



# **User manual**

## **Meeting room models**

# Introduction

---

This document provides installation and deployment information for administrators planning to deploy the **meetingroom models** compatible with Innes connector applets. The **meetingroom models** is a set of views for connector applets as *ics-calendar*, *ms-exchange-2003-calendar*, *ews-calendar*, *domino-calendar*, *grr-calendar*.

A **meetingroom model** is a view able to display the calendar information configured by the calendar connector.

Available masks:

- Meetingroom - Mono – Landscape
- Meetingroom - Summarization – Landscape
- Meetingroom - Summarization - Portrait

The next steps must be followed to guarantee a correct deployment.

- Install and configure your connector.
- Install the models
- Configure the variable
- Configure the view

# System Requirements

---

The requirements must be met before the installation.

- Innes Plugncast Server V2.50.24 or higher
- Innes Plugncast Studio V2.50.24 or higher
- Innes Playzilla Player V2.50.24 or higher

Please refer to the 'Meeting room models history.txt' file provided for latest compatibility requirements.

# Installation

---

Copy the following files to maff models location in the plugncast server:

- landscape.mono.mr.mask.maff
- mono.mr.form.swf
- landscape.multi.mr.mask.maff
- portrait.multi.mr.mask.maff
- multi.mr.form.swf

These files can be found under the directory .models\medias\maffs\

The maff model location server is:

**<Innes PlugnCast Data Folder>\Server\accounts<domain>\.models\medias\maffs\**

The **<Innes PlugnCast Data Folder>** is the data folder of Innes PlugnCast.

The **<domain>** is your currently domain.

## Examples

In these examples **<domain>** value is **my\_company.com**.

### Vista Example

**C:\Users\Public\Documents\Innes**

**PlugnCast\Server\accounts\my\_company.com\.models\medias\maffs\**

### XP Example

**C:\Documents and Settings\All Users\Documents\Innes PlugnCast\Server\accounts\my\_company.com\.models\medias\maffs\**

# Variable Configuration

A variable must be created and configured. The variable is used to make the correspondence between the resource form the connector applet and the player(s).

## 2.1 Create a variable.

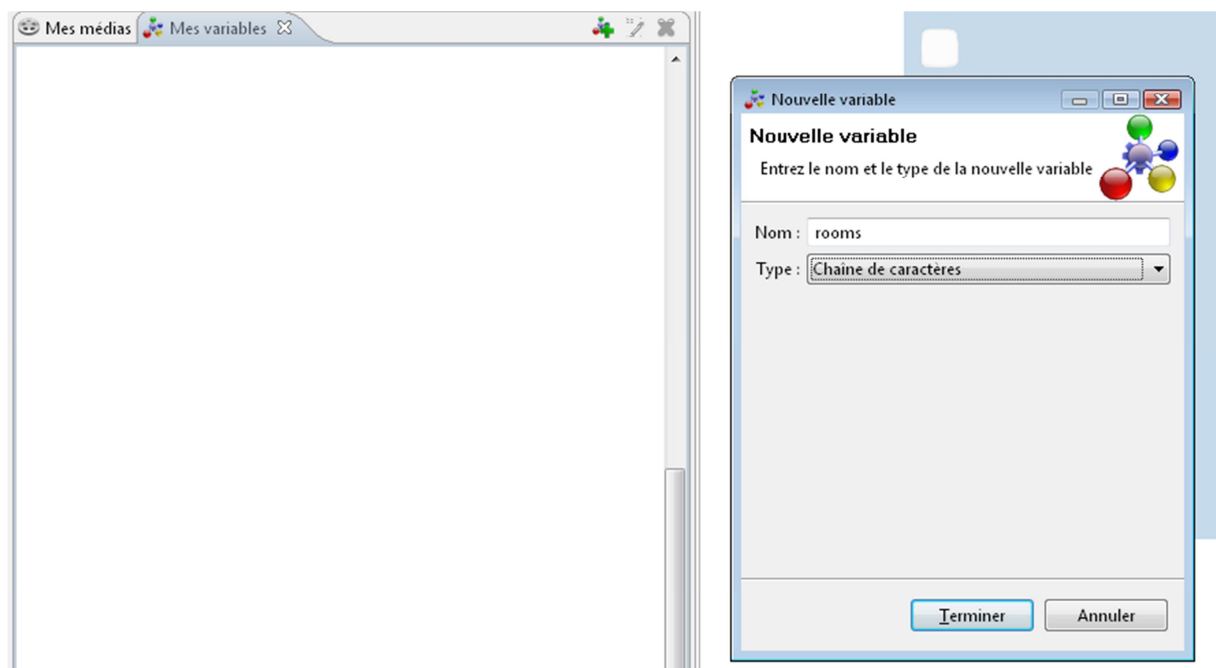
You can view your variables panel by clicking "Settings->Show View->My Variables".

