# **UDP Commands User Manual**

Model No. ET-FMP50 ET-FMP20 ET-SBFMP10





# **Panasonic**

## ■How to use UDP commands

## Overview

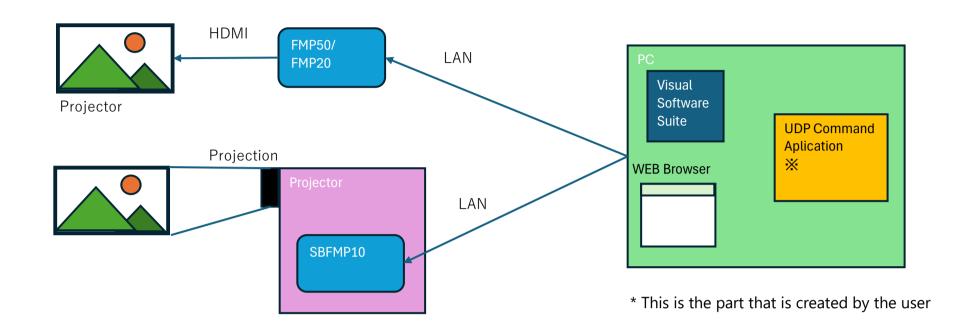
Media playback operations using the Timeline function of the media processor (hereinafter referred to as "FMP") can be performed with commands using UDP communication.

Media playback is performed on a playlist (\*) basis.

Up to four playlists can be registered to FMP, and you can switch between registered playlists to perform media playback operations. It also allows synchronous playback of media by multiple FMPs.

\*Playlists can contain multiple individual media (videos and still images), and playback operations can be performed as a single piece of media.

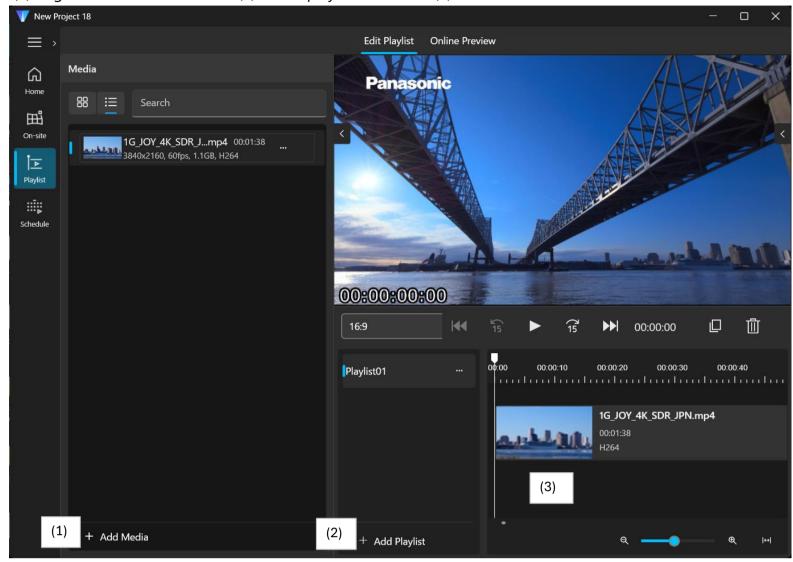
# •Configuration diagram



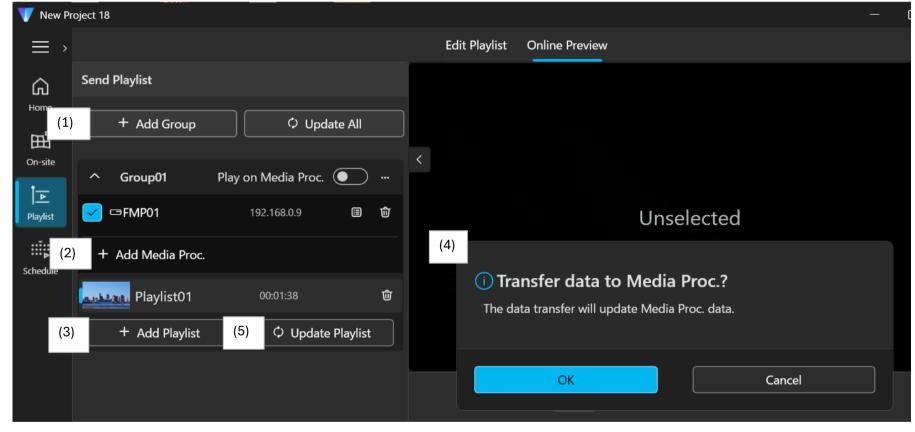
• Create playlists using Visual Software Suite (VSS) and send playlists to FMP

\*For details, refer to the Visual Software Suite (VSS) manual.

- 1. Playlist creation is done on the playlist editing screen below.
  - (1) Add media to be registered in the playlist.
  - (2) Create a playlist.
  - (3) Register the media added in (1) to the playlist created in (2).

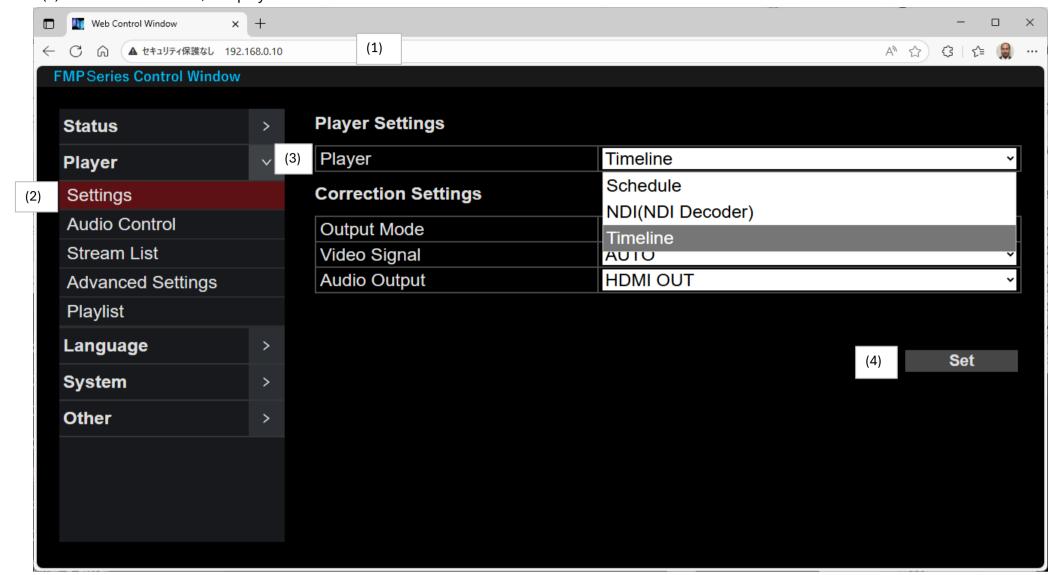


- 2. The created playlist will be sent to FMP on the online preview screen.
  - (1) Add a group to send the playlist to.
  - (2) Add FMP as a media processor to send playlists.
  - (3) Add the playlist created by the playlist editor.
  - (4) After (3) is implemented, a pop-up window for data transfer will appear, so click "OK" to send the playlist to FMP.
  - (5) If the pop-up does not appear, click "Update Playlist" to display the pop-up (4). (Hereafter, the same operation as (4) is performed.)



