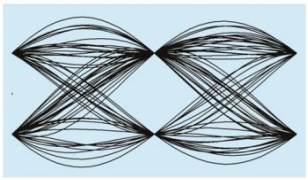

MaxEye Digital Audio and Video Signal Generation

DRM Signal Generation Toolkit

Version 1.0.0

Remote C API User Manual

Contents

1 Introduction

MaxEye Technologies provides generation API's in C for generating the standard complaint signals for various digital audio and video broadcasting standards. This guide explains Remote C structure and how to make use of the API's to control the DRM Signal Generation toolkit Soft Front Panel (SFP) remotely, to run programming examples by using NI hardware Vector Signal Generator (NI VSG), and Vector Signal Transceiver (NI VST).

The Digital Radio Mondiale (DRM) signal generation is based on the ETSI standard ES 201 980 (Digital Radio Mondiale (DRM); System Specification).

2 Installed File Location

The remote C API's documentation file is located in, C:\Program Files(x86)\MaxEye\Digital Video Toolkits\DRM Generation\Documentation.

(***Note***: - *For 32-bit Operating System*, C Documentation is located in C:\Program Files\MaxEye\Digital Video Toolkits\DRM Generation\Documentation).

The remote C Examples are located in, C:\Program Files(x86)\MaxEye\Digital Video Toolkits\DRM Generation\Examples\C.

(***Note***: - *For 32-bit Operating System*, remote C Examples are located in, C:\Program Files\MaxEye\Digital Video Toolkits\DRM Generation\Examples\C.)

The DRM Signal Generation Soft Front Panel (SFP) is located in, C:\Program Files(x86)\MaxEye\Digital Video Toolkits\DRM Generation\Application.

(***Note***: - *For 32-bit Operating System*, Soft Front Panel (SFP) is located in C:\Program Files\MaxEye\Digital Video Toolkits\DRM Generation\Application.)

You can also find a shortcut to the above location from the windows start menu. **Start->All Programs->MaxEye->Digital Radio Toolkits->DRM Generation**
Note: - *For Windows 10*, **Start ->MaxEye**

3 Remote C APIs

The Remote C API's allow user to configure and control the DRM Signal Generation Soft Front Panel (SFP) remotely through TCP connection. The SFP running in the signal generation hardware acts as a TCP Client and the test program running in the remote system built using the remote C API's acts as a TCP Server. MaxEye DRM Signal Generation Toolkit provides set of C API's to establish connection, configure parameters, initiating and stopping the signal generation and to read the output parameters. The DRM Signal Generation SFP operates in two modes, remote and local. To control the SFP from remote system the SFP should be in remote mode.

3.1 MaxEye DRM SG Remote TCP Open Connection

NAME MaxEye_DRM_SG_Remote_TCP_Open_Connection

DESCRIPTION Opens the TCP network connection between DRM SFP Client and Server application.

FUNCTION PROTOTYPE

```
void __cdecl MaxEye_DRM_SG_Remote_TCP_Open_Connection
(
        char            IPAddress[],
        uint16_t        PortNumber,
        int32_t         TimeoutMs,
        int32_t         ErrorCodeIn,
        LVRefNum        *ConnectionIDOut,
        int32_t         *ErrorCodeOut,
)
```

INPUT PARAMETERS

- ErrorCodeIn – Specifies the error code. The ErrorCodeIn can accept error information from previously called C API. Use this information to decide if any functionality should be bypassed in the event of errors from other C APIs.
- TimeoutMs – Specifies TCP Network connection timeout, in milliseconds, that the function waits to complete and return an error. The default value is 20s. A value of -1 indicates to wait indefinitely.
- PortNumber – Specifies the port number to establish network connection from server to client system. The default value is 7072.
- IPAddress – Specifies the IP Address or network name of the remote system.

OUTPUT PARAMETERS

- ConnectionIDOut – Returns the TCP connection reference. Connection ID Out is a network connection reference that uniquely identifies the TCP connection. Use this value to refer to this connection in subsequent API calls.
- ErrorCodeOut – Returns the error code**,** passes error or warning information out of an API to be used by other C APIs.

