Nitro GFXEngine API Specifications Document

Version: 0.7.1

Date: 6-Dec-2019

Copyright © 2020 Workflowlabs

Table of Contents

Copyright © 2020 Workflowlabs

1. Introduction

Nitro G-series servers come with a built-in 2D GFX engine with support for basic GFX templates - such as Logo (static and animated), Ticker, Bug, Aston, L-Band, Animations (PNG and TGA sequences) etc.

All GFX assets will be mainly PNG / TGA / TIFF images or sequences so they can be created from any 3rd party Graphics creation tools.

This API specification document describes the communication method and the messages structure for listing, triggering, updating and stopping GFX instances from a 3rd party GFX controller application.

2. Communication method

Copyright © 2020 Workflowlabs

Nitro GFX Engine accepts TCP/IP socket connections from controller applications.

Every Playout Channel will be associated with its own dedicated GFXEngine service waiting for connections on a different TCP/IP port.

Playout Channel	GFX Engine	TCP/IP Port
Channel X	GFXEngine service X	8880
Channel Y	GFXEngine service Y	8881
Channel Z	GFXEngine service Z	8882
….		

The controller application should use the IP address of the Nitro server and the appropriate TCP/IP port number to communicate with the GFXEngine of the required Playout channel.

Socket communication with the GFXEngine can be tested using telnet command as follows:

```
user@HOST:~$ telnet 192.168.1.101 8880
Trying 192.168.1.101...
Connected to 192.168.1.101.
Escape character is '^]'.
{"cmd": "list_tpl"}

RESPONSE:
{"templates": {"LBand": … }}
```

Copyright © 2020 Workflowlabs

3. Messages structure

Nitro GFXEngine communication messages (both commands and responses) follow JSON message format.

https://en.wikipedia.org/wiki/JSON

Every message needs to be on *single line* and should have a newline (“\n” or “\r\n”) character at the end of message (messages delimiter = \n).

[✕] Wrong:

```
{
    "cmd": "stop_cg",
    "_id": "5828f303-95df-4c3a-8da4-94c16f1bb53f",
}
```

[✓] Correct:

```
{ "cmd": "stop_cg", "_id": "5828f303-95df-4c3a-8da4-94c16f1bb53f" }
```

Copyright © 2020 Workflowlabs

4. Commands and Responses

4.1. List Templates

This command is used for listing the GFX templates currently available on the GFXEngine. The response message is a JSON object with key = template_id and value = template object.

Each template object will have following fields:

- property-fields: List of available properties fields on the Template in triplet format: [property_name, data_type, property_descr]
- data-fields: List of available data fields on the Template in triplet format: [data_field_name, data_type, data_field_descr]

Command:

```
{"cmd": "list_tpl"}
```

Example Response:


```
{"templates": {"LBand": {"property-fields": [["left", "int", "Number of pixels to
squeeze from left"], ["bottom", "int", "Number of pixels to squeeze from bottom"],
["right", "int", "Number of pixels to squeeze from right"], ["top", "int", "Number
of pixels to squeeze from top"], ["trans_dur", "int", "DVE transition duration (in
number of frames)"]], "type": "l-band", "data-fields": [["image", "path", "LBand
image path"]]}, "clock": {"property-fields": [["x", "int", "X coordinate"], ["y",
"int", "Y coordinate"], ["bg_color", "rgb", "Background color"], ["text_color",
"rgb", "Text color"], ["font_family", "font", "Font family"], ["font_size", "int",
"Font size"]], "type": "clock", "data-fields": [["timezone", "str", "Timezone
(example: \"Asia/Kolkata\", \"US/Eastern\" etc\")"], ["24hrs", "bool", "Display
time as 24-hours clock? (yes/no)"], ["seconds", "bool", "Should display seconds?
(yes/no)"], ["fmt", "str", "Time display format (optional)"]]}, "png_sequence":
{"data-fields": [], "property-fields": []}, "logo": {"property-fields": [["x",
"int", "X coordinate of top-left corner of logo image"], ["y", "int", "Y
coordinate of top-left corner of logo image"], ["width", "int", "Desired width of
logo image in pixels (optional)"], ["height", "int", "Desired height of logo image
in pixels (optional)"], ["alpha", "float", "Transparency level, between 0
(transparent) and 1 (opaque)"]], "type": "logo", "data-fields": [["image", "path",
"Logo image path"]]}, "ticker": {"property-fields": [["x", "int", "X coordinate of
top-left corner"], ["y", "int", "Y coordinate of top-left corner"], ["width",
"int", "Width of ticker (in pixels)"], ["height", "int", "Height of ticker (in
pixels)"], ["bg_image", "path", "Background PNG Image (optional)"], ["bg_color",
"rgb", "Background color"], ["text_color", "rgb", "Text color"], ["font_family",
"font", "Font family"], ["font_size", "int", "Font size"], ["speed", "int", "Speed
of ticker (min: 1, max: 100)"], ["sep", "str", "Separator text"]], "type":
"scroll", "data-fields": [["line1", "str", ""], ["line2", "str", ""], ["line3",
"str", ""], ["line4", "str", ""], ["line5", "str", ""], ["line6", "str", ""],
["line7", "str", ""], ["line8", "str", ""], ["line9", "str", ""], ["line10",
"str", ""], ["line11", "str", ""], ["line12", "str", ""], ["line13", "str", ""],
["line14", "str", ""], ["line15", "str", ""], ["line16", "str", ""], ["line17",
"str", ""], ["line18", "str", ""], ["line19", "str", ""], ["line20", "str",
""]]}}}
```


Copyright © 2020 Workflowlabs

4.2. List Resources

This command is used for listing the GFX resources currently uploaded and known to the GFXEngine. The response message is a list of JSON objects having path and type information about each resource.

Command:

```
{"cmd": "list_res"}
```

Example Response:


```
{"resources": [{"path": "L-bands/ORTHO1.png", "type": "PNG Image"}, {"path": "L-
bands/SHOPPER.png", "type": "PNG Image"}, {"path": "TickerBG/Ticker bottom.png",
"type": "PNG Image"}, {"path": "Logo/Republic_TV_short.png", "type": "PNG Image"},
{"path": "Logo/Gravity deep icon.png", "type": "PNG Image"}, {"path": "Logo/R-
Bharat-Logo.png", "type": "PNG Image"}, {"path": "Logo/R-Republic-Bharat-
Logo.png", "type": "PNG Image"}, {"path": "Logo/Republic_Bharat_Logo.png", "type":
"PNG Image"}, {"path": "Logo/Republic_TV_long.png", "type": "PNG Image"}, {"path":
"Bugs/Powered by.png", "type": "PNG Image”}]}
```


Copyright © 2020 Workflowlabs

4.3. Play CG

This command is used for triggering (loading and playing) a CG template.

Following are the important mandatory fields:

- cg-id: CG template_id that needs to be played (one of the keys of the JSON object returned by “list_tpl” command)

  Example: “logo”, “bug”, “ticker”, “clock”, “LBand” etc

- _id: CG instance id chosen by controller application which can be later used to refer to this CG instance while using the “update_cg” and “stop_cg” commands.

- layer: Layer on which this CG instance needs to be loaded (layer 1 will be the top-most layer, followed by layer 2 and so on)

Example Command 1: Load and play a static logo image uploaded on Nitro server at the location “Logo/R-Republic-Bharat-Logo.png" positioned at X coordinate = 1558 and Y coordinate = 64 and using the actual width and height of the PNG logo image.

```
{ "cmd" : "play_cg", "cg-id" : "logo", "_id" : "main_logo", "layer" : 1,
"properties" : {"x" : 1558, "y" : 64}, "data" : { "image" : "Logo/R-
Republic-Bharat-Logo.png" }}
```

Response:

```
{ "cmd": "play_cg", "_id": "main_logo", "status": "OK" }
```

Example Command 2: Load and play a static bug image uploaded on Nitro server at the location “Bugs/Powered by TATA.png" positioned at X coordinate = 1700 and Y coordinate = 820 and resizing the original PNG image to 200x200 and with transparency level = 75%

```
{ "cmd" : "play_cg", "cg-id" : "bug", "_id" : "powered_by_tata_bug",
"layer" : 5, "properties" : {"x" : 1700, "y" : 820, "width": 200,
"height": 200, "alpha": 0.75 }, "data" : { "image" : "Bugs/Powered by
TATA.png" }}
```

Response:

```
{ "cmd": "play_cg", "_id": "powered_by_tata_bug", "status": "OK" }
```

Example Command 3: Load and play a static LBand commercial uploaded on Nitro server at the location “L-bands/SHOPPER.png" for a duration of 10 seconds with DVE squeeze of 320 pixels towards left and 180 pixels towards bottom and with a squeeze/unsqueeze duration of 1 seconds each (= 25 frames as per PAL standard)

```
{"cmd" : "play_cg", "cg-id" : "LBand", "_id" : "lband1", "layer" : 4,
"duration" : 10, "properties" : { "left": 320, "bottom" : 180, "trans_dur"
: 25}, "data" : { "image" : "L-bands/SHOPPER.png" }}
```

Response:

```
{ "cmd": "play_cg", "_id": "lband1", "status": "OK" }
```

Copyright © 2020 Workflowlabs

4.4. Stop CG

This command is used for stopping and unloading a CG instance

This command requires the “_id” (CG instance id) field which was used in the play_cg command to load the CG instance.

Example Command 1: Stop and unload the logo instance of the Play CG example 1

```
{ "cmd" : "stop_cg", "_id" : "main_logo" }
```

Response:

```
{ "cmd": "stop_cg", "_id": "main_logo", "status": "OK" }
```

Copyright © 2020 Workflowlabs

4.5. Update CG

This command is used for dynamically updating the properties and data fields of a playing CG instance (without stopping and playing it again).

This command also requires the “_id” (CG instance id) field which was used in the play_cg command to load the CG instance.

Rest of the fields of this command are exactly same as play_cg command.

Example Command 1: Update the X and Y coordinates of the main_logo instance of the Play CG example 1 without stopping and playing it.

```
{ "cmd" : "update_cg", "cg-id" : "logo", "_id" : "main_logo", "layer" : 1, "properties" : {"x" : 1600, "y" : 70}, "data" : { "image" : "Logo/R-Republic-Bharat-Logo.png" }}
```

Response:

```
{ "cmd": "update_cg", "_id": "main_logo", "status": "OK" }
```

Copyright © 2020 Workflowlabs

4.6. Stop All CG

This command is used for stopping all active CG instances except those playing on VIP layers (configurable, by default layers 1 and 2 will be configured as VIP layers).

Command:

```
{ "cmd": "stop_all_cg" }
```

Response:

```
{ "cmd": "stop_all_cg", "status": "OK" }
```

Copyright © 2020 Workflowlabs