

# **PS-API**

---

## **Interface Specifications for DLL**

---

Edition 12.80 R01  
Sep. 30, 2022

i-PRO Co., Ltd.

With regard to the contents of this document.

- It is prohibited to reproduce part or all in this document.

- i-PRO Co., Ltd. reserves the right, at its discretion, to change, modify, add, or remove portions of the contents of this document at any time.

## Revision Record

Version	Revised Date	Content of Revision
0.9	Feb. 1, 2009	First Edition
1.0	Feb. 16, 2009	Baseline
1.1	Feb. 25, 2009	Install ActiveX control of Network Camera is supported.
1.2	Jul 27, 2009	5.4.1.4. Add Search condition. 5.5.1.10. Add DecodeImage method. 5.5.3.3. Add Compressed video image type to OnImage event. 5.5.3.4. Add Compressed video image type SetImageListener method. 5.6.1.4. Add preset functions to CameraOperation method. 6.10. Add operation procedure about Compressed video image.
1.2 R02	Sep. 30, 2009	Correct the following points of this document. 2.3. Change the firmware version of DG-ND200/WJ-ND200. 5.6.1.1. Change the description of tilt speed.
2.0 R01	Oct 16, 2009	1.6. Add the description of Download function to Overview of Functions. 1.7. Add the description of SearchEx method, OnSearchExCB, FTP functions to Function list. 1.8. Add SearchEx method, FTP functions to Compatible chart by models. 2.3. Add HD600/700 series and NP502/NW502Sseries to supported devices. 3.1. Update Product list. 4.4.2. Add the description of FTP function. 5.1. Add ISearchResultEx class. 5.1.2.2. Add SearchEx function. 5.1.2.2. Add GetDevTimeZone function. 5.1.2.2. Add SetH264Port function.. 5.1.2.2. Add GetH264Port function. 5.1.2.2. Add SetMultiAutoConf function. 5.1.2.2. Add GetMulticastAutoConf function. 5.1.2.2. Add SetH264Resolution function. 5.1.2.2. Add GetH264Resolution function. 5.1.2.2. Add SetStreamNumber function. 5.1.2.2. Add GetStreamNumber function. 5.1.2.2. Add SetTransFrameRate function. 5.1.2.2. Add GetTransFrameRate function. 5.1.2.2. Add FtpGet function. 5.1.2.2. Add FtpCancel function. 5.1.2.2. Add FtpServerClose function. 5.1.2.2. Add SetFtpPort function. 5.1.2.2. Add GetFtpPort function. 5.1.2.2. Add SetFtpTransMode function. 5.1.2.2. Add GetFtpTransMode function. 5.1.2.2. Add GetFtpStatus function. 5.1.2.2. Add GetFtpTransRate function. 5.1.2.2. Add GetFtpTransByte function. 5.1.2.4. Add OnSearchExCB callback. 5.1.2.4. Add OnFtpStatusCB callback. 5.1.2.6. Add ISearchResultEx class.

Version	Revised Date	Content of Revision
2.0 R01	Oct 16, 2009	<p>5.3.1.1. Add the description of HD600/700 to Open method.</p> <p>5.3.2.1. Add the description of HD600/700 to DeviceType property.</p> <p>5.3.2.8. Add the description of HD600/700 to UID property.</p> <p>5.4.1.1. Add the description of HD600/700 to GetDevStatus method.</p> <p>5.4.1.2. Add the description of HD600/700 to RecCtrl method.</p> <p>5.4.1.3. Add the description of HD600/700 to GetRecStatus method.</p> <p>5.4.1.4. Add the description of HD600/700 to Search method.</p> <p>5.4.1.4. Add condition type to Search method.</p> <p>5.4.1.5. Add SearchEx method.</p> <p>5.4.1.6. Add GetDevTimeZone method.</p> <p>5.4.3.1. Add the description of HD600/700 to OnDevStatus.</p> <p>5.4.3.3. Add the description of HD600/700 to OnRecStatus.</p> <p>5.4.4.3. Add OnSearchExCB.</p> <p>5.5.1.1. Add the description of HD600/700 to GetFrameTime method.</p> <p>5.5.1.2. Add the description of HD600/700 and H.264 to PlayLive method.</p> <p>5.5.1.3. Add the description of HD600/700 and skip to the latest record to Play method.</p> <p>5.5.1.5. Add the description of HD600/700 PlayControl method.</p> <p>5.5.1.7. Add the description of HD600/700 GetPlaySpeed method.</p> <p>5.5.1.8. Add the description of HD600/700 GetFrameRate method.</p> <p>5.5.1.10 Add the description of H.264 to DecodeImage method.</p> <p>5.5.2.2. Add H264Resolution property.</p> <p>5.5.2.3. Add the description of H.264 to MulticastAddr property.</p> <p>5.5.2.5. Add H264Resolution property.</p> <p>5.5.2.6. Add the resolution type 2048 to JPEGResolution property.</p> <p>5.5.2.7. Add the description of H.264 to StreamFormat property.</p> <p>5.5.2.7. Add the description about getting property automatically to StreamFormat.</p> <p>5.5.2.10 Add MulticastAutoConf property.</p> <p>5.5.2.11. Add StreamNumber property.</p> <p>5.5.2.12. Add TransFrameRate property.</p> <p>5.5.3.1. Add the description of HD600/700 to OnPlayStatus.</p> <p>5.5.3.3. Add the description of HD600/700 to OnImage.</p> <p>5.6.1.1. Add the description of HD600/700 to CameraControl method.</p> <p>5.6.1.4. Add the description of HD600/700 to CameraOperation method.</p> <p>5.6.1.5. Add the description of HD600/700 to GetCamOpStatus method.</p> <p>5.6.2.1. Add the description of HD600/700 to OnOpStatus.</p> <p>5.6.3.1. Add the description of HD600/700 to OnOpStatusCB.</p>

Version	Revised Date	Content of Revision
2.0 R01	Oct 16, 2009	<p>5.7.1.1. Add the description of HD600/700 to AlmOperation method.</p> <p>5.7.2.1. Add the description of HD600/700 to OnAlmStatus.</p> <p>5.8. Add the description of FTP group functions.</p> <p>cedure and Sequence of FtpGet.</p> <p>6. Add the description of HD600/700 to Operation Procedure and Sequence.</p> <p>6. Add the description of H.264 to Operation Procedure and Sequence.</p> <p>6.11. Add Operation Pro</p>
2.0 R01	Oct. 30, 2009	<p>Correct the following points of this document.</p> <p>5.6.1.1 Add the description of Pan/Tilt direction depending on camera setting condition.</p> <p>5.6.1.2 Add the description of Pan/Tilt direction depending on camera setting condition.</p> <p>5.6.1.3 Add the description of Pan/Tilt direction depending on camera setting condition.</p> <p>5.3.2.4 Add the description of HttpTimeout.</p>
2.0 R01	Nov. 9, 2009	Update Package contents list
2.0 R02	Dec. 1, 2009	Update by QC indication.
2.0 R02	Dec. 24 2009	Correct the spelling.
2.0 R03	Jan. 26, 2010	Change company name.
3.0 R01	Jun. 23, 2010	<p>1.4 Deleted Visual C++ 6.0 from Abbreviations.</p> <p>1.6 Added "Image recognition" to Overview of Functions.</p> <p>1.6 Added "Digital zoom, Overlay" to Overview of Functions.</p> <p>1.6 Added "Snap Shot" to Overview of Functions.</p> <p>1.6 Added "Audio reception and transmission" to Overview of Functions.</p> <p>1.6 Added "Auto Back Focus functions, Super Dynamic function" to camera control function of Overview of Functions.</p> <p>1.6 Added "Control AUX" to Overview of Functions.</p>

Version	Revised Date	Content of Revision
3.0 R01	Jun. 23, 2010	<p>1.7 Added ClearWaitingFunc、GetWaitingFuncCount method to the function list.</p> <p>1.7 Added VMDSearchEx method to the function list.</p> <p>1.7 Added SearchCancel method to the function list.</p> <p>1.7 Added GetDeviceLog method to the function list.</p> <p>1.7 Added GetDevCurrentInfo method and GetInfoString method to the function list.</p> <p>1.7 Added PlayControlByTime method to the function list.</p> <p>1.7 Added GetImageResolution method to the function list.</p> <p>1.7 Added ClearImage method to the function list.</p> <p>1.7 Added SnapShot method to the function list.</p> <p>1.7 Added TitleOperation method, GetTitle method and BoxOperation method to the function list.</p> <p>1.7 Added DigitalZoomMove method to the function list.</p> <p>1.7 Added SetIntelligentView method, GetIntelligentView method, SetIntelligentViewColor method, GetIntelligentViewColor method, SetIntelligentViewSize method, GetIntelligentViewSize method, SetIntelligentViewTrackTime method and GetIntelligentTrackTime method to the function list.</p> <p>1.7 Added SetRecordListener method to the function list.</p> <p>1.7 Added AudioSend method, GetAudioSendStatus method to the function list.</p> <p>1.7 Added CameraCentering method to the function list.</p> <p>1.7 Added CameraAuxControl method and GetCameraAuxStatus method to the function list.</p> <p>1.7 Added OnRecordStatus method to the function list.</p> <p>1.8 Updated the list of compatible chart by models.</p> <p>2.1 Added Windows 7 Professional to System Environment.</p> <p>2.1 Deleted Pentium4 from System Environment.</p> <p>2.2 Deleted VisualC++ 6.0 from System Environment.</p> <p>4.2 Added network playback to the function that works by using shared UID.</p> <p>4.3 Added the restrictions to use a shared UID.</p> <p>5.1.2.2 Added ClearWaitingFunc method and GetWaitingFuncCount method.</p> <p>5.1.2.2 Added VMDSearchEx method and SearchCancel method.</p> <p>5.1.2.2 Added GetDeviceLog method.</p> <p>5.1.2.2 Added GetDevCurrentInfo method and GetInfoString method.</p> <p>5.1.2.2 Added SetSearchMultiChMask method and GetSearchMultiChMask method.</p> <p>5.1.2.2 Added PlayControlByTime method.</p> <p>5.1.2.2 Added GetImageResolution method.</p> <p>5.1.2.2 Added ClearImage method.</p> <p>5.1.2.2 Added SnapShot method.</p>

Version	Revised Date	Content of Revision
3.0 R01	Jun. 23, 2010	<p>5.1.2.2 Added TitleOperation method, GetTitle method and BoxOperation method.</p> <p>5.1.2.2 Added DigitalZoomMove method.</p> <p>5.1.2.2 Added SetBackColor method and GetBackColor method.</p> <p>5.1.2.2 Added SetDigitalZoom method, GetDigitalZoom method, GetDigitalZoomPosition method, SetDigitalZoomMode method and GetDigitalZoomMode method.</p> <p>5.1.2.2 Added SetIntelligentView method, GetIntelligentView method, SetIntelligentViewColor method, GetIntelligentViewColor method, SetIntelligentViewSize method, GetIntelligentViewSize method, SetIntelligentViewTrackTime method and GetIntelligentViewTrackTime method.</p> <p>5.1.2.2 Added SetSkipRecordGap method and GetSkipRecordGap method.</p> <p>5.1.2.2 Added SetRecordListener method.</p> <p>5.1.2.2 Added SetMultiScreenChannel method and GetMultiScreenChannel method.</p> <p>5.1.2.2 Added AudioSend method, GetAudioSendStatus method, SetAudioRcvEnable method, GetAudioRcvEnable method, SetAudioRcvVolume method, GetAudioRcvVolume method, SetAudioRcvMute method, GetAudioRcvMute method, SetAudioSendVolume method, GetAudioSendVolume method, SetAudioSendMute method and GetAudioSendMute method.</p> <p>5.1.2.2 Added CameraCentering method.</p> <p>5.1.2.2 Added CameraAuxControl method.</p> <p>5.1.2.2 Added GetCameraAuxStatus method.</p> <p>5.1.2.3 Added OnRecordStatus method.</p> <p>5.1.2.7 Added IPSAPIPicture class.</p> <p>5.3.1.5 Added ClearWaitingFunc method.</p> <p>5.3.1.6 Added GetWaitingFuncCount method.</p> <p>5.4.1.6 Added VMDSearchEx method.</p> <p>5.4.1.7 Added SearchCancel method.</p> <p>5.4.1.8 Added GetDeviceLog method.</p> <p>5.4.1.10 Added GetDevCurrentInfo method.</p> <p>5.4.1.11 Added GetInfoString method.</p> <p>5.4.2.2 Added SearchMultiChMask method.</p> <p>5.5.1.5 Added Next recoed and Previous record to PlayControl method.</p> <p>5.5.1.6 Added PlayControlByTime method.</p> <p>5.5.1.10 Added GetImageResolution method.</p> <p>5.5.1.13 Added ClearImage method.</p> <p>5.5.1.14 Added SnapShot method.</p> <p>5.5.1.15 Added TitleOperation method.</p> <p>5.5.1.16 Added GetTitle method.</p> <p>5.5.1.17 Added BoxOperation method.</p> <p>5.5.1.18 Added DigitalZoomMove method.</p> <p>5.5.1.19 Added GetDigitalZoomPosition method.</p> <p>5.5.1.20 Added SetIntelligentView method.</p>

Version	Revised Date	Content of Revision
3.0 R01	Jun. 23, 2010	<p>5.5.1.21 Added GetIntelligentView method.</p> <p>5.5.1.22 Added SetIntelligentViewColor method.</p> <p>5.5.1.23 Added GetIntelligentViewColor method.</p> <p>5.5.1.24 Added SetIntelligentViewSize method.</p> <p>5.5.1.25 Added GetIntelligentViewSize method.</p> <p>5.5.1.26 Added SetIntelligentViewTrackTime method.</p> <p>5.5.1.27 Added GetIntelligentViewTrackTime method.</p> <p>5.5.2.13 Added BackColor property.</p> <p>5.5.2.14 Added DigitalZoom property.</p> <p>5.5.2.15 Added DigitalZoomMode property.</p> <p>5.5.2.16 Added SkipRecordGap property.</p> <p>5.5.2.17 Added MultiScreenChannel property.</p> <p>5.5.3.3 Added OnRecordStatus event.</p> <p>5.5.3.4 Added SetRecordListener method.</p> <p>5.6.1.1 Added AudiSend method.</p> <p>5.6.1.2 Added GetAudioSendStatus method.</p> <p>5.6.2.1 Added AudioRcvEnable property.</p> <p>5.6.2.2 Added AudioRcvVolume property.</p> <p>5.6.2.3 Added AudioRcvMute property.</p> <p>5.6.2.4 Added AudioSendVolume property.</p> <p>5.6.2.5 Added AudioSendMute property.</p> <p>5.7.1.4 Added Auto back Focus functions and Super Dynamic function to CameraOperation method.</p> <p>5.7.1.6 Added CameraCentering method.</p> <p>5.7.1.7 Added CameraAuxControl method.</p> <p>5.7.1.8 Added GetCameraAuxStatus method.</p>
3.0 R02	Jul. 30, 2010	<p>1.7 Added MultiSyncPause method and MultiSyncTime method.</p> <p>1.8 Deleted ND300 from the list of device that can specify the playback speed directly.</p> <p>4.3 Added MultiSyncPause method and MultiSyncTime method to the network playback of "Restrictions of the usage when using the shared UID" section.</p> <p>5.1.2.2 Added the argument of summer time information to PlayControlByTime method.</p> <p>5.1.2.2 Added MultiSyncPausemethod and MultiSyncTime method.</p> <p>5.1.1.1, 5.1.2.3 Added the argument of summer time information to OnRecordStatus method.</p> <p>5.5.1.5 Deleted ND300 from the list of device that can specify the playback speed directly.</p> <p>5.5.1.6 Added the argument of summer time information to PlayControlByTime method.</p> <p>5.5.1.28 Added MultiSyncPause method.</p> <p>5.5.1.29 Added MultiSyncTime method</p> <p>5.5.3.3 Added the argument of summer time information to OnRecordStatus.</p> <p>5.5.3.3 Added the description that OnRecordStatus doesn't occur with file playback.</p> <p>5.6.1.1 Added the description about audio transmission in case that send audio to the full duplex mode device or the half duplex mode device.</p>



Version	Revised Date	Content of Revision
3.0 R02	Jul. 30, 2010	6.12 Added the operation procedure and the sequence for Audio. 6.13 Added the operation procedure and the sequence for SnapShot. 6.14 Added the operation procedure and the sequence for Overlay. 6.15 Added the operation procedure and the sequence for VMDSearchEx.
3.0 R02	Aug. 4, 2010	2.3 Update Supported Panasonic Products.
3.0 R02	Aug. 17, 2010	3.1.1 Update Package contents list. 3.2 Added the VC++ runtime install procedure. 3.3 Added the VC++ runtime uninstall procedure.
4.0 R01	Dec. 14, 2010	1.7. Add the description of GetPicturePosition method to Function list. 1.8. Add GetPicturePosition to Compatible chart by models. 2.3. Add NV200 series, SC385 series and SP105/SP102 series to supported devices. Separate Install/Uninstall procedure. 3.4 Update the description of UID in Restrictions. 4.2 Added the limitation description about NV200 connection. 5.1.2.2 Added GetPicturePosition method. 5.1.2.2 Added FilePassword property. 5.1.2.2 Added PictureFitMode property. 5.4.1.2 Added the description that NV200 doesn't support Manual recording to Note of RecCtrl method. 5.5.1.4 Added the description about FilePassword property to Note of PlayFile method. 5.5.1.10 Added GetPicturePosition method. 5.5.2.5 Added 16:9 resolution to H264Resolution property. 5.5.2.6 Added 16:9 resolution to JPEGResolution property. 5.5.2.10 Added FilePassword property. 5.5.2.14 Added PictureFitMode property. 5.7.1.4 Added Preset sequence, Auto sort, Patrol functions to CameraOperation method. 6.3.1 Added FilePassword to Operation Procedure of PlayFile. 6.3.2 Added FilePassword to Sequence of PlayFile. 7 Update error code list.
4.1 R01	Aug. 23, 2011	2.1 Added Windows 7 Professional SP1 to System Environment. 2.1 Added Microsoft® Windows Server® 2003 to System Environment. 2.3. Added SW355 series, SC384 series, SW395 and SF340 series to supported devices. 5.1.2.2 Added InternetMode property. 5.5.2.13 Added InternetMode property. 7 Updated error code list.

Version	Revised Date	Content of Revision
5.0 R01	Dec. 21, 2011	<p>2.1 Added Microsoft® Windows Server® 2008 R2 to System Environment.</p> <p>2.3 Separate supported devices to another document (PS-API Supported Product List).</p> <p>5.3.2.2 Added IPv6 description.</p> <p>5.4.1.12 Added SetCameraTime method.</p> <p>5.5.1.17 Added TitleOperationEx method.</p> <p>5.5.1.20 Added BoxOperationEx method.</p> <p>5.5.1.21 Added BitmapOperationEx method.</p> <p>5.5.2.3 Added RtpPortMode property.</p> <p>5.5.2.4 Added RtpPortRange property.</p> <p>5.5.2.16 Added FastPlayMode property.</p>
5.0 R02	Feb. 16, 2012	5.7.1.4 SC386 and SW396 support Patrol function of CameraOperation.
6.0 R01	Apr. 27, 2012	<p>1.7. Add the description of GetLoginStatus method, GetUIDInfo method, GetSIDInfo method, GetStatisticsData method and SetUIDPriority method to Function list.</p> <p>1.8. Add GetLoginStatus method, GetUIDInfo method, GetSIDInfo method, GetStatisticsData method and SetUIDPriority method to Compatible chart by models.</p> <p>4.3 Added the description of StreamID.</p> <p>5.3.1.7 Added GetLoginStatus method.</p> <p>5.3.1.8 Added GetUIDInfo method.</p> <p>5.3.1.9 Added GetSIDInfo method.</p> <p>5.4.1.13 Added GetStatisticsData method.</p> <p>5.4.1.14 Added SetUIDPriority method.</p> <p>5.5.2.23 Added SIDMode property.</p> <p>7 Updated error code list.</p>
6.0 R02	Jul. 7, 2012	Add SF539 series and SF549 series to supported devices.

Version	Revised Date	Content of Revision
7.0 R01	Dec. 25, 2012	<p>1.7. Added the description of CamSnapShot method to Function list.</p> <p>1.7.Deleted the description of SetIntelligentView method, GetIntelligentView method, SetIntelligentViewColor method, GetIntelligentViewColor method, SetIntelligentViewSize method , GetIntelligentViewSize method, SetIntelligentViewTrackTime method and GetIntelligentTrackTime method from Function list.</p> <p>1.8. Added CamSnapShot method to Compatible chart by models.</p> <p>4.7 Added the description of 360-degree network camera.</p> <p>5.1.2.2. Deleted SetIntelligentView method, GetIntelligentView method, SetIntelligentViewColor method, GetIntelligentViewColor method, SetIntelligentViewSize method , GetIntelligentViewSize method, SetIntelligentViewTrackTime method and GetIntelligentTrackTime method from list.</p> <p>5.1.2.2 Added CamSnapShot method to list.</p> <p>5.5.1.24 Deleted SetIntelligentView method.</p> <p>5.5.1.25 Deleted GetIntelligentView method.</p> <p>5.5.1.26 Deleted SetIntelligentViewColor method.</p> <p>5.5.1.27 Deleted GetIntelligentViewColor method.</p> <p>5.5.1.28 Deleted SetIntelligentViewSize method.</p> <p>5.5.1.29 Deleted GetIntelligentViewSize method.</p> <p>5.5.1.30 Deleted SetIntelligentViewTrackTime method.</p> <p>5.5.1.31 Deleted GetIntelligentViewTrackTime method.</p> <p>5.5.1.34 Added CamSnapShot method.</p> <p>7 Updated error code list.</p>
7.1 R01	Mar. 21, 2013	<p>1.7 Add the description of SetCameraImageCap method to Function list.</p> <p>1.8 Add SetCameraImageCap method to Compatible chart by models.</p> <p>1.8 Added the limitation description about RecCtrl method.</p> <p>2.1 Added Microsoft® Windows® 8 Pro to System Environment.</p> <p>5.1.2.2 Added SetCameraImageCap method.</p> <p>5.5.1.9 Updated the note of GetFrameRate method.</p> <p>5.7.1.9 Added Add SetCameraImageCap method.</p> <p>7 Updated error code list.</p>

Version	Revised Date	Content of Revision
7.2 R01	Jul. 12, 2013	<p>1.8. Added new model of 360-degree Network Camera.</p> <p>4.7. Added new model of 360-degree Network Camera .</p> <p>5.5.1.2. Updated Argument and Note of PlayLive method.</p> <p>5.5.1.34. Updated Note of CamSnapShot method.</p> <p>5.5.2.14. Updated Note of StreamNumber property.</p> <p>5.5.3.1. Updated Argument of OnPlayStatus listener.</p> <p>5.7.1.1. Updated Argument and Note of CameraControl method.</p> <p>5.7.1.2. Updated Argument and Note of SetCameraPosition method</p> <p>5.7.1.3. Updated Argument and Note of GetCameraPosition method.</p> <p>5.7.1.4. Updated Argument and Note of CameraOperation method.</p> <p>5.7.1.5. Updated Argument of GetCamOpStatus method.</p> <p>5.7.1.6. Updated Argument and Note of CameraCentering method.</p> <p>5.7.2.1. Updated Argument of OnOpStatus listener.</p> <p>5.8.1.1. Updated Argument of AlmOperation method.</p> <p>5.8.2.1. Updated Argument of OnAlmStatus listener.</p> <p>6.5.1 Updated Description of Absolute position camera control (7 SetCameraPosition).</p> <p>5.5.2.6. Updated Argument for SET of MPEG4Resolution property.</p> <p>5.5.2.7. Updated Argument for SET of H264Resolution property.</p> <p>5.5.2.8. Updated Argument for SET of JPEGResolution property.</p> <p>Correction of Typographical Error.</p>
7.4 R01	Mar. 26, 2014	<p>2.1 Added Microsoft® Windows® 8.1 Pro to System Environment.</p> <p>2.1 Update System Environment.</p> <p>4.7 Updated the description of 360-degree network camera.</p> <p>5.5.2.7. Updated Argument for SET of H264Resolution property.</p> <p>5.5.2.8. Updated Argument for SET of JPEGResolution property.</p> <p>5.5.2.14. Updated Argument for SET of StreamNumber property.</p> <p>5.5.2.25. Added DecResolutionMode property</p> <p>5.6.1.1. Updated Note of AudioSend method.</p> <p>5.4.1.12. Updated Note of SetCameraTime method.</p> <p>5.7.1.4. Updated Argument for CameraOperation method.</p> <p>7 Updated error code list.</p>

Version	Revised Date	Content of Revision
7.5 R01	Jul. 14, 2014	<p>1.8 Updated Compatible chart by models.</p> <p>3.4 Updated Restrictions</p> <p>5.3.1.8 Updated Note of GetUIDInfo method.</p> <p>5.3.1.9 Updated Note of GetSIDInfo method.</p> <p>5.4.1.1 Updated Description , Argument ,Note of GetDeviceStatus method.</p> <p>5.4.1.2 Updated Note of RecCtrl method.</p> <p>5.4.1.3 Updated Argument , Note of GetRecStatus method.</p> <p>5.4.1.4 Updated Argument of Search method.</p> <p>5.4.1.5 Updated Argument of SearchEx method.</p> <p>5.4.1.6 Updated Argument of VMDSearchEx method.</p> <p>5.4.1.11 Updated Note of GetInfoString method.</p> <p>5.4.1.12 Updated Note of SetCameraTime method.</p> <p>5.4.1.13 Updated Description of GetStatisticsData method.</p> <p>5.4.1.14 Updated Description of SetUIDPriority method.</p> <p>5.4.3.1 Updated Description , Argument of OnDevStatus event.</p> <p>5.4.3.3 Updated Description , Argument of OnRecStatus event.</p> <p>5.5.1.1 Updated Note of GetFrameTime method.</p> <p>5.5.1.2 Updated Argument of PlayLive method.</p> <p>5.5.1.3 Updated Argument of Play method.</p> <p>5.5.1.5 Updated Note of PlayControl method.</p> <p>5.5.1.8 Updated Note of GetPlaySpeed method.</p> <p>5.5.2.15 Updated Note of InternetMode property.</p> <p>5.5.2.16 Updated Note of FastPlayMode property.</p> <p>5.5.3.1 Updated Note of OnPlayStatus event.</p> <p>5.7.1.1 Updated Argument of CameraControl method.</p> <p>5.7.1.4 Updated Argument of CameraOperation method.</p> <p>5.7.1.5 Updated Argument of GetCameraOpStatus method.</p> <p>5.7.1.6 Updated Argument of CameraCentering method.</p> <p>5.7.2.1 Updated Argument of OnOpStatus event.</p> <p>5.8.1.1 Updated Argument of AlmOperation event.</p> <p>5.8.2.1 Updated Argument of OnAlmStatus event.</p> <p>5.9.1.1 Updated Argument of FTPGet method.</p> <p>5.9.1.3 Updated Note of FTPServerClose method.</p> <p>7 Updated error code list.</p>

Version	Revised Date	Content of Revision
8.1 R01	Jan. 20, 2015	<p>1.8 Updated Compatible chart by models.</p> <p>2.1 Update System Environment.</p> <p>3.4 Updated Restrictions</p> <p>5.3.1.8 Updated Note of GetUIDInfo method.</p> <p>5.3.1.9 Updated Note of GetSIDInfo method.</p> <p>5.4.1.1 Updated Description , Argument ,Note of GetDeviceStatus method.</p> <p>5.4.1.2 Updated Note of RecCtrl method.</p> <p>5.4.1.3 Updated Argument , Note of GetRecStatus method.</p> <p>5.4.1.4 Updated Argument of Search method.</p> <p>5.4.1.5 Updated Argument of SearchEx method.</p> <p>5.4.1.6 Updated Argument of VMDSearchEx method.</p> <p>5.4.1.11 Updated Note of GetInfoString method.</p> <p>5.4.1.12 Updated Note of SetCameraTime method.</p> <p>5.4.1.13 Updated Description of GetStatisticsData method.</p> <p>5.4.1.14 Updated Description of SetUIDPriority method.</p> <p>5.4.3.1 Updated Description , Argument of OnDevStatus event.</p> <p>5.4.3.3 Updated Description , Argument of OnRecStatus event.</p> <p>5.5.1.1 Updated Note of GetFrameTime method.</p> <p>5.5.1.2 Updated Argument of PlayLive method.</p> <p>5.5.1.3 Updated Argument of Play method.</p> <p>5.5.1.5 Updated Note of PlayControl method.</p> <p>5.5.1.8 Updated Note of GetPlaySpeed method.</p> <p>5.5.2.7 Updated Argument of H264Resolution property.</p> <p>5.5.2.8 Updated Argument of JPEGResolution property.</p> <p>5.5.2.15 Updated Note of InternetMode property.</p> <p>5.5.2.16 Updated Note of FastPlayMode property.</p> <p>5.5.3.1 Updated Note of OnPlayStatus event.</p> <p>5.7.1.1 Updated Argument of CameraControl method.</p> <p>5.7.1.4 Updated Argument of CameraOperation method.</p> <p>5.7.1.5 Updated Argument of GetCameraOpStatus method.</p> <p>5.7.1.6 Updated Argument of CameraCentering method.</p> <p>5.7.1.9 Updated Argument,Note of SetCameraImageCap.</p> <p>5.7.2.1 Updated Argument of OnOpStatus event.</p> <p>5.8.1.1 Updated Argument of AlmOperation event.</p> <p>5.8.2.1 Updated Argument of OnAlmStatus event.</p> <p>5.9.1.1 Updated Argument of FTPGet method.</p> <p>5.9.1.3 Updated Note of FTPServerClose method.</p>
9.0R01	July.17, 2015	<p>2.1 Update System Environment.</p> <p>5.5.1.11 Updated H264 black image size of GetImageResolution.</p> <p>5.5.2.7 Updated Argument of H264Resolution property.</p> <p>5.5.2.8 Updated Argument of JPEGResolution property.</p>

Version	Revised Date	Content of Revision
9.2 R01	Oct. 14, 2015	<p>1.7. Updated Function list.</p> <p>1.8. Updated Compatible chart by models.</p> <p>2.1 Added Microsoft® Windows® 10 Pro to System Environment.</p> <p>4.8.1 Added the restriction when using “Smart Coding” function of network camera.</p> <p>5.5.1.35 Added SetCroppingRect method.</p> <p>5.5.1.36 Added GetCroppingRect method.</p> <p>5.5.1.37 Added SetCroppingDrawRect method.</p> <p>5.5.1.38 Added GetCroppingDrawRect method.</p> <p>5.5.1.39 Added SetCroppingDrawEnable method.</p> <p>5.5.1.40 Added GetCroppingDrawEnable method.</p> <p>5.5.1.41 Added SetCroppingMarker method.</p> <p>5.5.1.42 Added GetCroppingMarker method.</p> <p>5.5.2.26 Added CroppingEnabled property</p> <p>6.16 Added Cropping function sequesnce</p> <p>7 Updated error code list.</p>
9.3 R01	Jan. 22, 2016	<p>2.1 Added Microsoft® Windows Server® 2012 Standard to System Environment.</p> <p>2.1 Added Microsoft® Windows Server® 2012 R2 Standard to System Environment.</p> <p>2.1 Deleted Microsoft® Windows Server® 2003 Standard 64 bit Edition to System Environment.</p> <p>2.1 Deleted Microsoft® Windows Server® 2003 Standard 32 bit Edition to System Environment.</p> <p>2.1 Deleted Microsoft® Windows Server® 2003 Enterprise 64 bit Edition to System Environment.</p> <p>2.1 Deleted Microsoft® Windows Server® 2003 Enterprise 32 bit Edition to System Environment.</p>
9.4 R01	Apr. 8, 2016	<p>5.4.1.12 Updated Note of SetCameraTime method.</p>

Version	Revised Date	Content of Revision
10.0 R01	Feb. 08, 2017	<p>1.7. Updated Function list.</p> <p>1.8. Updated Compatible chart by models.</p> <p>2.1. Updated System Environment</p> <p>3.4. Updated Restrictions</p> <p>5.3.1.8 Updated Note of GetUIDInfo</p> <p>5.3.1.9 Updated Note of GetSIDInfo method.</p> <p>5.3.2.4 Updated Note of HttpTimeout method.</p> <p>5.4.1.1 Updated Description , Argument ,Note of GetDeviceStatus method.</p> <p>5.4.1.2 Updated Note of RecCtrl method.</p> <p>5.4.1.3 Updated Argument,Note of GetRecStatus method.</p> <p>5.4.1.4 Updated Argument of Search method.</p> <p>5.4.1.5 Updated Argument of SearchEx method.</p> <p>5.4.1.6 Updated Argument of VMDSearchEx method.</p> <p>5.4.1.11 Updated Note of GetInfoString method.</p> <p>5.4.1.12 Updated Note of SetCameraTime method.</p> <p>5.4.1.13 Updated Description of GetStatisticsData method.</p> <p>5.4.1.14 Updated Description of SetUIDPriority method.</p> <p>5.4.3.1 Updated Description , Argument of OnDevStatus event.</p> <p>5.4.3.3 Updated Description , Argument of OnRecStatus event.</p> <p>5.5.1.1 Updated Note of GetFrameTime method.</p> <p>5.5.1.2 Updated Argument of PlayLive method.</p> <p>5.5.1.3 Updated Argument,Note of Play method.</p> <p>5.5.1.5 Updated Note of PlayControl method.</p> <p>5.5.1.8 Updated Note of GetPlaySpeed method.</p> <p>5.5.2.7 Updated Argument of H264Resolution property.</p> <p>5.5.2.8 Updated Argument of JPEGResolution property.</p> <p>5.5.2.15 Updated Note of InternetMode property.</p> <p>5.5.2.16 Updated Note of FastPlayMode property.</p> <p>5.7.1.1 Updated Argument of CameraControl method.</p> <p>5.7.1.4 Updated Argument of CameraOperation method.</p> <p>5.7.1.5 Updated Argument of GetCameraOpStatus method.</p> <p>5.7.1.6 Updated Argument of CameraCentering method.</p> <p>5.7.1.9 Updated Argument,Note of SetCameraImageCap method.</p> <p>5.7.1.10 Added CameraWiperControl method.</p> <p>5.7.2.1 Updated Argument of OnOpStatus event.</p> <p>5.8.1.1 Updated Argument of AlmOperation method.</p> <p>5.8.2.1 Updated Argument of OnAlmStatus event.</p> <p>5.9.1.1 Updated Argument,Note of FTPGet method.</p> <p>5.9.1.3 Updated Note of FTPServerClose method.</p> <p>7 Updated error code list.</p>



Version	Revised Date	Content of Revision
10.1 R01	Jun. 19, 2017	<p>Change company name.</p> <p>2.1 Deleted Microsoft® Windows® XP Professional SP3 from System Environment.</p> <p>2.1 Deleted Microsoft® Windows Vista® Business SP2 32 bit Edition from System Environment.</p> <p>3.4. Updated Restrictions</p> <p>5.3.2.2 Updated Note of IPAddr</p> <p>5.3.1.5 Updated Note of ProxyName</p> <p>5.3.1.8 Updated Note of GetUIDInfo</p> <p>5.3.1.9 Updated Note of GetSIDInfo method.</p> <p>5.4.1.1 Updated Description , Argument ,Note of GetDeviceStatus method.</p> <p>5.4.1.2 Updated Note of RecCtrl method.</p> <p>5.4.1.3 Updated Argument,Note of GetRecStatus method.</p> <p>5.4.1.4 Updated Argument of Search method.</p> <p>5.4.1.5 Updated Argument of SearchEx method.</p> <p>5.4.1.6 Updated Argument of VMDSearchEx method.</p> <p>5.4.1.11 Updated Note of GetInfoString method.</p> <p>5.4.1.12 Updated Note of SetCameraTime method.</p> <p>5.4.1.13 Updated Description of GetStatisticsData method.</p> <p>5.4.1.14 Updated Description of SetUIDPriority method.</p> <p>5.4.3.1 Updated Description , Argument of OnDevStatus event.</p> <p>5.4.3.3 Updated Description , Argument of OnRecStatus event.</p> <p>5.5.1.1 Updated Note of GetFrameTime method.</p> <p>5.5.1.2 Updated Argument of PlayLive method.</p> <p>5.5.1.3 Updated Argument,Note of Play method.</p> <p>5.5.1.5 Updated Note of PlayControl method.</p> <p>5.5.1.8 Updated Note of GetPlaySpeed method.</p> <p>5.5.2.15 Updated Note of InternetMode property.</p> <p>5.5.2.16 Updated Note of FastPlayMode property.</p> <p>5.7.1.1 Updated Argument of CameraControl method.</p> <p>5.7.1.4 Updated Argument of CameraOperation method.</p> <p>5.7.1.5 Updated Argument of GetCameraOpStatus method.</p> <p>5.7.1.6 Updated Argument of CameraCentering method.</p> <p>5.7.1.10 Added CameraWiperControl method.</p> <p>5.7.2.1 Updated Argument of OnOpStatus event.</p> <p>5.8.1.1 Updated Argument of AlmOperation method.</p> <p>5.8.2.1 Updated Argument of OnAlmStatus event.</p> <p>5.9.1.1 Updated Argument,Note of FTPGet method.</p> <p>5.9.1.3 Updated Note of FTPServerClose method.</p>

Version	Revised Date	Content of Revision
10.3 R01	Dec. 12, 2017	<p>1.6 Updated Overview of functions.</p> <p>1.7 Updated Function List.</p> <p>1.8 Updated Compatible chart by models.</p> <p>2.1 Updated System Environment.</p> <p>3.4 Updated Restrictions.</p> <p>4.2 Updated Connect to the device.</p> <p>4.3 Updated Restrictions of the usage when using the shared UID.</p> <p>4.5.3 Added MP4 file download.</p> <p>4.7 Updated image capture mode.</p> <p>4.8.1 Updated Smart Cording.</p> <p>5.1.2.2 Updated IPSAPI.</p> <p>5.1.2.3 Updated IAppListener.</p> <p>5.3.1.8 Updated Note of GetUIDInfo method.</p> <p>5.3.1.9 Updated Note of GetSIDInfo method.</p> <p>5.4.1.1 Updated Description , Argument ,Note of GetDevStatus method.</p> <p>5.4.1.2 Updated Note of RecCtrl method.</p> <p>5.4.1.3 Updated Argument ,Note of GetRecStatus method.</p> <p>5.4.1.4 Updated Argument, Note(sub stream of NX Series) of Search method.</p> <p>5.4.1.5 Updated Argument, Note(sub stream of NX Series) of SearchEx method.</p> <p>5.4.1.6 Updated Argument, Note(sub stream of NX Series) of VMDSearchEx method.</p> <p>5.4.1.11 Updated Note of GetInfoString method.</p> <p>5.4.1.12 Updated Note of SetCameraTime method.</p> <p>5.4.1.13 Updated Description of GetStatisticsData method.</p> <p>5.4.1.14 Updated Description of SetUIDPriority method.</p> <p>5.4.3.1 Updated Description, Argument of OnDevStatus Listener.</p> <p>5.4.3.3 Updated Description, Argument of OnRecStatus Listener.</p> <p>5.5.1.1 Updated Note of GetFrameTime method.</p> <p>5.5.1.2 Updated Argument, Note of PlayLive method.</p> <p>5.5.1.3 Updated Argument, Note(sub stream of NX Series) of Play method.</p> <p>5.5.1.4 Updated Note of PlayFile method.</p> <p>5.5.1.5 Updated Note of PlayControl method.</p> <p>5.5.1.8 Updated Note of GetPlaySpeed method.</p> <p>5.5.1.34 Updated Note of CamSnapShot method.</p> <p>5.5.1.43 Added HttpMP4Download method.</p> <p>5.5.1.44 Added GetMP4DownloadStatus method.</p> <p>5.5.1.45 Added GetMP4downloadTransRate method.</p> <p>5.5.2.7 Updated Argument of H264Resolution property.</p> <p>5.5.2.8 Updated Arugument of JPEGResolution property.</p> <p>5.5.2.14 Updated Note of StreamNumber.</p> <p>5.5.2.15 Added NXStreamNumber property.</p> <p>5.5.2.16 Updated Note of InternetMode property.</p> <p>5.5.2.17 Updated Note of FastPlayMode property.</p> <p>5.5.2.28 Added RcvAudioDec property.</p> <p>5.5.3.1 Updated Argument, Note of OnPlayStatus Listener.</p> <p>5.5.3.7 Added OnMP4DownloadStatus Listener.</p> <p>5.5.3.8 Added SetMP4DownloadListener.</p> <p>5.7.1.1 Updated Argument, Note of CameraControl method.</p> <p>5.7.1.2 Updated Note of SetCameraPosition method.</p>

Version	Revised Date	Content of Revision
10.3 R01	Dec. 12, 2017	5.7.1.4 Updated Argument, Note of CameraOperation method. 5.7.1.5 Updated Argument of GetCameraOpStatus method. 5.7.1.6 Updated Argument, Note of CameraCentering method. 5.7.1.9 Updated Argument, Note of SetCameraImageCap method. 5.7.1.10 Updated Argument of CameraWiperControl method. 5.7.2.1 Updated Argument of OnOpStatus Listener. 5.8.1.1 Updated Argument of AlmOperation method. 5.8.2.1 Updated Argument of OnAlmStatus Listener. 5.9.1.1.Updated Argument, Note of FTPGet method. 5.9.1.3 Updated Note of FTPServerClose method. 6.17 Added Operation Procedure, Sequence of HttpMP4Download. 7 Updated Error Code List.
11.00 R01	Mar. 23, 2018	1.8 Updated Compatible chart by models. 3.4 Updated Restrictions. 4.7 Updated image capture mode. 5.4.1.12 Updated Note of SetCameraTime method. 5.5.1.2 Updated Note of PlayLive method. 5.5.1.3 Updated Note of Play method. 5.5.1.4 Updated Note of PlayFile method. 5.5.2.14 Updated Note of StreamNumber property. 5.5.2.28 Updated Note of RcvAudioDec property. 5.7.1.1 Updated Note of CameraControl method. 5.7.1.2 Updated Argument of SetCameraPosition method. 5.7.1.3 Updated Argument of GetCameraPosition method. 5.7.1.4 Updated Argument,Note of CameraOperation method. 5.7.1.6 Updated Note of CameraCentering method. 5.7.1.9 Updated Argument of SetCameraImageCap method.
11.10 R01	Jun. 29, 2018	3.4. Updated Restrictions. 5.4.1.2 Updated Description of RecCtrl method. 5.5.1.2 Updated Argument of PlayLive method. 5.5.2.7 Updated Argument of H264Resolution property. 5.5.2.8 Updated Arugument of JPEGResolution property. 5.7.1.1 Updated Arugument of CameraControl method. 5.7.1.4 Updated Argument of CameraOperation method.

Version	Revised Date	Content of Revision
11.30 R01	Mar. 28, 2019	3.4 Updated Restrictions. 5.3.1.8 Updated Note of GetUIDInfo method. 5.3.1.9 Updated Note of GetSIDInfo method. 5.4.1.1 Updated Description , Argument of GetDevStatus method. 5.4.1.3 Updated Argument of GetRecStatus method. 5.4.1.4 Updated Argument of Search method. 5.4.1.5 Updated Argument of SearchEx method. 5.4.1.6 Updated Argument of VMDSearchEx method. 5.4.1.12 Updated Note of SetCameraTime method. 5.4.3.1 Updated Argument of OnDevStatus Listener. 5.4.3.3 Updated Argument of OnRecStatus Listener. 5.5.1.2 Updated Argument of PlayLive method. 5.5.1.3 Updated Argument of Play method. 5.5.1.5 Updated Note of PlayControl method. 5.5.1.8 Updated Note of GetPlaySpeed method. 5.5.1.34 Updated Argument of CamSnapShot method. 5.5.1.43 Updated Argument of HttpMP4Download method. 5.5.2.16 Updated Note of InternetMode property. 5.5.2.17 Updated Note of FastPlayMode property. 5.5.3.1 Updated Argument of OnPlayStatus Listener. 5.7.1.1 Updated Argument of CameraControl method. 5.7.1.4 Updated Argument of CameraOperation method. 5.7.1.5 Updated Argument of GetCameraOpStatus method. 5.7.1.6 Updated Argument of CameraCentering method. 5.7.1.10 Updated Argument of CameraWiperControl method. 5.7.2.1 Updated Argument of OnOpStatus Listener. 5.8.1.1 Updated Argument of AlmOperation method. 5.8.2.1 Updated Argument of OnAlmStatus Listener. 5.9.1.1.Updated Argument of FTPGet method.
11.40 R01	Jun. 26, 2019	1.4 Updated Abbreviations. 2.2 Updated Development Environment. 5.5.2.7. Updated Argument for SET of H264Resolution property. 5.5.2.8. Updated Argument for SET of JPEGResolution property. 5.7.1.1 Updated Note of CameraControl method. 5.7.1.2 Updated Note of SetCameraPosition method. 5.7.1.4 Updated Note of CameraOperaton method. 5.7.1.6 Updated Note of CameraCentering method. 7 Updated error code list.
11.60 R01	May. 29, 2020	Change company name. 2.1 Updated System Environment. 4.3 Updated Restrictions of the usage when using the shared UID. 5.3.2.11 Added SecureCommunicationMode property. 5.5.2.29 Added TransIntervalMode property. 5.5.2.30 Added DecBufferNum property. 5.5.3.5. Updated Argument of OnImage Listener. 5.5.3.6. Updated Argument of SetImageListener. 6.18 Added Operation Procedure, Sequence of SSL. 7 Updated error code list.

Version	Revised Date	Content of Revision
12.00 R01	May.31, 2021	1.5 Updated Structures of PS-API. 1.6 Updated Overview of functions. 1.7 Updated Function List. 1.8 Updated Compatible chart by models. 4.7 Updated the description of 360-degree network camera. 5.1.2.2 Updated IPSAPI. 5.3.2.11 Updated Note for SecureCommunicationMode property. 5.4.1.12 Updated Note of SetCameraTime method. 5.5.1.3 Updated Note of Play method. 5.5.1.44. Added HttpDownload method. 5.5.1.45. Updated Description of GetMP4DownloadStatus method. 5.5.1.46. Updated Description of GetMP4DownloadTransRate method. 5.5.2.7. Updated Argument for SET of H264Resolution property. 5.5.2.8. Updated Argument for SET of JPEGResolution property. 5.5.3.7. Updated Description of OnMP4DownloadStatus listener. 5.5.3.8 Updated Description of SetMP4DownloadListener listener. 5.7.1.4 Updated Argument of CameraOperaton method. 5.7.1.9 Updated Argument of SetCameraImageCap method. 5.8.1.1. Updated Argument , note of AlmOperation method 7 Updated error code list.
12.10 R01	Aug.31, 2021	5.4.1.12 Updated Note of SetCameraTime method. 5.5.2.16 Updated Note of InternetMode property
12.30 R01	Jan.31, 2022	4.7 Updated the description of 360-degree network camera. 5.4.1.12 Updated Note of SetCameraTime method. 5.5.2.7. Updated Argument for SET of H264Resolution property. 5.5.2.8. Updated Argument for SET of JPEGResolution property. 5.7.1.4 Updated Argument of CameraOperaton method. 5.9.1.1.Updated Description of FTPGet method. 7 Updated error code list.
12.50 R01	Apr.1, 2022	Change company name. 2.1 Added Microsoft® Windows® 11 Pro and Windows Server® 2019 Standard to System Environment. 5.3.2.2 Updated Note of IPAddr. 5.3.2.11 Updated Note of SecureCommunicationMode property. 5.4.1.12 Updated Note of SetCameraTime method. 5.5.1.2 Updated Argument of PlayLive method. 5.5.1.34 Updated Argument of CamSnapShot method. 5.7.1.1 Updated Argument of CameraControl method. 5.7.1.4 Updated Argument of CameraOperation method. 5.9.1.1.Updated Description of FTPGet method. 6.19 Added Operation Procedure,Sequence of MultiPlayLive_SID.
12.70 R01	Jun.30, 2022	4.3 Updated Restrictions of the usage when using the shared UID. 5.4.1.12 Updated Note of SetCameraTime method. 5.7.1.1 Updated Note of CameraControl method. 5.7.1.2 Updated Note of SetCameraPosition method. 5.7.1.4 Updated Argument , Note of CameraOperaton method. 5.7.1.6 Updated Note of CameraCentering method.

Version	Revised Date	Content of Revision
12.80 R01	Sep.30, 2022	2.1 Added Microsoft® Windows Server® 2022 Standard to System Environment. 2.1 Deleted Microsoft® Windows® 8 Pro from System Environment. 5.6.1.1. Updated Note of AudioSend method. 5.6.2.1. Updated Note of AudioRcvEnable property.

# INDEX

1. Preface .....	1
1.1. What's PS-API .....	1
1.2. Trademarks and Registered Trademarks .....	1
1.3. Limitation of liability .....	1
1.4. Abbreviations .....	1
1.5. Structures of PS-API .....	2
1.6. Overview of Functions .....	3
1.7. Function List .....	4
1.8. Compatible chart by models .....	13
2. System requirements .....	16
2.1. System Environment .....	16
2.2. Development Environment .....	17
2.3. Supported i-PRO Products .....	17
3. Setup .....	18
3.1. Product .....	18
3.2. Install .....	20
3.3. UnInstall .....	20
3.4. Restrictions .....	21
4. Overview of Library .....	23
4.1. Connect to the device .....	23
4.2. Connect to the device (Advanced usage for connection with recorder) .....	24
4.3. Restrictions of the usage when using the shared UID .....	25
4.4. Steps to control the device with PS-API .....	29
4.5. Relationship between PS-API and Device .....	30
4.5.1. Display video images .....	30
4.5.2. FTP download .....	33
4.5.3. MP4 or n3 file download via HTTP .....	34
4.6. Sync/Async Transaction Operation .....	35
4.6.1. Synchronous method call .....	35
4.6.2. Asynchronous method call .....	36
4.6.3. Receiving a notification by using Listener .....	37
4.7. About 360-degree Network Camera .....	38
4.7.1. How to get the image capture mode of 360-degree network camera .....	38
4.7.2. The explanation if image capture mode of 360-degree network camera .....	40
4.8. Restriction by device settings .....	46
4.8.1. Smart Cording .....	46
5. Details of DLL Class and Method .....	47
5.1. Class .....	47
5.1.1. Class Definition .....	47
5.1.2. Class Diagram .....	47
5.1.2.1. Global Function .....	48
5.1.2.2. IPSAPI .....	48
5.1.2.3. IAppListener .....	52
5.1.2.4. IAppCallBack .....	52
5.1.2.5. ISearchResult .....	53
5.1.2.6. ISearchResultEx .....	53
5.1.2.7. IPSAPIPicture .....	53
5.2. Global Function .....	54
5.2.1. GetIPSAPI .....	54
5.2.2. DeleteIPSAPI .....	56
5.3. PS Builder Group .....	58
5.3.1. Method .....	58
5.3.1.1. Open .....	58
5.3.1.2. Connect .....	60

5.3.1.3.	Close .....	62
5.3.1.4.	Disconnect .....	64
5.3.1.5.	ClearWaitingFunc.....	66
5.3.1.6.	GetWaitingFuncCount.....	68
5.3.1.7.	GetLoginStatus .....	70
5.3.1.8.	GetUIDInfo .....	72
5.3.1.9.	GetSIDInfo .....	74
5.3.2.	Property .....	76
5.3.2.1.	DeviceType .....	76
5.3.2.2.	IPAddr .....	78
5.3.2.3.	HttpPort .....	80
5.3.2.4.	HttpTimeout.....	82
5.3.2.5.	ProxyName .....	84
5.3.2.6.	ProxyPort .....	86
5.3.2.7.	AccessType.....	88
5.3.2.8.	UID .....	90
5.3.2.9.	UserName .....	92
5.3.2.10.	Password .....	94
5.3.2.11.	SecureCommunicationMode .....	96
5.3.3.	Application Listener .....	98
5.3.3.1.	OnError .....	98
5.3.3.2.	SetErrListener .....	100
5.4.	Device Group .....	102
5.4.1.	Method.....	102
5.4.1.1.	GetDevStatus .....	102
5.4.1.2.	RecCtrl .....	104
5.4.1.3.	GetRecStatus .....	106
5.4.1.4.	Search.....	108
5.4.1.5.	SearchEx.....	114
5.4.1.6.	VMDSearchEx .....	120
5.4.1.7.	SearchCancel .....	125
5.4.1.8.	GetDeviceLog .....	127
5.4.1.9.	GetDevTimeZone.....	129
5.4.1.10.	GetDevCurrentInfo.....	131
5.4.1.11.	GetInfoString .....	133
5.4.1.12.	SetCameraTime .....	135
5.4.1.13.	GetStatisticsData .....	136
5.4.1.14.	SetUIDPriority .....	138
5.4.2.	Property .....	140
5.4.2.1.	DevModel.....	140
5.4.2.2.	SearchMultiChMask.....	142
5.4.3.	Application Listener .....	144
5.4.3.1.	OnDevStatus .....	144
5.4.3.2.	SetDevListener .....	146
5.4.3.3.	OnRecStatus .....	148
5.4.3.4.	SetRecListener .....	150
5.4.4.	Application Callback .....	152
5.4.4.1.	OnRecCB.....	152
5.4.4.2.	OnSearchCB.....	154
5.4.4.3.	OnSearchExCB.....	156
5.5.	Video Group .....	158
5.5.1.	Method.....	158
5.5.1.1.	GetFrameTime .....	158
5.5.1.2.	PlayLive.....	161
5.5.1.3.	Play .....	163
5.5.1.4.	PlayFile .....	166
5.5.1.5.	PlayControl.....	168



5.5.1.6.	PlayControlByTime .....	172
5.5.1.7.	GetPlayStatus .....	174
5.5.1.8.	GetPlaySpeed .....	176
5.5.1.9.	GetFrameRate .....	179
5.5.1.10.	GetPicturePosition .....	181
5.5.1.11.	GetImageResolution .....	183
5.5.1.12.	SetVideoWindow.....	186
5.5.1.13.	DecodeImage.....	188
5.5.1.14.	ClearImage.....	190
5.5.1.15.	SnapShot .....	192
5.5.1.16.	TitleOperation.....	195
5.5.1.17.	TitleOperationEx .....	198
5.5.1.18.	GetTitle.....	201
5.5.1.19.	BoxOperation .....	203
5.5.1.20.	BoxOperationEx.....	205
5.5.1.21.	BitmapOperationEx.....	207
5.5.1.22.	DigitalZoomMove .....	209
5.5.1.23.	GetDigitalZoomPosition .....	211
5.5.1.24.	SetIntelligentView .....	213
5.5.1.25.	GetIntelligentView .....	213
5.5.1.26.	SetIntelligentViewColor.....	213
5.5.1.27.	GetIntelligentViewColor .....	213
5.5.1.28.	SetIntelligentViewSize .....	213
5.5.1.29.	GetIntelligentViewSize .....	213
5.5.1.30.	SetIntelligentViewTrackTime .....	213
5.5.1.31.	GetIntelligentViewTrackTime .....	213
5.5.1.32.	MultiSyncPause .....	214
5.5.1.33.	MultiSyncTime.....	216
5.5.1.34.	CamSnapShot.....	218
5.5.1.35.	SetCroppingRect.....	220
5.5.1.36.	GetCroppingRect .....	222
5.5.1.37.	SetCroppingDrawRect .....	224
5.5.1.38.	GetCroppingDrawRect.....	226
5.5.1.39.	SetCroppingDrawEnabled .....	228
5.5.1.40.	GetCroppingDrawEnabled.....	230
5.5.1.41.	SetCroppingMarker.....	232
5.5.1.42.	GetCroppingMarker .....	235
5.5.1.43.	HttpMP4Download .....	238
5.5.1.44.	HttpDownload .....	240
5.5.1.45.	GetMP4DownloadStatus.....	242
5.5.1.46.	GetMP4DownloadTransRate .....	243
5.5.2.	Property .....	244
5.5.2.1.	MPEG4Port.....	244
5.5.2.2.	H264Port .....	246
5.5.2.3.	RtpPortMode .....	248
5.5.2.4.	RtpPortRange .....	250
5.5.2.5.	MulticastAddr .....	252
5.5.2.6.	MPEG4Resolution .....	254
5.5.2.7.	H264Resolution .....	256
5.5.2.8.	JPEGResolution.....	258
5.5.2.9.	StreamFormat .....	260
5.5.2.10.	FilePassword .....	262
5.5.2.11.	ImageHeight.....	264
5.5.2.12.	ImageWidth .....	266
5.5.2.13.	MulticastAutoConf.....	268
5.5.2.14.	StreamNumber .....	270
5.5.2.15.	NXStreamNumber.....	272

5.5.2.16.	InternetMode .....	274
5.5.2.17.	FastPlayMode .....	276
5.5.2.18.	TransFrameRate .....	279
5.5.2.19.	PictureFitMode .....	281
5.5.2.20.	DigitalZoom .....	283
5.5.2.21.	DigitalZoomMode .....	285
5.5.2.22.	SkipRecordGap .....	287
5.5.2.23.	MultiScreenChannel .....	289
5.5.2.24.	SIDMode .....	291
5.5.2.25.	BackColor .....	293
5.5.2.26.	DecResolutionMode .....	295
5.5.2.27.	CroppingEnabled .....	297
5.5.2.28.	RcvAudioDec .....	299
5.5.2.29.	TransIntervalMode .....	301
5.5.2.30.	DecBufferNum .....	303
5.5.3.	Application Listener .....	304
5.5.3.1.	OnPlayStatus .....	304
5.5.3.2.	SetPlayListener .....	306
5.5.3.3.	OnRecordStatus .....	308
5.5.3.4.	SetRecordListener .....	310
5.5.3.5.	OnImage .....	312
5.5.3.6.	SetImageListener .....	318
5.5.3.7.	OnMP4DownloadStatus .....	320
5.5.3.8.	SetMP4DownloadListener .....	322
5.5.4.	Application Callback .....	324
5.5.4.1.	OnPlayStatusCB .....	324
5.6.	Audio Group .....	326
5.6.1.	Method .....	326
5.6.1.1.	AudioSend .....	326
5.6.1.2.	GetAudioSendStatus .....	328
5.6.2.	Property .....	330
5.6.2.1.	AudioRcvEnable .....	330
5.6.2.2.	AudioRcvVolume .....	332
5.6.2.3.	AudioRcvMute .....	334
5.6.2.4.	AudioSendVolume .....	336
5.6.2.5.	AudioSendMute .....	338
5.7.	Operation Group .....	340
5.7.1.	Method .....	340
5.7.1.1.	CameraControl .....	340
5.7.1.2.	SetCameraPosition .....	342
5.7.1.3.	GetCameraPosition .....	344
5.7.1.4.	CameraOperation .....	346
5.7.1.5.	GetCamOpStatus .....	349
5.7.1.6.	CameraCentering .....	351
5.7.1.7.	CameraAuxControl .....	353
5.7.1.8.	GetCameraAuxStatus .....	355
5.7.1.9.	SetCameraImageCap .....	357
5.7.1.10.	CameraWiperControl .....	360
5.7.2.	Application Listener .....	362
5.7.2.1.	OnOpStatus .....	362
5.7.2.2.	SetOpListener .....	364
5.7.3.	Application Callback .....	366
5.7.3.1.	OnOpStatusCB .....	366
5.8.	Alarm Group .....	368
5.8.1.	Method .....	368
5.8.1.1.	AlmOperation .....	368
5.8.1.2.	GetAlarmStatus .....	370

5.8.2.	Application Listener .....	372
5.8.2.1.	OnAlmStatus .....	372
5.8.2.2.	SetAlmListener.....	374
5.8.3.	Application Callback .....	376
5.8.3.1.	OnAlmStatusCB .....	376
5.9.	FTP Group .....	378
5.9.1.	Method.....	378
5.9.1.1.	FtpGet .....	378
5.9.1.2.	FtpCancel.....	383
5.9.1.3.	FtpServerClose .....	385
5.9.1.4.	GetFtpStatus .....	387
5.9.1.5.	GetFtpTransRate .....	389
5.9.1.6.	GetFtpTransByte.....	391
5.9.2.	Property .....	393
5.9.2.1.	FtpPort .....	393
5.9.2.2.	FtpTransMode.....	395
5.9.3.	Application Callback .....	397
5.9.3.1.	OnFtpStatusCB .....	397
6.	Operation Procedure and Sequence .....	399
6.1.	PlayLive.....	399
6.1.1.	Operation Procedure .....	399
6.1.2.	Sequence.....	402
6.2.	Play .....	404
6.2.1.	Operation Procedure .....	404
6.2.2.	Sequence.....	408
6.3.	PlayFile .....	412
6.3.1.	Operation Procedure .....	412
6.3.2.	Sequence.....	414
6.4.	CameraControl.....	416
6.4.1.	Operation Procedure .....	416
6.4.2.	Sequence.....	417
6.5.	CameraOperation .....	418
6.5.1.	Operation Procedure .....	418
6.5.2.	Sequence.....	421
6.6.	AlmOperation .....	423
6.6.1.	Operation Procedure .....	423
6.6.2.	Sequence.....	424
6.7.	Search.....	425
6.7.1.	Operation Procedure .....	425
6.7.2.	Sequence.....	427
6.8.	RecCtrl .....	428
6.8.1.	Operation Procedure .....	428
6.8.2.	Sequence.....	429
6.9.	MultiPlayLive.....	430
6.9.1.	Operation Procedure .....	430
6.9.2.	Sequence.....	431
6.10.	DecodelImage.....	433
6.10.1.	Operation Procedure.....	433
6.10.2.	Sequence .....	435
6.11.	FtpGet.....	436
6.11.1.	Operation Procedure .....	436
6.11.2.	Sequence.....	438
6.12.	Audio .....	439
6.12.1.	Operation Procedure.....	439
6.12.2.	Sequence .....	444
6.13.	SnapShot.....	448
6.13.1.	Operation Procedure.....	448

6.13.2.	Sequence .....	449
6.14.	Overlay .....	451
6.14.1.	Operation Procedure.....	451
6.14.2.	Sequence .....	453
6.15.	VMDSearch .....	455
6.15.1.	Operation Procedure.....	455
6.15.2.	Sequence .....	458
6.16.	Cropping .....	459
6.16.1.	Operation Procedure.....	459
6.16.2.	Sequence .....	461
6.17.	HttpMP4Download .....	462
6.17.1.	Operation Procedure.....	462
6.17.2.	Sequence .....	464
6.18.	SSL.....	465
6.18.1.	Operation Procedure.....	465
6.18.2.	Sequence .....	466
6.19.	MultiPlayLive_SID .....	467
6.19.1.	Operation Procedure.....	467
6.19.2.	Sequence .....	469
7.	Error Code List .....	471

## 1. Preface

### 1.1. What's PS-API

PS-API is the software library which is provided to help to develop the application to control i-PRO Co., Ltd. made security products (Network cameras, Network Disk Recorders, Digital Disk Recorders, Network Interface Units) by the unified method and to get video image.

PS-API is provided as the dynamic link library (DLL).

### 1.2. Trademarks and Registered Trademarks

Microsoft and Windows are registered trademarks of Microsoft Corporation in U.S. and/or other countries. Other names of companies and product contained in these operating instructions may be trademarks or registered trademarks of their respective owners.

### 1.3. Limitation of liability

- PS-API provides interfaces for the application software that controls i-PRO Co., Ltd. made security products (Network cameras, Network Disk Recorders, Digital Disk Recorders, Network Interface Units), and is not designed to protect against "theft" or "crime" independently.

- The provided sample programs are designed to instruct users how to use the SDK. They are not developed for the purpose of actual surveillance system.

- In not event shall i-PRO Co., Ltd. be liable to any party or any person, except for replacement or reasonable maintenance of the product, for the cases, including but not limited to below;

[1] Any damage and loss, including without limitation, direct or indirect, special, sequential or exemplary, arising out of or relating to the product;

[2] Personal injury or any damage caused by inappropriate use or neglect operation of the user;

[3] Any problems, consequential inconvenience, or loss or damage, arising out of the reverse compiling or reverse engineering of the product;

[4] Any loss or damage, or claims arising out from loss or leak of PS data including video data in the PC;

[5] Any claim or action for damages, brought by any person or organization being a photogenic subject, due to violation of privacy with the result of that surveillance camera's picture, including saved data, for some reason, becomes public or is used for the purpose other than surveillance;

### 1.4. Abbreviations

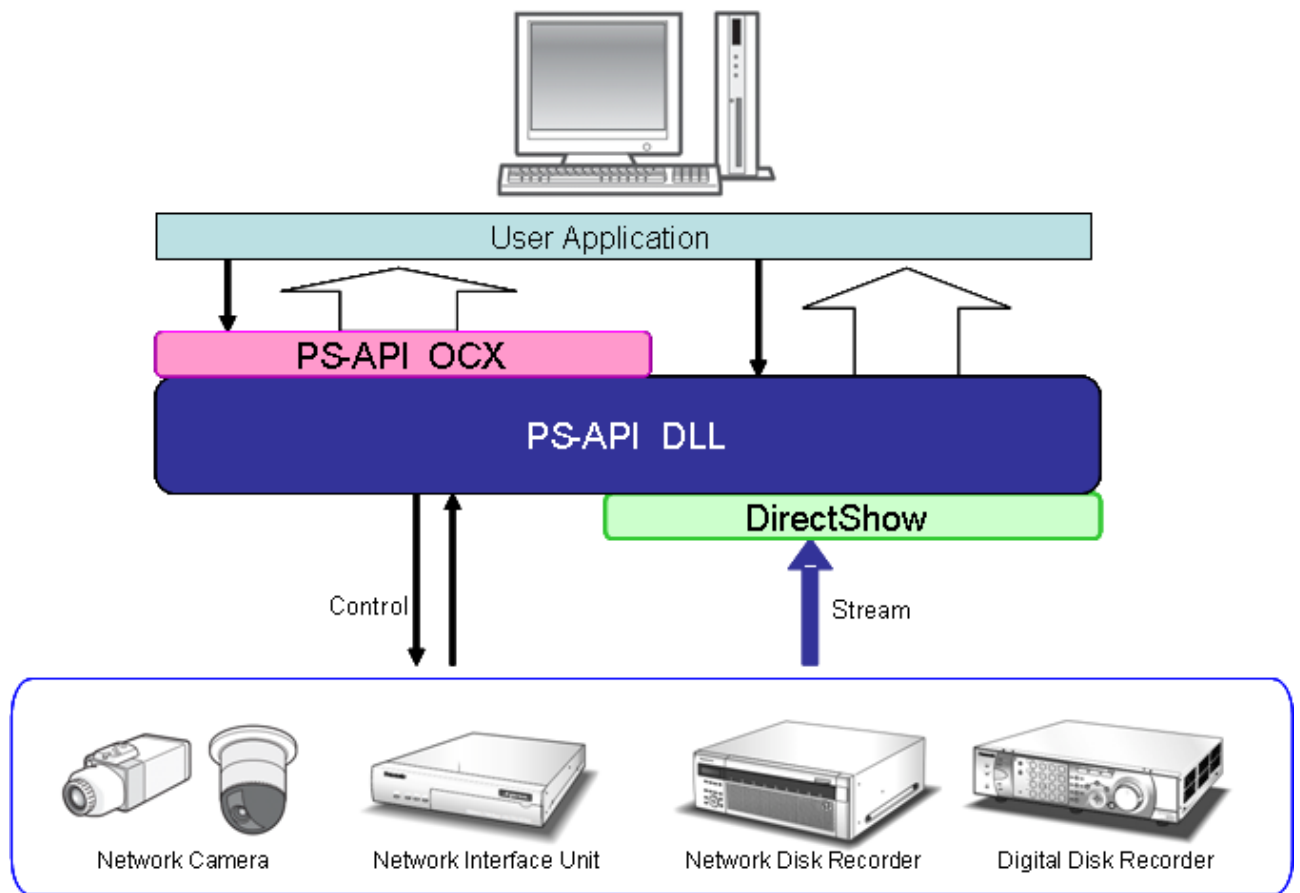
The following abbreviations are used in these operating instructions.

Microsoft® Visual C++ 2005 is described as Visual C++ 2005.

Microsoft® Visual C++ 2012 is described as Visual C++ 2012.

### 1.5. Structures of PS-API

Figure 1-1 shows the PS-API structures.



**Figure 1-1 PS-API Structures**

- \* Hereafter Network Camera is referred as Camera.  
Camera models are listed only for representative models of the series  
Please refer to [PS-API Supported Product List for English] document for camera series
- \* Hereafter Network Interface Unit is referred as Encoder.
- \* Hereafter Network Disk Recorder is referred as NWDR.
- \* Hereafter Network Disk Recorder (NX series) is referred as NX series.
- \* Hereafter Digital Disk Recorder (HD300 series) is referred as HD300.
- \* Hereafter Digital Disk Recorder (HD600 series, HD700 series) is referred as HD600/700.

## 1.6. Overview of Functions

Table 1-1 shows the overview of functions.

**Table 1-1 Function Overview**

No.	Overview	Reference
1	Create and Delete the IPSAPI instance	Global function
2	Login / Logout	PS Build Group
3	Search the recorded video data. (Network Disk Recorder, Digital Disk Recorder)	Device Group
4	Control recording. (Network Disk Recorder, Digital Disk Recorder)	
5	Display live image.	Video Group
6	Display playback image via network. (Network Disk Recorder, Digital Disk Recorder)	
7	Control playback such as Fast Forward and Rewind. (Network Disk Recorder, Digital Disk Recorder)	
8	Get and draw Compressed video image for one frame.	
9	Image recognition. (Only for supported devices)	
10	Digital zoom, Overlay	
11	Snap Shot	
12	MP4 or n3 file download (HTTP)	
13	Audio reception and transmission.	Audio Group
14	Control a camera. (Pan, Tilt, Zoom, Auto Back Focus functions, Super Dynamic function, etc)	Operation Group
15	Control AUX (Camera and Encoder)	
16	Control alarms.	Alarm Group
17	FTP download	FTP Group

## 1.7. Function List

Table 1-2 shows the functions provided in this PS-API.

**Table 1-2 List of Functions**

Method

No.	Class	Method	Overview	Reference
<i>Global Function</i>				
1	-	GetIPSAPI	Create the IPSAPI instance.	
2	-	DeleteIPSAPI	Delete the IPSAPI instance.	
<i>PS Builder Group</i>				
1	IPSAPI	Open	Log in to the device, and get UID.	
2	IPSAPI	Connect	Connect to the device with the specified UID which other instance got by the login.	
3	IPSAPI	Close	Stop the HTTP connection, and log out from the device.	
4	IPSAPI	Disconnect	Stop the HTTP connection.	
5	IPSAPI	ClearWaitingFunc	Cancel the waiting function of async type method.	
6	IPSAPI	GetWaitingFuncCount	Get the waiting function of async type method count.	
7	IPSAPI	GetLoginStatus	Get login result. If login failure, the reason of failure returns.	
8	IPSAPI	GetUIDInfo	Get the maximum number of UID that recorder can issue and the number of UID which is in use.	
9	IPSAPI	GetSIDInfo	Get the information of StreamID support, the maximum number of StreamID that recorder can issue, and the number of StreamID which is in use.	
10	IPSAPI	SetErrListener	Register OnError Listener function for an application.	



No.	Class	Method	Overview	Reference
<i>Device Group</i>				
1	IPSAPI	GetDevStatus	Get the device status.	
2	IPSAPI	RecCtrl	Turn on and off the manual recording.	
3	IPSAPI	GetRecStatus	Get recording status on a target recorder.	
4	IPSAPI	Search	Search the recording data on recording device. Time zone information is NOT included in search results.	
5	IPSAPI	SearchEx	Search the recording event. Time zone information is included in search results.	
6	IPSAPI	VMDSearchEx	Do VMD search.	
7	IPSAPI	SearchCancel	Cancel the executing Search/SearchEx/VMDSearchEx function.	
8	IPSAPI	GetDeviceLog	Get the recording device logs.	
9	IPSAPI	GetDevTimeZone	Get time zone information and summer time In/Out time table.	
10	IPSAPI	GetDevCurrentInfo	Get the device information and hold it.	
11	IPSAPI	GetInfoString	Get a value from the kept information that was gotten by GetDevCurrentInfo method.	
12	IPSAPI	SetCameraTime	Set specified time to a target device.	
13	IPSAPI	GetStatisticsData	Get statistics data from NV200, NV250, NV300 and NX Series.	
14	IPSAPI	SetUIDPriority	Change the UID priority mode of ND400, NV200, NV250, NV300, NX Series and HD600/700.	
15	IPSAPI	SetDevListener	Register OnDevStatus Listener function for an application.	
16	IPSAPI	SetRecListener	Register OnRecStatus Listener function for an application.	

No.	Class	Method	Overview	Reference
<i>Video Group</i>				
1	IPSAPI	GetFrameTime	Get the frame time and date of the recorded video play.	
2	IPSAPI	PlayLive	Start live video play.	
3	IPSAPI	Play	Start recorded video play.	
4	IPSAPI	PlayFile	Start video play from a specified video file.	
5	IPSAPI	PlayControl	Control video play that was started by PlayLive/Play/PlayFile method. (Forward, Pause, etc)	
6	IPSAPI	PlayControlByTime	Jump to the specified date and time,	
7	IPSAPI	GetPlayStatus	Get current video play status.	
8	IPSAPI	GetPlaySpeed	Get current video play speed.	
9	IPSAPI	GetFrameRate	Get current playing video stream frame rate.	
10	IPSAPI	GetPicturePosition	Get the position of the displayed picture except black panels.	
11	IPSAPI	GetImageResolution	Get the displayed image resolution.	
12	IPSAPI	SetVideoWindow	Specify a window handle for video display .	
13	IPSAPI	DecodeImage	Draw a Compressed video image of one frame on the window specified by using SetVideoWindow.	
14	IPSAPI	ClearImage	Clear the image that is displayed on the screen.	
15	IPSAPI	SnapShot	Get the image that is displayed on the screen.	
16	IPSAPI	TitleOperation	Draw text strings on the displayed image.	
17	IPSAPI	TitleOperationEx	Draw text strings on the displayed image. (with transmissivity)	
18	IPSAPI	GetTitle	Get the text strings that is displayed on the video image by using TitleOperation.	
19	IPSAPI	BoxOperation	Display frame lines on the video image.	
20	IPSAPI	BoxOperationEx	Display frame lines on the video image. (with transmissivity)	
21	IPSAPI	BitmapOperationEx	Display bitmap image on the video image. (with transmissivity)	

No.	Class	Method	Overview	Reference
22	IPSAPI	DigitalZoomMove	Move the displayed area during working a digital zoom.	
23	IPSAPI	GetDigitalZoomPosition	Get the current position of the displayed area in the original image with digital zoom.	
24	IPSAPI	MultiSyncPause	When using PlayControl with multi screen, pause to synchronize with other screen.	
25	IPSAPI	MultiSyncTime	When using PlayControl with multi screen, set the sync time to synchronize with other screen.	
26	IPSAPI	SetPlayListener	Register OnPlayStatus Listener function for an application.	
27	IPSAPI	SetRecordListener	Register OnRecordStatus Listener function for an application.	
28	IPSAPI	SetImageListener	Register OnImage Listener function for an application.	
29	IPSAPI	CamSnapShot	Get the snapshot image from NW camera or Encoder, and display it.	
30	IPSAPI	SetCroppingRect	Set the Cropping Position.	
31	IPSAPI	GetCroppingRect	Get the Cropping Position.	
32	IPSAPI	SetCroppingDrawRect	Set the Draw Position in Cropping function.	
33	IPSAPI	GetCroppingDrawRect	Get the Draw Position in Cropping function.	
34	IPSAPI	SetCroppingDrawEnabled	Set Hide/Show each id of Cropping function.	
35	IPSAPI	GetCroppingDrawEnabled	Get Hide/Show each id of Cropping function.	
36	IPSAPI	SetCroppingMarker	Set the Marker that indicate the Cropping Position.	
37	IPSAPI	GetCroppingMarker	Get the Marker that indicate the Cropping Position.	