DEPENDENCIES

- Header – DRM Generation.h, extcode.h, fundtype.h, platdefines.h, lv_epilog.h
- Library – DRM Generation.lib
- DLL – DRM Generation.dll

3.2 MaxEye DRM SG Remote Set Number of Carriers

NAME MaxEye_DRM_SG_Remote_Set_Number_of_Carriers

DESCRIPTION Configures the number of carriers to the DRM Client SFP Application through TCP Network Connection. The DRM Signal Generation Toolkit supports generation of multiple DRM carriers. The maximum number of carriers supported is 15. In the real-time mode only one carrier is supported.

FUNCTION PROTOTYPE

```
void __cdecl MaxEye_DRM_SG_Remote_Set_Number_of_Carriers
(
        LVRefNum      *ConnectionIDIn,
        int32_t       NumberOfCarriers,
        int32_t       ErrorCodeIn,
        LVRefNum      *ConnectionIDOut
        int32_t       *ErrorCodeOut
)
```

INPUT PARAMETERS

- ConnectionIDIn – Specifies the TCP connection reference. Connection ID In is a network connection reference that uniquely identifies the TCP connection.
- NumberOfCarriers – Specifies the number of carriers needs to be generated. The default value is 1.
- ErrorCodeIn – Specifies the error code. The ErrorCodeIn can accept error information from previously called C API. Use this information to decide if any functionality should be bypassed in the event of errors from other C APIs.

OUTPUT PARAMETERS

- ConnectionIDOut – Returns the TCP connection reference. Connection ID Out is a network connection reference that uniquely identifies the TCP connection. Use this value to refer to this connection in subsequent function calls.
- ErrorCodeOut – Returns the error code**,** passes error or warning information out of an API to be used by other C APIs.

DEPENDENCIES

- Header – DRM Generation.h, extcode.h, fundtype.h, platdefines.h, lv_epilog.h
- Library – DRM Generation.lib
- DLL – DRM Generation.dll

3.3 MaxEye DRM SG Remote Set Number of Super Frames

NAME MaxEye_DRM_SG_Remote_Set_Number_of_Super_Frames

DESCRIPTION Configures the total number of Transmission Super Frames to the DRM Remote SFP Application through TCP Network Connection.

The generator uses the same frame configuration for all the frames and the payload is continuous across frames.

FUNCTION PROTOTYPE

```
void __cdecl MaxEye_DRM_SG_Remote_Set_Number_of_Super_Frames
(
        int32_t         ErrorCodeIn,
        uint32_t        NumberOfSuperFrames,
        LVRefNum        *ConnectionIDIn,
        int32_t         *ErrorCodeOut,
        LVRefNum        *ConnectionIDOut
)
```

INPUT PARAMETERS

- ConnectionIDIn – Specifies the TCP connection reference. Connection ID In is a network connection reference that uniquely identifies the TCP connection.
- NumberOfSuperFrames – Specifies the required number of Transmission Super Frames. This parameter defines the length of the waveform to be generated. To generate longer duration of the waveform, increase the Number of Tx Super Frames. The default value is 1.
- ErrorCodeIn – Specifies the error code. The ErrorCodeIn can accept error information from previously called C API. Use this information to decide if any functionality should be bypassed in the event of errors from other C APIs.

OUTPUT PARAMETERS

- ConnectionIDOut – Returns the TCP connection reference. Connection ID Out is a network connection reference that uniquely identifies the TCP connection. Use this value to refer to this connection in subsequent function calls.
- ErrorCodeOut – Returns the error code**,** passes error or warning information out of an API to be used by other C APIs.

DEPENDENCIES

- Header – DRM Generation.h, extcode.h, fundtype.h, platdefines.h, lv_epilog.h
- Library – DRM Generation.lib
- DLL – DRM Generation.dll

3.4 MaxEye DRM SG Remote Set Generation Mode

NAME MaxEye_DRM_SG_Remote_Set_Generation_Mode

DESCRIPTION Configures the Generation Mode to the DRM Remote SFP Application through TCP Network Connection.

