CGManager

Technical Reference Sheet

CGManager

Technical Reference Sheet

TRS-1040-01
Aveco
www.aveco.com

Publication Date: October 2024

Table of Contents

OVERVIEW

The **Aveco CGManager** is a standalone software application designed to manage and administrate graphical templates, primarily for the Redwood WHITE server but not limited to it. It operates as a web-based application hosted on an embedded web server and can run on Linux within a Docker container or as a Windows 64-bit service. The **CGManager** integrates with ASTRA NUE and newsroom computer systems (NRCS) as a plugin using the widely adopted MOS HTML standard.

FEATURES

- **Web-Based Interface**: Accessible through any modern web browser, providing an intuitive user interface for managing graphic templates.
- **Template Management**: Displays template names, creation dates, and thumbnails on the landing page.
- **Sorting and Filtering**: Allows sorting by “Author,” “Category,” or “Created” fields and filtering using a search function.
- **Template Data Editing**: Clicking on a template opens an edit window with preview functionality.
- **Dynamic Field Preview**: The Aveco HTML templates can be previewed with their dynamic fields.
- **Validation**: Checks the validity of timecode fields and alerts users to any whitespace issues in text fields.
- **Plugin Integration**: Features an “Insert to Rundown” button for seamless integration with NRCS and MCR systems, adhering to the HTML MOS standard.

INSTALLATION

SYSTEM REQUIREMENTS

Requirement Type	Minimum Specification	Recommended Specification
CPU	1 CPU Core	2-4 CPU Cores
Memory (RAM)	1 GB	4-8 GB
Storage	20 GB HDD/SSD	100 GB SSD
Network	100 Mbps Ethernet	1 Gbps Ethernet
Operating System	openSUSE Leap (Latest) or Windows 64-bit	openSUSE Leap (Latest) or Windows 64-bit

DEPLOYMENT OPTIONS

- **Linux**: Runs within a Docker container on a Linux system.
- **Windows**: Can be installed as a Windows 64-bit service.

INTEGRATION

SUPPORTED TEMPLATE FORMAT

To display GFX templates within **CGManager**, templates must be properly placed and accompanied by a sidecar XML file. The XML format specifies the parameters and attributes that ASTRA can send to the GFX inserter, enabling dynamic content rendering and control within the broadcasting environment.

XML DESCRIPTION FILE FOR GRAPHICAL TEMPLATES

This section outlines the structure and format of the XML metadata used by graphical templates to communicate with the ASTRA system. Each XML file for a graphical template defines several **`parameter`** tags, each with three attributes:

- **`id`**: The name of the variable.
- **`type`**: Defines the type of the variable.
- **`info`**: Provides additional information or context for the variable.

PARAMETER TYPES

- **`type="string"`**: Represents a standard one-line dynamic field, displayed as a textbox. The info attribute is used as a tooltip.
- **`type="text"`**: Represents a multi-line dynamic text field, displayed as a textbox.
- **`type="integer"`**: Displayed as a textbox but input is restricted to integers. The info attribute serves as a tooltip.
- **`type="image"`**: Specifies a file URL. Displayed as a dropdown allowing the user to select a file located in a preconfigured folder.
- **`type="list"`**: Represents a choice of predefined values, displayed as a combo box. The info attribute contains a semicolon-separated list of possible values.
- **`type="info"`**: Used to provide the automation system with hints (e.g., which layer to play the graphical template). Variables of this type must start with astra_ and are not visible in the user interface.
- **`type="auto"`**: Fields automatically populated by ASTRA to convey runtime information back to the graphical inserter.

EXAMPLE OF XML DESCRIPTION FILE

```
<template version="2.0.0" authorName="Aveco"
 authorEmail="info@aveco.com" templateInfo=""
 originalWidth="1920" originalHeight="1080"
 originalFrameRate="50">
  <components/>
  <keyframes/>
  <instances/>
  <parameters>
    <parameter id="Nametag" type="string" info="Bottom text line
 below the main text"/>
    <parameter id="Photobox" type="image" info="File URL for an
 image inside the photobox."/>
```

```
    <parameter id="AUX_Input" type="list"
 info="PST;IN1;IN2;IN3;IN4;PLAY1;PLAY2;PLAY3;PLAY4"/>
    <parameter id="PGM_Volume" type="integer" info="Volume of the
 PGM in dB, default to 0"/>
    <parameter id="astra_category" type="info" info="LowerThird"/
>
    <parameter id="astra_output" type="auto" info="Channel, as
 set by the automation"/>
    <parameter id="astra_layer" type="auto" info="Layer, as set
 by the automation"/>
  </parameters>
</template>
```