No.	Class	Method	Overview	Reference
38	IPSAPI	HttpMP4Download	Download the MP4 file via HTTP from a NX Series.	
39	IPSAPI	HttpDownload	Download the MP4 or n3 file via HTTP from a NX Series.	
40	IPSAPI	GetMP4DownloadStatus	Get MP4 or n3 file downloading status.	
41	IPSAPI	GetMP4DownloadtransRate	Get transfer speed in MP4 or n3 file downloading.	
42	IPSAPI	SetMP4DownloadListener	Register OnMP4DownloadStatus Listener function for an application.	

No.	Class	Method	Overview	Reference
<i>Audio Group</i>				
1	IPSAPI	AudioSend	Start or Stop the audio transmission.	
2	IPSAPI	GetAudioSendStatus	Get the audio transmission status.	
<i>Operation Group</i>				
1	IPSAPI	CameraControl	Send Pan, Tilt, Focus, Zoom, Iris command to the specified camera.	
2	IPSAPI	SetCameraPosition	Set Pan, Tilt, Zoom and Focus values in absolute angle.	
3	IPSAPI	GetCameraPosition	Get Pan, Tilt, Zoom and Focus values in absolute angle.	
4	IPSAPI	CameraOperation	Operate camera such as auto track, auto pan, auto focus.	
5	IPSAPI	SetOpListener	Register OnOpStatus Listener function for an application.	
6	IPSAPI	CameraCentering	Control the camera to the specified position becomes the center.	
7	IPSAPI	CameraAuxControl	Control the AUX terminal of camera and encoder to OPEN/CLOSE.	
8	IPSAPI	GetCameraAuxStatus	Get AUX terminal status of camera and encoder.	
9	IPSAPI	SetCameraImageCap	Set a Image capture mode and installation of camera.	
10	IPSAPI	CameraWiperControl	Operate camera wiper function.	
<i>Alarm Group</i>				
1	IPSAPI	AlmOperation	Reset current device alarm state. Or trigger a device alarm by the application.	
2	IPSAPI	GetAlarmStatus	Get current alarm information.	
3	IPSAPI	SetAlmListener	The Set OnAlmStatus Listener function is to register a listener for an application so that PS-API can notify any alarm status changes.	

No.	Class	Method	Overview	Reference
<i>FTP Group</i>				
1	IPSAPI	FtpGet	Download the video/ audio/ recording event list via FTP from a recording device.	
2	IPSAPI	FtpCancel	Cancel FTP download.	
3	IPSAPI	FtpServerClose	Turn off the FTP server mode on a target device.	
4	IPSAPI	GetFtpStatus	Get Ftp downloading status.	
5	IPSAPI	GetFtpTransRate	Get transfer speed in FTP downloading.	
6	IPSAPI	GetFtpTransByte	Get transferred byte count in FTP downloading.	

## Application Listener

No.	Class	Method	Overview	Reference
<i>PS Builder Group</i>				
1	IAppListener	OnError	The OnError notification function is to pass error code from PS-API to a specified application.	
<i>Device Group</i>				
1	IappListener	OnDevStatus	The OnDevStatus notification function is to pass device connection status between recorder and camera from PS-API to a specified application.	
2	IappListener	OnRecStatus	The OnRecStatus notification function is to pass Recorder status from PS-API to a specified application.	
<i>Video Group</i>				
1	IappListener	OnPlayStatus	The OnPlayStatus notification function is to pass play status from PS-API to a specified application.	
2	IappListener	OnRecordStatus	The OnRecordStatus notification function is to pass the recording data status (beginning or the ending) from PS-API to a specified application.	
3	IappListener	OnImage	The OnImage notification function is to pass image data with specified type from PS-API to a specified application.	
4	IappListener	OnMP4DownloadStatus	The OnMP4DownloadStatus notification function is to pass MP4 or n3,file downloading status change from PS-API to a specified application.	
<i>Operation Group</i>				
1	IAppListener	OnOpStatus	The OnOpStatus notification function is to pass camera operation status from PS-API to a specified application.	
<i>Alarm Group</i>				
1	IAppListener	OnAlmStatus	The alarm status change notification function is to pass alarm status from PS-API to a specified application.	

## Application Callback

No.	Class	Method	Overview	Reference
<i>Device Group</i>				
1	IAppCallBack	OnRecCB	The Recording control callback function is to pass application recording control result from PS-API to a specified application.	
2	IAppCallBack	OnSearchCB	The Search callback function is to send the notification from PS-API to a specified application when the search finishes.	
3	IAppCallBack	OnSearchExCB	The SearchEx / VMDSearchEx callback function is to send the notification from PS-API to a specified application when the search finishes.	
<i>Video Group</i>				
1	IAppCallBack	OnPlayStatusCB	The play status callback function is to pass video play status from PS-API to a specified application.	
<i>Operation Group</i>				
1	IAppCallBack	OnOpStatusCB	The operation status callback function is to pass camera operation status from PS-API to a specified application request.	
<i>Alarm Group</i>				
1	IAppCallBack	OnAlmStatusCB	The alarm status callback function is to pass AlmOperation state from PS-API to a specified application request.	
<i>FTP Group</i>				
1	IAppCallBack	OnFtpStatusCB	The FTP status callback function is to pass FtpGet state from PS-API to a specified application request.	



## 1.8. Compatible chart by models

The following list shows the compatible chart by models.

**Table 1-3 Compatible Chart by Models**

No.	Method	Camera	NWDR	HD300	HD600/700	Encoder	NX Series	Remarks
<i>PS Builder Group</i>								
1	Open	Yes	Yes	Yes	Yes	Yes	Yes	
2	Connect	Yes	Yes	Yes	Yes	Yes	Yes	
3	Close	Yes	Yes	Yes	Yes	Yes	Yes	
4	Disconnect	Yes	Yes	Yes	Yes	Yes	Yes	
5	ClearWaitingFunc	Yes	Yes	Yes	Yes	Yes	Yes	
6	GetWaitingFuncCount	Yes	Yes	Yes	Yes	Yes	Yes	
7	GetLoginStatus	Yes	Yes	Yes	Yes	Yes	Yes	
8	GetUIDInfo	Yes	Yes	Yes	Yes	Yes	Yes	
9	GetSIDInfo	Yes	Yes	Yes	Yes	Yes	Yes	
10	SetErrListener	Yes	Yes	Yes	Yes	Yes	Yes	
<i>Device Group</i>								
1	GetDevStatus	Yes	Yes	Yes	Yes	Yes	Yes	
2	RecCtrl	-	*1	Yes	Yes	-	-	
3	GetRecStatus	-	Yes	Yes	Yes	-	Yes	
4	Search	-	Yes	Yes	Yes	-	Yes	
5	SearchEx	-	Yes	Yes	Yes	-	Yes	
6	VMDSearchEx	-	*2	Yes	Yes	-	*2	
7	SearchCancel	-	Yes	Yes	Yes	-	Yes	
8	GetDeviceLog	-	Yes	Yes	Yes	-	Yes	
9	GetDevTimeZone	Yes	Yes	*3	Yes	Yes	Yes	
10	GetDevCurrentInfo	Yes	Yes	Yes	Yes	Yes	Yes	
11	GetInfoString	Yes	Yes	Yes	Yes	Yes	Yes	
12	SetCameraTime	Yes	-	-	-	Yes	-	
13	GetStatisticsData	—	*12	—	—	—	Yes	
14	SetUIDPriority	—	*13	—	Yes	—	Yes	
15	SetDevListener	Yes	Yes	Yes	Yes	Yes	Yes	
16	SetRecListener	-	Yes	Yes	Yes	-	Yes	

No.	Method	Camera	NWDR	HD300	HD600/700	Encoder	NX Series	remarks
<b>Video Group</b>								
1	GetFrameTime	*4	*5	*6	Yes	Yes	Yes	
2	PlayLive	Yes	Yes	Yes	Yes	Yes	Yes	
3	Play	-	Yes	Yes	Yes	-	Yes	
4	PlayFile	-	-	-	-	-	-	
5	PlayControl	Yes	Yes	*7	Yes	Yes	Yes	
6	PlayControlByTime	-	Yes	Yes	Yes	-	Yes	
7	GetPlayStatus	Yes	Yes	Yes	Yes	Yes	Yes	
8	GetPlaySpeed	Yes	Yes	Yes	Yes	Yes	Yes	
9	GetFrameRate	*8	*9	*8	*8	*8	*9	
10	GetPicturePosition	Yes	Yes	Yes	Yes	Yes	Yes	
11	GetImageResolution	Yes	Yes	Yes	Yes	Yes	Yes	
12	SetVideoWindow	Yes	Yes	Yes	Yes	Yes	Yes	
13	DecodeImage	-	-	-	-	-	-	
14	ClearImage	Yes	Yes	Yes	Yes	Yes	Yes	
15	SnapShot	Yes	Yes	Yes	Yes	Yes	Yes	
16	TitleOperation	Yes	Yes	Yes	Yes	Yes	Yes	
17	TitleOperationEx	Yes	Yes	Yes	Yes	Yes	Yes	
18	GetTitle	Yes	Yes	Yes	Yes	Yes	Yes	
19	BoxOperation	Yes	Yes	Yes	Yes	Yes	Yes	
20	BoxOperationEx	Yes	Yes	Yes	Yes	Yes	Yes	
21	BitmapOperationEx	Yes	Yes	Yes	Yes	Yes	Yes	
22	DigitalZoomMove	Yes	Yes	Yes	Yes	Yes	Yes	
23	GetDigitalZoomPosition	Yes	Yes	Yes	Yes	Yes	Yes	
24	MultiSyncPause	-	Yes	-	Yes	-	Yes	
25	MultiSyncTime	-	Yes	-	Yes	-	Yes	
26	SetPlayListener	Yes	Yes	Yes	Yes	Yes	Yes	
27	SetRecordListener	-	Yes	Yes	Yes	-	Yes	
28	SetImageListener	Yes	Yes	Yes	Yes	Yes	Yes	
29	CamSnapShot	Yes	-	-	-	Yes	-	
30	SetCroppingRect	Yes	Yes	Yes	Yes	Yes	Yes	
31	GetCroppingRect	Yes	Yes	Yes	Yes	Yes	Yes	
32	SetCroppingDrawRect	Yes	Yes	Yes	Yes	Yes	Yes	
33	GetCroppingDrawRect	Yes	Yes	Yes	Yes	Yes	Yes	
34	SetCroppingDrawEnabled	Yes	Yes	Yes	Yes	Yes	Yes	
35	GetCroppingDrawEnabled	Yes	Yes	Yes	Yes	Yes	Yes	
36	SetCroppingMarker	Yes	Yes	Yes	Yes	Yes	Yes	
37	GetCroppingMarker	Yes	Yes	Yes	Yes	Yes	Yes	
38	HttpMP4Download	-	-	-	-	-	Yes	
39	HttpDownload	-	-	-	-	-	Yes	
40	GetMP4DownloadStatus	-	-	-	-	-	Yes	
41	GetMP4DownloadTransRate	-	-	-	-	-	Yes	
<b>Audio Group</b>								
1	AudioSend	Yes	-	-	-	Yes	-	
2	GetAudioSendStatus	Yes	-	-	-	Yes	-	

No.	Method	Camera	NWDR	HD300	HD600/700	Encoder	NX Series	remarks
<b>Operation Group</b>								
1	CameraControl	*10	*10	*10	*10	*10	*10	
2	SetCameraPosition	*10	-	-	-	-	-	
3	GetCameraPosition	*10	-	-	-	-	-	
4	CameraOperation	*10	*10	*10	*10	*10	*10	
5	CameraCentering	*10	*10	*10	*10	*10	*10	
6	CameraAuxControl	Yes	-	-	-	Yes	-	
7	GetCameraAuxStatus	*10	-	-	-	*10	-	
8	SetOpListener	*10	*10	*10	*10	*10	*10	
9	SetCameraImageCap	*14	—	—	—	—	—	
10	CameraWiperControl	*15	*15	—	—	—	*15	
<b>Alarm Group</b>								
1	AlmOperation	*11	Yes	*11	Yes	*11	Yes	
2	GetAlarmStatus	Yes	Yes	Yes	Yes	Yes	Yes	
3	SetAlmListener	Yes	Yes	Yes	Yes	Yes	Yes	
<b>FTP Group</b>								
1	FtpGet	-	Yes	Yes	Yes	-	Yes	
2	FtpCancel	-	Yes	Yes	Yes	-	Yes	
3	FtpServerClose	-	Yes	Yes	Yes	-	Yes	
4	GetFtpStatus	-	Yes	Yes	Yes	-	Yes	
5	GetFtpTransRate	-	Yes	Yes	Yes	-	Yes	
6	GetFtpTransByte	-	Yes	Yes	Yes	-	Yes	

\*1 : NV200 ,NV250 and NV300 is not supported.

\*2 : ND200 and ND300 are not supported.

\*3 : In case of HD300, time zone information cannot be got and is set to "0".

\*4 : NP1000 is not supported.

\*5 : In case that the target device is ND200, time zone depends on the device's time zone.

And in case that the target device is ND300 or ND400, time zone depends on the connected camera's time zone

\*6 : Time zone cannot be got and is set to "0".

\*7 : In case of network playback, to specify playback speed directly is not supported for HD300.

\*8 : In case of PlayLive, frame rate cannot be got and is set to "0".

\*9 : In case of MPEG-4/H.264/H.265(not JPEG), frame rate cannot be got and is set to "0" or "300".

\*10 : Depends on models whether PTZ, Preset, Auto back focus, Super Dynamic, AUX control, preset sequence, auto sort and patrol are supported or not.

\*11 : Can specify No-operation and Reset control only.

\*12 : When NVF20,NVF30 is available, the function of statistics download works.

\*13 : Only ND400 , NV200 ,NV250 and NV300 is supported.

\*14 : Only 360-degree Network Camera (SF438 series, SW458 series, SF448 series, SFV481 series, S4550 series, X4571 series, S4551 series, X4573 series) is supported.

\*15 : SUD638 is only supported.

## 2. System requirements

### 2.1. System Environment

The following table shows the PC specification that is needed for installing PS-API.

**Table 2-1 OS**

OS	Microsoft® Windows® 7 Professional SP1 32 bit Edition(*5)
	Microsoft® Windows® 7 Professional SP1 64 bit Edition(*5)
	Microsoft® Windows® 8.1 Pro 32 bit Edition(*2)
	Microsoft® Windows® 8.1 Pro 64 bit Edition(*2)
	Microsoft® Windows® 10 Pro 32 bit Edition
	Microsoft® Windows® 10 Pro 64 bit Edition
	Microsoft® Windows® 11 Pro
	Microsoft® Windows Server® 2008 R2 Standard SP1 (*1)(*5)
	Microsoft® Windows Server® 2008 R2 Enterprise SP1 (*1)(*5)
	Microsoft® Windows Server® 2012 Standard(*1)
	Microsoft® Windows Server® 2012 R2 Standard(*1)
	Microsoft® Windows Server® 2016 Standard(Desktop Experience)
	Microsoft® Windows Server® 2019 Standard(Desktop Experience)
	Microsoft® Windows Server® 2022 Standard(Desktop Experience)

**Table 2-2 Minimum System requirements**

Processor(*3)	Intel® Core™2 Quad 2.66GHz or more
Memory(*3)	2.0GB or more
Hard drive	10GB or more
LAN	100Mbps or more

**Table 2-3 Recommended system requirements**

Processor(*3)	Intel® Core™i7-4790
Memory(*3)	8.0GB or more(*4)
Hard drive	10GB or more
LAN	100Mbps or more

(\*1) When using PS-API, it is necessary to install “Desktop Experience” to use Direct Show Filter functions.

(\*2) Modern UI is not supported.

(\*3) Requirement spec depends on the screen number and the video setting.  
When using for the multiple screen application and/or retrieving H.264/H.265 video image,  
please use PS-API with the recommended system requirements.

In case of H.264/H.265 video streaming, you may be able to display mode video streams by the PS-API settings which DecResolutionMode is set to 3 and the value of H264Resolution is reduced.

(\*4) Please use two same Memories that dual-channel supported.

(\*5) Windows® Internet Explorer® 11 or later is recommended.

## 2.2. Development Environment

The following table shows the development environment list that is supported by PS-API DLL.

**Table 2-4 Development Environment**

Target	Development Tool
PS-API DLL	Visual C++ 2005 SP1
	Visual C++ 2012

## 2.3. Supported i-PRO Products

Please refer to **[PS-API Supported Product List for English]** document.

### 3. Setup

#### 3.1. Product

Please refer to [PS-API Installation Guide for DLL] document.

**Table 3-1 The Overview of Files**

[DLL folder]

Directory Name	Objective
PS-API¥ For Development	Header files / lib files For development. <b>NOT REDISTRIBUTABLE.</b>
PS-API¥ Redistributable	DLL files Redistributable.
PS-API¥ Setup	Installer for DirectShow filters. For development. <b>NOT REDISTRIBUTABLE.</b>
PS-API¥ Tool	Test tool. By using this tool, you can confirm connecting to a target device with PS-API. <b>NOT REDISTRIBUTABLE.</b>
PS-API¥ Document	The users manual. (This document) <b>NOT REDISTRIBUTABLE.</b>

Directory Name	Objective
PS-ALARM¥ For Development	Header files / lib files For development. <b>NOT REDISTRIBUTABLE.</b>
PS-ALARM¥ Redistributable	DLL files Redistributable.
PS-ALARM¥ Document	The users manual. <b>NOT REDISTRIBUTABLE.</b>

Directory Name	Objective
PS-LOOKUP¥ For Development	Header files / lib files For development. <b>NOT REDISTRIBUTABLE.</b>
PS-LOOKUP¥ Redistributable	DLL files Redistributable.
PS-LOOKUP¥ Document	The users manual. <b>NOT REDISTRIBUTABLE.</b>

[DLL-Sample folder]

Directory Name	Objective
Sample Program	Sample Programs. <b>NOT REDISTRIBUTABLE.</b>

### 3.2. Install

Please refer to [**PS-API Installation Guide for DLL**] document.

### 3.3. UnInstall

Please refer to [**PS-API Installation Guide for DLL**] document.



### 3.4. Restrictions

- (1) Do not use the Power Schemes or System Standby mode of Windows®.
- (2) PS-API instance is NOT thread safe.
- (3) Use Multibyte Character Set.
- (4) PS-API controls the Digital Disk Recorder/Network Disk Recorder with a UID provided from the Digital Disk Recorder/Network Disk Recorder. The maximum number of provided UIDs is shown on the following table.

Recorder	Maximum UID number
HD300	8
ND200	4
ND300	8
ND400	16
NV200 (*1)	4
NV250(*1)	4
NV300(*1)	8
NX100(*1)	16
NX200(*1)	16
NX300(*1)	16
NX400(*1)	16
HD600/700	8

\*1 : In case of NV200, NV250, NV300, NX Series up to 16 instances can work PlayLive/Play with the sharing UID at the same time.

Refer to "4.2 Connect to the device (Advanced usage for connection with recorder)" for further information about UID.

- (5) In case of connection with Digital Disk Recorder/Network Disk Recorder, when 8 UIDs (in case of ND400 : 16 UIDs) are already used and a new user is going to get a UID, the priority of UID publication depends on the target device setting.
- (6) When "ON" is selected for "User Authentication" on the Digital Disk Recorder/Network Disk Recorder, a user with a higher user level will have a priority to have a valid UID and a UID given for the instance of a user with a lower user level will be invalidated. (Refer to the operating instructions of the Digital Disk Recorder/Network Disk Recorder for further information about "User Authentication").
- (7) The limitation of stream count that transmit live video from network camera or encoder is different between models or usage conditions. Regarding the details, please refer to CGI specification document of network camera.
- (8) The audio reception supports G.726 and AAC. The audio transmission supports only G.726.
- (9) In case of using multiple instances, please do not use AAC Format.
- (10) The application that includes PS-API must be compiled by 32bit(x86) setting.

- (11) Video delay may occur in the case of [resolution:2992\*2992, frame rate:30ips] stream.  
In that case, it may be improved with one of the following settings.

Proposal(1) Set the transmission priority of the camera to [VBR].

\*When there is temporary large number of video changes, video delay occurs.  
After that, when the video change decreases, the delay is gradually eliminated.

Proposal(2) Set the camera bit rate to 3.5Mbps or less.

\*Please check the image delay and image quality and set appropriate values.

Proposal(3) please use the camera's frame rate "15ips(default)".

- (12) [resolution: 2192x2192, frame rate:30ips] stream if it is displayed on 4K monitor, video delay may occur. In that case, it may be improved by adding a graphic board.  
(\*Operation confirmed graphic board: NVIDIA Quadro P600)

## 4. Overview of Library

### 4.1. Connect to the device

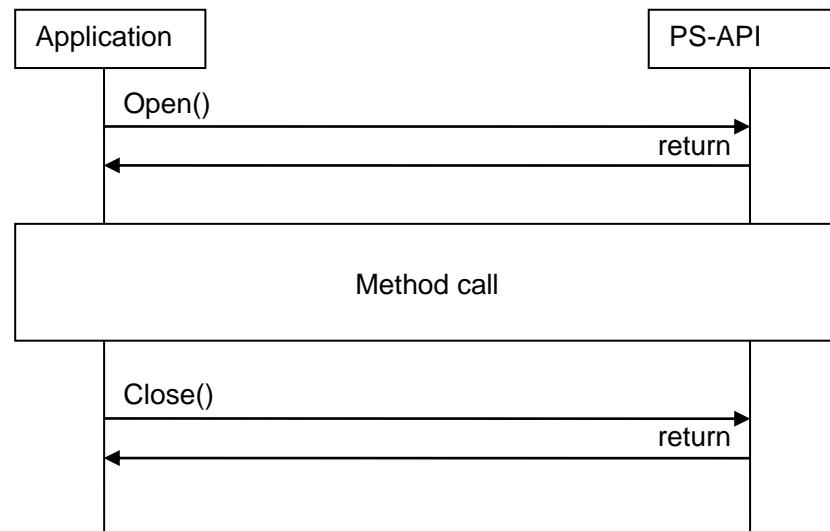
When the application connects to a network camera or a recorder by using PS-API, it is necessary that IP address, User name and Password are set first, and then Open() method is called.

(Refer to 5.3.1.1 Open.)

After execute Open() method, Application can use every method.

When stop connecting to the device, Application disconnects by using Close() method.

(Refer to 5.3.1.3 Close.)



**Figure 4-1 Connect to The Device**

#### 4.2. Connect to the device (Advanced usage for connection with recorder)

A recorder has the limitation of user connection count, and the connection count is managed by the count of UID that is delivered by a device.

When using Open() method, application gets one UID.

If you want to save the connection count (UID) when displaying multiple live images for plural channels, you can use Connect() method with one UID that one PS-API instance gets by using Open() method.

(Refer to 5.3.1.2 Connect)

If stop the connection that is started by using Connect(), Application disconnects by using Disconnect() method.

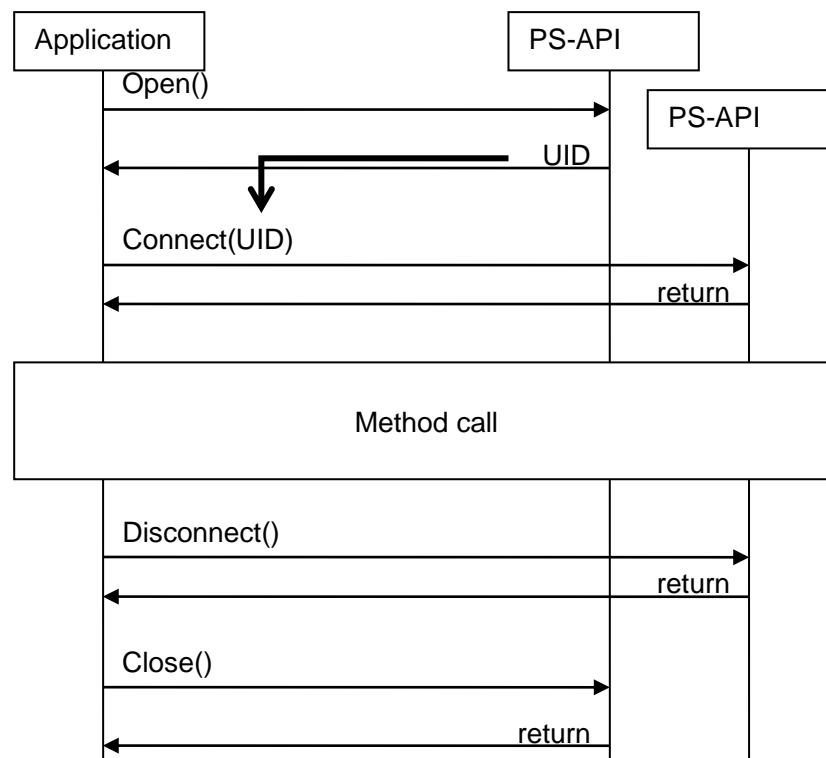
(Refer to 5.3.1.4 Disconnect)

#### NOTE

The following functions can work by using the shared UID. (\*1) But when using the shared UID, there are some restrictions of the usage. Please refer to 4.3 Restrictions of the usage when using the shared UID.

- (1) Live.
- (2) FTP download
- (3) Network playback
- (4) MP4 or n3 file download via HTTP

\*1 : In case of NV200, NV250, NV300, NX Series up to 16 instances can work PlayLive/Play with the sharing UID at the same time.



**Figure 4-2 Connect to The Device (Advanced usage for connection with recorder)**

#### 4.3. Restrictions of the usage when using the shared UID

The instance that starts connection with Open() (hereafter referred as Instance-A) and the instance that starts connections with Connect() (hereafter referred as Instance-B) can be used for only Live, for only FTP download, for only MP4 or n3 file download via HTTP or for only network playback.

##### (1) Live

###### **Case:**

###### **SIDMode property is 1, and Recorder that supports StreamID Mode**

- On Instance-A and Instance-B, user can operate Live freely.

###### **Case:**

###### **SIDMode property is 0, or Recorder that does not support StreamID Mode**

- When using Instance-A for Live, Instance-B should be used for Live.
- When Instance-A and Instance-B share the same UID, they cannot show the same channel.

About StreamID Mode supported device or not is refer to [GetSIDInfo] method.

##### (2) FTP download

- When using Instance-A for FTP download, Instance-B should be used for FTP download.
- There is a limit to the number of the files which API can download at the same time.

##### (3) MP4 or n3 file download via HTTP

- When using Instance-A for file download, Instance-B cannot file download.

(4) Network playback

**Case:**

**SIDMode property is 1, Recorder that supports StreamID Mode**

- On Instance-A and Instance-B, user can operate Network playback freely.

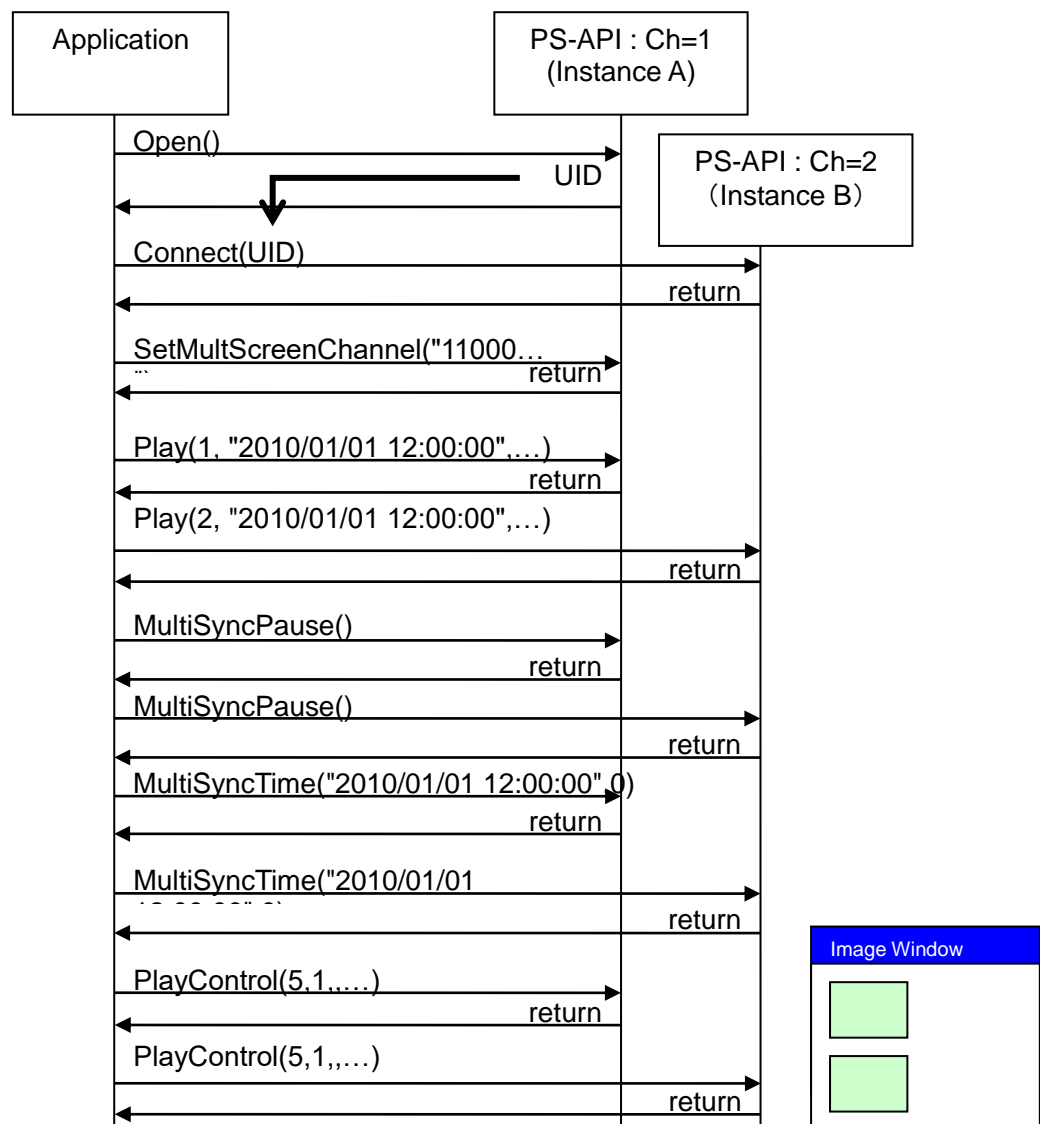
**Case:**

**SIDMode property is 0, or Recorder that does not support StreamID Mode**

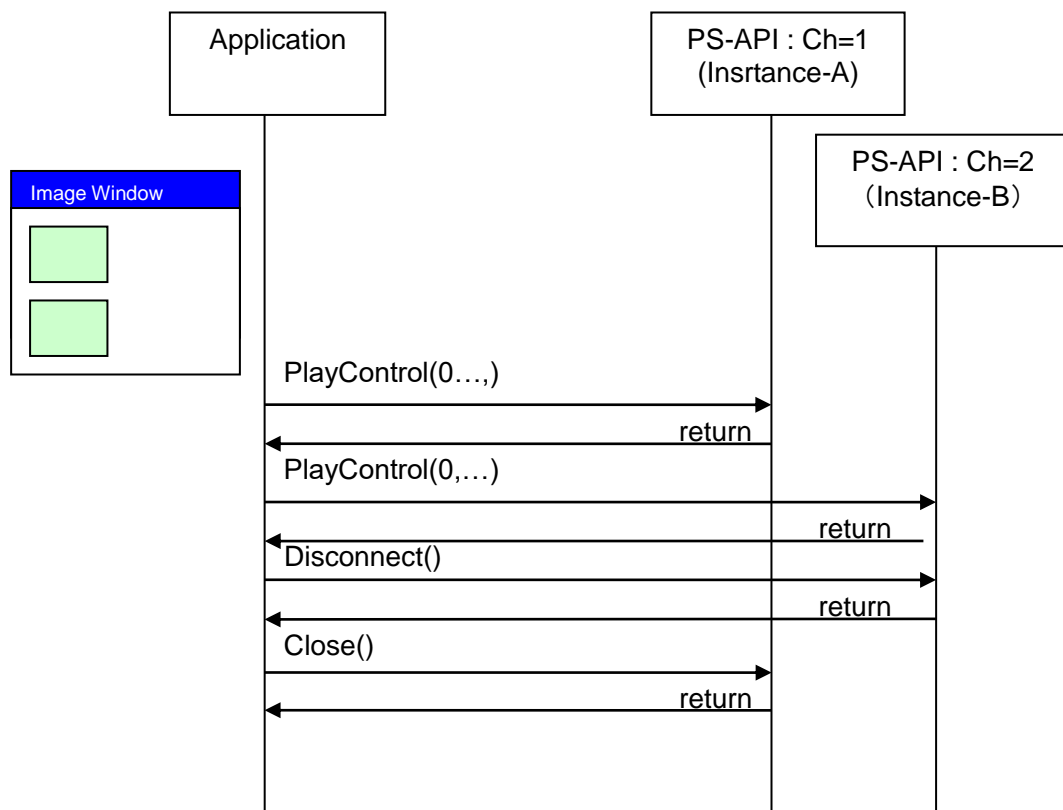
- When using Instance-A for network playback and playback control, Instance-B should be used for network playback and playback control.
- Before starts network playback, please specify the channels that you want to start playback by using SetMultiChannelScreen(). This method should be called at one instance.  
If you call SetMultiChannelScreen() during network playback, unexpected result will be happen.
- When Instance-A and Instance-B share the same UID, they cannot show the same channel.
- When starts network playback with Play(), specify the same time and date to all instances that share UID.
- When starts playback control (Playback, Backward, Fast forward, Rewind) with PlayControl(), please calling MultiSyncPause method and MultiSyncTime method, and then specify the same operation to all instances that share UID. But PlayControl() cannot do Next frame, Previous frame, Next record, Previous record control.

About StreamID Mode supported device or not is refer to [GetSIDInfo] method.

[ND200 and ND300]



**Figure 4-3 Start a playback with Multiple-Channel(Advanced usage for connection with recorder)**



**Figure 4-4 Stop a playback with Multiple-Channel(Advanced usage for connection with recorder)**



#### 4.4. Steps to control the device with PS-API

The following chart shows the flow that create instance, connect to device, register a window, start live, control PTZ and stop.

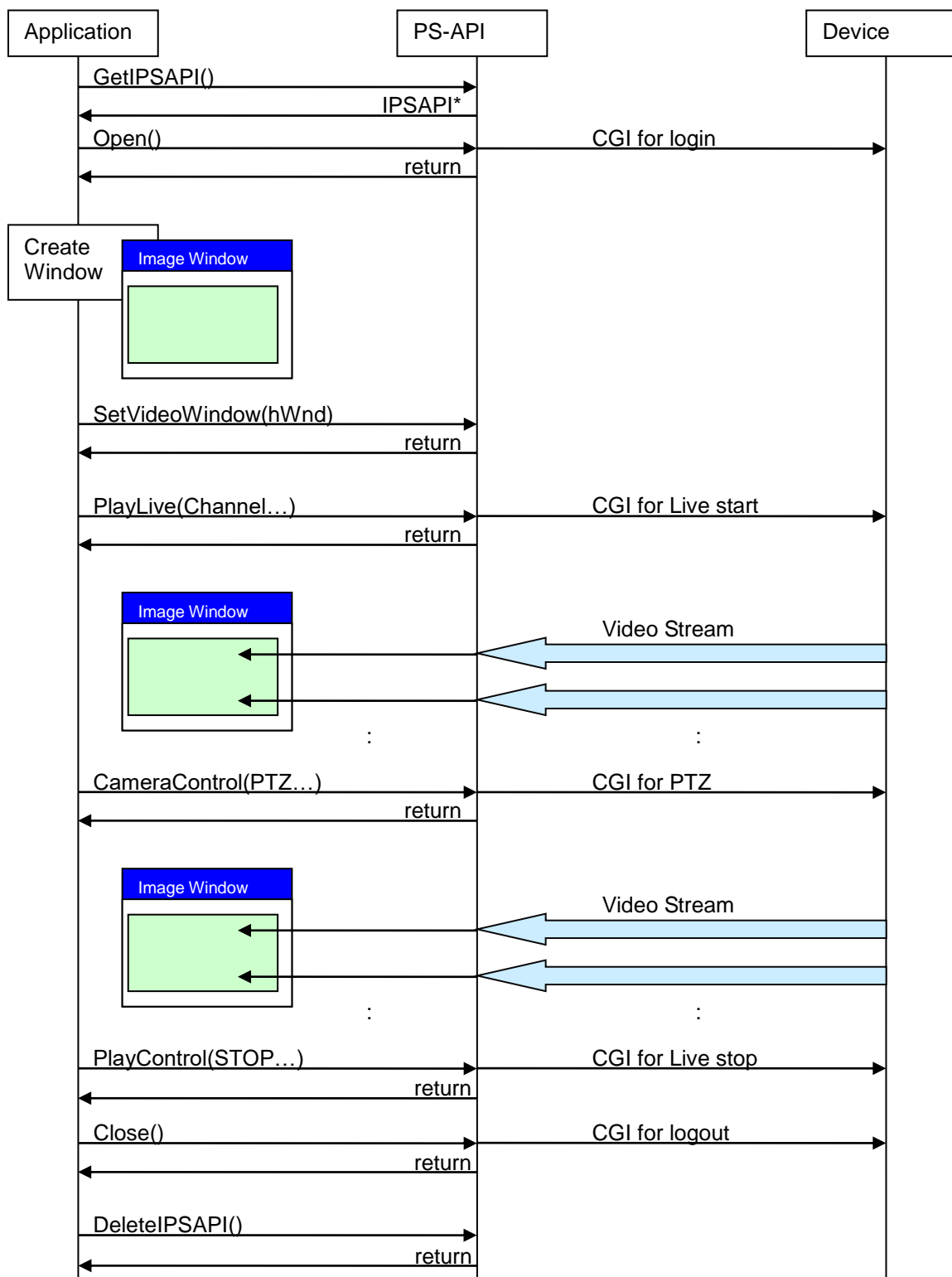


Figure 4-5 Steps to Control the Device with PS-API

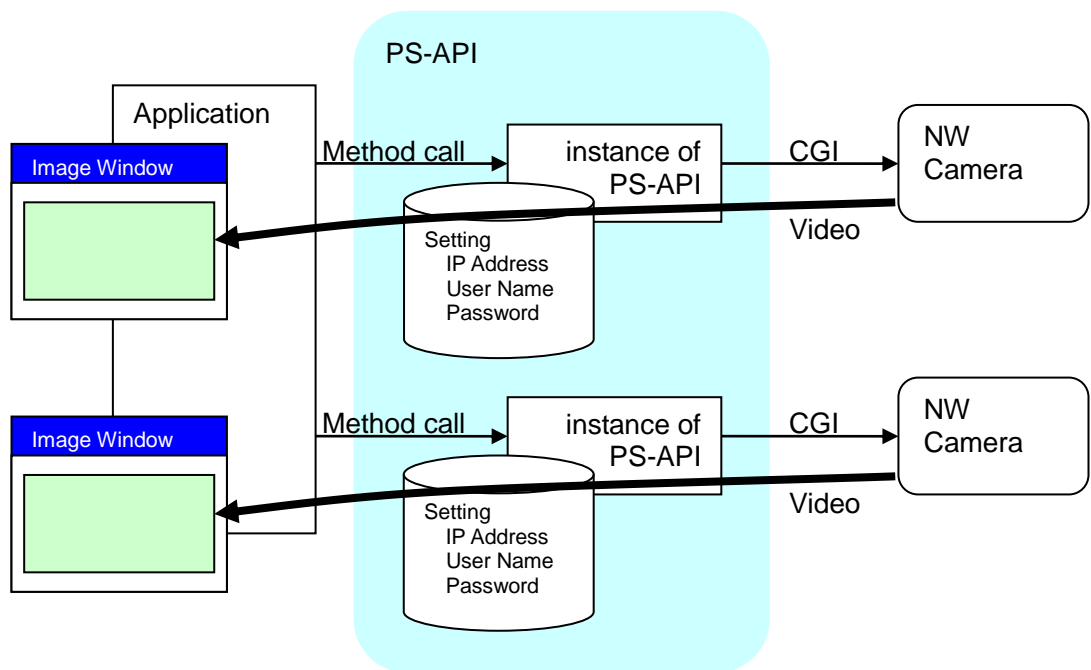
#### 4.5. Relationship between PS-API and Device

This chapter shows the relationship between PS-API and a device in case that an application connects to Network Camera or Recorder by using PS-API.

##### 4.5.1. Display video images

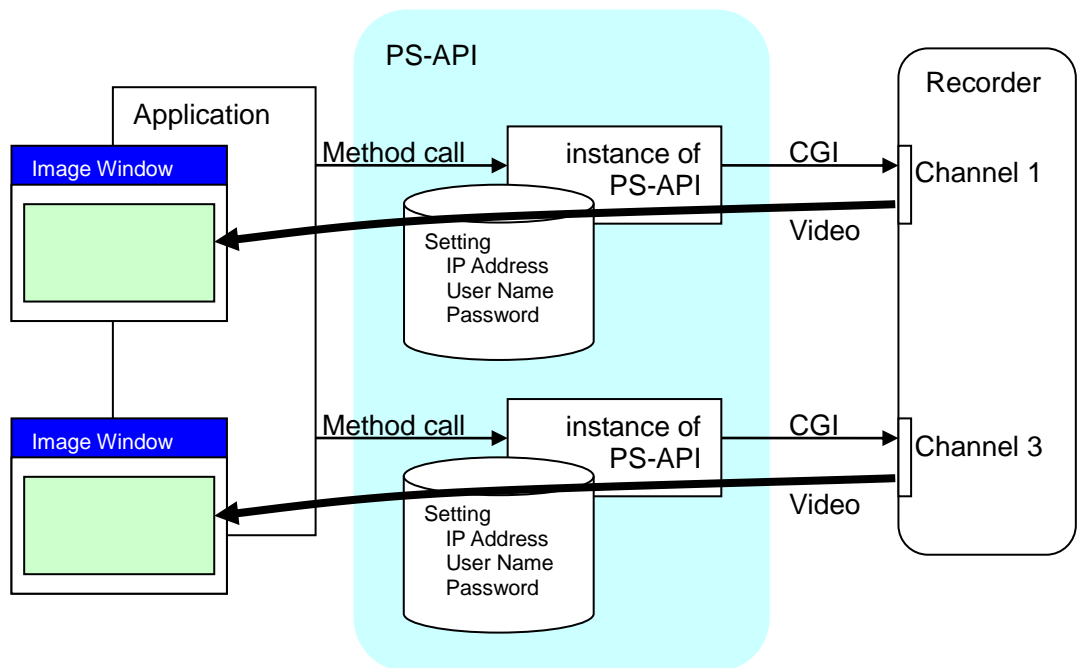
One PS-API instance displays one video stream. The plural video streams cannot be displayed at the same time by using one PS-API instance.

- Connect to network cameras



**Figure 4-6 Connect to Network Cameras**

- Connect to a recording device



**Figure 4-7 Connect to the Recording Device**

It is possible to control plural devices from one instance.

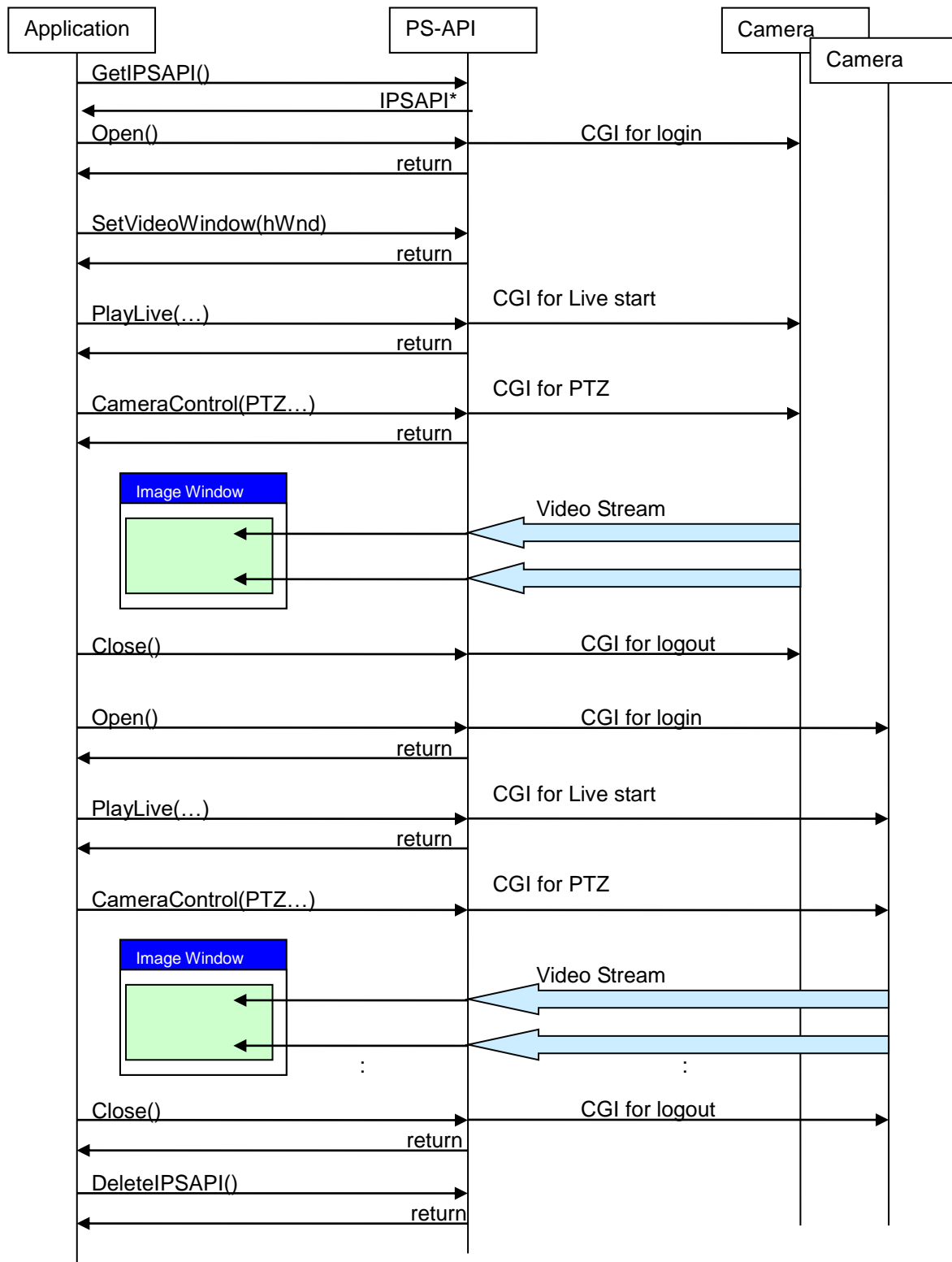
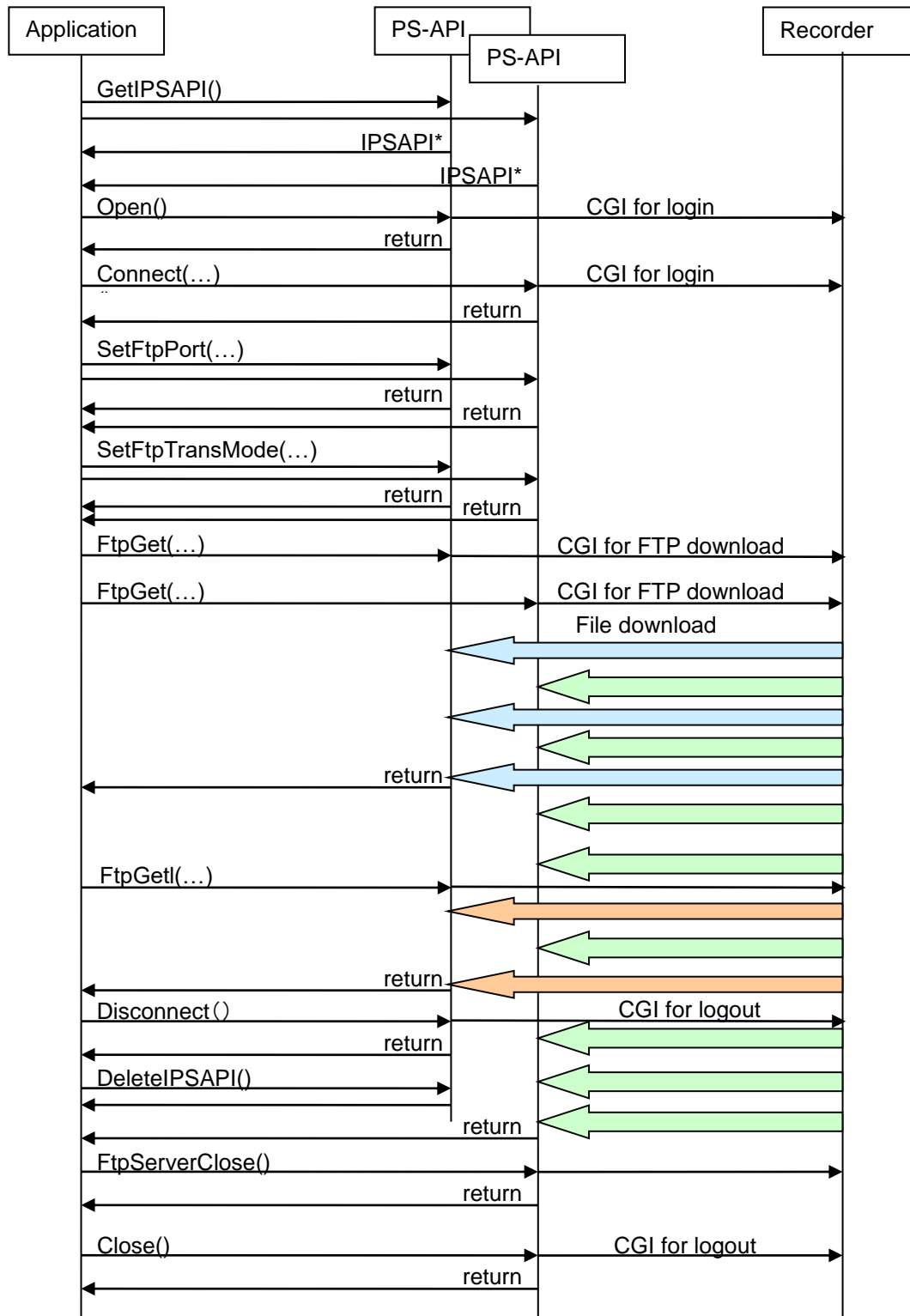


Figure 4-8 Control Plural Devices from One Instance

#### 4.5.2. FTP download

Multiple files cannot be downloaded by using one PS-API instance at the same time. Please call the next download method after finish downloading. If using multiple instances, you can download several files at the same time.



### Figure 4-9 FTP Download from Multiple Instances

#### 4.5.3. MP4 or n3 file download via HTTP

Multiple files cannot be downloaded by using one PS-API instance at the same time. Please call the next download method after finish downloading. If using multiple instances, you can download several files at the same time.

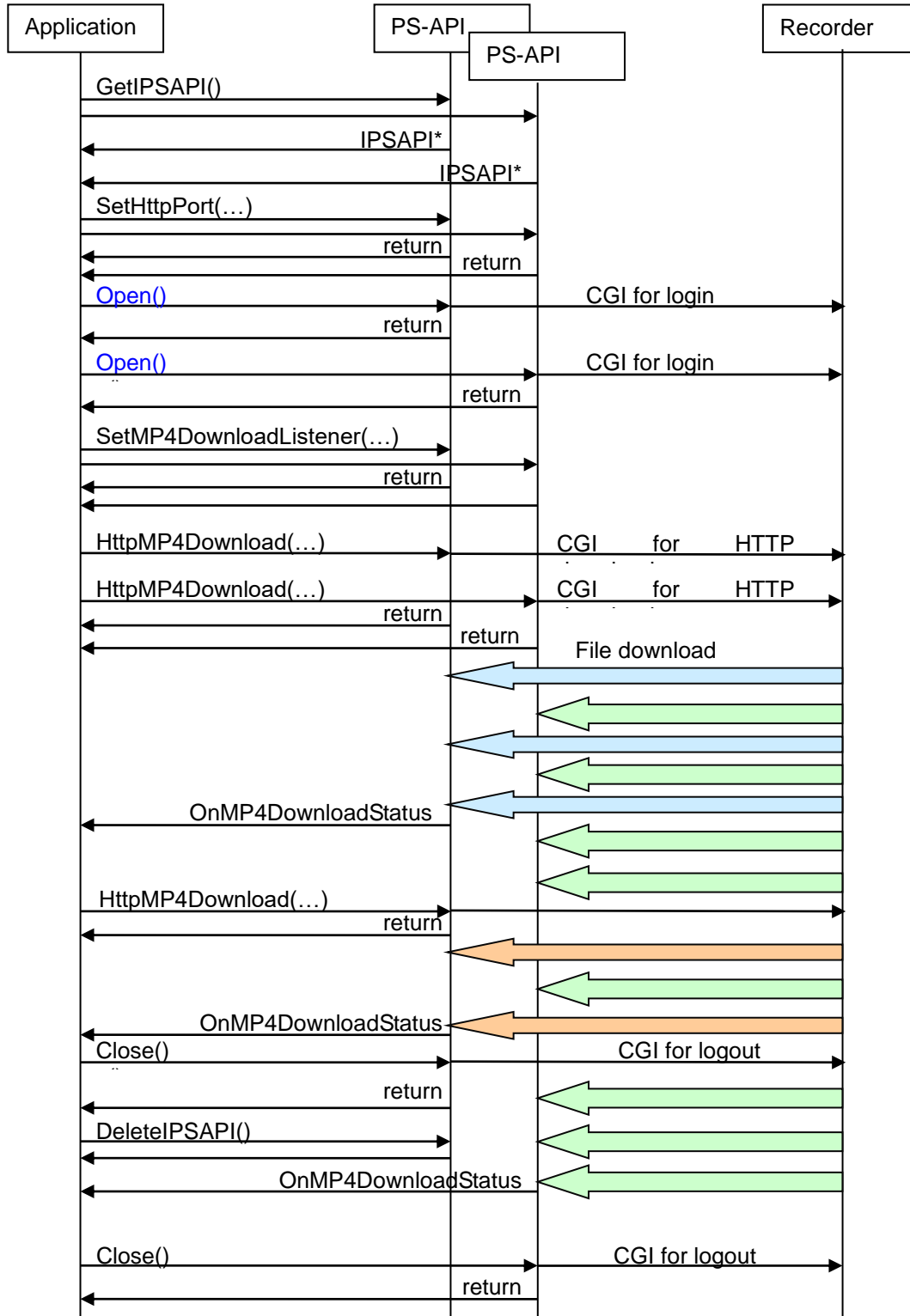


Figure 4-10 MP4 Download from Multiple Instances

#### 4.6. Sync/Async Transaction Operation

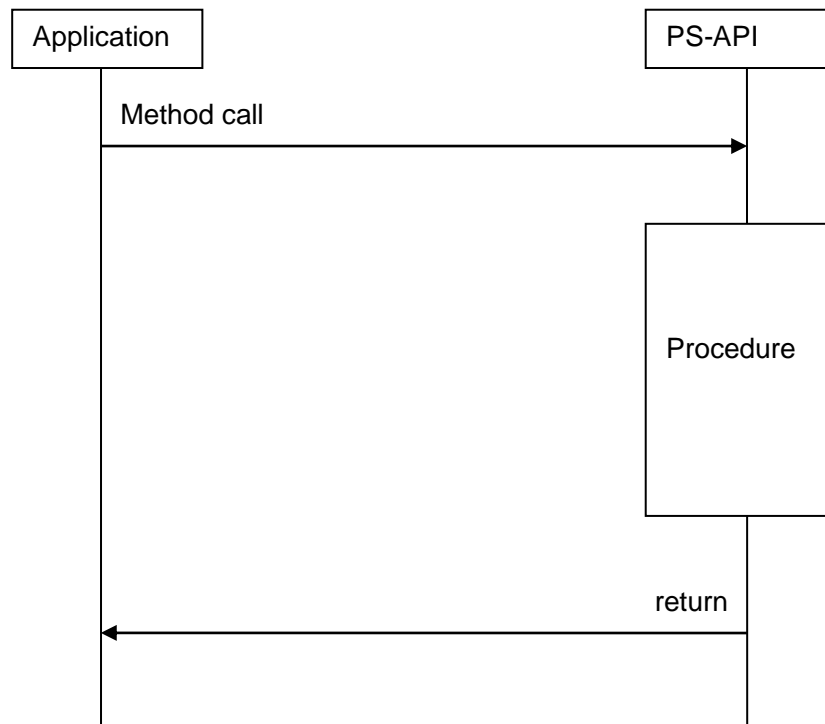
In the case of using a PS-API instance, there are two ways of calling a method and a way of receiving a notification.

1. Synchronous method call
2. Asynchronous method call
3. Receiving a notification by using Listener

##### 4.6.1. Synchronous method call

Synchronous method call is the general way. When a method of PS-API instance is called, the operation is blocked until the processing will be finished.

When calling the synchronous method during processing the asynchronous method, the synchronous method returns error because of “in processing”.



**Figure 4-11 Synchronous Method Call**

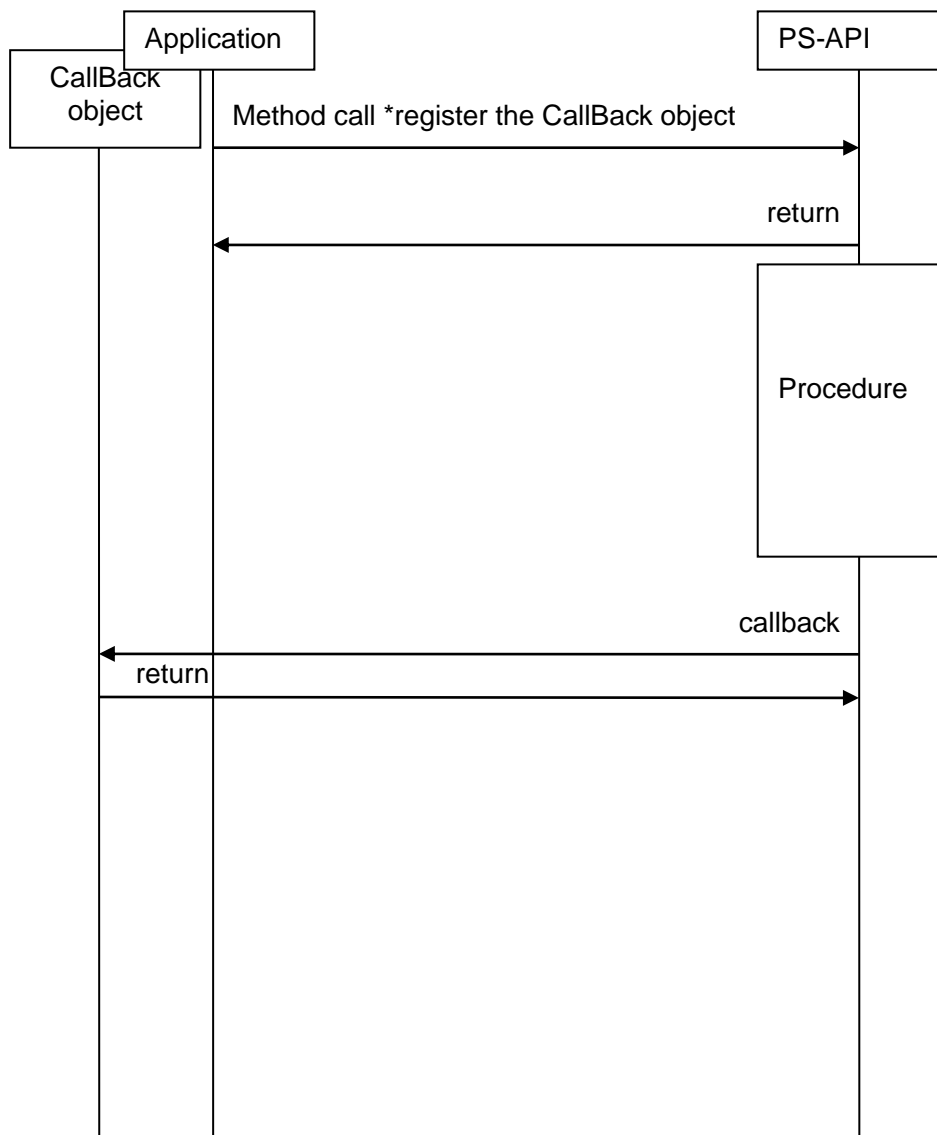
#### 4.6.2. Asynchronous method call

Some methods can be used with the asynchronous mode. When the method of PS-API instance is called with asynchronous mode, PS-API starts processing, and then returns the return value before finishing processing.

When calling the asynchronous method during processing the asynchronous method, the method is in the queue and it will process on after another.

The asynchronous processing result of PS-API is notified via Callback object. Callback object is an instantiated Callback class that Callback interface class prepared by PS-API is implemented.

\* Don't destruct the callback object before the callback object called.



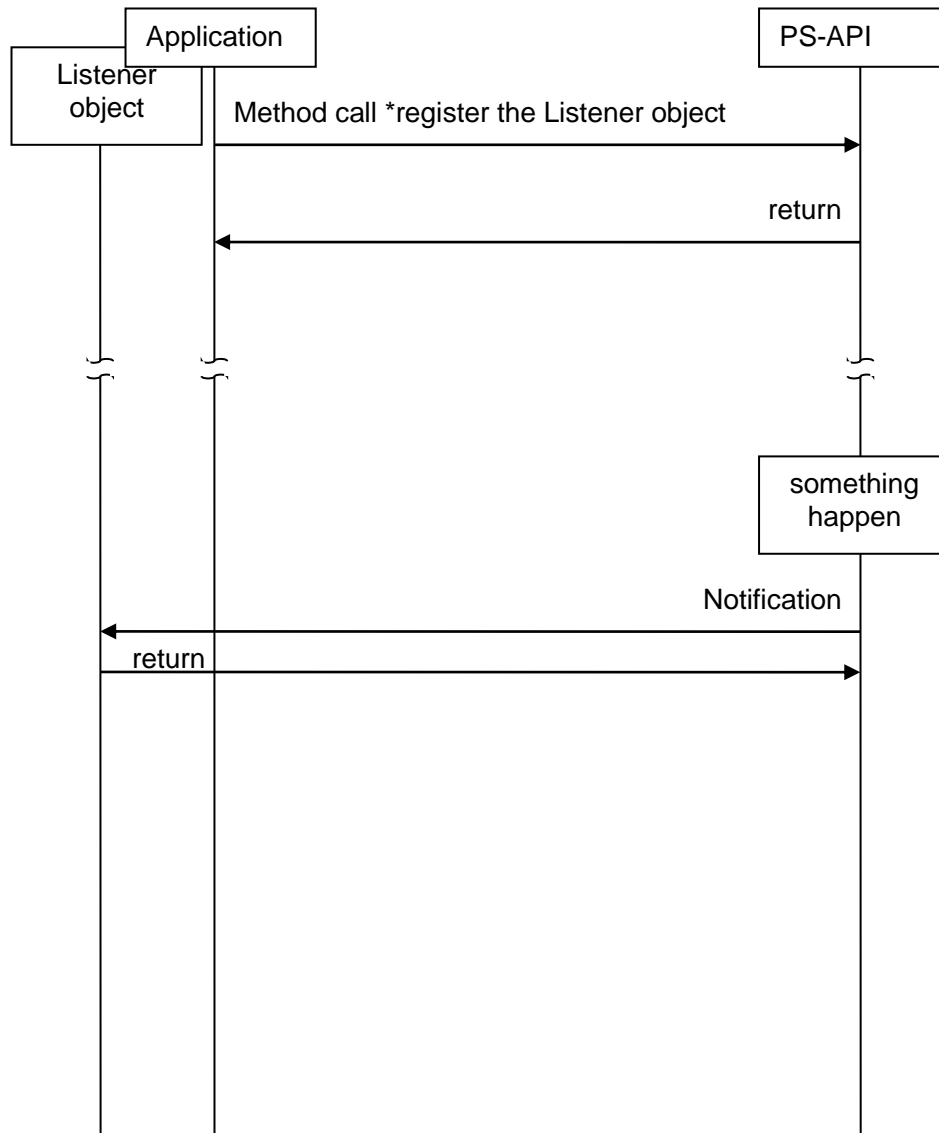
**Figure 4-12 Asynchronous Method Call**



#### 4.6.3. Receiving a notification by using Listener

Application can receive the notification by registering Listener object to PS-API. Listener object is an instantiated Listener class that Listener interface class prepared by PS-API is implemented.

\* On the application side, don't destruct the listener object that is registering



**Figure 4-13 Receiving a Notification by Using Listener**

#### 4.7. About 360-degree Network Camera

When using 360-degree network camera, the behavior depends on the image capture mode of the Network Camera.

##### 4.7.1. How to get the image capture mode of 360-degree network camera

The procedure of getting the image capture mode from 360-degree network camera is as follows.

- (1) Execute "GetDevCurrentInfo()" method.
- (2) Execute "GetInfoString()" method with "Fisheye\_ImageMode" as keyword.

[SF438 series, SW458 series, SF448 series]

Image capture mode	Value
3M Fisheye mode	3m_fisheye
1.3M Fisheye mode	1.3m_fisheye
1.3M Single PTZ mode	SinglePTZ
1.3M Quad PTZ mode	QuadPTZ
2M Panorama mode	Panorama
1M Panorama mode	1m_panorama
2M Double panorama mode	wpanorama
1M Double panorama mode	1m_wpanorama
Panorama and Single PTZ	panorama_SinglePTZ
Panorama and Quad PTZ	panorama_QuadPTZ
Double panorama and Single PTZ	wpanorama_SinglePTZ
Double panorama and Quad PTZ	wpanorama_QuadPTZ
2M Double panorama and Fisheye	wpanorama_1.3m_fisheye
1M Double panorama and Fisheye	1m_wpanorama_1.3m_fisheye
Fisheye and Quad PTZ	1.3m_fisheye_QuadPTZ
2M Panorama and Fisheye	panorama_1.3m_fisheye
1M Panorama and Fisheye	1m_panorama_1.3m_fisheye
VGA 4 stream mode	4stream

[SFV481 series]

Image capture mode	Value
9M Fisheye mode	9m_fisheye
4M Fisheye mode	4m_fisheye
Single PTZ mode	SinglePTZ
Quad PTZ mode	QuadPTZ
Panorama mode	panorama
Double panorama mode	wpanorama
8MFisheye and DoublePanorama	8m_fisheye_wpanorama
4MFisheye and DoublePanorama	4m_fisheye_wpanorama
8MFisheye and Quad PTZ	8m_fisheye_QuadPTZ
4MFisheye and QuadPTZ	4m_fisheye_QuadPTZ
8MFisheye and Panorama	8m_fisheye_panorama
4MFisheye and Panorama	4m_fisheye_panorama
4 stream mode	4stream

[S4550 series, S4551 series, S4556 series]

Image capture mode	Value
Fisheye mode	5m fisheye
Single PTZ mode	5m SinglePTZ
Quad PTZ mode	5m QuadPTZ
Panorama mode	5m panorama
Double panorama mode	5m wpanorama
Fisheye and Double panorama mode	5m fisheye wpanorama
Fisheye and QuadPTZ mode	5m fisheye QuadPTZ
Fisheye and Panorama mode	5m fisheye panorama
4 stream mode	4stream

[X4571 series, X4573 series, S4576 series]

Image capture mode	Value
Fisheye mode	9m fisheye
Single PTZ mode	SinglePTZ
Quad PTZ mode	QuadPTZ
Panorama mode	panorama
Double panorama mode	wpanorama
Fisheye and Double panorama mode	9m fisheye wpanorama
Fisheye and QuadPTZ mode	9m fisheye QuadPTZ
Fisheye and Panorama mode	9m fisheye panorama
4 stream mode	4stream

4.7.2. The explanation if image capture mode of 360-degree network camera  
The explanation of the image capture mode and the remarks are as follows.