FUNCTION PROTOTYPE

```
void __cdecl MaxEye_DRM_SG_Remote_Set_Generation_Mode
(
        LVRefNum        *ConnectionIDIn,
        uint16_t        GenerationMode,
```

```
                int32_t         ErrorCodeIn,
                LVRefNum        *ConnectionIDOut
                int32_t         *ErrorCodeOut,
)
```

INPUT PARAMETERS

- ConnectionIDIn – Specifies the TCP connection reference. Connection ID In is a network connection reference that uniquely identifies the TCP connection.
- GenerationMode – Specifies the generation mode of the MaxEye DRM Signal Generator. The default value is 2. The possible values are given below.
  - 0 – Generate and Play Waveform (Real Time)
  - 1 – Generate and Play Waveform
  - 2 – Generate and Save Waveform
  - 3 – Play Waveform from File
- ErrorCodeIn – Specifies the error code. The ErrorCodeIn can accept error information from previously called C API. Use this information to decide if any functionality should be bypassed in the event of errors from other C APIs.

OUTPUT PARAMETERS

- ConnectionIDOut – Returns the TCP connection reference. Connection ID Out is a network connection reference that uniquely identifies the TCP connection. Use this value to refer to this connection in subsequent function calls.
- ErrorCodeOut – Returns the error code**,** passes error or warning information out of an API to be used by other C APIs.

DEPENDENCIES

- Header – DRM Generation.h, extcode.h, fundtype.h, platdefines.h, lv_epilog.h
- Library – DRM Generation.lib
- DLL – DRM Generation.dll

3.5 MaxEye DRM SG Remote Set Hardware Settings

NAME MaxEye_DRM_SG_Remote_Set_Hardware_Settings

DESCRIPTION Configures the VSG/VST hardware settings to the DRM Remote SFP Application through TCP Network Connection

FUNCTION PROTOTYPE

```
void __cdecl MaxEye_DRM_SG_Remote_Set_Hardware_Settings
(
        LVRefNum        *ConnectionIDIn,
        DRM_Hardware    *HardwareSettings,
        int32_t         ErrorCodeIn,
        LVRefNum        *ConnectionIDOut,
        int32_t         *ErrorCodeOut
```

)

INPUT PARAMETERS

- ConnectionIDIn – Specifies the TCP connection reference. Connection ID In is a network connection reference that uniquely identifies the TCP connection.
- HardwareSettings – Specifies the Hardware Settings parameters

```
typedef struct
{
        LStrHandle      RFSGResource;
        double          PowerLevelDBm;
        double          ArbPreFilterGainDB;
        double          ExternalAttenuationDB;
        LStrHandle      RefClockSource;
        double          FrequencyHz;
        LStrHandle      ClockOutputTerminal;
} DRM_Hardware
```

- RFSGResource – Specifies the Resource Name. Select the name used in NI Measurement and Automation Explorer (NI MAX) for the NI PXIe-5672/5673/5673E or NI 5840 device.
- PowerLevelDBm – Specifies the Average Power level of the signal in dBm. The default value is -10.00dBm
- ExternalAttenuationDB – Specifies the external amplification or attenuation, if any, between the NI RF signal generator and the device under test. Positive values for this property represent amplification, and negative values for this property represent attenuation. The default value is 0.
- HaedroomDB – Needs to be added here
- ArbPreFilterGainDB – Specifies the AWG Pre-filter Gain. The pre-filter gain is applied to the waveform data before any other signal processing. Reduce this value to prevent overflow in the AWG interpolation filters. Other gains on the NI-RFSG device are automatically adjusted to compensate for non-unity AWG pre-filter gain. The default value is -1 dB
- RefClockSource – Specifies the source of the Reference Clock signal. The default value is 0. Given below are the possible values
  - 0 – OnboardClock
  - 1 – RefIn
  - 2 – PXI_CLK
  - 3 – ClkIn
- FrequencyHz – Specifies the Reference Clock rate, in Hertz (Hz). The default value is 10MHz.
- outputTerminal – Specifies the terminal where the signal will be exported. The default value is 0. Given below are the possible values
  - 0 – Do not export signal
  - 1 – RefOut
  - 2 – RefOut2
  - 3 – ClkOut
  - 4 – PFI0

- 5 – PFI1
- 6 – PFI4
- 7 – PFI5
- 8 – PXI_Trig0
- 9 – PXI_Trig1
- 10 – PXI_Trig2

For more information about this, please refer NI RFSG Signal Generators help file.