## ●In the WEB (FMP Series Control Window),how to set the Timeline mode

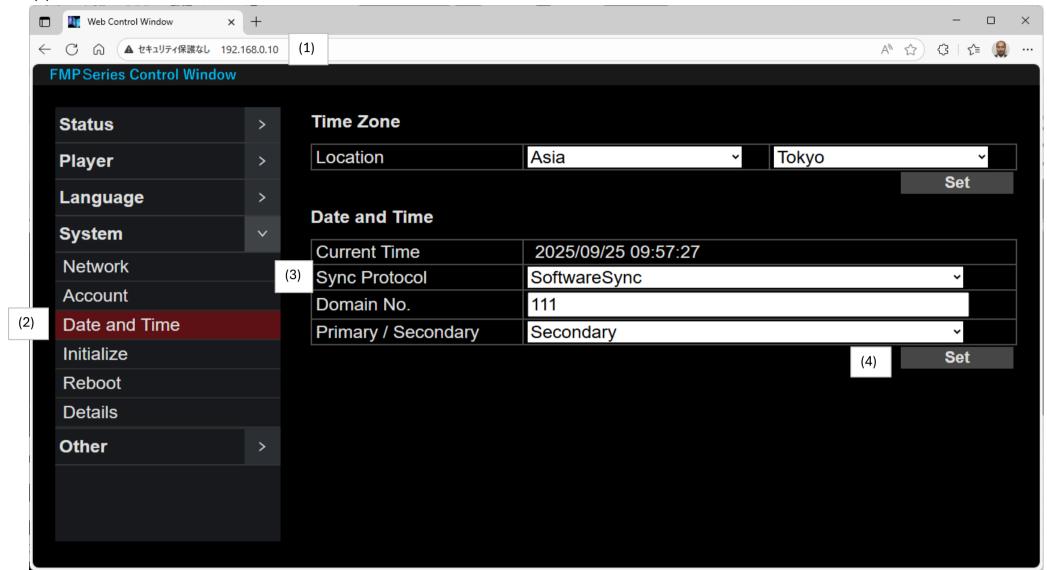
- (1) Use a web browser to access the IP address of the FMP.
- (2) Click "Player"  $\rightarrow$  "Settings" from the left menu to move to the following screen.
- (3) Select "Timeline" from "Player" in the player settings.
- (4) Click the "Set" button.
- (5) After a few seconds, the player "Timeline" starts on FMP.



\* This section describes the settings for synchronized media playback by two media processors. Please refer to the FMP manual for details.

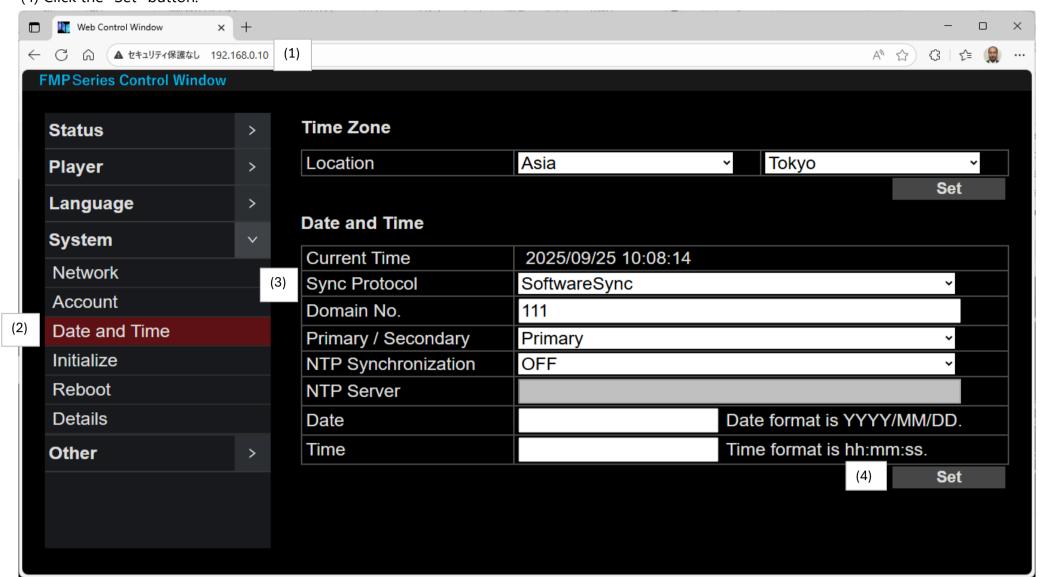
## Secondary side

- (1) With a web browser, access the IP address of the FMP to be used as a secondary.
- (2) Click "System"  $\rightarrow$  "Date and Time" from the left menu to move to the following screen.
- (3) Select "SoftwareSync" from "Synchronization Protocol" for date and time, Set the "Domain Number" in the player settings to any value ("111" here), Select "Secondary" from "Primary/Secondary".
- (4) Click the "Set" button.



## Primary side

- (1) Use a web browser to access the IP address of the FMP to be the primary.
- (2) Click "System"  $\rightarrow$  "Date and Time" from the left menu to move to the following screen.
- (3) Select "SoftwareSync" from "Synchronization Protocol" for date and time, Set the "Domain Number" in the player settings to any value ("111" here), Select "Primary" from "Primary/Secondary".
- (4) Click the "Set" button.



## • UDP Command Specification

## **FMP side connection conditions**

The FMP side is waiting at the following sockets.

IP address: IP address set in FMP Listening port number: **65432** Source address: **INADDR\_ANY** 

Maximum transmit and receive size: 1024byte

## **List of operation commands**

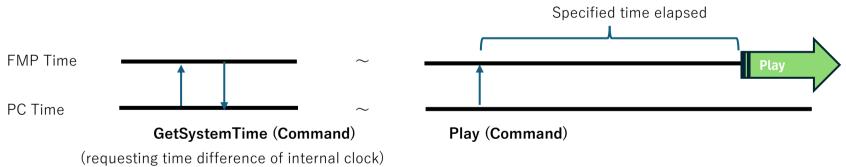
### 情報系Command一覧

command	description
GetSystemTime	Time adjustment between UDP
	command execution side and FMP side
Play	Playlist playback
Pause	Stop playlist playback
	Stop playback at the specified position
Stop	Stop playlist playback
	Playback stops when the command is received
	(You do not need to specify the stop position.)
Seek	Seeking playlist playback
Skip	Skipping media in playlists
SelectPlayList	Playlist Selection
Loop	Playlist loop settings
DisconnectReq	Restarting the Timeline Module
ResetPlaylist	Reload a playlist
SetSyncTime	FMP Time adjustment setting (initial value is 3 seconds)

報系Command一覧	
command	description
GetPlaylistInfo	Retrieving Playlist Information
GetEachPlaylistInfo	Retrieving media information in a playlist
GetPlayingInfo	Retrieving playlist playback status

# Steps to play a playlist (when specifying a synchronized playback start time)

# [Prosessing Image]



- (1) Execute GetSystemTime (time adjustment between UDP command execution side and FMP side)
- (2) Obtain the time difference between the internal clock of the PC executing the UDP command and the FMP side, and add it to the parameter "playback start synchronization time" of the subsequent UDP command.
- (3) Execute SelectPlayList (playlist selection)
- (4) Execute Loop (playlist loop setting)
- (5) Execute Play (playlist playback)

After that, the playlist playback is stopped with the Pause, Stop, Seek, and Skip commands.

# [Command Example]

- (1) GetSystemTime CurrentTimeVss=1715237197598000
- (2)
- (3) SelectPlayList PlayListName=Playlist01
- (4) Loop EnableLoop=0
- (5) Play SyncTime=1727684152203481&PlayPosition=0

# PC Time No need to execute (3 seconds elapsed (changeable with SetSyncTime)) Play (Command) (no SyncTime)

- (1) Execute SelectPlayList (playlist selection)
- (2) Execute Loop (playlist loop setting)
- (3) Execute Play (playlist playback)

After that, the playlist playback is stopped with the Stop, Seek, and Skip commands.