[SF438 series, SW458 series, SF448 series]

Image capture mode	Remarks
3M Fisheye mode 1.3M Fisheye mode	<ul style="list-style-type: none"> <li>- Channel parameter can be set "1 to 4".</li> <li>- PlayLive function works as channel 1 regardless of the specified channel value.</li> </ul>
1.3M Single PTZ mode	<ul style="list-style-type: none"> <li>- Channel parameter can be set "1 to 4".</li> <li>- PlayLive function works as channel 1 regardless of the specified channel value.</li> <li>- CameraControl/CameraCentering/CameraOperation works as channel 1 regardless of the specified channel value.</li> </ul>
1.3M Quad PTZ mode	<ul style="list-style-type: none"> <li>- Channel parameter can be set "1 to 4".</li> <li>- PlayLive function works as channel 1 regardless of the specified channel value.</li> <li>- CameraCentering/CameraOperation works as channel 1 regardless of the specified channel value.</li> <li>- CameraControl works with channel parameter (1 to 4) to do Pan/Tilt/Zoom of each area.</li> <li>- When specify 1 or after number as the preset position, the screen of area 1 moves to the specified position.</li> <li>- When specify 0 (home position) as the preset position, the all screen move to the home position.</li> </ul>
2M Panorama mode 1M Panorama mode	<ul style="list-style-type: none"> <li>- Channel parameter can be set "1 to 4".</li> <li>- PlayLive function works as channel 1 regardless of the specified channel value.</li> <li>- CameraControl/CameraCentering/CameraOperation is not supported with panorama mode.</li> <li>- If you call CamSnaoShot method with fisheye mode, a fisheye snapshot image cannot be gotten. (A panorama image will be displayed.)</li> </ul>
2M Double panorama mode 1M Double panorama mode	<ul style="list-style-type: none"> <li>- Channel parameter can be set "1 to 4".</li> <li>- PlayLive function works as channel 1 regardless of the specified channel value.</li> <li>- CameraControl/CameraCentering/CameraOperation is not supported with Double panorama mode.</li> <li>- If you call CamSnaoShot method with fisheye mode, a fisheye snapshot image cannot be gotten. (A panorama image will be displayed.)</li> </ul>

Image capture mode	Remarks
2 stream mode  Panorama and Single PTZ Panorama and Quad PTZ	<ul style="list-style-type: none"> <li>- Channel parameter can be set "1 to 4".</li> <li>- PlayLive function works as channel 1 regardless of the specified channel value.</li> <li>- Stream 2 (Single PTZ or Quad PTZ) is not supported.</li> <li>- CameraControl/CameraCentering/CameraOperation is not supported with panorama mode.</li> </ul>
2 stream mode  Double panorama and Single PTZ Double panorama and Quad PTZ	<ul style="list-style-type: none"> <li>- Channel parameter can be set "1 to 4".</li> <li>- PlayLive function works as channel 1 regardless of the specified channel value.</li> <li>- Stream 2 (Single PTZ or Quad PTZ) is not supported.</li> <li>- CameraControl/CameraCentering/CameraOperation is not supported with Double panorama mode.</li> </ul>
2 stream mode  2M Double panorama and Fisheye 1M Double panorama and Fisheye	<ul style="list-style-type: none"> <li>- Channel parameter can be set "1 to 4".</li> <li>- PlayLive function works as channel 1 regardless of the specified channel value.</li> <li>- Stream 2 (Fisheye) is not supported.</li> <li>- CameraControl/CameraCentering/CameraOperation is not supported with Double panorama mode</li> </ul>
2 stream mode  Fisheye and Quad PTZ	<ul style="list-style-type: none"> <li>- Channel parameter can be set "1 to 4".</li> <li>- PlayLive function works as channel 1 regardless of the specified channel value.</li> <li>- Stream 2 (Quad PTZ) is not supported.</li> </ul>
2 stream mode  2M Panorama and Fisheye 1M Panorama and Fisheye	<ul style="list-style-type: none"> <li>- Channel parameter can be set "1 to 4".</li> <li>- PlayLive function works as channel 1 regardless of the specified channel value.</li> <li>- Stream 2 (Fisheye) is not supported.</li> <li>- CameraControl/CameraCentering/CameraOperation is not supported with panorama mode</li> </ul>
VGA 4 stream mode	<ul style="list-style-type: none"> <li>- Channel parameter can be set "1 to 4".</li> <li>- Each channel (1 to 4) can be displayed with PlayLive function.</li> <li>- There is no 2<sup>nd</sup> stream.</li> <li>- CameraControl/CameraCentering/CameraOperation is not supported with panorama mode.</li> </ul>

[SFV481 series]

Image capture mode	Remarks
9M Fisheye mode 4M Fisheye mode	<ul style="list-style-type: none"> <li>- Channel parameter can be set "1 to 4".</li> <li>- PlayLive function works as channel 1 regardless of the specified channel value.</li> </ul>
Single PTZ mode	<ul style="list-style-type: none"> <li>- Channel parameter can be set "1 to 4".</li> <li>- PlayLive function works as channel 1 regardless of the specified channel value.</li> <li>- CameraControl/CameraCentering/CameraOperation works as channel 1 regardless of the specified channel value.</li> </ul>
Quad PTZ mode	<ul style="list-style-type: none"> <li>- Channel parameter can be set "1 to 4".</li> <li>- PlayLive function works as channel 1 regardless of the specified channel value.</li> <li>- CameraOperation works as channel 1 regardless of the specified channel value.</li> <li>- CameraCentering works with channel parameter (1 to 4) to do centering of each area.</li> <li>- CameraControl works with channel parameter (1 to 4) to do Pan/Tilt/Zoom of each area.</li> <li>- When specify "0 to 4" as the preset position, the screen of area 1 moves to the position1, the screen of area 2 moves to the position2. the screen of area 3 moves to the position3, the screen of area 4 moves to the position4. When specify "5 to 8" as the preset position, the screen of area 1 moves to the position5, the screen of area 2 moves to the position6. the screen of area 3 moves to the position7, the screen of area 4 moves to the position8. the same is true as for "9-12","13-16"</li> </ul>
Panorama mode	<ul style="list-style-type: none"> <li>- Channel parameter can be set "1 to 4".</li> <li>- PlayLive function works as channel 1 regardless of the specified channel value.</li> <li>- CameraControl/CameraCentering/CameraOperation is not supported with panorama mode.</li> <li>- If you call CamSnaoShot method with fisheye mode, a fisheye snapshot image cannot be gotten. (A panorama image will be displayed.)</li> </ul>
Double panorama mode	<ul style="list-style-type: none"> <li>- Channel parameter can be set "1 to 4".</li> <li>- PlayLive function works as channel 1 regardless of the specified channel value.</li> <li>- CameraControl/CameraCentering/CameraOperation is not supported with Double panorama mode.</li> <li>- If you call CamSnaoShot method with fisheye mode, a fisheye snapshot image cannot be gotten. (A panorama image will be displayed.)</li> </ul>

Image capture mode	Remarks
2 stream mode  8M Fisheye and Double panorama 4M Fisheye and Double panorama	<ul style="list-style-type: none"> <li>- Channel parameter can be set "1 to 4".</li> <li>- PlayLive function works as channel 1 regardless of the specified channel value.</li> <li>- Stream 2 (Double panorama) is not supported.</li> <li>- CameraControl/CameraCentering/CameraOperation is not supported with Fisheye mode</li> </ul>
2 stream mode  8M Fisheye and Quad PTZ 4M Fisheye and Quad PTZ	<ul style="list-style-type: none"> <li>- Channel parameter can be set "1 to 4".</li> <li>- PlayLive function works as channel 1 regardless of the specified channel value.</li> <li>- Stream 2 (Quad PTZ) is not supported.</li> <li>- CameraControl/CameraCentering/CameraOperation is not supported with Fisheye mode.</li> </ul>
2 stream mode  8M Fisheye and panorama 4M Fisheye and panorama	<ul style="list-style-type: none"> <li>- Channel parameter can be set "1 to 4".</li> <li>- PlayLive function works as channel 1 regardless of the specified channel value.</li> <li>- Stream 2 (panorama) is not supported.</li> <li>- CameraControl/CameraCentering/CameraOperation is not supported with Fisheye mode</li> </ul>
4 stream mode	<ul style="list-style-type: none"> <li>- Channel parameter can be set "1 to 4".</li> <li>- Each channel (1 to 4) can be displayed with PlayLive function.</li> <li>- Stream 2 is not supported.</li> <li>- CameraCentering/CameraOperation works as channel 1 regardless of the specified channel value.</li> <li>- CameraControl works with channel parameter (1 to 4) to do Pan/Tilt/Zoom of each area.</li> <li>- When specify "0 to 4" as the preset position, the screen of area 1 moves to the position1, the screen of area 2 moves to the position2. the screen of area 3 moves to the position3, the screen of area 4 moves to the position4. When specify "5 to 8" as the preset position, the screen of area 1 moves to the position5, the screen of area 2 moves to the position6. the screen of area 3 moves to the position7, the screen of area 4 moves to the position8. the same is true as for "9-12", "13-16"</li> </ul>

[S4550 series, S4551 series, X4571 series, X4573 series, S4576 series, S4556 series]

Image capture mode	Remarks
Fisheye mode	<ul style="list-style-type: none"> <li>- Channel parameter can be set "1 to 4".</li> <li>- PlayLive function works as channel 1 regardless of the specified channel value.</li> </ul>
Single PTZ mode	<ul style="list-style-type: none"> <li>- Channel parameter can be set "1 to 4".</li> <li>- PlayLive function works as channel 1 regardless of the specified channel value.</li> <li>- CameraControl/CameraCentering/CameraOperation works as channel 1 regardless of the specified channel value.</li> </ul>
Quad PTZ mode	<ul style="list-style-type: none"> <li>- Channel parameter can be set "1 to 4".</li> <li>- PlayLive function works as channel 1 regardless of the specified channel value.</li> <li>- CameraOperation works as channel 1 regardless of the specified channel value.</li> <li>- CameraCentering/CameraControl works with channel parameter (1 to 4) to do Pan/Tilt/Zoom of each area.</li> <li>- When specify "0 to 4" as the preset position, the screen of area 1 moves to the position1, the screen of area 2 moves to the position2. the screen of area 3 moves to the position3, the screen of area 4 moves to the position4. When specify "5 to 8" as the preset position, the screen of area 1 moves to the position5, the screen of area 2 moves to the position6. the screen of area 3 moves to the position7, the screen of area 4 moves to the position8. the same is true as for "9-12", "13-16"</li> </ul>
Panorama mode	<ul style="list-style-type: none"> <li>- Channel parameter can be set "1 to 4".</li> <li>- PlayLive function works as channel 1 regardless of the specified channel value.</li> <li>- CameraControl/CameraCentering/CameraOperation is not supported with panorama mode.</li> <li>- If you call CamSnaoShot method with fisheye mode, a fisheye snapshot image cannot be gotten. (A panorama image will be displayed.)</li> </ul>
Double panorama mode	<ul style="list-style-type: none"> <li>- Channel parameter can be set "1 to 4".</li> <li>- PlayLive function works as channel 1 regardless of the specified channel value.</li> <li>- CameraControl/CameraCentering/CameraOperation is not supported with Double panorama mode.</li> <li>- If you call CamSnaoShot method with fisheye mode, a fisheye snapshot image cannot be gotten. (A panorama image will be displayed.)</li> </ul>



Image capture mode	Remarks
2 stream mode  Fisheye and Double panorama	<ul style="list-style-type: none"> <li>- Channel parameter can be set "1 to 4".</li> <li>- PlayLive function works as channel 1 regardless of the specified channel value.</li> <li>- Stream 2 (Double panorama) is not supported.</li> <li>- CameraControl/CameraCentering/CameraOperation is not supported with Fisheye mode</li> </ul>
2 stream mode  Fisheye and Quad PTZ	<ul style="list-style-type: none"> <li>- Channel parameter can be set "1 to 4".</li> <li>- PlayLive function works as channel 1 regardless of the specified channel value.</li> <li>- Stream 2 (Quad PTZ) is not supported.</li> <li>- CameraControl/CameraCentering/CameraOperation is not supported with Fisheye mode.</li> </ul>
2 stream mode  Fisheye and panorama	<ul style="list-style-type: none"> <li>- Channel parameter can be set "1 to 4".</li> <li>- PlayLive function works as channel 1 regardless of the specified channel value.</li> <li>- Stream 2 (panorama) is not supported.</li> <li>- CameraControl/CameraCentering/CameraOperation is not supported with Fisheye mode</li> </ul>
4 stream mode	<ul style="list-style-type: none"> <li>- Channel parameter can be set "1 to 4".</li> <li>- Each channel (1 to 4) can be displayed with PlayLive function.</li> <li>- Stream 2 is not supported.</li> <li>- CameraCentering/CameraOperation works as channel 1 regardless of the specified channel value.</li> <li>- CameraControl works with channel parameter (1 to 4) to do Pan/Tilt/Zoom of each area.</li> <li>- When specify "0 to 4" as the preset position, the screen of area 1 moves to the position1, the screen of area 2 moves to the position2. the screen of area 3 moves to the position3, the screen of area 4 moves to the position4. When specify "5 to 8" as the preset position, the screen of area 1 moves to the position5, the screen of area 2 moves to the position6. the screen of area 3 moves to the position7, the screen of area 4 moves to the position8. the same is true as for "9-12", "13-16"</li> </ul>

#### 4.8. Restriction by device settings

##### 4.8.1. Smart Cording

There are following restrictions when "Smart Coding mode" is set to "ON", "ON(Low)", "ON(Mid)", "ON(High)", "ON(Advanced)" in the "Transmission priority" setting of the camera.

###### (1) Live

[In the case of "ON" or "ON(Low)"]

- It may take 8 seconds at maximum for images to be displayed.

[In the case of "ON(Mid)" or "ON(High)"]

- It may take 16 seconds at maximum for images to be displayed.

###### (2) Network playback

[In the case of "ON" or "ON(Low)"]

- For playback, jump, and skip, it may take 8 seconds at maximum for images to be displayed.
- The speed rate for the fast playback may be increased compared to the standard setting.  
(For example, images will be forwarded 32 seconds at maximum in Step 2.)
- The images may be seen stopped for 8 seconds at maximum in the frame playback or reverse frame playback.
- In playback with the alarm time and date, alarm images may not be displayed during playback.

[In the case of "ON(Mid)" or "ON(High)"]

- For playback, jump, and skip, it may take 16 seconds at maximum for images to be displayed.
- The speed rate for the fast playback may be increased compared to the standard setting.  
(For example, images will be forwarded 64 seconds at maximum in Step 2.)
- The images may be seen stopped for 16 seconds at maximum in the frame playback or reverse frame playback.
- In playback with the alarm time and date, alarm images may not be displayed during playback.

[In the case of "ON(Frame rate control)"]

- It does not supported to the stream delivered with this setting.

## 5. Details of DLL Class and Method

This chapter describes the detail specification of this library.

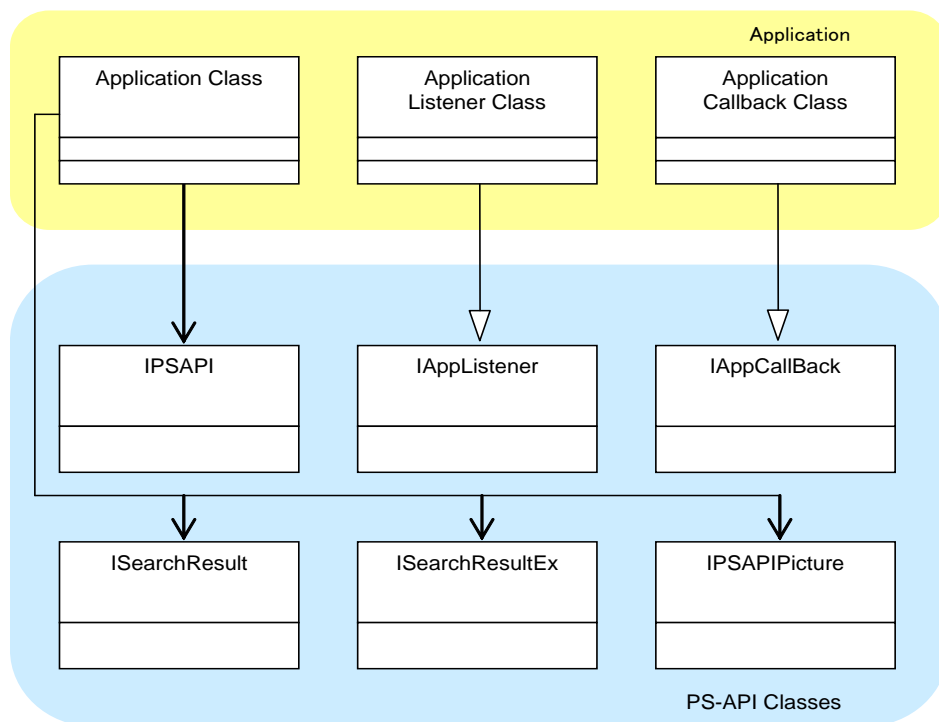
### 5.1. Class

#### 5.1.1. Class Definition

**Table 5-1 Class Definition**

No.	Class Name	Overview
1	IPSAPI	It is the interface class for using PS-API functions from an application. Control a device by calling the method of IPSAPI.
2	IAppListener	It is the interface class for receiving the notification from PS-API. The application can implement the process for the notification by inheriting this class and by implementing the notification method. Also the application can receive the notification by registering to PS-API the instance that is implemented by the application.
3	IAppCallBack	It is the interface class for receiving the result when using the asynchronous method of PS-API. The application can implement the process for the result by inheriting this class and by implementing the notification method. Also the application specifies the instance that is implemented by the application when calling the asynchronous method.
4	ISearchResult	It is the class to store the search result by Search method.
5	ISearchResultEx	It is the class to store the search result by SearchEx method.
6	IPSAPIPicture	It is the class to store the data when getting image with SnapShot method.

#### 5.1.2. Class Diagram



**Figure 5-1 Class Diagram**

#### 5.1.2.1. Global Function

```
IPSAPI* GetIPSAPI( );  
void DeleteIPSAPI(IPSAPI* ipsapi);
```

#### 5.1.2.2. IPSAPI

```
long Open( );  
long Connect(long uid);  
void Close( );  
void Disconnect( );  
long ClearWaitingFunc();  
long GetWaitingFuncCount();  
long GetLoginStatus();  
long GetUIDInfo(SUID_INFO* uidInfo);  
long GetSIDInfo(SSID_INFO* sidInfo);  
long SetDeviceType(long type);  
long GetDeviceType();  
long SetIPAddr(char* ipaddr);  
void GetIPAddr(char* ipaddr, long size);  
long SetHttpPort(long port);  
long GetHttpPort();  
long SetHttpTimeout(long time);  
long GetHttpTimeout();  
long SetProxyName(char* name);  
void GetProxyName(char* name, long size);  
long SetProxyPort(long port);  
long GetProxyPort();  
long SetAccessType(long type);  
long GetAccessType();  
long GetUID();  
long SetUserName(char* userName);  
void GetUserName(char* userName, long size);  
long SetPassword(char* password);  
void GetPassword(char* password, long size);  
long SetErrListener(IAppListener* pReceiver);  
long GetDevStatus(long channel);  
long RecCtrl(long channel, long command, long& state, long mode,  
    IAppCallBack* pSender);  
long Search(long channel, char* startDate, char* endDate, long type,  
    ISearchResult* searchResult, long mode, IAppCallBack* pSender);  
long SearchEx(long channel, char* startDate, char* endDate, long type,  
    ISearchResultEx* searchResult, long mode, IAppCallBack* pSender);  
long VMDSearchEx(long channel, char* startDate, char* endDate, long mask,  
    long aSensitivity, long axTopLeft, long ayTopLeft, long axBottomRight,  
    long ayBottomRight, long bSensitivity, long bxTopLeft, long byTopLeft,  
    long bxBottomRight, long byBottomRight, long cSensitivity, long cxTopLeft,  
    long cyTopLeft, long cxBottomRight, long cyBottomRight, long dSensitivity,  
    long dxTopLeft, long dyTopLeft, long dxBottomRight, long dyBottomRight,  
    long imageWidth, long imageHeight, ISearchResultEx* searchResultEx,  
    long mode, IAppCallBack pSender);  
long SearchCancel();  
long GetDeviceLog(long type, SLOGRESULT* resultlist, long maxCount, long& resultCount);  
void GetDevTimezone(STIMEZONE_INFO* timezone);  
long GetDevCurrentInfo();  
long GetInfoString(char* key, char* value, long size);
```

```

long    SetCameraTime(char* timeDate, long isDst);
long    GetStatisticsData(char* startTimeDate, char* endTimeDate, char* fileName);
long    SetUIDPriority(long command);
void    GetDevModel(char* model, long size);
long    SetSearchMultiChMask(char* mask);
long    GetSearchMultiChMask(char* mask, long size);
long    SetDevListener(IAppListener* pReceiver);
long    SetRecListener(IAppListener* pReceiver);
void    GetFrameTime(SVIDEO_FRAME_TIME* frametime );
long    PlayLive(long channel, long& status, long mode, IAppCallBack* pSender);
long    Play(long channel, char* timeData, long& status, long mode, IAppCallBack* pSender);
long    PlayFile(char* fileName, long& status, long mode, IAppCallBack* pSender);
long    PlayControl(long command, long speed, long& status, long mode,
                    IAppCallBack* pSender);
long    PlayControlByTime(char* timeData, long isDst, long& status, long mode,
                    IAppCallBack* pSender);
long    GetPlayStatus( );
long    GetPlaySpeed( );
long    GetFrameRate( );
long    GetPicturePosition(long& xTopLeft, long& yTopLeft,
                           long& xBottomRight, long& yBottomRight);
long    GetImageResolution(long& imagWidth, long& imageHeight);
long    SetVideoWindow(hWnd handle);
long    DecodeImage(unsigned char* pBuffer, long size);
void    ClearImage();
long    Snapshot(IPSAPIPicture* pContainer)
long    TitleOperation(long id, long command, char* text, long xPosition, long yPosition,
                      long align, char* font, long fontSize, long foreColor, long borderColor,
                      long style);
long    TitleOperationEx(long id, long command, char* text, long xPosition, long yPosition,
                        long align, char* font, long fontSize, long foreColor, long borderColor,
                        long style, long transmissivity);
long    GetTitle(long id, char* pBuffer, long size);
long    BoxOperation(long id, long command, long color, long size,
                    long xTopLeft, long yTopLeft, long xBottomRight, long yBottomRight);
long    BoxOperationEx(long id, long command, long color, long size,
                      long xTopLeft, long yTopLeft, long xBottomRight, long yBottomRight,
                      long transmissivity);
long    BitmapOperationEx(long id, long command, char* filename,
                          long xPosition, long yPosition, long maskColor, long transmissivity);
long    DigitalZoomMove(long xPosition, long yPosition);
long    GetDigitalZoomPosition(long& xPosition, long& yPosition);
long    MultiSyncPause();
long    MultiSyncTime(char* syncTime, long isDst);
long    CamSnapShot(long channel, long imageMode);
long    SetCroppingRect (long id, long ltX, long ltY, long rbX, long rbY);
long    GetCroppingRect (long id, long& ltX, long& ltY, long& rbX, long& rbY);
long    SetCroppingDrawRect (long id, long ltX, long ltY, long rbX, long rbY);
long    GetCroppingDrawRect (long id, long& ltX, long& ltY, long& rbX, long& rbY);
long    SetCroppingDrawEnabled (long id, long mode);
long    GetCroppingDrawEnabled (long id, long& mode);
long    SetCroppingMarker(long id, long mode, long ltX, long ltY, long rbX, long rbY,
                          long lineSize, long lineColor, long ellipseSize, long ellipseColor);
long    SetCroppingMarker(long id, long& mode, long& ltX, long& ltY, long& rbX, long& rbY,
                          long& lineSize, long& lineColor, long& ellipseSize, long& ellipseColor);
long    HttpMP4Download(long channel, long command, char* startTimeDate,
                       char* endTimeDate, long audioMode, char* filename);

```

```

long    HttpDownload (long  channel, long  command, char*  startTimeDate, long  isDstSt,
                    char*  endTimeDate, long  isDstEt, long  dataType, char*  filename );
long    GetMP4DownloadStatus();
long    GetMP4DownloadTransrate();
long    SetMPEG4Port(long  port);
long    GetMPEG4Port();
long    SetH264Port(long  port);
long    GetH264Port();
long    SetRtpPortMode(long  mode);
long    GetRtpPortMode();
long    SetRtpPortRange(long  range);
long    GetRtpPortRange();
long    SetMulticastAutoConf(long  auotconf);
long    GetMulticastAutoConf();
long    SetMulticastAddr(char*  ipAddr);
void    GetMulticastAddr(char*  ipAddr, long  size);
long    SetMPEG4Resolution(long  resolution);
long    GetMPEG4Resolution();
long    SetH264Resolution(long  resolution);
long    GetH264Resolution();
long    SetJPEGResolution(long  resolution);
long    GetJPEGResolution();
long    SetStreamFormat(long  format);
long    GetStreamFormat();
long    SetFilePassword(char*  password);
long    GetFilePassword();
long    SetImageHeight(long  height);
long    GetImageHeight();
long    SetImageWidth(long  width);
long    GetImageWidth();
long    SetMulticastAutoConf(long  autoconf);
long    GetMulticastAutoConf();
long    SetStreamNumber(long  no);
long    GetStreamNumber();
long    SetInternetMode(long  mode);
long    GetInternetMode();
long    SetFastPlayMode(long  mode);
long    GetFastPlayMode();
long    SetTransFrameRate(long  transFrameRate);
long    GetTransFrameRate();
long    SetBackColor(long  color);
long    GetBackColor();
long    SetPictureFitMode(long  mode);
long    GetPictureFitMode();
long    SetDigitalZoom(long  zoom);
long    GetDigitalZoom();
long    SetDigitalZoomMode(long  mode);
long    GetDigitalZoomMode();
long    SetSkipRecordGap(long  mode);
long    GetSkipRecordGap();
long    SetMultiScreenChannel(char*  channel);
long    GetMultiScreenChannel(char*  channel, long  size)
long    SetSIDMode(long  mode);
long    GetSIDMode();
long    SetDecResolutionMode(long  mode);
long    GetDecResolutionMode();
long    SetCroppingEnabled(long  mode);

```

```

long    GetCroppingEnabled();
long    SetDecResolutionMode (long  mode);
long    GetDecResolutionMode ();
long    SetRcvAudioDec (long  format);
long    GetRcvAudioDec ();
long    SetPlayListener(IAppListener* pReceiver);
long    SetRecordListener(IAppListener* pReceiver);
long    SetImageListener(IAppListener* pReceiver);
long    SetMP4DownloadListener (IAppListener* pReceiver);
long    AudioSend(long  command);
long    GetAudioSendStatus();
long    SetAudioRcvEnable(long  mode);
long    GetAudioRcvEnable();
long    SetAudioRcvVolume(long  volume);
long    GetAudioRcvVolume();
long    SetAudioRcvMute(long  mute);
long    GetAudioRcvMute();
long    SetAudioSendVolume(long  volume);
long    GetAudioSendVolume();
long    SetAudioSendMute(long  mute);
long    GetAudioSendMute();
long    CameraControl(long  channel, long  pan, long  tilt, long  zoom, long  focus, long  iris);
long    SetCameraPosition(long  channel, long  pan, long  tilt, long  zoom, long  focus,);
long    GetCameraPosition(long  channel, long&  pan, long&  tilt, long&  zoom, long&  focus);
long    CameraOperation(long  channel, long  command, long  data, long&  status, long  mode,
        IAppCallBack* pSender);
long    GetCamOpStatus(long  channel);
long    CameraCentering(long  channel, long  xPosition, long  yPosition, long  imageWidth,
        long  imageHeight);
long    CameraAuxControl(long  channel, long  alarmTrmNo, long  command);
long    GetCameraAuxStatus(long  channel, long  alarmTrmNo);
long    SetCameraImageCap(long  mode, long  installation);
long    CameraWiperControl(long  channel, long  command);
long    SetOpListener(IAppListener* pReceiver);
long    AlmOperation(long  channel, long  command, long&  status, long  mode,
        IAppCallBack* pSender);
long    GetAlarmStatus( );
long    SetAlmListener(IAppListener* pReceiver);
long    FtpGet(long  channel, char*  startTimeDate, char*  endTimeDate, long  data, long  type,
        char*  filename, long&  Status,  long  mode, IAppCallBack* pSender);
long    FtpCancel();
long    FtpServerClose();
long    SetFtpPort(long  port);
long    GetFtpPort();
long    SetFtpTransMode(long  mode);
long    GetFtpTransMode();
long    GetFtpStatus();
long    GetFtpTransRate();
long    GetFtpTransByte();

```

#### 5.1.2.3. IAppListener

```
virtual void OnError(long errorCode, const char* description) {};  
virtual void OnDevStatus(long channel, long status) {};  
virtual void OnRecStatus(long channel, long status) {};  
virtual void OnPlayStatus(long channel, long status) {};  
virtual void OnRecordStatus(long recType, char* timeDate, long isDst,  
                             char* nextRecTime, long isDstNext);  
virtual void OnImage(long type, unsigned char* pBuffer, long size) {};  
virtual void OnOpStatus (long channel, long status) {};  
virtual void OnAlmStatus (long channel, long type, char* timeDate, long status) {};  
virtual void OnMP4DownloadStatus(long status, char* fileName) {};
```

#### 5.1.2.4. IAppCallBack

```
virtual void OnRecCB(long status) {};  
virtual void OnSearchCB( ) {};  
virtual void OnSearchExCB() {};  
virtual void OnPlayStatusCB(long status) {};  
virtual void OnOpStatusCB (long status) {};  
virtual void OnAlmStatusCB (long status) {};  
virtual void OnFtpStatusCB(long status) {};
```



#### 5.1.2.5. ISearchResult

```
long    GetListCount( );
long    GetResult(long lIndex, SEARCHRSLT_INFO*    stSetInfo);
long    GetAndDelete(SEARCHRSLT_INFO*    stSetInfo);
long    Clear( );
```

#### 5.1.2.6. ISearchResultEx

```
long    GetListCount( );
long    GetResult(long lIndex, SEARCHRSLT_INFO_EX*    stSetInfo);
long    GetAndDelete(SEARCHRSLT_INFO_EX*    stSetInfo);
long    Clear( );
```

#### 5.1.2.7. IPSAPIPicture

```
long    GetBitmapSize();
long    GetBitmapImage(char* pBuffer, long size);
long    SaveBitmapImage(char* fileName);
long    GetJpegSize();
long    GetJpegImage(char* pBuffer, long size);
long    SaveJpegImage(char* fileName);
```

## 5.2. Global Function

### 5.2.1. GetIPSAPI

<b>Class</b>	-
--------------	---

<b>Function</b>	<b>GetIPSAPI</b>
-----------------	------------------

IPSAPI*	GetIPSAPI( );
---------	---------------

#### Description

Create the IPSAPI instance.

#### Argument

None

#### Return value

IPSAPI*	Pointer of created IPSAPI instance
---------	------------------------------------

#### Error

## Note

---

## Sequence

---

### 6.1 PlayLive

## Sample program code

---

[Visual C++ 2012] ..¥Sample Program¥PS-API¥Visual C++ 2012¥001\_PlayLive

## Reference

---

### 5.2.2. DeleteIPSAPI

**Class** -

---

**Function** DeleteIPSAPI

---

void DeleteIPSAPI(IPSAPI\* ipsapi)

---

**Description**

Delete the IPSAPI instance.

---

**Argument**

ipsapi	Pointer of IPSAPI instance	Specify the pointer of the IPSAPI instance.
--------	----------------------------	---

---

**Return value**

None

---

**Error**

## Note

---

## Sequence

---

### 6.1 PlayLive

## Sample program code

---

[Visual C++ 2012] ..¥Sample Program¥PS-API¥Visual C++ 2012¥001\_PlayLive

## Reference

---

### 5.3. PS Builder Group

#### 5.3.1. Method

##### 5.3.1.1. Open

<b>Class</b>	<b>IPSAPI</b>
<b>Method</b>	<b>Open</b>
long	Open( );

#### **Description**

Log in to the device, and get UID.

#### **Argument**

None

#### **Return value**

Positive value	Success in connecting to the device. (HD300, NWDR, HD600/700, NX Series) The return value is UID.
0	Success in connecting to the device.(network camera, encoder)
Negative value	Failure to connect to the device

#### **Error**

Get the error information by OnError event.

## Note

---

## Sequence

---

### 6.1 PlayLive

## Sample program code

---

[Visual C++ 2012] ..¥Sample Program¥PS-API¥Visual C++ 2012¥001\_PlayLive

## Reference

---

#### 5.3.1.2. Connect

**Class**                      **IPSAPI**

**Method**                    **Connect**

```
long      Connect(  
                 long uid  
                 );
```

---

#### **Description**

Connect to the device with the specified UID which other instance got by the login.

---

#### **Argument**

uid	Positive value	UID which other instance got.
-----	----------------	-------------------------------

---

#### **Return value**

0	Success in connecting to the device.
Except 0	Error code

---

#### **Error**

Error is defined by the return value.  
Get the error information by OnError event.



## Note

---

## Sequence

---

### 6.9 MultiPlayLive

## Sample program code

---

[Visual C++ 2012] ..¥Sample Program¥PS-API¥Visual C++ 2012¥009\_MultiPlayLive

## Reference

---

#### 5.3.1.3. Close

<b>Class</b>	<b>IPSAPI</b>
--------------	---------------

<b>Method</b>	<b>Close</b>
---------------	--------------

void	Close( );
------	-----------

#### Description

Stop the HTTP connection, and log out from the device.  
UID is annulled.

#### Argument

None

#### Return value

None

#### Error

Get the error information by OnError event.

## **Note**

---

## **Sequence**

---

### **6.1 PlayLive**

## **Sample program code**

---

[Visual C++ 2012] ..¥Sample Program¥PS-API¥Visual C++ 2012¥001\_PlayLive

## **Reference**

---

#### 5.3.1.4. Disconnect

<b>Class</b>	<b>IPSAPI</b>
--------------	---------------

<b>Method</b>	<b>Disconnect</b>
---------------	-------------------

void	Disconnect( );
------	----------------

---

#### Description

Stop the HTTP connection.  
UID is NOT annulled.

---

#### Argument

None

---

#### Return value

None

---

#### Error

Get the error information by OnError event.

## Note

---

## Sequence

---

### 6.9 MultiPlayLive

## Sample program code

---

[Visual C++ 2012] ..¥Sample Program¥PS-API¥Visual C++ 2012¥009\_MultiPlayLive

## Reference

---

#### 5.3.1.5. ClearWaitingFunc

<b>Class</b>	<b>IPSAPI</b>
--------------	---------------

<b>Method</b>	<b>ClearWaitingFunc</b>
---------------	-------------------------

long	ClearWaitingFunc ( );
------	-----------------------

#### Description

Cancel the waiting function of async type method.

#### Argument

None

#### Return value

0	Success
---	---------

Except 0	Error code
----------	------------

#### Error

Error is defined by the return value.  
Get the error information by OnError event.

### Note

---

The function in processing cannot be deleted.

The callback of the deleted async mode method is not notified.

If you clear the waiting queue by using this method, the callback of the processing async mode method is notified

Please don't destruct the callback object before the callback object is called.

### Sequence

---

### Sample program code

---

### Reference

---

#### 5.3.1.6. GetWaitingFuncCount

**Class**                **IPSAPI**

---

**Method**             **GetWaitingFuncCount**

---

long      GetWaitingFuncCount ( );

---

#### **Description**

Get the waiting function of async type method count.

---

#### **Argument**

None

---

#### **Return value**

0 and over      The waiting method count.

Negative  
value            Error code

---

#### **Error**

Error is defined by the return value.  
Get the error information by OnError event.



**Note**

---

**Sequence**

---

**Sample program code**

---

**Reference**

---

### 5.3.1.7. GetLoginStatus

**Class**                **IPSAPI**

**Method**            **GetLoginStatus**

long      GetLoginStatus( );

#### **Description**

Get login result. If login failure, the reason of failure returns.

#### **Argument**

None

#### **Return value**

-1	Open/Connect not executed.
0	Open/Connect success
1	(Recorder response) User excess
2	(Recorder response) Authentication error
3	(Recorder response) Network impossible
4	(Recorder response) Host attestation
5	(Recorder response) Config state
6	(Recorder response) In measurement
7	(Recorder response) Config user attestation
8	(Recorder response) Restarting state
9	(Recorder response) Sleep state
10	(Recorder response) Play only state
11	(Recorder response) No disk
12	(Recorder response) IP easy setup connected state
13	(Recorder response) Being configuration state
14	(Recorder response) Blackout
15	Other errors

#### **Error**

Get the error information by OnError event.

**Note**

---

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.3.1.8. GetUIDInfo

**Class**            **IPSAPI**

**Method**          **GetUIDInfo**

long      GetUIDInfo( SUID\_INFO\*   uidInfo );

---

#### **Description**

Get the maximum number of UID that recorder can issue and the number of UID which is in use.

---

#### **Argument**

uidInfo	Specify a pointer to a structure describing the UID information. Here is the definition of the structure:
---------	--

```
struct  SUID_INFO{  
    // The maximum number of UID  
    long  m_IMaxCount;  
  
    // The number of UID which is in use  
    long  m_IInUseCount;  
}
```

---

#### **Return value**

0	Success
---	---------

Except 0	Error code
----------	------------

---

#### **Error**

Error is defined by the return value.  
Get the error information by OnError event.

**Note**

Maximum number of UID depends on device model.

Model	Maximum number	Remarks
HD300	Not support.	Getting information is not supported. Maximum number of UID is 8
ND200	Not support.	Getting information is not supported. Maximum number of UID is 4
ND300	Not support.	Getting information is not supported. Maximum number of UID is 8
ND400	16	Firmware version V3.10 or later can support the getting information.
NV200	4	Firmware version V1.40 or later can support the getting information.
NV250	4	Firmware version V1.00 or later can support the getting information.
NV300	8	Firmware version V1.02 or later can support the getting information.
NX100	16	Firmware version V1.00 or later can support the getting information.
NX200	16	Firmware version V1.00 or later can support the getting information.
NX300	16	Firmware version V1.00 or later can support the getting information.
NX400	16	Firmware version V1.00 or later can support the getting information.
HD600/700	8	Firmware version V2.20 or later can support the getting information.

**Sequence****Sample program code****Reference**

#### 5.3.1.9. GetSIDInfo

**Class**                **IPSAPI**

**Method**             **GetSIDInfo**

long      GetSIDInfo( SSID\_INFO\*    sidInfo );

#### Description

Get the information of StreamID support, the maximum number of StreamID that recorder can issue, and the number of StreamID which is in use.

#### Argument

sidInfo	Specify a pointer to a structure describing the UID information. Here is the definition of the structure:
---------	--

```
struct  SSID_INFO{  
    // The information of StreamID support  
    // 0 : Target device doesn't support StreamID  
    // 1 : Target device support StreamID  
    long  m_IMode;  
  
    // The maximum number of StreamID  
    long  m_IMaxCount;  
  
    // The number of StreamID which is in use  
    long  m_InUseCount;  
}
```

#### Return value

0	Success
---	---------

Except 0	Error code
----------	------------

#### Error

Error is defined by the return value.  
Get the error information by OnError event.

**Note**

Maximum number of StreamID depends on device model.

Model	Maximum number	Remarks
HD300	Not support.	StreamID mode is not supported.
ND200	Not support.	StreamID mode is not supported.
ND300	Not support.	StreamID mode is not supported.
ND400	64	Firmware version V3.10 or later can support the getting information.
NV200	16	Firmware version V1.40 or later can support the getting information.
NV250	24	Firmware version V1.00 or later can support the getting information.
NV300	32	Firmware version V1.02 or later can support the getting information.
NX100	64	Firmware version V1.00 or later can support the getting information.
NX200	64	Firmware version V1.00 or later can support the getting information.
NX300	64	Firmware version V1.00 or later can support the getting information.
NX400	64	Firmware version V1.00 or later can support the getting information.
HD600/700	64	Firmware version V2.20 or later can support the getting information.

**Sequence****Sample program code****Reference**

### 5.3.2. Property

#### 5.3.2.1. DeviceType

**Class**            **IPSAPI**

---

**Property**        **DeviceType**

---

long      SetDeviceType(  
                                 long   type  
                                 );

long      GetDeviceType();

---

#### Description

Set a device type of a target device into PS-API.

Get a device type of a target device from PS-API.

---

#### Argument for SET

type	0 : HD300	Digital Disk Recorder (HD300)
	1 : NWDR	Network Disk Recorder(except for NX Series)
	2 : Camera	Network Camera
	3 : Encoder	Network Interface Unit
	4 : HD600/700	Digital Disk Recorder (HD600/700)
	6 : NX Series	Network Disk Recorder(NX Series)

Default value is 2.

---

#### Return value for SET

0	Success
---	---------

Except 0	Error code
----------	------------

---

#### Return value for GET

Get a Device type for a target device.

---

#### Error

---



## Note

---

## Sequence

---

### 6.1 PlayLive

## Sample program code

---

[Visual C++ 2012] ..¥Sample Program¥PS-API¥Visual C++ 2012¥001\_PlayLive

## Reference

---

#### 5.3.2.2. IPAddr

<b>Class</b>	<b>IPSAPI</b>
--------------	---------------

<b>Property</b>	<b>IPAddr</b>
-----------------	---------------

long	SetIPAddr( char* ipaddr );
void	GetIPAddr( char* ipaddr, long size );

#### Description

Set IP Address or host name of a target device into PS-API.  
Get IP Address or host name of a target device from PS-API.

#### Argument for SET

ipaddr	Character strings (255bytes or less)	IP Address or host name The strings must be terminated with null character. e.g. 192.168.0.10  Default value is "192.168.0.10".
--------	---	--

#### Argument for GET

ipaddr	Character strings	Get IP Address Application needs to allocate and deallocate memory. If getting value failed, it is set to NULL.
size	Buffer size of ipaddr	Specify the buffer size for ipaddr. (Byte)

#### Return value for SET

0	Success
Except 0	Error code

#### Error

---

**Note**

When 0 is added at the top of segment like as 192.168.000.010, it is treated with octal number.  
When port number is written after : (colon) like as 192.168.0.10:8080, error occurs.

PS-API does not support DDNS name resolver.

When both IPv6 address and IPv4 address are set as host name, connect with IPv6 address.

In the case of IPv6 address, you can set interface number by using "%", like as "2001::1%1"

In the case of IPv6 address and stream format are H265 / H264, the video may not be displayed.

---

**Sequence****6.1 PlayLive**

---

**Sample program code**

[Visual C++ 2012] ..¥Sample Program¥PS-API¥Visual C++ 2012¥001\_PlayLive

---

**Reference**

---

### 5.3.2.3. HttpPort

**Class**            **IPSAPI**

**Property**        **HttpPort**

long      SetHttpPort(  
                         long   port  
                         );

long      GetHttpPort();

#### **Description**

Set HTTP server port number of a target device into PS-API.

Get HTTP server port number of a target device from PS-API.

#### **Argument for SET**

port	1 to 65535	HTTP server port.
		Default value is 80.

#### **Return value for SET**

0	Success
Except 0	Error code

#### **Return value for GET**

Get HTP server port number for a target device.

#### **Error**

## Note

---

## Sequence

---

### 6.1 PlayLive

## Sample program code

---

[Visual C++ 2012] ..¥Sample Program¥PS-API¥Visual C++ 2012¥001\_PlayLive

## Reference

---

#### 5.3.2.4. HttpTimeout

**Class**            **IPSAPI**

**Property**        **HttpTimeout**

long      SetHttpTimeout(  
                                 long   time  
                                 );

long      GetHttpTimeout();

#### **Description**

Set HTTP communication timeout value for the HTTP server into PS-API.  
Get HTTP communication timeout value for the HTTP server from PS-API.

#### **Argument for SET**

time	1 to 60 Second scale.	Http server timeout value.  Default value is 10 seconds .
------	--------------------------	---

#### **Return value for SET**

0	Success
Except 0	Error code

#### **Return value for GET**

Get HTTP timeout value for the HTTP server.

#### **Error**

**Note**

---

When PS-API fails communication because of time-out or other reason, PS-API try to communicate again. Therefore the time-out error occurs after about 2 times interval of the time-out value set in HttpTimeout.

Depends on the version of installed Internet Explorer, the timeout error occurs before 2 times interval of the time-out value set in HttpTimeout.

In case of a NX Series, when operate the Search method, SearchEx method, VMDSearchEx method, and FtpGet method, the timeout value is fixed at 60 seconds.

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.3.2.5. ProxyName

**Class**            **IPSAPI**

**Property**        **ProxyName**

long	SetProxyName( char* name );
void	GetProxyName( char* name, long size );

#### **Description**

Set Proxy name or network address into PS-API.  
Get Proxy name or network address from PS-API.

#### **Argument for SET**

name	Character strings (255bytes or less)	Proxy name or Network address The strings must be terminated with null character.  Default value is NULL.
------	---	---

#### **Argument for GET**

name	Character strings	Get Proxy name or Network address Application needs to allocate and deallocate memory. If getting value failed, it is set to NULL.
size	Buffer size of name	Specify the buffer size for name. (Byte)

#### **Return value for SET**

0	Success
Except 0	Error code

#### **Error**



**Note**

---

When 0 is added at the top of segment like as 192.168.000.010, it is treated with octal number.  
When port number is written after : (colon) like as 192.168.0.10:8080, error occurs.

PS-API does not support DDNS name resolver.

When both IPv6 address and IPv4 address are set as host name, connect with IPv6 address.

Interface number cannot be specified to ProxyName.

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.3.2.6. ProxyPort

**Class**            **IPSAPI**

**Property**        **ProxyPort**

long      SetProxyPort(  
                                long   port  
                                );

long      GetProxyPort();

#### **Description**

Set Proxy port number into PS-API.  
Get Proxy port number from PS-API.

#### **Argument for SET**

port	1 to 65535	Proxy port number
		Default value is 80.

#### **Return value for SET**

0	Success
Except 0	Error code

#### **Return value for GET**

Get Proxy port number in an application client station.  
When the function returns for GetProxyName() is null, it indicates that there is no proxy name specified and application should ignore the proxy port number setting.

#### **Error**

**Note**

---

**Sequence**

---

**Sample program code**

---

**Reference**

---

### 5.3.2.7. AccessType

**Class**            **IPSAPI**

---

**Property**        **AccessType**

---

long      SetAccessType(  
                                 long type  
                                 );

long      GetAccessType();

---

#### **Description**

Set the type of an access method into PS-API to communicate with its server.

Get the type of an access method from PS-API to communicate with its server.

---

#### **Argument for SET**

type	0 : Depends on IE setting 1 : Direct access 2 : Proxy server access	It retrieves setting from Internet Explorer. The web server must be configured to listen to the desired ports . Via proxy server access.  Default value is 0.
------	---	--

---

#### **Return value for SET**

0	Success
Except 0	Error code

---

#### **Return value for GET**

Get Access type to access the web server.

---

#### **Error**

---

**Note**

---

When ProxyName isn't specified though AccessType is set to "2", it retrieves setting from Internet Explorer.

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.3.2.8. UID

<b>Class</b>	<b>IPSAPI</b>
--------------	---------------

<b>Property</b>	<b>UID</b>
-----------------	------------

long	GetUID();
------	-----------

#### Description

Get UID that is retrieved from a target device.

#### Argument for SET

None

#### Return value for SET

None

#### Return value for GET

	Get UID.
-1	Not login
0	Logined (Login to Network Camera or Encoder)
1 to 65535	Logined UID (Login to NWDR, HD300, HD600/700, NX Series).

#### Error

**Note**

---

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.3.2.9. UserName

**Class**                      **IPSAPI**

**Property**                      **UserName**

```
long      SetUserName(
                char*    userName
            );

void      GetUserName(
                char*    username
                long    size
            );
```

#### Description

Set a user name into PS-API for logging in to a target device.  
Get a user name from PS-API for logging in to a target device.

#### Argument for SET

userName	Character strings (255bytes or less)	Login user name. The strings must be terminated with null character. If user authentication is set to off, please set UserName to "".  Default value is    NULL.
----------	---	--

#### Argument for GET

userName	Character strings	Get login user name Application needs to allocate and deallocate memory. If getting value failed, it is set to NULL.
size	Buffer size of name	Specify the buffer size for userName. (Byte)

#### Return value for SET

0	Success
Except 0	Error code

#### Error



## Note

---

## Sequence

---

### 6.1 PlayLive

## Sample program code

---

[Visual C++ 2012] ..¥Sample Program¥PS-API¥Visual C++ 2012¥001\_PlayLive

## Reference

---

### 5.3.2.10. Password

**Class**                      **IPSAPI**

**Property**                      **Password**

```
void      SetPassword(
                char*   password
            );

void      GetPassword(
                char*   password,
                long    size
            );
```

#### Description

Set a user password into PS-API for logging in to a target device.

Get a user password from PS-API for logging in to a target device.

#### Argument for SET

password	Character strings (255bytes or less)	Login password. The strings must be terminated with null character. If user authentication is set to off, please set password to "".  Default value is   NULL.
----------	---	--

#### Argument for GET

password	Character strings	Get login password Application needs to allocate and deallocate memory. If getting value failed, it is set to NULL.
size	Buffer size of password	Specify the buffer size for password. (Byte)

#### Return value for SET

0	Success
Except 0	Error code

#### Error

## Note

---

## Sequence

---

### 6.1 PlayLive

## Sample program code

---

[Visual C++ 2012] ..¥Sample Program¥PS-API¥Visual C++ 2012¥001\_PlayLive

## Reference

---

### 5.3.2.11. SecureCommunicationMode

**Class**            **IPSAPI**

**Property**        **SecureCommunicationMode**

long      SetSecureCommunicationMode(  
                                 long mode  
                                 );

long      GetSecureCommunicationMode ();

#### **Description**

Set the HTTP Communication protocol (HTTP or HTTPS) to PS-API.

Get the HTTP Communication protocol (HTTP or HTTPS) to PS-API

#### **Argument for SET**

mode	0 : HTTP	Set the HTTP protocol.
	1 : HTTPS	
		Default value is 0.

#### **Return value for SET**

0	Success
---	---------

Except 0	Error code
----------	------------

#### **Return value for GET**

Get the HTTP protocol
0 : HTTP
1 : HTTPS

#### **Error**

## Note

- Set target device to HTTPS settings. (\*Refer to each device manuals)
- A Self-signed Certificate or CA certificate is installed on the target device, only HTTP communication will be encrypted.
- Confirm the following settings in Internet Explore-Internet Options-Advanced
  - “Use SSL2.0”, “Use SSL3.0” is unchecked.
  - “TLS1.3” is only compatible with Windows 11.
- HTTPS is supported for the following communications.
  - (1) CGI Send and receive
  - (2) JPEG stream reception (including audio reception / transmission)
  - (3) H264/H265/MPEG4 stream reception in InternetMode (including audio reception / transmission)
- FTPS protocol is not supported.
- Use HTTPPort property for Port number of HTTPS.  
Set HTTPPort property according to the HTTPS port number of the connecting device.
- Set SecureCommunicationMode before Open method.
- HTTPS communication availability for each device type is below.

DeviceType	HTTPS communication	Remarks
0 : HD300	×	
1 : NWDR	×	
2 : Camera	○	Below models are not deprecated NP502/SP300/SF330/SP100/SW350/SC384/SC385/SF340 /SF135/SW155/SW396/SW316/SC386/SP509/SW559 /SF539/SF549/SF438/SW458/SF448/SW598/SP307 /SF337/SW115/SW374/SW397/SF105A/SC384B
3 : Encoder	○	Below models are not deprecated GXE100
4 : HD600/700	×	
6 : NX Serise	○	

- When setting HTTPS, if there is something abnormal such as slow method response (over 10 seconds), please update the driver of Network adapter with Windows update.

## Sequence

### 6.18. SSL

## Sample program code

## Reference

### 5.3.3. Application Listener

#### 5.3.3.1. OnError

Class	IAppListener
Listener	OnError
void	OnError( long errorCode, const char* description );

#### Description

Notify the error to the specified application.

Application needs to create the listener class that IAppListener and to implement OnError method.

#### Argument

errorCode	Negative value	Error code number
description	Character strings	Detailed error description <b>* This parameter is valid in the inside of OnError function. When OnError function is finished, this memory is freed.</b>

#### Return value

None
------

#### Error

**Note**

---

Please don't use PS-API functions in the notified listener function.

**Sequence**

---

**Sample program code**

---

**Reference**

---

### 5.3.3.2. SetErrListener

**Class**                      **IPSAPI**

**Listener**                      **SetErrListener**

```
long      SetErrListener(  
                         IAppListener*   pReceiver  
                         );
```

#### **Description**

Set the instance of listener class that implement “5.3.3.1 OnError” .  
After setting the listener, PS-API can notify any error information by calling “5.3.3.1 OnError”,  
when it happens.

#### **Argument**

pReceiver	Pointer	Pointer for Listener class. Don't delete the registered object by your application.
	NULL	If pReceiver is set to NULL, PS-API unregister the Listener class.

#### **Return value**

0	Success
Except 0	Error code

#### **Error**

Error is defined by the return value.



**Note**

---

Don't delete the registered listener object by your application.

**Sequence**

---

**Sample program code**

---

**Reference**

---

## 5.4. Device Group

### 5.4.1. Method

#### 5.4.1.1. GetDevStatus

Class	IPSAPI
Method	GetDevStatus
long	GetDevStatus( long channel );

#### Description

Get the device status.

In case of a NWDR, get the status of the device that connect to the specified channel.

In case of HD300, NV200, NV250, NV300, NX Series, a network camera, an encoder or HD600/700, get the status of device itself.

#### Argument

channel	1 : Network Camera 1 to 4 : Encoder NX100 1 to 16 : ND200, HD300, HD600/700 1 to 24 : NV200, NV250 1 to 32 : ND300, NV300 NX200, NX300 1 to 64 : ND400 1 to 128 : NX400	Specify the video channel.
---------	--	----------------------------

#### Return value

-1	Fail to get status.
0	There is no device on the specified channel
1	Does properly connect to the device.
2	Does not properly connect to the device.

#### Error

Get the error information by OnError event.
---

**Note**

---

In case of ND400, NV200, NV250, NV300, NX Series and HD600/700, there is another way to get the connection status. For details, refer to 5.4.1.11 `GetInfoString()`.

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.4.1.2. RecCtrl

**Class**                      **IPSAPI**

**Method**                      **RecCtrl**

```
long      RecCtrl(
           long    channel,
           long    command,
           long&   state,
           long    mode,
           IAppCallBack*   pSender
           );
```

#### **Description**

Turn on and off the manual recording for the recorder with manual recording function.

#### **Argument**

channel	0 : All channels 1 to 16 : ND200, HD300 HD600/700 1 to 32 : ND300 1 to 64 : ND400	Specify the video channel. When it set to zero, it refers to all channels. If specifying the channel for HD300, ND200, ND300 or HD600/700, all channels recording will be started.
command	0 : Turn off recording 1 : Turn on recording	
state	-1 : Fail to get status. 0 : OFF 1 : Manual Recording 2 : Event Recording 3 : Schedule Recording 4 : Emergency Recording	In case of blocking mode, the current status is set by PS-API.
mode	0 : Blocking Except 0 : Non-blocking	Blocking mode.

pSender

Callback interface

Specify a callback interface for the recorder control operation. Please refer to application callback section and look for function OnRecCtrl() for details.  
Don't delete the registered object by your application.

---

**Return value**

0	Success
Except 0	Error code

---

**Error**

Error is defined by the return value.  
Get the error information by OnError event.

---

**Note**

**Don't delete the registered callback object by your application.**

NV200 ,NV250, NV300,NX Series doesn't support manual recording.

---

**Sequence**

6.8 RecCtrl

---

**Sample program code**

[Visual C++ 2012] ..¥Sample Program¥PS-API¥Visual C++ 2012¥008\_RecCtrl

---

**Reference**

---

### 5.4.1.3. GetRecStatus

**Class**                **IPSAPI**

**Method**            **GetRecStatus**

```
long     GetRecStatus(
           long   channel
           );
```

#### Description

Get the recording status.

#### Argument

channel	0 : All channels 1 to 4 : NX100 1 to 16 : ND200, HD300, HD600/700 1 to 24 : NV200, NV250 1 to 32 : ND300, NV300, NX200, NX300 1 to 64 : ND400 1 to 128 : NX400	Specify the video channel. In case of "0 : All channels", even if only one channel is in recording, return value is "1 : In recording".
---------	--	--

#### Return value

	[Specified a channel]
-1	Fail to get status.
0	OFF
1	Manual Recording
2	Event Recording
3	Schedule Recording
4	Emergency Recording
	[Specified All channels]
-1	Fail to get status.
0	OFF
1	In recording

#### Error

Get the error information by OnError event.

**Note**

---

In case of NWDR, NX Series, if a camera is not registered on the specified channel, GetRecStatus returns 0.

NV200, NV250, NV300, NX Series doesn't support manual recording.

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.4.1.4. Search

Class	IPSAPI
Method	Search
long	Search( long channel, char* startTimeDate, char* endTimeDate, long type, ISearchResult* searchResult, long mode, IAppCallback* pSender );

#### Description

Search the recording data on a target recording device.

**Time zone information is NOT included in search results.**

#### Argument

channel	0 : All channels 1 to 4 : NX100 1 to 16 : ND200, HD300, HD600/700 1 to 24 : NV200, NV250 1 to 32 : ND300, NV300, NX200, NX300 1 to 64 : ND400 1 to 128 : NX400	Specify the video channel. When it set to zero, it refers to all channels. When it set to zero and SearchMultiChMask property is set the value, Search method works with the channels that is specified by SearchMultiChMask property.
startTimeDate	yyyy/mm/dd hh:mm:ss	Search the recording data that started recording after the specified time. The strings must be terminated with null character.
endTimeDate	yyyy/mm/dd hh:mm:ss	Search the recording data that started recording before the specified time. The strings must be terminated with null character.



eventType

Bit 0 to Bit 14 :

Specify the recording event kind by 32 digit binary.

- Bit 0 : Emergency (EMR)
- Bit 1 : Manual (MAN)
- Bit 2 : Schedule (SCH)
- Bit 3 : Terminal (TRM)
- Bit 4 : Command alarm (COM)
- Bit 5 : Camera site alarm (CAM/SITE)  
\* In case of HD300, Bit5 is VMD.
- Bit 6 : SD Backup (SD)
- Bit 7 : Video Loss (LOSS)
- Bit 8 : VMD alarm (VMD)
- Bit 9 : Motion alarm (CMTN)
- Bit 10 : Loitering alarm (CSTY)
- Bit 11 : Removal alarm (CRMV)
- Bit 12 : Scene change alarm (CSCD)
- Bit 13 : Terminal alarm (CTRM)
- Bit 14 : Direction alarm (CDRT)
- Bit 15 to Bit 32 : Reserved

Bit	HD300	ND200 ND300 ND400	NV200 NV250 NV300	HD600 HD700	NX100 NX200 NX300 NX400
0	EMR	EMR	-	EMR	<b>EMR</b>
1	MAN	MAN	-	MAN	-
2	SCH	SCH	SCH	SCH	<b>SCH</b>
3	TRM	TRM	TRM	TRM	<b>TRM</b>
4	COM	COM	COM	COM	<b>COM</b>
5	<b>(VMD)</b>	<b>CAM</b>	<b>CAM</b>	<b>SITE</b>	<b>CAM</b>
6	-	<b>SD</b>	<b>SD</b>		<b>SD</b>
7	<b>LOSS</b>	-	-	<b>LOSS</b>	-
8	<b>VMD</b>	-	-	<b>VMD</b>	-
9	-	-	-	<b>CMTN</b>	-
10	-	-	-	<b>CSTY</b>	-
11	-	-	-	<b>CRMV</b>	-
12	-	-	-	<b>CSCD</b>	-
13	-	-	-	<b>CTRM</b>	-
14	-	-	-	<b>CDRT</b>	-

Bit map format

MSB	-	-	-	-	-	-	LSB
0	0	0	...	0	0	0	0
Bit 31	Bit 30	Bit 29	...	Bit 3	Bit 2	Bit 1	Bit 0

e.g. Emergency + Terminal  
[Binary] 0001001 ---> type = 9

Except for NX Series, SD backup search and the other type search cannot be used together at the same time.  
If specifying "1" to the other bits, SD backup are ignored.  
In case of NWDR and NX Series, Bit7 is ignored if it is set to "1".  
In case of HD300, Bit6 is ignored if it is set to "1".  
In case of HD300, Bit5 or Bit8 is VMD alarm event search.

---

searchResult pointer

Set the pointer of ISearchResult class forgetting the search result.  
ISearchResult class is defined as follows.  
Please create and destruct an instance in application.

```
-- isearchresult.h --

// Create the ISearchResult instance
ISearchResult* GetISearchResult();
// Delete the ISearchResult instance
void DeleteISearchResult( ISearchResult* isearchresult);

class ISearchResult {
public:
//Delete all search result elements
long Clear();

// Return the number of search result elements
long GetListCount(long& ICount);

// Get a search result element with index (without memory de-allocation.)
long GetResult(long IIndex, SSEARCHRSULT_INFO *stSetInfo);

// Get a search result element with memory deallocation.
long GetAndDelete(SSEARCHRSULT_INFO *stSetInfo);
}
```

```
--psapidef.h --
```

```
typedef struct {
    long        m_lChannel;
    TCHAR       m_cStartTD[32];
    TCHAR       m_cEndDT[32];
    TCHAR       m_cRecKind[8];
    long        m_lAudio;
} SSEARCHRSLT_INFO;
```

m_lChannel	Channel
m_cStartTD[32]	Recording start date(yyyy/mm/dd hh:mm:ss)
m_cEndTD[32]	Recording stop date(yyyy/mm/dd hh:mm:ss) If the recording stop date cannot be got, it is set to the recording start date.
m_cRecKind[8]	Recording event kind EMR : emergency event MAN : manual event SCH : schedule event TRM : terminal event COM : command alarm event VMD : VMD alarm event (HD300, HD600/700) CAM : camera site alarm event (NWDR, HD600/700 , NX Series) SD : SD backup (NWDR , NX Series) LOSS : Camera loss alarm event (only HD600/700) CMTN : Motion alarm event(only HD600/700) CSTY : Loitering alarm event (only HD600/700) CRMV : Removal alarm event (only HD600/700) CSCD : Scene change alarm event (only HD600/700) CTRM : Terminal alarm event(only HD600/700) CDRT : Direction alarm event (only HD600/700)
m_lAudio	Audio ON/OFF 0 : OFF 1 : ON

mode	0 : Blocking Except 0 : Non-blocking	Blocking mode
pSender	Callback interface	Specify a callback interface for the recorder control operation. Please refer to application callback section and look for function OnSearch() for details. Don't delete the registered object by your application.

## Return value

0	Success
Except 0	Error code

## Error

Error is defined by the return value.  
Get the error information by OnError event.

## Note

**Don't delete the registered callback object by your application.**

The recording stop date of SSEARCHSLT\_INFO (m\_cEndDT) and the audio data (m\_lAudio) is supported ND400 (Firmware version 1.20 or later). In case of unsupported device, the recording stop date is set to the same value as the recording start date and the audio mode is set to 0.

**In case that Search method is called during the network playback or the local file playback, please stop the network playback and the local file playback before calling Search method.**

The maximum number of search result count is 3000.

When TRM/CMD/CAM/SITE/VMD/LOSS/CMTN/CSTY/CRMV/CSCD/CTRM/CDRT are specified in search condition, all event pre recording data that exist between the specified time are included in the search result list.

In case of a NX Series, when NXStreamNumber property is set to "2", the sub-stream recording is searched.

The sub-stream recording supports H.265, H.264.

When the "HDD Standby Control" setting is "ON" in the NX Series, please specify the duration between startTimeDate and endTimeDate within 120 minutes.

The following methods are related to ISearchResult class.

Method	Description
ISearchResult* GetISearchResult();	Create ISearchResult class instance and return the pointer. The created instance should be destructed by DeleteISearchResult method.
void DeleteISearchResult( ISearchResult* isearchresult );	Destruct isearchresult.
long ISearchResult::Clear();	Delete the SSEARCHSLT_INFO structure array that is gotten by searching.
long ISearchResult::GetListCount( long& lCount );	Get the array count of the SSEARCHSLT_INFO structure array that is gotten by searching.
long ISearchResult::GetResult( long lIndex, SSEARCHSLT_INFO *stSetInfo );	Copy the lIndex th element from the SSEARCHSLT_INFO structure array that is gotten by searching. to stSetInfo.
long ISearchResult::GetAndDelete( SSEARCHSLT_INFO *stSetInfo );	Copy the first element from the SSEARCHSLT_INFO structure array that is gotten by searching. to stSetInfo, and delete the element from ISearchResult class.

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.4.1.5. SearchEx

**Class**                      **IPSAPI**

**Method**                    **SearchEx**

```
long     SearchEx(
           long   channel,
           char*   startTimeDate,
           char*   endTimeDate,
           long   type,
           ISearchResult*   searchResult,
           long   mode,
           IAppCallback*   pSender
           );
```

#### Description

Search the recording data on a target recording device.  
**Time zone information is included in search results.**

#### Argument

channel	0 : All channels 1 to 4 : NX100 1 to 16 : ND200, HD300, HD600/700 1 to 24 : NV200, NV250 1 to 32 : ND300, NV300, NX200, NX300 1 to 64 : ND400 1 to 128 : NX400	Specify the video channel. When it set to zero, it refers to all channels. When it set to zero and SearchMultiChMask property is set the value, Search method works with the channels that is specified by SearchMultiChMask property.
startTimeDate	yyyy/mm/dd hh:mm:ss	Search the recording data that started recording after the specified time. The strings must be terminated with null character.
endTimeDate	yyyy/mm/dd hh:mm:ss	Search the recording data that started recording before the specified time. The strings must be terminated with null character.

eventType

Bit 0 to Bit 14 :

Specify the recording event kind by 32 digit binary.

- Bit 0 : Emergency (EMR)
- Bit 1 : Manual (MAN)
- Bit 2 : Schedule (SCH)
- Bit 3 : Terminal (TRM)
- Bit 4 : Command alarm (COM)
- Bit 5 : Camera site alarm (CAM/SITE)  
\* In case of HD300, Bit5 is VMD.
- Bit 6 : SD Backup (SD)
- Bit 7 : Video Loss (LOSS)
- Bit 8 : VMD alarm (VMD)
- Bit 9 : Motion alarm (CMTN)
- Bit 10 : Loitering alarm (CSTY)
- Bit 11 : Removal alarm (CRMV)
- Bit 12 : Scene change alarm (CSCD)
- Bit 13 : Terminal alarm (CTRM)
- Bit 14 : Direction alarm (CDRT)
- Bit 15 to Bit 32 : Reserved

Bit	HD300	ND200 ND300 ND400	NV200 NV250 NV300	HD600 HD700	NX100 NX200 NX300 NX400
0	EMR	EMR	-	EMR	<b>EMR</b>
1	MAN	MAN	-	MAN	-
2	SCH	SCH	SCH	SCH	<b>SCH</b>
3	TRM	TRM	TRM	TRM	<b>TRM</b>
4	COM	COM	COM	COM	<b>COM</b>
5	<b>(VMD)</b>	<b>CAM</b>	<b>CAM</b>	<b>SITE</b>	<b>CAM</b>
6	-	<b>SD</b>	-	-	<b>SD</b>
7	<b>LOSS</b>	-	-	<b>LOSS</b>	-
8	<b>VMD</b>	-	-	<b>VMD</b>	-
9	-	-	-	<b>CMTN</b>	-
10	-	-	-	<b>CSTY</b>	-
11	-	-	-	<b>CRMV</b>	-
12	-	-	-	<b>CSCD</b>	-
13	-	-	-	<b>CTRM</b>	-
14	-	-	-	<b>CDRT</b>	-

Bit map format

MSB	-	-	-	-	-	-	LSB
0	0	0	...	0	0	0	0
Bit	Bit	Bit	...	Bit	Bit	Bit	Bit
31	30	29		3	2	1	0

e.g. Emergency + Terminal  
[Binary] 0001001 ---> type = 9

Except for NX Series, SD backup search and the other type search cannot be used together at the same time.  
If specifying "1" to the other bits, SD backup are ignored.  
In case of NWDR and NX Series, Bit7 is ignored if it is set to "1".  
In case of HD300, Bit6 is ignored if it is set to "1".  
In case of HD300, Bit5 or Bit8 is VMD alarm event search.

---

searchResultEx pointer

Set the pointer of ISearchResultEx class forgetting the search result.  
ISearchResultEx class is defined as follows.  
Please create and destruct an instance in application.

```
-- isearchresult.h --  
  
// Create the ISearchResultEx instance  
ISearchResult*Ex GetISearchResultEx ();  
// Delete the ISearchResultEx instance  
void DeleteISearchResultEx ( ISearchResultEx* isearchresultex);  
  
class ISearchResultEx {  
public:  
//Delete all search result elements  
long Clear();  
  
// Return the number of search result elements  
long GetListCount(long& lCount);  
  
// Get a search result element with index (without memory de-allocation.)  
long GetResult(long lIndex, SSEARCHRSULT_INFO_EX *stSetInfo);  
  
// Get a search result element with memory deallocation.  
long GetAndDelete(SSEARCHRSULT_INFO_EX *stSetInfo);  
}
```



```
--psapidef.h --
```

```
typedef struct {
    long        m_IChannel;
    TCHAR       m_cStartTD[32];
    TCHAR       m_cEndDT[32];
    long        m_ITimeZone;
    long        m_ITimeMode;
    TCHAR       m_cRecKind[8];
    long        m_IAudio;
} SSEARCHRSLT_INFO;
```

m_IChannel	Channel
m_cStartTD[32]	Recording start date(yyyy/mm/dd hh:mm:ss)
m_cEndTD[32]	Recording stop date(yyyy/mm/dd hh:mm:ss) If the recording stop date cannot be got, it is set to the recording start date.
m_ITimeZone	Time zone information of recorded video. (plus or minus, minutes scale) e.g. In case of GMT+9:00, m_ITimeZone=540
m_ITimeMode	Time mode information of recorded video. Summer time (0:Normal, 1:Summer time)
m_cRecKind[8]	Recording event kind EMR : emergency event MAN : manual event SCH : schedule event TRM : terminal event COM : command alarm event VMD : VMD alarm event (HD300, HD600/700) CAM : camera site alarm event (NWDR, HD600/700 , NX Series) SD : SD backup (NWDR , NX Series) LOSS : Camera loss alarm event (HD300, HD600/700) CMTN : Motion alarm event(only HD600/700) CSTY : Loitering alarm event (only HD600/700) CRMV : Removal alarm event (only HD600/700) CSCD : Scene change alarm event (only HD600/700) CTRM : Terminal alarm event(only HD600/700) CDRT : Direction alarm event (only HD600/700)
m_IAudio	Audio ON/OFF 0 : OFF 1 : ON

mode            0 : Blocking  
                 Except 0 : Non-blocking

Blocking mode

pSender        Callback interface

Specify a callback interface for the recorder control operation. Please refer to application callback section and look for function OnSearchExCB() for details. Don't delete the registered object by your application.

## Return value

0	Success
Except 0	Error code

## Error

Error is defined by the return value.  
Get the error information by OnError event.

## Note

**Don't delete the registered callback object by your application.**

The recording stop date of SSEARCHRSLT\_INFO\_EX (m\_cEndDT), the audio data (m\_lAudio) and the time zone information (m\_lTimeZone) are supported ND400 (Firmware version 1.20 or later). In case of unsupported device, the recording stop date is set to the same value as the recording start date, and the audio mode and the time zone information are set to 0.

**In case that Search method is called during the network playback or the local file playback, please stop the network playback and the local file playback before calling Search method.**

The maximum number of search result count is 3000.

When TRM/CMD/CAM/SITE/VMD/LOSS/CMTN/CSTY/CRMV/CSCD/CTRM/CDRT are specified in search condition, all event pre recording data that exist between the specified time are included in the search result list.

In case of a NX Series, when NXStreamNumber property is set to "2", the sub-stream recording is searched.

The sub-stream recording supports H.265, H.264.

When the "HDD Standby Control" setting is "ON" in the NX Series, please specify the duration between startTimeDate and endTimeDate within 120 minutes.

The following methods are related to ISearchResultEx class.

Method	Description
ISearchResultEx* GetISearchResultEx();	Create ISearchResultEx class instance and return the pointer. The created instance should be destructed by DeleteISearchResultEx method.
void DeleteISearchResultEx( ISearchResultEx* isearchresultex );	Destruct isearchresultex.
long ISearchResultEx::Clear();	Delete the SSEARCHRSLT_INFO_EX structure array that is gotten by searching.
long ISearchResultEx::GetListCount( long& lCount );	Get the array count of the SSEARCHRSLT_INFO_EX structure array that is gotten by searching.
long ISearchResultEx::GetResult( long lIndex, SSEARCHRSLT_INFO_EX *stSetInfo );	Copy the lIndex th element from the SSEARCHRSLT_INFO_EX structure array that is gotten by searching. to stSetInfo.
long ISearchResultEx::GetAndDelete( SSEARCHRSLT_INFO_EX *stSetInfo );	Copy the first element from the SSEARCHRSLT_INFO_EX structure array that is gotten by searching. to stSetInfo, and delete the element from ISearchResultEx class.

## **Sequence**

---

**6.7** Search

## **Sample program code**

---

[Visual C++ 2012] ..¥Sample Program¥PS-API¥Visual C++ 2012¥007\_Search

## **Reference**

---

#### 5.4.1.6. VMDSearchEx

Class	IPSAPI
Method	VMDSearchEx
void	VMDSearchEx( long channel, char* startTimeDate, char* endTimeDate, long mask, long aSensitivity, long axTopLeft, long ayTopLeft, long axBottomRight, long ayBottomRight, long bSensitivity, long bxTopLeft, long byTopLeft, long bxBottomRight, long byBottomRight, long cSensitivity, long cxTopLeft, long cyTopLeft, long cxBottomRight, long cyBottomRight, long dSensitivity, long dxTopLeft, long dyTopLeft, long dxBottomRight, long dyBottomRight, long imageWidth, long imageHeight, ISearchResultEx* searchResultEx, long mode, IAppCallBack pSender );
Description	Do VMD search. <b>Time zone information is included in search results.</b>

## Argument

channel	1 to 4 : NX100 1 to 16 : ND200, HD300, HD600/700 1 to 24 : NV200, NV250 1 to 32 : NV300, NX200, NX300 1 to 64 : ND400 1 to 128 : NX400	Specify the video channel.
startTimeDate	yyyy/mm/dd hh:mm:ss	Search the recording data that started recording after the specified time. The strings must be terminated with null character.
endTimeDate	yyyy/mm/dd hh:mm:ss	Search the recording data that started recording before the specified time. The strings must be terminated with null character.
mask	[HD300, HD600/700] 0 : 1[second] 1 : 1[minute] 2 : 1[hour] 3 : 24[hour]  [NV200, NV250, NV300, NX100, NX200 , NX300, NX400] 0 : 1[second] 1 : 1[minute] 2 : 5[minute] 3 : 10[minute]  [ND400] 0 – 3 : 1[minute]	masking duration When the mask duration is set, any motion will not be detected for the set duration after a motion detection. Use the mask duration when it takes time to detect motion or when motion is detected frequently.
aSensitivity	0 : OFF 1 : Low 2 : Middle 3 : High	Sensitivity of area A In case of ND400, if any value (1, 2, or 3) is set, it works with same sensitivity.
axTopLeft	0 and over	X position of the top left corner of area A
ayTopLeft	0 and over	Y position of the top left corner of area A
axBottomRight	0 and over	X position of the bottom right corner of area A
ayBottomRight	0 and over	Y position of the bottom right corner of area A

**Argument**

bSensitivity	0 : OFF 1 : Low 2 : Middle 3 : High	Sensitivity of area B In case of ND400, if any value (1, 2, or 3) is set, it works with same sensitivity.
bxTopLeft	0 and over	X position of the top left corner of area B
byTopLeft	0 and over	Y position of the top left corner of area B
bxBottomRight	0 and over	X position of the bottom right corner of area B
byBottomRight	0 and over	Y position of the bottom right corner of area B
cSensitivity	0 : OFF 1 : Low 2 : Middle 3 : High	Sensitivity of area C In case of ND400, if any value (1, 2, or 3) is set, it works with same sensitivity.
cxTopLeft	0 and over	X position of the top left corner of area C
cyTopLeft	0 and over	Y position of the top left corner of area C
cxBottomRight	0 and over	X position of the bottom right corner of area C
cyBottomRight	0 and over	Y position of the bottom right corner of area C
dSensitivity	0 : OFF 1 : Low 2 : Middle 3 : High	Sensitivity of area D In case of ND400, if any value (1, 2, or 3) is set, it works with same sensitivity.
dxTopLeft	0 and over	X position of the top left corner of area D
dyTopLeft	0 and over	Y position of the top left corner of area D
dxBottomRight	0 and over	X position of the bottom right corner of area D
dyBottomRight	0 and over	Y position of the bottom right corner of area D
imageWidth	0 and over	Width of video displayed area.
imageHeight	0 and over	Height of video displayed area.

---

**Argument**

---

searchResultEx	pointer	Set the pointer of ISearchResultEx class forgetting the search result. Please refer to 5.4.1.5 SearchEx about the details of ISearchResultEx class. Please create and destruct an instance in application.
mode	0 : Blocking Except 0 : Non-blocking	Blocking mode
pSender	Callback interface	Specify a callback interface for the recorder control operation. Please refer to application callback section and look for function OnSearchExCB() for details. Don't delete the registered object by your application.

---

**Return value**

---

0	Success
Except 0	Error code

---

**Error**

---

Error is defined by the return value.  
Get the error information by OnError event.

## Note

**Don't delete the registered callback object by your application.**

ND200 and ND300 don't support VMD search.

When using digital zoom function, please specify the positions that is translated to the once position.

The recording stop date of SSEARCHRSLT\_INFO\_EX (m\_cEndDT), the audio data (m\_IAudio) and the time zone information (m\_ITimeZone) are supported ND400 (Firmware version 1.20 or later). In case of unsupported device, the recording stop date is set to the same value as the recording start date, and the audio mode and the time zone information are set to 0.

**In case that VMDSearchEx method is called during the network playback or the local file playback, please stop the network playback and the local file playback before calling VMDSearchEx method.**

The maximum number of search result count is 200.

This function is not support the sub-stream recording of NX Series. When NXStreamNumber property is set to "2", it is the same operation as set to "1".

When the "HDD Standby Control" setting is "ON" in the NX Series, please specify the duration between startTimeDate and endTimeDate within 120 minutes.

**The following methods are related to ISearchResultEx class.**

Method	Description
ISearchResultEx* GetISearchResultEx();	Create ISearchResultEx class instance and return the pointer. The created instance should be destructed by DeleteISearchResultEx method.
void DeleteISearchResultEx( ISearchResultEx* isearchresultex );	Destruct isearchresultex.
long ISearchResultEx::Clear();	Delete the SSEARCHRSLT_INFO_EX structure array that is gotten by searching.
long ISearchResultEx::GetListCount( long& ICount );	Get the array count of the SSEARCHRSLT_INFO_EX structure array that is gotten by searching.
long ISearchResultEx::GetResult( long IIndex, SSEARCHRSLT_INFO_EX *stSetInfo );	Copy the IIndex th element from the SSEARCHRSLT_INFO_EX structure array that is gotten by searching. to stSetInfo.
long ISearchResultEx::GetAndDelete( SSEARCHRSLT_INFO_EX *stSetInfo );	Copy the first element from the SSEARCHRSLT_INFO_EX structure array that is gotten by searching. to stSetInfo, and delete the element from ISearchResultEx class.

