

# **Media Production Suite**

## **License 状態取得**

## **Web API 仕様書**

**Version 1.0**

**Nov, 2025**

**Panasonic Entertainment & Communication Co., Ltd.**

## 目次

1	はじめに	3
1.1	本書の目的	3
1.2	Web API 使用時の注意事項	3
1.3	更新履歴	3
2	インターフェース	4
2.1	仕様	4
2.2	フォーマット	4
2.3	対応コマンド一覧	5
3	コマンド詳細	6
3.1	Get License Data	6
4	異常系処理	11

# 1 はじめに

## 1.1 本書の目的

本書は、Media Production Suite ソフトウェアの有償プラグインのライセンス状態を取得する Web API の通信 IF 仕様を規定したものです。

## 1.2 Web API 使用時の注意事項

特になし

## 1.3 更新履歴

本書の更新履歴と対応する Media Production Suite(MPS)のバージョンを記載します。

表 1.3-1 更新履歴

文書バージョン	サポートする MPS バージョン	変化点
1.0	1.11 以降	初版

## 2 インタフェース

### 2.1 仕様

本書にて Web API として規定する通信 IF は HTTP1.1 の通信仕様に準拠し、本ソフトウェアに対する制御を Web サーバ上の URL に対する HTTP リクエストとして実現しています。HTTP リクエストは、全て GET メソッドとなります。

### 2.2 フォーマット

Web API として提供する制御コマンドのフォーマット(URL)は以下となります。

[送信]

**http://[IP Address]:[Port No]/cgi-bin/license?cmd=[コマンド]&[パラメータ]=[値]&...**

表 2.2-1 送信フォーマット詳細

No	名称	説明
1	IP Address	Web Application の IP アドレス
2	Port No	Web Application の待受用ポート番号。 現在は 1337 固定。
3	コマンド	制御コマンド文字列。 2.3 対応コマンド一覧において使用可能なコマンド一覧を規定する。
4	パラメータ	制御コマンドパラメータ。"amp;"を使用し、複数指定可能とする。 各コマンドにおいて規定されるため、3 コマンド詳細の各節を参照ください。
5	値	制御コマンドパラメータに設定する値。 各コマンドにおいて規定されるため、3 コマンド詳細の各節を参照ください。

[受信]

コマンド発行元に返されるレスポンスデータのフォーマットは JSON 形式(content\_type "application/json")です。

レスポンスデータの内容は 3 コマンド詳細の各節を参照ください。

## 2.3 対応コマンド一覧

対応するコマンドの一覧は以下となります。

各コマンドの詳細は 3 コマンド詳細を参照ください。

表 2.3-1 対応コマンド一覧

No	名称	説明
1	Get License Data	すべて有償プラグインのライセンス状態取得
2		
3		

## 3 コマンド詳細

### 3.1 Get License Data

#### 説明

Get License Data コマンドは、Media Production Suite ソフトウェアのすべての有償プラグインのライセンス状態を取得するときに使用します。

#### 送信パラメータ

送信パラメータの一覧は以下になります。

表 3.1-1 送信パラメータ一覧

名称	使用	形式	説明
cmd	必須	文字列	GetLicenseData

#### コマンド使用例 :

IP アドレス 192.168.0.200 の PC で実行されている Media Production Suite ソフトウェアにコマンドを送信する場合の例です。

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

#### コマンドレスポンス

コマンドレスポンスは以下の JSON データになります。

表 3.1-2 レスポンス JSON 定義

JSON key	形式	説明
Command	文字列	送信されたコマンド
Response	文字列	ack / nack
NACKDetail	文字列	Responseがnackの場合、詳細を返す
LicenseData	JSONオブジェクトの配列	Responseがackの場合 下記表「3.1-3 LicenseDataオブジェクトの構造」で 定義されるJSONオブジェクトの配列  Responseがnackの場合 本Keyは存在しない

表 3.1-3 LicenseData オブジェクトの構造

JSON key	形式	説明
PluginName	文字列	プラグインの名称
LicenseState	文字列	<p>ライセンス状態を示す文字列          ※ライセンスを複数登録可能なプラグインの場合、すべてのライセンスを統合した状態を示す</p> <p>Initial : 初期状態 (ライセンス無効)          Activated : アクティベーション済 (ライセンス有効)          Deactivated : アクティベーション解除済 (ライセンス無効)          In Trial : 試用期間中 (ライセンス有効)          Trial Expired : 試用期間切れ (ライセンス無効)          License Expired : ライセンス期限切れ (ライセンス無効)          Duplicated : 他のPCのアクティベーション情報が使用されている (ライセンス無効)</p>
RemainDays	整数値	<p>ライセンスの残日数 (期間限定ライセンス/試用期間中のとき)</p> <p>※期間限定ライセンスが複数登録されている場合、残り日数が最も短いライセンスの残日数が返される</p> <p>※アクティベートされているライセンスが期限を持たない場合、かつ試用期間中ではない場合、本値はnullとなる。</p>
TotalLicenseCount	整数値	<p>有効な全ライセンス数</p> <p>※プラグインがAuto Tracking (SF100), Auto Tracking (SF200), Advanced Auto Framingのとき、本値には追尾に使用可能なカメラ台数が格納される。それ以外のプラグインの場合、本値はライセンス有効時 : 1、ライセンス無効時 : 0となる。</p>
UsedLicenseCount	整数値	<p>使用中のライセンス数</p> <p>※プラグインがAuto Tracking (SF100), Auto Tracking (SF200), Advanced Auto Framingのとき、本値にはプラグインの自動追尾で使用中のカメラ台数が格納される。それ以外のプラグインの場合、本値はライセンス有効時 : 1、ライセンス無効時 : 0となる。</p>
LicensedDevice	JSONオブジェクトの配列	<ul style="list-style-type: none"> <li>・ プラグインがAuto Tracking (SF100), Auto Tracking (SF200), Advanced Auto Framingのとき           <p>プラグインの自動追尾で使用中のカメラがある場合、下記表「3.1-4 LicensedDeviceオブジェクトの構造」で定義されるJSONオブジェクトの配列が格納される。</p> <p>自動追尾で使用中のカメラが無い場合、本値はnullとなる。</p> </li> <li>・ プラグインがAuto Tracking (SF100), Auto Tracking (SF200), Advanced Auto Framing以外のとき           <p>本値はnullとなる。</p> </li> </ul>

表 3.1-4 LicensedDevice オブジェクトの構造

JSON key	形式	説明
IP Address	文字列	デバイスのIPアドレス
Name	文字列	デバイスの名称

## コマンドレスポンス例（正常応答）

```
{
  "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  
}  
]
```

コマンドレスポンス例（異常応答）

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

## 4 異常系処理

「3 コマンド詳細」に規定されていない、異常なコマンド送信に対する受信レスポンスには、以下の JSON データが送信されます。

表 4.1 異常レスポンス JSON 定義

Object	JSON key	形式	説明
	Command	文字列	送信されたコマンド文字列
	Response	文字列	nack
	NACKDetail	文字列	エラーの詳細を返す

コマンドレスポンス例

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