The FMP will perform playback, etc., 3 seconds after receiving the command (modifiable with SetSyncTime).

## [Command Example]

- (1) SelectPlayList PlayListName=Playlist01
- (2) Loop EnableLoop=0
- (3) Play

## Checking the playback status of a playlist

- (1) Set the playlist playback state
- (2) Execute GetPlayingInfo (Get playlist playback status) at any time
- (3) Judge the playback status of the playlist by increasing the CurrentPosition (current position of the playlist) in the return of GetPlayingInfo (get playlist playback status)

## Reload a playlist after updating a playlist

- (1) Perform playlist update in FMP from Visual Software Suite (VSS)
- (2) Execute DisconnectReq (module restart) or ResetPlaylist (reload playlist)
- (3) Execute GetPlaylistInfo (to get playlist information) and GetEachPlaylistInfo (to get media information in the playlist) and confirm that the playlist has been updated
- (4) Execute "Procedure for playlist playback"

## API List of UDP Command

Time adjustment between UDP command execution side and FMP side

#### Command GetSystemTime

## Description

To determine the time difference between the internal clock on the PC executing the UDP command and the FMP side.

The time difference should be calculated in microseconds, and this value should be added to the parameter "playback start synchronization time" of the subsequent UDP command.

Request			
Parameter	Required	Description	Remarks
GetSystemTime	0	Command	
CurrentTimeVss	0		Linux time format (including usec after the decimal point) multiplied by 1000000

## Example Request

GetSystemTime CurrentTimeVss=1715237197598000

- \*1: The separator between UDP commands and parameters is " " (space), and the value of the parameter is after "=". \*2: 1715237197598000 = 2024/05/09 06:46:37.598000

Return Value			
Parameter	Always	Description	Remarks
OK	0	Result	Always [OK] is returnd
CurrentTimeVss	0	Execution time set at the time of request	
CurrentTimeFmp	0		Linux time format (including usec after the decimal point) multiplied by 1000000

# Example Return

On Success

OK CurrentTimeVss=1715237197598000&CurrentTimeFmp=1715237196004200

- \*1: The separator between the execution result and the parameter is "" (space), the delimiter between the parameters is "&", and the value of the parameter is after "=".
- \*2: 1715237197598000=2024/05/09 06:46:37.598000

1715237196004200 = 2024/05/09 06:46:36.004200

## On Failure

none

Playlist playback
-------------------

#### Command Play

# Description

To play the specified playlist registered in FMP.

Request			
Parameter	Required	Description	Remarks
Play	0	Command	
		IRADANARALIAN SIARI IIMA (LINUVIIMA + USAC)	Linux time format (including usec after the decimal point) multiplied by 1000000
SyncTime			Specify a time that is at least 1 seconds ahead of the current time  Also, include the time difference in GetSystemTime
PlayPosition			Linux time format (including usec after the decimal point) multiplied by 1000000

# Example Request

When specifying synchronized playback start time: Play SyncTime=1727684152203481&PlayPosition=0

When not specifying synchronized playback start time: Play PlayPosition=0 or Play

- \*1: The separator between the execution result and the parameter is "" (space), the delimiter between the parameters is "&", and the value of the parameter is after "=".
- \*2: If the playback position of the playlist is set to something other than 0, playback can be started from the middle of the playlist. \*3:If synchronized playback start time is not specified, the "SyncTime" and "PlayPosition" parameters can be omitted. In that case, the FMP will start playback from the beginning of the playlist 3 seconds after receiving this command (changeable with SetSyncTime).

Return Value			
Parameter	Always	Description	Remarks
OK/NG	0	Result	
NG Parameter		Cause of NG	

## Example Return On Success

# On Failure

- \*: NG may be due to the following factors.
- Playlist is not specified.
- The playback start synchronization time points to the past.
- The playback position of the playlist points outside the range of the playlist.