## Sequence

## Sample program code

## Reference



#### 5.4.1.7. SearchCancel

**Class**                **IPSAPI**

---

**Method**            **SearchCancel**

---

long      SearchCancel();

---

#### **Description**

Cancel the executing Search/SearchEx/VMDSearchEx function.

The search result that are already gotten are stored to ISearchResult class or ISearchResultEx class.

---

#### **Argument**

None

---

#### **Return value**

0                      Success

Except 0          Error code

---

#### **Error**

Error is defined by the return value.

Get the error information by OnError event.

**Note**

---

The canceled callback of Search method, SearchEx method and VMDSearchEx method is notified.  
The Search method, SearchEx method and VMDSearchEx in the waiting queue are not canceled.

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.4.1.8. GetDeviceLog

**Class**                      **IPSAPI**

**Method**                    **GetDeviceLog**

```
long     GetDeviceLog(  
         long type,  
         SLOGRESULT* resultlist[],  
         long maxCount,  
         long& resultCount  
         );
```

#### **Description**

Get the recording device logs.

#### **Argument**

type	0 : Access Log 1 : Network Log 2 : Trouble Log 3 : Event Log	Log kind
SLOGRESULT	Pointer of SLOGRESULT structure array	Specify the pointer of SLOGRESULT structure array for getting logs.

```
struct SLOGRESULT {  
    //Log number  
    long m_INumber;  
  
    //Time and Date  
    //yyyy/mm/dd hh:mm:ss  
    char m_cTimeDate[20];  
  
    //Summer time mode  
    //0 : Normal time , 1 : Summer time (DST)  
    long m_isDST;  
  
    // Time zone information (plus or minus, minutes scale)  
    // e.g. In case of GMT+9:00, m_IZone=540  
    long m_IZone;  
  
    //Protocol  
    char m_cProtocol[16];  
  
    //Log description  
    char m_cMsg[128];  
};
```

---

**Argument**

maxCount	Maximum count of SLOGRESULT structure	Specify maximum count of SLOGRESULT structure. Please specify 1 or over. The maximum number of each logs depends on the target device's model.
resultCount	Total count of received log count.	The gotten log count is set.

---

**Return value**

0	Success
Except 0	Error code

---

**Error**

Error is defined by the return value.  
Get the error information by OnError event.

---

**Note**

---

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.4.1.9. GetDevTimeZone

**Class** IPSAPI

**Method** GetDevTimeZone

void GetDevTimeZone(  
STIMEZONE\_INFO\* timezone  
);

#### Description

Get the time zone and summer time IN/OUT table information from a target device.

#### Argument

timezoneinfo Specify a pointer to a structure describing the time zone information. Here is the definition of the timezoneinfo structure:

```
struct SDSTINFO{  
    //DST in time  
    //yyyy/mm/dd hh:mm:ss  
    char    m_cInTime[20];  
  
    //DST out time  
    //yyyy/mm/dd hh:mm:ss  
    char    m_cOutTime[20];  
}  
  
struct STIMEZONEINFO{  
    //Time zone (plus or minus, minutes scale)  
    //e.g. In case of GMT+9:00, m_lTimezone=540  
    long    m_lTimezone;  
  
    //Current DST mode  
    // 0 : Normal, 1 : Summer time(DST)  
    // 2 : Auto setting  
    long    m_isDst;  
  
    //DST range list  
    SDSTINFO stDstList[10]  
};
```

---

**Return value**

---

None

---

**Error**

---

Get the error information by OnError event.

---

**Note**

---

In case of HD300, the time zone information (m\_lTimezone) cannot be gotten. m\_lTimezone is set to 0.

In case of camera or encoder, the DST range list (stDstList) cannot be gotten.

---

**Sequence**

---

---

**Sample program code**

---

---

**Reference**

---

#### 5.4.1.10. GetDevCurrentInfo

**Class**                      **IPSAPI**

---

**Method**                      **GetDevCurrentInfo**

---

long      GetDevCurrentInfo();

---

#### **Description**

Get the device information and hold it.  
When refer the gotten information, please use GetInfoString method.

---

#### **Argument**

None

---

#### **Return value**

0	Success
Except 0	Error code

---

#### **Error**

Error is defined by the return value.  
Get the error information by OnError event.

**Note**

---

Depends on network load or environment, it takes about 10 seconds to finish this method.

**Sequence**

---

**Sample program code**

---

**Reference**

---



#### 5.4.1.11. GetInfoString

**Class**                      **IPSAPI**

**Method**                      **GetInfoString**

```
long      GetInfoString(  
                         char*   Key,  
                         char*   value,  
                         long   size  
                         );
```

#### **Description**

Get a value from the kept information that was gotten by GetDevCurrentInfo method.

#### **Argument**

key	Character strings (15 bytes or less)	Specify the key name. The strings must be terminated with null character.
value	Character strings (255 bytes or less)	Get the pair to specified key. Application needs to allocate and deallocate memory. If getting value failed, it is set to NULL.
size	Buffer size of value	Specify the buffer size for value. (Byte)

#### **Return value**

0	Success
Except 0	Error code

#### **Error**

Error is defined by the return value.  
Get the error information by OnError event.

**Note**

---

If there is no pair to specified key, this method returns an error and the value is set to NULL.  
In case of ND400, NV200, NV250, NV300, NX Series and HD600/700, it is possible to get the connection status of the device , using the key “API\_CAM\_CONNECT\_XXCH” .

**Key**

“API\_CAM\_CONNECT\_XXCH”      XX is channel number

**value**

[ND400, NV200, NV250, NV300, NX Series]

0: camera is not registration

1: camera is connected

2: camera is not connected

[HD600/700]

1: camera is connected

2: camera is not connected

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.4.1.12. SetCameraTime

**Class**                      **IPSAPI**

**Method**                    **SetCameraTime**

```
long     SetCameraTime (
                char*   timeDate,
                long   isDst
            );
```

#### Description

Set specified time to a target device.

#### Argument

timeDate	yyyy/mm/dd hh:mm:ss	Specify a time and date character string to change the setting of a target device. The strings must be terminated with null character.
isDst	0:Normal 1:Summer time	Summer time information of the time that is specified to timeDate.

#### Return value

0	Success
Except 0	Error code

#### Error

Error is defined by the return value.  
Get the error information by OnError event.

#### Note

Time display format is set to "24h" except for the following models  
NP240 , NW484 , NS202 , NS950 , NP304 , NP502 , SP300 , SP100 , SC385 , SF340 , SW350 ,  
SW559 , SW155 , SW316 , SC386 , SF539 , SF438 , SW458 , SP307 , SF337 , SW598 , SW158  
SF138 , ST165 , SW175 , SW397 , SW374 , SW115 , SC384B series  
GXE500 , GXE100 series .

#### Sequence

#### Sample program code

#### Reference

#### 5.4.1.13. GetStatisticsData

**Class**                      **IPSAPI**

**Method**                      **GetStatisticsData**

```
long     GetStatisticsData(
           char*   startTimeDate,
           char*   endTimeDate,
           char*   fileName,
           );
```

#### Description

Get statistics data from NV200 ,NV250 ,NV300 ,NX Series.  
When Statistical information license is available, the function of statistics download works.

#### Argument

startTimeDate	yyyy/mm/dd hh:mm:ss	Get the statistics data after the specified time. The strings must be terminated with null character.
endTimeDate	yyyy/mm/dd hh:mm:ss	Get the statistics data before the specified time. Please specify the duration between startTimeDate and endTimeDate less than a day. The strings must be terminated with null character.  e.g.) startTimeDate = 2012/04/01 00:00:00 endTimeDate = 2012/04/01 23:59:59
fileName	Character strings (255 byte or less)	Specify a complete file path, and file name for storing the downloaded data.

#### Return value

0	Success
Except 0	Error code

#### Error

Error is defined by the return value.  
Get the error information by OnError event.

**Note**

---

In case that `GetStatisticsData` method is called during the network playback, please stop the network playback before calling `GetStatisticsData` method.

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.4.1.14. SetUIDPriority

**Class**                      **IPSAPI**

**Method**                    **SetUIDPriority**

long      SetUIDPriority(  
                                 long    command  
                                 );

#### **Description**

Change the UID priority mode of ND400, NV200, NV250, NV300, NX Series and HD600/700.

#### **Argument**

command	0 : First-Come-First-Serve mode	Specify the UID priority mode.
	1 : Last-Come-First-Serve mode	

#### **Return value**

0	Success
Except 0	Error code

#### **Error**

Error is defined by the return value.  
Get the error information by OnError event.

**Note**

---

[First-Come-First-Serve mode]

The user who accessed before is more prioritized.

The user who accessed after cannot access.

[Last-Come-First-Serve mode]

The user who accessed after is more prioritized.

The connection is cut into the user who was previously accessing.

When setting user access level, the connection for high level user is prioritized.

**Sequence**

---

**Sample program code**

---

**Reference**

---

### 5.4.2. Property

#### 5.4.2.1. DevModel

**Class**                      **IPSAPI**

**Property**                **DevModel**

void	GetDevModel( char* model, long size );
------	---

---

#### **Description**

Get a device manufacture model number.

---

#### **Argument for SET**

None

---

#### **Argument for GET**

model	Character strings (255 bytes or less)	Get the model number. Application needs to allocate and deallocate memory. If getting value failed, it is set to NULL.
size	Buffer size of model	Specify the buffer size for model. (Byte)

---

#### **Error**



**Note**

---

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.4.2.2. SearchMultiChMask

**Class**                      **IPSAPI**

**Property**                      **SearchMultiChMask**

long              SetSearchMultiChMask(  
   char\*    mask  
   );

long              GetSearchMultiChMask(  
   char\*    mask  
   long    size  
   )

#### **Description**

Set the channel information into PS-API when searching (Search/SearchEx) at multiple channel at the same time..

Get the channel information from PS-API.

#### **Argument for SET**

mask	Character strings (128 bytes or less)	Specify "1 : ON", "0 : OFF" for each channels. The strings must be terminated with null character.  Default value is NULL.  When the channel argument of Search/SearchEx method is set to "0: all channels", this property is valid. the channel argument of Search/SearchEx method is set to "0: all channels" and this property is set to NULL, all channels are the target of search.  When specified 128 bytes character strings, the top (left side) of character is pair to channel 1, and the end (right side) of character is pair to channel 128.  e.g. In case of searching channel 1, 3 and 6. "10100100....."
------	--	--

#### **Argument for GET**

mask	Character strings	Get the channel information. Application needs to allocate and deallocate memory. If getting value failed, it is set to NULL.
size	Buffer size of mask	Specify the buffer size for mask. (Byte)

---

**Return for SET**

---

0                  Success

Except 0        Error code

---

**Return for GET**

---

0                  Success

Except 0        Error code

---

**Error**

---

---

**Note**

---

The specified value is shorter than the channel number that target device has, the lack value changes as "0 : OFF".

The specified value is over the channel number that target device has, the over value is ignored.

All specified values that are pair to the target device's channel numbers are set to "0 : OFF", Searchb/SearchEx method returns error.

---

**Sequence**

---

---

**Sample program code**

---

---

**Reference**

---

### 5.4.3. Application Listener

#### 5.4.3.1. OnDevStatus

Class	IAppListener
-------	--------------

Listener	OnDevStatus
----------	-------------

void	OnDevStatus( long channel, long status );
------	--

#### Description

Notify the device connection status change to the specified application.

In case of a NWDR, get the status of the device that connect to the specified channel.

In case of HD300, HD600/700, NV200, NV250, NV300, NX Series or an encoder, get the status of device itself.

Application needs to create the listener class that inherits IAppListener and to implement OnDevStatus method.

#### Argument

channel	1 : Network Camera Encoder, HD300 HD600/700 1 to 4 : NX100 1 to 16 : ND200 1 to 24 : NV200, NV250 1 to 32 : ND300, NV300, NX200, NX300 1 to 64 : ND400 1 to 128 : NX400	Specify the video channel.
status	0 : There is no device on the specified channel 1 : Does properly connect to the device. 2 : Does not properly connect to the device.	Specify a device connection status.

#### Return value

None

#### Error

**Note**

---

Please don't use PS-API functions in the notified listener function.

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.4.3.2. SetDevListener

**Class**                      **IPSAPI**

**Listener**                      **SetDevListener**

```
long      SetDevListener(  
                            IAppListener*   pReceiver  
                            );
```

#### **Description**

Set the instance of listener class that implement “5.4.3.1 OnDevStatus” .  
After setting the listener, PS-API can notify any device status change by calling “5.4.3.1 OnDevStatus”, when it happens.

#### **Argument**

pReceiver	Pointer	Pointer for Listener class. Don't delete the registered object by your application.
	NULL	If pReceiver is set to NULL, PS-API unregister the Listener class.

#### **Return value**

0	Success
Except 0	Error code

#### **Error**

Error is defined by the return value.

**Note**

---

Don't delete the registered listener object by your application.

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.4.3.3. OnRecStatus

**Class**                    **IAppListener**

**Listener**                **OnRecStatus**

```
void      OnRecStatus (
           long    channel,
           long    status
           );
```

#### Description

Notify the recording status change to the specified application.

Application needs to create the listener class that inherits IAppListener and to implement OnRecStatus method.

#### Argument

channel	1 to 4 : NX100 1 to 16 : ND200, HD300, HD600/700 1 to 24 : NV200, NV250 1 to 32 : ND300, NV300, NX200, NX300 1 to 64 : ND400 1 to 128 : NX400	Specify the video channel.
status	0 : OFF 1 : Manual Recording 2 : Event Recording 3 : Schedule Recording 4 : Emergency Recording	Specify whether a reorder device in recording or not.

#### Return value

None

#### Error



**Note**

---

Please don't use PS-API functions in the notified listener function.  
NV200, NV250, NV300, NX Series doesn't support manual recording.

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.4.3.4. SetRecListener

**Class**                      **IPSAPI**

**Listener**                      **SetRecListener**

```
long      SetRecListener(  
                            IAppListener* pReceiver  
                            );
```

#### **Description**

Set the instance of listener class that implement “5.4.3.3 OnRecStatus” .  
After setting the listener, PS-API can notify any recording status change by calling “5.4.3.3 OnRecStatus”, when it happens.

#### **Argument**

pReceiver	Pointer	Pointer for Listener class. Don't delete the registered object by your application.
	NULL	If pReceiver is set to NULL, PS-API unregister the Listener class.

#### **Return value**

0	Success
Except 0	Error code

#### **Error**

Error is defined by the return value.

**Note**

---

Don't delete the registered listener object by your application.

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.4.4. Application Callback

##### 5.4.4.1. OnRecCB

**Class**                **IAppCallBack**

**Callback**           **OnRecCB**

```
void      OnRecCB(  
                 long   status  
                 );
```

#### **Description**

Notify the recording status change to the specified application.

Application needs to create the callback class that inherits IAppCallBack and to implement OnRecCB method.

#### **Argument**

status	-1 : Fail to get status. 0 : OFF 1 : Manual Recording 2 : Event Recording 3 : Schedule Recording 4 : Emergency Recording	Specify whether a reorder device in recording or not.
--------	---	---

#### **Return value**

None

#### **Error**

**Note**

---

Please don't use PS-API functions in the notified callback function.

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.4.4.2. OnSearchCB

<b>Class</b>	<b>IAppCallBack</b>
--------------	---------------------

<b>Callback</b>	<b>OnSearchCB</b>
-----------------	-------------------

void	OnSearchCB( );
------	----------------

#### Description

Notify the completion of search by Search method to the specified application.  
Application needs to create the callback class that inherits IAppCallBack and to implement OnSearchCB method.

#### Argument

None

#### Return value

None

#### Error

**Note**

---

Please don't use PS-API functions in the notified callback function.

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.4.4.3. OnSearchExCB

<b>Class</b>	<b>IAppCallBack</b>
--------------	---------------------

<b>Callback</b>	<b>OnSearchExCB</b>
-----------------	---------------------

void	OnSearchExCB( );
------	------------------

#### Description

Notify the completion of search by SearchEx method or VMDSearch method to the specified application.

Application needs to create the callback class that inherits IAppCallBack and to implement OnSearchExCB method.

#### Argument

None

#### Return value

None

#### Error



**Note**

---

Please don't use PS-API functions in the notified callback function.

**Sequence**

---

**Sample program code**

---

**Reference**

---

## 5.5. Video Group

### 5.5.1. Method

#### 5.5.1.1. GetFrameTime

**Class**                      **IPSAPI**

---

**Method**                      **GetFrameTime**

---

void                      GetFrameTime( SVIDEO\_FRAME\_TIME\*    frametime );

---

#### Description

Get the frame time and date of the current displaying image, when getting the live image of network camera/recorder/encoder or the recording image of recorder.

---

#### Argument

frametime	Specify a pointer to a structure describing the current frame time information. Here is the definition of the frametime structure:
-----------	--

```
struct SVIDEO_FRAME_TIME{
    //yyyy/mm/dd hh:mm:ss:fff
    char   m_cLocaltime[24];

    //Time zone (plus or minus, minutes scale)
    //e.g. In case of GMT+9:00, m_ITimezone=540
    long   m_ITimezone;

    //Summer time (0:Nomal, 1:Summer time)
    long   m_ITimemode;
};
```

---

#### Return value

None

## Error

Get the error information by OnError event.

## Note

According to the target device, m\_cLocaltime, m\_ITimezone and m\_ITimemode

Depends on the target device whether m\_cLocaltime, m\_ITimezone and m\_ITimemode are unable to be got or not.

Refer to the following table.

### -PlayLive

	m_cLocaltime	m_ITimezone	m_ITimemode
HD300	local time *4	0(Fixed)	summer time
NWDR	local time *4	time zone *2	summer time
Camera *1	local time	time zone *3	summer time
Encoder	local time	time zone *3	summer time
HD600/700	local time *4	time zone of HD600/700	summer time
NX Series	local time *4	time zone *2	summer time

\*1 NP1000 doesn't support the frame time and date of the current displaying image.

Frame time is set to "".

\*2 ND200, NV200, NV250, NV300, NX Series : time zone of recorder

ND300, ND400 : time zone of the camera

\*3 In case of NTP setting of camera is manual, time zone is set to 0.

In case of NTP setting of camera is the sync mode with NTP server,  
time zone is the network camera's time zone.

\*4 In case of PlayLive with HD300, NWDR, HD600/700 or , NX Series,  
milli-second is set to "000 (Fixed)".

### -Play

	m_cLocaltime	m_ITimezone	m_ITimemode
HD300	local time *5	0(Fixed)	summer time
NWDR	local time	time zone *3	summer time
HD600/700	local time	time zone of HD600/700	summer time
NX Series	local time	time zone *3	summer time

\*3 ND200, NV200, NV250, NV300, NX Series : time zone of recorder

ND300, ND400 : time zone of the camera

\*5 In case of Play with HD300, milli-second is set to "000 (Fixed)".

### -PlayFile

	m_cLocaltime	m_ITimezone	m_ITimemode
HD300	local time *5	0(Fixed)	summer time
NWDR	local time *5	time zone *3	summer time
HD600/700	local time	time zone of HD600/700	summer time
NX Series	local time *5	time zone *3	summer time

\*3 ND200, NV200, NV250, NV300, NX Series : time zone of recorder

ND300, ND400 : time zone of the camera

\*5 In case of PlayFile with n3r(JPEG) or h3r file, milli-second is set to "000 (Fixed)".

**Sequence**

---

**Sample program code**

---

**Reference**

---

### 5.5.1.2. PlayLive

Class		IPSAPI
Method		PlayLive
long	PlayLive( long channel, long& status, long mode, IAppCallBack* pSender );	
<b>Description</b>		
Start live video play.		
<b>Argument</b>		
channel	1 : Network Camera, GXE100 1 to 3 : S8573, 1 to 4 : Encoder, X8570, X8571 S8574 NX100 1 to 16 : ND200, HD300, HD600/700 1 to 24 : NV200, NV250 1 to 32 : ND300, NV300, NX200, NX300 1 to 64 : ND400 1 to 128 : NX400	Specify the video channel.  In case of blocking mode, the current status is set by PS-API.
status	-1 : Invalid status (Run with non-blocking or not displaying Playback and Live image) 0 : stop 1 : pause 2 : play	
mode	0 : blocking Except 0 : Non-blocking	Blocking mode.
pSender	Callback interface	Specify a callback interface for the PlayLive operation. Please refer to application callback section and look for function OnPlayStatusCB() for details. Don't delete the registered object by your application.

---

**Return value**

---

0	Success
Except 0	Error code

---

**Error**

---

Error is defined by the return value.  
Get the error information by OnError event.

---

**Note**

---

**Don't delete the registered callback object by your application.**

When connecting to NWDR, NX Series and displaying MPEG-4/H.264/H.265 Live, please specify the different port number for receiving the MPEG-4/H.264 video stream to each channel.

About the setting of MPEG-4/ H.264/H.265 port number, please refer to "5.5.2.1 SetMPEG4Port" or "5.5.2.2 SetH264Port".

When connecting to NWDR, NX Series and displaying MPEG-4/H.264/H.265 Live, the response of starting Live may be late if the refresh rate is long.

In case of HD600/700, when playing Live with H264Resolution set to 320, OSD is not displayed.

When using 360-degree Network Camera, please refer to "4.7 About 360-degree Network Camera".

In case H.264/H.265 video is not display, please try setting as bellow.

(1)Set DecResolutionMode property to 3.

(2)Set H264Resolution property smaller than video stream resolution it has being receiving.(\*However, the resolution will be low.)

If a network fault occurs during Internet mode live, even if the network fault recovery, you may not hear audio.In case make audio hear, please execute PlayLive again.

---

**Sequence**

---

**6.1 PlayLive**

---

**Sample program code**

---

[Visual C++ 2012] ..¥Sample Program¥PS-API¥Visual C++ 2012¥001\_PlayLive

---

**Reference**

---

### 5.5.1.3. Play

**Class**                      **IPSAPI**

**Method**                    **Play**

```
long     Play(
        long   channel,
        char*   timeDate,
        long&   status,
        long   mode,
        IAppCallBack*   pSender
    );
```

#### Description

Start recorded video play.

#### Argument

channel	1 to 4 : NX100 1 to 16 : ND200, HD300, HD600/700 1 to 24 : NV200, NV250 1 to 32 : ND300, NV300, NX200, NX300 1 to 64 : ND400 1 to 128 : NX400	Specify the video channel.
timeDate	yyyy/mm/dd hh:mm:ss NULL or Empty string	Specify a time and date character string for the recorded video playback. The strings must be terminated with null character. If "NULL" or "Empty String" is specified, PS-API skip to the latest recording time and start playing.
status	-1 : Invalid status (Run with non-blocking mode or not displaying Playback and Live image) 0 : stop 1 : pause 2 : play	In case of blocking mode, the current status is set by PS-API.
mode	0 : blocking Except 0 : Non-blocking	Blocking mode.
pSender	Callback interface	Specify a callback interface for the Play operation. Please refer to application callback section and look for function OnPlayStatus() for details. Don't delete the registered object by your application.

---

**Return value**

0	Success
Except 0	Error code

---

**Error**

Error is defined by the return value.  
Get the error information by OnError event.

---

**Note**

**Don't delete the registered callback object by your application.**

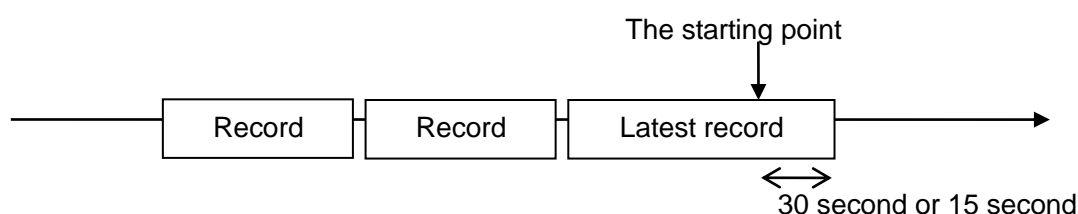
In case of HD300, it is necessary that the total recording frame rate of all playback stream is 15 ips or less.

If the total is over 15 ips, the playback isn't smooth.

If specify the NULL or empty string to timeDate, playback starts from the following point.

Time between the starting point and the record terminal depends on the device setup. Default value is 30 second(except for NX Series), 15 second(in case of NX Series).

However if you repeatedly execute Play method on this setting in the NX Series, audio might not be heard.



In the case a NX Series by designating date & time, playback may start from a point several seconds before or after the designated date & time.

In case H.264/H.265 video is not display, please try setting as bellow.

(1)Set DecResolutionMode property to 3.

(2)Set H264Resolution property smaller than video stream resolution it has being receiving.(\*However, the resolution will be low.)

In case of a NX Series, when SIDMode property is set to '0' and NXStreamNumber property is set to '2', the sub-stream recording is played.

The sub-stream recording supports H.265, H.264.

To execute Play mehod right after HttpMP4Download or HttpDownload method, execute PlayLive method once and then execute Play method again.

If there is a difference between "RcvAudioDec property value" and "Recorder's audio format setting or recording data format", multiple OnError may be notified when SetErrListener is enabled.

---

**Sequence**

---

**6.2 Play**

---

**Sample program code**

[Visual C++ 2012] ..¥Sample Program¥PS-API¥Visual C++ 2012¥002\_Play

---

**Reference**





#### 5.5.1.4. PlayFile

**Class**                      **IPSAPI**

**Method**                    **PlayFile**

```
long     PlayFile(
           char*   fileName,
           long&   status,
           long   mode,
           IAppCallBack*   pSender
           );
```

#### **Description**

Start video play from a specified video file.

#### **Argument**

fileName	Character strings (255 bytes or less)	Specify a file name and its full path for playing recorded video. The strings must be terminated with null character.
status	-1 : Invalid status (Run with non-blocking mode or not displaying Playback and Live image) 0 : stop 1 : pause 2 : play	In case of blocking mode, the current status is set by PS-API.
mode	0 : blocking Except 0 : Non-blocking	Blocking mode.
pSender	Callback interface	Specify a callback interface for the PlayFile operation. Please refer to application callback section and look for function OnPlayStatusCB() for details. Don't delete the registered object by your application.

#### **Return value**

0	Success
Except 0	Error code

---

**Error**

Error is defined by the return value.  
Get the error information by OnError event.

---

**Note**

**Don't delete the registered callback object by your application.**

If Open method or Connect method is executed during PlayFile working, PlayFile stops.  
When doing PlayFile with the n3r file which has a password, please set password to "5.5.2.10  
FilePassword" property.

---

**Sequence**

6.3 PlayFile

---

**Sample program code**

[Visual C++ 2012] ..¥Sample Program¥PS-API¥Visual C++ 2012¥003\_PlayFile

---

**Reference**

---

#### 5.5.1.5. PlayControl

**Class**                      **IPSAPI**

**Method**                    **PlayControl**

```
long      PlayControl(
                long    command,
                long    speed,
                long&   status,
                long    mode,
                IAppCallBack* pSender
            );
```

#### **Description**

Control video play. (Forward, Pause, etc)

#### **Argument**

command	0 : stop play	Specify the playback kind such as stop or play. When using 8: step fast forward, 9: step fast rewind, playback speed is stepped up one by one to 7th step. Please refer to NOTE about the detailed description of playback speed. When restart the playback after pause, please specify 4, 5, 8 or 9 to command. Please refer to NOTE about playback speed of each Step. 10 : next record and 11 : previous record is valid in network playback, is invalid in playback file. When using 10 : next recoed and 11 : previous record, playback direction and speed will be kept. Next recoed and previous record are not supported for HD300.
	1 : stop live	
speed	2 : stop file playback	
	3 : pause play	
	4 : forward play	
	5 : rewind play	
	6 : next frame	
	7:previous frame	
	8 : step fast forward	
	9 : step fast rewind	
	10 : next record	
	11 : previous record	
	1 : Step1	Specify the play speed. When command is set to "4" or "5", speed is valid. To specify playback speed directly is not supported for HD300.
speed	2 : Step2	
	3 : Step3	
	4 : Step4	
	5 : Step5	
	6 : Step6	
	7 : Step7	

---

**Argument**

status	-1 : Invalid status (Run with non-blocking mode or not displaying Playback and Live image) 0 : stop 1 : pause 2 : play	In case of blocking mode, the current status is set by PS-API.
mode	0 : blocking Except 0 : Non-blocking	Blocking mode.
pSender	Callback interface	Specify a callback interface for the PlayControl operation. Please refer to application callback section and look for function OnPlayStatus() for details. Don't delete the registered object by your application.

---

**Return value**

0	Success
Except 0	Error code

---

**Error**

Error is defined by the return value.  
Get the error information by OnError event.

**Note**


---

**Don't delete the registered callback object by your application.**

The description of the playback speed is as follows.

[HD300] \*1

Resolution	Step1	Step2	Step3	Step4	Step5	Step6	Step7
FRAME	x 1	x 2	x 4	x 8	x 16	x 32	x 48
FIELD	x 1	x 4	x 8	x 16	x 32	x 48	x96
SIF	x 1	x 8	x16	x 32	x 64	x128	x132

\*1) To specify playback speed directly is not supported for HD300.

[NWDR]

	Step1	Step2	Step3	Step4	Step5	Step6	Step7
ND200	x 1	x 4	x 8	x 16	x 32	x 48	x96
ND300							
ND400							
NV200							
NV250							
NV300							

[HD600/700]

	Step1	Step2	Step3	Step4	Step5	Step6	Step7
HD600/700	x 1	x 2	x 5	x 10	x 20	x 50	x100

[NX Series]

	Step1	Step2	Step3	Step4	Step5	Step6	Step7
NX100	x 1	x 4	x 8	x 16	x 32	x 48	x96
NX200							
NX300							
NX400							

When using FastPlayMode = 1 (High rate mode, the description of the playback speed is as follows.

[NWDR]

	Step1	Step2	Step3	Step4	Step5	Step6	Step7
ND400	x 1	x 2	x 4	x 8	x 16	x 32	x48
NV200							
NV250							
NV300							

[HD600/700]

	Step1	Step2	Step3	Step4	Step5	Step6	Step7
HD600/700	x 1	x 2	x 4	x 8	x 16	x 32	x48

[NX Series]

	Step1	Step2	Step3	Step4	Step5	Step6	Step7
NX100	x 1	x 2	x 4	x 8	x 16	x 32	x48
NX200							
NX300							
NX400							

---

---

**Note**

In case of multi screen playback, please refer to the following notes.

When using “Next frame/Previous frame” with the records that are recorded with different rate, the playback cannot synchronize. Because the duration between frames of each channel is different.

When using “Next record/Previous record” with the records that are separated at different time, the playback cannot synchronize. Because the starting point of playback are different.

When the duration of the record which is recorded with MPEG-4/H.264/H.265 is shorter than the refresh interval (I-frame interval), next record and previous record in keeping direction and speed doesn't work.

In this case, please do pause before next record and previous record.

In the case a NX Series by designating frame, playback may start from a point several seconds before or after the designated frame.

---

**Sequence**

---

**6.2 Play**

---

**Sample program code**

---

[Visual C++ 2012] ..¥Sample Program¥PS-API¥Visual C++ 2012¥002\_Play

---

**Reference**

---

#### 5.5.1.6. PlayControlByTime

**Class**                      **IPSAPI**

**Method**                    **PlayControlByTime**

```
long      PlayControlByTime(
                                char*   timeDate,
                                long   isDst,
                                long&   status,
                                long   mode,
                                IAppCallBack* pSender
                                );
```

#### Description

Jump to the specified date and time,  
Playback direction and speed is hold..

#### Argument

timeDate	yyyy/mm/dd hh:mm:ss	Specify a time and date character string for the recorded video playback. The strings must be terminated with null character. If "NULL" or "Empty String" is specified, error occurs.
isDst	0:Normal 1:Summer time	Summer time information of the time that is specified to timeDate.
status	-1 : Invalid status (Run with non-blocking mode or not displaying Playback and Live image) 0 : stop 1 : pause 2 : play	In case of blocking mode, the current status is set by PS-API.
mode	0 : blocking Except 0 : Non-blocking	Blocking mode.
pSender	Callback interface	Specify a callback interface for the Play operation. Please refer to application callback section and look for function OnPlayStatus() for details. Don't delete the registered object by your application.

#### Return value

0	Success
Except 0	Error code



## **Error**

---

Error is defined by the return value.  
Get the error information by OnError event.

## **Note**

---

**Don't delete the registered callback object by your application.**

HD300 doesn't support PlayControlByTime method.

In case of network playback, jump function is valid during displaying black image. When the network playback video is displayed, this method returns error.

In case of network playback, the specifiable time and date is the time between the current playback point and the beginning point of the next record.

In case of file playback, it can jump to any optional time by using this method.

## **Sequence**

---

## **Sample program code**

---

## **Reference**

---

#### 5.5.1.7. GetPlayStatus

**Class**                **IPSAPI**

**Method**            **GetPlayStatus**

long      GetPlayStatus( );

---

#### **Description**

Get current video play status.

---

#### **Argument**

None

---

#### **Return value**

-1	Invalid status (Run with Non-blocking mode or not displaying Playback and Live image)
0	Stop
1	Pause
2	Play
3	Preparing for Play

---

#### **Error**

Get the error information by OnError event.

**Note**

---

When the "HDD Standby Control" setting is "ON" in the NX Series, it may take time to play back. In that case, "status = 3 (Preparing for Play)" may be notified.

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.5.1.8. GetPlaySpeed

**Class**                **IPSAPI**

**Method**            **GetPlaySpeed**

long      GetPlaySpeed( );

---

#### **Description**

Get current video play speed.

---

#### **Argument**

None

---

#### **Return value**

	Get current play speed.
-1	Fail to get status.
1	Step1
2	Step2
3	Step3
4	Step4
5	Step5
6	Step6
7	Step7

---

#### **Error**

Get the error information by OnError event.

**Note**

The description of the playback speed is as follows.

[HD300] \*1

Resolution	Step1	Step2	Step3	Step4	Step5	Step6	Step7
FRAME	x 1	x 2	x 4	x 8	x 16	x 32	x 48
FIELD	x 1	x 4	x 8	x 16	x 32	x 48	x96
SIF	x 1	x 8	x16	x 32	x 64	x128	x132

\*1) To specify playback speed directly is not supported for HD300.

[NWDR]

	Step1	Step2	Step3	Step4	Step5	Step6	Step7
ND200	x 1	x 4	x 8	x 16	x 32	x 48	x96
ND300							
ND400							
NV200							
NV250							
NV300							

[HD600/700]

	Step1	Step2	Step3	Step4	Step5	Step6	Step7
HD600/700	x 1	x 2	x 5	x 10	x 20	x 50	x100

[NX Series]

	Step1	Step2	Step3	Step4	Step5	Step6	Step7
NX100	x 1	x 4	x 8	x 16	x 32	x 48	x96
NX200							
NX300							
NX400							

When using FastPlayMode = 1 (High rate mode, the description of the playback speed is as follows.

[NWDR]

	Step1	Step2	Step3	Step4	Step5	Step6	Step7
ND400	x 1	x 2	x 4	x 8	x 16	x 32	x48
NV200							
NV250							
NV300							

[HD600/700]

	Step1	Step2	Step3	Step4	Step5	Step6	Step7
HD600/700	x 1	x 2	x 4	x 8	x 16	x 32	x48

[NX Series]

	Step1	Step2	Step3	Step4	Step5	Step6	Step7
NX100	x 1	x 2	x 4	x 8	x 16	x 32	x48
NX200							
NX300							
NX400							

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.5.1.9. GetFrameRate

<b>Class</b>	<b>IPSAPI</b>
--------------	---------------

<b>Method</b>	<b>GetFrameRate</b>
---------------	---------------------

long	GetFrameRate( );
------	------------------

#### Description

Get the set frame rate value on the recorder at the time of recording

#### Argument

None

#### Return value

-1	Fail to get status.
0	Frame rate [ips]
:	About the detailed description of frame rate, please refer to Note.
600	

#### Error

Get the error information by OnError event.

## Note

According to target device, frame rate cannot be gotten.  
Refer to the following table.

	PlayLive	Play	PlayFile
HD300	0(Fixed)	Recording rate	Recording rate
NWDR	JPEG:Live rate MPEG-4:0(Fixed) H.264:0(Fixed)	JPEG: Recording rate MPEG-4: 300(Fixed) *1 H.264:300(Fixed) *1	JPEG: Recording rate MPEG-4:300(Fixed) H.264:300(Fixed)
Camera	0(Fixed)		
Encoder	0(Fixed)		
HD600/700	0(Fixed)	0(Fixed)	H.264:300(Fixed)
NX Series	JPEG:Live rate H.264:0(Fixed) H.265:0(Fixed)	JPEG: Recording rate H.264:300(Fixed) *1 H.265:300(Fixed)	JPEG: Recording rate H.264:300(Fixed) H.265:300(Fixed)

\*1 In case of I-frame (I-picture) of Play with ND200 and ND300,  
frame rate of MPEG-4 and H.264 is set to 0 (fixed).

## Sequence

## Sample program code

## Reference



#### 5.5.1.10. GetPicturePosition

**Class**                      **IPSAPI**

**Method**                    **GetPicturePosition**

```
long     GetPicturePosition(
           long&   xTopLeft,
           long&   yTopLeft,
           long&   xBottomRight,
           long&   yBottomRight
           );
```

#### **Description**

Get the position of the displayed picture except black panels.

#### **Argument**

xTopLeft	0 and over	The top-left x position of the displayed picture except black panels. (pixels)
yTopLeft	0 and over	The top-left y position of the displayed picture except black panels. (pixels)
xBottomRight	0 and over	The bottom-right x position of the displayed picture except black panels. (pixels)
yBottomRight	0 and over	The bottom-right y position of the displayed picture except black panels. (pixels)

#### **Return value**

0	Success
Except 0	Error code

#### **Error**

Error is defined by the return value.  
Get the error information by OnError event.

**Note**

The following points can be gotten as xTopLeft, yTopLeft, xBottomRight and yBottomRight position.

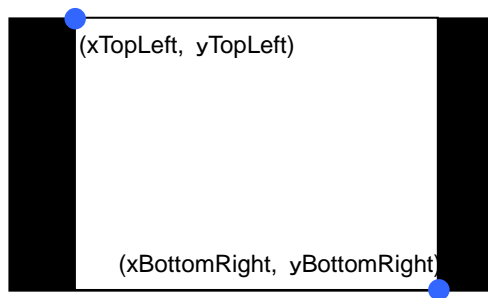


Figure 5-1 Black Panel on the Left and Right

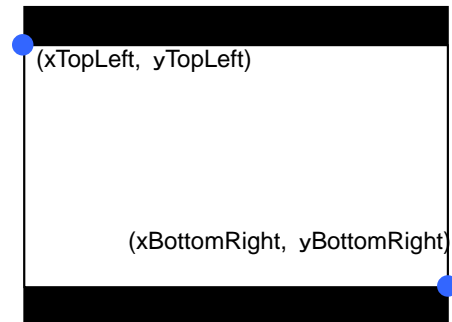


Figure 5-2 Black Panel on the Top and Bottom

**Sequence****Sample program code****Reference**

#### 5.5.1.11. GetImageResolution

**Class**                      **IPSAPI**

---

**Method**                      **GetImageResolution**

---

long      GetImageResolution(  
                                 long&    imageWidth  
                                 long&    imageHeight  
                                 );

---

#### **Description**

Get the displayed image resolution of live video and recorded video image.

---

#### **Argument**

imageWidth	0 and over	The total pixels of the horizontal direction
imageHeight	0 and over	The total pixels of the vertical direction

---

#### **Return value**

0	Success
Except 0	Error code

---

#### **Error**

Error is defined by the return value.  
Get the error information by OnError event.

**Note**

The gotten resolution depends on a target device and stream format.  
Please refer to the following table.

	PlayLive	Play	PlayFile	DecodeImage
HD300	imageWidth: 640 imageHeight: 480	imageWidth: 640 imageHeight: 480	imageWidth: 640 imageHeight: 480	
NWDR	JPEG : Resolution of video stream MPEG-4 : *1 H.264 : *2	JPEG : Resolution of video stream MPEG-4 : *1 H.264 : *2	JPEG : Resolution of video stream MPEG-4 : *1 H.264 : *2	Not support
Network camera	JPEG : Resolution of video stream MPEG-4 : *1 H.264 : *2 H.265 : Resolution of video stream			Not support
Encoder	JPEG : Resolution of video stream MPEG-4 : *1 H.264 : *2			Not support
HD600/700	H.264 : *2	H.264 : *2	H.264 : *2	Not support
NX Series	JPEG : Resolution of video stream H.264 : Resolution of video stream H.265 : Resolution of video stream	JPEG : Resolution of video stream H.264 : Resolution of video stream H.265 : Resolution of video stream	JPEG : Resolution of video stream H.264 : Resolution of video stream H.265 : Resolution of video stream	Not support

\*1 : Get the resolution of a displayed image.

In case that the installed ActiveX control doesn't support the function to get resolution,  
the resolution that is set to MPEG4Resolution property is gotten.

\*2 : Get the resolution of a displayed image.

In case that the installed ActiveX control doesn't support the function to get resolution,  
the resolution that is set to H264Resolution property is gotten.

## Note

---

- \* Whether the installed ActiveX control supports the function to get resolution or not, that depends on the version of the installed ActiveX control.  
Please refer to the following table about supported version of the function to get resolution.

Program name *3	Version *3
Network Camera View3	Not supported
Network Camera View 4	v.4.0.0.18 or later
Network Camera View 4S	v.4.0.0.16 or later
WebVideo ActiveX	v.5.0.4.0 or later
WebVideo2 ActiveX	v.4.0.18.0 or later
WebVideo ActiveX NX	v.6.0.11.0 or later

- \*3 : Program name and version can be confirmed by Add or Remove Programs (or Programs and Features) in Control Panel.

When a black image is displayed, a resolution value is gotten depends on stream format.

	imageWidth	imageHeight
JPEG	10	10
MPEG-4	352	288
H.264	32	32
H.265	64	64

## Sequence

---

## Sample program code

---

## Reference

---

#### 5.5.1.12. SetVideoWindow

**Class**                **IPSAPI**

**Method**             **SetVideoWindow**

long     SetVideoWindow(  
                                 hWnd   handle  
                                 );

#### **Description**

Specify a Windows handle for video display .

#### **Argument**

handle	Window handle	Specify a Windows handle from the application to the PS-API for video play.
--------	---------------	---

#### **Return value**

0	Success
Except 0	Error code

#### **Error**

Error is defined by the return value.  
Get the error information by OnError event.

## Note

---

## Sequence

---

### 6.1 PlayLive

## Sample program code

---

[Visual C++ 2012] ..¥Sample Program¥PS-API¥Visual C++ 2012¥001\_PlayLive

## Reference

---

#### 5.5.1.13. DecodeImage

**Class**                **IPSAPI**

**Method**            **DecodeImage**

```
long    DecodeImage(  
         unsigned char*   pBuffer  
         long   size  
         );
```

#### **Description**

Decode a compressed video image that is gotten by OnImage.

The decoded image data can be drawn on the window specified by SetVideoWindow.

When getting the decoded image data, the application can get the decoded image data by setting the video image type of SetImageListener to "1 : decoded video image" and using OnImage.

#### **Argument**

pBuffer	Pointer of buffer	Specify the buffer of a compressed video image for one frame including header.
size	Compressed video image size .[Byte]	

#### **Return value**

0	Success
Except 0	Error code

#### **Error**

Error is defined by the return value.  
Get the error information by OnError event.



### **Note**

---

The StreamFormat must be set before calling DecodeImage. If the StreamFormat and the format of Compressed video image does not match, this function returns error.

If Play/PlayFile/PlayLive is already started DecodeImage returns error.

For MPEG-4/H.264/H.265 drawing, the first Compressed video image must be I frame (I picture).

Unless I-frame (I-picture) is specified first, the drawn image is disorderly or isn't shown correctly.

When decoding the compressed video image that its Compression (in Compression method) is 0x0004 : H.264(type2), 0x0006 : H.265, cash memory for 2 frames (pictures) is needed. 1st frame (picture) is decoded when executing DecodeImage for 3rd frame (picture).

### **Sequence**

---

### **Sample program code**

---

### **Reference**

---

#### 5.5.1.14. ClearImage

<b>Class</b>	<b>IPSAPI</b>
--------------	---------------

<b>Method</b>	<b>ClearImage</b>
---------------	-------------------

void	ClearImage();
------	---------------

#### Description

Drawing area is painted over with background color specified by BackColor property.  
If using this method during Live and Play, error occurs.

#### Argument

None

#### Return value

None

#### Error

Get the error information by OnError event.

**Note**

---

**Sequence**

---

**Sample program code**

---

**Reference**

---

### 5.5.1.15. SnapShot

**Class**                    **IPSAPI**

**Method**                **SnapShot**

long      SnapShot(  
                 IPSAPIPicture\*   pContainer  
                 );

### Description

Get or save image that is displayed on the screen with JPEG / BITMAP format.

### Argument

pContainer      pointer

Specify the pointer of IPSAPIPicture class for getting snapshot. IPSAPIPicture class is defined as follows. Please create and destruct an instance in application.

[ ipsapipicture.h ]

```
// Create IPSAPIPicture instance
IPSAPIPicture*  GetIPSAPIPicture();
// Delete IPSAPIPicture instance
void  DeletelIPSAPIPicture(IPSAPIPicture*  pContainer);

class IPSAPIPicture {
public:
// Get data size of BMP format image
long  GetBitmapSize();

// Get data of BMP format image
long  GetBitmapImage(char*  pBuffer, long  size);

// Save BMP format image to a file
long  SaveBitmapImage(char*  fileName);

// Get data size of JPEG format image
long  GetJpegSize();

// Get data of JPEG format image
long  GetJpegImage(char*  pBuffer, long  size);

// Save JPEG format image to a file
long  SaveJpegImage(char*  fileName);
}
```

pBuffer	Binary data
size	pBuffer size
fileName	Completed file path

**Return value**

0	Success
Except 0	Error code

**Error**

Error is defined by the return value.  
Get the error information by OnError event.

**Note**

The size of a image gotten by SnapShot is same as the size gotten by GetImageResolution.  
Regarding on the overlay texts and frames by Overlay functions, position and size are different between the displayed image and SnapShot.  
When displaying video stream with kept aspect ratio, the black panels are not included in the gotten picture.

The following methods are related to IPSAPIPicture class.

Method	Description
IPSAPIPicture* GetIPSAPIPicture();	Create IPSAPIPicture class instance and return the pointer. The created instance should be destructed by DeleteIPSAPIPicture method.
void DeleteIPSAPIPicture( IPSAPIPicture* pContainer );	Destruct pContainer.
long IPSAPIPicture::GetBitmapSize();	Get the data size of a BITMAP format image gotten by SnapShot method. It is Byte scale.
long IPSAPIPicture::GetBitmapImage( char* pBuffer, long size );	Copy the BITMAP format image gotten by SnapShot method to pBuffer Specified pContainer size. It is Byte scale.
long IPSAPIPicture::SaveBitmapImage( char* fileName );	Save the BITMAP format image gotten by SnapShot method to a file. Specify a completed file path to fileName.
long IPSAPIPicture::GetJpegSize();	Get the data size of a JPEG format image gotten by SnapShot method. It is Byte scale.
long IPSAPIPicture::GetJpegImage( char* pBuffer, long size );	Copy the JPEG format image gotten by SnapShot method to pBuffer Specified pContainer size. It is Byte scale.
long IPSAPIPicture::SaveJpegImage( char* fileName );	Save the JPEG format image gotten by SnapShot method to a file. Specify a completed file path to fileName.

**Sequence**

---

**Sample program code**

---

**Reference**

---

### 5.5.1.16. TitleOperation

**Class**                **IPSAPI**

**Method**            **TitleOperation**

```
long    TitleOperation(
        long   id,
        long   command,
        char*   text,
        long   xPosition,
        long   yPosition,
        long   align,
        char*   font,
        long   fontSize,
        long   foreColor,
        long   borderColor,
        long   style
    );
```

#### Description

Display text strings on the video image.

#### Argument

id	1 to 6	ID for management ID is shared by TitleOperation and TitleOperationEx.
command	0 : Non display 1 : Display	
text	Character strings (Half size character : 256 bytes or less Full size character : 128 bytes or less	Specify overlay character strings. The strings must be terminated with null character.
xPosition	0 and over	X position of displayed text.
yPosition	0 and over	Y position of displayed text.
align	0 : Left and Top 1 : Center and Top 2 : Right and Top 3 : Left and Bottom 4 : Center and Bottom 5 : Right and Bottom	The placement of the text

font	Character strings (Half size character : 256 bytes or less Full size character : 128 bytes or less)	Font name The strings must be terminated with null character. e.g.) "MS UI Gothic" "Century"
fontsize	8 to 128	Font size (pt)
foreColor	0 to 16777215	Text color The sum of R (Red), G (Green) and B (Blue) R, G and B are specified with the following value.  R :        0    -        255    (0x000000 – 0x0000FF) G :       256   -       65280   (0x000100 – 0x00FF00) B :    65536   -    16711680   (0x010000 – 0xFF0000)
borderColor	0 to 16777215	Edge color The sum of R (Red), G (Green) ad B (Blue) R, G and B are specified with the following value.  R :        0    -        255    (0x000000 – 0x0000FF) G :       256   -       65280   (0x000100 – 0x00FF00) B :    65536   -    16711680   (0x010000 – 0xFF0000)
style	0 : Standard 1 : Bold 2 : Italic 3 : Bold and Italic	Text style

#### Return value

0	Success
Except 0	Error code

#### Error

Error is defined by the return value.  
Get the error information by OnError event.



**Note**

---

Please specify xPosition and yPosition based on the display size by “5.5.2.11 ImageHeight” and “5.5.2.12 ImageWidth”. Cannot display the text out of drawing area.

The text is not displayed on the image gotten by OnImage.

**Sequence**

---

**Sample program code**

---

**Reference**

---

### 5.5.1.17. TitleOperationEx

**Class**                **IPSAPI**

**Method**            **TitleOperationEx**

```
long    TitleOperationEx(
        long   id,
        long   command,
        char*   text,
        long   xPosition,
        long   yPosition,
        long   align,
        char*   font,
        long   fontSize,
        long   foreColor,
        long   borderColor,
        long   style,
        long   transmissivity
    );
```

#### **Description**

Display text strings with transmissivity on the video image.

#### **Argument**

id	1 to 6	ID for management ID is shared by TitleOperation and TitleOperationEx.
command	0 : Non display 1 : Display	
text	Character strings (Half size character : 256 bytes or less Full size character : 128 bytes or less	Specify overlay character strings. The strings must be terminated with null character.
xPosition	0 and over	X position of displayed text.
yPosition	0 and over	Y position of displayed text.
align	0 : Left and Top 1 : Center and Top 2 : Right and Top 3 : Left and Bottom 4 : Center and Bottom 5 : Right and Bottom	The placement of the text

font	Character strings (Half size character : 256 bytes or less Full size character : 128 bytes or less)	Font name The strings must be terminated with null character. e.g.) "MS UI Gothic" "Century"
fontsize	8 to 128	Font size (pt)
foreColor	0 to 16777215	Text color The sum of R (Red), G (Green) and B (Blue) R, G and B are specified with the following value.  R :       0   -       255   (0x000000 – 0x0000FF) G :       256   -       65280   (0x000100 – 0x00FF00) B :   65536   -   16711680   (0x010000 – 0xFF0000)
borderColor	0 to 16777215	Edge color The sum of R (Red), G (Green) ad B (Blue) R, G and B are specified with the following value.  R :       0   -       255   (0x000000 – 0x0000FF) G :       256   -       65280   (0x000100 – 0x00FF00) B :   65536   -   16711680   (0x010000 – 0xFF0000)
style	0 : Standard 1 : Bold 2 : Italic 3 : Bold and Italic	Text style
transmissivity	0 to 255	Transmissivity of overlay character strings. 0 (0x00) : transparent 127 (0x7F) : translucent 255 (0xFF) : opaque

#### Return value

0	Success
Except 0	Error code

#### Error

Error is defined by the return value.  
Get the error information by OnError event.

**Note**

---

Please specify xPosition and yPosition based on the display size by “5.5.2.11 ImageHeight” and “5.5.2.12 ImageWidth”. Cannot display the text out of drawing area.

The text is not displayed on the image gotten by OnImage.

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.5.1.18. GetTitle

**Class**                **IPSAPI**

**Method**            **GetTitle**

```
long     GetTitle(  
         long   id,  
         char*   pBuffer,  
         long   size  
         );
```

#### **Description**

Get the text strings of specified ID.

#### **Argument**

id	1 to 6	ID for management
pBuffer	Character strings (256 bytes or less)	Get the text string of specified ID. If getting value failed, it is set to NULL.
size	pBuffer size	Specify the size of pBuffer

#### **Return value**

0	Success
Except 0	Error code

#### **Error**

Error is defined by the return value.  
Get the error information by OnError event.

**Note**

---

**Sequence**

---

**Sample program code**

---

**Reference**

---

### 5.5.1.19. BoxOperation

**Class**                    **IPSAPI**

**Method**                **BoxOperation**

```
long    BoxOperation(
        long   id,
        long   command,
        long   color,
        long   size,
        long   xTopLeft,
        long   yTopLeft,
        long   xBottomRight,
        long   yBottomRight
        );
```

#### **Description**

Display frame lines on the video image.

#### **Argument**

id	1 to 9	ID for management ID is shared by BoxOperation and BoxOperationEx.
command	0 : Non display 1 : Solid line 2 : Dotted line 3 : Painting out	Type of frame
color	0 to 16777215	Frame color The sum of R (Red), G (Green) and B (Blue) R, G and B are specified with the following value.  R :        0   -        255    (0x000000 – 0x0000FF) G :       256   -       65280   (0x000100 – 0x00FF00) B :    65536   -    16711680   (0x010000 – 0xFF0000)
size	1 to 5	Width of frame line
xTopLeft	0 and over	Specify the xposition for top-left corner of frame.
yTopLeft	0 and over	Specify the yposition for top-left corner of frame.
xBottomRight	0 and over	Specify the xposition for bottom-right corner of frame.
yBottomRight	0 and over	Specify the yposition for bottom-right corner of frame.

---

**Return value**

---

0	Success
Except 0	Error code

---

**Error**

---

Error is defined by the return value.  
Get the error information by OnError event.

---

**Note**

---

Please specify xTopLeft, yTopLeft, xBottomRight and yBottomRight based on the display size by "5.5.2.11 ImageHeight" and "5.5.2.12 ImageWidth". Cannot display the frame lines out of drawing area.

The frame lines is not displayed on the image gotten by OnImage.

---

**Sequence**

---

---

**Sample program code**

---

---

**Reference**

---



### 5.5.1.20. BoxOperationEx

**Class**                      **IPSAPI**

**Method**                    **BoxOperationEx**

```
long    BoxOperationEx(
        long   id,
        long   command,
        long   color,
        long   size,
        long   xTopLeft,
        long   yTopLeft,
        long   xBottomRight,
        long   yBottomRight,
        long   transmissivity
    );
```

#### **Description**

Display frame lines with transmissivity on the video image.

#### **Argument**

id	1 to 9	ID for management ID is shared by BoxOperation and BoxOperationEx.
command	0 : Non display 1 : Solid line 2 : Dotted line 3 : Painting out	Type of frame
color	0 to 16777215	Frame color The sum of R (Red), G (Green) and B (Blue) R, G and B are specified with the following value. <div style="margin-left: 40px;">                         R :        0   -        255    (0x000000 – 0x0000FF)                          G :       256   -       65280   (0x000100 – 0x00FF00)                          B :      65536   -      16711680   (0x010000 – 0xFF0000)                     </div>
size	1 to 5	Width of frame line
xTopLeft	0 and over	Specify the xposition for top-left corner of frame.
yTopLeft	0 and over	Specify the yposition for top-left corner of frame.
xBottomRight	0 and over	Specify the xposition for bottom-right corner of frame.
yBottomRight	0 and over	Specify the yposition for bottom-right corner of frame.

transmissivity	0 to 255	Transmissivity of overlay frame lines. 0 (0x00) : transparent 127 (0x7F) : translucent 255 (0xFF) : opaque
----------------	----------	---

---

#### Return value

0	Success
Except 0	Error code

---

#### Error

Error is defined by the return value.  
Get the error information by OnError event.

---

#### Note

Please specify xTopLeft, yTopLeft, xBottomRight and yBottomRight based on the display size by "5.5.2.11 ImageHeight" and "5.5.2.12 ImageWidth". Cannot display the frame lines out of drawing area.

The frame lines is not displayed on the image gotten by OnImage.

---

#### Sequence

---

#### Sample program code

---

#### Reference

### 5.5.1.21. BitmapOperationEx

**Class**                **IPSAPI**

**Method**            **BitmapOperationEx**

```
long    BitmapOperationEx(
        long   id,
        long   command,
        char*   filename,
        long   xPosition,
        long   yPosition,
        long   maskColor,
        long   transmissivity
    );
```

#### Description

Display bitmap image with transmissivity on the video image.

#### Argument

id	1 to 4	ID for management
command	0 : Non display 1 : Display	
filename	Character strings (Half size character : 256 bytes or less)	Specify a file name and its full path for displaying on video image. The strings must be terminated with null character.
xPosition	0 and over	X position of displayed bitmap (top right corner).
yPosition	0 and over	Y position of displayed bitmap (top right corner)..
maskColor	-1 : No mask 0 to 16777215 : Mask	Mask color (undisplayed color) The sum of R (Red), G (Green) ad B (Blue) R, G and B are specified with the following value.  <div style="margin-left: 40px;"> R :        0   -        255    (0x000000 – 0x0000FF)  G :       256   -       65280   (0x000100 – 0x00FF00)  B :      65536   -      16711680   (0x010000 – 0xFF0000) </div> No mask is specified with “-1 (0xFFFFFFFF)”.
transmissivity	0 to 255	Transmissivity of overlay bitmap image. 0 (0x00) : transparent 127 (0x7F) : translucent 255 (0xFF) : opaque

---

**Return value**

---

0	Success
Except 0	Error code

---

**Error**

---

Error is defined by the return value.  
Get the error information by OnError event.

---

**Note**

---

Please specify xPosition and yPosition based on the display size by “5.5.2.11 ImageHeight” and “5.5.2.12 ImageWidth”. Cannot display the bitmap out of drawing area.  
The bitmap is not displayed on the image gotten by OnImage.

---

**Sequence**

---

---

**Sample program code**

---

---

**Reference**

---

#### 5.5.1.22. DigitalZoomMove

**Class**                **IPSAPI**

**Method**            **DigitalZoomMove**

```
long     DigitalZoomMove(  
                         long   xPosition,  
                         long   yPosition  
                         );
```

#### **Description**

Move the displayed area during working a digital zoom.

#### **Argument**

xPosition	Integer	Specify distance of x direction.
yPosition	Integer	Specify distance of y direction.

#### **Return value**

0	Success
Except 0	Error code

#### **Error**

Error is defined by the return value.  
Get the error information by OnError event.

**Note**

---

Please specify the direction of x direction and y direction by using pixel scale value of displayed image with digital zoom.

If specified position is out of original image, the displayed area moves to an edge. When reaches to the edge, error occurs.

When displaying the video image with black panel and executing this method, the position and width of black panel is kept.

**Sequence**

---

**Sample program code**

---

**Reference**

---

### 5.5.1.23. GetDigitalZoomPosition

**Class**                **IPSAPI**

---

**Method**              **GetDigitalZoomPosition**

---

long      GetDigitalZoomPosition(  
                                 long & xPosition,  
                                 long & yPosition  
                                 );

---

#### **Description**

Get the current position of the displayed area in the original image with digital zoom.  
Coordinate origin is the top-left corner position of original image, and the gotten value by this method is the top-left corner position of the displayed area.

---

#### **Argument**

xPosition	0 and over	Get position of x direction. (Pixel scale)
yPosition	0 and over	Get position of y direction. (Pixel scale)

---

#### **Return value**

0	Success
Except 0	Error code

---

#### **Error**

Error is defined by the return value.  
Get the error information by OnError event.

**Note**

---

**Sequence**

---

**Sample program code**

---

**Reference**

---



5.5.1.24. SetIntelligentView  
Not Supported

5.5.1.25. GetIntelligentView  
Not Supported

5.5.1.26. SetIntelligentViewColor  
Not Supported

5.5.1.27. GetIntelligentViewColor  
Not Supported

5.5.1.28. SetIntelligentViewSize  
Not Supported

5.5.1.29. GetIntelligentViewSize  
Not Supported

5.5.1.30. SetIntelligentViewTrackTime  
Not Supported

5.5.1.31. GetIntelligentViewTrackTime  
Not Supported

### 5.5.1.32. MultiSyncPause

**Class**                **IPSAPI**

---

**Method**            **MultiSyncPause**

---

long        MultiSyncPause();

---

#### **Description**

When using PlayControl with multi screen, pause to synchronize with other screen.  
When using the device which supports StreamID mode and setting On to SID mode, this method doesn't need to be executed.

---

#### **Argument**

None

---

#### **Return value**

0                Success

Except 0       Error code

---

#### **Error**

Error is defined by the return value.  
Get the error information by OnError event.

---

**Note**

When using the device which supports StreamID mode and setting On to SID mode, this method doesn't need to be executed.

It is necessary to call MultiSyncTime method after executing this method.

If using multiple screen playback with shared UID without executing this method, PS-API works with unexpected behavior.

---

**Sequence**

---

---

**Sample program code**

---

---

**Reference**

---

#### 5.5.1.33. MultiSyncTime

**Class**                **IPSAPI**

**Method**             **MultiSyncTime**

```
long     MultiSyncTime(  
             char*   syncTime,  
             long   isDst  
             );
```

#### Description

When using PlayControl with multi screen, set the sync time to synchronize with other screen. When using the device which supports StreamID mode and setting On to SID mode, this method doesn't need to be executed.

#### Argument

syncTime	yyyy/mm/dd hh:mm:ss	Specify a time and date character string for the base to synchronize. The strings must be terminated with null character. If "NULL" or "Empty String" is specified, error occurs.
isDst	0:Normal 1:Summer time	Summer time information of the time that is specified to syncTime.

#### Return value

0	Success
Except 0	Error code

#### Error

Error is defined by the return value.  
Get the error information by OnError event.

---

**Note**

When using the device which supports StreamID mode and setting On to SID mode, this method doesn't need to be executed.

It is necessary to call MultiSyncPause method before executing this method.

If using multiple screen playback with shared UID without executing this method, PS-API works with unexpected behavior.

---

**Sequence**

---

---

**Sample program code**

---

---

**Reference**

---

#### 5.5.1.34. CamSnapShot

**Class**                      **IPSAPI**

**Method**                      **CamSnapShot**

long      CamSnapShot (  
                 long    channel,  
                 long    imageMode  
                 );

#### **Description**

Get the snapshot image from NW camera or Encoder, and display it.

#### **Argument**

channel	1 : NW camera, GXE100 1 to 3 : S8573, 1 to 4 : Encoder X8570, X8571 S8574	Specify the video channel It doesn't support network disk recorder and digital disk recorder
imageMode	0 : Fisheye image 1 : 320x240 or 320x180 2 : 640x480 or 640x360	Specify the mode of snapshot. The aspect ratio of the gotten image depends on the NW camera and encoder configuration.

#### **Return value**

0	Success
Except 0	Error code

#### **Error**

Error is defined by the return value.  
Get the error information by OnError event.

**Note**

---

The NW camera that can get fisheye images with “imageMode=0” is as follows.

\*SW458, SF448, SF438

When using mode “0” with NW camera or encoder other than above, the aspect ratio of the gotten image is the aspect ratio that is set for the live video transmission.

---

**Sequence**

---

---

**Sample program code**

---

---

**Reference**

---

### 5.5.1.35. SetCroppingRect

**Class**                **IPSAPI**

**Method**              **SetCroppingRect**

```
long     SetCroppingRect (
           long   id,
           long   ltX,
           long   ltY,
           long   rbX,
           long   rbY
           );
```

#### Description

Set the Cropping area from the panorama area.

When enable the "5.5.2.26 CroppingEnabled", this setting will be reflected in the video.

#### Argument

id	1 to 5:Cropping frame	Cropping area id
ltX	integer	Specify the top-left X position of Cropping area.
ltY	integer	Specify the top-left Y position of Cropping area.
rbX	integer	Specify the bottom-right X position of Cropping area.
rbY	integer	Specify the bottom-right Y position of Cropping area.
* Negative value can be set on XY position, but there is no images in the negative value area, so the area will be back ground color.		

#### Return value

0	Success
Except 0	Error code

#### Error

Error is defined by the return value.  
Get the error information by OnError event.



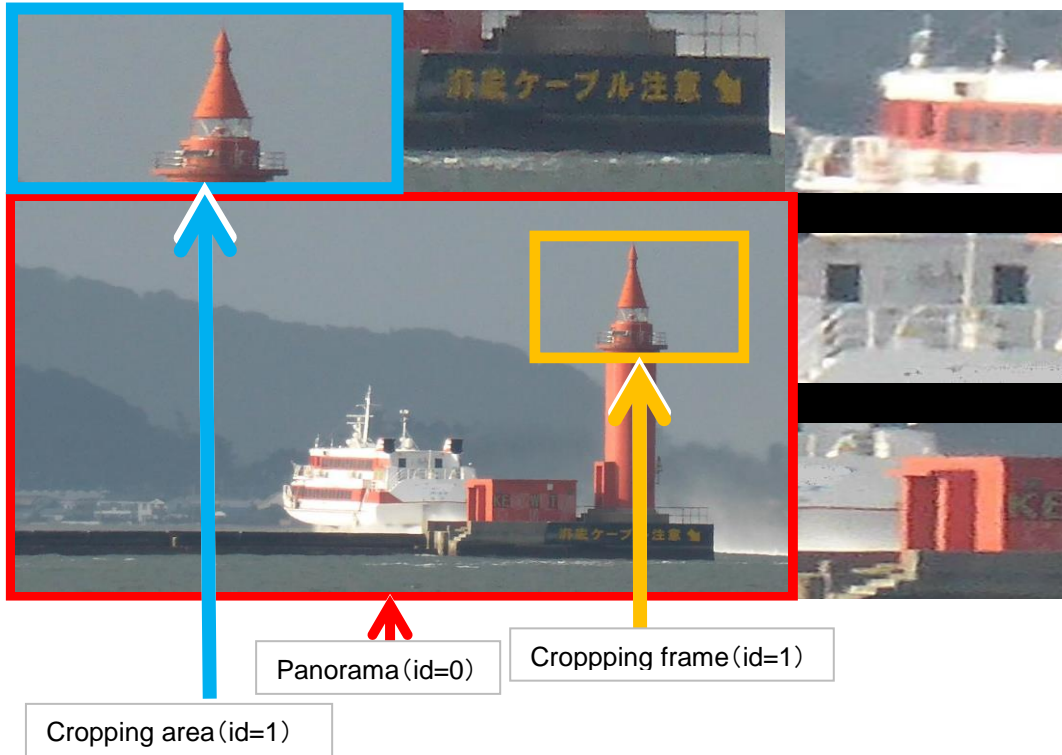
## Note

The coordinates of the Cropping area should be specified the position in the panorama area which is set by "5.5.1.37 SetCroppingDrawRect" method, and the coordinate is based on the display area which is set by "5.5.2.11 ImageHeight" and "5.5.2.12 ImageWidth" property.

When the coordinates is specified out of "id=0 area(panorama)" range, it will be displayed the back ground color on the screen.

The bottom-right position value should be bigger than top-left position value.

If need to display the frame line, please set the frame position by "5.5.1.41 SetCroppingMarker".



## Sequence

6.16 Cropping

## Sample program code

## Reference

### 5.5.1.36. GetCroppingRect

**Class**                **IPSAPI**

**Method**              **GetCroppingRect**

```
long    GetCroppingRect (
        long   id,
        long&  ltX,
        long&  ltY,
        long&  rbX,
        long&  rbY
    );
```

#### **Description**

Get the Cropping frame that has been set by the "5.5.1.35 SetCroppingRect".

#### **Argument**

id	1 to 5:Cropping frame	Cropping area id
ltX	integer	Get the top-left X position of Cropping area.
ltY	integer	Get the top-left Y position of Cropping area.
rbX	integer	Get the bottom-right X position of Cropping area.
rbY	integer	Get the bottom-right Y position of Cropping area.

#### **Return value**

0	Success
Except 0	Error code

#### **Error**

Error is defined by the return value.  
Get the error information by OnError event.

**Note**

---

The coordinates of Cropping frame is based on display size of “5.5.2.11 ImageHeight” and “5.5.2.12 ImageWidth”.

**Sequence**

---

**Sample program code**

---

**Reference**

---

### 5.5.1.37. SetCroppingDrawRect

**Class**                **IPSAPI**

**Method**              **SetCroppingDrawRect**

```
long     SetCroppingDrawRect (
           long   id,
           long   ltX,
           long   ltY,
           long   rbX,
           long   rbY
           );
```

#### **Description**

Set the drawing position of the panorama image and the cropping images which are specified by "5.5.1.35 SetCroppingRect".

When enable the "5.5.2.26 CroppingEnabled", this setting will be reflected in the video.

#### **Argument**

id	0 :Panorama image 1 to 5:Cropping frame	Cropping area id
ltX	integer	Specify the top-left X position of Cropping area.
ltY	integer	Specify the top-left Y position of Cropping area.
rbX	integer	Specify the bottom-right X position of Cropping area.
rbY	integer	Specify the bottom-right Y position of Cropping area.
* Negative value can be set on XY position, but the negative value area will not be displayed.		

#### **Return value**

0	Success
Except 0	Error code

#### **Error**

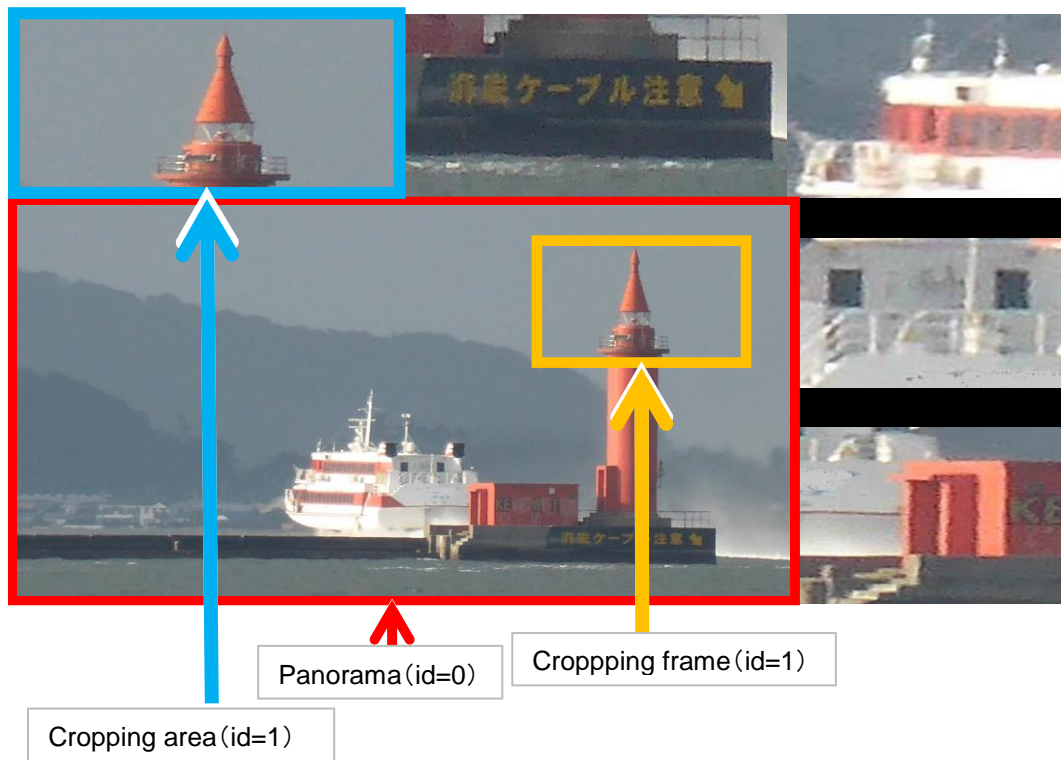
Error is defined by the return value.  
Get the error information by OnError event.

## Note

The coordinates of the Cropping area should be specified the position which is based on the display area which is set by “5.5.2.11 ImageHeight” and “5.5.2.12 ImageWidth” property.

If the drawing position is overlap, the video of the large value of id is displayed in the front.

The bottom-right position value should be bigger than top-left position value.



## Sequence

6.16 Cropping

## Sample program code

## Reference

### 5.5.1.38. GetCroppingDrawRect

**Class**                **IPSAPI**

**Method**            **GetCroppingDrawRect**

```
long    GetCroppingDrawRect (
        long   id,
        long&   ltX,
        long&   ltY,
        long&   rbX,
        long&   rbY
    );
```

#### **Description**

Get the drawing position that has been set by the "5.5.1.37 SetCroppingDrawRect".

#### **Argument**

id	0 :Panorama image 1 to 5:Cropping frame	Cropping area id
ltX	integer	Get the top-left X position of Cropping area.
ltY	integer	Get the top-left Y position of Cropping area.
rbX	integer	Get the bottom-right X position of Cropping area.
rbY	integer	Get the bottom-right Y position of Cropping area.

#### **Return value**

0	Success
Except 0	Error code

#### **Error**

Error is defined by the return value.  
Get the error information by OnError event.

**Note**

---

The coordinates of drawing position is based on display size of “5.5.2.11 ImageHeight” and “5.5.2.12 ImageWidth”.

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.5.1.39. SetCroppingDrawEnabled

**Class**                **IPSAPI**

---

**Method**                **SetCroppingDrawEnabled**

---

long        SetCroppingDrawEnabled (  
                                 long id,  
                                 long mode  
                                 );

---

#### **Description**

Set the "Hide" or "Show" mode for each drawing area which is specified with "5.5.1.37 SetCroppingDrawRect".

When enable the "5.5.2.26 CroppingEnabled", this setting will be reflected in the video.

---

#### **Argument**

id	0 : Panorama image 1 to 5: Cropping frame	Cropping area id
mode	0 : Hide 1 : Show	

---

#### **Return value**

0	Success
Except 0	Error code

---

#### **Error**

Error is defined by the return value.  
Get the error information by OnError event.



