scheduling is active.



Caller detail

Name	LAYOUT_CALLER
Caller type	Layout-based
Description	-
Base layout	Cross test layout
Base service	CW
Base algorithm	Collaborative
Similar item algorithm	Content
Also viewed algorithm	Collaborative
Fallback algorithm	EmptyRecommendation

Callers: caller detail portlet.

Create a new caller

This section describes how to add a new caller in the system.

To create a new caller click the *New* button in the Callers portlet.

You will be required to select a provider for which the new caller will operate.

Definition

Define the general settings of the caller

- *Name*: unique caller identifier.

> ⚠ Valid characters are **[A-Za-z0-9]**, **-**, _, **.**

- *Type*: the type of the caller. Once a caller has been defined, its type cannot be changed. Possible values:
  - Plain: a caller that will have to provide standard recommendations.
  - Layout-based: a caller that will operate with layouts.
- *Description*: a short caller description (optional).
- *Caller groups*: the list of caller groups the caller must belong to. It can be empty.

Scope

Define the application domain of the caller

- *Subdomain*: the caller default subdomain. (only plain callers)
- *Base service*: the caller default service.
- *Base layout*: the default layout that is used by the caller if no scheduling is active. (only layout-based callers)

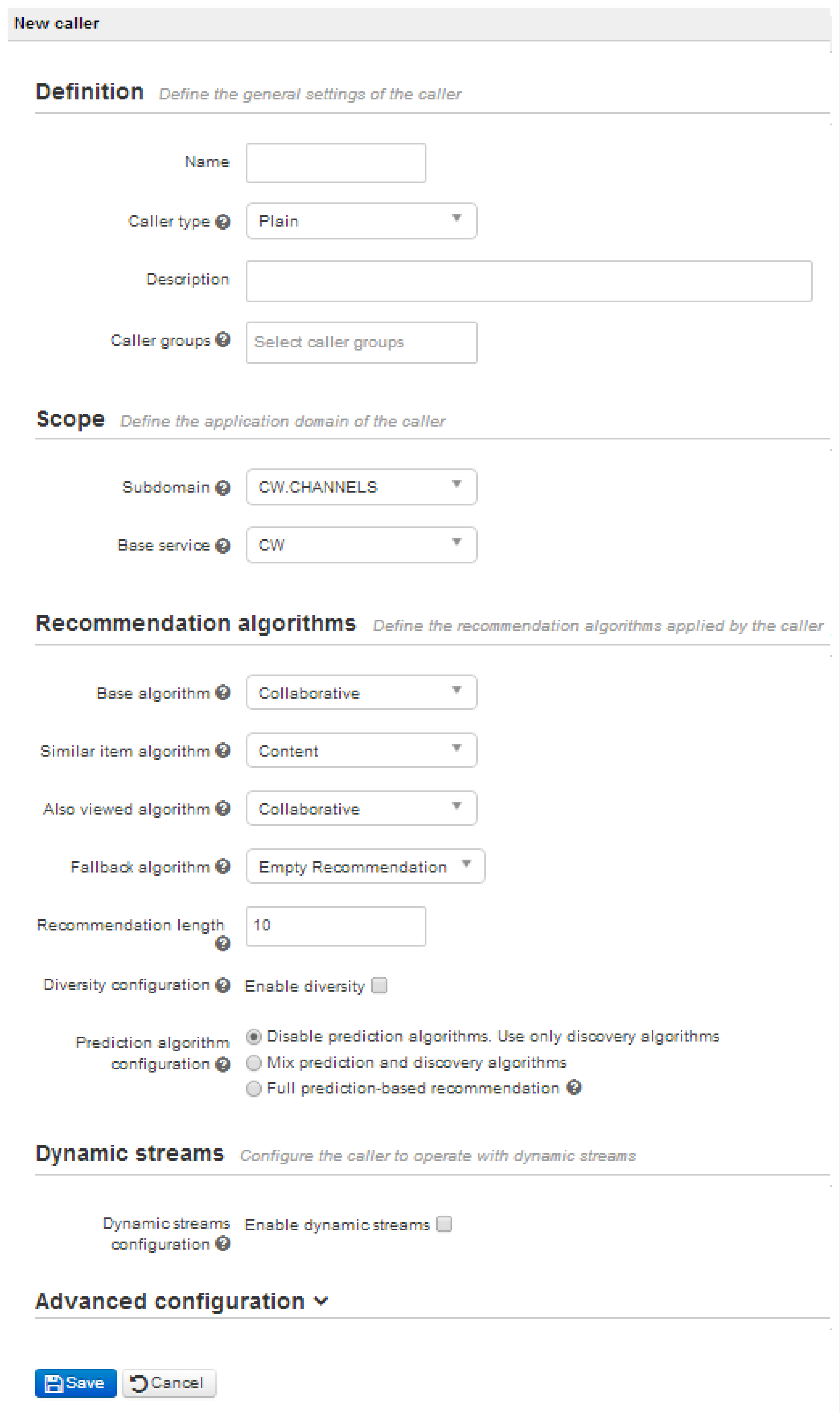
Recommendation algorithms

Define the recommendation algorithms applied by the caller

- *Base algorithm*: the caller default algorithm.
- *Similar Item Algorithm*: the algorithm applied by getSimilarItem APIs.
- *Also Viewed Algorithm*: the algorithm applied by getItemAlsoViewed APIs.
- *Fallback strategy*: the recommendation algorithm applied as fallback policy.
- *Recommendation length*: the caller default length of a recommendation.
- *Diversity configuration*: configure recommendation diversity to enable diversification of results among time.
- *Prediction algorithm configuration*: define the prediction algorithm configuration. It applies only to live events recommendations, both singledomain and crossdomain.

Advanced configuration

A set of advanced options that are not mandatory. Caller editor provides a contextual description of each option.



New caller

Definition *Define the general settings of the caller*

Name

Caller type: Plain ▾

Description

Caller groups: Select caller groups

Scope *Define the application domain of the caller*

Subdomain: CW.CHANNELS ▾

Base service: CW ▾

Recommendation algorithms *Define the recommendation algorithms applied by the caller*

Base algorithm: Collaborative ▾

Similar item algorithm: Content ▾

Also viewed algorithm: Collaborative ▾

Fallback algorithm: Empty Recommendation ▾

Recommendation length: 10

Diversity configuration: Enable diversity □

Prediction algorithm configuration:
- ● Disable prediction algorithms. Use only discovery algorithms
- ○ Mix prediction and discovery algorithms
- ○ Full prediction-based recommendation

Dynamic streams *Configure the caller to operate with dynamic streams*

Dynamic streams configuration: Enable dynamic streams □

Advanced configuration ⌄

Callers: Caller editor portlet.