Command	In.		
	Pause		
<b>Description</b> To stop playback o	f a playlist	at the specified stop position of the playlist.	
Stop while display	ing the vide	eo of the stop position.	
	<b>_</b>		
Request Parameter	Required	Description	Remarks
Pause	0	Command	Linux time format (including usec after the decimal point) multiplic
StopPosition	0	Playlist stop position (elapsed time from the beginning)	by 1000000
Example Request Pause StopPosition	=3800490		
		commands and parameters is "" (space), and the value of	the narameter is after "-"
The Separator b	CLWCCII ODI (	(Space), and the value of	the parameter is after — .
Return Value			
Parameter OK/NG	Always	<b>Description</b> Result	Remarks
VII/ ITM			
Example Return On Success			
OK Success			
On Failure			
NG			
Nu			
*: NG may be due t	o the follow	ving factors.	
- The Stop posit	ion of the p	playlist points outside the range of the playlist.	
	Stop playl	ist playback	
	Stop playl	ist playback	
Command	Stop playl	ist playback	
Description	Stop		
<b>Description</b> To stop playback o	Stop       f the play!	ist at the position at the time of receiving the command.	
<b>Description</b> To stop playback o	Stop       f the play!		
<b>Description</b> To stop playback o	Stop       f the play!	ist at the position at the time of receiving the command.	
Description To stop playback o The video at the t	Stop  f the playl ime of rece	st at the position at the time of receiving the command.  Iving the command is displayed and stopped.	
Description To stop playback o The video at the t  Request Parameter	Stop  f the playl ime of rece	ist at the position at the time of receiving the command.	Remarks
Description To stop playback o The video at the t  Request Parameter	Stop  f the playl ime of rece  Required	ist at the position at the time of receiving the command.  Iving the command is displayed and stopped.  Description	Remarks
Description To stop playback o The video at the t  Request Parameter	Stop  f the playl ime of rece  Required	ist at the position at the time of receiving the command.  Iving the command is displayed and stopped.  Description	Remarks
Description To stop playback o The video at the t  Request Parameter	Stop  f the playl ime of rece  Required	ist at the position at the time of receiving the command.  Iving the command is displayed and stopped.  Description	Remarks
Description To stop playback o The video at the t  Request Parameter Stop  Example Request	Stop  f the playl ime of rece  Required	ist at the position at the time of receiving the command.  Iving the command is displayed and stopped.  Description	Remarks
Description To stop playback o The video at the t  Request Parameter Stop  Example Request	Stop  f the playl ime of rece  Required	ist at the position at the time of receiving the command.  Iving the command is displayed and stopped.  Description	Remarks
Description To stop playback o The video at the t  Request Parameter Stop  Example Request	Stop  f the playl ime of rece  Required	ist at the position at the time of receiving the command.  Iving the command is displayed and stopped.  Description	Remarks
Description To stop playback o The video at the t  Request Parameter Stop  Example Request	Stop  f the playl ime of rece  Required	ist at the position at the time of receiving the command.  Iving the command is displayed and stopped.  Description	Remarks
Description To stop playback of The video at the tension of the video at the video at the tension of the video at	Stop  f the playl ime of rece  Required	st at the position at the time of receiving the command.  In the command is displayed and stopped.  Description  Command	
Description To stop playback of The video at the tension of the video at	Stop  f the playl ime of rece  Required	ist at the position at the time of receiving the command.  Iving the command is displayed and stopped.  Description	Remarks
Description To stop playback of The video at the tendence of the video at the video at the tendence of the video at the video at the video at the video at the tendence of the video at the video a	Stop  f the playl ime of rece  Required  Always  O	st at the position at the time of receiving the command. ving the command is displayed and stopped.  Description Command  Description	Remarks  Linux time format (including usec after the decimal point) multiplic
Description To stop playback of The video at the tendence of the video at the vid	Stop  f the playl ime of rece  Required  O  Always	st at the position at the time of receiving the command.  ving the command is displayed and stopped.  Description  Command  Description  Result	Remarks  Linux time format (including usec after the decimal point) multiplic by 1000000  If you set this value to the playback position of playlist playback.
Command  Description To stop playback of The video at the text the video at the video at the text the video at the text the video at the vi	Stop  f the playl ime of rece  Required  Always  O	st at the position at the time of receiving the command.  ving the command is displayed and stopped.  Description  Command  Description  Result	Remarks  Linux time format (including usec after the decimal point) multiplic
Description To stop playback of The video at the terms of the video at the video at the terms of the video at the terms of the video at	Stop  f the playl ime of rece  Required  Always  O	st at the position at the time of receiving the command.  ving the command is displayed and stopped.  Description  Command  Description  Result	Remarks  Linux time format (including usec after the decimal point) multiplic by 1000000  If you set this value to the playback position of playlist playback.
Description To stop playback of The video at the term video at the term of the video at the	Stop  f the playl ime of rece  Required  Always  O	st at the position at the time of receiving the command.  ving the command is displayed and stopped.  Description  Command  Description  Result	Remarks  Linux time format (including usec after the decimal point) multiplic by 1000000  If you set this value to the playback position of playlist playback,
Description To stop playback of The video at the tendence of the video at the video at the tendence of the video at the video at the video at the video at the tendence of the video at the video a	Stop  f the playl ime of rece  Required  Always  O	st at the position at the time of receiving the command.  ving the command is displayed and stopped.  Description  Command  Description  Result	Remarks  Linux time format (including usec after the decimal point) multiplic by 1000000  If you set this value to the playback position of playlist playback.
Description To stop playback of The video at the text the video at the text the text the video at the text the text the video at the video at the text the video at the text the video at the	Stop  f the playl ime of rece  Required  Always  O  100000000  between the	Description Command  Description Playlist stop position (elapsed time from the beginning)  execution result and the parameter is "" (space), and t	Remarks  Linux time format (including usec after the decimal point) multiplic by 1000000  If you set this value to the playback position of playlist playback the playlist will be played from the stopped position.
Description To stop playback of The video at the text the video at the text the text the video at the text the text the video at the video at the text the video at the text the video at the	Stop  f the playl ime of rece  Required  Always  O  100000000  between the	Description Command  Description Playlist stop position (elapsed time from the beginning)  execution result and the parameter is "" (space), and t	Remarks  Linux time format (including usec after the decimal point) multiplic by 1000000  If you set this value to the playback position of playlist playback, the playlist will be played from the stopped position.
Description To stop playback of The video at the term video at the	Stop  f the playl ime of rece  Required  Always  O  100000000  between the	Description Command  Description Playlist stop position (elapsed time from the beginning)  execution result and the parameter is "" (space), and t	Remarks  Linux time format (including usec after the decimal point) multiplic by 1000000  If you set this value to the playback position of playlist playback the playlist will be played from the stopped position.
Description To stop playback of The video at the terms of the video at	Stop  f the playlime of rece  Required  Always  O  100000000  between the seconds = 10	Description Command  Description Playlist stop position (elapsed time from the beginning)  execution result and the parameter is "" (space), and t	Remarks  Linux time format (including usec after the decimal point) multiplic by 1000000  If you set this value to the playback position of playlist playback, the playlist will be played from the stopped position.
Description To stop playback of The video at the terms of the video at the video at the video at the video at the terms of the video at	Stop  f the playl ime of rece  Required  O  Always  O  =100000000  between the seconds = 10  =-1  o the follow	Description Command  Description  Playlist stop position (elapsed time from the beginning)  execution result and the parameter is "" (space), and to seconds	Remarks  Linux time format (including usec after the decimal point) multiplic by 1000000  If you set this value to the playback position of playlist playback the playlist will be played from the stopped position.
Description To stop playback of The video at the terms of the video at	Stop  f the playl ime of rece  Required  Required  O  Always  O  100000000  between the seconds = 100000000000000000000000000000000000	Description Command  Description Result Playlist stop position (elapsed time from the beginning)  execution result and the parameter is "" (space), and to seconds	Remarks  Linux time format (including usec after the decimal point) multiplic by 1000000  If you set this value to the playback position of playlist playback, the playlist will be played from the stopped position.

Seeking playlist playback
seeking playing playeack

#### Command Seek

## Description

To play a playlist from the playlist playback position specified during playlist playback.

If playlist playback is stopped, it will only move to the playback position of the specified playlist.

Request			
Parameter	Required	Description	Remarks
Seek	0	Command	
		Regeneration start time (LinuxTime + usec)	Linux time format (including usec after the decimal point) multiplied by 1000000
SyncTime			Specify a time that is at least 1 seconds ahead of the current time
			Also, include the time difference in GetSystemTime
PlayPosition	1 ()	Playlist playback position (elapsed time from the beginning)	Play from the beginning with O

## Example Request

When specifying synchronized playback start time: Seek SyncTime=1727669330084099&PlayPosition=0

When not specifying synchronized playback start time: Seek PlayPosition=0

- \*1: The separator between UDP commands and parameters is "" (space), the separator between parameters is "&", and the parameter value is after "=".
- \*2: If the playback position of the playlist is set to something other than 0, it can be played or moved from the middle of the playlist.

\*3: If synchronized playback start time is not specified, the "SyncTime" parameter can be omitted. In that case, the FMP will start playback 3 seconds after receiving this command (changeable with SetSyncTime).

Return Value			
Parameter	Always	Description	Remarks
OK/NG	0	Result	
NG Parameter		Cause of NG	

## Example Return On Success

## On Failure

- \*: NG may be due to the following factors.
- Playlist not specified -> Playlist not selected
- Playlist playback position is outside the playlist range -> Out of range PlayPosition: (Specified playback position)
- Another command is in progress -> Other command running

Skipping media in playlists	
-----------------------------	--

#### Command Skip

# Description

to skip from the current media to the next in a playlist.

If the playlist playback is stopped, skipping will not be performed.

Also, if there is only one piece of media registered in the playlist, it will not be skipped and will stop playing.

(When the loop setting is turned on, it returns to the beginning.)

Request			
Parameter	Required	Description	Remarks
Skip	0	Command	

# Example Request

\*1: The separator between UDP commands and parameters is "" (space), and the value of the parameter is after "=".
\*2:If synchronized playback start time is not specified, the "SyncTime" parameter can be omitted. In that case, the FMP will start playback 3 seconds after receiving this command (changeable with SetSyncTime).