- ErrorCodeIn – Specifies the error code. The ErrorCodeIn can accept error information from previously called C API. Use this information to decide if any functionality should be bypassed in the event of errors from other C APIs.

OUTPUT PARAMETERS

- ConnectionIDOut – Returns the TCP connection reference. Connection ID Out is a network connection reference that uniquely identifies the TCP connection. Use this value to refer to this connection in subsequent function calls.
- ErrorCodeOut – Returns the error code**,** passes error or warning information out of an API to be used by other C APIs.

DEPENDENCIES

- Header – DRM Generation.h, extcode.h, fundtype.h, platdefines.h, lv_epilog.h
- Library – DRM Generation.lib, labview.lib
- DLL – DRM Generation.dll

3.6 MaxEye DRM SG Remote Set Output Waveform Settings

NAME MaxEye_DRM_SG_Remote_Set_Output_Waveform_Settings

DESCRIPTION Configures the DRM Output Waveform Settings to the DRM Remote SFP Application through TCP Network Connection

FUNCTION PROTOTYPE

```
void __cdecl MaxEye_DRM_SG_Remote_Set_Output_Waveform_Settings
(
        LVRefNum      *ConnectionIDIn
        uint16_t      SampleWidth,
        double        HeadroomDB,
        uint16_t      OversamplingEnabled,
        double        OutputSamplingRate,
        int32_t       ErrorCodeIn,
        LVRefNum      *ConnectionIDOut,
        int32_t       *ErrorCodeOut
)
```

INPUT PARAMETERS

- ConnectionIDIn – Specifies the TCP connection reference. Connection ID In is a network connection reference that uniquely identifies the TCP connection.
- SampleWidth – Specifies the sample width to be used to generate waveform file. The default value is 1(16 bit). MaxEye recommend using 16-bits sample width for better signal quality of the generated waveform. Given below are the possible values

➢ 0 – 8 bit
➢ 1 – 16 bit

- HeadroomDB – Specifies the Headroom value. The generator uses this value for scaling the waveform. If PAPR of the signal is higher than the Headroom value then the generator clips the signal. To avoid clipping, the Headroom value should be higher than the PAPR of the signal. The default value is 12 dB.
- OverSamplingEnabled – Specifies whether the Oversampling Property is enabled or not. If this property is set to True then the generator resamples the generated signal based on the value configured by the user for the Output Sampling Rate property.The default value is 0 (False). Given below are the possible values

➢ 0 – False
➢ 1 – True

- OutputSamplingRate – Specifies the Output Sampling Rate. The generator resamples the generated signal to a sampling rate equal to the Output Sampling Rate only if the Over Sampling Enabled property is set to True.
- ErrorCodeIn – Specifies the error code. The ErrorCodeIn can accept error information from previously called C API. Use this information to decide if any functionality should be bypassed in the event of errors from other C APIs.

OUTPUT PARAMETERS

- ConnectionIDOut – Returns the TCP connection reference. Connection ID Out is a network connection reference that uniquely identifies the TCP connection. Use this value to refer to this connection in subsequent function calls.
- ErrorCodeOut – Returns the error code**,** passes error or warning information out of an API to be used by other C APIs.