**Note**

---

**Sequence**

---

6.16 Cropping

---

**Sample program code**

---

**Reference**

---

#### 5.5.1.40. GetCroppingDrawEnabled

**Class**                **IPSAPI**

**Method**            **GetCroppingDrawEnabled**

long      GetCroppingDrawEnabled (  
                                 long id,  
                                 long& mode  
                                 );

#### **Description**

Get the "Hide" or "Show" mode for each drawing area which is specified with "5.5.1.39 SetCroppingDrawEnabled".

#### **Argument**

id	0 : Panorama image 1 to 5: Cropping frame	Cropping area id
mode	0 : Hide 1 : Show	

#### **Return value**

0	Success
Except 0	Error code

#### **Error**

Error is defined by the return value.  
Get the error information by OnError event.

**Note**

---

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.5.1.41. SetCroppingMarker

**Class**            **IPSAPI**

**Method**           **SetCroppingMarker**

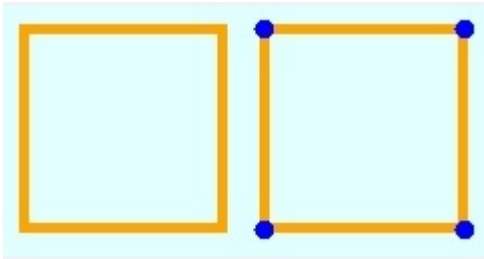
```
long    SetCroppingMarker (
        long    id,
        long    mode
        long    ltX,
        long    ltX,
        long    rbX,
        long    rbY,
        long    lineSize,
        long    lineColor,
        long    ellipseSize,
        long    ellipseColor
    );
```

#### **Description**

Set the color, size and style of the frame for showing the Cropping area.

When enable the "5.5.2.26 CroppingEnabled", this setting will be reflected in the video.

## Argument

id	1 to 5:Cropping frame	Cropping area id
mode	0 : Not display the frame 1 : Display the frame without dot on the corner. 2 : Display the frame with dot on the corner.	<div></div> <div>mode=1mode=2</div>
ltX	integer	Specify the top-left X position of Cropping area.
ltY	integer	Specify the top-left Y position of Cropping area.
rbX	integer	Specify the bottom-right X position of Cropping area.
rbY	integer	Specify the bottom-right Y position of Cropping area.
		* Negative value can be set on XY position, but the negative value area will not be displayed.
lineSize	1 to 10	Width of frame line
lineColor	0 to 16777215	Frame color The sum of R (Red), G (Green) and B (Blue) R, G and B are specified with the following value. R : 0 - 255 (0x000000 - 0x0000FF) G : 256 - 65280 (0x000100 - 0x00FF00) B : 65536 - 1671168 (0x010000 - 0xFF0000)
ellipseSize	1 to 10	Width of vertex Circle
ellipseColor	0 to 16777215	Circle color The sum of R (Red), G (Green) and B (Blue) R, G and B are specified with the following value. R : 0 - 255 (0x000000 - 0x0000FF) G : 256 - 65280 (0x000100 - 0x00FF00) B : 65536 - 1671168 (0x010000 - 0xFF0000)

---

**Return value**

0	Success
Except 0	Error code

---

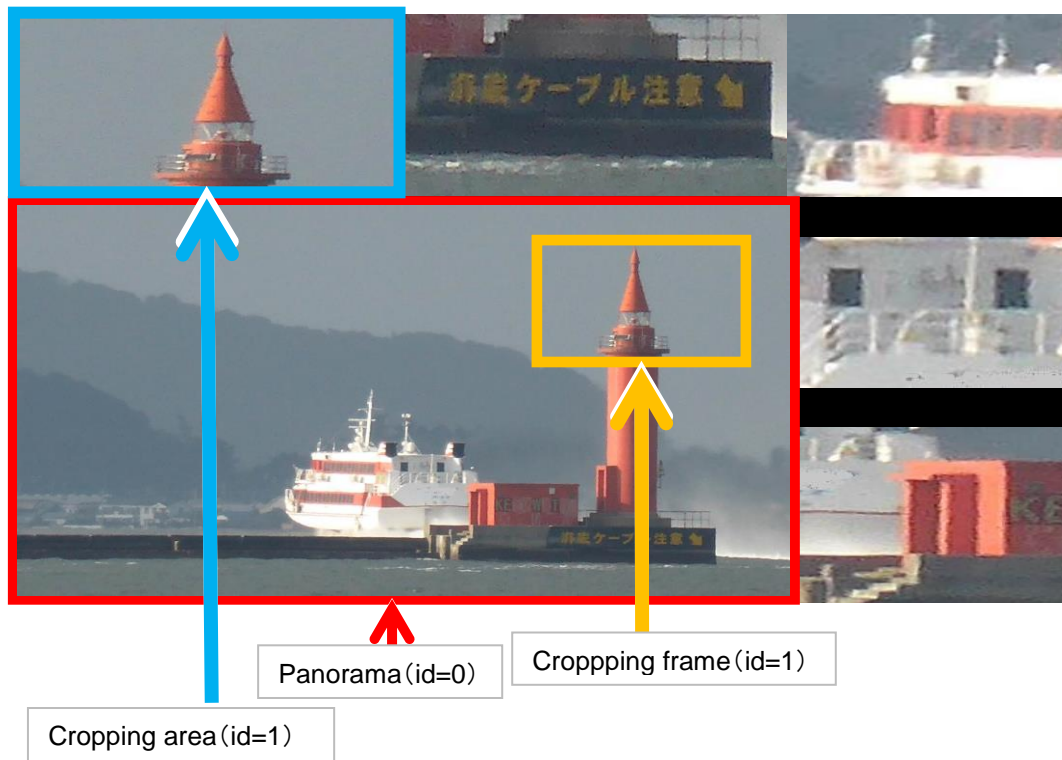
**Error**

Error is defined by the return value.  
Get the error information by OnError event.

---

**Note**

The coordinates of the Cropping frame should be specified the position which is based on the display area.



---

**Sequence**

6.16 Cropping

---

**Sample program code**

---

**Reference**

---

#### 5.5.1.42. GetCroppingMarker

**Class**            **IPSAPI**

---

**Method**           **GetCroppingMarker**

---

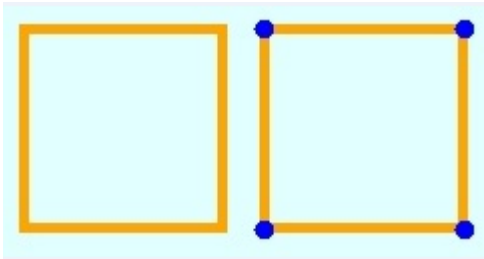
long    GetCroppingMarker (  
                 long   id,  
                 long&   mode  
                 long&   ltX,  
                 long&   ltX,  
                 long&   rbX,  
                 long&   rbY,  
                 long&   lineSize,  
                 long&   lineColor,  
                 long&   ellipseSize,  
                 long&   ellipseColor  
                 );

---

**Description**

Get the the color ,size and style in the line of Cropping frame that has been set by "5.5.1.41 SetCroppingMarker".

## Argument

id	1 to 5:Cropping frame	Cropping area id
mode	0 : Not display the frame 1 : Display the frame without dot on the corner. 2 : Display the frame with dot on the corner.	<div></div> <div>mode=1mode=2</div>
ltX	integer	Get the top-left X position of Cropping area.
ltY	integer	Get the top-left Y position of Cropping area.
rbX	integer	Get the bottom-right X position of Cropping area.
rbY	integer	Get the bottom-right Y position of Cropping area.
		* Negative value can be set on XY position, but the negative value area will not be displayed.
lineSize	1 to 10	Width of frame line
lineColor	0 to 16777215	Frame color The sum of R (Red), G (Green) and B (Blue) R, G and B are specified with the following value. R : 0 - 255 (0x000000 - 0x0000FF) G : 256 - 65280 (0x000100 - 0x00FF00) B : 65536 - 1671168 (0x010000 - 0xFF0000)
ellipseSize	1 to 10	Width of vertex Circle
ellipseColor	0 to 16777215	Circle color The sum of R (Red), G (Green) and B (Blue) R, G and B are specified with the following value. R : 0 - 255 (0x000000 - 0x0000FF) G : 256 - 65280 (0x000100 - 0x00FF00) B : 65536 - 1671168 (0x010000 - 0xFF0000)



---

**Return value**

0	Success
Except 0	Error code

---

**Error**

Error is defined by the return value.  
Get the error information by OnError event.

---

**Note**

The drawing position of Cropping frame is based on display size of “5.5.2.11 ImageHeight” and “5.5.2.12 ImageWidth”.

---

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.5.1.43. HttpMP4Download

**Class** **IPSAPI**

**Method** **HttpMP4Download**

```
long    HttpMP4Download (
        long    channel,
        long    command,
        char*   startTimeDate,
        char*   endTimeDate,
        long    audiomode,
        char*   fileName
    );
```

#### Description

Download MP4 file from NX Series via HTTP.

#### Argument

channel	1 to 4 : NX100 1 to 32 : NX200, NX300 1 to 128 : NX400	Specify the video channel.
command	0 : Stop MP4 file download 1 : Start MP4 file download	Specify Stop MP4 file download / Start MP4 file download.
startTimeDate	yyyy/mm/dd hh:mm:ss	Download the recording data that started recording after the specified NX Series local time. The strings must be terminated with null character.
endTimeDate	yyyy/mm/dd hh:mm:ss	Download the recording data that start recording before the specified NX Series local time. Please specify the duration between startTimeDate and endTimeDate within 60 minutes. The strings must be terminated with null character.
audioMode	0 : video data only 1: video data and audio data	
fileName	Character strings (221 byte or less)	Specify a complete file path, and file name without extension for storing the downloaded video. The start date and time of the recorded data acquired from the NX Series is also added to the file name. An extension(mp4) is added automatically. The strings must be terminated with null character.

---

**Return value**

---

0	Success
Except 0	Error code

---

**Error**

---

Error is defined by the return value.  
Get the error information by OnError event.

---

**Note**

---

NWDR other than NX Series does not have MP4 download function.  
Depending on the NX Series specification, it may be divided into multiple MP4 files.  
(Example :If the video format, resolution or recording type is different, the file will be split)  
Depending on the NX Series specification, JPEG will be excluded from download.  
While downloading, a ".tmp" file is created in the specified folder, but until downloading is completed do not manually delete ".tmp" file.

---

**Sequence**

---

---

**Sample program code**

---

---

**Reference**

---

#### 5.5.1.44. HttpDownload

**Class**                      **IPSAPI**

**Method**                    **HttpDownload**

```
long      HttpDownload (
           long      channel,
           long      command,
           char*     startTimeDate,
           long      isDstSt,
           char*     endTimeDate,
           long      isDstEt,
           long      dataType,
           char*     fileName
           );
```

#### Description

Download MP4 / n3r / n3a / n3n file from NX Series via HTTP.

#### Argument

channel	1 to 4 : NX100 1 to 32 : NX200, NX300 1 to 128 : NX400	Specify the video channel.
command	0 : Stop download 1 : Start download	Specify Stop file download / Start file download.
startTimeDate	yyyy/mm/dd hh:mm:ss	Download the recording data that started recording after the specified NX Series local time. The strings must be terminated with null character.
isDstSt	0:Normal 1:Summer time	Summer time information of the time that is specified to startTimeDate.
endTimeDate	yyyy/mm/dd hh:mm:ss	Download the recording data that start recording before the specified NX Series local time. Please specify the duration between startTimeDate and endTimeDate within 60 minutes. The strings must be terminated with null character.
isDstEt	0:Normal 1:Summer time	Summer time information of the time that is specified to endTimeDate.
dataType	1: MP4 (video only ) 2: MP4 (video and audio ) 3: n3r (video only ) 4: n3r +n3a (video and audio ) 5: n3n (Recording event information)	

fileName	Character strings (221 byte or less)	Specify a complete file path, and file name without extension for storing the downloaded video. The start date and time of the recorded data acquired from the NX Series is also added to the file name. An extension is added automatically. The strings must be terminated with null character.
----------	---	--

---

#### Return value

0	Success
Except 0	Error code

---

#### Error

Error is defined by the return value.  
Get the error information by OnError event.

---

#### Note

NWDR other than NX Series does not have HTTP download function.

Depending on the NX Series specification, it may be divided into multiple files.

(Example :If the video format, resolution or recording type is different, the file will be split)

In case of datatype=1 or 2 (MP4) , JPEG will be excluded from download.

While downloading, a ".tmp" file is created in the specified folder, but until downloading is completed do not manually delete ".tmp" file.

---

#### Sequence

---

#### Sample program code

---

#### Reference

#### 5.5.1.45. GetMP4DownloadStatus

**Class**                **IPSAPI**

**Method**             **GetMP4DownloadStatus**

long      GetMP4DownloadStatus();

---

#### **Description**

Get current file downloading status when using HttpMP4Download and HttpDownload method.

---

#### **Argument**

None

---

#### **Return value**

-1	Fail to get status.
0	No getting file via HTTP.
1	In getting file via HTTP.

---

#### **Error**

Get the error information by OnError event.

---

#### **Note**

---

#### **Sequence**

---

#### **Sample program code**

---

#### **Reference**

#### 5.5.1.46. GetMP4DownloadTransRate

**Class**                    **IPSAPI**

**Method**                **GetMP4DownloadTransRate**

long      GetMP4DownloadTransRate();

---

#### **Description**

Get current transmission speed [byte/s] of file downloading when using HttpMP4Download and HttpDownload method.

---

#### **Argument**

None

---

#### **Return value**

-1	Fail to get status.
0 or more	Transfer rate [byte/s]

---

#### **Error**

Get the error information by OnError event.

---

#### **Note**

---

#### **Sequence**

---

#### **Sample program code**

---

#### **Reference**

### 5.5.2. Property

#### 5.5.2.1. MPEG4Port

**Class**                **IPSAPI**

**Property**           **MPEG4Port**

long      SetMPEG4Port(  
                                 long port  
                                 );

long      GetMPEG4Port();

#### **Description**

Set a UDP port number to receive MPEG-4 video stream into PS-API.

Get a UDP port number to receive MPEG-4 video stream from PS-API.

#### **Argument for SET**

port	The even number. Valid range : 1024 to 65534	UDP port number for MPEG-4 video communication.  Default value is 12500.
------	---	---

#### **Return value for SET**

0	Success
Except 0	Error code

#### **Return value for GET**

Get a UDP port number for MPEG-4 video communication.

#### **Error**



**Note**

---

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.5.2.2. H264Port

**Class**            **IPSAPI**

**Property**        **H264Port**

long      SetH264Port(  
                         long   port  
                         );

long      Get H264Port();

#### **Description**

Set a UDP port number to receive H.264, H.265 video stream into PS-API.

Get a UDP port number to receive H.264, H.265 video stream from PS-API.

#### **Argument for SET**

port	The even number. Valid range : 1024 to 65534	UDP port number for H.264, H.265 video communication.
------	---	---

The default value is 12500.

#### **Return value for SET**

0	Success
---	---------

Except 0	Error code
----------	------------

#### **Return value for GET**

Get a UDP port number for H.264, H.265 video communication.

#### **Error**

**Note**

---

**Sequence**

---

**Sample program code**

---

**Reference**

---

### 5.5.2.3. RtpPortMode

**Class**            **IPSAPI**

**Property**        **RtpPortMode**

long      SetRtpPortMode(  
                                 long   mode  
                                 );

long      GetRtpPortMode();

#### **Description**

Set mode whether use a fixed port to receive MPEG-4/H.264/H.265 stream or not into PS-API.  
Get mode whether use a fixed port to receive MPEG-4/H.264/H.265 stream or not from PS-API.

#### **Argument for SET**

mode	0 : Disable	Specify the fixed port mode.  The default value is 0.
	1 : Enable (Use fixed port)	

#### **Return value for SET**

0	Success
Except 0	Error code

#### **Return value for GET**

Get mode whether use a fixed port to receive MPEG-4/H.264/H.265 stream or not

#### **Error**

**Note**

---

When RtpPortMode is set to "1", MPEG-4/H.264/H.265 stream is received at the port which is specified in MPEG4Port/H264Port.

When RtpPortMode is set to "1" and set the range of searchable port in RtpPortRange, it can limit a port range to use for the MPEG-4/H.264/H.265 stream reception.

The audio receiving port is MPEG-4/H.264/H.265 receiving port +1000.

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.5.2.4. RtpPortRange

**Class**            **IPSAPI**

---

**Property**        **RtpPortRange**

---

long      SetRtpPortRange(  
                                 long   range  
                                 );

long      GetRtpPortRange();

---

#### **Description**

Set available port range with a fixed MPEG-4/H.264/H.265 port mode into PS-API.  
Get available port range with a fixed MPEG-4/H.264/H.265 port mode from PS-API.  
\*Search available port from "MPEG4Port/H.264Port" to "MPEG4Port/H.264Port + RtpPortRange".

---

#### **Argument for SET**

range	0 - 65534	Available port range The default value is 0.
-------	-----------	---

---

#### **Return value for SET**

0	Success
Except 0	Error code

---

#### **Return value for GET**

Get available port range with a fixed MPEG-4/H.264/H.265 port mode

---

#### **Error**

---

**Note**

---

When RtpPortMode is set to “1” and set the range of searchable port in RtpPortRange, it can limit a port range to use for the MPEG-4/H.264/H.265 stream reception.  
The audio receiving port is MPEG-4/H.264/H.265 receiving port +1000.

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.5.2.5. MulticastAddr

**Class**                **IPSAPI**

**Property**           **MulticastAddr**

```
long      SetMulticastAddr(
                                char*  ipAddr
                                );

void      GetMulticastAddr(
                                char*  ipAddr,
                                long    size
                                );
```

#### Description

Set a multicast address to receive MPEG-4/H.264/H.265 video stream into PS-API.  
 Get a multicast address to receive MPEG-4/H.264/H.265 video stream from PS-API.  
 MulticastAddr setting is valid when MulticastAutoConf is set to "0".

**\* Only IPv4 address is acceptable.**

#### Argument for SET

ipAddr	Character Strings (255 bytes or less)	IPv4 multicast IP address for a specified video stream. The strings must be terminated with null character. e.g. 239.192.0.20  Default value is NULL.
--------	--	---

#### Argument for GET

ipAddr	Character strings	Get the multicast IP address for MPEG-4/H.264/H.265 video stream. Application needs to allocate and deallocate memory. If getting value failed, it is set to NULL.
size	Buffer size of ipAddr	Specify the buffer size for ipAddr. (Byte)

#### Return value for SET

0	Success
Except 0	Error code

#### Error



**Note**

---

When 0 is added at the top of segment like as 192.168.000.010, it is treated with octal number.  
When port number is written after : (colon) like as 192.168.0.10:8080, it is ignored and the port number follows the setting.

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.5.2.6. MPEG4Resolution

**Class**                      **IPSAPI**

---

**Property**                      **MPEG4Resolution**

---

long      SetMPEG4Resolution(  
   long   resolution  
   );

long      GetMPEG4Resolution();

---

#### Description

Set a MPEG-4 video image resolution into PS-API.

It is necessary that MPEG-4Resolution is same value of the device setting.

Get a MPEG-4 video image resolution from PS-API.

---

#### Argument for SET

resolution	320 : 320x240 (QVGA) 640 : 640x480 (VGA) 720 : 720x480(D1:NTSC) 720x576(D1:PAL) 960 : 960x720	Specify a MPEG-4 image resolution for a retrieving video image.  Default value is    640.
------------	---	--

---

#### Return value for SET

0	Success
---	---------

Except 0	Error code
----------	------------

---

#### Return value for GET

Get a retrieving video image resolution.

---

#### Error

---

---

**Note**

If the resolution of a received image is different from the specified resolution, this property is not updated. Then this property is not same as the value gotten by GetImageResolution method.

---

**Sequence**

6.1 PlayLive

---

**Sample program code**

[Visual C++ 2012] ..¥Sample Program¥PS-API¥Visual C++ 2012¥001\_PlayLive

---

**Reference**

---

#### 5.5.2.7. H264Resolution

**Class**                      **IPSAPI**

**Property**                      **H264Resolution**

long      SetH264Resolution(  
   long   resolution  
   );

long      GetH264Resolution();

#### Description

Set a H.264/H.265 video image resolution into PS-API.

It is necessary that JPEGResolution is same value of the device setting.

Get a H.264/H.265 video image resolution from PS-API.

#### Argument for SET

resolution	160 : 160x120 (4:3) 160x90 (16:9) 320 : 320x240 (QVGA) 320x180(16:9) 320x320(1:1) 180x320(9:16) 400 : 400x300 (4:3) 640 : 640x480 (VGA) 640x360(16:9) 640x640(1:1) 360x640 (9:16) 720 : 720x480(D1:NTSC) 720x576(D1:PAL) 800 : 800x600 960 : 960x720 1280 : 1280x960 (4:3) 1280x720 (16:9) 1280x1280 (1:1) 720x1280 (9:16) 1600 : 1600x1200 (4:3) 1920 : 1920x1080(16:9) 1080x1920 (9:16) 2048 : 2048x1536 (4:3) 2048x2048 (1:1) 2192: 2192x2192 (1:1) 2560 : 2560x1920 (4:3) 2560x1440 (16:9) 2688 : 2688x1520(16:9) 2816 : 2816x2816 (1:1) 2992 : 2992x2992 (1:1) 3072 : 3072x1728 (16:9) 3072x2304 (4:3) 1728x3072 (9:16)	Specify a H.264/H.265 image resolution for a retrieving video image.  Default value is 640.
------------	--	--

---

**Argument for SET**

---

3328 : 3328x1872(16:9)  
1872x3328(9:16)  
3840 : 3840x2160(16:9)  
2160x3840(9:16)  
4000 : 4000x3000 (4:3)

---

**Return value for SET**

---

0	Success
Except 0	Error code

---

**Return value for GET**

---

Get a H.264/H.265 image resolution for a retrieving video image.

---

**Error**

---

---

**Note**

---

If the resolution of a received image is different from the specified resolution, this property is not updated. Then this property is not same as the value gotten by GetImageResolution method.

---

**Sequence**

---

6.1 PlayLive

---

**Sample program code**

---

[Visual C++ 2012] ..¥Sample Program¥PS-API¥Visual C++ 2012¥001\_PlayLive

---

**Reference**

---

### 5.5.2.8. JPEGResolution

**Class**                      **IPSAPI**

**Property**                      **JPEGResolution**

long      SetJPEGResolution(  
   long   resolution  
   );

long      GetJPEGResolution();

### Description

Set a JPEG video image resolution into PS-API.

It is necessary that JPEGResolution is same value of the device setting.

Get a JPEG video image resolution from PS-API.

### Argument for SET

resolution	160 : 160x120 (4:3) 160x90 (16:9) 320 : 320x240 (QVGA) 320x180(16:9) 320x320(1:1) 180x320(9:16) 400 : 400x300 (4:3) 640 : 640x480 (VGA) 640x360(16:9) 640x640(1:1) 360x640 (9:16) 720 : 720x480(D1:NTSC) 720x576(D1:PAL) 800 : 800x600 960 : 960x720 1280 : 1280x960(4:3) 1280x720 (16:9) 1280x1280 (1:1) 720x1280 (9:16) 1600 : 1600x1200 (4:3) 1920 : 1920x1080(16:9) 1080x1920 (9:16) 2048 : 2048x1536(4:3) 2048x2048 (1:1) 2192: 2192x2192 (1:1) 2560 : 2560x1920(4:3) 2560x1440 (16:9) 2688 : 2688x1520(16:9) 2816 : 2816x2816(1:1) 2992 : 2992x2992(1:1) 3072 : 3072x1728 (16:9) 3072x2304 (4:3) 1728x3072 (9:16)	Specify a JPEG image resolution for a retrieving video image.  Default value is    640.
------------	---	---

---

**Argument for SET**

resolution	3328 : 3328x1872(16:9) 1872x3328(9:16) 3840 : 3840x2160(16:9) 2160x3840(9:16) 4000 : 4000x3000 (4:3)
------------	--

---

**Return value for SET**

0	Success
Except 0	Error code

---

**Return value for GET**

Get a JPEG image resolution for a retrieving video image.

---

**Error**

---

**Note**

---

**Sequence**

6.1 PlayLive

---

**Sample program code**

[Visual C++ 2012] ..¥Sample Program¥PS-API¥Visual C++ 2012¥001\_PlayLive

---

**Reference**

---

#### 5.5.2.9. StreamFormat

**Class**            **IPSAPI**

---

**Property**        **StreamFormat**

---

long      SetStreamFormat(  
                                 long   format  
                                 );

long      GetStreamFormat();

---

#### **Description**

Set image format of video stream into PS-API.  
Get image format of video stream from PS-API.

---

#### **Argument for SET**

format	0 : JPEG	Specify a video stream format for a displaying video image.
	1 : MPEG-4	
	2 : HD300	Default value is 0.
	3 : H.264	
	6 : H.265	

---

#### **Return value for SET**

0	Success
---	---------

Except 0	Error code
----------	------------

---

#### **Return value for GET**

Get a video stream format for a displaying video image.

---

#### **Error**

---



**Note**

---

In case of PlayLive/Play/PlayFile or DecodeImage with NWDR, NX Series PS-API gets stream format from the received video image and set StreamFormat property automatically.

When "6: H.265" is set as StreamFormat, the following properties are used for H.265 instead of H.264.

- H264Port
- H264Resolution

**Sequence**

---

6.1 PlayLive

**Sample program code**

---

[Visual C++ 2012] ..¥Sample Program¥PS-API¥Visual C++ 2012¥001\_PlayLive

**Reference**

---

#### 5.5.2.10. FilePassword

**Class**            **IPSAPI**

**Property**        **FilePassword**

```
long     SetFilePassword(
           char*   password
           );

void     GetFilePassword(
           char* password,
           long   size
           );
```

#### Description

Set a password to playback the downloaded video file which has a password into PS-API.  
Get a password to playback the downloaded video file which has a password from PS-API.

#### Argument for SET

password	Character Strings (15 bytes or less)	Password of the video data file. The strings must be terminated with null character.  Default value is   NULL.
----------	---	---

#### Argument for GET

password	Character strings	Get password of the video data file. Application needs to allocate and deallocate memory. If getting value failed, it is set to NULL.
size	Buffer size of password	Specify the buffer size for password. (Byte)

#### Return value for SET

0	Success
Except 0	Error code

#### Error

## Note

---

## Sequence

---

6.3 PlayFile

## Sample program code

---

[Visual C++ 2012] ..¥Sample Program¥PS-API¥Visual C++ 2012¥003\_PlayFile

## Reference

---

#### 5.5.2.11. ImageHeight

**Class**            **IPSAPI**

**Property**        **ImageHeight**

long      SetImageHeight(  
                                 long   height  
                                 );

long      GetImageHeight();

#### **Description**

Set into PS-API the vertical pixels of the video image, that is displayed on the window set by SetVideoWindow.

Get from PS-API the vertical pixels of the video image.

#### **Argument for SET**

height	Greater than 0	Specify vertical pixels for a displaying video image.
--------	----------------	---

The default value is 480.

#### **Return value for SET**

0	Success
---	---------

Except 0	Error code
----------	------------

#### **Return value for GET**

Get vertical pixels for a displaying video image.

#### **Error**

**Note**

---

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.5.2.12. ImageWidth

**Class**            **IPSAPI**

**Property**        **ImageWidth**

long      SetImageWidth(  
                                 long   width  
                                 );

long      GetImageWidth();

#### **Description**

Set into PS-API the horizontal pixels of the video image, that is displayed on the window set by SetVideoWindow.

Get from PS-API the horizontal pixels of the video image.

#### **Argument for SET**

width	Greater than 0	Specify horizontal pixels for a displaying video image.
-------	----------------	---

The default value is 640.

#### **Return value for SET**

0	Success
---	---------

Except 0	Error code
----------	------------

#### **Return value for GET**

Get horizontal pixels for a displaying video image.

#### **Error**

**Note**

---

**Sequence**

---

**Sample program code**

---

**Reference**

---

### 5.5.2.13. MulticastAutoConf

**Class**            **IPSAPI**

**Property**        **MulticastAutoConf**

long      SetMulticastAutoConf(  
   long    autoconf  
   );

long      GetMulticastAutoConf();

#### **Description**

Set the multicast setting mode into PS-API whether use the multicast setting gotten from a target device or not.

Get the multicast setting mode from PS-API whether use the multicast setting gotten from a target device or not.

#### **Argument for SET**

autoconf	0 : Not use auto setting 1 : Use auto setting	Specify the multicast setting mode whether get multicast address and port number from a target device or not.
----------	--	---

The default value is 0.

#### **Return value for SET**

0	Success
---	---------

Except 0	Error code
----------	------------

#### **Return value for GET**

Get the stream number from PS-API.

#### **Error**



**Note**

---

MulticastAutoConf setting is valid when using MPEG-4/H.264/H.265 with camera and encoder. When the camera setting or encoder setting is unicast mode, MulticastAutoConf setting is ignored and PS-API works with unicast mode.

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.5.2.14. StreamNumber

**Class**            **IPSAPI**

**Property**        **StreamNumber**

long      SetStreamNumber(  
                                 long no  
                                 );

long      GetStreamNumber ();

#### **Description**

Set the stream No. for multi streaming device.  
Get the stream No. from PS-API.

#### **Argument for SET**

no	1: MPEG-4/H.264/H.265 stream 1	Stream number
	2: MPEG-4/H.264/H.265 stream 2	Default value is 1.
	3: H.264/H.265 stream 3	
	4: H.264/H.265 stream 4	

#### **Return value for SET**

0	Success
Except 0	Error code

#### **Return value for GET**

Get the stream number from PS-API.

#### **Error**

**Note**

---

When setting InternetMode property to “1 (using HTTP)” and receiving live video image with a device which doesn’t support 2<sup>nd</sup> stream of MPEG-4/H.264, this parameter has to be set to “1”.

When setting InternetMode property to “0 (not using HTTP)” and receiving live video image with a device which doesn’t support 2<sup>nd</sup> stream of MPEG-4/H.264, this parameter is ignored.

When using 360-degree Network Camera, please refer to “4.7 About 360-degree Network Camera”.

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.5.2.15. NXStreamNumber

**Class**            **IPSAPI**

**Property**        **NXStreamNumber**

long      SetNXStreamNumber(  
                                 long   no  
                                 );

long      GetNXStreamNumber ();

#### **Description**

Set the NX Series stream No. of NX Series for multi streaming device.  
Get the stream No. of NX Series from PS-API.

#### **Argument for SET**

no	1: H.264/H.265 stream 1	Stream number
	2: H.264/H.265 stream 2	
		Default value is 1.

#### **Return value for SET**

0	Success
Except 0	Error code

#### **Return value for GET**

Get the stream number of NX Series from PS-API.

#### **Error**

**Note**

---

This property is valid when the “Play”, “Search”, “SearchEx” and “VMDSearchEx” method is executed.

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.5.2.16. InternetMode

Class	IPSAPI
Property	InternetMode
long	SetInternetMode (long mode);
long	GetInternetMode ();

### Description

Set the mode to receive MPEG-4/H.264/H.265 live via HTTP into PS-API.

Get the mode to receive MPEG-4/H.264/H.265 live via HTTP from PS-API.

## Argument for SET

mode	0 : Don't use HTTP for receiving MPEG-4/H.264/H.265 live. 1 : Use HTTP for receiving MPEG-4/H.264/H.265 live.	Mode to receive MPEG-4/H.264/H.265 live.  Default value is 0.
------	--	---

### Return value for SET

0	Success
Except 0	Error code

### Return value for GET

Get the mode to receive MPEG-4/H.264/H.265 live via HTTP from PS-API

## Error

## Note

The device which supports MPEG-4/H.264/H.265 transmission via HTTP is as follows.

		Live				Play
		JPEG	MPEG-4	H.264	H.265	
HD300		Not support				Not support
NWDR	ND200	Not support	Not support	Not support		Not support
	ND300	Not support	Not support	Not support		Not support
	ND400	Not support	Support	Support		Not support
	NV200	Not support	Support	Support		Not support
	NV250	Not support		Support		Not support
	NV300	Not support		Support		Not support
NW Camera *1		Not support	Support	Support	Support	
Encoder *1		Not support	Support	Support		
HD600/700				Support		Not support
NX Series	NX100	Not support		Support	Support	Not support
	NX200	Not support		Support	Support	Not support
	NX300	Not support		Support	Support	Not support
	NX400	Not support		Support	Support	Not support

\*1 : NP1000, NP304, NT304 and NT314 don't support MPEG-4 transmission via HTTP

**When InternetMode property is set to ON, NWcamera or encoder's "Internet mode" has to set to ON. \*1**

**InternetMode property needs to be set to same value of NWcamera or encoder's "Internet mode". \*1**

\*1 In the case of S1136 series, there does not have "Internet mode" setting. Stream reception follows InternetMode property.

**PS-API does not support DDNS name resolver.**

**InterenetMode cannot use with IPv6 environment.**

## Sequence

## Sample program code

## Reference

### 5.5.2.17. FastPlayMode

**Class**            **IPSAPI**

**Property**        **FastPlayMode**

long      SetFastPlayMode(  
                                 long   mode  
                                 );

long      GetFastPlayMode();

#### **Description**

Set the viewing mode of fast forward/rewind rate in network playback into PS-API.

Get the viewing mode of fast forward/rewind rate in network playback from PS-API.

#### **Argument for SET**

mode	0 : Normal mode 1 : High rate mode	Mode of fast forward/rewind rate in network playback
		Default value is 0.

#### **Return value for SET**

0	Success
Except 0	Error code

#### **Return value for GET**

Get the viewing mode of fast forward/rewind rate in network playback  
from PS-API

#### **Error**



**Note**

ND200 and ND300 are not supported this property.

When using this property with ND400, NV200, NV250, NV300, NX Series and HD600/700, please confirm the device's firmware version and it is necessary to use the firmware which support this function.

If the device's firmware doesn't support this function. PS-API works as normal mode.

This property is valid with network playback.

In rewind, regardless of playback speed, only I pictures are displayed.

The description of the playback speed is as follows.

In the case of FastPlayMode = 0 (Normal mode)

[NWDR]

	Step1	Step2	Step3	Step4	Step5	Step6	Step7
ND400 NV200 NV250 NV300	x 1	x 4	x 8	x 16	x 32	x 48	x96

[HD600/700]

	Step1	Step2	Step3	Step4	Step5	Step6	Step7
HD600/700	x 1	x 2	x 5	x 10	x 20	x 50	x100

[NX Series]

	Step1	Step2	Step3	Step4	Step5	Step6	Step7
NX100 NX200 NX300 NX400	x 1	x 4	x 8	x 16	x 32	x 48	x96

In the case of FastPlayMode = 1 (High rate mode)

[NWDR]

	Step1	Step2	Step3	Step4	Step5	Step6	Step7
ND400 NV200 NV250 NV300	x 1	x 2	x 4	x 8	x 16	x 32	x48

[HD600/700]

	Step1	Step2	Step3	Step4	Step5	Step6	Step7
HD600/700	x 1	x 2	x 4	x 8	x 16	x 32	x48

[NX Series]

	Step1	Step2	Step3	Step4	Step5	Step6	Step7
NX100 NX200 NX300 NX400	x 1	x 2	x 4	x 8	x 16	x 32	x48

When using high rate mode, the frames of all I-picture/P-picture/B-picture are displayed in the case of up to Step3. Therefore a delay may occur for drawing when doing fast forward/rewind a video image which is recorded with high resolution and high frame rate.

e.g. ) When doing "Step3" fast forward the video image which is recorded with 4VGA resolution and 30 ips frame rate

**Sequence**

---

**Sample program code**

---

**Reference**

---

### 5.5.2.18. TransFrameRate

**Class**            **IPSAPI**

**Property**        **TransFrameRate**

long      SetTransFrameRate(  
                                 long rate  
                                 );

long      GetTransFrameRate();

#### Description

Set frame rate for video stream into PS-API.

Get frame rate for video stream from PS-API.

TransFrameRate is valid when using JPEG PlayLive with camera and encoder, or using PlayLive/Play with HD300.

#### Argument for SET

rate	0 : Use device setting 1 to 300 : Frame rate	Specify the optional frame rate for video stream. e.g.)    1 : 0.1 [ips] 100 : 10 [ips] 300 : 30 [ips]
------	---	---

Default value is 0.

In case of camera or encoder, the following frame rate are supported.

1, 2, 3, 5, 10, 20, 30, 50, 60, 100, 150, 300

\* If specifying "3", PS-API works as 0.33[ips].

#### Return value for SET

0            Success

Except 0    Error code

#### Return value for GET

Get the frame rate setting for video stream from PS-API.

#### Error

**Note**

---

In case of camera or encoder, if the specified frame rate is not supported on a target device, an error occurs from PS-API when calling PlayLive.

In case of HD300, the frame rate of received video stream may be less than the specified frame rate.

In case of Play with HD300, if specifying frame rate is less than the recording frame rate, playback speed becomes slow.

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.5.2.19. PictureFitMode

**Class**            **IPSAPI**

---

**Property**        **PictureFitMode**

---

long      SetPictureFitMode(  
                                 long   mode  
                                 );

long      GetPictureFitMode();

---

#### **Description**

Set the view mode to keep its aspect ratio of the video stream, or to fit to the drawing area.  
Get the view mode from PS-API.

---

#### **Argument for SET**

type	0: Keep its aspect ratio of the video stream 1: Fit to the drawing area	Specify the view mode to keep its aspect ratio of the video stream, or to fit to the drawing area.  Default value is 1.
------	---	---

---

#### **Return value for SET**

0	Success
Except 0	Error code

---

#### **Return value for GET**

Get the view mode to keep its aspect ratio of the video stream, or to fit  
to the drawing area.

---

#### **Error**

---

**Note**

When PictureFitMode is set to “0” and the aspect ratio of video stream is different of the aspect ratio of drawing area, the black panels are shown on top and bottom or on the both side.



Figure 5-3 Black Panel on the Left and Right



Figure 5-2 Black Panel on the Top and Bottom

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.5.2.20. DigitalZoom

**Class**            **IPSAPI**

**Property**        **DigitalZoom**

long      SetDigitalZoom(  
                                long    zoom  
                                );

long      GetDigitalZoom();

#### **Description**

Do digital zoom of the displayed image by specifying magnification.  
Get the magnification of digital zoom from PS-API.

#### **Argument for SET**

zoom	10 to 80	Specify the magnification of digital zoom. e.g. 10 : x 1 80 : x 8
		Default value is 10.

#### **Return value for SET**

0	Success
Except 0	Error code

#### **Return value for GET**

Get the magnification of digital zoom from PS-API.

#### **Error**

**Note**

---

When starting digital zoom, the center of image is displayed. For moving the displayed area, please use DigitalZoomMove method.

When displaying the video image with black panel and executing this method, the position and width of black panel is kept.

**Sequence**

---

**Sample program code**

---

**Reference**

---



#### 5.5.2.21. DigitalZoomMode

**Class**            **IPSAPI**

---

**Property**        **DigitalZoomMode**

---

long      SetDigitalZoomMode(  
                                 long   mode  
                                 );

long      GetDigitalZoomMode();

---

#### **Description**

Set an enlarged / reduction method of digital zoom into PS-API.  
Get an enlarged / reduction method of digital zoom from PS-API.

---

#### **Argument for SET**

mode	0 : No complement 1 : Complement	Specify the mode of complement when drawing image.  Default value is 0.
------	-------------------------------------	--

---

#### **Return value for SET**

0	Success
Except 0	Error code

---

#### **Return value for GET**

Get an enlarged / reduction method of digital zoom

---

#### **Error**

---

**Note**

---

When using the mode except “0 : No complement”, the processing load to draw increases.

**Sequence**

---

**Sample program code**

---

**Reference**

---

### 5.5.2.22. SkipRecordGap

**Class**            **IPSAPI**

---

**Property**        **SkipRecordGap**

---

long      SetSkipRecordGap(  
                                 long   mode  
                                 );

long      GetSkipRecordGap();

---

#### **Description**

Set the mode whether skip the black images period that there is no recorded video between records into PS-API.

Get the mode whether skip the black images period that there is no recorded video between records from PS-API.

SkipRecordGap is valid with Play method. In case of PlayFile method, this property is ignored and the duration that there is no recorded video is skipped.

---

#### **Argument for SET**

mode	0 : No skip 1 : Use skip	Specify the mode whether skip the black images period that there is no recorded video between records.
------	-----------------------------	--

Default value is 1.

---

#### **Return value for SET**

0	Success
---	---------

Except 0	Error code
----------	------------

---

#### **Return value for GET**

Get the mode whether skip the black images period that there is no recorded video between records.

---

#### **Error**

---

**Note**

---

It does not depend on the setting of this property, the black image is displayed when playback reaches the time that there is no recorded video between records. The resolution of black image is different from the resolution of recorded video image, then the notification of changing resolution occurs by OnError.

When the duration that there is no recorded video between records is less than 13 seconds, it doesn't skip.

In case of HD300, this property is ignored and the duration that there is no recorded video between records is skipped.

**Sequence**

---

**Sample program code**

---

**Reference**

---

### 5.5.2.23. MultiScreenChannel

**Class**            **IPSAPI**

**Property**        **MultiScreenChannel**

```
long      SetMultiScreenChannel(
                                char*  channel
                                );

long      GetMultiScreenChannel(
                                char*  channel
                                long  size
                                )
```

#### **Description**

Set the channel information when doing network playback on the multiple channel with same UID.

Get the channel information from PS-API.

When using the device which supports StreamID mode and setting On to SID mode, this method doesn't need to be executed.

#### **Argument for SET**

channel	Character strings (128 bytes or less)	Specify "1 : ON", "0 : OFF" for each channels. The strings must be terminated with null character.  Default value is "000...000". (128 digits).  When specified 128 bytes character strings, the top (left side) of character is pair to channel 1, and the end (right side) of character is pair to channel 128.  e.g. In case of playing channel 1, 3 and 6. "10100100....."
---------	--	--

#### **Argument for GET**

channel	Character strings	Get the channel information. Application needs to allocate and deallocate memory. If getting value failed, it is set to NULL.
size	Buffer size of channel	Specify the buffer size for channel. (Byte)

---

**Return value for SET**

---

0            Success

Except 0    Error code

---

**Return value for GET**

---

0            Success

Except 0    Error code

---

**Error**

---

---

**Note**

---

When using the device which supports StreamID mode and setting On to SID mode, this method doesn't need to be executed.

This property is valid with HD300, NWDR, HD600/700 and NX Series.

Please set this property after executing Open/Connect method and before executing Play method.

This property is shared between all PS-API instances that share the same UID.

This property reflects the specified value to a target device when property is set. If error occurs, please set this property again.

---

**Sequence**

---

---

**Sample program code**

---

---

**Reference**

---

#### 5.5.2.24. SIDMode

**Class**            **IPSAPI**

**Property**        **SIDMode**

long      SetSIDMode(  
                         long   mode  
                         );

long      GetSIDMode();

#### **Description**

Set the StreamID mode or normal (UID) mode into PS-API.

Get the StreamID mode or normal (UID) mode from PS-API.

#### **Argument for SET**

mode	0 : Normal (use UID) mode 1 : StreamID mode	Specify the StreamID mode or normal (UID) mode.
------	--	---

Default value is 0.

#### **Return value for SET**

0	Success
---	---------

Except 0	Error code
----------	------------

#### **Return value for GET**

Get the StreamID mode or normal (UID) mode.

#### **Error**

**Note**

---

When DeviceType is Network camera, Encoder or HD300, Live/Network playback works with normal (UID) mode regardless of SIDMode setting.

When DeviceType is NWDR or HD600/700 or NX Series and the target device doesn't support StreamID, Live/Network playback doesn't work with StreamID mode.

**Sequence**

---

**Sample program code**

---

**Reference**

---



### 5.5.2.25. BackColor

**Class**            **IPSAPI**

**Property**        **BackColor**

long      SetBackColor(  
                                 long   color  
                                 );

long      GetBackColor();

### Description

Set the background color of displayed area into PS-API.  
Get the background color of displayed area from PS-API.

### Argument for SET

color            0 to 16777215

The sum of R (Red), G (Green) ad B (Blue)  
R, G and B are specified with the following value.

R	:	0	-	255	(0x000000 – 0x0000FF)
G	:	256	-	65280	(0x000100 – 0x00FF00)
B	:	65536	-	16711680	(0x010000 – 0xFF0000)

Default value is 0.

### Return value for SET

0                Success

Except 0       Error code

### Return value for GET

Get the background color of displayed area.

### Error

**Note**

---

**Sequence**

---

**Sample program code**

---

**Reference**

---

### 5.5.2.26. DecResolutionMode

**Class**            **IPSAPI**

**Property**        **DecResolutionMode**

long      SetDecResolutionMode(  
                                 long mode  
                                 );

long      DecResolutionMode ();

#### **Description**

Set the maximum decode resolution for H.264, H.265 into PS-API.

Get the maximum decode resolution for H.264, H.265 from PS-API.

#### **Argument for SET**

mode	0 : Disable the setting of maximum decode resolution. 3 : Enable the setting of maximum decode resolution	Default value is 0.
------	--	---------------------

#### **Return value for SET**

0	Success
Except 0	Error code

#### **Return value for GET**

Get the maximum decode resolution for H.264, H.265 from PS-API.

#### **Error**

## Note

When you use this function, it is necessary to install the ActiveX control, and the device's firmware is required the version which is showed in the Supported Product list or later.  
ND200 and ND300 don't support "3:Enable the setting of maximum decode resolution".

This function is available for H.264/H.265. If set "3:Enable the setting of maximum decode resolution", video image is decoded by the resolution which is set to H264Resolution as maximum.  
e.g.) When DecResoluitonMode is 3 and H264Resolution is 640(VGA), the resolution of the decoded video image is up to VGA, even if the original video image resolution which is transmitted by camera is 1280(4VGA).

When you enable this function, the snapshot image size is upto the size which is set to H264Resolution.

e.g.)

Decoded video image size		Original H.264/H.265 resolution (Camera transmission setting)		
		320	640	1280
Maximum decode size setting (H264Resolution with DecResolutionMode = 3)	320	320	<b>320</b>	<b>320</b>
	640	320	640	<b>640</b>
	1280	320	640	1280

## Sequence

## Sample program code

## Reference

#### 5.5.2.27. CroppingEnabled

**Class**            **IPSAPI**

**Property**        **CroppingEnabled**

long      SetCroppingEnabled (  
                                 long   mode  
                                 );

long      GetCroppingEnabled ();

#### **Description**

Set the Cropping mode.

Get the Cropping mode.

#### **Argument for SET**

mode	0 : Don't use Cropping	Default value is 0.
	1 : Use Cropping	

#### **Return value for SET**

0	Success
---	---------

Except 0	Error code
----------	------------

#### **Return value for GET**

Get the Cropping mode.

#### **Error**

#### **Note**

#### **Sequence**

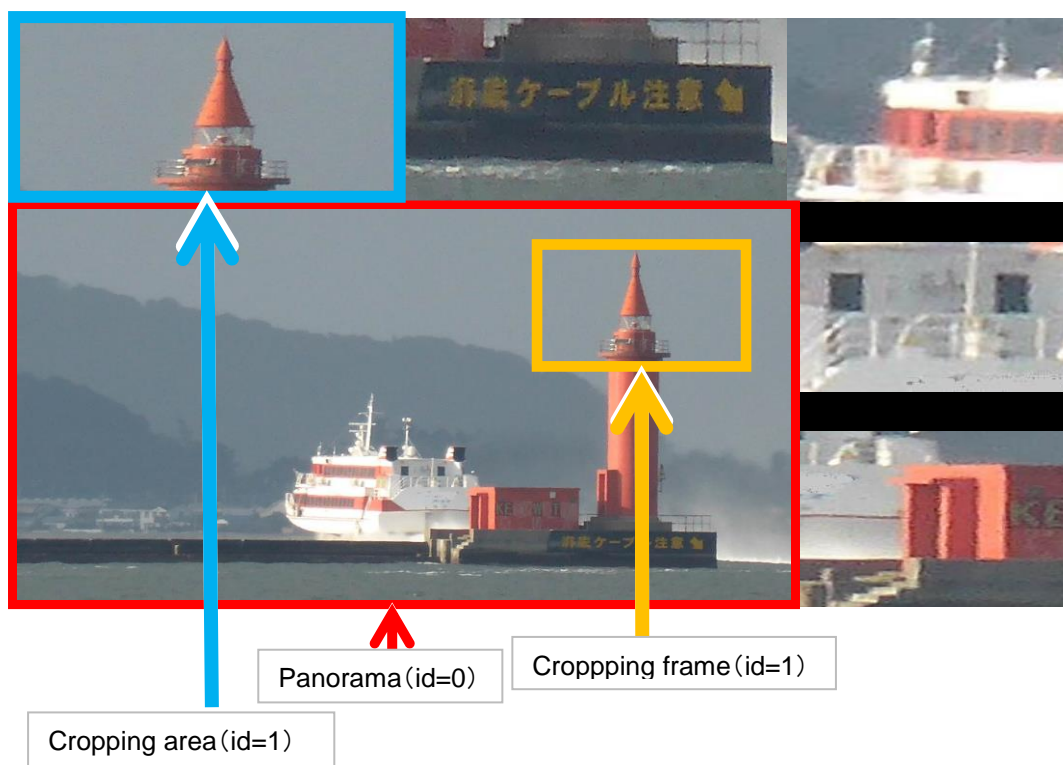
6.16 Cropping

#### **Sample program code**

#### **Reference**

## Note

Set “5.5.1.35 SetCroppingRect”, “5.5.1.37 SetCroppingDrawRect” and “5.5.1.41 SetCroppingMarker”, and turn on the Cropping function, then the setting value will be refrected.



## Sequence

6.16 Cropping

## Sample program code

## Reference

#### 5.5.2.28. RcvAudioDec

**Class**            **IPSAPI**

---

**Property**        **RcvAudioDec**

---

long      SetRcvAudioDec(  
                                 long   format  
                                 );

long      GetRcvAudioDec();

---

#### **Description**

Set audio format into PS-API.  
Get audio format from PS-API.

---

#### **Argument for SET**

format	0 : auto	
	1 : G.726	
	3 : AAC	Default value is 1.

---

#### **Return value for SET**

0	Success
Except 0	Error code

---

#### **Return value for GET**

Get audio format from PS-API.

---

#### **Error**

---

## Note

Even if "auto" or correct audio format is specified, audio may not be heard.  
Please see the table below for details.

	PlayLive	Play	PlayFile
G.726	Support	Support *2	Support *3
AAC *1	Support	Support *2	Support *3

\*1 When the installed ActiveX control does not support AAC, the audio is not heard.

\*2 When is changed of the audio setting of NX Series between record creation and playback audio may not be heard.

\*3 If both the G.726 and AAC audio format are recorded in the file to be played, audio may not be heard.

\*Whether or not the installed ActiveX control is compatible with AAC,  
It depends on the version of ActiveX control installed.  
Refer to the table below for the version compatible with AAC.

Program name *4	Version *4
Network Camera value3	Not support
Network Camera value4	Not support
Network Camera Value4S	V4.37.04 or later
WebVideo ActiveX	Not support
WebVideo2 ActiveX	Not support
Web Video ActiveX NX	V6.0.20.0 or later

\*4 The program name and version can be confirmed from "the addition and deletion" of program of the control panel.

## Sequence

## Sample program code

## Reference



#### 5.5.2.29. TransIntervalMode

Class	IPSAPI
Property	TransIntervalMode
long	SetTransIntevalMode (long mode);
long	GetTransIntervalMode();

### Description

Set the mode to correct the transmission interval of decoded pictures.

Get the mode to correct the transmission interval of decoded pictures.

## Argument for SET

mode	0:OFF 1:ON	Default value is 0.
------	---------------	---------------------

### Return value for SET

0	Success
Except 0	Error code

### Return value for GET

---

Get the mode to correct the transmission interval of decoded pictures.

## Error

### Note

- JPEG/HD300 are not supported
- Not supported when the frame rate is less than 1fps  
Even if the rate temporarily drops less 1fps due to a network error or other error, the function stops.  
Execute the PlayLive/Play/PlayFile method to resume the function.

When TransIntervalMode = 1, it has the effect of suppressing the video unsmooth movement. The effect can be adjusted by the setting value of DecBufferNum property. Set DecBufferNum property according to the operating environment.

supplementary note:

- (1) This function may not be effective.  
e.g.) video unsmooth movement due to network delay
- (2) It takes a little more time from operation until the first image appears.  
But does not mean that the video delay spread.
- (3) It would be increased memory usage, because it suppress the video unsmooth movement by accumulateing image.

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.5.2.30. DecBufferNum

Class	IPSAPI
Property	DecBufferNum
long	SetDecBufferNum (long bufCount);
long	GetDecBufferNum();

### Description

Set decode buffer number to PS-API
Get decode buffer number from PS-API

### Argument for SET

bufCount	4 - 20 : buffer number	Default value is 4.
----------	------------------------	---------------------

### Return value for SET

0	Success
---	---------

Except 0	Error code
----------	------------

### Return value for GET

---

Get decode buffer number from PS-API

## Error

## Note

## Sequence

### Sample program code

## Reference

### 5.5.3. Application Listener

#### 5.5.3.1. OnPlayStatus

Class	IAppListener
Listener	OnPlayStatus
void	OnPlayStatus( long channel, long status );

#### Description

Notify the play status change to the specified application.  
Application needs to create the listener class that inherits IAppListener and to implement OnPlayStatus method.

#### Argument

channel	0 : In case of file playback 1 : Network Camera, GXE100 1 to 4 : Encoder NX100 1 to 16 : ND200, HD300, HD600/700 1 to 24 : NV200, NV250 1 to 32 : ND300, NV300, NX200, NX300 1 to 64 : ND400 1 to 128 : NX400	Specify the video channel.
status	-1 : Invalid status (Run with non-blocking mode or not displaying Playback and Live image) 0 : Live 1 : Pause 2 : Play 3 : Preparing for Play	Specify the current video play status.

#### Return value

None

#### Error

**Note**

---

Please don't use PS-API functions in the notified listener function.

When the "HDD Standby Control" setting is "ON" in the NX Series, it may take time to play back.  
In that case, "status = 3 (Preparing for Play)" may be notified.

**Sequence**

---

**Sample program code**

---

**Reference**

---

### 5.5.3.2. SetPlayListener

**Class**                      **IPSAPI**

**Listener**                      **SetPlayListener**

```
long      SetPlayListener(  
                                IAppListener* pReceiver  
                                );
```

#### **Description**

Set the instance of listener class that implement “5.5.3.1 OnPlayStatus” .  
After setting the listener, PS-API can notify any play status change by calling “5.5.3.1 OnPlayStatus”, when it happens.

#### **Argument**

pReceiver	Pointer	Pointer for Listener class. Don't delete the registered object by your application.
	NULL	If pReceiver is set to NULL, PS-API unregister the Listener class.

#### **Return value**

0	Success
Except 0	Error code

#### **Error**

Error is defined by the return value.

**Note**

---

Don't delete the registered listener object by your application.

**Sequence**

---

**Sample program code**

---

**Reference**

---

### 5.5.3.3. OnRecordStatus

**Class** **IAppListener**

**Listener** **OnRecordStatus**

```
void OnRecordStatus(
    long recType,
    char* timeDate,
    long isDst,
    char* nextRecTime,
    long isDstNext
);
```

### Description

In case of network playback, notify the beginning point of the record and the end point of the record to the specified application.

Application needs to create the listener class that inherits IAppListener and to implement OnRecordStatus method.

### Argument

recType	1 : The beginning point 2 : The end point	The beginning point and the end point of the record.
timeDate	yyyy/mm/dd hh:mm:ss	Time-and-date of the played record.
isDst	0 : Normal 1 : Summer time	Summer time information of the time that is specified to timeDate.
nextRecTime	yyyy/mm/dd hh:mm:ss	When recType is set to "2", the beginning time-and-date of the next record is set. If there is no next record or if recType is set to "1", NULL is set.
isDstNext	0 : Normal 1 : Summer time	Summer time information of the time that is specified to nextRecTime.

### Return value

None

### Error



**Note**

---

Please don't use PS-API functions in the notified listener function.

If there is no record at the specified time-and-date by using Play method, this event notify with "recType = 2".

In case of HD300, even if OnRecordStatus listener is registered, OnRecordStatus is not notified.

In case of playback with file, even if OnRecordStatusEnable is set to "ON", OnRecordStatus event is not notified.

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.5.3.4. SetRecordListener

**Class**                      **IPSAPI**

**Listener**                      **SetRecordListener**

```
long      SetRecordListener(  
                                IAppListener*   pReceiver  
                                );
```

#### **Description**

Set the instance of listener class that implement “5.5.3.3 OnRecordStatus” .  
After setting the listener, PS-API can notify any play status change by calling “5.5.3.3 OnRecordStatus ”, when it happens.

#### **Argument**

pReceiver	Pointer	Pointer for Listener class. Don't delete the registered object by your application.
	NULL	If pReceiver is set to NULL, PS-API unregister the Listener class.

#### **Return value**

0	Success
Except 0	Error code

#### **Error**

Error is defined by the return value.

**Note**

---

Don't delete the registered listener object by your application.

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.5.3.5. OnImage

**Class**            **IAppListener**

**Listener**        **OnImage**

```
void      OnImage(  
                 long   type,  
                 unsigned char*   pBuffer,  
                 long   size  
                 );
```

#### Description

Notify the updated image to the specified application.

Application needs to create the listener class that inherits IAppListener and to implement OnImage method.

#### Argument

type	1 : decoded video image (RGB32) 2 : Compressed video image	Specify a video image type.  “2 : Compressed video image” supports PlayLive of network camera and encoder, and PlayLive/Play of NWDR, HD600/700 and NX Series.PlayFile is not supported. About Compressed video image format, please refer to Note.
pBuffer	Pointer of buffer	Specify a pointer to an image buffer to hold video image data. When OnImage function finishes, memory will be deallocated.
size	Data size [Byte]	

#### Return value

None

#### Error

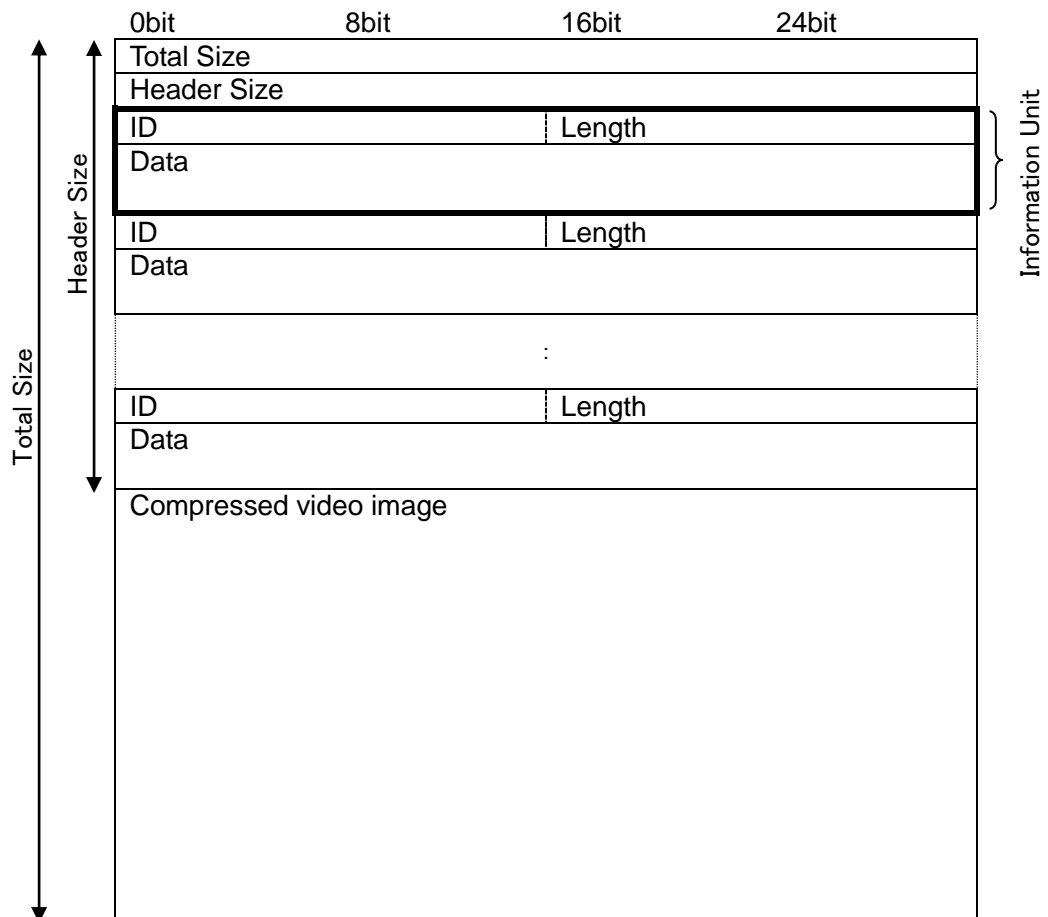
**Note**

Please don't use PS-API functions in the notified listener function.

**Data format of Compressed video image (JPEG/MPEG-4/H.264/H.265)**

Compressed video image consists of Header part and Data part. Header part consists of some "Information Unit". Data part consists of an compressed video image.

Information Unit is recognized by "ID". Information Unit may be added, and the order may be changed in future.



\* Each value is put with big endian.

No.	Name	Size	Description
1	Total Size	4Bytes	The total data size of Compressed video image e.g.) 0x0001 86A0 -> 100,000Bytes
2	Header Size	4Bytes	The header data size of Compressed video image e.g.) 0x0000 0034 -> 52 Bytes

## Note

Information Unit is defined as follows.

### [Time information]

offset	0bit	8bit	16bit	24bit
0x00	ID=0x0011		Length	
0x04	Clock			
0x08	TimeZoneDirection	TimeZoneHour	TimeZoneMinute	SummerTime

No.	Name	Size	Description
1	ID	2Bytes	0x0011 (Fixed)
2	Length	2Bytes	The data size of time information. (ID and Length data size are included) Byte unit. e.g.) 0x000C -> 12Bytes
3	Clock	4Bytes	A multiplication second from 1970
4	TimeZoneDirection	1Byte	Time zone (Plus or minus) 0x00 : - (minus) 0x01 : + (plus)
5	TimeZoneHour	1Byte	Time zone (Hour) 0x00: 0:00, 0x01: 1:00 0x02: 2, 0x03: 3:00 0x04: 4:00, 0x05: 5:00, 0x06: 6:00, 0x07: 7:00 0x08: 8:00, 0x09: 9:00, 0x0a: 10:00 0x0b: 11:00 0x0c: 12:00, 0x0d: 13:00, 0x0e: 14:00, 0x0f: 15:00 0x10: 16:00, 0x11: 17:00, 0x12: 18:00, 0x13: 19:00 0x14: 20:00, 0x15: 21:00, 0x16: 22:00, 0x17: 23:00
6	TimeZoneMinute	1Byte	Time zone (Minute) 0x00: 0:00, 0x01: 0:01, 0x02: 0:02, ....., 0x39: 0:57, 0x3a: 0:58, 0x3b: : 0:59
7	SummerTime	1 Byte	Summer time (Daylight saving time) 0x00 : off 0x01 : Summer time (DST)

## Note

### [Frame time information (millisecond)]

offset	0bit	8bit	16bit	24bit
0x00	ID=0x0012		Length	
0x04	FrameTime		Padding 0x0000 (Fixed)	

No.	Name	Size	Description
1	ID	2Bytes	0x0012 (Fixed)
2	Length	2Bytes	The data size of frame time information. (ID and Length data size are included) Byte unit. e.g.) 0x0008 -> 8 Bytes
3	FrameTime	2Bytes	Milliseconds (10milliseconds unit) 0x0000: 0 [ms],      0x0001: 10 [ms], ..... 0x0062: 980 [ms],   0x0063: 990 [ms]  * In case of PlayLive with NWDR, HD600/700 or NX Series FrameTime is 0x0000 (Fixed).
4	Padding	2Bytes	0x0000 (Fixed)

### [Compression method]

offset	0bit	8bit	16bit	24bit
0x00	ID=0x1001		Length	
0x04	Compression		Padding 0x0000 (Fixed)	

No.	Name	Size	Description
1	ID	2Bytes	0x1001 (Fixed)
2	Length	2Bytes	The data size of compression method information. (ID and Length data size are included) Byte unit. e.g.) 0x0008 -> 8 Bytes
3	Compression	2Bytes	0x0000 :JPEG 0x0001 :MPEG-4 0x0003 : H.264(type1) 0x0004 : H.264(type2) 0x0006 : H.265
4	Padding	2Bytes	0x0000 (Fixed)

## Note

### [Frame type information]

offset	0bit	8bit	16bit	24bit
0x00	ID=0x1002		Length	
0x04	FrameType		Padding 0x0000 (Fixed)	

No.	Name	Size	Description
1	ID	2Bytes	0x1002 (Fixed)
2	Length	2Bytes	The data size of frame type information. (ID and Length data size are included) Byte unit. e.g.) 0x0008 -> 8 Bytes
3	FrameType	2Bytes	0x0000 :JPEG frame 0x0001 :MPEG-4 I frame / H.264 I picture H.265 I picture(display) 0x0002 :MPEG-4 P frame H.264 P picture / H.264 B picture H.265 P picture 0x0003 :H.265 I picture(non-display)
4	Padding	2Bytes	0x0000 (Fixed)

### [Resolution information]

offset	0bit	8bit	16bit	24bit
0x00	ID=0x1003		Length	
0x04	ImageWidth		ImageHight	

No.	Name	Size	Description
1	ID	2Bytes	0x1003 (Fixed)
2	Length	2Bytes	The data size of resolution information. (ID and Length data size are included) Byte unit. e.g.) 0x0008 -> 8 Bytes
3	ImageWidth	2Bytes	0x0000 - 0xFFFF. e.g.) 1024x768 -> 0x0400
4	ImageHight	2Bytes	0x0000 - 0xFFFF. e.g.) 1024x768, -> 0x0300



---

**Note**

---

The resolution of compressed video image gotten by OnImage depends on the device setting regardless of the resolution setting in PS-API.

When the resolution setting in PS-API is set to 320 and PlayLive runs with HD600/700, the resolution of compressed video image gotten by OnImage is QVGA(320x240) regardless of the device setting.

---

**Sequence**

---

---

**Sample program code**

---

---

**Reference**

---

### 5.5.3.6. SetImageListener

**Class**                      **IPSAPI**

**Listener**                      **SetImageListener**

```
long      SetImageListener(
                IAppListener* pReceiver,
                long   type
            );
```

#### **Description**

Set the instance of listener class that implement "OnImage" .  
After setting the listener, PS-API can notify an image data by calling "OnImage", when the image update happens.

#### **Argument**

pReceiver	Pointer	Pointer for Listener class. Don't delete the registered object by your application.
	NULL	If pReceiver is set to NULL, PS-API unregister the Listener class.
type	1 : decoded video image (RGB32) 2 : Compressed video image	Specify a video image type. "2 : Compressed video image" supports PlayLive of network camera and encoder, and PlayLive/Play of NWDR, HD600/700 and NX Series. PlayFile is not supported.About Compressed video image format, please refer to 5.5.3.5 OnImage Note.

#### **Return value**

0	Success
Except 0	Error code

#### **Error**

Error is defined by the return value.

## Note

---

**Don't delete the registered listener object by your application.**

If the StreamFormat is "2(HDR)" and the "type" of this function is 2, this function returns error.

\* Similarly, SetStreamFormat should return error, if the ImageListener is set with "type=2".

## Sequence

---

## Sample program code

---

## Reference

---

#### 5.5.3.7. OnMP4DownloadStatus

**Class**                      **IAppListener**

**Listener**                      **OnMP4DownloadStatus**

```
void      OnMP4DownloadStatus (
                long      status,
                char*      fileName
            );
```

#### **Description**

Notify the status of HttpMP4Download and HttpDownload method to the specified application. Application needs to create the listener class that inherits IAppListener and to implement OnMP4DownloadStatus method.

#### **Argument**

status	0 : Completed file download. 1 : File downloading.(split file) 2 : Forced stop of file download	Result of file download
fileName	Pointer to character strings	Output file name

#### **Return value**

None

## **Error**

---

## **Note**

Please don't use PS-API functions in the notified listener function.

## **Sequence**

---

## **Sample program code**

---

## **Reference**

---

#### 5.5.3.8. SetMP4DownloadListener

**Class**                      **IPSAPI**

**Listener**                      **SetMP4DownloadListener**

```
long      SetMP4DownloadListener(  
                                 IAppListener*   pReceiver  
                                 );
```

#### **Description**

Set the instance of listener class that implement “5.5.3.7 OnMP4DownloadStatus” .  
After setting the listener, PS-API can notify any file download status change by calling “5.5.3.7 OnMP4DownloadStatus”, when it happens.

#### **Argument**

pReceiver	Pointer	Pointer for Listener class. Don't delete the registered object by your application.
	NULL	If pReceiver is set to NULL, PS-API unregister the Listener class.

#### **Return value**

0	Success
Except 0	Error code

#### **Error**

Error is defined by the return value.

**Note**

---

Don't delete the registered listener object by your application.

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.5.4. Application Callback

##### 5.5.4.1. OnPlayStatusCB

Class	IAppCallBack
Callback	OnPlayStatusCB
void	OnPlayStatusCB( long status );

#### Description

Notify the play status change to the specified application.  
Application needs to create the callback class that inherits IAppCallBack and to implement OnPlayStatusCB method.

#### Argument

status	-1 : Invalid status (Run with non-blocking or not displaying Playback and Live image) 0 : Live 1 : Pause 2 : Play 3 : Preparing for Play	Specify the current video play status.
--------	---	--

#### Return value

None

#### Error



**Note**

---

Please don't use PS-API functions in the notified callback function.  
When the "HDD Standby Control" setting is "ON" in the NX Series, it may take time to play back.  
In that case, "status = 3 (Preparing for Play)" may be notified.

**Sequence**

---

**Sample program code**

---

**Reference**

---

## 5.6. Audio Group

### 5.6.1. Method

#### 5.6.1.1. AudioSend

**Class**                **IPSAPI**

**Method**            **AudioSend**

```
long    AudioSend(  
              long    command  
              );
```

---

#### **Description**

Start or Stop the audio transmission during Live.

---

#### **Argument**

command	0 : Stop audio transmission 1 : Start audio transmission
---------	---

---

#### **Return value**

0	Success
Except 0	Error code

---

#### **Error**

Error is defined by the return value.  
Get the error information by OnError event.

## **Note**

---

The audio transmission can be used when doing PlayLive with Network camera and encoder. When send audio to the devices with full duplex mode or half duplex mode, please set AudioRcvEnable to "1".

The audio transmission is disconnected by duration configured at the device(default : 5 minutes). If you need to send audio continuously, please find out the disconnection by OnError or AudioSendStatus, and then execute AudioSend method again.

This method uses the sound recording device on PC. Therefore to use this method may affect other applications.

Only one PS-API instance can use the audio transmission function. When the other instance uses the audio transmission function, error occurs by calling this method.

If you cannot work audio transmission , please check the following Windows settings.

- [Windows+S] Services - Windows Audio Status : Running
- [Windows+S] Privacy settings - Microphone - Allow apps to access your microphone : On

## **Sequence**

---

## **Sample program code**

---

## **Reference**

---

#### 5.6.1.2. GetAudioSendStatus

**Class**                **IPSAPI**

**Method**            **GetAudioSendStatus**

long      GetAudioSendStatus();

---

#### **Description**

Get the audio transmission status.

---

#### **Argument**

None

---

#### **Return value**

0	Stop sending audio
1	In sending audio
2	Busy (The other instance uses the audio transmission)
Negative value	Error code

---

#### **Error**

Error is defined by the return value.  
Get the error information by OnError event.

**Note**

---

**Sequence**

---

**Sample program code**

---

**Reference**

---

## 5.6.2. Property

### 5.6.2.1. AudioRcvEnable

**Class**                **IPSAPI**

**Property**            **AudioRcvEnable**

long        SetAudioRcvEnable(  
                                 long   mode  
                                 );

long        GetAudioRcvEnable();

#### **Description**

Set a mode whether receiving audio stream with PlayLive or Play into PS-API.

Get a audio reception mode from PS-API.

#### **Argument for SET**

mode	0 : Not receive audio stream	Specify a audio reception mode.
	1 : Receive audio stream	Default value is 0.

#### **Return value for SET**

0	Success
---	---------

Except 0	Error code
----------	------------

#### **Return value for GET**

Get a audio reception mode from PS-API.

#### **Error**

## Note

After PlayLive and Play started, the setting cannot be changed. Please adjust by AudioRcvVolume. Some instances receives audio stream at the same time, the sound is composed.

The audio reception support depends on a target device.  
Please refer to the following table.

	HD300	NWDR	Camera	Encoder	HD600/700	NX Series
PlayLive	-	*1	Yes	Yes	Yes	Yes
Play	-	*2			Yes	Yes
PlayFile	-	Yes			Yes	Yes

\*1 : ND200 and ND300 don't support it.

\*2 : ND300 doesn't support it.

If you cannot work audio reception , please check the following Windows settings.

- [Windows+S] Services - Windows Audio Status : Running

## Sequence

## Sample program code

## Reference

### 5.6.2.2. AudioRcvVolume

**Class**                **IPSAPI**

---

**Property**            **AudioRcvVolume**

---

long        SetAudioRcvVolume(  
   long   volume  
   );

long        GetAudioRcvVolume();

---

#### **Description**

Set a volume of the audio reception into PS-API.  
Get a volume of the audio reception from PS-API.

---

#### **Argument for SET**

volume	0 (minimum) to 100 (maximum)	Volume of the audio reception.  Default value is 10.
--------	---------------------------------	--

---

#### **Return value for SET**

0	Success
Except 0	Error code

---

#### **Return value for GET**

Get a volume of the audio reception

---

#### **Error**

---



### **Note**

---

The volume can be set each instance.  
Some instances receives audio stream at the same time, the sound is composed.  
When the audio reception needs to set mute, please use AudioRcvMute property.  
Even if AudioRcvVolume is set to "0", it is not mute.

### **Sequence**

---

### **Sample program code**

---

### **Reference**

---

### 5.6.2.3. AudioRcvMute

**Class**                **IPSAPI**

**Property**           **AudioRcvMute**

long        SetAudioRcvMute(  
                                 long   mute  
                                 );

long        GetAudioRcvMute();

#### **Description**

Set a mute mode setting of the audio reception into PS-API.