Return Value			
Parameter	Always	Description	Remarks
OK/NG	0	Result	
NG Parameter		Cause of NG	

## Example Return On Success

# On Failure

- \*: NG may be due to the following factors.
- Playlist not specified -> Playlist not selected
- Playlist is not playing -> Playlist not playing
- Another command is in progress -> Other command running

Commercial	lo-1Pi	ul i a t	
Command	SelectPla	yL18t	
<b>Description</b> To select a play	list to play	in a playlist registered in FMP.	
Request Parameter	Required	Description	Remarks
SelectPlayList PlayListName	0	Command Playlist name to select	Nomal Ro
1 Tay E To EName		Trayriot name to coroct	
<b>Example Request</b> SelectPlayList P	lavlistName=P	lavlist01	
		commands and parameters is "" (space), and the va	alue of the parameter is after "="
The doparator	BOCHOON ODI	Copaddy, and the ve	True of the parameter to arter
Return Value			
Return value Parameter OK/NG	Always	Description	Remarks
ON/ NU	0	Result	
Evenn la Deter		l	l
Example Return On Success			
OK			
On Failure NG			
NG	to the follow	wing factors	
	to the follow does not exist	wing factors. st in FMP.	
NG	to the followedoes not exis	wing factors. st in FMP.	
NG	to the follow does not exist	wing factors. st in FMP.	
NG			
NG		wing factors. st in FMP. op settings	
NG			
NG  *: NG may be due  - The playlist  Command  Description	Playlist loo	op settings	
NG  *: NG may be due  - The playlist  Command  Description  To choose whethe	Playlist loo Loop r to loop the	playlist playback within the playlist.	
NG  *: NG may be due  - The playlist  Command  Description	Playlist loo Loop r to loop the	playlist playback within the playlist.	
NG  *: NG may be due  - The playlist  Command  Description To choose whethe The initial value	Playlist loo Loop r to loop the	playlist playback within the playlist.	
NG  *: NG may be due  - The playlist  Command  Description To choose whethe The initial value  Request Parameter	Playlist loo  Loop  r to loop the e is no loop  Required	playlist playback within the playlist. (stop at end).	Remarks
NG  *: NG may be due  - The playlist  Command  Description To choose whethe The initial value  Request	Playlist loo  Loop  r to loop the e is no loop  Required	playlist playback within the playlist.  (stop at end).	Remarks No loop(stop at the end), loop(play again from the beginning)
NG  *: NG may be due  - The playlist  Command  Description  To choose whethe  The initial value  Request  Parameter  Loop	Playlist loo  Loop  r to loop the e is no loop  Required	playlist playback within the playlist. (stop at end).  Description Command	
NG  *: NG may be due  - The playlist  Command  Description  To choose whethe  The initial value  Request  Parameter  Loop	Playlist loo  Loop  r to loop the e is no loop  Required	playlist playback within the playlist. (stop at end).  Description Command	
*: NG may be due  - The playlist  Command  Description To choose whethe The initial value  Request Parameter Loop EnableLoop  Example Request	Playlist loo  Loop  r to loop the e is no loop  Required	playlist playback within the playlist. (stop at end).  Description Command	
*: NG may be due  - The playlist  Command  Description To choose whethe The initial value  Request Parameter Loop EnableLoop  Example Request Loop EnableLoop=	Playlist loo  Loop  r to loop the e is no loop  Required  O	playlist playback within the playlist.  (stop at end).  Description Command Loop setting (0: no loop, 1: with loop)	No loop(stop at the end), loop(play again from the beginning)
*: NG may be due  - The playlist  Command  Description To choose whethe The initial value  Request Parameter Loop EnableLoop  Example Request Loop EnableLoop=	Playlist loo  Loop  r to loop the e is no loop  Required  O	playlist playback within the playlist. (stop at end).  Description Command	No loop(stop at the end), loop(play again from the beginning)
*: NG may be due  - The playlist  Command  Description To choose whethe The initial value  Request Parameter Loop EnableLoop  Example Request Loop EnableLoop=	Playlist loo  Loop  r to loop the e is no loop  Required  O	playlist playback within the playlist.  (stop at end).  Description Command Loop setting (0: no loop, 1: with loop)	No loop(stop at the end), loop(play again from the beginning)
*: NG may be due  - The playlist  Command  Description To choose whethe The initial value  Request Parameter Loop EnableLoop  Example Request Loop EnableLoop=  *: The separator	Playlist loo  Loop  r to loop the e is no loop  Required  O	playlist playback within the playlist.  (stop at end).  Description Command Loop setting (0: no loop, 1: with loop)	No loop(stop at the end), loop(play again from the beginning)
*: NG may be due  - The playlist  Command  Description To choose whethe The initial value  Request Parameter Loop EnableLoop  *: The separator  Return Value Parameter	Playlist loo  Loop  r to loop the e is no loop  Required  O  O  between UDP o	playlist playback within the playlist.  (stop at end).  Description  Command  Loop setting (0: no loop, 1: with loop)  commands and parameters is "" (space), and the value of the playlist.	No loop(stop at the end), loop(play again from the beginning)  alue of the parameter is after "=".  Remarks
*: NG may be due  - The playlist  Command  Description To choose whethe The initial value  Request Parameter Loop EnableLoop  Example Request Loop EnableLoop=  *: The separator  Return Value	Playlist loo  Loop  r to loop the e is no loop  Required  O  O  between UDP o	playlist playback within the playlist.  (stop at end).  Description  Command  Loop setting (0: no loop, 1: with loop)  commands and parameters is "" (space), and the variable of the setting of the sett	No loop(stop at the end), loop(play again from the beginning)  alue of the parameter is after "=".
*: NG may be due  - The playlist  Command  Description To choose whethe The initial value  Request Parameter Loop EnableLoop  *: The separator  Return Value Parameter	Playlist loo  Loop  r to loop the e is no loop  Required  O  O  between UDP o	playlist playback within the playlist.  (stop at end).  Description  Command  Loop setting (0: no loop, 1: with loop)  commands and parameters is "" (space), and the value of the playlist.	No loop(stop at the end), loop(play again from the beginning)  alue of the parameter is after "=".  Remarks
*: NG may be due  - The playlist  Command  Description To choose whethe The initial value  Request Parameter Loop EnableLoop  *: The separator  Return Value Parameter OK	Playlist loo  Loop  r to loop the e is no loop  Required  O  O  between UDP o	playlist playback within the playlist.  (stop at end).  Description  Command  Loop setting (0: no loop, 1: with loop)  commands and parameters is "" (space), and the value of the playlist.	No loop(stop at the end), loop(play again from the beginning)  alue of the parameter is after "=".  Remarks
*: NG may be due  - The playlist  Command  Description To choose whethe The initial value  Request Parameter Loop EnableLoop  *: The separator  Return Value Parameter OK  Example Return	Playlist loo  Loop  r to loop the e is no loop  Required  O  O  between UDP o	playlist playback within the playlist.  (stop at end).  Description  Command  Loop setting (0: no loop, 1: with loop)  commands and parameters is "" (space), and the value of the playlist.	No loop(stop at the end), loop(play again from the beginning)  alue of the parameter is after "=".  Remarks
*: NG may be due  - The playlist  Command  Description To choose whethe The initial value  Request Parameter Loop EnableLoop  *: The separator  Return Value Parameter OK	Playlist loo  Loop  r to loop the e is no loop  Required  O  O  between UDP o	playlist playback within the playlist.  (stop at end).  Description  Command  Loop setting (0: no loop, 1: with loop)  commands and parameters is "" (space), and the value of the playlist.	No loop(stop at the end), loop(play again from the beginning)  alue of the parameter is after "=".  Remarks
*: NG may be due  - The playlist  Command  Description To choose whethe The initial value  Request Parameter Loop EnableLoop  *: The separator  Return Value Parameter OK  Example Return ON Seccess	Playlist loo  Loop  r to loop the e is no loop  Required  O  O  between UDP o	playlist playback within the playlist.  (stop at end).  Description  Command  Loop setting (0: no loop, 1: with loop)  commands and parameters is "" (space), and the value of the playlist.	No loop(stop at the end), loop(play again from the beginning)  alue of the parameter is after "=".  Remarks
NG  *: NG may be due  - The playlist  Command  Description To choose whethe The initial value  Request Parameter Loop EnableLoop  *: The separator  Return Value Parameter OK  Example Return ON  Con Failure	Playlist loo  Loop  r to loop the e is no loop  Required  O  O  between UDP o	playlist playback within the playlist.  (stop at end).  Description  Command  Loop setting (0: no loop, 1: with loop)  commands and parameters is "" (space), and the value of the playlist.	No loop(stop at the end), loop(play again from the beginning)  alue of the parameter is after "=".  Remarks
NG  *: NG may be due  - The playlist  Command  Description To choose whethe The initial value  Request Parameter Loop EnableLoop  *: The separator  Return Value Parameter OK  Example Return On Seccess OK	Playlist loo  Loop  r to loop the e is no loop  Required  O  O  between UDP o	playlist playback within the playlist.  (stop at end).  Description  Command  Loop setting (0: no loop, 1: with loop)  commands and parameters is "" (space), and the value of the playlist.	No loop(stop at the end), loop(play again from the beginning)  alue of the parameter is after "=".  Remarks