The variable types allowed are: String, Json. Follow the configuration instructions to choose type of variable. Json type is highly recommended.

Create so many values as different player configurations needed.

### 2.1.1 String Variable Value Configuration

Create a new variable. Choose the type **String**.



Value : **<calendarId>**

**<calendarId>** : The calendar id configured in the 'configuration.xml' file, from your applet connector.

#### 2.1.1.1 Basic Example

Extract from configuration.xml :

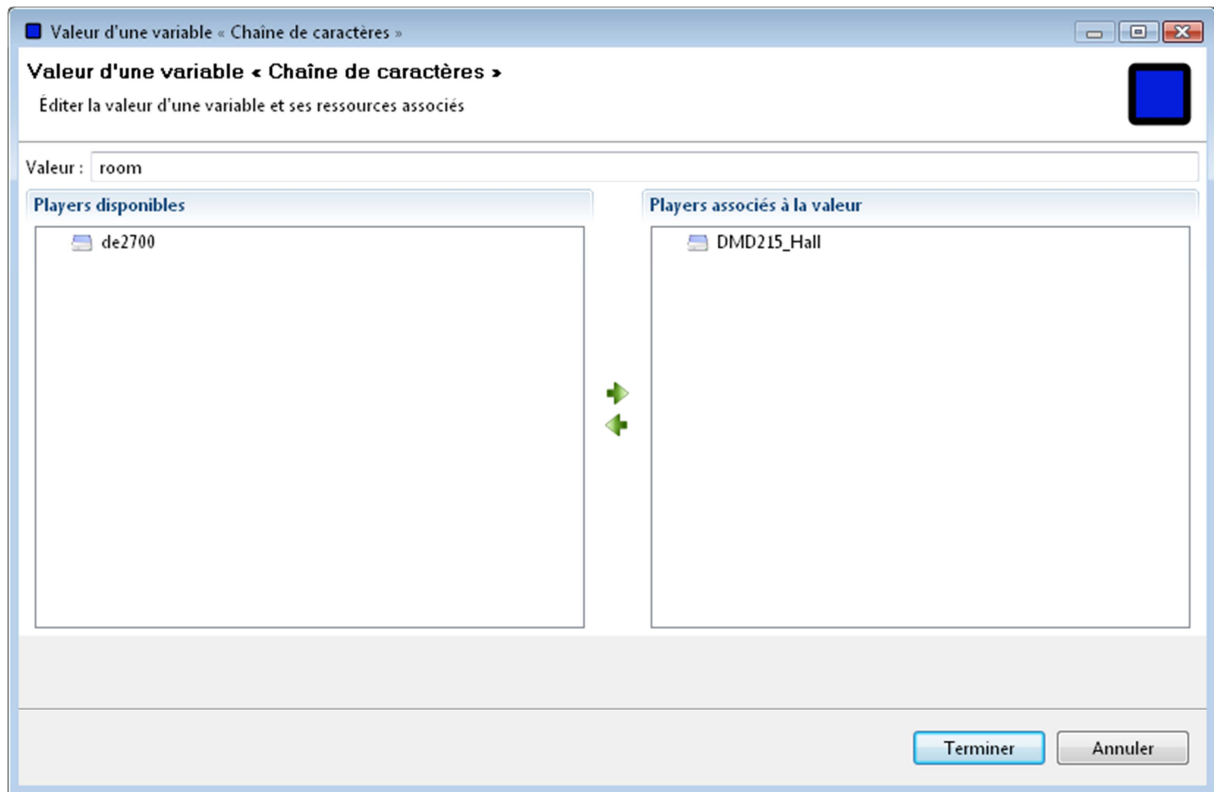
...

```
<scc:calendar id="room" server="url(#server)">
    <param name="resource">resourceURL</param>
</scc:calendar>
```

...

The **<calendarId>** is 'room' (The calendar id configured in the configuration.xml).

Value: room



The associated player is **DMD215\_HALL**. The player **DMD215\_HALL** will display the calendar information from calendar **room**.

