MAINCONCEPT™

MainConcept 2GO OTT Live Encoder

User Guide

MainConcept 2GO OTT Live Encoder

User Guide

Contents

Product version 2.3, last updated: **June 14, 2021**

1. Introduction

The MainConcept 2GO OTT Live Encoder is an optimized Docker container for live encoding of UDP, RTSP and RTMP network streams into adaptive streaming formats Apple HLS and MPEG-DASH as AVC/H.264 and HEVC/H.265 for playback on iOS and Android devices. The included packager creates ready-to-use manifest and playlist files for direct upload to a CDN (Akamai & CloudFront) or HTTP servers.

Features:

- Live Adaptive Bitrate Stream Encoding to MPEG-DASH & Apple HLS up to 4K 10-bit
- Parallel MPEG-DASH & HLS Layers Packaging as well as manifest and playlist generation
- IP network streams input via UDP (MPEG-2 / H.264 in TS), authenticated RTSP, RTMP and HTTP
- Hybrid playlist generation and packaging for HLS AVC and HEVC

Presets:

- HLS AVC
- HLS HEVC
- DASH-264
- DASH-265

2. Installation

Before installing MainConcept 2GO, please ensure Docker is installed correctly.

2.1 Installing Docker

To download and install Docker on your server, please follow the instructions here:
www.docker.com/community-edition
After successful installation you should be able to run the Docker “hello-world” container.
Example output from the Docker hello-world container:

```
$ docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub. (amd64)
3. The Docker daemon created a new container from that image which runs the executable
that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your
terminal.
```

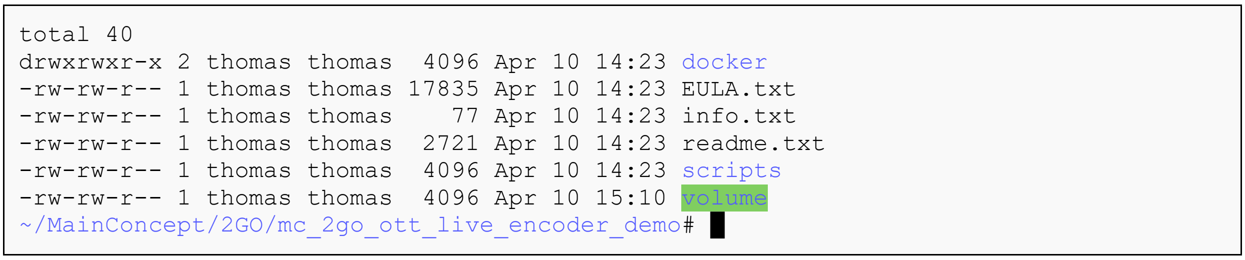
If you see different output, please refer to the Docker installation guide.

2.2 Resources for Docker

On Windows machines there is a default limitation of resources available for Docker. It could be adjusted in the "Settings > Advanced" section. For more information, please check the official website: docs.docker.com/docker-for-windows.

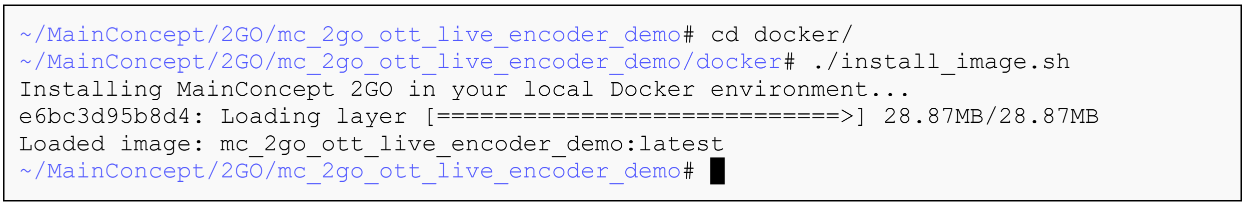
2.3 Installing MainConcept 2GO

1) Extract the MainConcept 2GO package
   To install your MainConcept 2GO product, first unpack the ZIP file you downloaded into a new folder on your computer. The files within the folder depend on the 2GO product.