	DisconnectReq			
Command	וען sconnect	LKEQ		
<b>Description</b> To restart the Ti	meline module	es running in the FMP.		
* Please restart	the Timeline	module when you replace the playlist.		
Request				
Parameter DisconnectReq		Description Command	Remarks	
Example Request			•	
DisconnectReq	•			
Return Value		ID	ID	
<b>Parameter</b> OK	Always O	Description Result	Remarks Always [OK] is returnd	
Example Return				
On Seccess OK				
On Failure				
On Failure none				
	Reload a p	playlist		
none				
Command	ResetPlay	list		
Command	ResetPlay	list		
Command  Description To update the pla	ResetPlay	list ation imported into the module running in the FMH		
Command  Description To update the pla	ResetPlay	list ation imported into the module running in the FMH	ayback is stopped. After that, you need to select a playlist.	
Command  Description To update the pla	ResetPlay	list ation imported into the module running in the FMH		
Command  Description To update the pla  * Please use this Further, when t	ResetPlay ylist informa command wher his Command i	list ation imported into the module running in the FMI n "Replace Playlist". is executed during playlist playback, playlist p	ayback is stopped. After that, you need to select a playlist.	
Command  Description To update the pla  * Please use this Further, when the sequest Parameter	ResetPlay  ylist informate command where this Command in the comma	list ation imported into the module running in the FMI n "Replace Playlist". is executed during playlist playback, playlist p		
Command  Description To update the pla  * Please use this Further, when the sequest Parameter	ResetPlay  ylist informate command where this Command in the comma	list ation imported into the module running in the FMI n "Replace Playlist". is executed during playlist playback, playlist p	ayback is stopped. After that, you need to select a playlist.	
Command  Description To update the pla  * Please use this Further, when the sequest Parameter	ResetPlay  ylist informate command where this Command in the comma	list ation imported into the module running in the FMI n "Replace Playlist". is executed during playlist playback, playlist p	ayback is stopped. After that, you need to select a playlist.	
Command  Description To update the pla  * Please use this Further, when the service of the servi	ResetPlay  ylist informate command where this Command in the comma	list ation imported into the module running in the FMI n "Replace Playlist". is executed during playlist playback, playlist p	ayback is stopped. After that, you need to select a playlist.	
Command  Description To update the pla  * Please use this Further, when t  Request Parameter ResetPlaylist  Example Request	ResetPlay  ylist informate command where this Command in the comma	list ation imported into the module running in the FMI n "Replace Playlist". is executed during playlist playback, playlist p	ayback is stopped. After that, you need to select a playlist.	
Command  Description To update the pla	ResetPlay  ylist informate command where this Command in the comma	list ation imported into the module running in the FMI n "Replace Playlist". is executed during playlist playback, playlist p	ayback is stopped. After that, you need to select a playlist.	
Command  Description To update the pla  * Please use this Further, when t  Request Parameter ResetPlaylist  Example Request	ResetPlay  ylist informate command where this Command in the comma	list ation imported into the module running in the FMI n "Replace Playlist". is executed during playlist playback, playlist p	ayback is stopped. After that, you need to select a playlist.	
Command  Description To update the plate of	ResetPlay  ylist informate command where this Command in the comma	list ation imported into the module running in the FMI n "Replace Playlist". is executed during playlist playback, playlist p	ayback is stopped. After that, you need to select a playlist.	
Command  Description To update the pla  * Please use this Further, when the set of the plant of	ResetPlay  ylist information command where this Command in the com	Iist  ation imported into the module running in the FMin "Replace Playlist". is executed during playlist playback, playlist p  Description  Command  Description	Remarks   Remarks	
Command  Description To update the pla  * Please use this Further, when the set Playlist  Example Request ResetPlaylist  ResetPlaylist  Return Value	ResetPlay  ylist information command where this Command in the com	list ation imported into the module running in the FMI n "Replace Playlist". is executed during playlist playback, playlist p	Remarks	
Command  Description To update the pla  * Please use this Further, when the set Playlist  Example Request ResetPlaylist  ResetPlaylist  Return Value Parameter	ResetPlay  ylist information command where this Command in the com	Iist  ation imported into the module running in the FMin "Replace Playlist". is executed during playlist playback, playlist p  Description  Command  Description	Remarks   Remarks	
Command  Description To update the pla  * Please use this Further, when the set of the plant of	ResetPlay  ylist information command where this Command in the com	Iist  ation imported into the module running in the FMin "Replace Playlist". is executed during playlist playback, playlist p  Description  Command  Description	Remarks   Remarks	
Command  Description To update the pla  * Please use this Further, when the standard	ResetPlay  ylist information command where this Command in the com	Iist  ation imported into the module running in the FMin "Replace Playlist". is executed during playlist playback, playlist p  Description  Command  Description	Remarks   Remarks	
Command  Description To update the pla  * Please use this Further, when the set Playlist  Example Request ResetPlaylist  Return Value Parameter OK  Example Return On Seccess	ResetPlay  ylist information command where this Command in the com	Iist  ation imported into the module running in the FMin "Replace Playlist". is executed during playlist playback, playlist p  Description  Command  Description	Remarks   Remarks	
Command  Description To update the pla * Please use this Further, when t  Request Parameter ResetPlaylist  Example Request ResetPlaylist  Return Value Parameter OK  Example Return	ResetPlay  ylist information command where this Command in the com	Iist  ation imported into the module running in the FMin "Replace Playlist". is executed during playlist playback, playlist p  Description  Command  Description	Remarks   Remarks	
Command  Description To update the pla  * Please use this Further, when the set Playlist  Example Request ResetPlaylist  Return Value Parameter OK  Example Return On Seccess	ResetPlay  ylist information command where this Command in the com	Iist  ation imported into the module running in the FMin "Replace Playlist". is executed during playlist playback, playlist p  Description  Command  Description	Remarks   Remarks	

	FMP Time Adjustment Setting

#### Command SetSyncTime

Description

Command to change the time adjustment value for command execution within the FMP when synchronized playback start time is not specified. \*The initial value is set to 3 seconds.

Also, this adjustment value will revert to 3 seconds when the FMP is restarted or the Timeline module is restarted.

Request			
Parameter	Required	Description	Remarks
SetSyncTime	0	Command	
Time	0		Specify an integer for the number of seconds. If less than 3 seconds, synchronized playback may not be possible.

## Example Request

SetSyncTime Time=5

\*: The separator between UDP commands and parameters is "" (space), and the value of the parameter is after "=".

Return Value			
Parameter	Always	Description	Remarks
OK	0	Result	Always [OK] is returnd

## Example Return On Seccess

## On Failure

none

Retrieving Playlist Information
---------------------------------

#### Command GetPlaylistInfo

Description

To get a list of playlist names registered in FMP.

Request			
Parameter	Required	Description	Remarks
GetPlaylistInfo	0	Command	
-d		Debugging Option	Notify with the ID part used in Visual Software Suite (VSS)

# Example Request

GetPlaylistInfo

GetPlaylistInfo -d

\*: The separator between Command and parameter is "" (space).