### 2.1.1.2 Accentuated calendarId Example

Extract from configuration.xml:

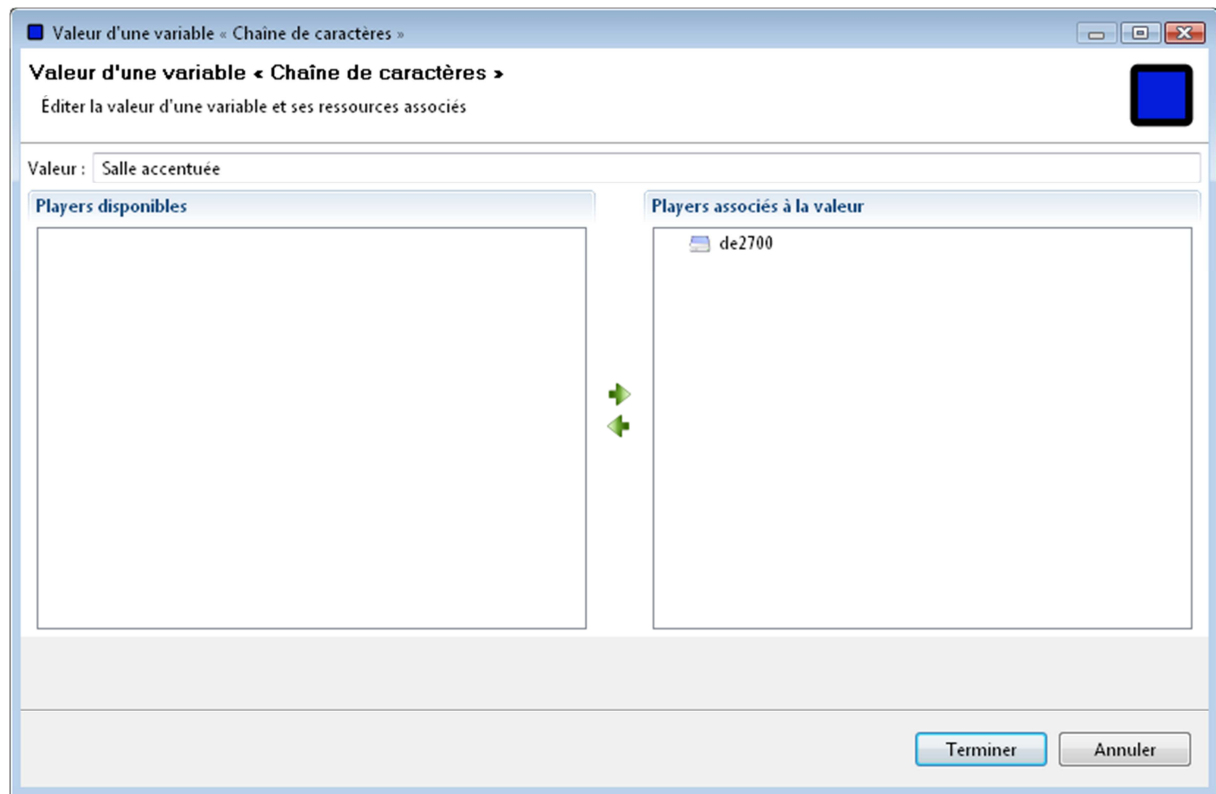
...

```
<scc:calendar id="Salle accentuée" server="url(#server)">
    <param name="resource">resourceURL</param>
</scc:calendar>
```

...

The **<calendarId>** is 'Salle accentuée' (The calendar id configured in the configuration.xml).