DEPENDENCIES

- Header – DRM Generation.h, extcode.h, fundtype.h, platdefines.h, lv_epilog.h
- Library – DRM Generation.lib, labview.lib
- DLL – DRM Generation.dll

3.7 MaxEye DRM SG Remote Set Carrier Configuration

NAME MaxEye_DRM_SG_Remote_Set_Carrier_Configuration

DESCRIPTION

Configures the Carrier Frequency and Signal Bandwidth for each carrier based on carrier index value to the DRM Remote SFP Application through TCP Network Connection

FUNCTION PROTOTYPE

```
void __cdecl MaxEye_DRM_SG_Remote_Set_Carrier_Configuration
(
        LVRefNum        *ConnectionIDIn
        int32_t         CarrierIndex,
        double          CarrierFrequencyHz,
        double          SignalBandwidthHz,
        int32_t         ErrorCodeIn,
        LVRefNum        *ConnectionIDOut,
        int32_t         *ErrorCodeOut
)
```

INPUT PARAMETERS

- ConnectionIDIn – Specifies the TCP connection reference. Connection ID In is a network connection reference that uniquely identifies the TCP connection.
- CarrierIndex – Specifies the index value of the selected carrier. The default value of the Carrier Index is 0 which corresponds to the first carrier. For generating multi carrier signal, configure the parameters for each carrier index.
- CarrierFrequencyHz – Specifies the Carrier Frequency for the selected carrier in Hz.
- SignalBandwidth – Specifies signal bandwidth of the selected carrier, in Hz. Configure the Bandwidth of the signal for the selected carrier. The generator internally uses the Carrier Frequency and Bandwidth property values internally to compute the overall bandwidth and sampling rate of the signal when more than one carrier is used.
- ErrorCodeIn – Specifies the error code. The ErrorCodeIn can accept error information from previously called C API. Use this information to decide if any functionality should be bypassed in the event of errors from other C APIs.

OUTPUT PARAMETERS

- ConnectionIDOut – Returns the TCP connection reference. Connection ID Out is a network connection reference that uniquely identifies the TCP connection. Use this value to refer to this connection in subsequent function calls.
- ErrorCodeOut – Returns the error code**,** passes error or warning information out of an API to be used by other C APIs.

DEPENDENCIES

- Header – DRM Generation.h, extcode.h, fundtype.h, platdefines.h, lv_epilog.h
- Library – DRM Generation.lib
- DLL – DRM Generation.dll

3.8 MaxEye DRM SG Remote Set Waveform Settings

NAME MaxEye_DRM_SG_Remote_Set_Waveform_Settings

DESCRIPTION Configures the DRM Waveform Settings for each carrier based on carrier index to the DRM Remote SFP Application through TCP Network Connection

FUNCTION PROTOTYPE

```
void __cdecl MaxEye_DRM_SG_Remote_Set_Waveform_Settings
(
        LVRefNum            *ConnectionIDIn,
        int32_t             CarrierIndex,
        DRM_Configuration   *DRMConfiguration,
        int32_t             ServiceConfigurationSize,
        int32_t             ErrorCodeIn,
        LVRefNum            *ConnectionIDOut,
        int32_t             *ErrorCodeOut
)
```

INPUT PARAMETERS

- ConnectionIDIn – Specifies the TCP connection reference. Connection ID In is a network connection reference that uniquely identifies the TCP connection.
- CarrierIndex – Specifies the index value of the selected carrier. The default value of the Carrier Index is 0 which corresponds to the first carrier. For generating multi carrier signal, configure the parameters for each carrier index.
- DRMConfiguration – Specifies the DRM Signal Configuration parameters

```
typedef struct
{
        uint16_t RobustnessMode;
        uint16_t SpectrumOccupancy;
        uint16_t ErrorProtectionType;
        uint16_t InterleaverDepth;
        uint16_t NumberOfServices;
        uint16_t ModulationScheme;
        uint16_t SDCModulationScheme;
        uint16_t MappingScheme;
        uint16_t HPPProtectionLevel;
        uint16_t LPPProtectionLevel;
        uint16_t VSPPProtectionLevel;
        uint16_t ProtectionLevel;
        uint16_t BaseEnhancementFlag;
} DRM_Configuration
```

- RobustnessMode – Specifies the Robustness Mode of the DRM signal. Supported modes are A, B, C, D and E. For the signal in a given Spectrum Occupancy, the different robustness mode along with the Spectrum Occupancy decides the data rate. The default value is 0 (Mode A). Given below are the possible values
  - 0 – Mode A