Return Value			
Parameter	Always	Description	Remarks
OK/NG	0	Result	
Playlist1		Playlist name	Name is not available if NG or unregistered
Playlist2		Playlist name	Name is not available if NG or unregistered
Playlist3		Playlist name	Name is not available if NG or unregistered
Playlist4		Playlist name	Name is not available if NG or unregistered

# Example Return

On Seccess

OK Playlist1=Weekday&Playlist2=Weekend

- \*1: The separator between the execution result and the parameter is "" (space), the delimiter between the parameters is "&", and the value of the parameter is after "=".
- \*2: In the above example, only two playlists are registered, so there is only "Playlist2".
- OK Playlist1=Weekday--(761e4d70-d6c7-4abd-aed0-889288240ef1).list&Playlist2=Weekend--(ea51550d-627f-48da-9776-f5808c0aac23).list
- \*3: When the debug option is specified, the ID part is added as described above.

# On Failure

- \*: NG may be due to the following factors.
- None of the playlists are registered.

Retrieving media information in a playlist

#### Command GetEachPlaylistInfo

Description

To get media information (media name, playback time) in a playlist registered in FMP.

Request			
Parameter	Required	Description	Remarks
GetEachPlaylistInfo	0	Command	
Playlist		The name of the playlist for which you want to retrieve media information	You don't need the ID part used by VisualSoftwareSuite (VSS)

Example Request

GetEachPlaylistInfo Playlist=Weekday

\*: The separator between Command and parameter is "" (space), and the value of the parameter is after "=".

Return Value				
Parameter	Always	Description	Remarks	
OK/NG	0	Result		
Playlist	0	Playlist name set at the time of Request		
Medianum	0	Number of media in the playlist		
Media1		First Media Name		
Duration1		Duration of the first media	The unit is "seconds"	
Media2		Second media name		
Duration2		Duration of the second media	The unit is "seconds"	
MediaN		"N"th Media Name		
DurationN		Duration of the "N"th media	The unit is "seconds"	

## Example Return

On Seccess

OK Playlist=PLAYLIST1&Medianum=3&Media1=MEDIA1.mp4&Duration1=60&Media2=MEDIA2.mp4&Duration2=60&Media3=MEDIA3.jpg&Duration3=10

- \*1: The separator between the execution result and the parameter is "" (space), the delimiter between the parameters is "&", and the value of the parameter is after "=".
- \*2: In the above example, only three media are registered in the playlist, so there are only "Media3" and "Duration3".
  \*3: If a large number of media is registered in the playlist, the maximum transmission and reception size: 1024 bytes may be exceeded, but in that case, up to 1024 bytes will be notified.

On Failure

NG Playlist=PLAYLIST1&Medianum=0

- \*: NG may be due to the following factors.
- The playlist does not exist.

Retrieving playlist playback status

#### GetPlayingInfo Command

Description

To get the status of playlist playback in FMP (determined by the current position of the playlist).

Request			
Parameter	Required	Description	Remarks
GetPlayingInfo	0	Command	

# Example Request

GetPlayingInfo

Return Value			
Parameter	Always	Description	Remarks
OK/NG	0	Result	
Playlist	0	Playlist name	
CurrentPosition	0	Playlist current location	When playback is stopped, "STOP" and other NGs are blank
Loop	0	Loop setting (0: no loop, 1: with loop)	No loop(stop at the end), loop(play again from the beginning)

# Example Return

On Seccess

OK Playlist=Weekday&CurrentPosition=100000000&Loop=0

- \*1: The separator between the execution result and the parameter is "" (space), the delimiter between the parameters is "&", and the value of the parameter is after "=".
- OK Playlist=PLAYLIST1&CurrentPosition=STOP&Loop=0
- \*2: When playlist playback is stopped (after Pause or Stop is executed), the current position of the playlist is notified by "STOP".

NG Playlist=&CurrentPosition=&Loop=0

- \*: NG may be due to the following factors.
- Playlist is not specified.

## ●Sample code (python)

Seek1 = 'Seek SyncTime='

If not specifying a synchronized playback start time, change the red-colored part of the code on the right from the code on the left. (Strikeout parts should be deleted)

```
*When specifying a synchronized playback start time
                                                                                    *When not specifying a synchronized playback start time (red text)
import socket
import time
M_SIZE = 1024
# Execution conditions
# · Python can be executed
# · LAN connection to FMP
# Pre-Execution Preparation
# Set the following items according to the FMP to be connected
# The address of the FMP and the port number of the connection destination (the port number is fixed)
# Use the broadcast address (e.g., 192.168.0.255) when connecting to multiple FMPs
fmp\_address = ('192.168.0.11', 65432)
# Playlist name to be selected in the playlist set for the FMP
PLAYLIST = 'Playlist01'
# How to execute
# (1) At the DOS prompt, move to the folder where this file (test_client.py) is located
# (2) Execute "py test_client.py"
# (3) Enter the number of the Command you want to execute
# (4) If you want to change the parameters of this file (test_client.py),
# please end with "0" once, edit this file, and continue from (2)
# GetSystemTime
# CurrentTimeVisual Software Suite (VSS) is set automatically
GetSystemTime = 'GetSystemTime CurrentTimeVss='
# SelectPlayList
# Change the playlist name to the actual name
# You can also check it with GetPlayingInfo
SelectPlayList = 'SelectPlayList PlayListName='
# Loop
# 0: No LOOP / 1: With LOOP
ENABLELOOP = '0'
Loop = 'Loop EnableLoop='
# Play
# SyncTime is automatically set by adding the diff_time (difference between FMP time) and marge_time (FMP processing time).
# When resuming playback after StopCommand, set the stop position in the response frame of StopCommand PLAY_POSITION
Play1 = 'Play SyncTime='
                                                                                    Play1 = 'Play'
Play2 = '&PlayPosition=0'
                                                                                    Play2 = '&PlayPosition=0'
Play3 = '&PlayPosition='
                                                                                    Play3 = 'PlayPosition='
                                                                                    \frac{diff_time}{=} 0
diff_time = 0
marge\_time = 2000000
                                                                                    marge\_time = 1000000
PLAY_POSITION = 0
# Pause
# StopPosition sets to 0 (the beginning of the playlist)
# Since the position during playback on the FMP is unknown, it is hardcoded.
Pause = 'Pause StopPosition=15000000' # Always stop at setting position
Pause2 = 'Pause StopPosition='
                                   # for Modify stop position
pause_position = 0
# Seek
# PlayPosition sets to 0 (the beginning of the playlist)
# SyncTime is automatically set by adding the diff_time (difference between FMP time) and marge_time (FMP processing time).
```