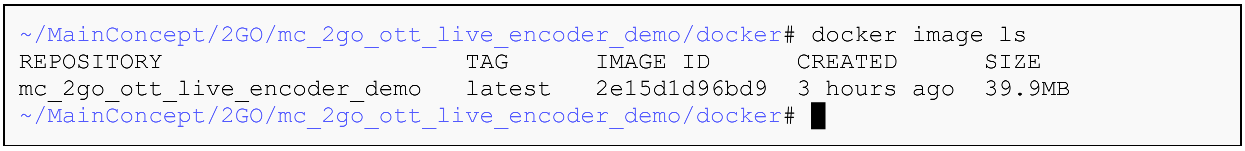
```
total 40
drwxrwxr-x 2 thomas thomas  4096 Apr 10 14:23 docker
-rw-rw-r-- 1 thomas thomas 17835 Apr 10 14:23 EULA.txt
-rw-rw-r-- 1 thomas thomas    77 Apr 10 14:23 info.txt
-rw-rw-r-- 1 thomas thomas  2721 Apr 10 14:23 readme.txt
-rw-rw-r-- 1 thomas thomas  4096 Apr 10 14:23 scripts
-rw-rw-r-- 1 thomas thomas  4096 Apr 10 15:10 volume
~/MainConcept/2GO/mc_2go_ott_live_encoder_demo#
```

2) Install the MainConcept 2GO docker image
   To install the MainConcept 2GO image in your local Docker environment, “cd” into the docker folder and run the “install_image” script:



```
~/MainConcept/2GO/mc_2go_ott_live_encoder_demo# cd docker/
~/MainConcept/2GO/mc_2go_ott_live_encoder_demo/docker# ./install_image.sh
Installing MainConcept 2GO in your local Docker environment...
e6bc3d95b8d4: Loading layer [============================>] 28.87MB/28.87MB
Loaded image: mc_2go_ott_live_encoder_demo:latest
~/MainConcept/2GO/mc_2go_ott_live_encoder_demo#
```

   Verify whether the 2GO container is installed properly by using the “docker image ls” command:



```
~/MainConcept/2GO/mc_2go_ott_live_encoder_demo/docker# docker image ls
REPOSITORY                     TAG       IMAGE ID       CREATED       SIZE
mc_2go_ott_live_encoder_demo   latest    2e15d1d96bd9   3 hours ago   39.9MB
~/MainConcept/2GO/mc_2go_ott_live_encoder_demo/docker#
```

NOTE:
For using both the demo and full version of MainConcept 2GO, you must allow the server running MC2GO a connection to https://taas-reporting-srv.mainconcept.com.

If you want to use the products offline (i.e. without internet connection), please contact customer.care@mainconcept.com. We will get in touch with you about the necessary steps.

3. Settings

3.1 Supported Formats

MainConcept 2GO OTT Live Encoder has support for the following formats.

Input:

- RTMP
- Authenticated RTSP (in MPEG TS)
- HTTP

Output:

- Apple HLS (including fMP4 packaging and playlist generation)
- MPEG-DASH (including fMP4 packaging and manifest file generation)
- Video Codec: AVC/H.264 (8-bit), HEVC/H.265 (8-bit) with up to 30 fps
- Audio Codec: AAC LC, 128 kbps, 2 channels

Packaging Formats:

- DASH
- MBR_HLS

Encoding Groups:

- DASH_HLS_H264
- DASH_HLS_H265

Encoding Presets for DASH_HLS_H264:

- 416x234_177kbps
- 480x270_429kbps
- 640x360_794kbps
- 768x432_1164kbps
- 960x540_2128kbps
- 1280x720_3128kbps
- 1280x720_4628kbps
- 1920x1080_6128kbps
- 1920x1080_7928kbps

Encoding Presets for DASH_HLS_H265:

- 416x234_177kbps
- 480x270_364kbps
- 640x360_724kbps
- 768x432_1054kbps
- 960x540_1828kbps
- 1280x720_2528kbps
- 1280x720_3328kbps
- 1920x1080_4628kbps
- 1920x1080_5928kbps

- 2560x1440_8228kbps
- 3840x2160_11728kbps
- 3840x2160_16928kbps

4. Configuration

With Docker installed most of the MainConcept 2GO configuration is complete. However, some 2GO products require shared folders, or specific parameters at startup.

4.1 Configuring shared folders

To read and write files located on the host computer, MainConcept 2GO uses shared volumes to access the filesystem of the host. Docker uses mounted volumes to share host folders with 2GO containers.

The "run" convenience script in the scripts folder runs the MainConcept 2GO container and automatically maps the required folders from your host computer into the 2GO container.

5. Usage

5.1 Starting MainConcept 2GO

Make sure you have successfully installed your MainConcept 2GO product on the computer by following the Installation instructions.