- 1 – Mode B
- 2 – Mode C
- 3 – Mode D
- 4 – Mode E

➢ SpectrumOccupancy – Specifies the spectrum occupancy as per the requirement. The default value is 5 (20 kHz). Given below are the possible values

- 0 – 4.5 kHz
- 1 – 5 kHz
- 2 – 9 kHz
- 3 – 10 kHz
- 4 – 18 kHz
- 5 – 20 Khz

➢ ErrorProtectionType – Specifies the Error Protection type of the DRM Signal. Equal error protection uses a single code rate to protect all the data in a channel. Unequal error protection can be used with two code rates to allow the data in the Main Service Channel to be assigned to the higher protected part and the lower protected part. The default value is 0 (Equal Error Protection). Given below are the possible values

- 0 **–** Equal Error Protection
- 1 – Unequal Error Protection

➢ InterleaverDepth – Specifies the Interleaver Depth. For robustness modes A, B, C and D user can choose short or long interleaving depth. For robustness mode E only Long Interleaving depth is supported by the standard. The default value is 1 (Short Interleaving). Given below are the possible values

- 0 – Long Interleaving
- 1 – Short Interleaving

➢ NumberOfServices – Specifies the required number of services. Maximum 4 services can be configured per carrier.

➢ ModulationScheme Specifies the Modulation Scheme for Main Service Channel (MSC) of the DRM signal. Supporting schemes are 64-QAM, 16-QAM and 4-QAM. The default value is 0 (64-QAM). Given below are the possible values

- 0 – 64-QAM
- 1 – 16-QAM
- 2 – 4-QAM

➢ ModulationScheme2 – Specifies the Modulation Scheme for Service Description Channel (SDC) of the DRM Signal. Supporting schemes are 64-QAM, 16-QAM and 4-QAM. The default value is 1 (16-QAM). Given below are the possible values

- 1 – 16-QAM
- 2 – 4-QAM

- MappingScheme – Specifies the mapping scheme for MSC. For the standard mapping and symmetrical hierarchical modulation (SM and HMsym), identical mappings shall be used for the real and imaginary components of the signal constellation. For the mixed mapping hierarchical modulation (HMmix) separate mappings shall be used for the real and imaginary components of the signal constellation. Hierarchical Mapping Scheme works only with 64-QAM Modulation Scheme. The default value is 0 (SM). Given below are the possible values
  - 0 – SM
  - 1 – Hmsym
  - 2 - HMmix
- HPPProtectionLevel – Specifies the protection level for the higher protected part of the MSC. This parameter is applicable only if the Error Protection Type is set to Unequal Error Protection. The default value is 0. Given below are the possible values
  - 0 (highest protection)
  - 1
  - 2
  - 3 (weakest protection)
- LPPProtectionLevel – Specifies the protection level for the lower protected part of the MSC. The default value is 3. Given below are the possible values
  - 0 (highest protection)
  - 1
  - 2
  - 3 (weakest protection)
- VSPPProtectionLevel – Specifies the protection level for the very strong protected part of the MSC. It specifies the code rate of the very strongly protected part. This parameter is applicable only if the Error Protection Type is set to Unequal Error Protection and the Mapping Scheme is set to any of the Hierarchical Mapping. The default value is 0. Given below are the possible values
  - 0 (highest protection)
  - 1
  - 2
  - 3 (weakest protection)
- ProtectionLevel – Specifies the protection level for SDC. The default value is 0. Choose 0 for 16-QAM SDC Modulation Scheme and 1 for 4-QAM SDC Modulation Scheme. Given below are the possible values
  - 0 (if SDC Modulation Scheme is 16-QAM)
  - 1 (if SDC Modulation Scheme is 4-QAM)
- BaseEnhancementFlag – Specifies whether the transmission is the Base or Enhancement layer. The default value is 0. Given below are the possible values

0: Base layer - decodable by all DRM receivers.
1:Enhancement layer - only decodable by receivers with enhancement layer capabilities

- ServiceConfigurationSize – Specifies the actual number of services including Audio & Data.
- ErrorCodeIn – Specifies the error code. The ErrorCodeIn can accept error information from previously called C API. Use this information to decide if any functionality should be bypassed in the event of errors from other C APIs.