Seek1 = 'Seek'

```
Seek2 = '&PlayPosition='
                                                                                      Seek2 = 'PlayPosition='
seek_position = 0
# Skip
# SyncTime is automatically set by adding the diff_time (difference between FMP time) and marge_time (FMP processing time).
Skip = 'Skip SyncTime='
                                                                                      Skip = 'Skip'
# DisconnectReq
# Disconnect from FMP. When the FMP side disconnects the connection, it restarts the timeline
# Reload the playlist by restarting the timeline
# (Since the information cannot be imported into Timeline just by sending the playlist, restart Timeline to import the playlist)
DisconnectReq = 'DisconnectReq'
# Stop
# This Command will stop at the position at which the Command is received
# Also, notify the stop position in the response frame (used when resuming playback)
Stop = 'Stop'
# Get Playback information
GetPlayingInfo = 'GetPlayingInfo'
GetPlaylistInfo = 'GetPlaylistInfo'
GetEachPlaylistInfo = 'GetEachPlaylistInfo Playlist='
ResetPlaylist = 'ResetPlaylist'
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
while True:
  try:
     print(")
     print('Input any Number')
     print('')
     print('1:GetSystemTime')
                                                                                      print('1:GetSystemTime')
     print('2:SelectPlayList')
     print('3:Loop')
     print('4:Play')
     print('5:Pause')
     print('6:Stop')
     print('7:Seek')
     print('8:Skip')
    print('9:DisconnectReq - timeline restart -')
     print('0:Exit')
     print('')
     print('a:GetPlaylistInfo')
     print('b:GetPlayingInfo')
     print('c:GetEachPlaylistInfo')
     print('d:ResetPlaylist')
     print(")
                                                                                         print('e:SetSyncTime')
     message = input()
     if message == 'end':
       print('closing socket')
       sock.close()
       print('done')
       break
     else:
       if message == '1':
                                                                                      if message == '1':
          start_u = int(time.time() * 1000000)
                                                                                          start_u = int(time.time() * 1000000)
         print("send : " + GetSystemTime + str(start_u))
                                                                                           print("send : " + GetSystemTime + str(start_u))
          send_data = GetSystemTime + str(start_u)
                                                                                           send_data = GetSystemTime + str(start_u)
          send_len = sock.sendto(send_data.encode('utf-8'), fmp_address)
                                                                                           <u>send_len = sock.sendto(send_data.encode('utf-8'), fmp_address)</u>
       elif message == '2':
         print("send : " + SelectPlayList + PLAYLIST)
          send_data = SelectPlayList + PLAYLIST
          send_len = sock.sendto(send_data.encode('utf-8'), fmp_address)
       elif message == '3':
         print('')
         print('Input any Number')
          print(")
          print('1:Loop disable')
          print('2:Loop enable')
          print(")
          selectnum = input()
          if selectnum == '1':
```

```
ENABLELOOP = '0'
  elif selectnum == '2':
    ENABLELOOP = '1'
  else:
    continue
  print("send : " + Loop + ENABLELOOP)
  send_data = Loop + ENABLELOOP
  send_len = sock.sendto(send_data.encode('utf-8'), fmp_address)
elif message == '4':
  print(")
  print('Input any Number')
  print(")
  print('1:Play from start position')
  print('2:Play from stop position')
  print(")
  selectnum = input()
  if selectnum == '1':
    PLAY POSITION = 0
  elif selectnum == '2':
    PLAY_POSITION = PLAY_POSITION
  else:
    continue
  start_u = int((time.time() * 1000000) + marge_time - diff_time)
                                                                                     -start_u = int((time.time() * 1000000) + marge_time - diff_time)
  print("send : " + Play1 + str(start_u) + Play3 + str(PLAY_POSITION))
                                                                                     print("send : " + Play1 + Play3 + str(PLAY_POSITION))
  send_data = Play1 + str(start_u) + Play3 + str(PLAY_POSITION)
                                                                                     send_data = Play1 + Play3 + str(PLAY_POSITION)
  send_len = sock.sendto(send_data.encode('utf-8'), fmp_address)
  play_start_time = start_u + diff_time
                                                                                     play_start_time = start_u + diff_time
# PauseCommand needs to specify the Pause position
# (in the sample below, the elapsed time from the Play start time is forcibly set to the Pause position)
elif message == '5':
  pause_position = int(((time.time() * 1000000) - play_start_time) + PLAY_POSITION)
                                                                                                  pause_position = 10000000
  print("send : " + Pause2 + str(pause_position))
  send_data = Pause2 + str(pause_position)
  send_len = sock.sendto(send_data.encode('utf-8'), fmp_address)
  PLAY_POSITION = pause_position
elif message == '6':
  print("send : " + Stop)
  send_data = Stop
  send_len = sock.sendto(send_data.encode('utf-8'), fmp_address)
elif message == '7':
  print(")
  print('Input Seek position(0~[us])')
  print(")
  selectnum = input()
  seek_position = int(selectnum)
  start_u = int(time.time() * 1000000 + marge_time - diff_time)
                                                                                       -start_u = int(time.time() * 1000000 + marge_time - diff_time)
                                                                                       print("send : " + Seek1 + Seek2 + str(seek_position))
  print("send : " + Seek1 + str(start_u) + Seek2 + str(seek_position))
  send_data = Seek1 + str(start_u) + Seek2 + str(seek_position)
                                                                                       send_data = Seek1 + Seek2 + str(seek_position)
  send_len = sock.sendto(send_data.encode('utf-8'), fmp_address)
elif message == '8':
  start_u = int(time.time() * 1000000 + marge_time - diff_time)
                                                                                       <del>-start_u = int(time.time() * 1000000 + marge_time - diff_time)</del>
  print("send : " + Skip + str(start_u))
                                                                                       print("send : " + Skip)
  send_data = Skip + str(start_u)
                                                                                       send_data = Skip
  send_len = sock.sendto(send_data.encode('utf-8'), fmp_address)
  PLAY_POSITION = 0
elif message == '9':
  print("send : " + DisconnectReq)
  send_data = DisconnectReq
  send_len = sock.sendto(send_data.encode('utf-8'), fmp_address)
elif message == '0':
  print('closing socket')
  sock.close()
  print('done')
  break
```

```
elif message == 'a':
      print("send : " + GetPlaylistInfo)
      send_data = GetPlaylistInfo
      send_len = sock.sendto(send_data.encode('utf-8'), fmp_address)
    elif message == 'b':
      print("send : " + GetPlayingInfo)
      send_data = GetPlayingInfo
      send_len = sock.sendto(send_data.encode('utf-8'), fmp_address)
    elif message == 'c':
      print(")
      print('Input Playlist name')
      print(")
      selectpl = input()
      print("send : " + GetEachPlaylistInfo + str(selectpl))
      send_data = GetEachPlaylistInfo + str(selectpl)
      send_len = sock.sendto(send_data.encode('utf-8'), fmp_address)
    # Even if you don't use ResetPlaylist, you can substitute DisconnectReq
    elif message == 'd':
      print("send : " + ResetPlaylist)
      send_data = ResetPlaylist
      send_len = sock.sendto(send_data.encode('utf-8'), fmp_address)
                                                                                           elif message == 'e':
                                                                                             print(")
                                                                                             print('Input Sync Time(0\sim[s])')
                                                                                             print(")
                                                                                             synctime = input()
                                                                                             print("send : SetSyncTime Time=" + str(synctime))
                                                                                             send_data = "SetSyncTime Time=" + str(synctime)
                                                                                              send_len = sock.sendto(send_data.encode('utf-8'), bload_addres)
    else:
      continue
    rx_meesage, addr = sock.recvfrom(M_SIZE)
    print(f"recive: {rx_meesage.decode(encoding='utf-8')}")
    if message == '1':
      fmp_kekka = rx_meesage.decode(encoding='utf-8')[0:2]
      if fmp_kekka == 'OK':
         fmp_time = rx_meesage.decode(encoding='utf-8')[-17:-1]
         diff_time = int(start_u) - int(fmp_time)
         print("diff_time : " + str(diff_time) + " us")
      else:
         continue
    elif message == '6':
      fmp_kekka = rx_meesage.decode(encoding='utf-8')[0:2]
      if fmp_kekka == 'OK':
         fmp_time = rx_meesage.decode(encoding='utf-8')[19:-1]
         PLAY_POSITION = int(fmp_time)
        print("PLAY_POSITION : " + str(PLAY_POSITION) + " us")
      else :
         continue
except KeyboardInterrupt:
  print('closing socket')
  sock.close()
  print('done')
  break
```