PLACEMENT AND NAMING

The sidecar XML file is expected to be in the subfolder of the template, using its name. The expected file structure is as follows:

File Location	Functionality
`template_name/template_name.html`	Actual GFX template (HTML format)
`template_name/template_name.swf`	Actual GFX template (SWF format)
`template_name/template_name.fla`	Actual GFX template (FLA format)
`template_name/template_name.xml`	Template description file
`template_name/template_name.png`	(Optional) Thumbnail for GUI display

API DOCUMENTATION

The **CGManager** provides a RESTful API for integration with third-party providers. The API documentation can be accessed at: **`http://<ip_address>:7200/docs`**

Replace **`<ip_address>`** with the actual hostname or IP address of the **CGManager** server.

AVAILABLE ENDPOINTS

The API allows for the following operations:

- **Get Image**: **`GET /api/img/{image_name}`** - Returns a full-size image from the library.
- **Get Thumbnail**: **`GET /api/thumb/{thumb_name}`** - Returns a thumbnail of the given image from the library.
- **Get Template List**: **`GET /api/cg`** - Returns a list of available templates. Supports sorting and filtering.
- **Get Template Details**: **`GET /api/cg/{template_name}`** - Returns detailed information about a specific template.
- **Get Template Thumbnail**: **`GET /api/cg/{template_name}/thumb`** - Returns a thumbnail image of the specified template.
- **Export MOS Object**: **`POST /api/cg/{template_name}/export`** - Creates a MOS message for the specified template.

EXAMPLE USAGE

RETRIEVING TEMPLATES

```
GET http://<ip_address>:7200/api/cg?
sort=name&sort_desc=false&query=lowerthird&reload=false
```

EXPORT MOS

```
POST http://<ip_address>:7200/api/cg/lowerthird/export
Content-Type: application/json
{
  "obj_id": "MOS12345",
  "item_id": "ITEM67890",
  "in_addr": "00:00:05:00",
  "duration": "00:00:10:00",
  "abstract": "Lower Third Graphic",
  "thumbnail_url": "http://<ip_address>:7200/api/cg/lowerthird/
thumb",
  "values": {
    "full_name": "Jane Smith",
    "title": "CTO"
  }
}
```

USER INTERFACE GUIDE

INTERFACE OVERVIEW

- **Landing Page**: Displays a list of templates with names, creation dates, and thumbnails.
- **Refresh Button (“X”)**: Located next to the search field; clicking it clears filters and refreshes the template list.
- **Sorting**: Click on the “Author,” “Category,” or “Created” headers to sort the template list accordingly.
- **Search Function**: Use the search field in the upper-right corner to filter templates.
- **Template Editing**: Click on a template to open the edit window, which includes preview functionality.
- **Validation**: The application checks the validity of timecode fields based on the **`objTB`** configuration parameter and alerts users to any whitespace issues in text fields.
- **Insert to Rundown**: Used for integrating templates into NRCS rundowns.

NAVIGATING THE INTERFACE

1. **Accessing the CGManager**: Open a web browser and navigate to the CGManager’s hostname.
2. **Viewing Templates**: The landing page displays all available templates.
3. **Filtering Templates**: Use the search bar to filter templates by name or other attributes.
4. **Sorting Templates**: Click on column headers to sort templates by “Author,” “Category,” or “Created” date.
5. **Editing a Template**:
   - Click on a template to open the edit window.
   - Modify the dynamic fields as needed.
   - Use the preview functionality to see changes in real-time (**be advised that preview functionality applies only to certain HTML templates**).
6. **Validation Alerts**:
   - If a timecode field is invalid an alert will appear.
   - Whitespace issues in text fields will also trigger an alert.
7. **Inserting into News Rundown or MCR Playlist**:
   - After editing, click on the “Insert to Rundown” button.