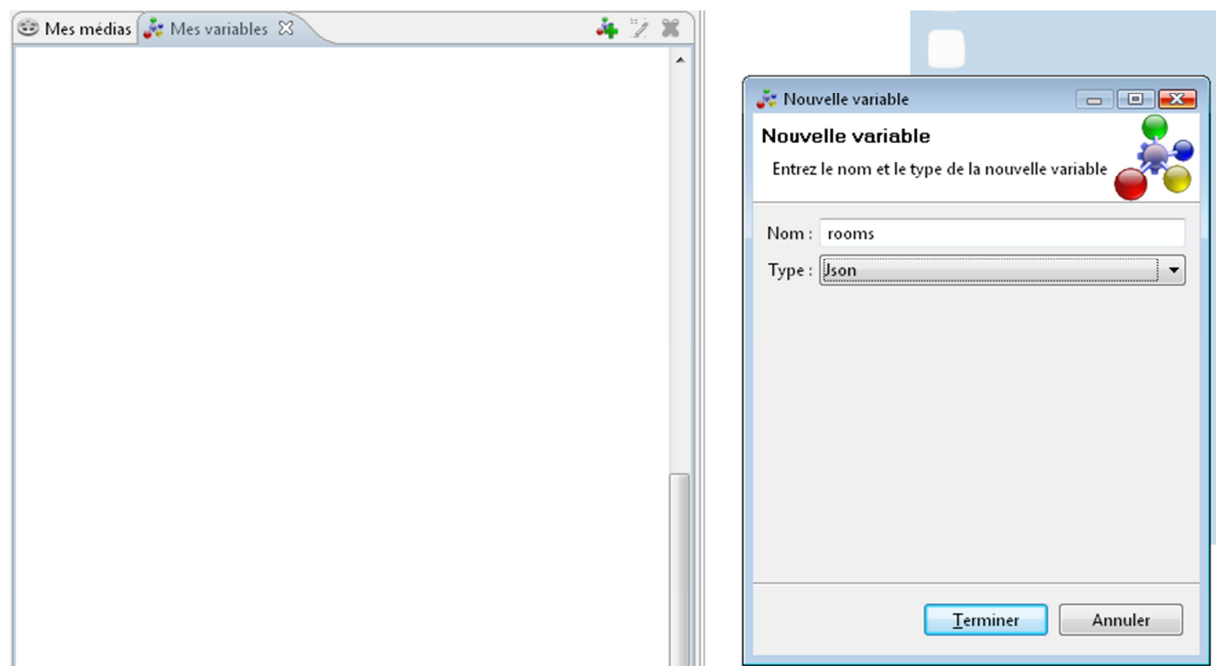
Value: salle accentuée



The associated player is **de2700**. The player **de2700** will display the calendar information from calendar **Salle accentuée**.

## 2.1.2 Json Variable Value Configuration

Create a new variable. Choose the type **Json**.



The precise syntax for json variable is given in annex.

### 2.1.2.1 Basic example

Extract from configuration.xml:

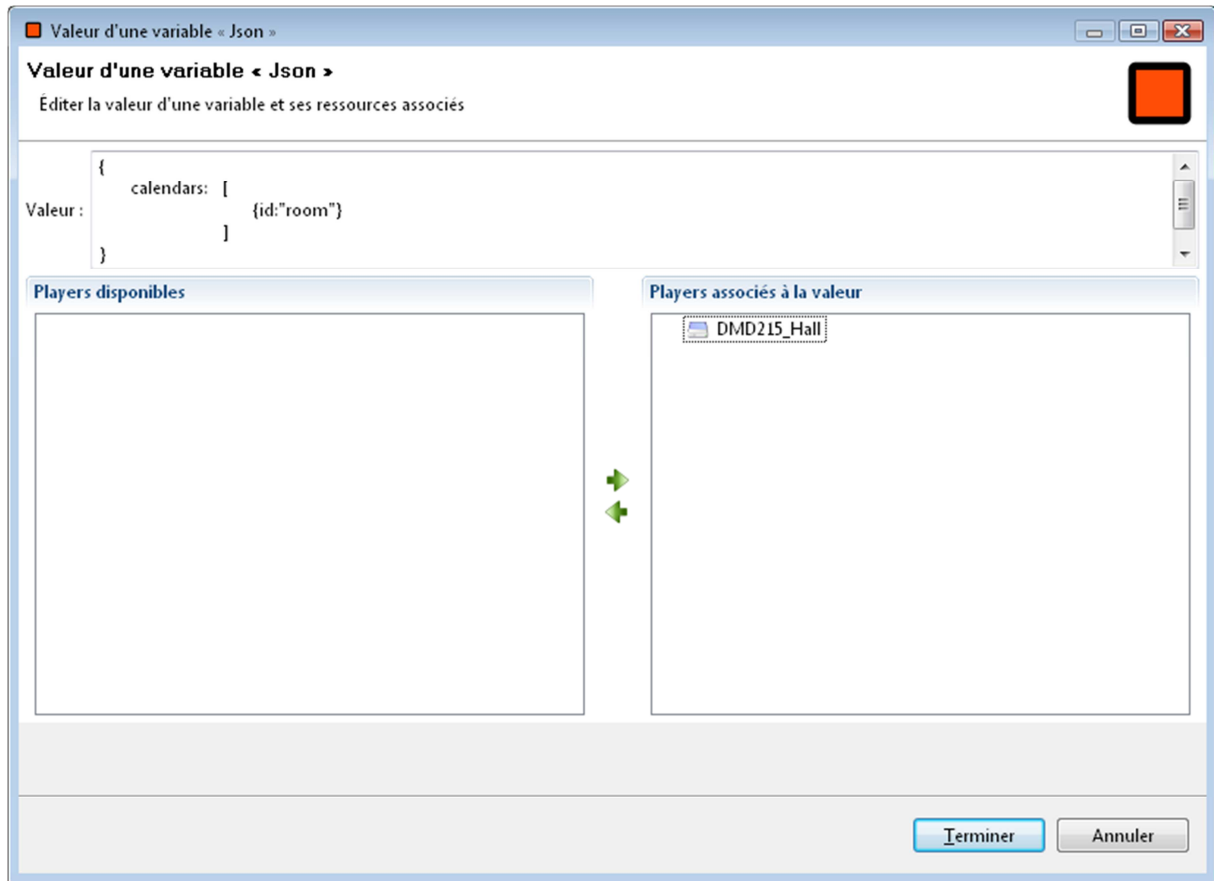
```
...
<scc:calendar id="room" server="url(#server)">
    <param name="resource">resourceURL</param>
</scc:calendar>
...
```