OUTPUT PARAMETERS

- ConnectionIDOut – Returns the TCP connection reference. Connection ID Out is a network connection reference that uniquely identifies the TCP connection. Use this value to refer to this connection in subsequent function calls.
- ErrorCodeOut – Returns the error code**,** passes error or warning information out of an API to be used by other C APIs.

DEPENDENCIES

- Header – DRM Generation.h, extcode.h, fundtype.h, platdefines.h, lv_epilog.h
- Library – DRM Generation.lib
- DLL – DRM Generation.dll

3.9 MaxEye DRM SG Remote Set Service Configuration

NAME MaxEye_DRM_SG_Remote_Set_ServiceConfiguration

DESCRIPTION Configures the Service parameters for the each carrier to the DRM Remote SFP Application through TCP Network Connection

FUNCTION PROTOTYPE

```
void __cdecl MaxEye_DRM_SG_Remote_Set_Waveform_Settings
(
        LVRefNum                    *ConnectionIDIn,
        int32_t                     CarrierIndex,
        int32_t                     ServiceIndex,
        DRM_Service_Configuration   *ServiceConfiguration,
        int32_t                     ErrorCodeIn,
        LVRefNum                    *ConnectionIDOut,
        int32_t                     *ErrorCodeOut
)
```

INPUT PARAMETERS

- ConnectionIDIn – Specifies the TCP connection reference. Connection ID In is a network connection reference that uniquely identifies the TCP connection.
- CarrierIndex – Specifies the index value of the selected carrier. The default value of the Carrier Index is 0 which corresponds to the first carrier. For generating multi carrier signal, configure the parameters for each carrier index.

- ServiceIndex – Specifies the index value of the selected service. The default value of the Service Index is 0 which corresponds to the first service. For generating multiple services, configure the parameters for each service index. Maximum 4 services can be configured per carrier.
- ServiceConfiguration – Specifies the Service Configuration parameters

```
typedef struct
{
	double		StreamID;
	double		ShortID;
	uint16_t	InputMode;
	LStrHandle	ServiceIdentifier;
	LStrHandle	ServiceLabel;
	uint16_t	Language;
	uint16_t	ProgrammeType;
	uint16_t	DisplayTextMessage;
	LStrHandle	TextMessage;
	uint16_t	AudioCAIndication;
	uint16_t	DataCAIndication;
} DRM_Service_Configuration
```

- StreamId – Specifies the stream Id of the stream which carries the data service (or data application). The valid values for stream Id is ranging from 0 to 4. Each Service must be given different Stream Ids.
- ShortId – Specifies the short identifier assigned to each service and used as a reference in the SDC. The valid values for short Id is ranging from 0 to 4. Each Audio Service must be given different Short Ids. If any Data Service is intended to be associated with any Audio Service, both the services shall be given same Short Id. Such associated Data Services need not to be accounted in the Number of Services parameter.
- InputMode – Specifies whether the intended service is of Audio type or Data type. The default value is 0 (Audio). Given below are the possible values
  - 0 – Audio
  - 1 – Data
- ServiceIdentifier – Specifies the unique identifier for each service. It is recommended that domestic services construct the 24-bit Service Identifier by allocating the 8-most significant bits to Extended Country Code (ECC), the next 4-bits to Country Code (CC), and the remaining 12 bits for individual services. Eg: - *f25236* where *f2* is ECC for India, *5* is the CC and *235* is random service id.
- ServiceLabel – Specifies label for a particular service.
- Language – Specifies the language of the target audience. The default value is 5 (English). Given below are the possible values

Decimal number	Language	Decimal number	Language
0	No language specified	8	Hindi
1	Arabic	9	Japanese
2	Bengali	10	Javanese
3	Chinese (Mandarin)	11	Korean
4	Dutch	12	Portuguese
5	English	13	Russian
6	French	14	Spanish
7	German	15	Other language