The "run" script inside the scripts folder makes starting MainConcept 2GO easy and lets you understand how 2GO docker containers are actually run. If you plan to run MainConcept 2GO through container management tools like Docker Compose, Kubernetes or Docker Swarm, it is recommended that you understand the parameters required for 2GO containers by reading the "run" script.

MainConcept 2GO products require passing configuration parameters to the container at startup. These can include input and output filenames, serial keys, shared volume folders or external URLs. These parameters can be specified in a properties file or via the command line.

5.2 Using job description file

MainConcept 2GO v2.1 introduces a REST API and provides users an interface that is more suitable for integration with their existing environment or tools. This API covers functionality to create jobs and query their status using standard REST API over HTTP. A job description file is posted to the endpoint, which must be in JSON format. This JSON file must contain all the necessary parameters required for the submitted jobs.

Option	Sample value	Description
SHARED_PATH_OUT		Folder for the encoding output packages
/	Ingest URL for encoding	
LOGIN		User name for RTMP / RTSP source, optional.
PASSWORD		User password for RTMP / RTSP source, optional.
OUTPUT_TYPE	MBR_HLS	Output packaging type as listed in the "Supported Formats" section. Multiple types can be specified as: OUTPUT_TYPE=<TYPE1>,<TYPE2> Valid options: • MBR_HLS • DASH
ENCODING_TYPE	Group: DASH_HLS_H264 Preset: 960x540_2128kbps DASH_HLS_H264:960x540_2128kbps	Encoding group(s) and preset(s) as listed in the "Supported Formats" section. Set multiple presets for one or more preset groups. Multiple values are separated with comma: ENCODING_TYPE=<GROUP1>:<PRESET1>,<PRESET2>,...;<GROUP2>:<PRESET1>,<PRESET2>,...
VERBOSITY	SILENT	Sets verbose level. Available options are: • SILENT (0) - Prints only error messages • DEFAULT (1) - Default level if verbose level is not specified. Prints information about input file, output file, preset. Prints information about REST API calls (target URLs with parameters, no BODY info). • FULL (2) - Prints all available information, including processing status, file transfer, REST API calls (target URLs with parameters with BODY info), complete information from processing step.
General parameters for CDN upload		
CDN	AKAMAI	Choose a CDN for direct upload. Besides general CDN settings, there are also some vendor specific CDN options for Akamai and Amazon CloudFront. Valid options:

		- AKAMAI - AMAZON
MAX_RETRY_COUNT	5	Number of retries before uploading fails.
UPLOAD_RECONNECT_DELAY_MS	500	Delay between reconnections in milliseconds.
Parameters for publishing to Akamai CDN (see also https://learn.akamai.com)		
ARY_PUBLISHING_URL	DASH: http://post.500002.testconfig.r.akamaientrypoint.net/dash/500002/test79/dash.mpd HLS: http://post.example-i.akamaihd.net/50002/test79	Specify the primary publishing URL to Akamai CDN.
AKAMAI_PRIMARY_PLAYBACK_URL	http://testconfig-i.akamaihd.net/dash/live/500002/test79/dash.mpd	Specify the primary playback URL for Akamai CDN.
AKAMAI_BACKUP_PLAYBACK_URL	http://testconfig-i.akamaihd.net/dash/live/500002-b/test79/dash.mpd	Specify the primary backup URL for Akamai CDN.
AKAMAI_USERNAME		User name for accessing Akamai Akamai CDN
AKAMAI_PASSWORD		Password for accessing Akamai Akamai CDN
Parameters for publishing to Amazon CloudFront CDN (http://aws.amazon.com/documentation/cloudfront/)		
AMAZON_BUCKET		Amazon S3 bucket name.
AMAZON_REGION	us-east-1	Region where bucket is situated.
AMAZON_S3_ENDPOINT		The Path in S3 Bucket is an optional parameter. The output files will be uploaded to the root of your bucket if this is empty. Note: *The bucket must be attached to the appropriate Amazon CloudFront distribution and must have read permissions to be visible in CloudFront.*
AMAZON_ACCESS_KEY_ID		To access certain resources on AWS, users need an Access Key ID and a Secret Access Key.

AMAZON_SECRET_ACCESS_KEY		To access certain resources on AWS, users need an Access Key ID and a Secret Access Key.

5.3 Using properties file

The easiest way to start MainConcept 2GO is by editing the “properties.txt” file to your needs and then executing the “run” script with this file.

```
~/MainConcept/2GO/mc_2go_ott_live_encoder_demo# ./scripts/run.sh scripts/properties.txt
```

The MainConcept 2GO OTT Live Encoder supports the following configuration options:

Option	**Sample value**	**Description**
ACCEPT_EULA	Y	The MainConcept End-User License Agreement (EULA) must be accepted before MainConcept 2GO can start. The license terms for this product can be found in the file "EULA.txt", which is included with the product package. You can accept the EULA by setting the ACCEPT_EULA parameter, e.g. ACCEPT_EULA=Y
CUSTOMER_ID		User's unique identifier, provided by MainConcept after purchase. Customer ID parameter can also be specified more concisely as `CID`
SHARED_PATH_OUT	/path/target_folder	Optional: Folder for the encoded output file. Content is copied to Volume first before uploading to Akamai or CloudFront.
JOBS	/path/to/json_file	Absolute path on host machine to JSON file with job description for immediate processing. The container will be stopped after processing. Note: This parameter is only used when you are planning to use MainConcept 2GO module to perform single job. REST API and server mode are unavailable.
PORT	8082	Specify port for REST API if default port (8080) is already in use. Note: This parameter is useful if MainConcept 2GO module is used in server mode.
AUTOSTART	TRUE, FALSE	Disable/enable job queue processing when Docker runs. FALSE sets the service to a stopped state after the Docker runs. Default value is TRUE.

CUSTOMER_ID		User's unique identifier, provided by MainConcept after purchase. Customer ID parameter can also be specified more concisely as `CID`.
REPORTING_SERVER		Name/IP address and port of self-hosted reporting server. Example: my-reporting-server-ip:443. If not specified, 2GO will use MainConcept's online reporting server. Demo modules always use MainConcept's server.
VERBOSITY	SILENT	Sets verbose level. Available options are: • SILENT (0) - Prints only error messages • DEFAULT (1) - Default level if verbose level is not specified. Prints information about input file, output file, preset. Prints information about REST API calls (target URLs with parameters, no BODY info). • FULL (2) - Prints all available information, including processing status, file transfer, REST API calls (target URLs with parameters with BODY info), complete information from processing step.

5.4 Using properties file

The easiest way to start MainConcept 2GO is by editing the “properties.txt” file to your needs and then executing the “run” script with this file.

```
~/MainConcept/2GO/mc_2go_ott_live_encoder_demo# ./scripts/run.sh scripts/properties.txt
```

To edit the parameters, use a text editor:

```
SHARED_PATH_IN==~/MainConcept/2GO/mc_2go_ott_live_encoder_demo/volume
SHARED_PATH_OUT=~/MainConcept/2GO/mc_2go_ott_live_encoder_demo/volume
URL=rtsp://5.196.138.6:1935/live/nrjbelgique_360p
LOGIN=
PASSWORD=
OUTPUT_TYPE=MBR_HLS
ENCODING_TYPE=DASH_HLS_H264:960x540_2128kbps,1280x720_4628kbps
VERBOSITY=DEFAULT
CDN=AKAMAI
MAX_RETRY_COUNT=5
```

```
UPLOAD_RECONNECT_DELAY_MS=500
AKAMAI_PRIMARY_PUBLISHING_URL=http://post.500002.testconfig.r.akamaientrypoint.net/dash/500002/test79/dash.mpd
AKAMAI_PRIMARY_PLAYBACK_URL=http://testconfig-i.akamaihd.net/dash/live/500002/test79/dash.mpd
AKAMAI_BACKUP_PLAYBACK_URL=http://testconfig-i.akamaihd.net/dash/live/500002-b/test79/dash.mpd
AKAMAI_USERNAME=
AKAMAI_PASSWORD=
```

Using command line options

Instead of editing a properties file, all configuration parameters can also be specified on the command line directly.

```
~/MainConcept/2GO/mc_2go_ott_live_encoder_demo/docker# ./scripts/run.sh \
>    SHARED_PATH_OUT=~/MainConcept/2GO/mc_2go_ott_live_encoder_demo/volume \
>    URL=rtsp://5.196.138.6:1935/live/nrjbelgique_360p \
>    LOGIN= \
>    PASSWORD= \
>    OUTPUT_TYPE=MBR_HLS \
>    ENCODING_TYPE=DASH_HLS_H264:960x540_2128kbps,1280x720_4628kbps \
>    VERBOSITY=DEFAULT█
>    CDN=AKAMAI \
>    MAX_RETRY_COUNT=5 \
>    UPLOAD_RECONNECT_DELAY_MS=500 \
>    AKAMAI_PRIMARY_PUBLISHING_URL=http://post.500002.testconfig.r.akamaientrypoint.net/dash/500002/test79/dash.mpd \
>    AKAMAI_PRIMARY_PLAYBACK_URL=http://testconfig-i.akamaihd.net/dash/live/500002/test79/dash.mpd \
>    AKAMAI_BACKUP_PLAYBACK_URL=http://testconfig-i.akamaihd.net/dash/live/500002-b/test79/dash.mpd \
>    AKAMAI_USERNAME= \
>    AKAMAI_PASSWORD=█
```

Using docker-compose

Docker Compose lets you start multiple replicas of the same image. It also significantly simplifies starting MainConcept 2GO products.

Docker Compose is a separate tool that must be installed in addition to Docker. Please refer to the documentation about how to install and setup Compose: docs.docker.com/compose/

To run MainConcept 2GO with Docker Compose you create a compose file in YAML. All parameters to run the 2GO product via Docker Compose are specified inside this YAML file. To simplify deployment of MainConcept 2GO containers, setting environment variables for the 2GO configuration and then using them in the YAML file is recommended.

This is a basic "docker-compose.yml" file showing how to configure MainConcept 2GO.

```
version: '2.2'
services:
  2go:
    image: 2go_ott_live_encoder_demo
    network_mode: host
    volumes:
    -${2GO_VOLUME}:/volume
    environment:
    -2GO_PARAMS=${2GO_PARAMS}
```

It uses two environment variables 2GO_VOLUME and 2GO_PARAMS. These must be set before starting docker-compose.

```
~# export 2GO_VOLUME=~/MainConcept/2GO/mc_2go_ott_live_encoder_demo/volume/
~# export 2GO_PARAMS="URL=rtsp://5.196.138.6:1935/live/nrjbelgique_360p LOGIN=123
PASSWORD=123 OUTPUT_TYPE=MBR_HLS ENCODING_TYPE=DASH_HLS_H264:960x540_2128kbps"
```

After that starting 2GO using Docker Compose is as easy as:

```
~/MainConcept/2GO/mc_2go_ott_live_encoder_demo# docker-compose up -d
Starting mc2goottliveencoderdemo_2go_1 ...
Starting mc2goottliveencoderdemo_2go_1 ... done
~/MainConcept/2GO/mc_2go_ott_live_encoder_demo#
```

5.5 Stopping MainConcept 2GO execution

MainConcept 2GO containers should be stopped using the convenience “stop” script provided in the scripts folder.

To execute the script on an active container you first need to know the container ID. This can be found from the first column of the docker ps command:

```
~/MainConcept/2GO/mc_2go_ott_live_encoder_demo# docker ps
CONTAINER ID   IMAGE                          COMMAND                      CREATED
bc45686deb0e   mc_2go_ott_live_encoder_demo   "/opt/bin/reporting_..."    4seconds ago
```

Then use the convenience “stop” script in the scripts folder to abort the 2GO execution:

```
~/MainConcept/2GO/mc_2go_ott_live_encoder_demo# ./scripts/stop.sh bc45686deb0e
Stopping the OTT Live Encoder...
OTT Live Encoder stopped.
~/MainConcept/2GO/mc_2go_ott_live_encoder_demo# █
```

If you are using the REST API, you also have the opportunity to shut down the container by using DELETE /service. If a job is currently active, the container is not shut down and this function is ignored. In this case you need to abort the job first using POST /jobs/{jobID}/abort

6. Technical Support

If you need additional assistance, the MainConcept Technical Support team is standing by to help. Send an e-mail to apps.support@mainconcept.com or go to the MainConcept Support page and we'll assist you as quickly as possible.

7. Credits

AAC's HE-AAC and HE-AAC v2 versions are regarded as today's most efficient general perceptual audio codecs. AAC has been standardized by ISO and IEC as part of the MPEG specifications. It is understood that it may be necessary to execute a patent license with the appropriate AAC licensing entities in order to obtain all rights necessary to create and exploit products utilizing AAC and it is recommended to contact the appropriate licensing entities, e.g. Via Licensing (www.vialicensing.com), and negotiate in good faith the adequate contracts, if any.

Fraunhofer Institute for Integrated Circuits IIS
Attention: Audio and Multimedia Departments - MC AAC LL
Am Wolfsmantel 33, 91058 Erlangen, Germany
www.iis.fraunhofer.de/amm
amm-info@iis.fraunhofer.de

MC2GO Product Version: 2.3
Last User Guide Update: June 14, 2021