The **<calendarId>** is 'room' (The calendar id configured in the configuration.xml).

Value:

```
{
  calendars: [
    {id:"room"}
  ]
}
```





The associated player is **DMD215\_HALL**. The player **DMD215\_HALL** will display the calendar information from calendar **room**.

### 2.1.2.2 Accentuated calendarId example

...

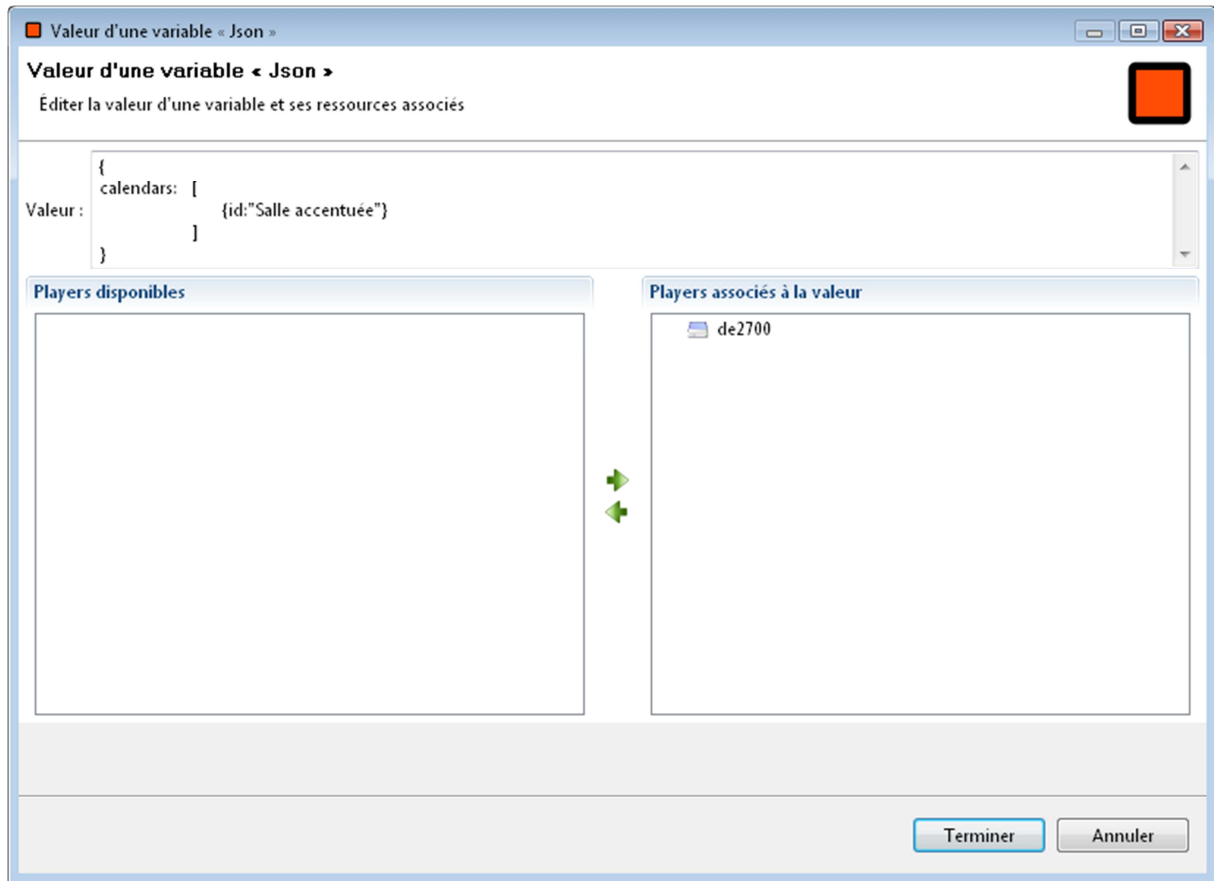
```
<scc:calendar id="Salle accentuée" server="url(#server)">
  <param name="resource">resourceURL</param>
</scc:calendar>
```

...