Get a mute mode setting of the audio reception from PS-API.

#### **Argument for SET**

mute	0 : Not mute mode 1 : Mute mode	Mute mode of the audio reception
		Default value is 0.

#### **Return value for SET**

0	Success
Except 0	Error code

#### **Return value for GET**

Get a mute mode setting of the audio reception

#### **Error**

**Note**

---

The volume can be set each instance.  
Some instances receives audio stream at the same time, the sound is composed.

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.6.2.4. AudioSendVolume

**Class**                **IPSAPI**

---

**Property**            **AudioSendVolume**

---

long        SetAudioSendVolume(  
   long   volume  
   );

long        GetAudioSendVolume();

---

#### **Description**

Set a volume of the audio transmission into PS-API.  
Get a volume of the audio transmission from PS-API.

---

#### **Argument for SET**

volume	0 (minimum) to 100 (maximum)	Volume of the audio transmission.  Default value is 10.
--------	---------------------------------	---

---

#### **Return value for SET**

0	Success
Except 0	Error code

---

#### **Return value for GET**

Get a volume of the audio transmission

---

#### **Error**

---

**Note**

---

This property changes the sound recording device on PC.  
When the audio transmission needs to set mute, please use AudioSendMute property.  
Even if AudioSendVolume is set to "0", it is not mute.

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.6.2.5. AudioSendMute

**Class**                **IPSAPI**

**Property**            **AudioSendMute**

long        SetAudioSendMute(  
                                 long   mute  
                                 );

long        GetAudioSendMute();

#### **Description**

Set a mute mode setting of the audio transmission into PS-API.  
Get a mute mode setting of the audio transmission from PS-API.

#### **Argument for SET**

0 : Not mute mode	Mute mode of the audio transmission.
1 : Mute mode	
	Default value is 0.

#### **Return value for SET**

0            Success

Except 0    Error code

#### **Return value for GET**

Get a audio reception mode from PS-API.

#### **Error**

**Note**

---

This property does NOT change the sound recording device on PC.

**Sequence**

---

**Sample program code**

---

**Reference**

---

## 5.7. Operation Group

### 5.7.1. Method

#### 5.7.1.1. CameraControl

Class	IPSAPI
Method	CameraControl
long	CameraControl( long channel, long pan, long tilt, long zoom, long focus, long iris );

#### Description

Send Pan, Tilt, Zoom, Focus, Iris command to the specified camera.

In case of sending CameraControl method repeatedly, the interval between commands needs to be set to 80 or more [milliseconds].

**\* About more detailed information, refer to Note.**

#### Argument

channel	1 : Network Camera, GXE100 1 to 3 : S8573 1 to 4 : Encoder, X8570, X8571, S8574 NX100 1 to 16 : ND200, HD300, HD600/700 1 to 24 : NV200, NV250 1 to 32 : ND300, NV300, NX200, NX300 1 to 64 : ND400 1 to 128 : NX400	Specify the video channel. X8570, X8571, S8573, S8574 can use "Iris" function for specified channel. However, when Image capture mode on S8573, S8574 is "Quad" or "Panorama", "Iris" function works as channel 1 regardless of the specified channel value.
pan	0 : Stop -256(Left) to 256(Right) : Pan speed	<b>* The direction of Pan/Tilt is for ceiling setting mode. (The camera is to be installed with the dome side down.)</b>
tilt	0 : Stop -256(Upper) to 256(Lower) : Tilt speed	
zoom	0 : Stop -4 (Wide) to 4 (Tele) : Zoom speed	



focus	0 : Stop -4 (Near) to 4 (Far) : Focus speed
iris	0 : Iris Stop 1 : Iris Open 2 : Iris Close 3 : Iris return to the default position

---

#### Return value

0	Success
Except 0	Error code

---

#### Error

Error is defined by the return value.  
Get the error information by OnError event.

---

#### Note

Depending on the camera model, a PoE + (compliant with IEEE802.3at) compatible hub or power supply device may be required. Please refer to the device manual.

In case of the NWDR, HD600/700 and NX Series, the network camera or the encoder, Focus cannot be used with Pan/Tilt/Zoom at the same time. If Pan/Tilt/Zoom and Focus is specified, Pan/Tilt/Zoom has priority, Focus is invalid.

When controlling the analog camera via a HD300, 2 or more operations (Pan and Tilt/ Zoom/ Focus/ Iris) cannot be used at the same time.

In case of a HD300, after sending the operation command, the analog camera behaves for 2 seconds and then stops behavior.

In case of the HD300, if the interval between the commands is short, the analog camera doesn't work.

When controlling the analog camera via a HD300, HD600/700 or encoder, error will not occur though the target analog camera doesn't support the operations (Pan and Tilt/ Zoom/ Focus/ Iris).

When using the camera with electronic zoom and setting the electronic zoom function to "ON", Zoom control doesn't work after reaching the edge of tele or the edge of wide. In this case, please send zoom stop command.

When Network Camera is set to "desktop setting mode" (the camera is to be installed with the dome side up), the direction of Pan/Tilt is same as "ceiling setting mode".

When using 360-degree Network Camera, please refer to "4.7 About 360-degree Network Camera".

---

#### Sequence

##### 6.4 CameraControl

---

#### Sample program code

[Visual C++ 2012] ..¥Sample Program¥PS-API¥Visual C++ 2012¥004\_CameraControl

---

#### Reference

### 5.7.1.2. SetCameraPosition

**Class**                **IPSAPI**

**Method**            **SetCameraPosition**

```
long    SetCameraPosition(
        long   channel
        long   pan,
        long   tilt,
        long   zoom,
        long   focus
    );
```

#### **Description**

Set Pan, Tilt, Zoom and Focus values in absolute angle.  
SetCameraPosition is valid to a Network Camera.

#### **Argument**

channel	1 (Fixed)	Specify the video channel.
pan	-475 to 3599	Specified Pan value -475 : -47.5 [degree] (Left) 3599 : 359.9 [degree] (Right)
tilt	-900 to 900	Specified Tilt value -900 : -90 [degree] (Upper) 900 : 90 [degree] (Lower)  <b>* The direction of Pan/Tilt is for ceiling setting mode. (The camera is to be installed with the dome side down.)</b>
zoom	10 to 9999	Specified Zoom value 10 : x1 9999 : x999.9
focus	1 to 9999	Specified Focus value 1 : 0.1 [m] 9999 : 999.9 [m]

---

**Return value**

---

0	Success
Except 0	Error code

---

**Error**

---

Error is defined by the return value.  
Get the error information by OnError event.

---

**Note**

---

Depending on the camera model, the valid range (angle, zoom) is different.  
Depending on the camera model, a PoE + (compliant with IEEE802.3at) compatible hub or power supply device may be required. Please refer to the device manual.

The got value by using GetCameraPosition is different from the set value by using SetCameraPosition.

When Network Camera is set to “desktop setting mode” (the camera is to be installed with the dome side up), the plus value of Tilt works as Lower and the minus value of Tilt works as Upper. The direction of Pan is same as “ceiling setting mode”.

---

**Sequence**

---

**6.5 CameraOperation**

---

**Sample program code**

---

[Visual C++ 2012] ..¥Sample Program¥PS-API¥Visual C++ 2012¥005\_CameraOperation

---

**Reference**

---

### 5.7.1.3. GetCameraPosition

**Class**                **IPSAPI**

**Method**            **GetCameraPosition**

```
long    GetCameraPosition(
                                long channel,
                                long& pan,
                                long& tilt,
                                long& zoom,
                                long& focus
                                );
```

#### **Description**

Get Pan, Tilt, Zoom and Focus values in absolute angle.  
GetCameraPosition is valid to a Network Camera.

#### **Argument**

channel	1 (Fixed)	Specify the video channel.
pan	-475 to 3599	Current Pan value -475 : -47.5 [degree] (Left) 3599 : 359.9 [degree] (Right)
tilt	-900 to 900	Current Tilt value -900 : -90 [degree] (Upper) 900 : 90 [degree] (Lower)  <b>* The direction of Pan/Tilt is for ceiling setting mode. (The camera is to be installed with the dome side down.)</b>
zoom	10 to 10800	Current Zoom value 10 : x1 10800 : x1080
focus	1 to 9999	Current Focus value 1 : 0.1 [m] 9999 : 999.9 [m]

---

**Return value**

---

0	Success
Except 0	Error code

---

**Error**

---

Error is defined by the return value.  
Get the error information by OnError event.

---

**Note**

---

Depending on the camera models, the valid range (angle, zoom) is different.  
The got value by using GetCameraPosition is different from the set value by using SetCameraPosition.

When Network Camera is set to “desktop setting mode” (the camera is to be installed with the dome side up), Lower is the plus value of Tilt and Upper is the minus value of Tilt. The direction of Pan is same as “ceiling setting mode”.

---

**Sequence**

---

6.5 CameraOperation

---

**Sample program code**

---

[Visual C++ 2012] ..¥Sample Program¥PS-API¥Visual C++ 2012¥005\_CameraOperation

---

**Reference**

---

#### 5.7.1.4. CameraOperation

Class		IPSAPI	
Method	CameraOperation		
long	CameraOperation(	long channel long command, long data, long& status, long mode, IAppCallBack* pSender );	
<b>Description</b>			
Operate camera such as auto track, auto pan, auto focus.			
<b>Argument</b>			
	channel	1 : Network Camera, GXE100 1 to 3 : S8573 1 to 4 : Encoder X8570,X8571 S8574 NX100 1 to 16 : ND200, HD300, HD600/700 1 to 24 : NV200, NV250 1 to 32 : ND300, NV300 NX200, NX300 1 to 64 : ND400 1 to 128 : NX400	Specify the video channel. X8570, X8571, S8573, S8574 can use “Super Dynamic” function for specified channel. However, when Image capture mode on S8573, S8574 is “Quad” or “Panorama”, “Super Dynamic” function works for all channels regardless of the specified channel value.
	command	0 : No operation 1 : Auto Track 2 : Auto Pan 3 : Auto Focus 4 : Set Preset 5 : Call Preset 6 : Delete Preset 7 : Auto Back Focus 8 : Super Dynamic 9 : Preset sequence 10 : Auto sort 11 : Patrol	Encoder, HD300, NWDR, HD600/700 and NX Series don’t support “6 : Delete Preset”, “7 : Auto Back Focus” and “8 : Super Dynamic”. NW camera, GXE500, GXE100, NWDR and NX Series don’t support “10 : Auto sort”.

## Argument

data	[Set / Call / Delete Preset position ] 0 : Home position 1 to 256 : NWDR, NX Series Encoder, HD300, Network Camera	Preset position number When command is specified to "0, 1, 2, 3, 7, 10", data is ignored even if having a value. When command is specified to "4, 6", "0 : Home position" cannot be set.
	[Super Dynamic] 0 : OFF 1 : ON(Normal) 2 : ON(High) 3 : ON(Super Dynamic)	
	[Preset sequence] 1 : HD300, NWDR, NX Series Encoder HD600/700 1 to 3 :Network Camera	For preset sequences, if data is specified as an out-of-range value, it works as data = 1.
	[Patrol] 1 : HD300, NWDR, NX Series NT304, NT314, HD600/700 1 to 4 :GXE500, GXE100, Network Camera	
status	-1 : Fail to get status. 0 : not used 1 : auto track 2 : auto pan 9 : Preset sequence 10 : Auto sort 11 : Patrol	In case of blocking mode, the current status is set by PS-API.
mode	0 : blocking Except 0 : Non-blocking	Blocking mode.
pSender	Callback interface	Specify a callback interface for the camera operation. Please refer to application callback section and look for function OnOpStatusCB() for details. Don't delete the registered object by your application.

---

**Return value**

---

0	Success
Except 0	Error code

---

**Error**

---

Error is defined by the return value.  
Get the error information by OnError event.

---

**Note**

---

The target device doesn't support the specified control, PS-API returns error.  
Depending on the camera model, a PoE + (compliant with IEEE802.3at) compatible hub or power supply device may be required. Please refer to the device manual.

When controlling HD300, HD600/700 or encoder, any control return success but the behavior depends on the camera models.

When controlling HD300, HD600/700 or encoder, Auto pan and Auto track return success but the behavior depends on the camera setting.

In case that the target device is HD300, HD600/700 or encoder, the got status information isn't same as the device behavior because the behavior depends on the camera setting.

In case of Call preset, if "data" (the preset position number) that doesn't have a position is specified, Camera will not work.

In case that the target device is HD300, HD600/700 or encoder, even if the related analog camera doesn't support 65 or higher preset position number, PS-API doesn't return error.

In case that the target device is X4571, even if the Super Dynamic doesn't support "2 : ON(High)" PS-API doesn't return error. When using Set preset during Auto pan, PS-API returns error.

When using Set Preset during Auto track, Auto track stops.

PS-API has the camera control status which is the result of controlled by itself. Therefore the behavior is different from the camera control status when a camera is controlled by other instance.

When using 360-degree Network Camera, please refer to "4.7 About 360-degree Network Camera".

---

**Sequence**

---

**6.5 CameraOperation**

---

**Sample program code**

---

[Visual C++ 2012] ..¥Sample Program¥PS-API¥Visual C++ 2012¥005\_CameraOperation

---

**Reference**

---



#### 5.7.1.5. GetCamOpStatus

**Class**                **IPSAPI**

**Method**            **GetCamOpStatus**

long      GetCamOpStatus(  
                                 long   channel  
                                 );

#### **Description**

Get the camera operation status such as auto track, auto pan.

#### **Argument**

channel	1 : Network Camera, GXE100 1 to 4 : Encoder NX100 1 to 16 : ND200, HD300, HD600/700 1 to 24 : NV200, NV250 1 to 32 : ND300, NV300, NX200, NX300 1 to 64 : ND400 1 to 128 : NX400	Specify the video channel.
---------	--	----------------------------

#### **Return value**

-1	Fail to get status.
0	not used
1	auto track
2	auto pan
9	Preset sequence
10	Auto sort
11	Patrol

#### **Error**

Get the error information by OnError event.

**Note**

---

In case that the target device is HD300, HD600/700 or encoder, the got status information isn't same as the device behavior because the behavior depends on the camera setting.

This method returns the status of camera control that is controlled from own instance. When the target camera is controlled by Web browser or other instances, the gotten status may be different from the device status.

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.7.1.6. CameraCentering

**Class**                      **IPSAPI**

**Method**                      **CameraCentering**

```
long      CameraCentering(
                long    channel,
                long    xPosition,
                long    yPosition,
                long    imageWidth,
                long    imageHeight
            );
```

#### Description

Control the camera to the specified position becomes the center.

#### Argument

channel	1 : Network Camera, GXE100 1 to 4 : Encoder NX100 1 to 16 : ND200, HD300, HD600/700 1 to 24 : NV200, NV250 1 to 32 : ND300, NV300, NX200, NX300 1 to 64 : ND400 1 to 128 : NX400	Specify the video channel.
xPosition	0 and over	Specified the x position that becomes center, when the top left corner of the displayed screen is as original position. xPosition value should be less than imageWidth value. (Pixel scale)
yPosition	0 and over	Specified the y position that becomes center, when the top left corner of the displayed screen is as original position. yPosition value should be less than imageHeight value. (Pixel scale)
imageWidth	0 and over	Specify the displayed screen width. (Pixel scale)
imageHeight	0 and over	Specify the displayed screen height. (Pixel scale)

#### Return value

0	Success
Except 0	Error code

---

**Error**

---

Error is defined by the return value.  
Get the error information by OnError event.

---

**Note**

---

The precision of centering depends on a target device.  
Depending on the camera model, a PoE + (compliant with IEEE802.3at) compatible hub or power supply device may be required. Please refer to the device manual.

When using digital zoom function, please specify the positions that is translated to the once position.

When controlling the analog camera via a HD300, HD600/700 or encoder, error will not occur though the target analog camera doesn't support the operations.

If specifying black panel area to xPosition and yPosition, error occurs.

When using 360-degree Network Camera, please refer to "4.7 About 360-degree Network Camera".

In the case of the X6533 and S6532, use a PoE+ hub / device that is compliant with IEEE802.3at standard when Return value = -32238707.

---

**Sequence**

---

---

**Sample program code**

---

---

**Reference**

---

### 5.7.1.7. CameraAuxControl

**Class**                **IPSAPI**

**Method**             **CameraAuxControl**

```
long CameraAuxControl(
    long channel,
    long alarmTrmNo,
    long command
);
```

#### **Description**

Control the AUX terminal of network camera and encoder to OPEN/CLOSE.

#### **Argument**

channel	1 : Network Camera 1to 4 : Encoder	Specify the video channel.
almTrmNo	0 : AUX terminal of network camera or encoder 1 : AUX terminal 1 of the camera that is connected with encoder 2 : AUX terminal 2 of the camera that is connected with encoder	Specify the controlled terminal number. In case of network camera, if this parameter is set to "1" or "2", it is ignored.
command	0 : CLOSE 1 : OPEN	Open or Close of the AUX terminal.

#### **Return value**

0	Success
Except 0	Error code

#### **Error**

Error is defined by the return value.  
Get the error information by OnError event.

**Note**

---

When controlling AUX terminal, the terminal alarm setting of network camera and encoder should be set to "AUX output". About the setup method, please refer to the operating Instructions for a target device.

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.7.1.8. GetCameraAuxStatus

**Class**            **IPSAPI**

**Method**           **GetCameraAuxStatus**

long    GetCameraAuxStatus(  
                                 long   channel,  
                                 long   alarmTrmNo,  
                                 );

#### **Description**

Get AUX terminal status of camera and encoder.

#### **Argument**

channel	1 : Network Camera 1to 4 : Encoder	Specify the video channel.
almtrmno	0 : AUX terminal of network camera or encoder	

#### **Return value**

-1	Fail to get status.
0	CLOSE
1	OPEN
2	AUX setting of camera is set to "OFF"
3	Status unsupported

#### **Error**

Error is defined by the return value.  
Get the error information by OnError event.

**Note**

---

In case of NT304 and NT314, it returns “3 : Status unsupported”.

**Sequence**

---

**Sample program code**

---

**Reference**

---



#### 5.7.1.9. SetCameraImageCap

**Class**                    **IPSAPI**

**Method**                **SetCameraImageCap**

long            SetCameraImageCap (  
                                  long   mode,  
                                  long   installation  
                                  );

#### **Description**

Set a Image capture mode and installation of camera.

#### **Argument**

mode	{SF438,SF448,SF458} 0 : 3M Fisheye 1 : 1.3M Fisheye 2 : 1.3M Single PTZ 3 : 1.3M Quad PTZ 4 : 2M Panorama 5 : 1M Panorama 6 : 2M Double Panorama 7 : 1M Double Panorama 8 : 4 Stream  [SFV481] 10 : 9M Fisheye 11 : 4M Fisheye 12 : Single PTZ 13 : Quad PTZ 14 : Panorama 15 : Double Panorama 16 : 4 Stream  [S4550,S4551] 20 : Fisheye 21 : Single PTZ 22 : Quad PTZ 23 : Panorama 24 : Double Panorama 25 : 4 Stream  [X4571、X4573] 30 : Fisheye 31 : Single PTZ 32 : Quad PTZ 33 : Panorama 34 : Double Panorama 35 : 4 Stream	Set a Image capture mode.
------	---	---------------------------

installation	0 : Ceiling 1 : Wall	Set a installation.
<b>Return value</b>		
0	Success	
Except 0	Error code	
<b>Error</b>		
	Error is defined by the return value. Get the error information by OnError event.	

## Note

Combination of Image Capture Mode and Installation Settings.  
{SF438,SF448,SF458}

Installation Mode	Ceiling	Wall
3M Fisheye	Yes	Yes
1.3M Fisheye	Yes	Yes
1.3M Single PTZ	Yes	Yes
1.3M Quad PTZ	Yes	Yes
2M Panorama	N/A	Yes
1M Panorama	N/A	Yes
2M Double Panorama	Yes	N/A
1M Double Panorama	Yes	N/A
4 Stream	Yes	N/A

[SFV481]

Installation Mode	Ceiling	Wall
9M Fisheye	Yes	Yes
4M Fisheye	Yes	Yes
Single PTZ	Yes	Yes
Quad PTZ	Yes	Yes
Panorama	N/A	Yes
Double Panorama	Yes	N/A
4 Stream	Yes	N/A

[S4550, S4551, X4571, X4573]

Installation Mode	Ceiling	Wall
Fisheye	Yes	Yes
Single PTZ	Yes	Yes
Quad PTZ	Yes	Yes
Panorama	N/A	Yes
Double Panorama	Yes	N/A
4 Stream	Yes	N/A

If SetCameraImageCap method is called during displaying the video images, the live video transmission from NW camera stops.

When setting image capture mode and installation, please stop playing the live with PlayControl method before SetCameraImageCap method is called.

## Sequence

## Sample program code

## Reference

#### 5.7.1.10. CameraWiperControl

**Class**                      **IPSAPI**

**Method**                      **CameraWiperControl**

```
long      CameraWiperControl(  
                                long    channel,  
                                long    command  
                            );
```

#### **Description**

Operate wiper functions of the camera.

#### **Argument**

channel	1 : Network Camera, 1 to 4 : NX100 1 to 24 : NV250 1 to 32 : NV300, NX200, NX300 1 to 64 : ND400 1 to 128 : NX400	Specify the video channel.
command	0 : Stop 1 : Low 2 : High 3 : 1 shot 4 : Washer	Specify the wiper operation mode.

#### **Return value**

0	Success
Except 0	Error code

#### **Error**

Error is defined by the return value.  
Get the error information by OnError event.

**Note**

---

SUD638 can be operated.

**Sequence**

---

**Sample program code**

---

**Reference**

---

## 5.7.2. Application Listener

### 5.7.2.1. OnOpStatus

Class	IAppListener
-------	--------------

Listener	OnOpStatus
----------	------------

void	OnOpStatus ( long channel, long status );
------	--

#### Description

Notify the camera operation status change to the specified application.  
Application needs to create the listener class that inherits IAppListener and to implement OnOpStatus method.

#### Argument

channel	1 : Network Camera, GXE100 1 to 4 : Encoder NX100 1 to 16 : ND200, HD300, HD600/700 1 to 24 : NV200, NV250 1 to 32 : ND300, NV300, NX200, NX300 1 to 64 : ND400 1 to 128 : NX400	Specify the video channel.
status	0 : No operation 1 : Auto track 2 : Auto pan 9 : Preset sequence 10 : Auto sort 11 : Patrol	Specify a device operation status value.

#### Return value

None

#### Error

**Note**

---

Please don't use PS-API functions in the notified listener function.

In case that the target device is HD300, HD600/700 or encoder, the got status information isn't same as the device behavior because the behavior depends on the camera setting.

**Sequence**

---

**Sample program code**

---

**Reference**

---

### 5.7.2.2. SetOpListener

**Class**                      **IPSAPI**

**Listener**                      **SetOpListener**

```
long      SetOpListener(  
                         IAppListener*   pReceiver  
                         );
```

#### **Description**

Set the instance of listener class that implement “5.7.2.1 OnOpStatus” .  
After setting the listener, PS-API can notify any camera operation status change by calling “5.7.2.1 OnOpStatus”, when it happens.

#### **Argument**

pReceiver	Pointer	Pointer for Listener class. Don't delete the registered object by your application.
	NULL	If pReceiver is set to NULL, PS-API unregister the Listener class.

#### **Return value**

0	Success
Except 0	Error code

#### **Error**

Error is defined by the return value.



**Note**

---

Don't delete the registered listener object by your application.

**Sequence**

---

**Sample program code**

---

**Reference**

---

### 5.7.3. Application Callback

#### 5.7.3.1. OnOpStatusCB

<b>Class</b>	<b>IAppCallBack</b>
--------------	---------------------

<b>Callback</b>	<b>OnOpStatusCB</b>
-----------------	---------------------

void	OnOpStatusCB (	long	status
		)	;

#### Description

Notify the camera operation status change to the specified application.  
Application needs to create the callback class that inherits IAppCallBack and to implement OnOpStatusCB method.

#### Argument

status	-1 : Fail to get status. 0 : No operation 1 : Auto track 2 : Auto pan 9 : Preset sequence 10 : Auto sort 11 : Patrol	Specify a device operation status value.
--------	--	--

#### Return value

None

#### Error

**Note**

---

Please don't use PS-API functions in the notified callback function.

In case that the target device is HD300, HD600/700 or encoder, the got status information isn't same as the device behavior because the behavior depends on the camera setting.

**Sequence**

---

**Sample program code**

---

**Reference**

---

## 5.8. Alarm Group

### 5.8.1. Method

#### 5.8.1.1. AlmOperation

Class	IPSAPI
Method	AlmOperation
long	AlmOperation( long channel, long command, long& status, long mode, IAppCallBack* pSender );

#### Description

Reset current device alarm state.  
Or trigger a device alarm by the application.  
Trigger operation is valid to only NWDR and NX Series.  
Not used and alarm reset is valid to all devices.

#### Argument

channel	[Trigger ON/OFF] 1 to 16 : ND200, HD600/700 1 to 24 : NV200 1 to 32 : ND300 1 to 64 : ND400 1 to 128 : NV300, NV250, NX100, NX200, NX300, NX400  [Alarm reset] 1 : Network Camera, GXE100 1 to 4 : Encoder 1 to 16 : ND200, HD300 HD600/700 1 to 24 : NV200 1 to 32 : ND300 1 to 64 : ND400 1 to 128 : NV300, NV250, NX100, NX200 NX300, NX400	Specify the video channel.
command	0 : not used. 1 : alarm reset 2 : trigger on 3 : trigger off all channels 4 : trigger off specified channel	

---

**Argument**

status	-1 : Fail to get status. 0 : Reset 1 : Active 2 : Status unsupported (In case of network camera or encoder)	In case of blocking mode, the current status is set by PS-API.
mode	0 : blocking Except 0 : Non-blocking	Blocking mode.
pSender	Callback interface	Specify a callback interface for the Alarm operation. Please refer to application callback section and look for function OnAlmStatusCB() for details. Don't delete the registered object by your application.

---

**Return value**

0	Success
Except 0	Error code

---

**Error**

Error is defined by the return value.  
Get the error information by OnError event.

---

**Note**

**Don't delete the registered callback object by your application.**

•In case of command=1/3/4 is executed ,Alarm and Event Recording status will be below.

command	Alarm Status	Event Recording *1
1	Reset	Turn off recording all channels
3	Reset	Turn off recording all channels
4	Reset	Turn off recording specified channel

\*1 Only Event Recording Turn off

---

**Sequence****6.6 AlmOperation**

---

**Sample program code**

[Visual C++ 2012] ..¥Sample Program¥PS-API¥Visual C++ 2012¥006\_AlmOperation

---

**Reference**

---

#### 5.8.1.2. GetAlarmStatus

<b>Class</b>	<b>IPSAPI</b>
--------------	---------------

<b>Method</b>	<b>GetAlarmStatus</b>
---------------	-----------------------

long	GetAlarmStatus( );
------	--------------------

---

#### Description

Get current alarm information.

---

#### Argument

None

---

#### Return value

-1	Fail to get status.
0	Reset
1	Active
2	Status unsupported (In case of Network Camera or Encoder)

---

#### Error

Get the error information by OnError event.

**Note**

---

**Sequence**

---

**Sample program code**

---

**Reference**

---

## 5.8.2. Application Listener

### 5.8.2.1. OnAlmStatus

Class	IAppListener
Listener	OnAlmStatus
void	OnAlmStatus ( <div>             long channel,              long type,              char* timeDate,              long status              );           </div>

#### Description

Notify the alarm status change to the specified application.  
 Application needs to create the listener class that inherits IAppListener and to implement OnAlmStatus method.

#### Argument

channle	0 : No channel 1 : Network Camera, GXE100 1 to 4 : Encoder 1 to 16 : ND200, HD300, HD600/700 1 to 24 : NV200 1 to 32 : ND300 1 to 64 : ND400 1 to 128 : NV300, NV250, NX100, NX200, NX300, NX400 1 to 32 : Terminal number	Specify a video channel or terminal number for the device.
type	0 : Terminal alarm 1 : Video loss alarm 2 : VMD alarm 3 : Command alarm 4 : Other	Specify an alarm trigger type.
tiemDate	Local time	Local time when an alarm trigger happens.
status	0 : Reset 1 : Active 2 : Status unsupported (In case of Network Camera or Encoder)	Specify the current alarm operation state.



**Return value**

---

None

**Error**

---

**Note**

---

Please don't use PS-API functions in the notified listener function.

In case of Network camera or encoder, even if an alarm occurs or alarm is reset, OnAlmStatus will not occur because alarm status information isn't supported.

In case of HD300, OnAlmStatus will not occur by emergency events.

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.8.2.2. SetAlmListener

**Class**                      **IPSAPI**

**Listener**                      **SetAlmListener**

```
long      SetAlmListener(  
                            IAppListener*   pReceiver  
                            );
```

#### **Description**

Set the instance of listener class that implement “5.8.2.1 OnAlmStatus” .  
After setting the listener, PS-API can notify any alarm information by calling “5.8.2.1 OnAlmStatus”, when it happens.

#### **Argument**

pReceiver	Pointer	Pointer for Listener class. Don't delete the registered object by your application.
	NULL	If pReceiver is set to NULL, PS-API unregister the Listener class.

#### **Return value**

0	Success
Except 0	Error code

#### **Error**

Error is defined by the return value.

**Note**

---

**Don't delete the registered listener object by your application.**

In case of Network camera or encoder, OnAlmStatus will not occur because alarm status information isn't supported.

**Sequence**

---

**Sample program code**

---

**Reference**

---

### 5.8.3. Application Callback

#### 5.8.3.1. OnAlmStatusCB

**Class**            **IAppCallBack**

**Callback**        **OnAlmStatusCB**

```
void      OnAlmStatusCB(  
                                long   status  
                                );
```

#### **Description**

Notify the alarm status change to the specified application.  
Application needs to create the callback class that inherits IAppCallBack and to implement OnAlmStatusCB method.

#### **Argument**

status	-1 : Fail to get status. 0 : Reset 1 : Active 2 : Status unsupported (In case of Network Camera or Encoder)	Specify the current alarm operation state.
--------	--	--

#### **Return value**

None

#### **Error**

**Note**

---

Please don't use PS-API functions in the notified callback function.

**Sequence**

---

**Sample program code**

---

**Reference**

---

## 5.9. FTP Group

### 5.9.1. Method

#### 5.9.1.1. FtpGet

Class	IPSAPI
Method	FtpGet
long	FtpGet( long channel, char* startTimeDate, char* endTimeDate, long dataType, long eventType, char* fileName, long& status, long mode, IAppCallBack* pSender );

#### Description

Get video, audio and recording event information data from a target device via FTP.

This method does not work with NX Series V5.00 and later versions.  
Please use "5.5.1.44 HttpDownload".

#### Argument

channel	1 to 4 : NX100 1 to 16 : ND200, HD300, HD600/700 1 to 24 : NV200, NV250 1 to 32 : ND300, NV300, NX200, NX300 1 to 64 : ND400 1 to 128 : NX400	Specify the video channel.
startTimeDate	yyyy/mm/dd hh:mm:ss	Download the recording data that started recording after the specified time. The strings must be terminated with null character.
endTimeDate	yyyy/mm/dd hh:mm:ss	Download the recording data that start recording before the specified time. Please specify the duration between startTimeDate and endTimeDate within 30 minutes(except for NX Series) or 60 minutes(in case of NX Series). The strings must be terminated with null character.
dataType	0 : Video 1 : Video and Audio 2 : Recording event information	Specify the downloaded data type



## Argument

eventType

Bit 0 to Bit 14 :

Specify the recording event kind by 32 digit binary.

- Bit 0 : Emergency (EMR)
- Bit 1 : Manual (MAN)
- Bit 2 : Schedule (SCH)
- Bit 3 : Terminal (TRM)
- Bit 4 : Command alarm (COM)
- Bit 5 : Camera site alarm (CAM/SITE)  
\* In case of HD300, Bit5 is VMD.
- Bit 6 : SD Backup (SD)
- Bit 7 : Video Loss (LOSS)
- Bit 8 : VMD alarm (VMD)
- Bit 9 : Motion alarm (CMTN)
- Bit 10 : Loitering alarm (CSTY)
- Bit 11 : Removal alarm (CRMV)
- Bit 12 : Scene change alarm (CSCD)
- Bit 13 : Terminal alarm (CTRM)
- Bit 14 : Direction alarm (CDRT)
- Bit 15 to Bit 32 : Reserved

Bit	HD300	ND200 ND300 ND400	NV200 NV250 NV300	HD600 HD700	NX100 NX200 NX300 NX400
0	EMR	EMR	-	EMR	EMR
1	MAN	MAN	-	MAN	-
2	SCH	SCH	SCH	SCH	SCH
3	TRM	TRM	TRM	TRM	TRM
4	COM	COM	COM	COM	COM
5	(VMD)	CAM	CAM	SITE	CAM
6	-	SD	-	-	SD
7	LOSS	-	-	LOSS	-
8	VMD	-	-	VMD	-
9	-	-	-	CMTN	-
10	-	-	-	CSTY	-
11	-	-	-	CRMV	-
12	-	-	-	CSCD	-
13	-	-	-	CTRM	-
14	-	-	-	CDRT	-

Bit map format

MSB	-	-	...	-	-	-	LSB
0	0	0	...	0	0	0	0
Bit	Bit	Bit	...	Bit	Bit	Bit	Bit
31	30	29	...	3	2	1	0

e.g. Emergency + Terminal  
[Binary] 0001001 ---> type = 9

Except for NX Series, SD backup search and the other type search cannot be used together at the same time.  
If specifying "1" to the other bits, SD backup are ignored.  
In case of NWDR and NX Series, Bit7 is ignored if it is set to "1".  
In case of HD300, Bit6 is ignored if it is set to "1".  
In case of HD300, Bit5 or Bit8 is VMD alarm event search.



---

**Argument**

---

fileName	Character strings (251 byte or less)	Specify a complete file path, and file name without extension for storing the downloaded video. <b>An extension is added automatically.</b> The strings must be terminated with null character.
status	-1 : Fail to get status. 0 : No getting data via ftp. 1 : In getting data via ftp	
mode	0 : blocking Except 0 : Non-blocking	Blocking mode.
pSender	Callback interface	Specify a callback interface for the port control. Please refer to application callback section and look for function OnFtpStatusCB() for details. Don't delete the registered object by your application.

---

**Return value**

---

0	Success
Except 0	Error code

---

**Error**

---

Error is defined by the return value.  
Get the error information by OnError event.

---

**Note**

---

---

**Don't delete the registered callback object by your application.**

While executing FtpGet method with non-blocking mode, FtpGet method cannot be called from the same instance. The second FTP download starts after the first FTP download finished.

By executing FtpGet method, FTP server mode of a target device turns on. If you want to turn off FTP server mode, please call FtpServerClose method.

If there is no recording data in a target device between the specified times, PS-API creates an empty file.

If there is the same name file on the specified complete file path, it is overwritten.

When TRM/CMD/CAM/SITE/VMD/LOSS/CMTN/CSTY/CRMV/CSCD/CTRM/CDRT are specified in FTP download condition, all event pre recording data that exist between the specified time are included in the downloaded file.

In case of ND400, FTP connection is disconnected after 1 hour from starting FTP download and the unfinished file is deleted.

In case of HD600/700, the start time of downloaded data is before 12 seconds of the specified startTimeDate time.

The size of one downloaded file should be up to 1.86 GByte.

When downloading the recorded video which is high resolution/bitrate, please set the duration to be short .

**It is necessary that “5.3.2.1 DeviceType”, “5.3.2.2 IPAddr”, “5.3.2.3 HttpPort”, “5.3.2.9 UserName” and “5.3.2.10 Password” are specified to download the recording data via FTP from HD300, NWDR, HD600/700 or NX Series.**

During FTP download form NX Series, FTP download may fail due to the effect of standby HDD.  
(FTP 100 Response: 551 Error on input file: Input/output error.) Please re-execute.

---

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.9.1.2. FtpCancel

<b>Class</b>	<b>IPSAPI</b>
--------------	---------------

<b>Method</b>	<b>FtpCancel</b>
---------------	------------------

long	FtpCancel();
------	--------------

#### Description

Cancel the ftp download.

This function doesn't disconnect from a target device.

If needs to log out, please call Close method or Disconnect method.

#### Argument

None

#### Return value

0	Success to regist an application listener.
---	--

Except 0	Error code
----------	------------

#### Error

Error is defined by the return value.

Get the error information by OnError event.

**Note**

---

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.9.1.3. FtpServerClose

**Class**                      **IPSAPI**

**Method**                    **FtpServerClose**

long      FtpServerClose();

---

#### **Description**

Turn off the FTP server mode of a target device.

---

#### **Argument**

None

---

#### **Return value**

0                      Success

Except 0            Error code

---

#### **Error**

Error is defined by the return value.  
Get the error information by OnError event.

**Note**

---

The device has one FTP server mode.

When the FTP server mode is turned off from one instance, other instances stop downloading even if they are getting data.

If calling FtpServerClose method without login to a target device(UID=-1), the PS-API instance gets a UID from PS-API in during changing the FTP server mode.

HD300, ND400, ND300, NV200, NV250, NV300, NX Series and HD600/700 can't turn off the FTP server mode by FtpServerClose() method.

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.9.1.4. GetFtpStatus

<b>Class</b>	<b>IPSAPI</b>
--------------	---------------

<b>Method</b>	<b>GetFtpStatus</b>
---------------	---------------------

long	GetFtpStatus();
------	-----------------

#### Description

Get current ftp downloading status.

#### Argument

None

#### Return value

-1	Fail to get status.
0	No getting data via ftp.
1	In getting data via ftp

#### Error

Get the error information by OnError event.

**Note**

---

**Sequence**

---

**Sample program code**

---

**Reference**

---



#### 5.9.1.5. GetFtpTransRate

**Class**                    **IPSAPI**

**Method**                **GetFtpTransRate**

long      GetFtpTransRate();

---

#### **Description**

Get current transmission speed [byte/s] of FTP downloading.

---

#### **Argument**

None

---

#### **Return value**

-1	Fail to get status.
0 or more	Transfer rate [byte/s]

---

#### **Error**

Get the error information by OnError event.

**Note**

---

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.9.1.6. GetFtpTransByte

**Class**                    **IPSAPI**

**Method**                **GetFtpTransByte**

long      GetFtpTransByte();

#### **Description**

Get the amount of transferred data [byte] from a target device.

#### **Argument**

None

None

#### **Return value**

-1

Fail to get status.

0 or more

Transferred bytes. [byte]

#### **Error**

Get the error information by OnError event.

**Note**

---

**Sequence**

---

**Sample program code**

---

**Reference**

---

## 5.9.2. Property

### 5.9.2.1. FtpPort

<b>Class</b>	<b>IPSAPI</b>
--------------	---------------

<b>Property</b>	<b>FtpPort</b>
-----------------	----------------

long	SetFtpPort( long port );
------	--------------------------------

long	GetFtpPort();
------	---------------

#### Description

Set FTP server port number of a target device into PS-API.  
Get FTP server port number of a target device from PS-API.

#### Argument for SET

port	1 to 65535	Ftp server port.
		Default value is 21.

#### Return value for SET

0	Success
Except 0	Error code

#### Return value for GET

Get FTP server port number for a target device.

#### Error

**Note**

---

It is necessary that “5.3.2.1 DeviceType”, “5.3.2.2 IPAddr”, “5.3.2.3 HttpPort”, “5.3.2.9 UserName” and “5.3.2.10 Password” are specified to download the recording data via FTP from HD300, NWDR, HD600/700 or NX Series.

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.9.2.2. FtpTransMode

**Class**            **IPSAPI**

---

**Property**        **FtpTransMode**

---

long      SetFtpTransMode(  
                                 long   transMode  
                                 );

long      GetFtpTransMode();

---

#### **Description**

Set ftp transmission mode into PS-API.  
Get ftp transmission mode from PS-API.

---

#### **Argument for SET**

ftpMode	0 : passive	Ftp transmission mode.
	1 : active	
		Default value is 0.

---

#### **Return value for SET**

0	Success
Except 0	Error code

---

#### **Return value for GET**

Get ftp transmission mode.

---

#### **Error**

---

**Note**

---

**Sequence**

---

**Sample program code**

---

**Reference**

---



### 5.9.3. Application Callback

#### 5.9.3.1. OnFtpStatusCB

<b>Class</b>	<b>IAppCallBack</b>
--------------	---------------------

<b>Callback</b>	<b>OnFtpStatusCB</b>
-----------------	----------------------

void	OnFtpStatusCB (	long	status
		)	;

#### Description

Notify the completion of download to the specified application.  
Application needs to create the callback class that inherits IAppCallBack and to implement OnFtpStatusCB method.

#### Argument

status	-1 : Fail to get status.	FTP download status
	0 : No getting data via ftp.	
	1 : In getting data via ftp	

#### Return value

None

#### Error

**Note**

---

Please don't use PS-API functions in the notified callback function.

**Sequence**

---

**Sample program code**

---

**Reference**

---

## 6. Operation Procedure and Sequence

### 6.1. PlayLive

#### 6.1.1. Operation Procedure

##### Start Live

No.	Property / Method	Parameter	Description
1	GetIPSAPI	-	Create instance of IPSAPI. 1 instance can control 1 target device.
2	SetIPAddr	IP Address (char*)	Set IP Address into PS-API.  e.g.) "192.168.0.10"
3	SetDeviceType	Device type (long)	Set the device type corresponding to a target device.  e.g.) 2
4	SetHttpPort	Port number (long)	Set Http port number into PS-API.  e.g.) 80
5	SetUserName	Character strings (char*)	Set user name into PS-API.  e.g.) admin
6	SetPassword	Character strings (char*)	Set password into PS-API.  e.g.) 12345
7	SetVideoWindow	WindowHandle (hWnd)	Set WinowHandle to display the video image.  e.g.) this->m_hWnd
8	SetImageWidth	Display image width (long)	Set the image width to display.  e.g.) 640
9	SetImageHeight	Display image height (long)	Set the image height to display.  e.g.) 480

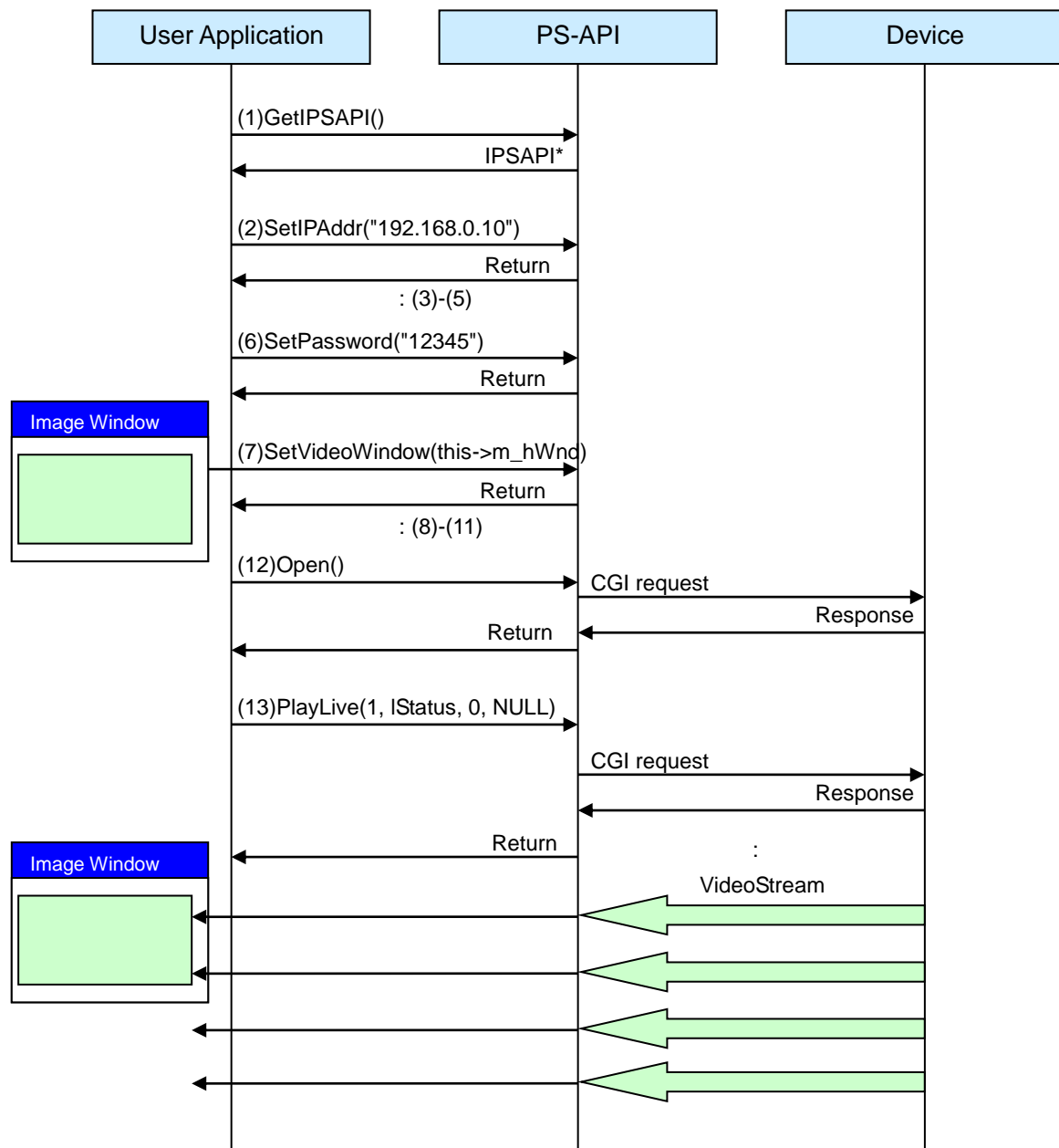
No.	Property / Method	Parameter	Description
10	SetStreamFormat	Stream type (long)	Set the stream type into PS-API. In case of NWDR, set the stream type that is same with a target channel that is wanted to display.  e.g.) 0
11	SetJPEGResolution/ SetMPEG4Resolution/ SetH264Resolution	Resolution (long)	It is necessary that the resolution setting is same value of the device setting.  e.g.) 640
12	Open/Connect	UID (long)	Connect to a target device. When getting a new UID, please use Open method.
13	PlayLive	Channel, Status, Blocking mode, Callback interface (long, long&, long, IAppCallBack*)	Start displaying live image. In case of network camera, please set the channel to "1". In case of NWDR, NX Series, HD600/700 or HD300, please set the channel to number that is wanted to display. Set the status to a long type variable . When using blocking mode, it is necessary that callback interface is set to "NULL". When using non-blocking mode, please set the callback interface to an instance to receive the result of async processing. (The callback interface can be set to "NULL", but application cannot receive the notification of finish processing.)  e.g.) 1, IStatus, 0, NULL

### Stop Live

No.	Property / Method	Parameter	Description
14	PlayControl	Command, Speed, Status, Blocking mode, Callback interface (long, long, long&, long, IAppCallBack*)	To stop live, set command to "1". Set speed to "1". Set the status to a long type variable . When using blocking mode, it is necessary that callback interface is set to "NULL". When using non-blocking mode, please set the callback interface to an instance to receive the result of async processing. (The callback interface can be set to "NULL", but application cannot receive the notification of finish processing.)  e.g.) 1, 1, IStatus, 0, NULL
15	Close/Disconnect	-	Stop the communication with the target device. When using Close method, UID will be annulled.
16	DeleteIPSAPI	Pointer of IPSAPI (IPSAPI*)	Delete the IPSAPI instance.

## 6.1.2. Sequence

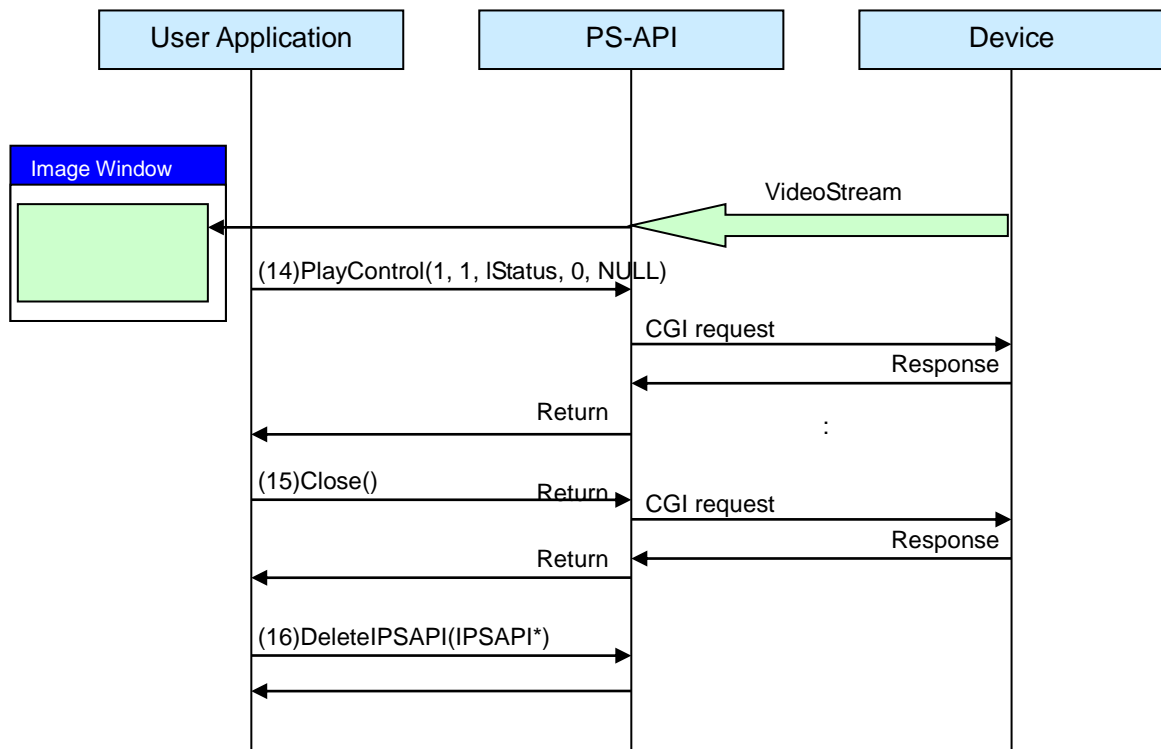
### Start Live



\* The timing of sending CGI command is not constant, depends on the device model or the device setting.

**Figure 6-1 Start Live**

## Stop Live



\* The timing of sending CGI command is not constant, depends on the device model or the device setting.

**Figure 6-2 Stop Live**

## 6.2. Play

### 6.2.1. Operation Procedure

#### Create the IPSAPI Instance and Login

No.	Property / Method	Parameter	Description
1	GetIPSAPI	-	Create instance of IPSAPI. 1 instance can control 1 target device.
2	SetIPAddr	IP Address (char*)	Set IP Address into PS-API.  e.g.) "192.168.0.250"
3	SetDeviceType	Device Type (long)	Set the device type corresponding to a target device.  e.g.) 1
4	SetHttpPort	Port number (long)	Set Http port number into PS-API.  e.g.) 80
5	SetUserName	Character strings (char*)	Set user name into PS-API.  e.g.) ADMIN
6	SetPassword	Character strings (char*)	Set password into PS-API.  e.g.) 12345
7	SetVideoWindow	WindowHandle (hWnd)	Set WinowHandle to display the video image.  e.g.) this->m_hWnd
8	SetImageWidth	Display image width (long)	Set the image width to display.  e.g.) 640
9	SetImageHeight	Display image height (long)	Set the image height to display.  e.g.) 480
10	SetStreamFormat	Stream type (long)	Set the stream type into PS-API. In case of NWDR, set the stream type that is same with a target channel that is wanted to display.  e.g.) 0



No.	Property / Method	Parameter	Description
11	SetJPEGResolution/ SetMPEG4Resolution/ SetH264Resolution	Resolution (long)	It is necessary that the resolution setting is same value of the device setting.  e.g.) 640
12	Open/Connect	UID (long)	Connect to a target device. When getting a new UID, please use Open method.

#### Start Recorded Video Play

No.	Property / Method	Parameter	Description
13	Play	Channel, Time and Data, Status, Blocking mode, Callback interface (long, char* long& long, IAppCallBack*)	Start displaying the recorded video image. In case of NWDR, NX Series, HD600/700 or HD300, please set the channel to optional number that is wanted to display. Specify the date that is wanted to play recorded video. Set the status to a long type variable . When using blocking mode, it is necessary that callback interface is set to "NULL". When using non-blocking mode, please set the callback interface to an instance to receive the result of async processing. (The callback interface can be set to "NULL", but application cannot receive the notification of finish processing.)  e.g.) 1, "2009/01/01 12:50:00", IStatus, 0, NULL

### Pause

No.	Property / Method	Parameter	Description
14	PlayControl	Command, Speed, Status, Blocking mode, Callback interface (long, long, long&, long, IAppCallBack*)	To pause the recorded video play, set command to "3". Set speed to "1". Set the status to a long type variable . When using blocking mode, it is necessary that callback interface is set to "NULL". When using non-blocking mode, please set the callback interface to an instance to receive the result of async processing. (The callback interface can be set to "NULL", but application cannot receive the notification of finish processing.)  e.g.) 3, 1, IStatus, 0, NULL

### Restart the network playback (or Backward)

No.	Property / Method	Parameter	Description
15	PlayControl	Command, Speed, Status, Blocking mode, Callback interface (long, long, long&, long, IAppCallBack*)	When restart the network playback, set the command to "4". When start Backward, set the command to "5". Set the playback speed to a long type variable. Set the status to a long type variable . When using blocking mode, it is necessary that callback interface is set to "NULL". When using non-blocking mode, please set the callback interface to an instance to receive the result of async processing. (The callback interface can be set to "NULL", but application cannot receive the notification of finish processing.)  e.g.) 5, 1, IStatus, 0, NULL

### Start the Fast Forward (or Rewind)

No.	Property / Method	Parameter	Description
16	PlayControl	Command, Speed, Status, Blocking mode, Callback interface (long, long, long&, long, IAppCallBack*)	When start Fast Forward, set the command to "8". When start Rewind, set the command to "9". Set speed to "1". Set the status to a long type variable . When using blocking mode, it is necessary that callback interface is set to "NULL". When using non-blocking mode, please set the callback interface to an instance to receive the result of async processing. (The callback interface can be set to "NULL", but application cannot receive the notification of finish processing.)  e.g.) 8, 1, IStatus, 0, NULL

**Display the next frame (or the previous frame)**

No.	Property / Method	Parameter	Description
17	PlayControl	Command, Speed, Status, Blocking mode, Callback interface (long, long, long&, long, IAppCallBack*)	When display the next frame, set the command to "6". When display the previous frame, set the command to "7". Set speed to "1". Set the status to a long type variable . When using blocking mode, it is necessary that callback interface is set to "NULL". When using non-blocking mode, please set the callback interface to an instance to receive the result of async processing. (The callback interface can be set to "NULL", but application cannot receive the notification of finish processing.)  e.g.) 6, 1, IStatus, 0, NULL

**Stop Recorded Video Play**

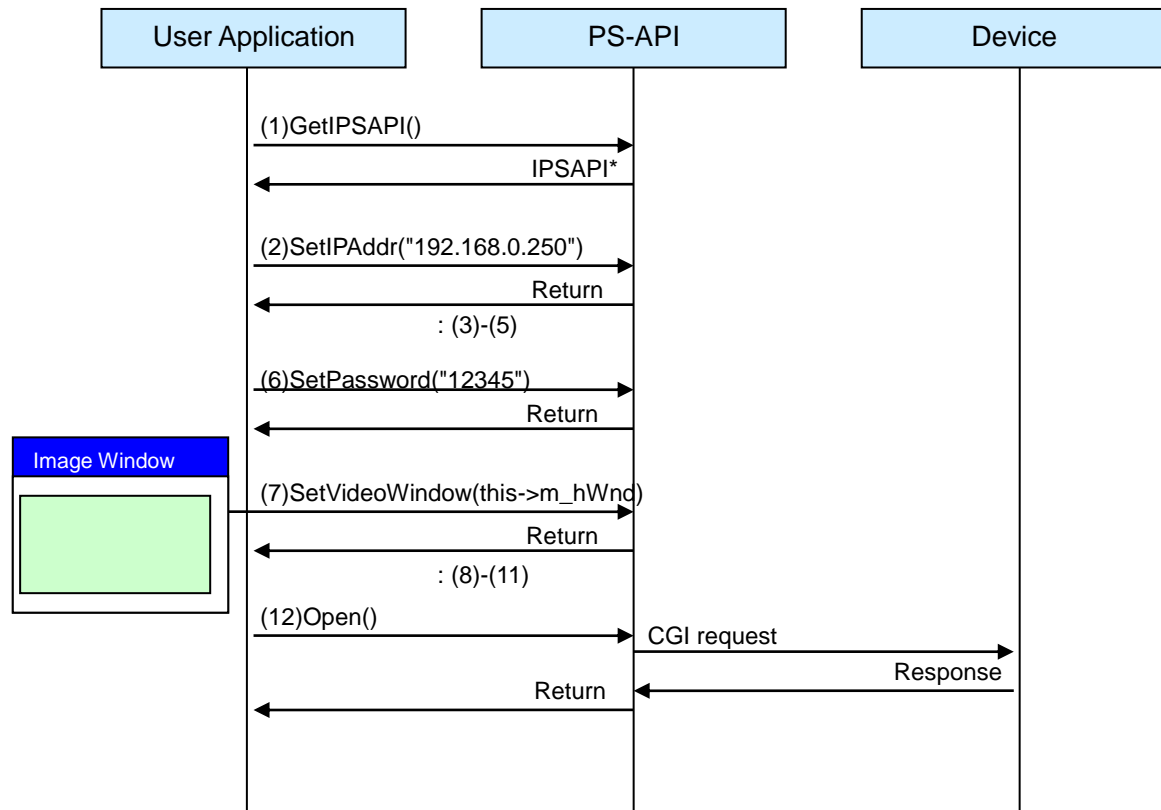
No.	Property / Method	Parameter	Description
18	PlayControl	Command, Speed, Status, Blocking mode, Callback interface (long, long, long&, long, IAppCallBack*)	When stop the network playback, set the command to "0". Set speed to "1". Set the status to a long type variable . When using blocking mode, it is necessary that callback interface is set to "NULL". When using non-blocking mode, please set the callback interface to an instance to receive the result of async processing. (The callback interface can be set to "NULL", but application cannot receive the notification of finish processing.)  e.g.) 0, 1, IStatus, 0, NULL

**Logout and Delete the IPSAPI Instance**

No.	Property / Method	Parameter	Description
19	Close/Disconnect	-	Stop the communication with the target device. When using Close method, UID will be annulled.
20	DeleteIPSAPI	Pointer of IPSAPI (IPSAPI*)	Delete the IPSAPI instance.

## 6.2.2. Sequence

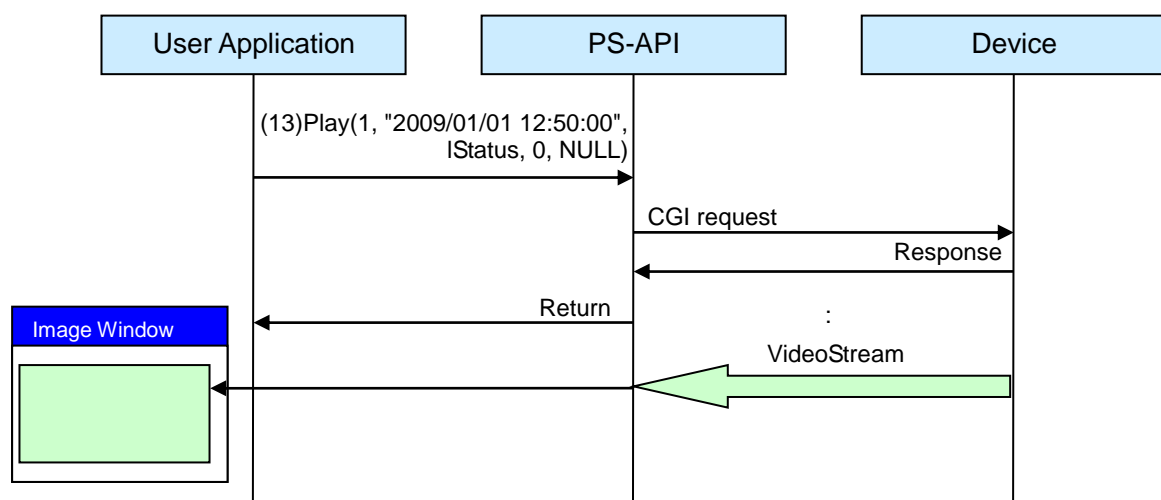
### Create the IPSAPI Instance and Login



\* The timing of sending CGI command is not constant, depends on the device model or the device setting.

**Figure 6-3 Create the IPSAPI Instance and Login**

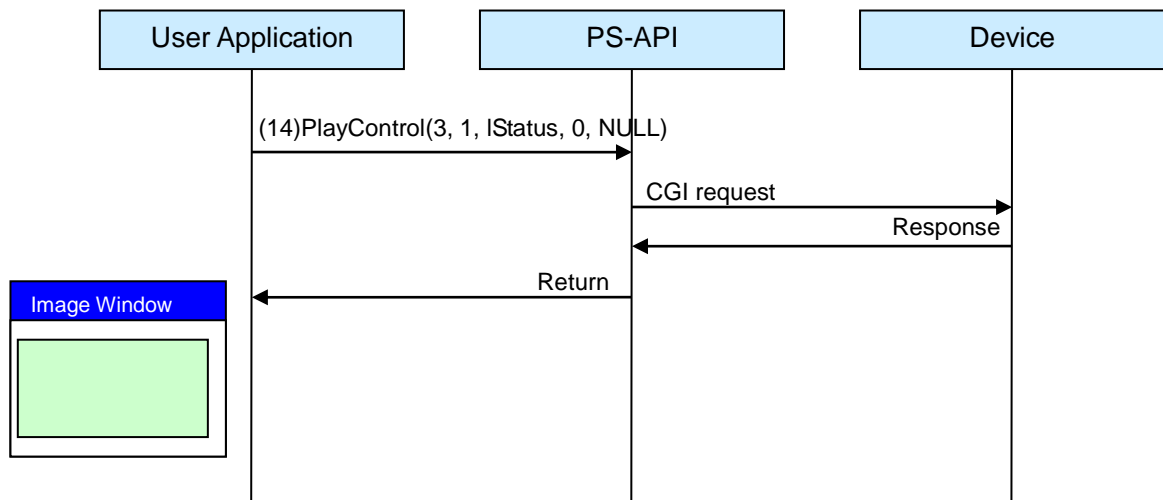
### Start Recorded Video Play



\* The timing of sending CGI command is not constant, depends on the device model or the device setting.

**Figure 6-4 Start Recorded Video Play**

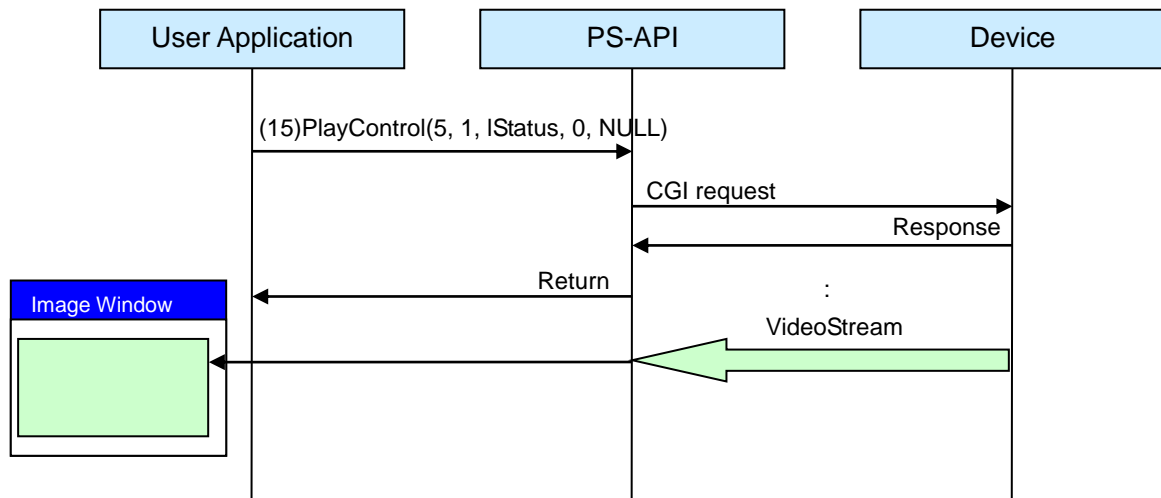
## Pause



\* The timing of sending CGI command is not constant, depends on the device model or the device setting.

**Figure 6-5 Pause**

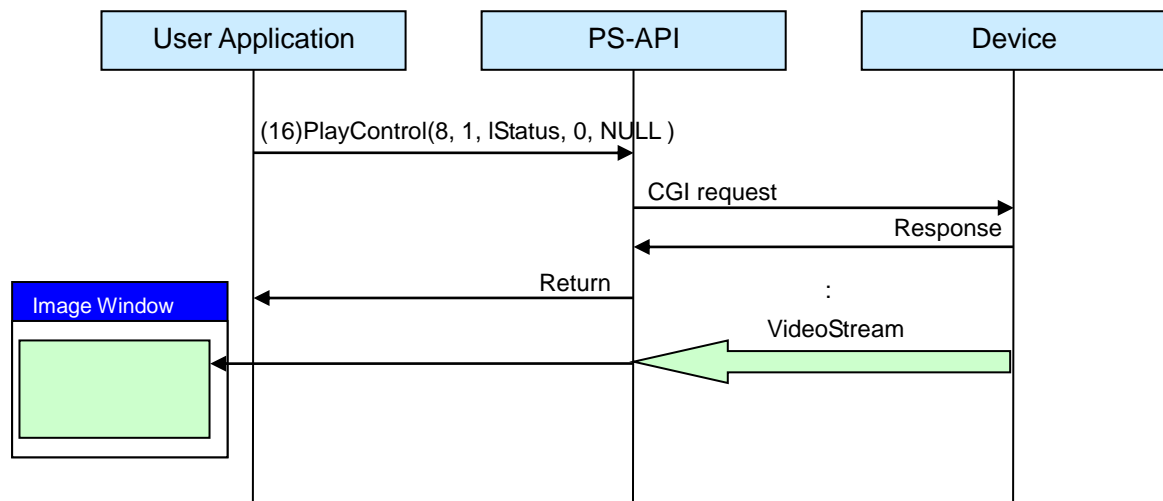
## Restart the network playback (or Backward)



\* The timing of sending CGI command is not constant, depends on the device model or the device setting.

**Figure 6-6 Restart the network playback (or Backward)**

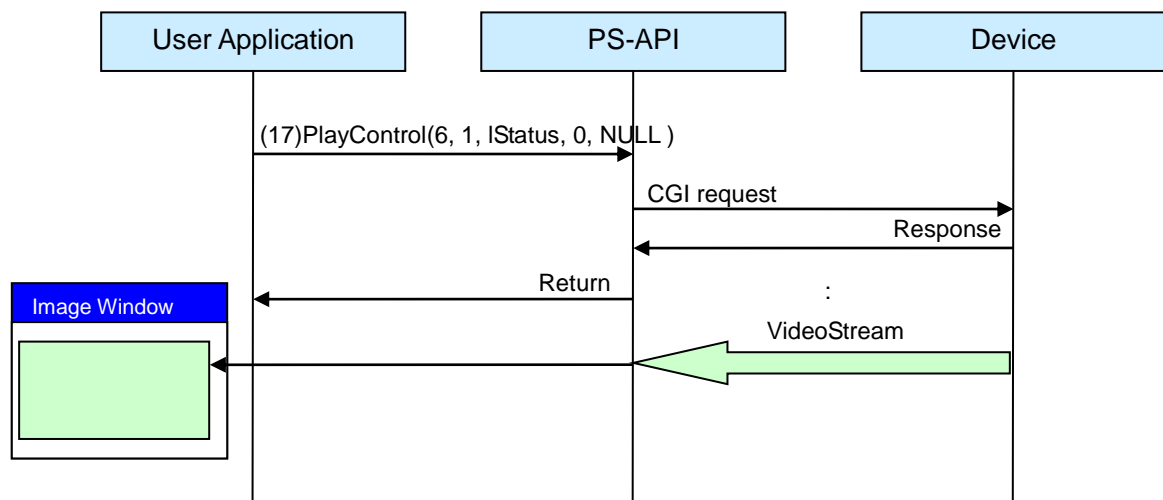
### Start the Fast Forward (or Rewind)



\* The timing of sending CGI command is not constant, depends on the device model or the device setting.

**Figure 6-7 Start the Fast Forward (or Rewind)**

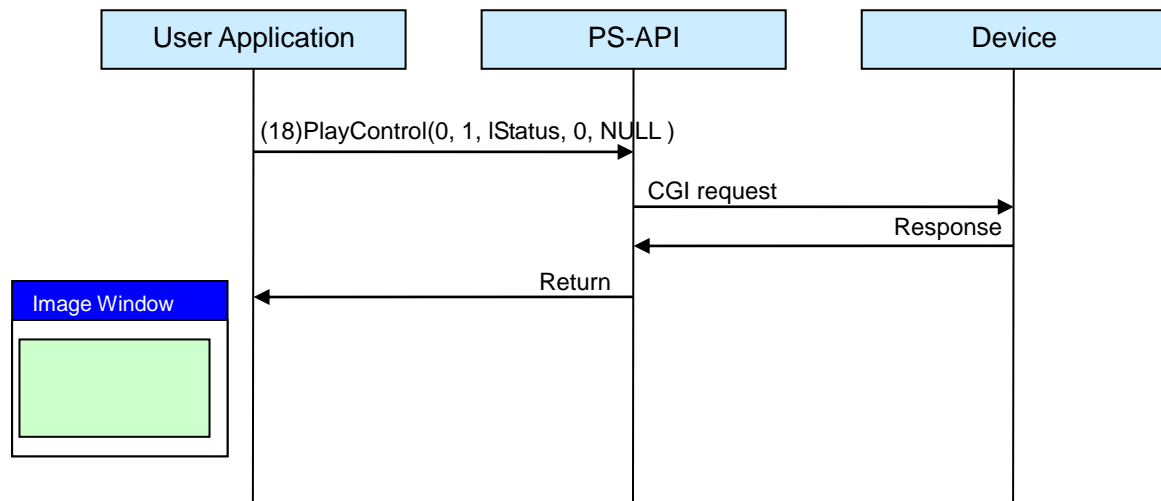
### Display the next frame (or the previous frame)



\* The timing of sending CGI command is not constant, depends on the device model or the device setting.

**Figure 6-8 Display the next frame (or the previous frame)**

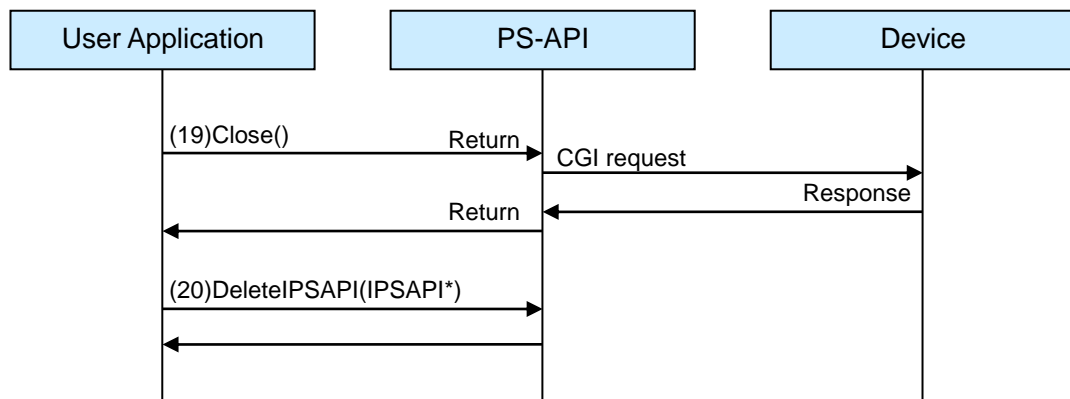
### Stop Recorded Video Play



\* The timing of sending CGI command is not constant, depends on the device model or the device setting.

**Figure 6-9 Stop Recorded Video Play**

### Logout and Delete the IPSAPI Instance



\* The timing of sending CGI command is not constant, depends on the device model or the device setting.