- ProgrammeType – Specifies the program type of an audio service. The default value is 5 (Education). Given below are the possible values

Decimal number	Programme type	Decimal number	Programme type
0	No programme type	16	Weather/meteorology
1	News	17	Finance/Business
2	Current Affairs	18	Children's programmes
3	Information	19	Social Affairs
4	Sport	20	Religion
5	Education	21	Phone In
6	Drama	22	Travel
7	Culture	23	Leisure
8	Science	24	Jazz Music
9	Varied	25	Country Music
10	Pop Music	26	National Music
11	Rock Music	27	Oldies Music
12	Easy Listening Music	28	Folk Music
13	Light Classical	29	Documentary
14	Serious Classical	30	*Not used*
15	Other Music	31	*Not used - skip indicator*

- DisplayTextMessage – Specifies whether to display text message or not in the receiver.
- TextMessage – Specifies the text message to be transmitted. Maximum 1024 characters are supported.
- DataConditionalAccessIndication – Specifies whether the data service uses data conditional access or not. The default value is 0 (False). Given below are the possible values
  - 0 – False
  - 1 – True
- AudioConditionalAccessIndication – Specifies whether the audio service uses audio conditional access or not. The default value is 0 (False). Given below are the possible values
  - 0 – False
  - 1 – True

- ErrorCodeIn – Specifies the error code. The ErrorCodeIn can accept error information from previously called C API. Use this information to decide if any functionality should be bypassed in the event of errors from other C APIs.

OUTPUT PARAMETERS

- ConnectionIDOut – Returns the TCP connection reference. Connection ID Out is a network connection reference that uniquely identifies the TCP connection. Use this value to refer to this connection in subsequent function calls.
- ErrorCodeOut – Returns the error code**,** passes error or warning information out of an API to be used by other C APIs.

DEPENDENCIES

- Header – DRM Generation.h, extcode.h, fundtype.h, platdefines.h, lv_epilog.h
- Library – DRM Generation.lib, labview.lib
- DLL – DRM Generation.dll

3.10 MaxEye DRM SG Remote Set Audio Payload Configuration

NAME MaxEye_DRM_SG_Remote_Set_AudioPayload_Configuration

DESCRIPTION Configures the Audio Payload properties for each carrier to the Client DRM SFP Application through TCP Network Connection

FUNCTION PROTOTYPE

```
void __cdecl MaxEye_DRM_SG_Remote_Set_AudioPayload_Configuration
(
        LVRefNum                    *ConnectionIDIn,
        int32_t                     CarrierIndex,
        int32_t                     ServiceIndex,
        DRM_Payload_Configuration   *DigitalAudioPayloadControl,
        int32_t                     NumberOfImages,
        int32_t                     ErrorCodeIn,
        LVRefNum                    *ConnectionIDOut,
        int32_t                     *ErrorCodeOut
)
```

INPUT PARAMETERS

- ConnectionIDIn – Specifies the TCP connection reference. Connection ID In is a network connection reference that uniquely identifies the TCP connection.
- NumberOfImages – Specifies the number of images added for the SlideShow data feature.
- CarrierIndex – Specifies the index value of the selected carrier. The default value of the Carrier Index is 0 which corresponds to the first carrier. For generating multi carrier signal, configure the parameters for each carrier index.
- ServiceIndex – Specifies the index value of the selected service. The default value of the Service Index is 0 which corresponds to the first service. For generating multiple services,

configure the parameters for each service index. Maximum 4 services can be configured per carrier.

- DigitalAudioPayloadControl – Specifies the Payload Configuration parameters

```
typedef struct
{
	uint16_t	PayloadMode;
	uint32_t	PayloadPNOrder;
	uint32_t	PayloadPNSeed;
	LStrHandle	PayloadUserDefinedBits;
	uint16_t	PayloadTestPattern;
	LStrHandle	AudioFilePath;
	int32_t		NumOfBytesInHPP;
}DRM_Payload_Configuration
```

➢ PayloadTestPattern – