Media Production Suite

License

Web API Specification

Version 1.0

Nov, 2025

Panasonic Entertainment & Communication Co., Ltd.

目次

1 Introduction

1.1 Purpose of this document

This document defines the communication interface specifications for the Web API that obtains the license status of paid plug-ins for Media Production Suite software.

1.2 Notes on using Web API

Nothing in particular

1.3 Change history

The update history of this document and the corresponding Media Production Suite (MPS) versions.

Table 1.3-1 Change History

Document version	Supported MPS Versions	Changing Points
1.0	1.11 or later	first edition

2 Interface

2.1 Specification

The communication IF specified as a Web API in this manual conforms to the communication specification of HTTP 1.1, and the control over this software is realized as an HTTP request to a URL on the Web server. All HTTP requests are GET methods.

2.2 Format

The format (URL) of the control command provided as Web API is as follows.

[Send]

http://[IP Address]:[Port No]/cgi-bin/license?cmd=[Command]&[Parameters]=[Value]&...

table 2.2-1 Transmission Format Details

No	name	explanation
1	IP Address	IP address of the web application
2	Port No	The listening port number of the Web application. Currently fixed at 1337.
3	command	The control command string. “2.3 List of Supported Commands” specifies a list of commands that can be used in .
4	parameter	Control command parameters. Use "&" to allow multiple specifications.
5	value	The value to set for the control command parameter.

[Receive]

The format of the response data returned to the command issuer is JSON format (content_type "application/json").

For the contents of the response data, please refer to each section in “3 Command Details”.

2.3 List of Supported Commands

The following is a list of commands supported.

For more information about each command, please refer to “3 Command Details”.

table 2.3-1 List of Supported Commands

No	name	explanation
1	Get License Data	Get license status of all paid plugins
2		
3		

3 Command Details

3.1 Get License Data

Description

The Get License Data command is used to retrieve the license status of all paid plug-ins in the Media Production Suite software.

Send Parameters

The send parameters are listed below.

table 3.1-1 List of Transmission Parameters

name	use	format	explanation
cmd	Mandatory	String	GetLicenseData

Command usage example:

This example shows how to send a command to Media Production Suite software running on a PC with IP address 192.168.0.200.

http://192.168.0.200:1337/cgi-bin/license?cmd=GetLicenseData

Command Response

The command response will be the following JSON data:

table 3.1-2 Response JSON Definition

JSON key	format	explanation
Command	String	Command sent
Response	String	ack / nack
NACKDetail	String	If Response is nack, return details
LicenseData	An array of JSON objects	If the response is ack: An array of JSON objects defined in "3.1-3 LicenseData Object Structure" below. If the response is nack: This key does not exist.

table 3.1-3 License Data Object Structure

JSON key	format	explanation
PluginName	String	Plugin Name
LicenseState	String	A string indicating the license status. *For plugins that allow multiple license registrations, this indicates the combined status of all licenses. Initial: Initial state (license invalid) Activated: Activated (license valid) Deactivated: Deactivated (license invalid) In Trial: In trial period (license valid) Trial Expired: Trial period expired (license invalid) License Expired: License expired (license invalid) Duplicated: Activation information from another PC is being used (license invalid)
RemainDays	Integer value	Remaining days of license (for time-limited licenses/trial periods) *If multiple time-limited licenses are registered, the remaining days of the license with the shortest remaining days will be returned. *If the activated license does not have an expiration date and is not in the trial period, this value will be null.
TotalLicenseCount	Integer value	Total number of active licenses *When using the Auto Tracking (SF100), Auto Tracking (SF200), or Advanced Auto Framing plugin, this value stores the number of cameras available for tracking. For other plugins, this value is 1 if the license is active and 0 if the license is inactive.
UsedLicenseCount	Integer value	Number of licenses in use *When using the Auto Tracking (SF100), Auto Tracking (SF200), or Advanced Auto Framing plugin, this value stores the number of cameras currently being used by the plugin. For other plugins, this value is 1 if the license is active and 0 if the license is inactive.
LicensedDevice	An array of JSON objects	When the plug-in is Auto Tracking (SF100), Auto Tracking (SF200), or Advanced Auto Framing If there are cameras in use with the plugin's auto tracking, an array of JSON objects defined in table "3.1-4 LicensedDevice object structure" below will be stored. If there are no cameras in use with auto tracking, this value will be null. When the plug-in is other than Auto Tracking (SF100), Auto Tracking (SF200), or Advanced Auto Framing this value will be null

table 3.1-4 LicensedDevice Object Structure

JSON key	format	explanation
IP Address	String	Device IP address
Name	String	Device Name

Command response example (normal response)


```
{
  "Command": "GetLicenseData",
  "Response": "ack",
  "LicenseData": [
    {
      "LicenseState": "Initial",
      "LicensedDevice": null,
      "PluginName": "Visual Preset",
      "RemainDays": null,
      "TotalLicenseCount": 0,
      "UsedLicenseCount": 0
    },
    {
      "LicenseState": "Deactivated",
      "LicensedDevice": null,
      "PluginName": "Auto Tracking (SF100)",
      "RemainDays": null,
      "TotalLicenseCount": 0,
      "UsedLicenseCount": 0
    },
    {
      "LicenseState": "In Trial",
      "LicensedDevice": [
          {
              "IP Address": "192.168.0.10",
              "Name": "AW-UE150"
          }
      ],
      "PluginName": "Auto Tracking (SF200)",
```



```
      "RemainDays": 16,
      "TotalLicenseCount": 7,
      "UsedLicenseCount": 1
    },
    {
      "LicenseState": "Activated",
      "LicensedDevice": null,
      "PluginName": "Video Mixer",
      "RemainDays": null,
      "TotalLicenseCount": 1,
      "UsedLicenseCount": 1
    },
    {
      "LicenseState": "Activated",
      "LicensedDevice": [
            {
                "IP Address": "192.168.0.20",
                "Name": "AW-UE160_1"
            },
            {
                "IP Address": "192.168.0.21",
                "Name": "AW-UE160_2"
            },
            {
                "IP Address": "192.168.0.22",
                "Name": "AW-UE160_3"
            }
      ],
      "PluginName": "Advanced Auto Framing",
      "RemainDays": 21,
      "TotalLicenseCount": 4,
      "UsedLicenseCount": 3
    }
  ]
}
```


Command response example (abnormal response)


```
{
  "Command": "GetLicenseData ",
  "NACKDetail": "unknown error",
  "Response": "nack"
}
```


4 Abnormal Handling

The following JSON data will be sent in response to an abnormal command not specified in "3 Command Details."

table 4.1 Abnormal response JSON definition

Object	JSON key	format	explanation
	Command	String	Command sent
	Response	String	nack
	NACKDetail	String	return error details

Command response example

```
{
  "Command": "LicenseCommand",
  "NACKDetail":"not supported API command",
  "Response": "nack"
}
```