The **<calendarId>** is 'Salle accentuée' (The calendar id configured in the configuration.xml).

Value:

```
{
  calendars: [
    {id:"Salle accentuée"}
  ]
}
```



The associated player is **de2700**. The player **de2700** will display the calendar information from calendar **Salle accentuée**.

### 2.1.2.2 Idle message Example

The **<calendarId>** is 'room' (The calendar id configured in the configuration.xml).

The **<idleMessage>** is "No events to show"

Value :

```
{
    calendars:  [
        {id:"rooms"}
    ],
    idle:{src:"data:text/plain,No events to show"}
}
```

### 2.1.2.3 Signs Example

This example shows how to add signs to the view.

...

```
<fcc:calendar id="rooms" server="url(#server)">
  <param name="resource">resourceURL-1</param>
  ...
  <param name="resource">resourceURL-N</param>
</fcc:calendar>

...
```

The **<calendarId>** is 'rooms' (The calendar id configured in the configuration.xml).

The property 'locations' is used to specify extra information of a resource (LOCATION iCalendar property)

This example specifies extra information for two resources (LOCATION iCalendar property), first 'Room', secondly 'Room A'.

Resource '**Room**':

- The way to go to the resource 'Room' is on the left side of the screen, then **<sign1Path>** specifies the path of an left arrow image 'arrows/arrowLeft.png' (relative to ".medias/").
- The resource 'Room' is on the floor above, then **<sign2Path>** specifies the path of an escalator up image 'arrows/escalatorUp.png' (relative to ".medias/").

Resource '**Room A**':

- The way to go to the resource 'Room A' is on the right side of the screen, then **<sign1Path>** specifies the path of an left arrow image 'arrows/arrowRight.png' (relative to ".medias/").
- The resource 'Room A' is on the floor below, then **<sign2Path>** specifies the path of an escalator down image 'arrows/escalatorDown.png' (relative to ".medias/").

The sign images must be linked medias of the model (see image below value example).

Value:

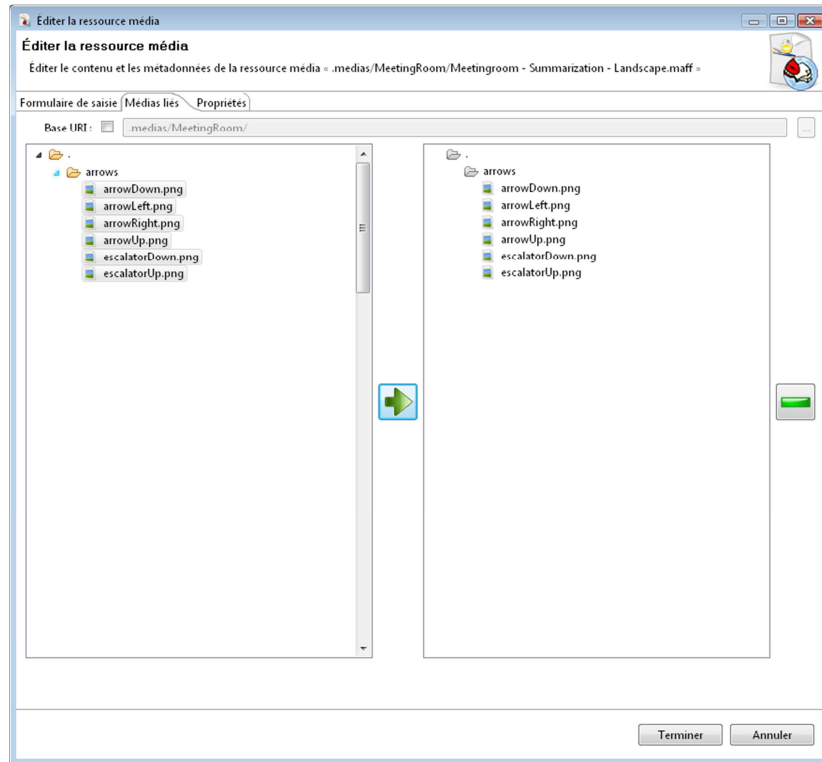
```
{
  calendars: [
    {id:"rooms"}
  ],
  locations: [
    {resource:"Room", sign1:"arrows/arrowLeft.png",
sign2:"arrows/escalatorUp.png"},
```

```

    {resource:"Room A", sign1:"arrows/arrowRight.png",
    sign2:"arrows/escalatorDown.png"}
  ]
}

```

Sign images as linked medias:



Examples of signs images are given in the file 'signs examples.zip'.

#### 2.1.2.4 - Range Example (Multiscreen Synchronization)

Use only with **Summarization** configurations.

This feature handles multiple screens synchronization showing the same calendar/s information.

Note: The range feature is not compatible with viewport feature!

The range feature split in small groups of events the whole group of events to display.

The `<calendarId>` is 'rooms' (The calendar id configured in the configuration.xml).

This example configures three screens with range feature, first '**Screen 1**', secondly '**Screen 2**', thirdly '**Screen 3**'.

Screen '**Screen 1**' (14 Lines):

- Screen 'Screen 1' handles the first 14 events to be displayed.

Screen '**Screen 2**' (14 Lines):

- Screen 'Screen 2' handles the events to be displayed from 15 to 28.

Screen '**Screen 3**' (14 Lines):

- Screen 'Screen 3' handles the events to be displayed from 29 to the end.

'Screen 1' Value:

```
{
  calendars: [
    {id:"rooms"}
  ],

  range:{position:"1",length:"14"}
}
```

'Screen 2' Value:

```
{
  calendars: [
    {id:"rooms"}
  ],

  range:{position:"15",length:"14"}
}
```

'Screen 3' Value:

```
{
  calendars: [
    {id:"rooms"}
  ],

  range:{position:"29"}
}
```

### 2.1.2.5 - Viewport Example (Multiscreen Synchronization)

Use only with **Summarization** configurations.

This feature handles multiple screens synchronization showing the same calendar/s information.

Note: The viewport feature is not compatible with range feature!

The viewport enables list view. The events to be displayed are handled as a vertical carousel between screens.

The **<calendarId>** is 'rooms' (The calendar id configured in the configuration.xml).

This example configures three screens with viewport feature, first '**Screen 1**', secondly '**Screen 2**', thirdly '**Screen 3**'.

Screen '**Screen 1**' (14 Lines):

- Screen 'Screen 1' starts at position 1 and the next screens can handle 28 lines.

Screen '**Screen 2**' (14 Lines):

- Screen 'Screen 2' starts at position 15 and the next screens can handle 7.

Screen '**Screen 3**' (14 Lines):

- Screen 'Screen 3' starts at position 29 and the are no more screens.

'**Screen 1**' Value:

```
{
  calendars: [
    {id:"rooms"}
  ],
  viewport:{"position":"1","remain":"28"}
}
```

'**Screen 2**' Value:

```
{
  calendars: [
    {id:"rooms"}
  ],
  viewport:{"position":"15","remain":"7"}
}
```

```
],
```

```
viewport:{"position":"15","remain":"14"}
```

```
}
```

'Screen 3' Value:

```
{
```

```
  calendars:  [
```

```
    {id:"rooms"}
```

```
  ],
```

```
viewport:{"position":"29"}
```

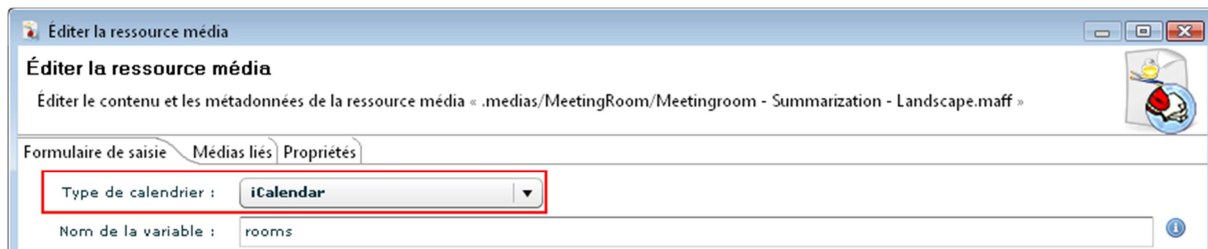
```
}
```

# Form Configuration

All **meetingroom maks** have a form where the next common parameters must be configured:

## 3.1 Calendar type

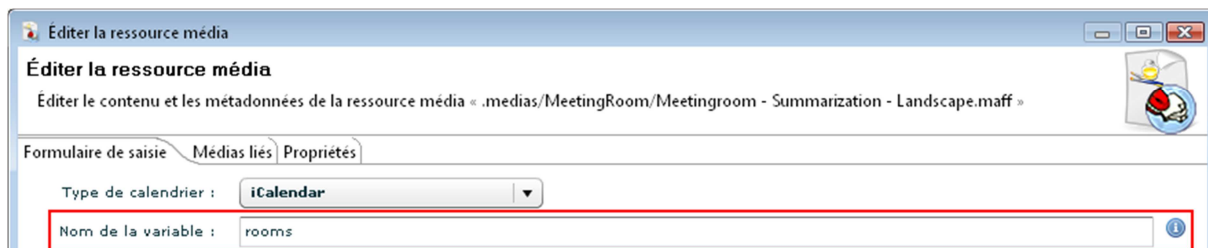
The connector calendar type configured in your system.



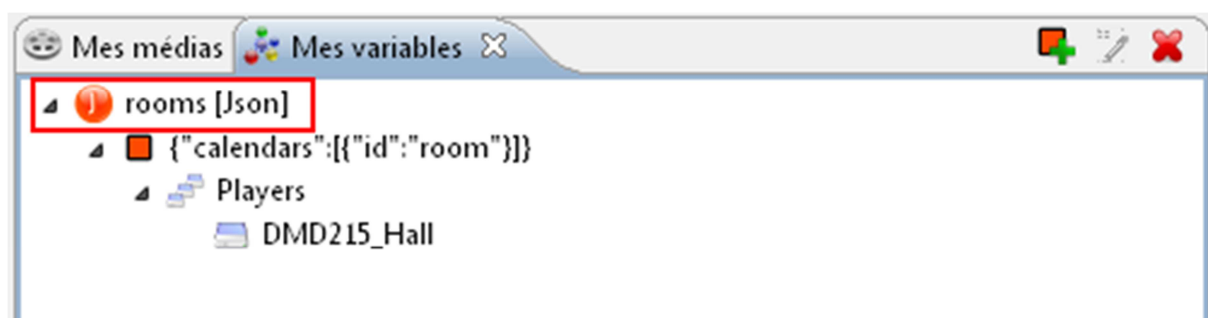
The screenshot shows a window titled "Éditer la ressource média". Below the title bar, there's a subtitle "Éditer le contenu et les métadonnées de la ressource média « .medias/MeetingRoom/Meetingroom - Summarization - Landscape.maff »". There are three tabs: "Formulaire de saisie", "Médias liés", and "Propriétés". The "Formulaire de saisie" tab is active. It contains two fields: "Type de calendrier :" with a dropdown menu set to "iCalendar", and "Nom de la variable :" with a text input field containing "rooms". A red box highlights the "Type de calendrier" dropdown.

## 3.2 Variable name

The name of the configured variable.



This screenshot is identical to the one above, showing the "Éditer la ressource média" window. In this view, a red box highlights the "Nom de la variable" text input field, which contains the text "rooms".



The screenshot shows a window with two tabs: "Mes médias" and "Mes variables". The "Mes variables" tab is active. It displays a tree structure of variables. The root variable is "rooms [Json]", which is highlighted with a red box. Below it, there's a sub-variable "calendars" with a value of [{"id": "room"}]. Under "calendars", there's a sub-variable "Players" with a value of "DMD215\_Hall".



# Annex

---

Syntax for json variable :

Value:

```
{
  //Calendars
  //Occurrence=1..n
  calendars: [
    configuration.xml //<calendarID> : The calendar id configured in the
                      {id:"<calendarID>"}
  ],

  //Idle state parameters
  //Occurrence=0..1
  idle:{
    //<idleMessage> : The idle message
    src:"data:text/plain,<idleMessage>"
  },

  //Locations extra information
  //Occurrence=0..1
  locations: [
    {
      //Occurrence=1
      //<resourceName> : The resource name.
      Generally is the room name (LOCATION iCalendar Property)
      resource:"<resourceName>",

```

```

                                //Occurrence=0..1
                                //<sign1Path> : The sign1 relative path to
".medias/"
                                sign1:"<sign1Path>",

                                //Occurrence=0..1
                                //<sign2Path> : The sign2 relative path to
".medias/"
                                sign2:"<sign2Path>"
                                }
                                ],

                                //Events range
                                //Occurrence=0..1
                                range:{
                                    //Occurrence=1
                                    //<position> : The first position of the range (firstPosition=1)
                                    position:"<position>",

                                    //Occurrence=0..1
                                    //<length> : The length of the range. If no length is specified,
the range length is infinite
                                    length:"7"
                                }

                                //Events viewport
                                //Occurrence=0..1 (not compatible with range)
                                viewport:{
                                    //Occurrence=1
                                    //<position> : The first position of the range (firstPosition=1)
                                    position:"<position>",

                                    //Occurrence=0..1
                                    //<remain> : The number of of event lines that remain after this
screen
                                    remain:"7"
                                }

```

}

}