**Figure 6-10 Logout and Delete the IPSAPI Instance**

### 6.3. PlayFile

#### 6.3.1. Operation Procedure

##### Create the IPSAPI Instance

No.	Property / Method	Parameter	Description
-	<b>Create the IPSAPI Instance (Refer to 6.2 Play)</b>		

##### Start local file playback

No.	Property / Method	Parameter	Description
1	SetFilePassword	Password (char*)	Set the password to playback the video file that has password.  e.g.) ""
2	PlayFile	File Name, Status, Blocking mode, Callback interface (char* long& long, IAppCallBack*)	Start local file playback. Specify the file name with full path. Set the status to a long type variable . When using blocking mode, it is necessary that callback interface is set to "NULL". When using non-blocking mode, please set the callback interface to an instance to receive the result of async processing. (The callback interface can be set to "NULL", but application cannot receive the notification of finish processing.)  e.g.) "c:\¥¥filesample.n3r", IStatus, 0, NULL

##### Pause

No.	Property / Method	Parameter	Description
-	<b>Pause (Refer to 6.2Play)</b>		



**Restart local file playback (or Backward)**

No.	Property / Method	Parameter	Description
-	<b>Restart local file playback (or Backward)</b> (Refer to 6.2 Play)		

**Start Fast Forward (or Rewind)**

No.	Property / Method	Parameter	Description
-	<b>Start Fast Forward (or Rewind)</b> (Refer to 6.2 Play)		

**Display the next frame (or the previous frame)**

No.	Property / Method	Parameter	Description
-	<b>Display the next frame (or the previous frame)</b> (Refer to 6.2 Play)		

**Stop local file playback**

No.	Property / Method	Parameter	Description
3	PlayControl	Command, Speed, Status, Blocking mode, Callback interface (long, long, long&, long, IAppCallBack*)	When stop the local file playback, set the command to "2". Set speed to "1". Set the status to a long type variable . When using blocking mode, it is necessary that callback interface is set to "NULL". When using non-blocking mode, please set the callback interface to an instance to receive the result of async processing. (The callback interface can be set to "NULL", but application cannot receive the notification of finish processing.)  e.g.) 2, 1, IStatus, 0, NULL

**Delete the IPSAPI Instance**

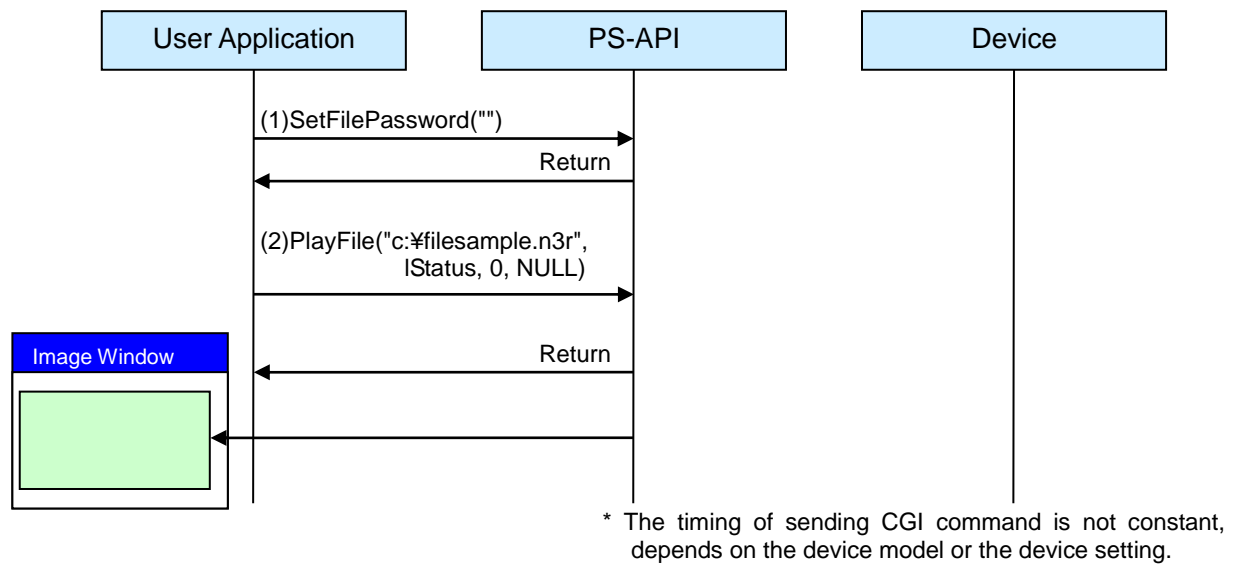
No.	Property / Method	Parameter	Description
-	<b>Delete the IPSAPI Instance</b> (Refer to 6.2 Play)		

### 6.3.2. Sequence

#### Create the IPSAPI Instance

Refer to 6.2 Play, Create the IPSAPI Instance

#### Start local file playback



**Figure 6-11 Start Local File Playback**

#### Pause

Refer to 6.2 Play, Pause

#### Restart local file playback (or Backward)

Refer to 6.2 Play, Restart the network playback (or Backward)

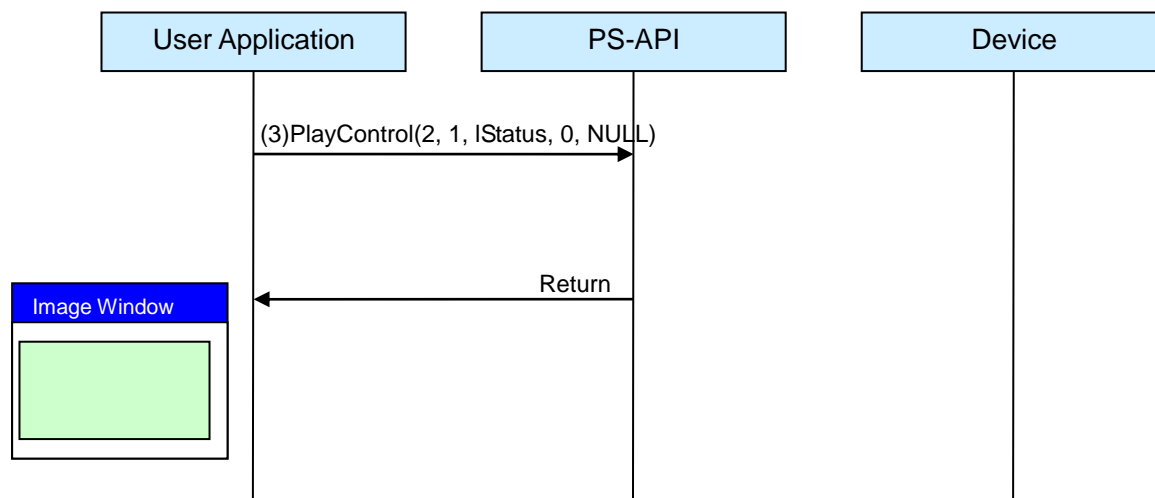
#### Start Fast Forward (or Rewind)

Refer to 6.2 Play, Start the Fast Forward (or Rewind)

#### Display the next frame (or the previous frame)

Refer to 6.2 Play, Display the next frame (or the previous frame)

### Stop local file playback



\* The timing of sending CGI command is not constant,  
depends on the device model or the device setting.

**Figure 6-12 Stop Local file Playback**

## 6.4. CameraControl

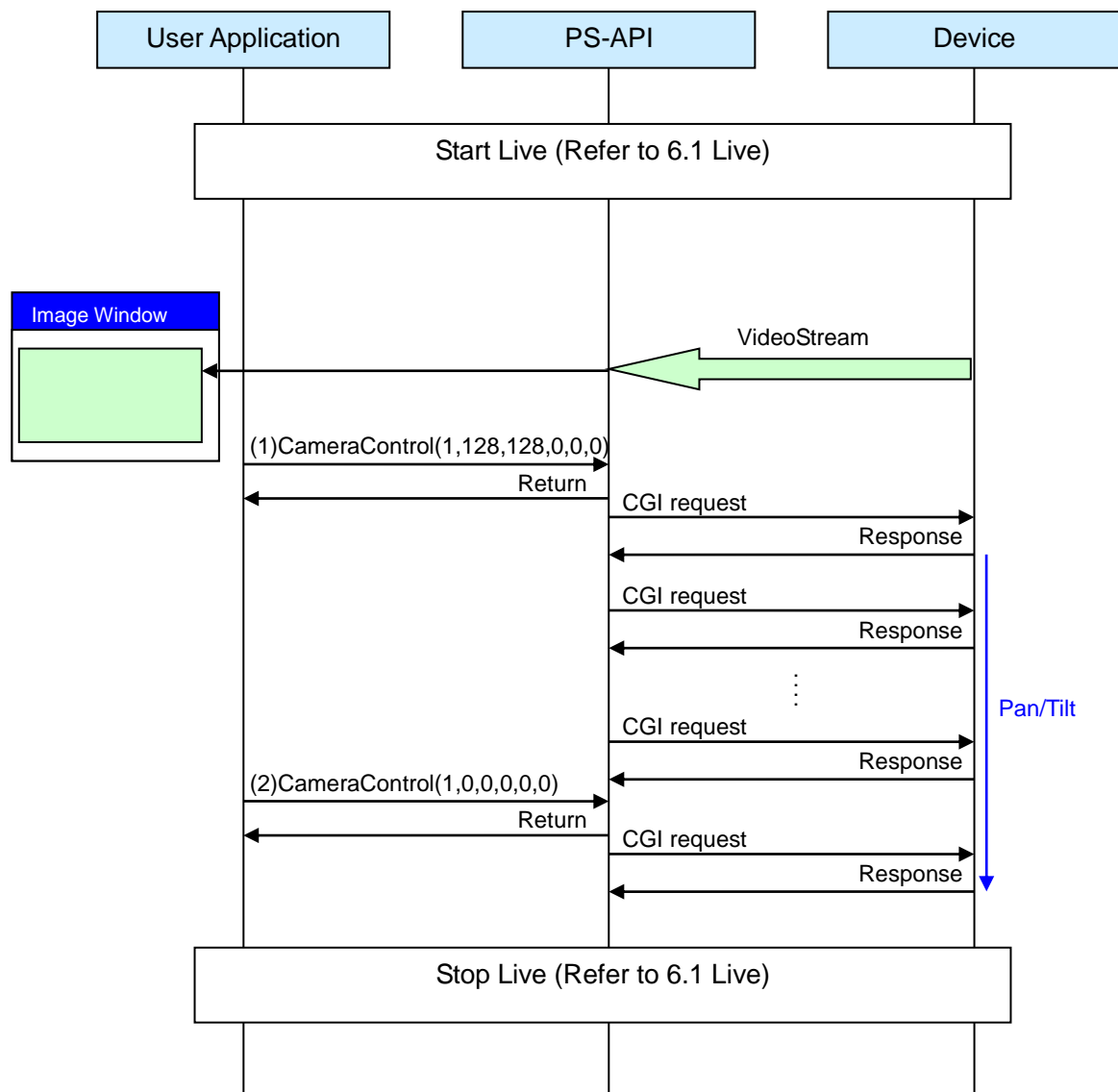
### 6.4.1. Operation Procedure

#### Camera Control

No.	Property / Method	Parameter	Description
-	<b>Start Live (Refer to 6.1 PlayLive)</b>		
1	CameraControl	Channel, Pan speed, Tilt speed, Zoom speed, Focus speed, Iris open/close (long, long, long, long, long, long)	Start the camera control. In controlling, PS-API communicate with a target device. CameraControl method can be used in async. If CameraControl method is called during controlling the camera, the latest CameraControl command is given first priority. When getting the error of CaneraControl method, please set the listener class by using SetErrListener.  e.g.) 1, 128, 128, 0, 0, 0
2	CameraControl	Channel, 0, 0, 0, 0, 0 (long, long, long, long, long, long)	Stop the camera control by setting to "0" the parameter that is needed to stop.  e.g.) 1, 0, 0, 0, 0, 0
-	<b>Stop Live (Refer to 6.1 PlayLive)</b>		

## 6.4.2. Sequence

### Camera Control



\* The timing of sending CGI command is not constant, depends on the device model or the device setting.

**Figure 6-13 Camera Control**

## 6.5. CameraOperation

### 6.5.1. Operation Procedure

#### Start Live

No.	Property / Method	Parameter	Description
-	<b>Start Live (Refer to 6.1 PlayLive)</b>		

#### Auto Track

No.	Property / Method	Parameter	Description
1	CameraOperation	Channel, Command, Data, Status, Blocking mode, Callback interface (long, long, long, long&, long, IAppCallBack*)	Start auto track. Set command to "1". Set data to "0". Set the status to a long type variable . When using blocking mode, it is necessary that callback interface is set to "NULL". When using non-blocking mode, please set the callback interface to an instance to receive the result of async processing. (The callback interface can be set to "NULL", but application cannot receive the notification of finish processing.) e.g.) 1, 1, 0, IStatus, 0, NULL
2	CameraOperation	Channel, Command, Data, Status, Blocking mode, Callback interface (long, long, long, long&, long, IAppCallBack*)	Stop auto track. Set command to "0". Set data to "0". Set the status to a long type variable . When using blocking mode, it is necessary that callback interface is set to "NULL". When using non-blocking mode, please set the callback interface to an instance to receive the result of async processing. (The callback interface can be set to "NULL", but application cannot receive the notification of finish processing.) e.g.) 1, 0, 0, IStatus, 0, NULL

**Auto Pan**

No.	Property / Method	Parameter	Description
3	CameraOperation	Channel, Command, Data, Status, Blocking mode, Callback interface (long, long, long, long&, long, IAppCallBack*)	Start auto pan. Set command to "2". Set data to "0". Set the status to a long type variable . When using blocking mode, it is necessary that callback interface is set to "NULL". When using non-blocking mode, please set the callback interface to an instance to receive the result of async processing. (The callback interface can be set to "NULL", but application cannot receive the notification of finish processing.) e.g.) 1, 2, 0, IStatus, 0, NULL
4	CameraOperation	Channel, Command, Data, Status, Blocking mode, Callback interface (long, long, long, long&, long, IAppCallBack*)	Stop auto pan. Set command to "0". Set data to "0". Set the status to a long type variable . When using blocking mode, it is necessary that callback interface is set to "NULL". When using non-blocking mode, please set the callback interface to an instance to receive the result of async processing. (The callback interface can be set to "NULL", but application cannot receive the notification of finish processing.) e.g.) 1, 0, 0, IStatus, 0, NULL

**Auto Focus**

No.	Property / Method	Parameter	Description
5	CameraOperation	Channel, Command, Data, Status, Blocking mode, Callback interface (long, long, long, long&, long, IAppCallBack*)	Start auto focus. Set command to "3". Set data to "0". Set the status to a long type variable . When using blocking mode, it is necessary that callback interface is set to "NULL". When using non-blocking mode, please set the callback interface to an instance to receive the result of async processing. (The callback interface can be set to "NULL", but application cannot receive the notification of finish processing.) e.g.) 1, 3, 0, IStatus, 0, NULL

**Absolute position camera control**

No.	Property / Method	Parameter	Description
6	GetCameraPosition	Channel, Pan value, Tilt value, Zoom value, Focus value (long, long& long& long& long&)	Get the absolute position value of Pan/ Tilt/ Zoom/ Focus. Specify the long type variable for Pan position, Tilt position, Zoom position, and Focus position.  e.g.1) 1, IPan, ITilt, IZoom, IFocus
7	SetCameraPosition	Channel, Pan value, Tilt value, Zoom value, Focus value, (long, long, long, long, long)	Specify the absolute value of Pan/ Tilt/ Zoom/ Focus. Specify the absolute value with Pan position(-475 - 3599), Tilt position (-450 - +900), Zoom position (10 - 9999), Focus position (1 - 9999).  e.g.1) 1, 0, 0, 10, 14 e.g.2) 1, 360, 360, 30, 300

**Stop Live**

No.	Property / Method	Parameter	Description
-	<b>Stop Live (Refer to 6.2 PlayLive)</b>		

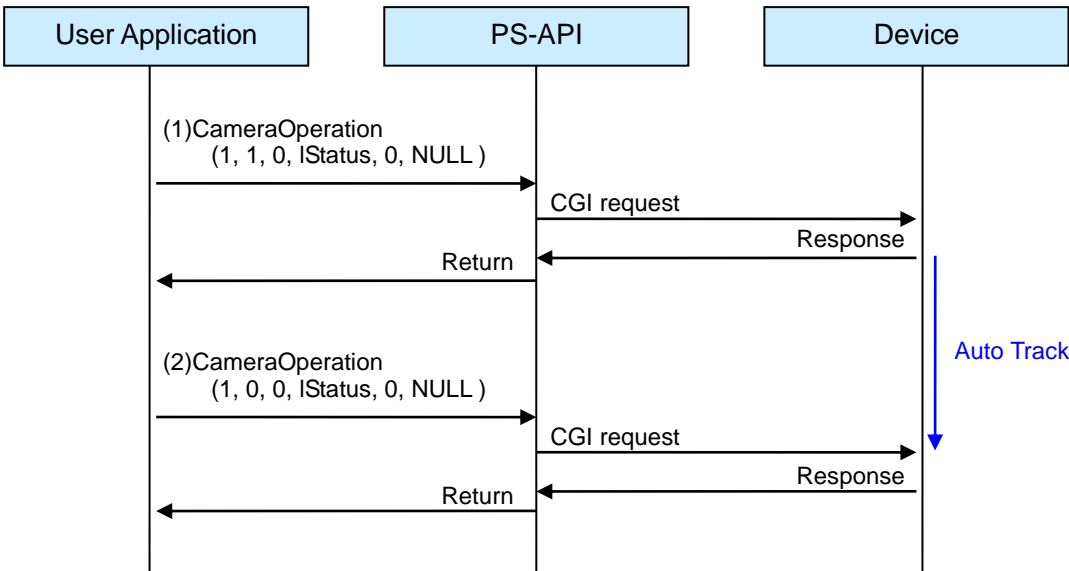


6.5.2. Sequence

Start Live

Refer to 6.1 PlayLive, Start Live

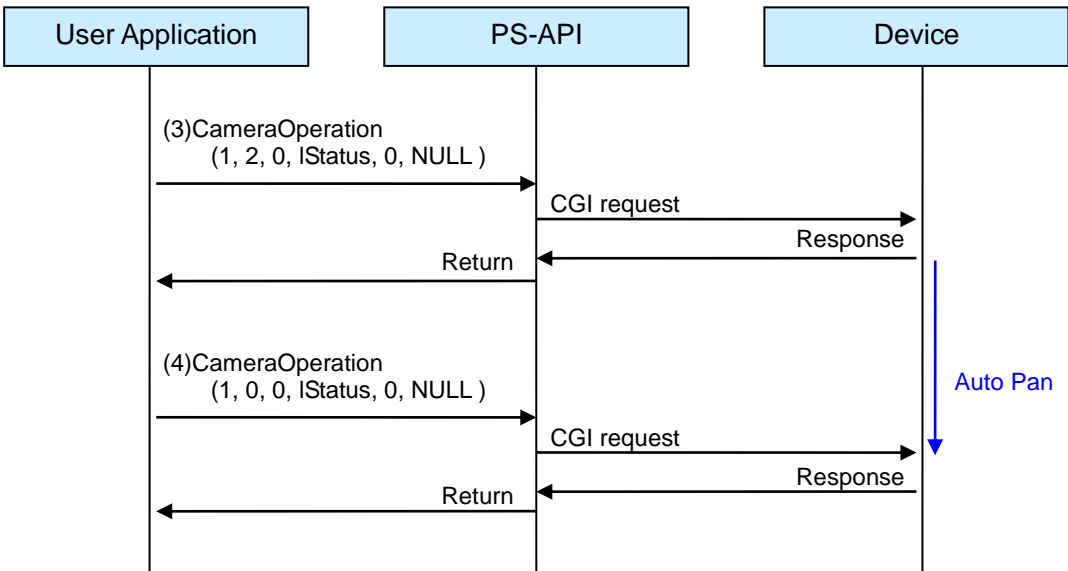
Auto track



\* The timing of sending CGI command is not constant, depends on the device model or the device setting.

Figure 6-14 Auto Track

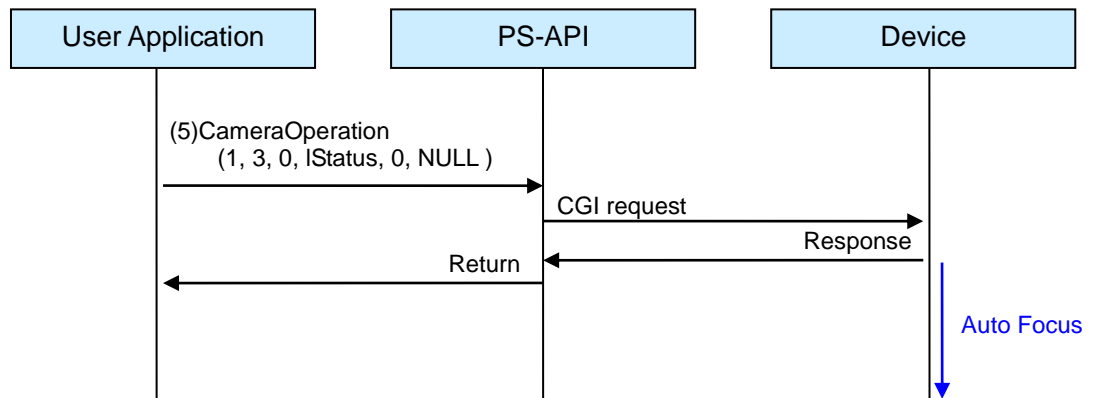
Auto Pan



\* The timing of sending CGI command is not constant, depends on the device model or the device setting.

Figure 6-15 Auto Pan

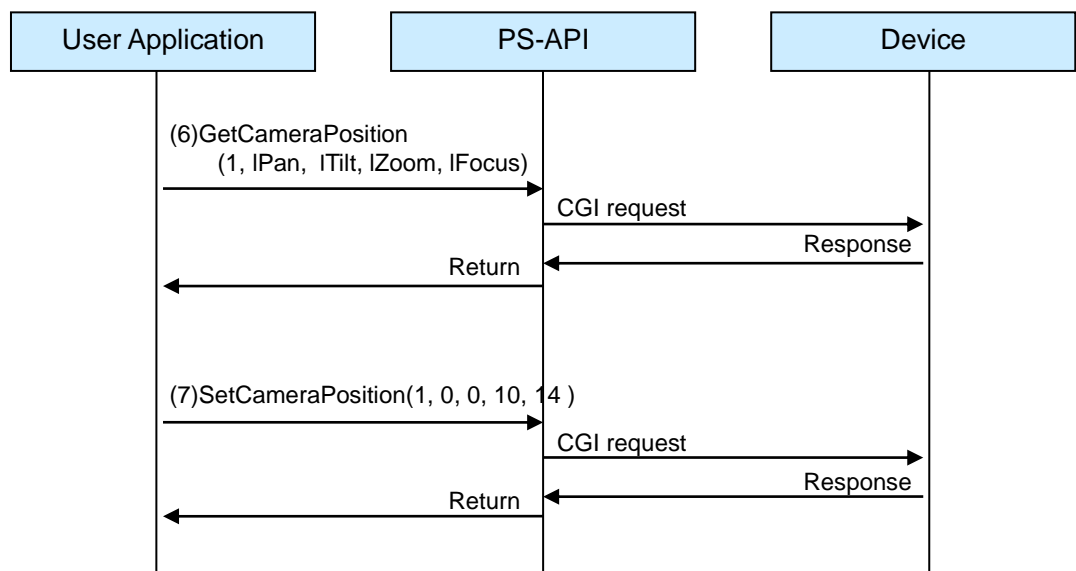
## Auto Focus



\* The timing of sending CGI command is not constant, depends on the device model or the device setting.

**Figure 6-16 Auto Focus**

## Absolute position camera control



\* The timing of sending CGI command is not constant, depends on the device model or the device setting.

**Figure 6-17 Absolute Position Camera Control**

## Stop Live

Refer to 6.2 PlayLive, Stop live

## 6.6. AlmOperation

### 6.6.1. Operation Procedure

#### Create the IPSAPI Instance and Login

No.	Property / Method	Parameter	Description
-	<b>Create the IPSAPI Instance and Login (Refer to 6.2 Play)</b>		

#### Reset alarm

No.	Property / Method	Parameter	Description
1	AlmOperation	Channel, Command, Status, Blocking mode, Callback interface (long, long, long&, long IAppCallBack*)	<p>Reset the alarms in the target device.</p> <p>Alarm reset cannot be used per channel. Even if channel is specified, all alarm that is occurred in device will be reset.</p> <p>In case of alarm reset, set the command to "1".</p> <p>Set the status to a long type variable .</p> <p>When using blocking mode, it is necessary that callback interface is set to "NULL".</p> <p>When using non-blocking mode, please set the callback interface to an instance to receive the result of async processing.</p> <p>(The callback interface can be set to "NULL", but application cannot receive the notification of finish processing.)</p> <p>e.g.) 1, 1, IStatus, 0, NULL</p>

#### Trigger ON

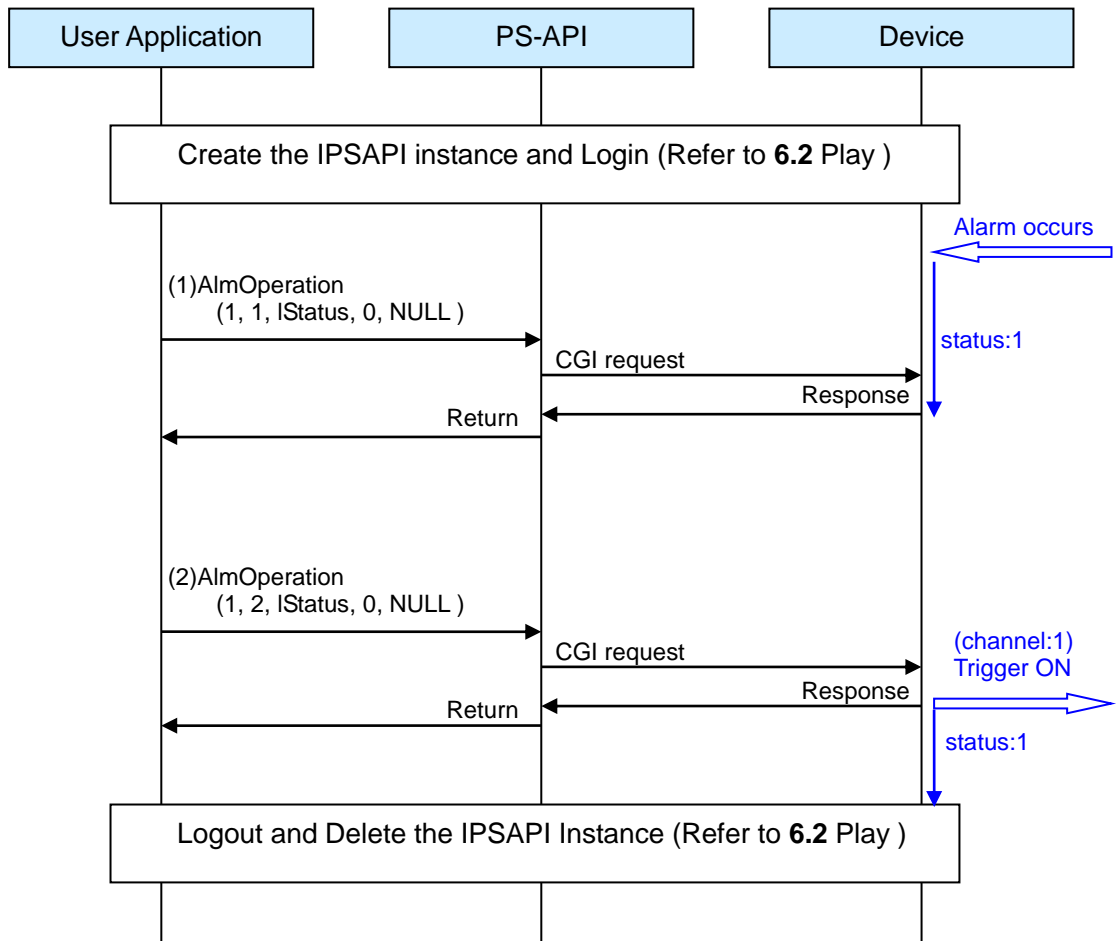
No.	Property / Method	Parameter	Description
2	AlmOperation	Channel, Command, Status, Blocking mode, Callback interface (long, long, long&, long IAppCallBack*)	<p>Trigger a device alarm by the application.</p> <p>Trigger operation is valid to NWDR and NX Series.</p> <p>In case of trigger ON, set the command to "2".</p> <p>Set the status to a long type variable .</p> <p>When using blocking mode, it is necessary that callback interface is set to "NULL".</p> <p>When using non-blocking mode, please set the callback interface to an instance to receive the result of async processing.</p> <p>(The callback interface can be set to "NULL", but application cannot receive the notification of finish processing.)</p> <p>e.g.) 1, 2, IStatus, 0, NULL</p>

#### Logout and Delete the IPSAPI Instance

No.	Property / Method	Parameter	Description
-	<b>Logout and Delete the IPSAPI Instance (Refer to 6.2 Play)</b>		

## 6.6.2. Sequence

### Reset alarm / Trigger ON



\* The timing of sending CGI command is not constant, depends on the device model or the device setting.

**Figure 6-18 Reset Alarm / Trigger ON**

## 6.7. Search

### 6.7.1. Operation Procedure

#### Create the IPSAPI Instance and Login

No.	Property / Method	Parameter	Description
-	<b>Create the IPSAPI Instance and Login (Refer to 6.2 Play)</b>		
1	GetISearchResultEx	-	Create the ISearchResultEx instance for getting search result list.

#### Search

No.	Property / Method	Parameter	Description
2	SearchEx	Channel, Strat date, End date, Event kind, Search result, Blocking mode, Callback interface (long, char*, char*, long, ISearchResult*, long IAppCallBack*)	Start searching. Specify channel, start date, end date and event kind as search condition. Specified the ISearchResultEx insrtance pointer, that is got by using GetISearchResultEx method, as argument to get search result. Set the status to a long type variable . When using blocking mode, it is necessary that callback interface is set to "NULL". When using non-blocking mode, please set the callback interface to an instance to receive the result of async processing. (The callback interface can be set to "NULL", but application cannot receive the notification of finish processing.)  e.g.) 1, "2008/12/01 00:00:00", "2008/12/31 23:59:59", 63, m_SearchResult, 0, NULL

#### Get search result (1)

No.	Property / Method	Parameter	Description
3	GetListCount	Total result count (long&)	The total of the search result.
4	GetAndDelete	Search result (SSEARCHRSLT_INFO*)	Get the search result. By using GetAndDelete method, the latest recording data information can be got, this recording data information is deleted from ISearchResultEx. Please get all search results by calling GetAndDelete repeatedly.

**Get search result (2)**

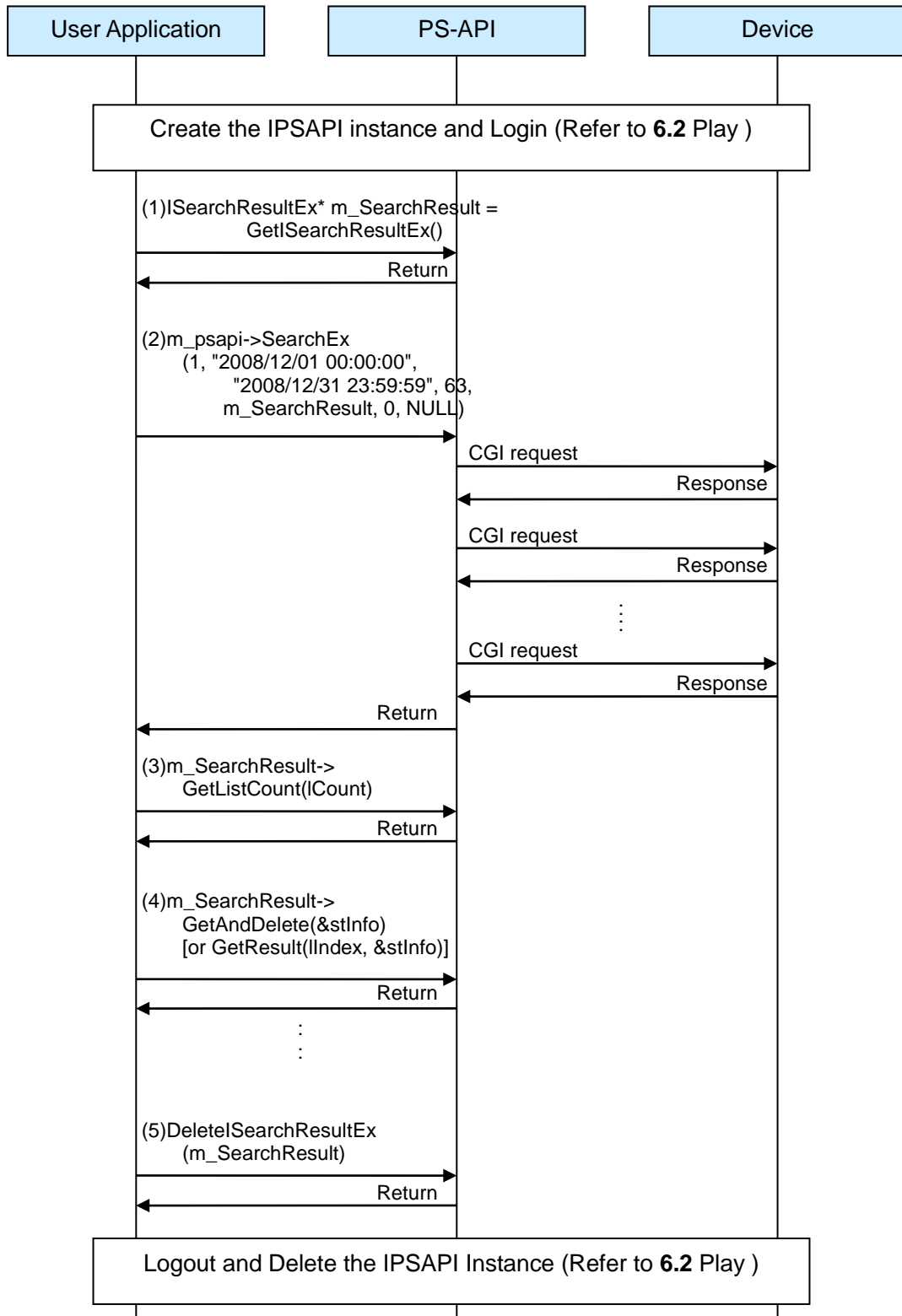
No.	Property / Method	Parameter	Description
3	GetListCount	Total result count (long&)	The total of the search result.
4	GetResult	Index, Search result (long SSEARCHRSLT_INFO*)	Get the search result. Specify the index of search result. The index of the latest recording data information is "1". Even if getting the search result by using GetResult, the recording data information isn't deleted from ISearchResultEx instance.  e.g.1)1, &stInfo

**Logout and Delete the IPSAPI Instance**

No.	Property / Method	Parameter	Description
5	DeleteISearchResultEx	Pointer of ISearchResultEx (ISearchResultEx*)	Delete the ISearchResultEx instance.
-	<b>Logout and Delete the IPSAPI Instance (Refer to 6.2 Play)</b>		

## 6.7.2. Sequence

### Search



\* The timing of sending CGI command is not constant, depends on the device model or the device setting.

**Figure 6-19 Search**

## 6.8. RecCtrl

### 6.8.1. Operation Procedure

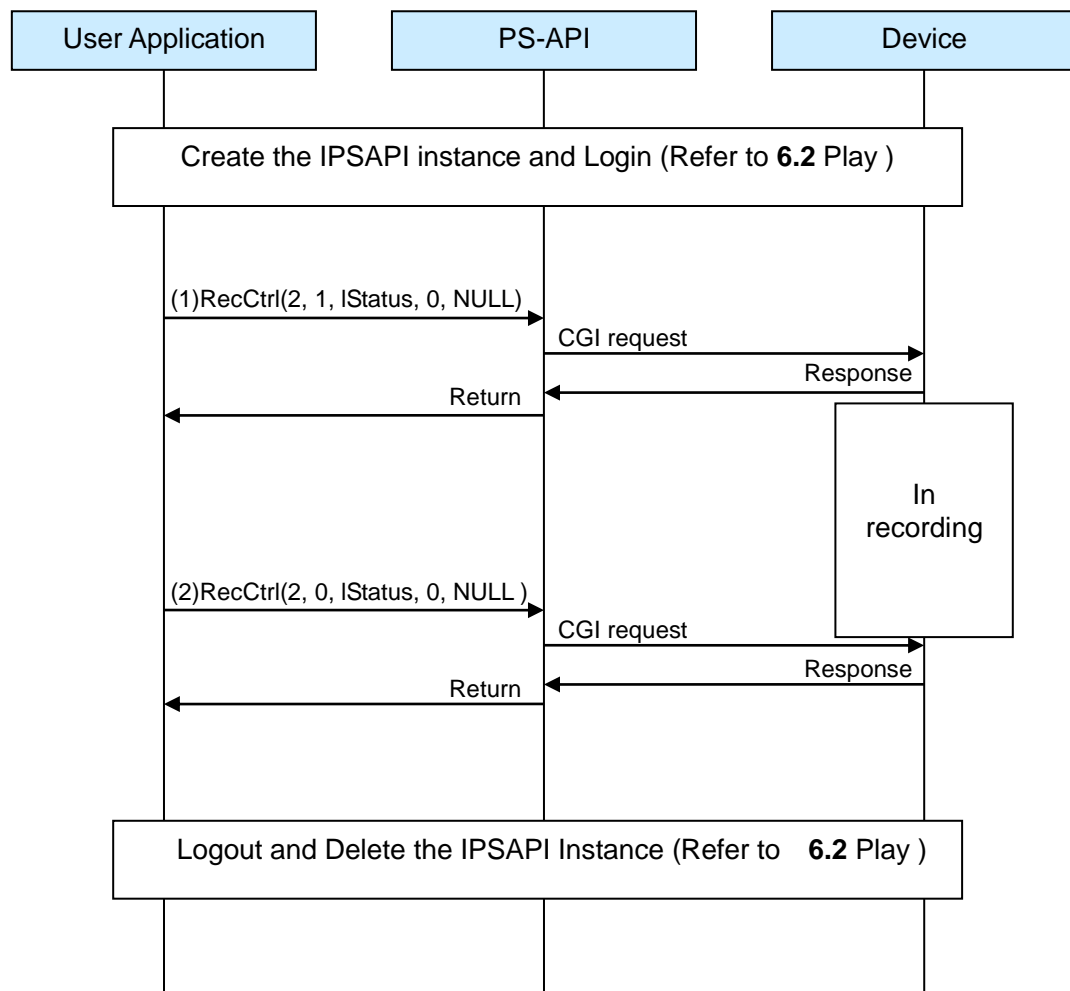
#### Manual Recording

No.	Property / Method	Parameter	Description
-	<b>Create the IPSAPI Instance and Login (Refer to 6.2 Play)</b>		
1	RecCtrl	Channel, Command, Status, Blocking mode, Callback interface (long, long long&, long, IAppCallBack*)	<p>Start manual recording.</p> <p>When starting manual recording for all channel, set the channel to "0". If specifying a channel for HD300, ND200, ND300 and HD600/700, all channels recording will be started.</p> <p>In case of recording start, set the command to "1".</p> <p>Set the status to a long type variable .</p> <p>When using blocking mode, it is necessary that callback interface is set to "NULL".</p> <p>When using non-blocking mode, please set the callback interface to an instance to receive the result of async processing.</p> <p>(The callback interface can be set to "NULL", but application cannot receive the notification of finish processing.)</p> <p>e.g.) 2, 1, IStatus, 0, NULL</p>
2	RecCtrl	Channel, Command, Status, Blocking mode, Callback interface (long, long long&, long, IAppCallBack*)	<p>Stop manual recording.</p> <p>In case of recording stop, set the command to "0".</p> <p>e.g.) 2, 0, IStatus, 0, NULL</p>
-	<b>Logout and Delete the IPSAPI Instance (Refer to 6.2 Play)</b>		



## 6.8.2. Sequence

### Manual Recording



\* The timing of sending CGI command is not constant, depends on the device model or the device setting.

**Figure 6-20 Manual Recording**

## 6.9. MultiPlayLive

### 6.9.1. Operation Procedure

#### Start Live

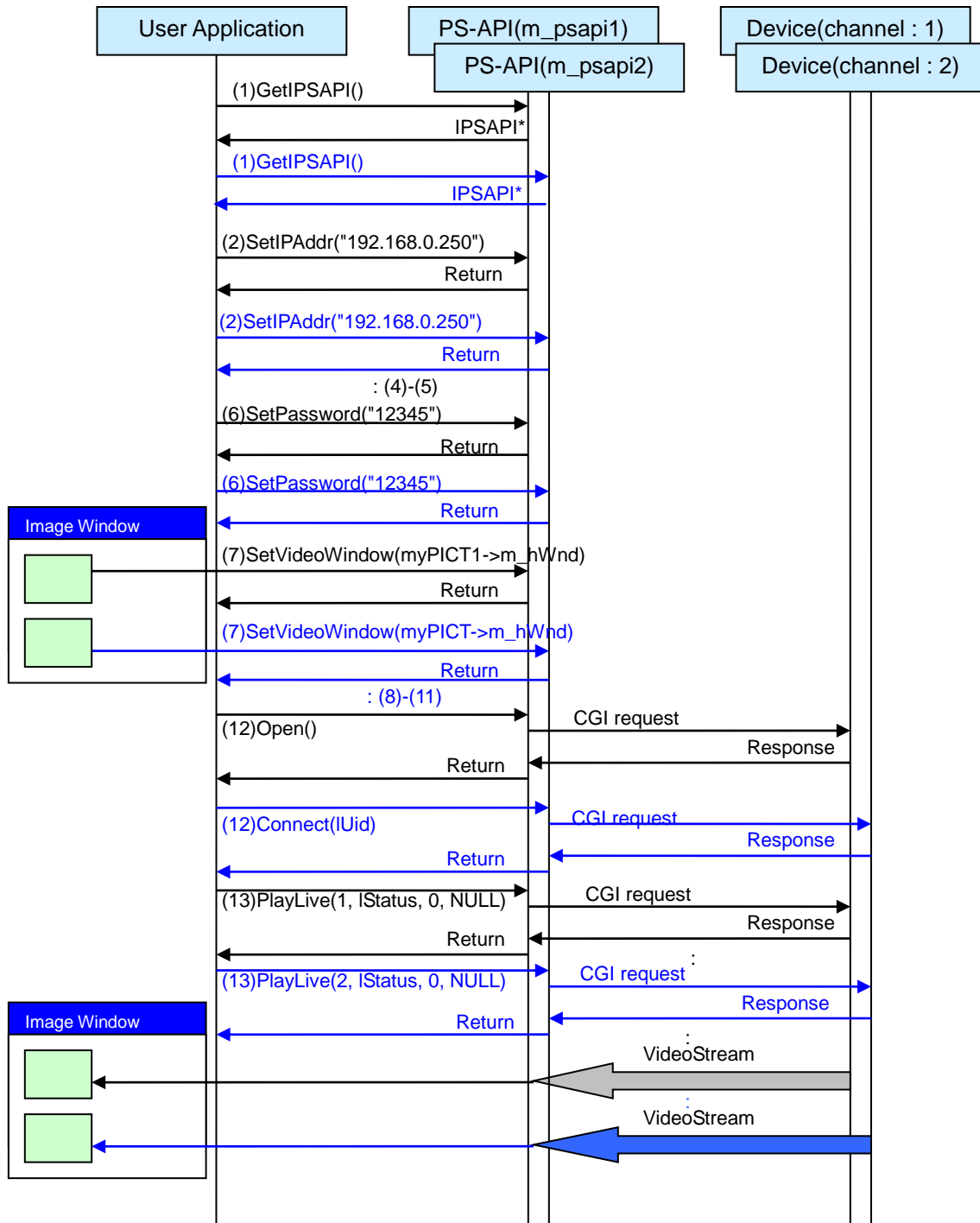
No.	Property / Method	Parameter	Description	Sample Code
1	GetIPSAPI	Refer to 6.1 PlayLive	Refer to 6.1 PlayLive	IPSAPI* m_psapi1 = GetIPSAPI( ); IPSAPI* m_psapi2 = GetIPSAPI( );
2	SetIPAddr	Refer to 6.1 PlayLive	Refer to 6.1 PlayLive	m_psapi1->SetIPAddr("192.168.0.250"); m_psapi2->SetIPAddr("192.168.0.250");
3	SetDeviceType	Refer to 6.1 PlayLive	Refer to 6.1 PlayLive	m_psapi1->SetDeviceType(1); m_psapi2->SetDeviceType(1);
4	SetHttpPort	Refer to 6.1 PlayLive	Refer to 6.1 PlayLive	m_psapi1->SetHttpPort(80); m_psapi2->SetHttpPort(80);
5	SetUserName	Refer to 6.1 PlayLive	Refer to 6.1 PlayLive	m_psapi1->SetUserName("ADMIN"); m_psapi2->SetUserName("ADMIN");
6	SetPassword	Refer to 6.1 PlayLive	Refer to 6.1 PlayLive	m_psapi1->SetPassword("12345"); m_psapi2->SetPassword("12345");
7	SetVideoWindow	Refer to 6.1 PlayLive	Refer to 6.1 PlayLive	CWnd* myPICT1 = GetDlgItem(IDC_STATIC1); m_psapi1->SetVideoWindow(myPICT1->m_hWnd);  CWnd* myPICT2 = GetDlgItem(IDC_STATIC2); m_psapi2->SetVideoWindow(myPICT2->m_hWnd);
8	SetImageWidth	Refer to 6.1 PlayLive	Refer to 6.1 PlayLive	m_psapi1->SetImageWidth(640); m_psapi2->SetImageWidth(640);
9	SetImageHeight	Refer to 6.1 PlayLive	Refer to 6.1 PlayLive	m_psapi1->SetImageHeight(480); m_psapi2->SetImageHeight(480);
10	SetStreamFormat	Refer to 6.1 PlayLive	Refer to 6.1 PlayLive	m_psapi1->SetStreamFormat(0); m_psapi2->SetStreamFormat(0);
11	SetJPEGResolution/ SetMPEG4Resolution/ SetH264Resolution	Refer to 6.1 PlayLive	Refer to 6.1 PlayLive	m_psapi1->SetJPEGResolution(640); m_psapi2->SetJPEGResolution(640);
12	Open/Connect	Refer to 6.1 PlayLive	Refer to 6.1 PlayLive	long IUID = m_psapi1->Open(); m_psapi2->Connect(IUID);
13	PlayLive	Refer to 6.1 PlayLive	Refer to 6.1 PlayLive	//channel : 1 m_psapi1->PlayLive(1, IStatus1, 0, NULL); //channel : 2 m_psapi2->PlayLive(2, IStatus, 0, NULL);

#### Stop Live

No.	Property / Method	Parameter	Description	Sample Code
14	PlayControl	Refer to 6.1 PlayLive	Refer to 6.1 PlayLive	//Stop PlayLive m_psapi1->PlayControl(1, 0, IStatus1, 0, NULL); m_psapi2->PlayControl(1, 0, IStatus1, 0, NULL);
15	Close/Disconnect	Refer to 6.1 PlayLive	Refer to 6.1 PlayLive	//m_psapi2, Disconnect is called first. m_psapi2->Disconnect(); m_psapi1->Close();
16	DeleteIPSAPI	Refer to 6.1 PlayLive	Refer to 6.1 PlayLive	DeleteIPSAPI(m_psapi1); DeleteIPSAPI(m_psapi2);

## 6.9.2. Sequence

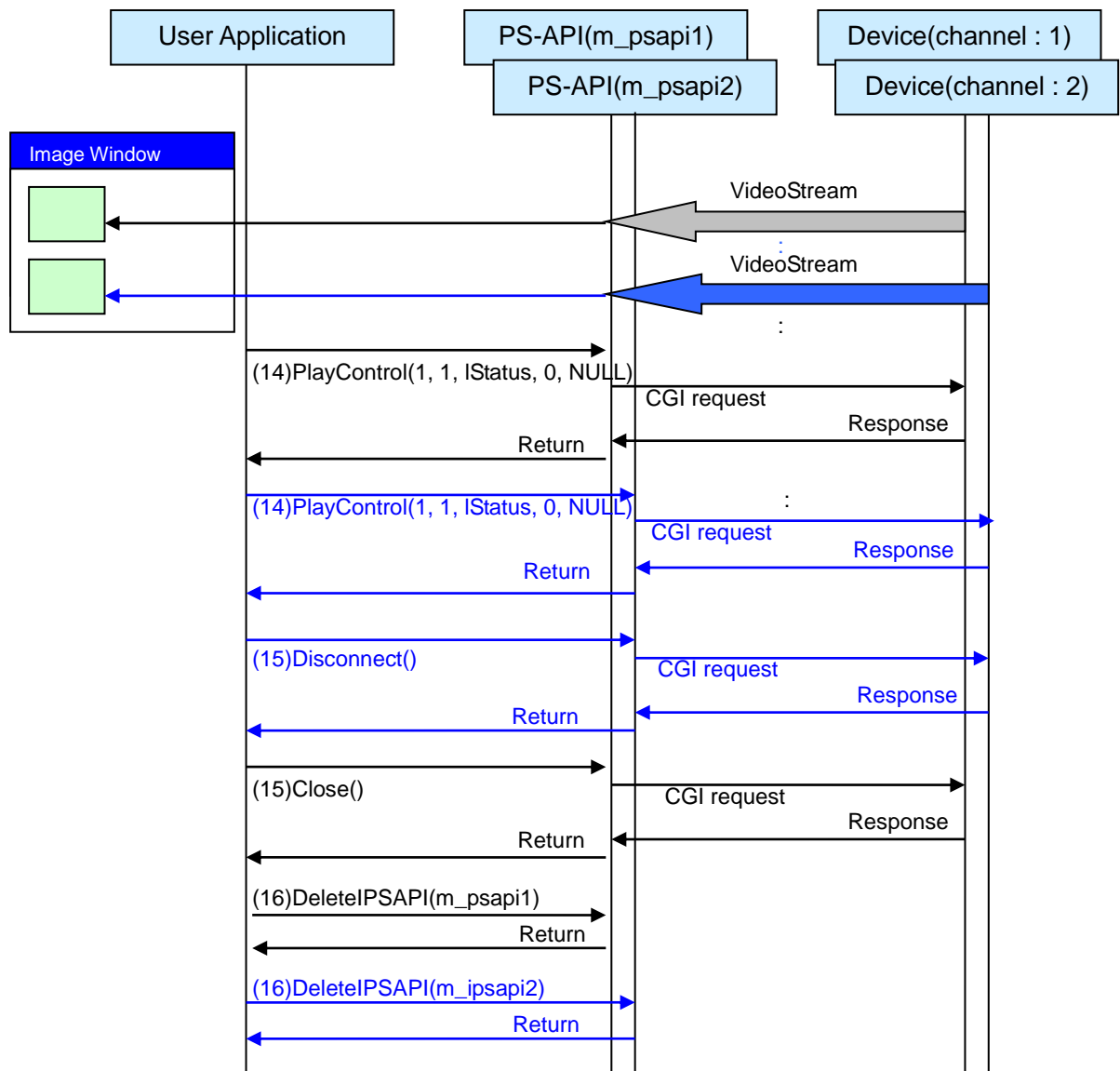
### Start Live



\* The timing of sending CGI command is not constant, depends on the device model or the device setting.

Figure 6-21 Start Live

## Stop Live



\* The timing of sending CGI command is not constant, depends on the device model or the device setting.

**Figure 6-22 Stop Live**

## 6.10. DecodeImage

### 6.10.1. Operation Procedure

#### Get Compressed video image

No.	Property / Method	Parameter	Description
-	<b>Create the IPSAPI Instance and Login (Refer to 6.1 PlayLive)</b>		
1	SetImageListener	Listener, Image type (IAppLisener*, long)	Set the instance of listener class that implement "OnImage" . PS-API can notify an image data by calling "OnImage", when the image update happens. Specify a video image type.  e.g.) pReceiver, 2
2	PlayLive	Channel, Status, Blocking mode, Callback interface (long, long&, long, IAppCallBack*)	Start displaying live image. In case of network camera, please set the channel to "1". In case of encoder, please set the channel to number that is wanted to display. Set the status to a long type variable . When using blocking mode, it is necessary that callback interface is set to "NULL". When using non-blocking mode, please set the callback interface to an instance to receive the result of async processing. (The callback interface can be set to "NULL", but application cannot receive the notification of finish processing.)  e.g.) 1, IStatus, 0, NULL
3	OnImage (Listener method)	Image type, Pointer, Size (long, unsigned char*, long)	Notify the updated image to the specified application with OnImage event. About Compressed video image format, please refer to "5.5.3.5 OnImage

4	PlayControl	Command, Speed, Status, Blocking mode, Callback interface (long, long, long&, long, IAppCallBack*)	To stop live, set command to "1". Set speed to "1". Set the status to a long type variable . When using blocking mode, it is necessary that callback interface is set to "NULL". When using non-blocking mode, please set the callback interface to an instance to receive the result of async processing. (The callback interface can be set to "NULL", but application cannot receive the notification of finish processing.)  e.g.) 1, 1, IStatus, 0, NULL
5	SetImageListener	Listener, Image type (IAppLisener*, long)	Unregister the Listener class by specifying NULL.  e.g.) NULL, 2

### Draw Compressed video image

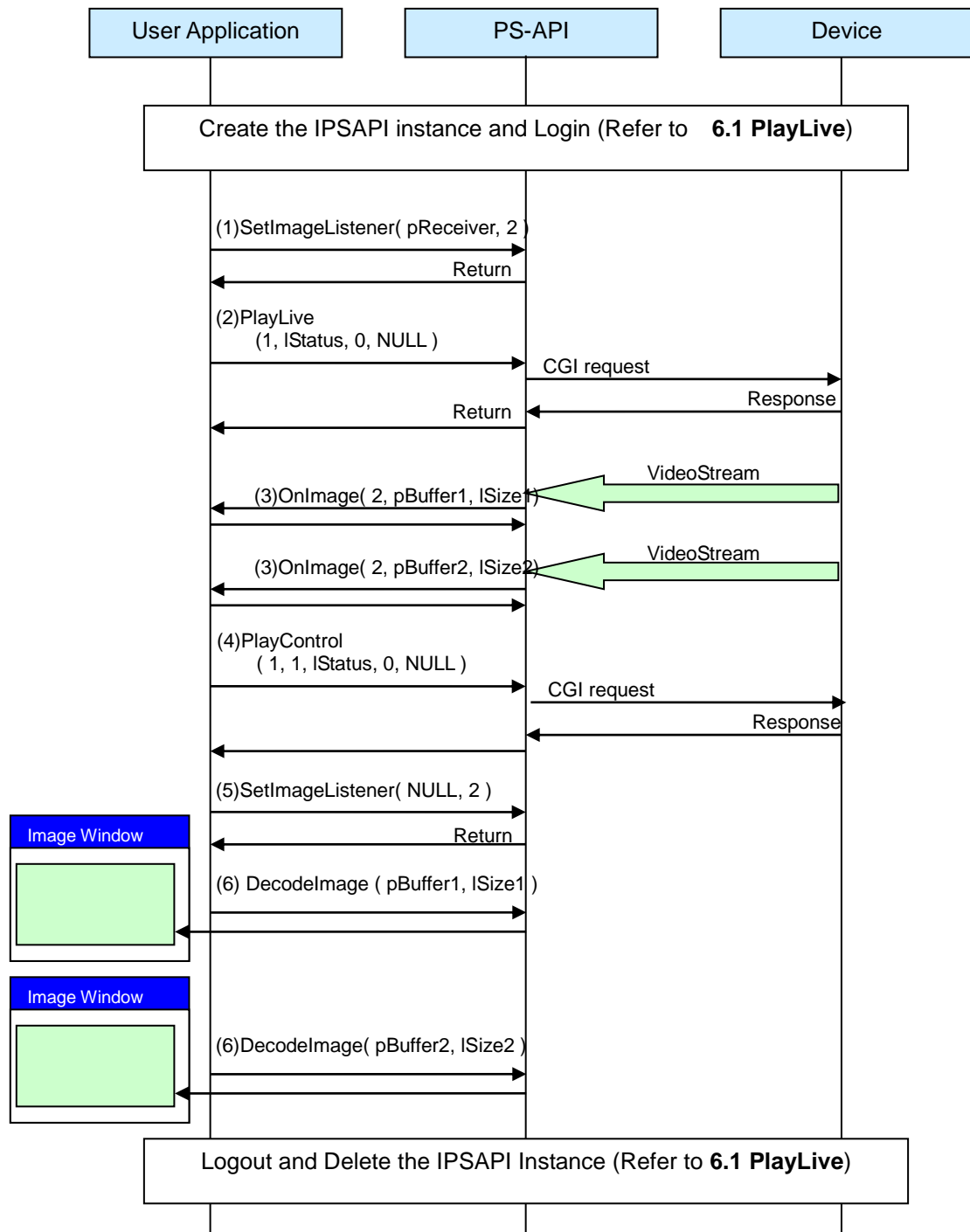
No.	Property / Method	Parameter	Description
-	SetVideoWindow	WindowHandle (hWnd)	Set WinowHandle to display the video image.  e.g.) this->m_hWnd
-	SetStreamFormat	Stream type (long)	Set the stream type into PS-API.  e.g.) 0
6	DecodeImage	Pointer, Size (char* long)	Set the argument to a Compressed video image that is gotten by OnImage. Compressed video image is drawn on the window that is specified by SetVideoWindow. The image format as StreamFormat and the image format of Compressed video image are mismatched, the image cannot be displayed. For MPEG-4/H.264/H.265 drawing, the first Compressed video image must be I frame (I picture). Unless I-frame (I-picture) is specified first, the drawn image is disorderly or isn't shown correctly. When decoding the compressed video image that its Compression (in Compression method) is 0x0004 : H.264(type2), 0x0006 : H.265),cash memory for 2 images is needed. 1st image is decoded when executing DecodeImage for 3rd image.

### Logout and Delete the IPSAPI Instance

No.	Property / Method	Parameter	Description
-	<b>Logout and Delete the IPSAPI Instance (Refer to 6.1 PlayLive)</b>		

## 6.10.2. Sequence

### Get Compressed video image / Draw Compressed video image



\* The timing of sending CGI command is not constant, depends on the device model or the device setting.

Figure 6-23 Get Compressed vide image / Draw Compressed vide image

## 6.11. FtpGet

### 6.11.1. Operation Procedure

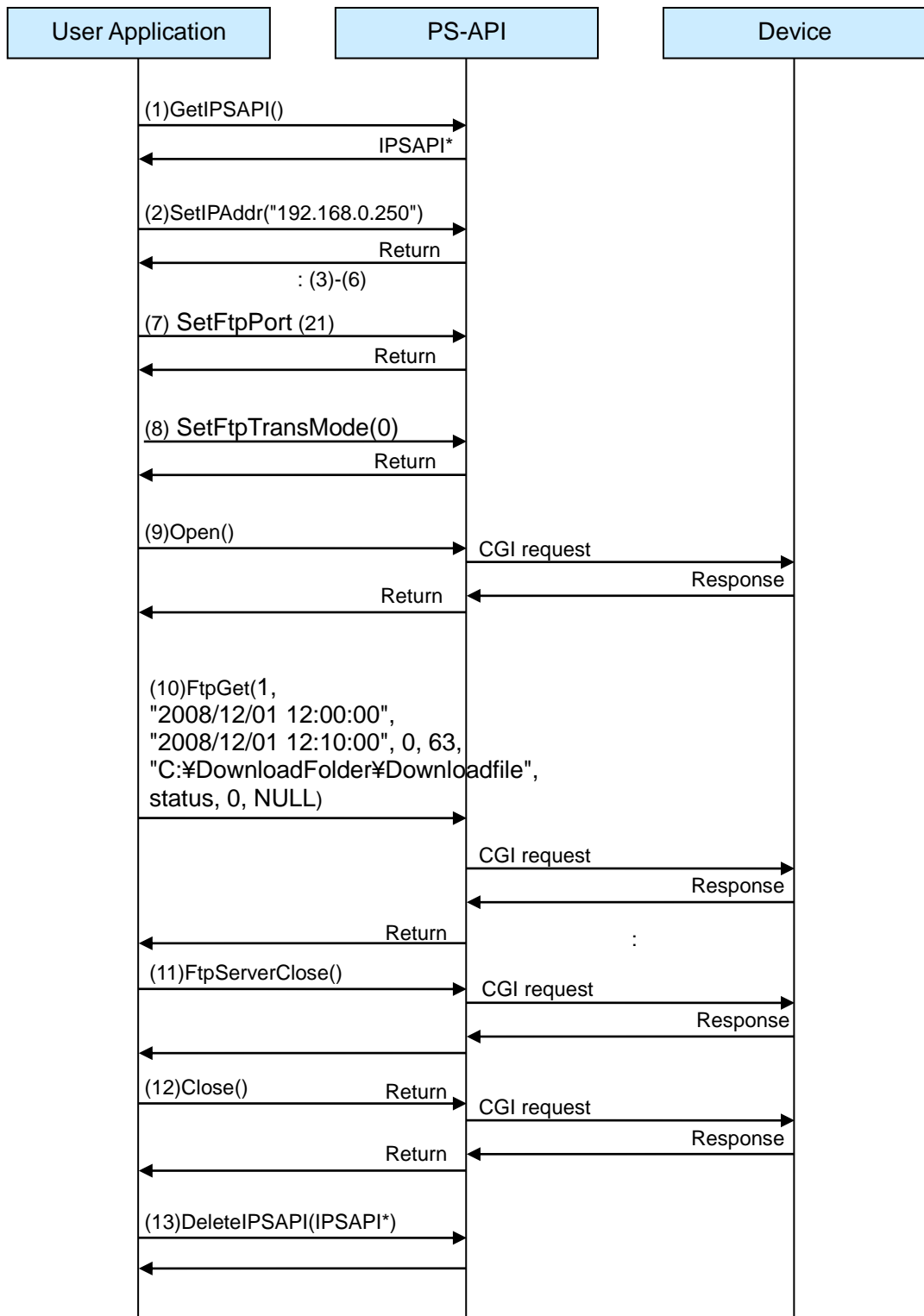
#### FTP download

No.	Property / Method	Parameter	Description
1	GetIPSAPI	-	Create instance of IPSAPI. 1 instance can control 1 target device.
2	SetIPAddr	IP Address (char*)	Set IP Address into PS-API.  e.g.) "192.168.0.10"
3	SetDeviceType	Device type (long)	Set the device type corresponding to a target device.  e.g.) 2
4	SetHttpPort	Port number (long)	Set Http port number into PS-API.  e.g.) 80
5	SetUserName	Character strings (char*)	Set user name into PS-API.  e.g.) admin
6	SetPassword	Character strings (char*)	Set password into PS-API.  e.g.) 12345
7	SetFtpPort	Port number (long)	Ftp server port.  e.g.) 21
8	SetFtpTransMode	Mode (long)	Ftp transmission mode  e.g.) 0
9	Open/Connect	UID (long)	Connect to a target device. When getting a new UID, please use Open method.



No.	Property / Method	Parameter	Description
10	FtpGet	Channel, Start time&date, End time&date, Data type Event type, Status File Name Blocking mode, Callback interface (long, char*, char*, long, long, char*, long&, long, IAppCallBack*)	Start FTP downloading. Specify channel, start date, end date, data type, event type and file name as download condition. Set the status to a long type variable . When using blocking mode, it is necessary that callback interface is set to "NULL". When using non-blocking mode, please set the callback interface to an instance to receive the result of async processing. (The callback interface can be set to "NULL", but application cannot receive the notification of finish processing.)  e.g.) 1, "2008/12/01 12:00:00", "2008/12/01 12:10:00", 0, 63, "C:¥DownloadFolder¥Downloadfile", status, 0, NULL
11	FtpServerClose	-	Turn off the FTP server mode of a target device. If the PS-API instance doesn't log in yet, the PS-API instance logs in during changing the FTP server mode.
12	Close/Disconnect	-	Stop the communication with the target device. When using Close method, UID will be annulled.
13	DeleteIPSAPI	Pointer of IPSAPI (IPSAPI*)	Delete the IPSAPI instance.

### 6.11.2. Sequence



\* The timing of sending CGI command is not constant, depends on the device model or the device setting.

**Figure 6-24 FTP download**

## 6.12. Audio

### 6.12.1. Operation Procedure

#### Start Live with audio reception

No.	Property / Method	Parameter	Description
1	GetIPSAPI	-	Create instance of IPSAPI. 1 instance can control 1 target device.
2	SetIPAddr	IP Address (char*)	Set IP Address into PS-API.  e.g.) "192.168.0.10"
3	SetDeviceType	Device type (long)	Set the device type corresponding to a target device.  e.g.) 2
4	SetHttpPort	Port number (long)	Set Http port number into PS-API.  e.g.) 80
5	SetUserName	Character strings (char*)	Set user name into PS-API.  e.g.) admin
6	SetPassword	Character strings (char*)	Set password into PS-API.  e.g.) 12345
7	SetVideoWindow	WindowHandle (hWnd)	Set WinowHandle to display the video image.  e.g.) this->m_hWnd
8	SetImageWidth	Display image width (long)	Set the image width to display.  e.g.) 640
9	SetImageHeight	Display image height (long)	Set the image height to display.  e.g.) 480

No.	Property / Method	Parameter	Description
10	SetStreamFormat	Stream type (long)	Set the stream type into PS-API. In case of NWDR, set the stream type that is same with a target channel that is wanted to display.  e.g.) 0
11	SetJPEGResolution/ SetMPEG4Resolution/ SetH264Resolution	Resolution (long)	It is necessary that the resolution setting is same value of the device setting.  e.g.) 640
12	Open/Connect	UID (long)	Connect to a target device. When getting a new UID, please use Open method.
13	SetAudioRcvEnable	Reception mode (long)	Set a mode whether receiving audio stream with PlayLive or Play into PS-API.  e.g.) 1
14	SetAudioRcvVolume	Volume (long)	Set a volume of the audio reception into PS-API.  e.g.) 10

No.	Property / Method	Parameter	Description
15	PlayLive	Channel, Status, Blocking mode, Callback interface (long, long& long, IAppCallBack*)	<p>Start displaying live image. In case of network camera, please set the channel to "1". In case of NWDR, NX Series, HD600/700 or HD300, please set the channel to number that is wanted to display.</p> <p>Set the status to a long type variable .</p> <p>When using blocking mode, it is necessary that callback interface is set to "NULL".</p> <p>When using non-blocking mode, please set the callback interface to an instance to receive the result of async processing.</p> <p>(The callback interface can be set to "NULL", but application cannot receive the notification of finish processing.)</p> <p>e.g.) 1, IStatus, 0, NULL</p>

#### Change volume for audio reception

No.	Property / Method	Parameter	Description
16	SetAudioRcvVolume	Volume (long)	<p>Change a volume of the audio reception into PS-API.</p> <p>e.g.) 50</p>

#### Mute setting for audio reception

No.	Property / Method	Parameter	Description
17	SetAudioRcvMute	Mute (long)	<p>Set a mute mode of the audio reception into PS-API.</p> <p>e.g.) 1</p>

**Start audio transmission**

No.	Property / Method	Parameter	Description
18	SetAudioSendVolume	Volume (long)	Set a volume of the audio transmission into PS-API. e.g.) 10
19	SetAudioSendMute	Mute (long)	Set a mute mode setting of the audio transmission into PS-API. e.g.) 0
20	AudioSend	command (long)	Start or Stop the audio transmission. When starting audio transmission, command is set to "1". e.g. 1

**Stop audio transmission**

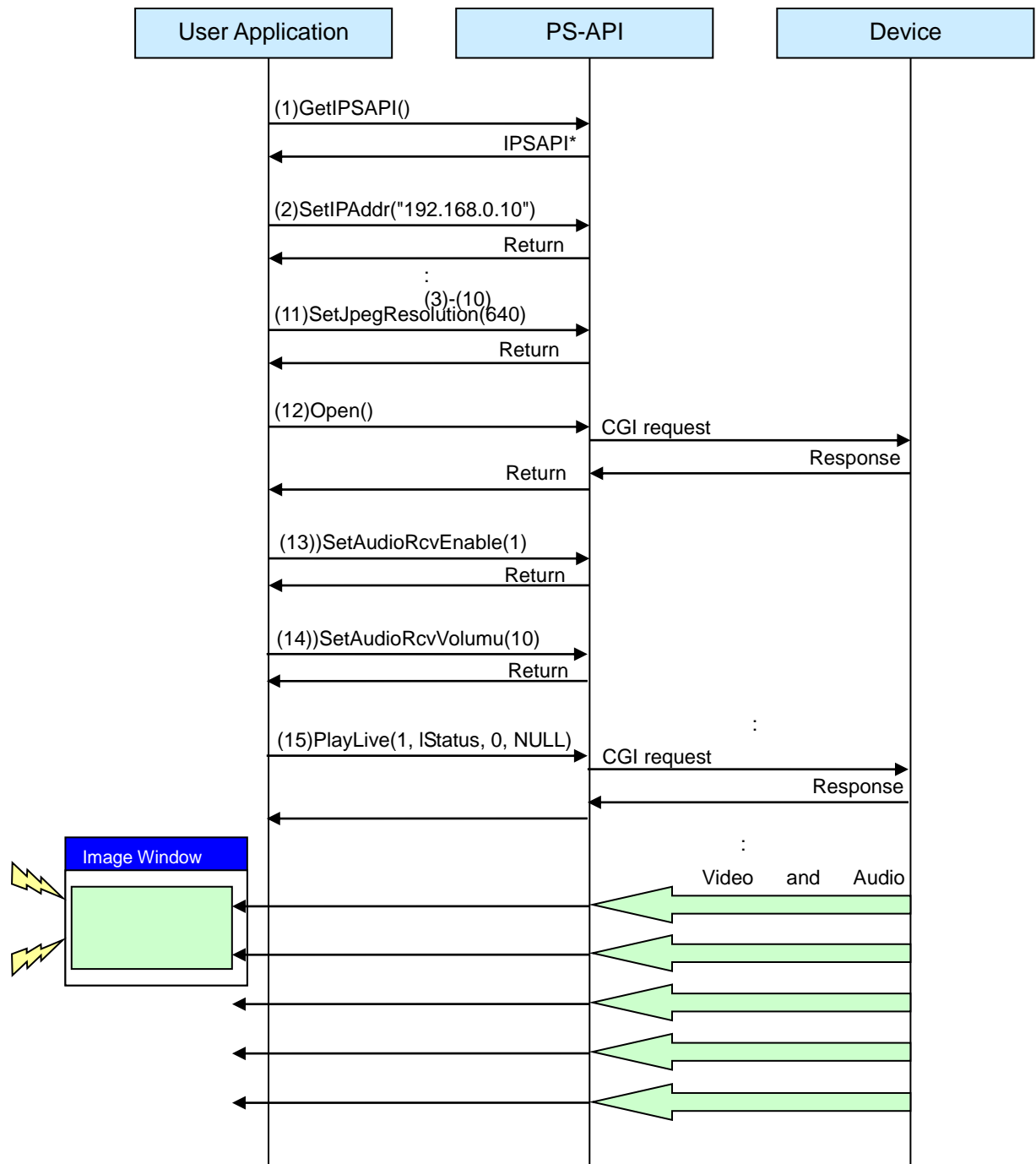
No.	Property / Method	Parameter	Description
21	AudioSend	command (long)	Start or Stop the audio transmission. When stops audio transmission, command is set to "0". e.g. 0

### Stop Live

No.	Property / Method	Parameter	Description
22	PlayControl	Command, Speed, Status, Blocking mode, Callback interface (long, long, long&, long, IAppCallBack*)	To stop live, set command to "1". Set speed to "1". Set the status to a long type variable . When using blocking mode, it is necessary that callback interface is set to "NULL". When using non-blocking mode, please set the callback interface to an instance to receive the result of async processing. (The callback interface can be set to "NULL", but application cannot receive the notification of finish processing.)  e.g.) 1, 1, IStatus, 0, NULL
23	SetAudioRcvEnable	Reception mode (long)	Set a mode whether receiving audio stream with PlayLive or Play into PS-API.  e.g.) 0
24	Close/Disconnect	-	Stop the communication with the target device. When using Close method, UID will be annulled.
25	DeleteIPSAPI	Pointer of IPSAPI (IPSAPI*)	Delete the IPSAPI instance.

## 6.12.2. Sequence

### Start Live with audio reception

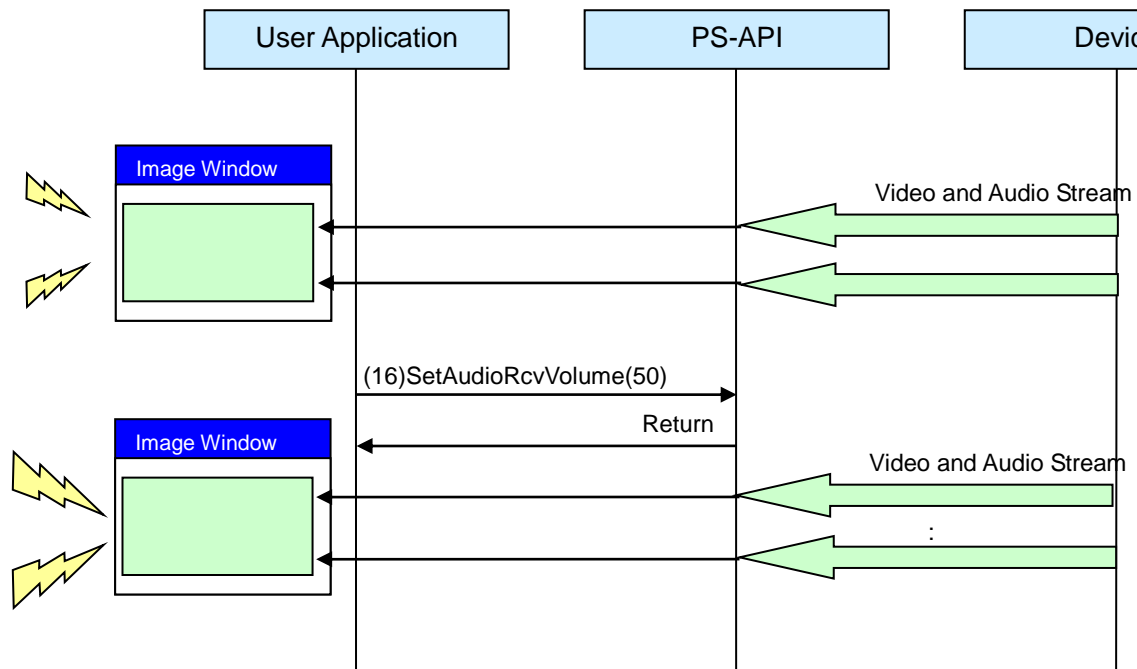


\* The timing of sending CGI command is not constant, depends on the device model or the device setting.

**Figure 6-25 Start Live with Audio**



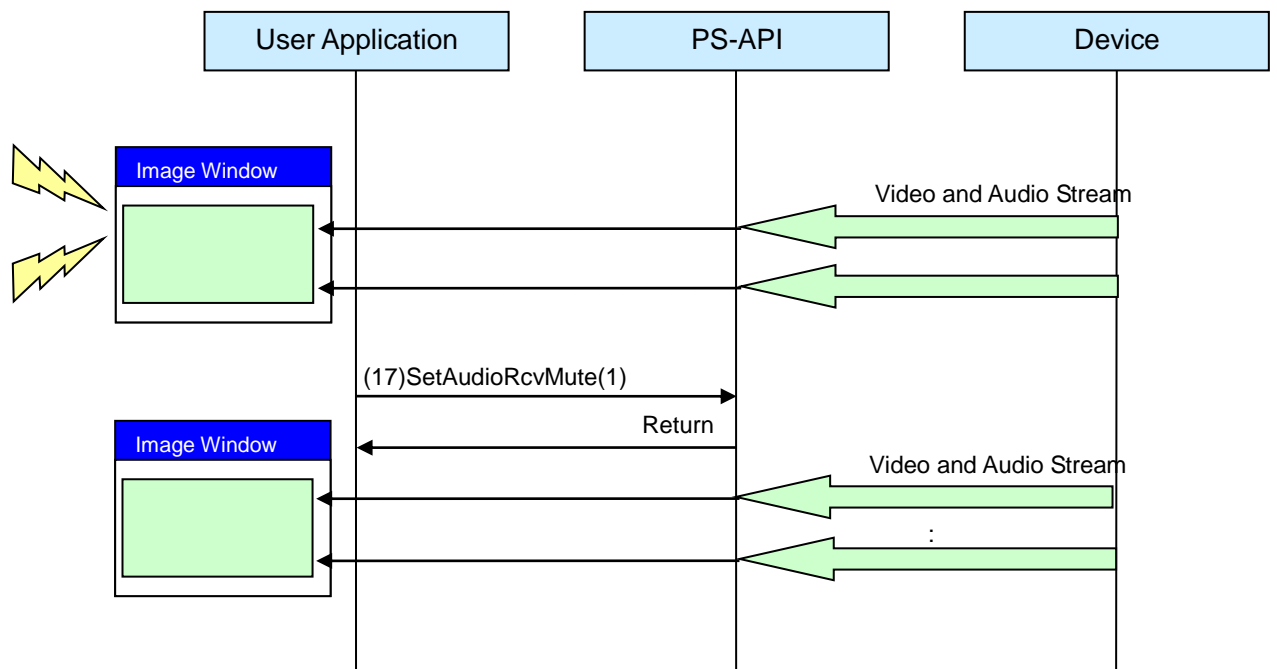
### Change volume for audio reception



\* The timing of sending CGI command is not constant, depends on the device model or the device setting.

**Figure 6-26 Change a volume**

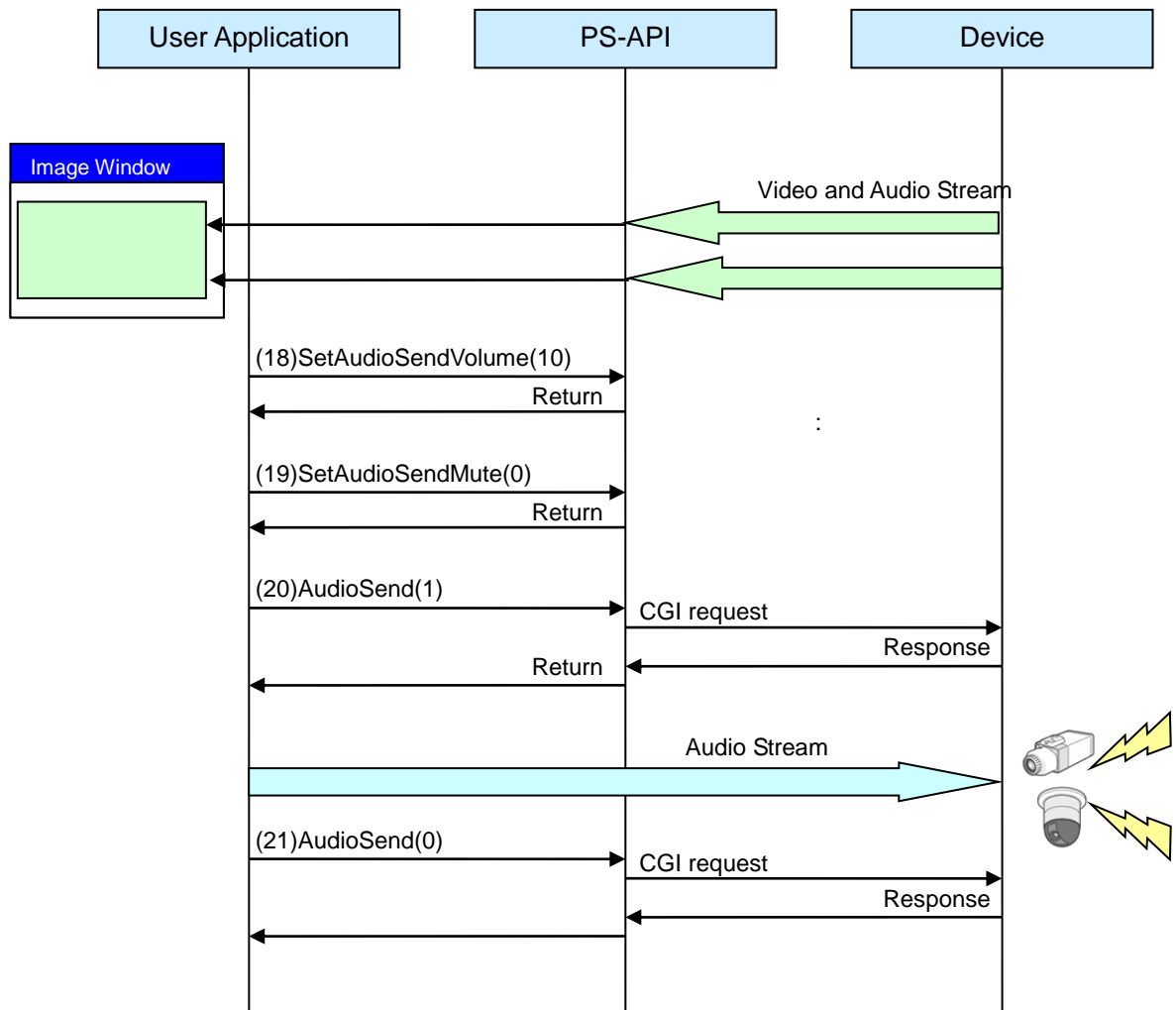
### Mute setting for audio reception



\* The timing of sending CGI command is not constant, depends on the device model or the device setting.

**Figure 6-27 Mute**

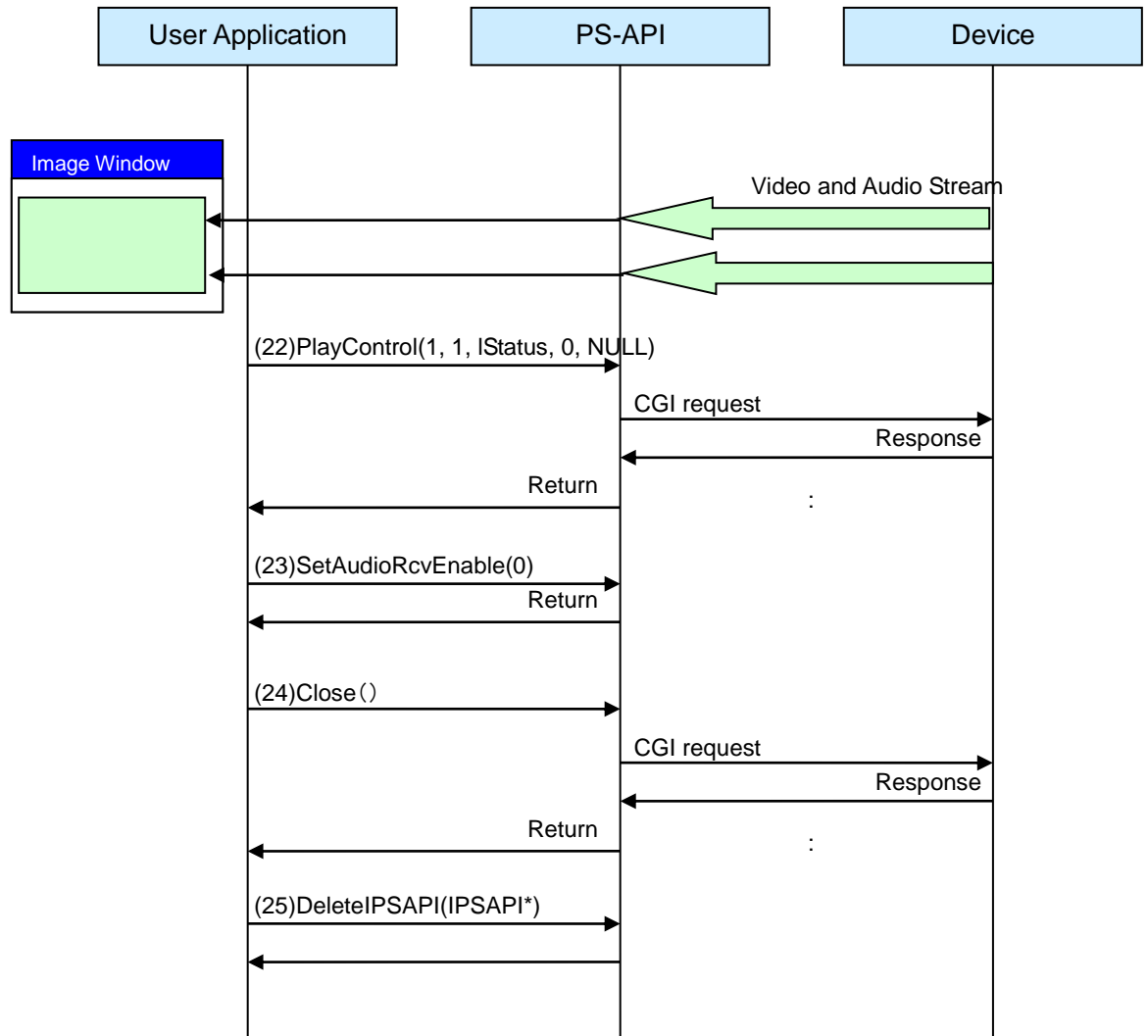
## Start and stop audio transmission



\* The timing of sending CGI command is not constant, depends on the device model or the device setting.

**Figure 6-28 Audio Transmission**

## Stop Live



\* The timing of sending CGI command is not constant, depends on the device model or the device setting.

**Figure 6-29 Stop Live**

## 6.13. SnapShot

### 6.13.1. Operation Procedure

#### Save Snapshot image

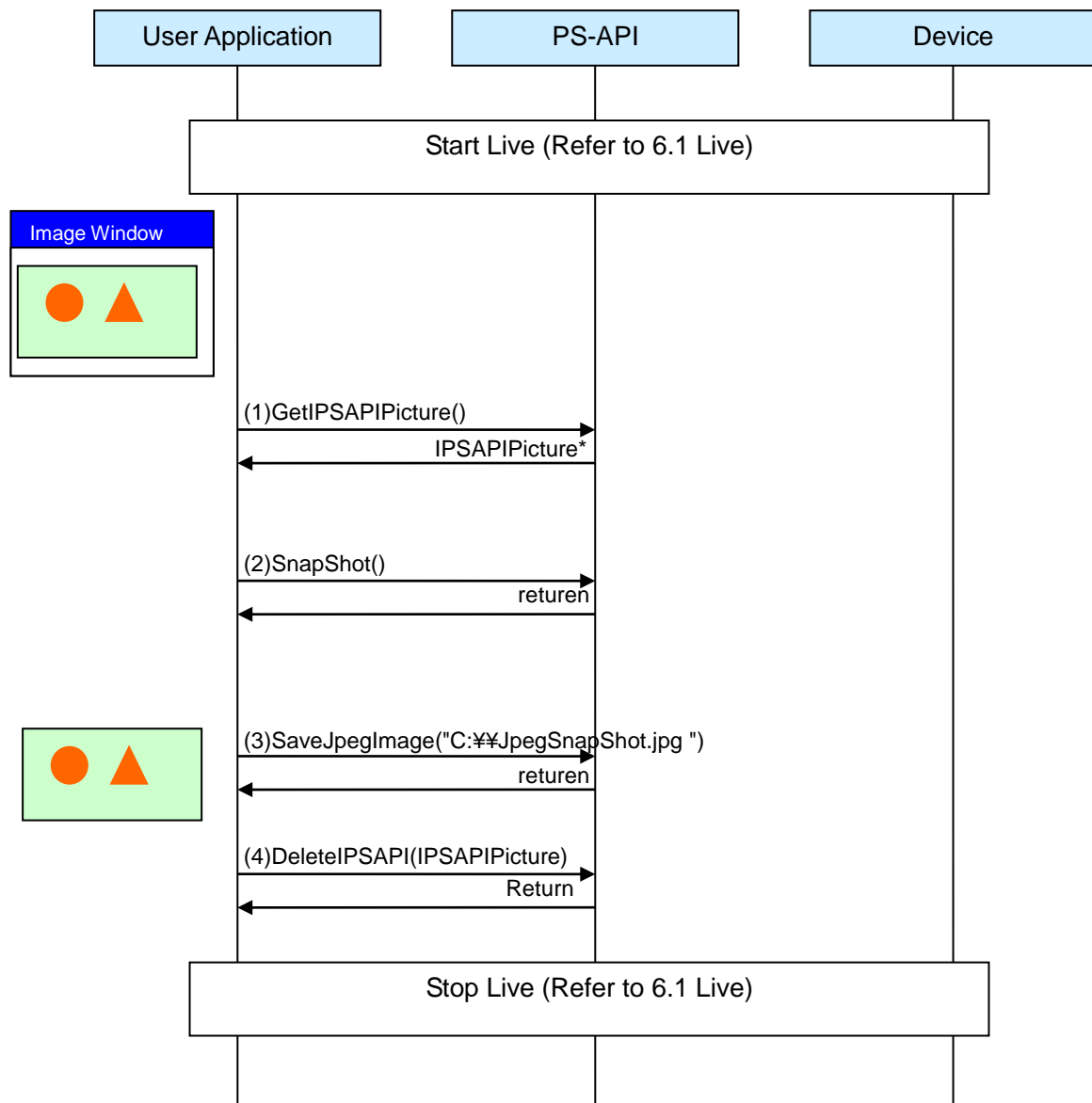
No.	Property / Method	Parameter	Description
-	<b>Start Live (Refer to 6.1 PlayLive)</b>		
1	GetIPSAPIPicture	—	Create the IPSAPIPicture instance for getting snapshot image. One instance is to get a snapshot image.
2	SnapShot	—	Store the displayed image on memory.
3	IPSAPIPicture::SaveJpegImage	File name (char*)	Store the displayed image in a jpeg format file. Specify the completed file path. "C:¥¥JpegSnapShot.jpg"
4	DeletelPSAPI	Pointer of IPSAPIPicture (IPSAPIPicture*)	Delete the IPSAPIPicture instance.
-	<b>Stop Live (Refer to 6.1 PlayLive)</b>		

#### Digital zoom

No.	Property / Method	Parameter	Description
-	<b>Start Live (Refer to 6.1 PlayLive)</b>		
1	DigitalZoom	Magnification (long)	Do digital zoom of the displayed image by specifying magnification. e.g.) 40
2	DigitalZoomMove	xPosition, yPosition  (long, long)	Move the displayed area during working a digital zoom. Specify distance of x direction and distance of y direction. e.g.) 320,0
-	<b>Stop Live (Refer to 6.1 PlayLive)</b>		

### 6.13.2. Sequence

#### Save Snapshot image



\* The timing of sending CGI command is not constant, depends on the device model or the device setting.

**Figure 6-30 SnapShot**

Digital zoom

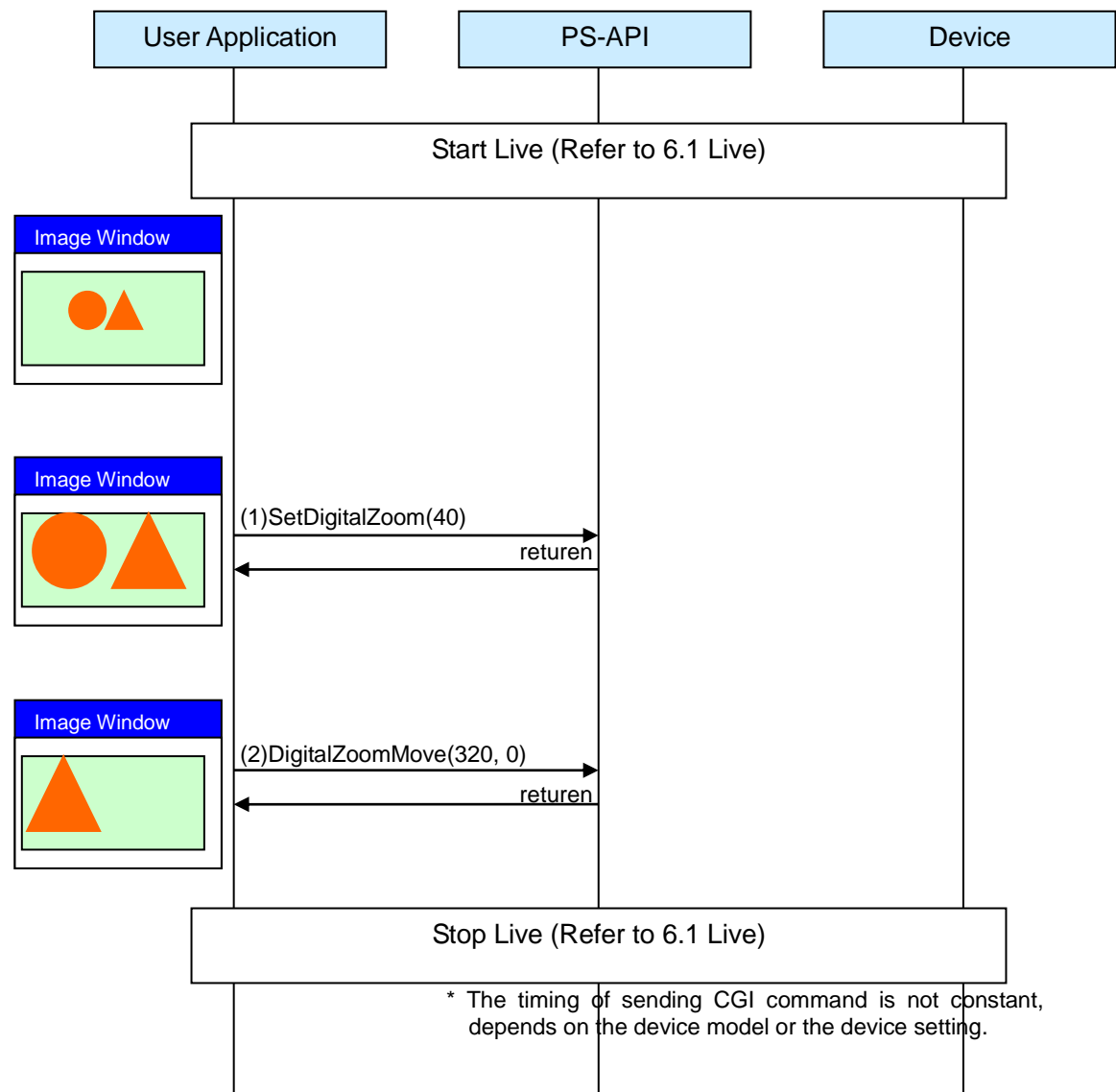


Figure 6-31 Digital Zoom

## 6.14. Overlay

### 6.14.1. Operation Procedure

#### Draw Title

No.	Property / Method	Parameter	Description
-	<b>Start Live (Refer to 6.1 PlayLive)</b>		
1	TitleOperation	id command text xPosition yPosition align font fontsize foreColor borderColor style (long, long, BSTR, long, long, long, BSTR, long, long, long, long)	Display text strings on the video image. Six texts can be displayed at the same time. Please use text id for recognizing each text. Cannot display the text out of PS-API control. Specify a font name that is installed on using OS.  e.g.) 1, 1, "PS-API", 100, 100, 0, "MS UI Gothic", 24, 0, 16777215, 2
-	<b>Stop Live (Refer to 6.1 PlayLive)</b>		

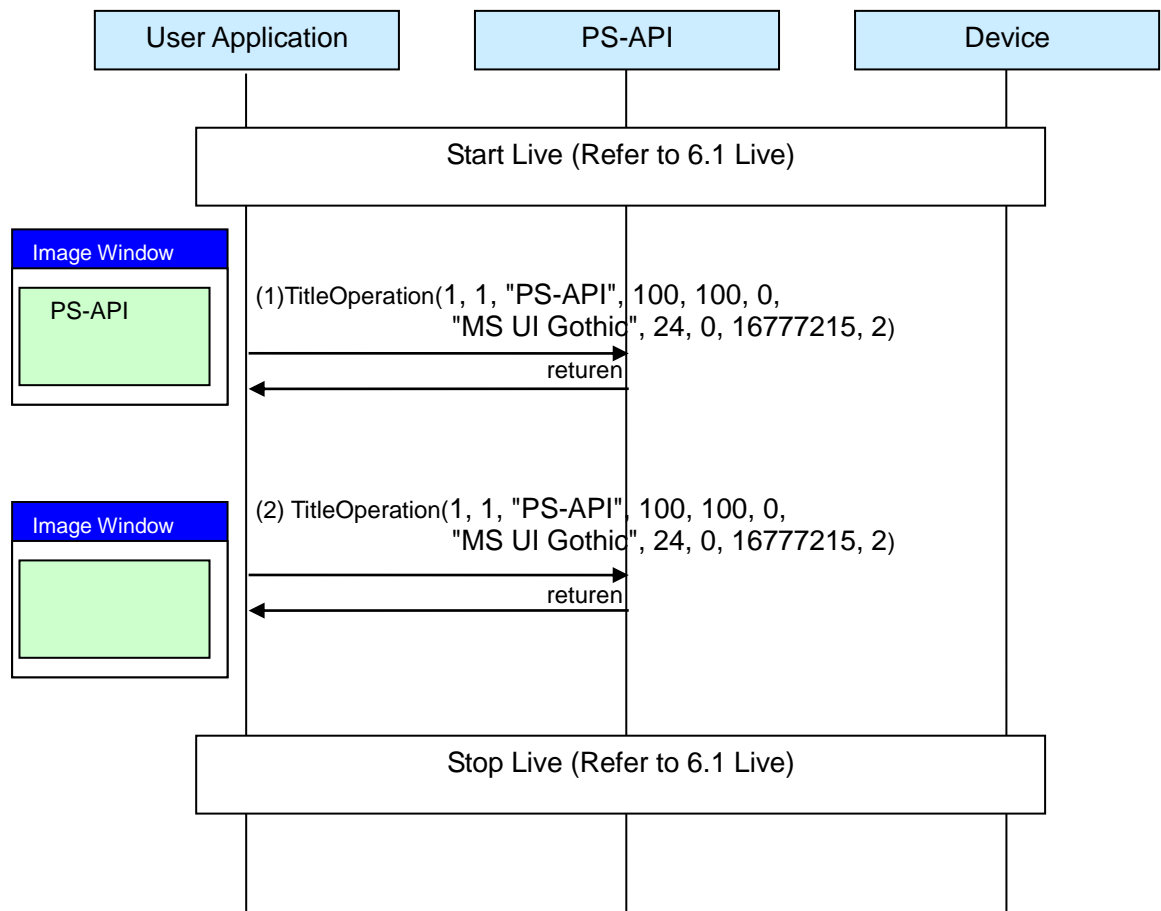
**Draw Box**

No.	Property / Method	Parameter	Description
-	<b>Start Live (Refer to 6.1 PlayLive)</b>		
1	BoxOperation	id command color size xTopLeft yTopLeft xBottomRight yBottomRight (long, long, long, long, long, long, long, long)	Display frame lines on the video image. Four boxes can be displayed at the same time. Please use box id for recognizing each box. Cannot display the frame lines out of PS-API control .  e.g.) 1, 2, 255, 3, 200, 200, 300, 300
-	<b>Stop Live (Refer to 6.1 PlayLive)</b>		



6.14.2. Sequence

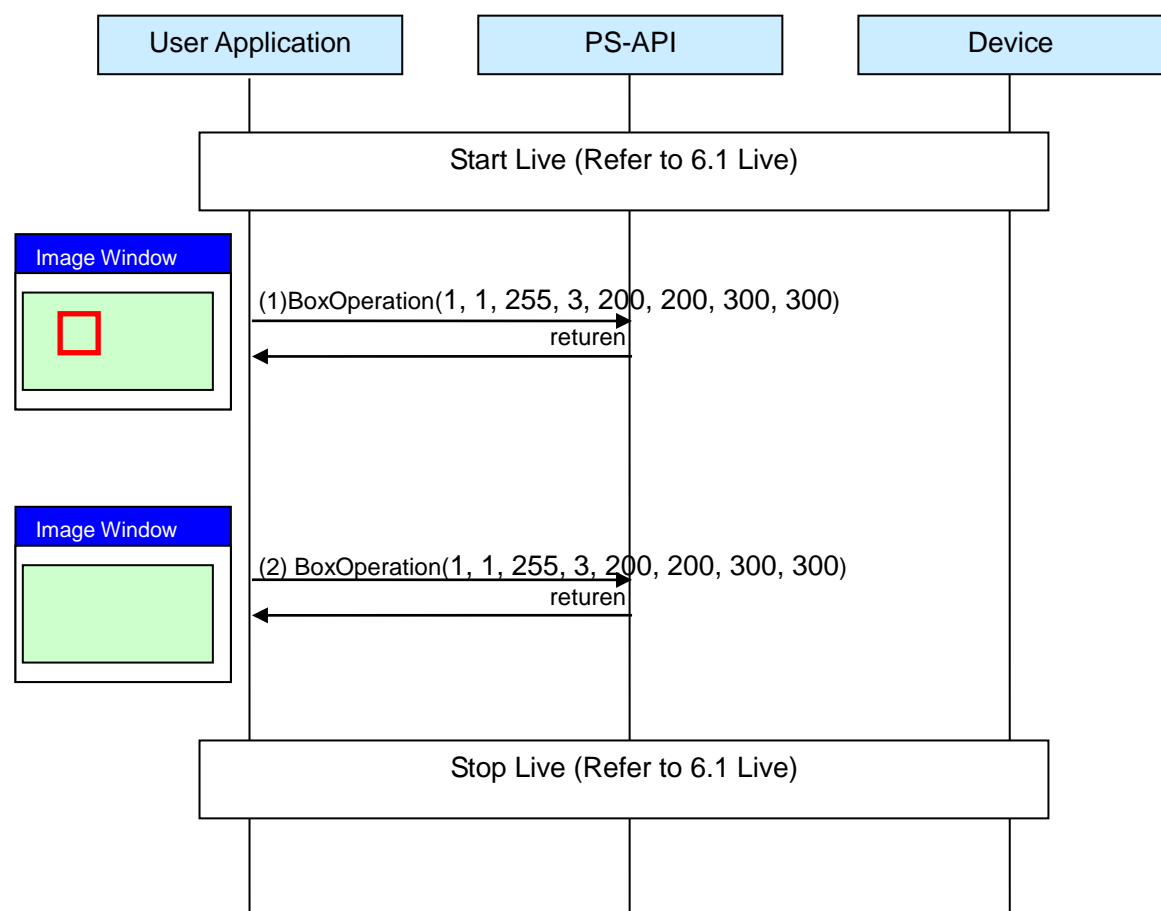
Draw Title



\* The timing of sending CGI command is not constant, depends on the device model or the device setting.

Figure 6-32 Display Text Strings

**Draw Box**



\* The timing of sending CGI command is not constant, depends on the device model or the device setting.

**Figure 6-33 Display Frame Lines**

## 6.15. VMDSearch

### 6.15.1. Operation Procedure

#### Create the IPSAPI Instance and Login

No.	Property / Method	Parameter	Description
-	<b>Create the IPSAPI Instance and Login (Refer to 6.2 Play)</b>		
1	GetISearchResultEx	-	Create the ISearchResultEx instance for getting search result list.

#### VMD search

No.	Property / Method	Parameter	Description
2	VmdSearchEx	channel startTimeDate endTimeDate mask aSensitivity axTopLeft ayTopLeft axBottomRight ayBottomRight bSensitivity bxTopLeft byTopLeft bxBottomRight byBottomRight cSensitivity cxTopLeft cyTopLeft cxBottomRight cyBottomRight dSensitivity dxTopLeft dyTopLeft dxBottomRight dyBottomRight imageWidth imageHeight Search result Blocking mode Callback (long, char*, char*, long, long, long, long, long, long, long, long, long, long, long, long,	Do VMD search. Specify channel, start time and date, end time and date, and searching area A – D as search condition. Specified the ISearchResultEx insrtance pointer, that is got by using GetISearchResultEx method, as argument to get search result. Set the status to a long type variable . When using blocking mode, it is necessary that callback interface is set to "NULL". When using non-blocking mode, please set the callback interface to an instance to receive the result of async processing. (The callback interface can be set to "NULL", but application cannot receive the notification of finish processing.)  e.g.) 1, "2010/07/01 00:00:00", "2010/07/01 23:59:59", 1, 1, 0, 0, 640, 480, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 640, 480, m_SearchResult, 0, NULL



**Get search result (1)**

No.	Property / Method	Parameter	Description
3	GetListCount	Total result count (long&)	The total of the search result.
4	GetAndDelete	Search result (SSEARCHRSLT_INFO*)	Get the search result. By using GetAndDelete method, the latest recording data information can be got, this recording data information is deleted from ISearchResultEx. Please get all search results by calling GetAndDelete repeatedly.

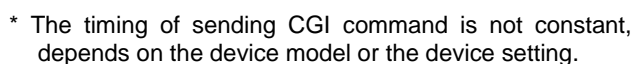
**Get search result (2)**

No.	Property / Method	Parameter	Description
3	GetListCount	Total result count (long&)	The total of the search result.
4	GetResult	Index, Search result (long SSEARCHRSLT_INFO*)	Get the search result. Specify the index of search result. The index of the latest recording data information is "1". Even if getting the search result by using GetResult, the recording data information isn't deleted from ISearchResultEx instance.  e.g.1)1, &stInfo

**Logout and Delete the IPSAPI Instance**

No.	Property / Method	Parameter	Description
5	DeleteISearchResultEx	Pointer of ISearchResultEx (ISearchResultEx*)	Delete the ISearchResultEx instance.
-	<b>Logout and Delete the IPSAPI Instance (Refer to 6.2 Play)</b>		

## VMD search



### Figure 6-34 VMD Search

## 6.16. Cropping

### 6.16.1.Operation Procedure

#### Start Live or Play or PlayFile

No.	Property / Method	Parameter	Description
-	Start Live (Refer to 6.1 PPlayLive) Play (Refer to 6.1 PPlay) PlayFile (Refer to 6.1 PPlayFile)		Cropping can be set even during execution.

#### Cropping

No.	Property / Method	Parameter	Description
1	SetCroppingEnabled	mode (long,)	1:Use Cropping
2	SetCroppingDrawRect	Id, ltX, ltY, rbX, rbY (long, long, long, long, long)	0 :Panorama image Specify the top-left X position of Cropping area. Specify the top-left Y position of Cropping area. Specify the bottom-right X position of Cropping area. Specify the bottom-right Y position of Cropping area.
3	SetCroppingRect	Id, ltX, ltY, rbX, rbY (long, long, long, long, long)	Cropping area id Specify the top-left X position of Cropping area. Specify the top-left Y position of Cropping area. Specify the bottom-right X position of Cropping area. Specify the bottom-right Y position of Cropping area.
4	SetCroppingDrawRect	Id, ltX, ltY, rbX, rbY (long, long, long, long, long)	Cropping area id Specify the top-left X position of Cropping area. Specify the top-left Y position of Cropping area. Specify the bottom-right X position of Cropping area. Specify the bottom-right Y position of Cropping area.

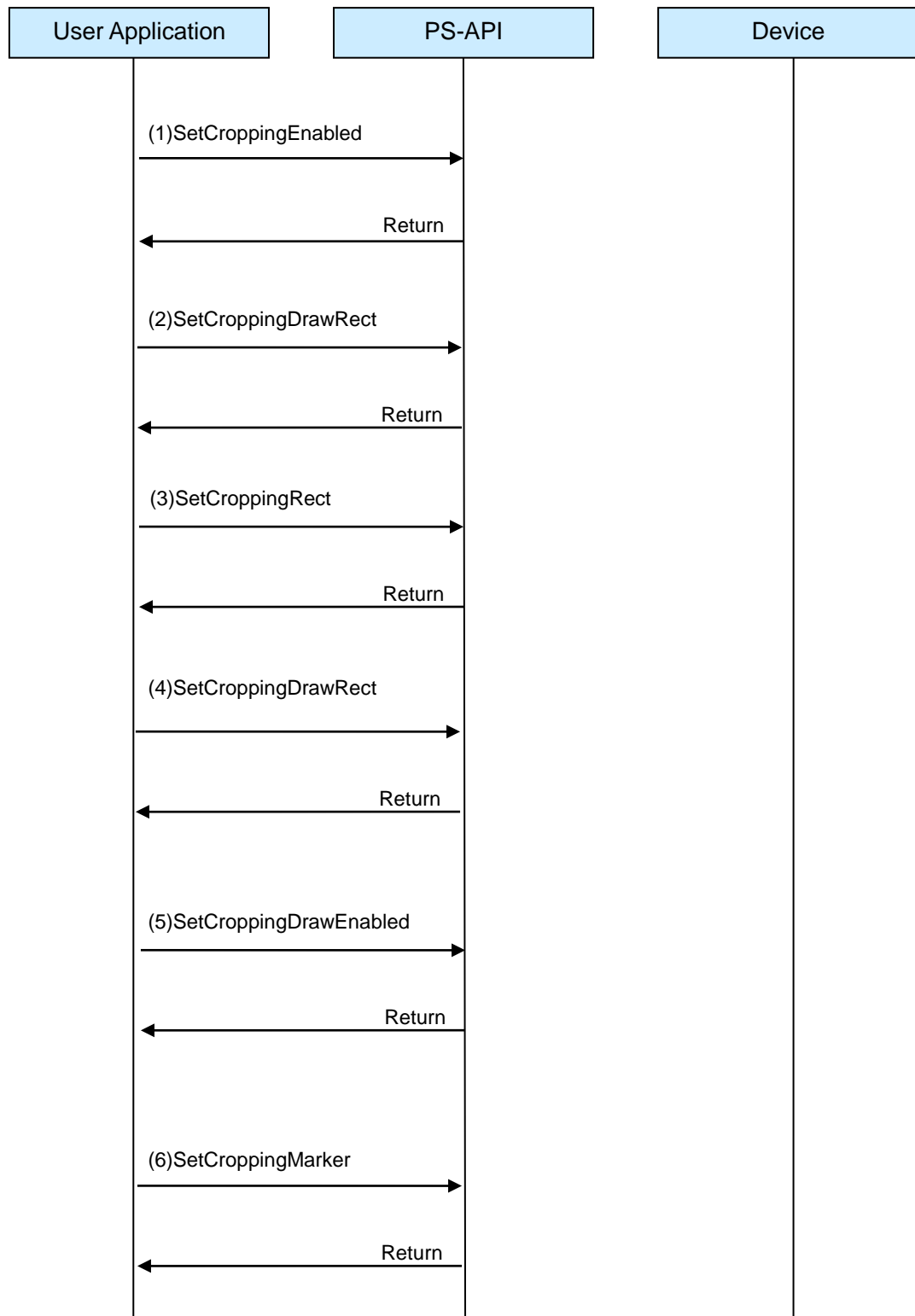
5	SetCroppingDrawEnabled	id, mode (long, long)	Cropping area id 1:show
6	SetCroppingMarker	id, mode, ltX, ltY, rbX, rbY, lineSize, lineColor, ellipseSize, ellipsecolor (long, long, long, long, long, long, long, long, long, long)	Cropping area id 1:Non edit mode Specify the top-left X position of Cropping area. Specify the top-left Y position of Cropping area. Specify the bottom-right X position of Cropping area. Specify the bottom-right Y position of Cropping area. Width of frame line Frame color Width of vertex Circle Circle color

#### Stop Live or Stop Play or Stop PlayFile

No.	Property / Method	Parameter	Description
-	Stop Live (Refer to 6.1 Live) Stop Play (Refer to 6.2 Play) Stop PlayFile (Refer to 6.3 PlayFile)		



### 6.16.2. Sequence



**Figure 6-35 Cropping**

## 6.17. HttpMP4Download

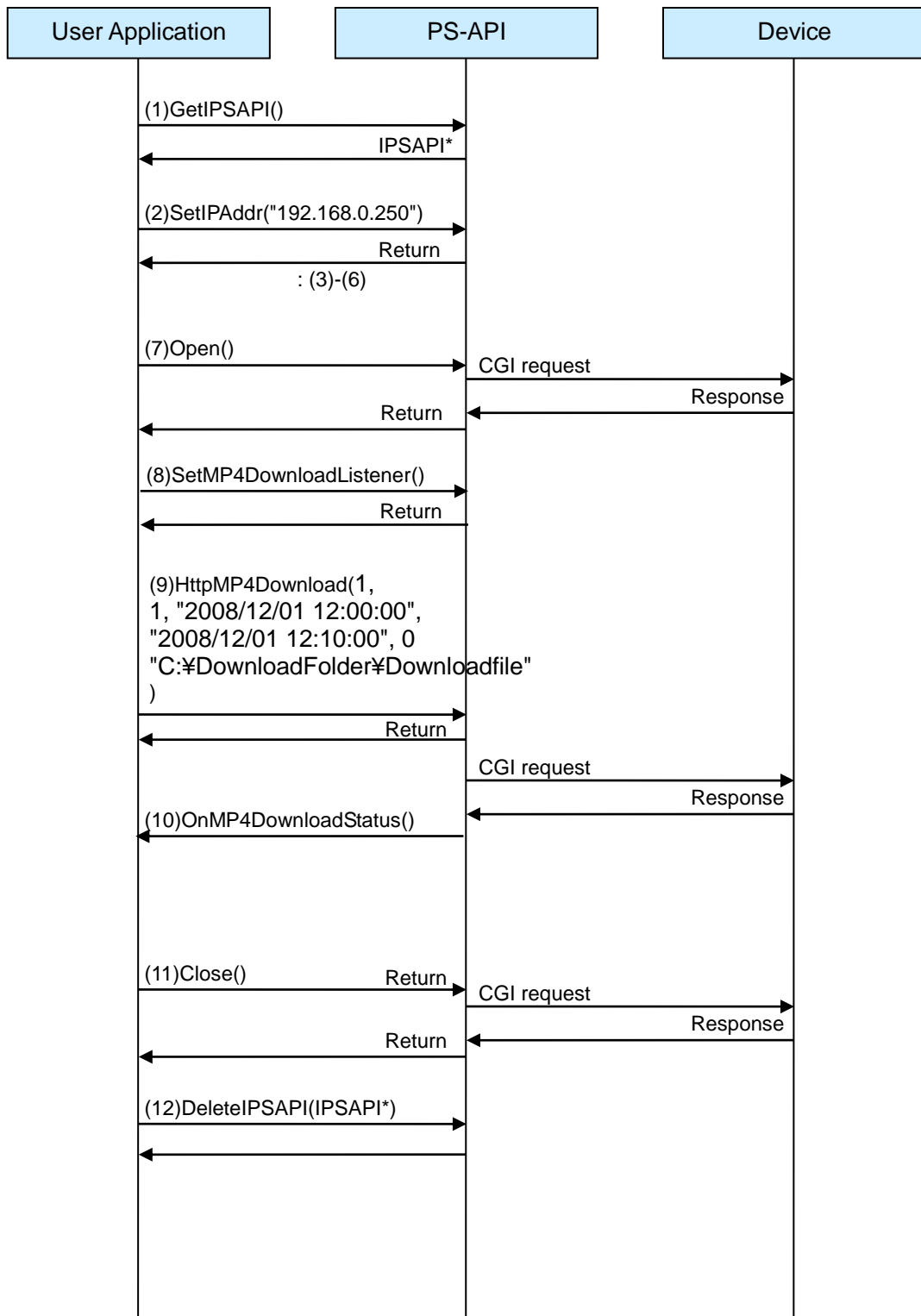
### 6.17.1. Operation Procedure

#### MP4 file Download(HTTP)

No.	Property / Method	Parameter	Description
1	GetIPSAPI	-	Create instance of IPSAPI. 1 instance can control 1 target device.
2	SetIPAddr	IP Address (char*)	Set IP Address into PS-API.  e.g.) "192.168.0.250"
3	SetDeviceType	Device type (long)	Set the device type corresponding to a NX series.  6
4	SetHttpPort	Port number (long)	Set Http port number into PS-API.  e.g.) 80
5	SetUserName	Character strings (char*)	Set user name into PS-API.  e.g.) ADMIN
6	SetPassword	Character strings (char*)	Set password into PS-API.  e.g.) 12345
7	Open/Connect	UID (long)	Connect to a target device. When getting a new UID, please use Open method.
8	SetMP4DownloadListener	Pointer of IAppListener (IAppListener*)	Set MP4 file download status notification.

No.	Property / Method	Parameter	Description
9	HttpMP4Download	channel, command, startTimeDate, endTimeDate, audioMode, fileName (long, long, char*, char*, long, char*)	Start HttpMP4Download. Specify channel, command, startTimeDate, endTimeDate, audioMode and fileName as download condition. (This method is non-blocking mode.)  e.g.) 1, 1, "2008/12/01 12:00:00", "2008/12/01 12:10:00", 0, "C:¥DownloadFolder¥Downloadfile"
10	OnMP4DownloadStatus	status filename (long char*)	Notified when MP4 download status changes.
11	Close/Disconnect	-	Stop the communication with the target device. When using Close method, UID will be annulled.
12	DeleteIPSAPI	Pointer of IPSAPI (IPSAPI*)	Delete the IPSAPI instance.

## 6.17.2. Sequence



\* The timing of sending CGI command is not constant, depends on the device model or the device setting.

**Figure 6-36 HttpMP4Download**

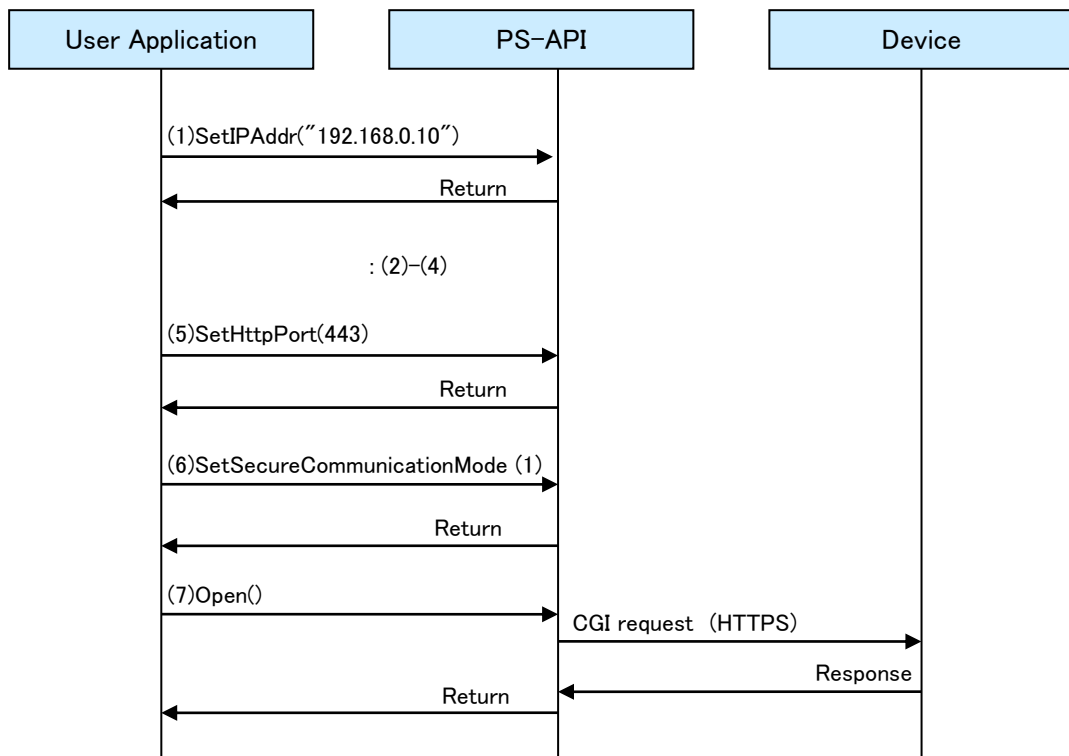
## 6.18. SSL

### 6.18.1. Operation Procedure

#### Open with SSL

手順	Property / Method	Parameter	Description
-	Set target device to HTTPS settings		
1	SetIPAddr	IP Address (BSTR)	Set IP Address into PS-API. e.g.) "192.168.0.10"
2	SetDeviceType	Device type (long)	Set the device type corresponding to a target device. e.g.) 2
3	SetUserName	Character strings (BSTR)	Set user name into PS-API. e.g.) admin
4	SetPassword	Character strings (BSTR)	Set password into PS-API. e.g.) 12345
5	SetHttpPort	Port number (long)	Set Https port number into PS-API. e.g.) 443
6	SetSecureCommunicationMode	HTTP/HTTPS (long)	Set Http or Https mode into PS-API. e.g.) 1
7	Open/Connect	UID (long)	Connect to a target device. When getting a new UID, please use Open method.
*	InternetMode	InternetMode (long)	Set internet mode into PS-API. e.g.) 1

### 6.18.2. Sequence



**Figure 6-37 Open with SSL**

## 6.19. MultiPlayLive\_SID

### 6.19.1. Operation Procedure

#### Start Live

No.	Property / Method	Parameter	Description
1	GetIPAPI	Refer to 6.1 PlayLive	Refer to 6.1 PlayLive
2	SetIPAddr	Refer to 6.1 PlayLive	Refer to 6.1 PlayLive
3	SetDeviceType	Refer to 6.1 PlayLive	Refer to 6.1 PlayLive
4	SetHttpPort	Refer to 6.1 PlayLive	Refer to 6.1 PlayLive
5	SetUserName	Refer to 6.1 PlayLive	Refer to 6.1 PlayLive
6	SetPassword	Refer to 6.1 PlayLive	Refer to 6.1 PlayLive
7	SetVideoWindow	Refer to 6.1 PlayLive	Refer to 6.1 PlayLive
8	SetImageWidth	Refer to 6.1 PlayLive	Refer to 6.1 PlayLive
9	SetImageHeight	Refer to 6.1 PlayLive	Refer to 6.1 PlayLive
10	SetStreamFormat	Refer to 6.1 PlayLive	Refer to 6.1 PlayLive
11	SetJPEGResolution/ SetMPEG4Resolution/ SetH264Resolution	Refer to 6.1 PlayLive	Refer to 6.1 PlayLive
12	Open/Connect	Refer to 6.1 PlayLive	Refer to 6.1 PlayLive Call Open method on the first instance. For the second and subsequent instances, connect using the UID obtained by the first Open method.
13	GetSIDInfo	-	Call method only one instance. Get the information of StreamID support, the maximum number of StreamID that recorder can issue, and the number of StreamID which is in use. If SIDMode is not supported, refer to the sequence of 6.9 MultiPlay Live.  e.g.) struct SSID_INFO{ long m_IMode=1 ( support StreamID ) long m_IMaxCount;= 64 ; long m_IInUseCount;10; }  
14	SetSIDMode	StreamID Mode(long)	Set SIDMode on every instance that perform PlayLive method.  e.g.) 1
13	PlayLive	Refer to 6.1 PlayLive	Refer to 6.1 PlayLive

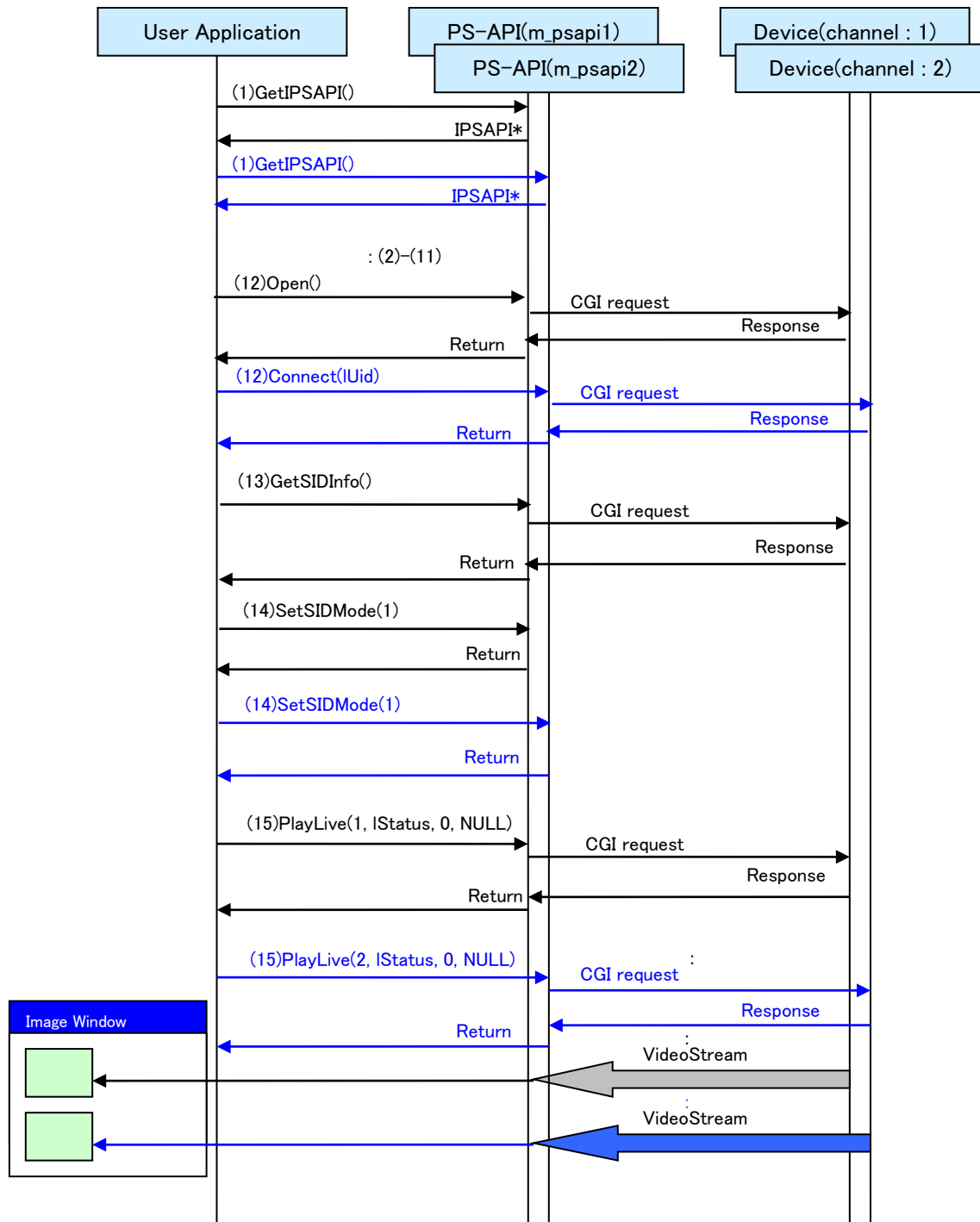
**Stop Live**

No.	Property / Method	Parameter	Description
14	PlayControl	Refer to 6.1 PlayLive	Refer to 6.1 PlayLive
15	Close/Disconnect	Refer to 6.1 PlayLive	Refer to 6.1 PlayLive
16	DeleteIPSAPI	Refer to 6.1 PlayLive	Refer to 6.1 PlayLive



## 6.19.2. Sequence

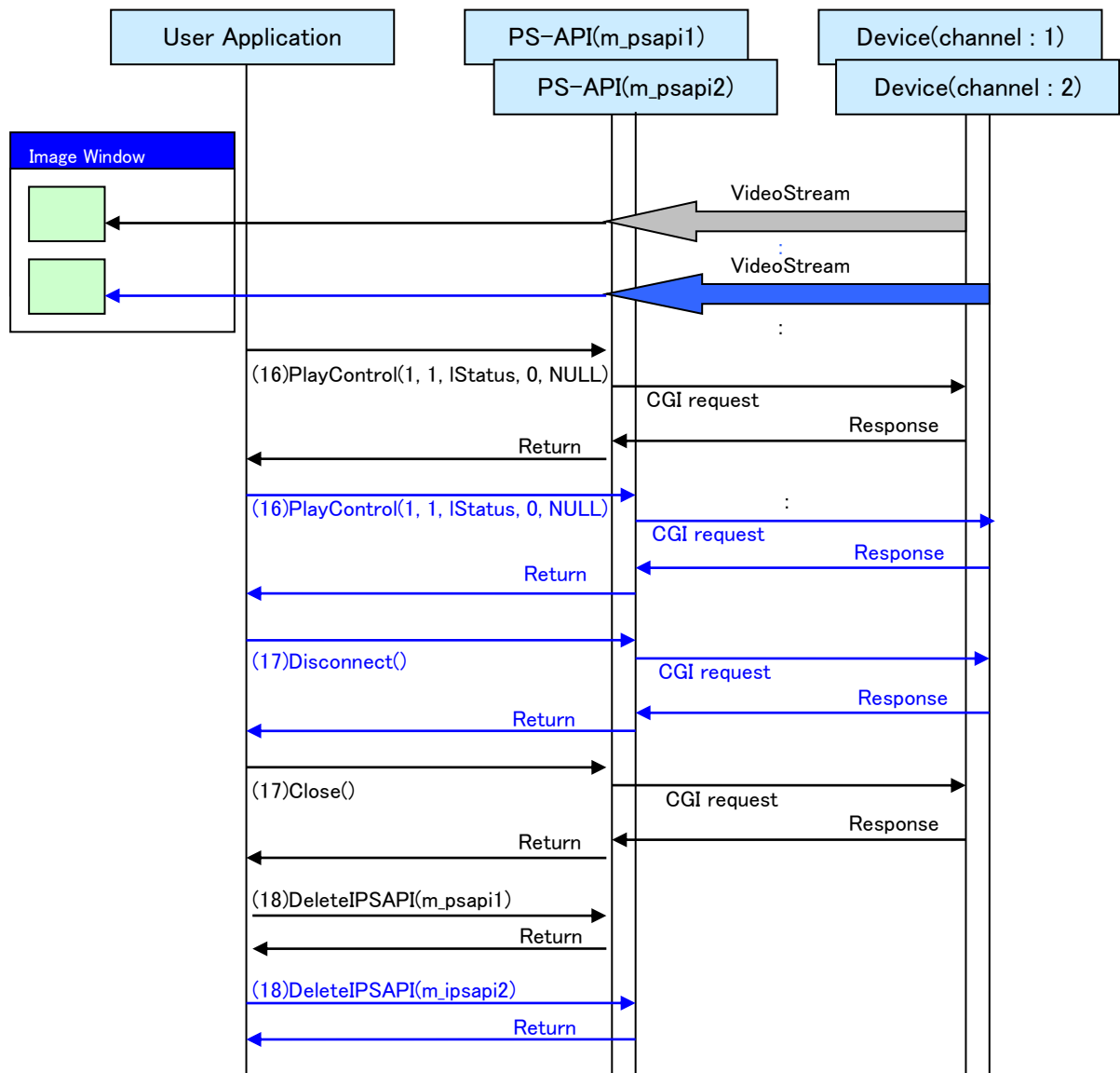
### Start Live



\* The timing of sending CGI command is not constant, depends on the device model or the device setting.

**Figure 6-38 Start Live**

## Stop Live



\* The timing of sending CGI command is not constant, depends on the device model or the device setting.

Figure 6-39 Stop Live

## 7. Error Code List

The error code is defined by the following format. (decimal, 8digits)

-D<sub>1</sub>A<sub>1</sub>C<sub>1</sub>M<sub>1</sub>M<sub>2</sub>Z<sub>1</sub>Z<sub>2</sub>Z<sub>3</sub>

-D <sub>1</sub> Category	A <sub>1</sub> Level	C <sub>1</sub> Source	M <sub>1</sub> M <sub>2</sub>	Z <sub>1</sub> Z <sub>2</sub> Z <sub>3</sub> Code	Error Description
-1: Common	1:error 2:warning	1:PS-API error 2:communication error 3:device error	(internal use)	001	Invalid Pointer
				002	Invalid State
				003	SDK Busy
				004	SDK internal error
				:	
				012	
				013	Request Not Suported
				014	Parameter is invalid
				015	Specific Memory Insufficient
				016	SDK internal error
				:	
				025	
				029	Async method queuing limit exceeded error

-D1 Category	A1 Level	C1 Source	M1M2	Z1Z2Z3 Code	Error Description
-2: Video	1:error 2:warning	1:PS-API error 2:communication error 3:device error 4:PS-API error 5:PS-API error	(internal use)	001 : 009	Video module initialize error
				010	Parameter Error (ProxyName, ProxyPort)
				011 : 030	Video module initialize error
				031	Failed to create decoder filter
				032 : 088	Video module initialize error
				089	Network error
				090	Open Socket error
				091	Close Socket error
				092 : 095	Network error for Live (MPEG-4/H.264/H.265)
				096 : 099	Network error for Play
				100 : 103	Error for FilePlay
				104 : 107	Network error for Live (JPEG)
				108 : 111	Network error for HDR
				112 : 129	SDK internal error
				130 131	Setting UID error (MPEG-4/H.264/H.265)
				132	Obtaining Device Info error(MPEG-4/H.264/H.265)
				133	Obtaining Play Status error(MPEG-4/H.264/H.265)
				134	MultiScreenChannel error (MPEG-4/H.264/H.265)
				135	SDK internal error
				136 137	Setting UID error (JPEG)
				138	Obtaining Device Info error(JPEG)
				139	Obtaining Play Status error(JPEG)
				140	MultiScreenChannel error (JPEG)
				141	SDK internal error

-D1 Category	A1 Level	C1 Source	M1M2	Z1Z2Z3 Code	Error Description
-2: Video	1:error 2:warning	1:PS-API error 2:communication error 3:device error 4:PS-API error 5:PS-API error	(internal use)	142 143	Setting UID error (HDR)
				144	Obtaining Device Info error(HDR)
				145	Obtaining Play Status error(HDR)
				146	MultiScreenChannel error (HDR)
				147	SDK internal error
				148	Live Operation Error
				149	Play Operation Error
				150	FilePlay Operation Error
				151	SDK internal error
				152	FileFormat Specified error
				153 : 163	SDK internal error
				164	Device UID is exceeded
				165	Specified video format is incorrect
				166	Specified video stream number is incorrect
				167	channel number is incorrect
				168 : 171	SDK internal error
				186 : 191	SDK internal error
				192	Internet mode setting is mismatch Error.

-D1 Category	A1 Level	C1 Source	M1M2	Z1Z2Z3 Code	Error Description
-2: Video	1:error 2:warning	1:PS-API error 2:communication error 3:device error 4:PS-API error 5:PS-API error	(internal use)	200	Parameter Error (DeviceType)
				201	Parameter Error (ImageFormat)
				202	SDK internal error
				203	Parameter Error (HttpPort)
				204	Parameter Error (AccessType error
				205	Parameter Error (ProxyPort)
				206	SDK internal error
				:	
				214	
				215	Parameter Error (Channel)
				216	SDK internal error
				217	Parameter Error (MulticastPort)
				218	SDK internal error
				:	
				223	
				224	Parameter Error (IpAddress)
				225	Parameter Error (ProxyName)
				226	Parameter Error (UID error
				227	Parameter Error (UserName)

-D1 Category	A1 Level	C1 Source	M1M2	Z1Z2Z3 Code	Error Description
-2: Video	1:error 2:warning	1:PS-API error 2:communication error 3:device error 4:PS-API error 5:PS-API error	(internal use)	228	Parameter Error (Password)
				229	SDK internal error
				230	Parameter Error (MultiCastAddress)
				231	SDK internal error
				232	Parameter Error (PlayFile file name)
				233	SDK internal error
				234	Device type and StreamFormat not mutch.
				235 : 237	SDK internal error
				238	Parameter Error (DateTime)
				239	Parameter Error (Resolution)
				240	Parameter Error (Play speed)
				241	CGI send error
				242	Parameter Error (StreamFormat)
				243	Parameter Error (Listener)
				244	Parameter Error (StreamNumber, NXStreamNumber)
				245	Parameter Error (TransFrameRate)
				246	SDK internal error
				247	Parameter Error (AutoMulticast)
				248	Parameter Error Data is NULL
				249	Parameter Error Data size invalid
				250	Parameter Error AudioRcvEnable/AudioSend
				251	Parameter Error AudioRcvVolume/ AudioSendVolume
				252	Parameter Error AudioRcvMute/ AudioSendMute
				253	SDK internal error
				254	Parameter Error SnapShot
				255	No data error (SnapShot)

-D1 Category	A1 Level	C1 Source	M1M2	Z1Z2Z3 Code	Error Description
-2: Video	1:error 2:warning	1:PS-API error 2:communication error 3:device error 4:PS-API error 5:PS-API error	(internal use)	256	Parameter Error Overlay ID
				257	Parameter Error Overlay ICommand
				258	Parameter Error Overlay IText
				259	Parameter Error Overlay lxPosition
				260	Parameter Error Overlay lyPosition
				261	Parameter Error Overlay lalign
				262	Parameter Error Overlay lfont
				263	Parameter Error Overlay lfontsize
				264	Parameter Error Overlay lforeColor
				265	Parameter Error Overlay lborderColor
				266	Parameter Error Overlay lstyle
				267	Parameter Error Overlay lcommand
				268	Parameter Error Overlay lcolor
				269	Parameter Error Overlay lsize
				270	Parameter Error Overlay lxTopLeft
				271	Parameter Error Overlay lyTopLeft
				272	Parameter Error Overlay lxBottomRight
				273	Parameter Error Overlay lyBottomRight
				274	Parameter Error GetTitle
				275	SDK internal error
				276	SDK internal error
				277	SDK internal error
				278	SDK internal error
				279	SDK internal error



-D1 Category	A1 Level	C1 Source	M1M2	Z1Z2Z3 Code	Error Description
-2: Video	1:error 2:warning	1:PS-API error 2:communication error 3:device error 4:PS-API error 5:PS-API error	(internal use)	280	Parameter Error DigitalZoom
				281	Parameter Error DigitalZoomMove
				282	Parameter Error DigitalZoomMode
				283	Parameter Error BackColor
				284 : 288	SDK internal error
				289	Parameter Error PlayControlByTime
				290	Parameter Error DST
				291	Parameter Error PictureFitMode
				292	Parameter Error DecResolutionMode
				293	Parameter Error FilePassword
				294	Parameter Error InternetMode
				297	Parameter Error Overlay transmissivity
				298	Parameter Error Overlay Bitmap file
				299	Parameter Error Overlay maskColor

-D1 Category	A1 Level	C1 Source	M1M2	Z1Z2Z3 Code	Error Description
-2: Video	1:error 2:warning	1:PS-API error 2:communication error 3:device error 4:PS-API error 5:PS-API error	(internal use)	300	Parameter is not set
				301	Duplicate Open error
				302	Already Open/Connect error
				303	Error occurred Not
				:	Open/Connect error
				305	
				306	Live operation fails because Live is not started
				307	Play operation fails because Play is not started
				308	PlayFile operation fails because PlayFile is not started
				309	Play/PlayFile/DecodeImage/ HttpMP4Download operation fails because Live is starting
				310	Live/PlayFile/DecodeImage/ HttpMP4Download operation fails because Live is starting
				311	Live/Play/DecodeImage/ HttpMP4Download operation fails because Live is starting
				312	Play for Camera/Encoder is not available
				313	SDK internal error
				:	
				315	
				316	Stopping AudioSend fails because AudioSend is already stopped
				317	SDK internal error
				318	Starting AudioSend fails because AudioSend is already started
				319	Receive Audio is disconnected
				320	Send Audio is disconnected
				321	Device does not support audio Device setting is invalid
				322	Device does not support format
				323	DigitalZoomMove error because DigitalZoom is x1.
				324	SDK internal error
				:	
				330	
				331	Specified password for n3r/n3a invalid.
				332	SDK internal error
				:	
				336	
				337	Video is not updating for a while.

-D1 Category	A1 Level	C1 Source	M1M2	Z1Z2Z3 Code	Error Description
-2: Video	1:error 2:warning	1:PS-API error 2:communication error 3:device error 4:PS-API error 5:PS-API error	(internal use)	342	Device doesn't support StreamID.
				343	Fail to start Live/Network playback by StreamID exhaustion.
				344	Fail to start Live/Network playback by the error except StreamID exhaustion.
				345	StreamID parameter error.
				346	Failure of keep-alive for StreamID.
				347	StreamID invalid.
				349	Decode by the upper limit of resolution is not supported.
				399	SSL Communication is not Available

-D1 Category	A1 Level	C1 Source	M1M2	Z1Z2Z3 Code	Error Description
-2: Video	1:error 2:warning	1:PS-API error 2:communication error 3:device error 4:PS-API error 5:PS-API error	(internal use)	401	Failed to create thread.
				402	Failed to create event handle
				403	Failed to create thread.
				404	
				405	Memory error
				406	Library load error.
				407	SDK internal error.
				408	
				409	Failed to allocate memory.
				501	Audio operation invalid.
				502	DeviceType is invalid for Audio operation.
				503	SDK internal error.
				504	
				505	Failed to create thread.
				506	Failed to delete thread.
				507	Failed to create thread.
				508	Failed to delete thread.
				509	SDK internal error.
				510 : 512	Device returns error for audio related CGI.
				513	Failed to create socket.
				514	Failed to delete socket.
				515	Failed to start AudioSend. Another client sending audio.
				516	Specified channel does not support Audio.
				517	Device does not support sending audio from client.
				518	Device does not support receiving audio by client.
				519	Device does not support sending audio from client.
				520 : 582	SDK internal error.
				583 : 590	The audio device of PC is not correctly set.

-D1 Category	A1 Level	C1 Source	M1M2	Z1Z2Z3 Code	Error Description
-2: Video	1:error 2:warning	1:PS-API error 2:communication error 3:device error 4:PS-API error 5:PS-API error	(internal use)	591 : 612	SDK internal error.
				702	Parameter Error Port range value
				704	Parameter Error GetSIDInfo value.
				705	Parameter Error SIDMode value.
				710	Parameter error CamSnapShot channel
				711	Parameter error CamSnapShot mode
				712	CamSnapShot Not supported
				752	Parameter is out of range. HttpMP4download channel
				753	Parameter is out of range. HttpMP4download command
				754	Parameter Error. (startTimeDate)
				755	Parameter Error. (endTimeDate)
				756	Parameter is out of range. HttpMP4download audioMode
				757	Parameter is illegal characters or exceeded of characters. HttpMP4download filename
				770	Time difference is 60 minutes or more. HttpMP4Download startTime and endTime
				779	Parameter is out of range Cropping id
				780	Parameter is out of range Cropping mode
				781	Parameter is out of range Cropping rate
				782	Parameter is out of range Cropping marker size
				783	Parameter is out of range Cropping marker color
				784	Position coordinate error CroppingDrawRect (ltX, ltY) > (rbX, rbY)
				785	Parameter is out of range Cropping MaxRate
				786	Position coordinate error CroppingMarker (ltX, ltY) > (rbX, rbY)
				787	Position coordinate error CroppingRect (ltX, ltY) > (rbX, rbY)

-D1 Category	A1 Level	C1 Source	M1M2	Z1Z2Z3 Code	Error Description
-2: Video	1:error 2:warning	1:PS-API error 2:communication error 3:device error 4:PS-API error 5:PS-API error	(internal use)	809	HttpMP4Download is not started
				810	HttpMP4Download is starting
				811	Recording data is JPEG only
				812	No recording data at the specified time
				813	Device MP4download number is exceed
				837	HttpDownload is not started
				838	HttpDownload is starting
				839	Recording data is JPEG only
				840	No recording data at the specified time
				841	Device download number is exceed

-D1 Category	A <sub>1</sub> Level	C <sub>1</sub> Source	M <sub>1</sub> M <sub>2</sub>	Z <sub>1</sub> Z <sub>2</sub> Z <sub>3</sub> Code	Error Description
-3: Control	1:error 2:warning	1:PS-API error 2:communication error 3:device error	(internal use)	201	Parameter Error (DeviceType)
				202	Parameter Error (AccessType)
				203	Parameter Error (IpAddr)
				204	Parameter Error (HttpPort)
				205	Parameter Error (HttpTimeout)
				206	Parameter Error (ProxyName)
				207	Parameter Error (ProxyPort)
				208	Parameter Error (Internal parameter)
				209	Parameter Error (Internal parameter)
				210	Parameter Error (UserName)
				211	Parameter Error (Password)
				212	Parameter Error (Internal parameter)
				310 : 349	Parameter Error

-D1 Category	A <sub>1</sub> Level	C <sub>1</sub> Source	M <sub>1</sub> M <sub>2</sub>	Z <sub>1</sub> Z <sub>2</sub> Z <sub>3</sub> Code	Error Description
-3: Control	1:error 2:warning	1:PS-API error 2:communication error 3:device error	(internal use)	601	Request failed by internal state.
				602	The response of the target device is incorrect.
				603	Username or password is incorrect.
				604	The method is executed for not-supported device.
				605	UID is invalid
				606	Specified control (Pan/Tilt or Zoom) is not supported for this device, specified control is invalid.
				607	Focus is not supported for this device, specified control is invalid.
				608	Pan/Tilt/Zoom has priority, specified control is invalid.
				609	There is no device information of the specified channel.
				610	Auto Pan/Auto Track/Auto Focus is not supported for this device, specified operation is invalid.
				611	This method is already executed.
				612	Trigger on/Trigger off is not supported for this device, specified operation is invalid.
				613	Trigger on/Trigger off is not supported for this device, specified operation is invalid.
				614	Search condition is not specified.
				615	DeviceType mismatch for the target device.
				616	Delete Preset is not supported for this device.
				617	SD backup condition doesn't have priority.
				618	StartTime and EndTime is same, or EndTime is smaller than StartTime.
				619	The specified time and date (2035/01/01 00:00:00 or after) is invalid.
				620	(Internal use)
				621	(Internal use)
				622	Set Preset is invalid during Auto Pan.
				623	Home position cannot be set/deleted.



-D1 Category	A <sub>1</sub> Level	C <sub>1</sub> Source	M <sub>1</sub> M <sub>2</sub>	Z <sub>1</sub> Z <sub>2</sub> Z <sub>3</sub> Code	Error Description
-3: Control	1:error 2:warning	1:PS-API error 2:communication error 3:device error	(internal use)	624	Login error (user excess)
				625	Login error (network impossible)
				626	Login error (host attestation)
				627	Login error (config state)
				628	Login error (measurement)
				629	Login error (config user attestation)
				630	Login error (rebooting state)
				631	Login error (sleep state)
				632	Login error (play only state)
				633	Login error (no disk)
				634	Login error (IP easy setting state)
				635	Login error (change setting state)
				636	Login error (blackout)
				637	Login error (system error)
				638	Login error
				639	Receive no-content response from a target device.
				640	ABF control is not supported for this device.
				641	Super Dynamic control is not supported for this device.
				642	Search method is not executed.
				643	There is no current device information.
				644	There is no Keyword in the current device information.
				645	Preset control is not supported for this device.
				646	In case of VMD search, multiple channel cannot be specified.

-D1 Category	A <sub>1</sub> Level	C <sub>1</sub> Source	M <sub>1</sub> M <sub>2</sub>	Z <sub>1</sub> Z <sub>2</sub> Z <sub>3</sub> Code	Error Description
-3: Control	1:error 2:warning	1:PS-API error 2:communication error 3:device error	(internal use)	647	Preset sequence control is not supported for this device.
				648	Auto sort control is not supported for this device.
				649	Patrol control is not supported for this device.
				650	Specified patrol number is not supported for this device.
				653	Time and date format is invalid.
				656	Failed to open file.
				657	Failed to write in file.
				658	The specified duration is invalid.
				659	The combination of parameter is invalid
				660	Specified Super Dynamic number is not supported for this device.
				663	SSL module is not exist
				701 : 704	Communication Error
				705	Failed to authentication of login.
				706	There is no specified URL.
				707	Access denied from device.
				708	Error occurred inside device.
				709	Error occurred with HTTP response or other.
				710	No response and timeout.
				711	Failed to HttpQueryInfo.
				712	Failed to InternetReadFile.
				901	Failed to allocate memory.
				902	Failed to create thread.
				903	Failed to create class.
				999	Exception Error

-D1 Category	A <sub>1</sub> Level	C <sub>1</sub> Source	M <sub>1</sub> M <sub>2</sub>	Z <sub>1</sub> Z <sub>2</sub> Z <sub>3</sub> Code	Error Description
-5: FTP	1:error 2:warning	1:PS-API error 2:communication error 3:device error	(internal use)	201	Parameter Error (Device Type)
				202	Parameter Error (Access Type)
				203	Parameter Error (Http URL)
				204	Parameter Error (Http Port Number)
				205	Parameter Error (Http TimeOut)
				206	Parameter Error (Proxy Name)
				207	Parameter Error (Proxy Port Number)
				208	Parameter Error (Retry Count)
				209	Parameter Error (Retry Time)
				210	Parameter Error (User Name)
				211	Parameter Error (Password)
				212	Parameter Error (Health Check)
				213	Parameter Error (UID)
				214	Parameter Error (UID OutPut Pointer Is NULL)
				215	Parameter Error (UID OutPut Size)
				216	Parameter Error (FTP Port Number)
				217	Parameter Error (FTP Port Number OutPut Pointer Is NULL)
				218	Parameter Error (Trans Mode)
				219	Parameter Error (Trans Mode OutPut Pointer Is NULL)
				220	Parameter Error (Device Type)

-D1 Category	A <sub>1</sub> Level	C <sub>1</sub> Source	M <sub>1</sub> M <sub>2</sub>	Z <sub>1</sub> Z <sub>2</sub> Z <sub>3</sub> Code	Error Description
-5: FTP	1:error 2:warning	1:PS-API error 2:communication error 3:device error	(internal use)	221	Parameter Error (FTP Status OutPut Pointer Is NULL)
				222	Parameter Error (FTP Transmission Rate OutPut Pointer Is NULL)
				223	Parameter Error (FTP Transmission Byte OutPut Pointer Is NULL)
				224	Parameter Error (Channel)
				225	Parameter Error (Start Date Time Pointer Is NULL)
				226	Parameter Error (End Date Time Pointer Is NULL)
				227	Parameter Error (Time)
				228	Parameter Error (Type)
				229	Parameter Error (File Name Pointer Is NULL)
				230	Parameter Error (Event Type)
				231	Parameter Error (File Name)
				232	Parameter Error (SD Event and the others cannot be specified together.)
				301	FTP Transmission State
				302	FTP Not Transmission State
				303	UID Setting State
				304	UID Not Setting State
				305	Network Parameter Not Setting State

-D1 Category	A <sub>1</sub> Level	C <sub>1</sub> Source	M <sub>1</sub> M <sub>2</sub>	Z <sub>1</sub> Z <sub>2</sub> Z <sub>3</sub> Code	Error Description
-5: FTP	1:error 2:warning	1:PS-API error 2:communication error 3:device error	(internal use)	601	Download Thread Stop Error
				701	InternetOpen Error
				702	InternetConnect Error
				703	HttpOpenRequest Error
				704	HttpSendRequest Error
				705	FtpOpenFile Error
				711	HttpQueryInfo Error
				712	InternetReadFile Error
				713	Get UID Error
				715	Data Damage Error
				720	No Audio File
				901	GetMessage Error
				902	Download Thread Create Error
				903	File Open Error
				904	File Write Error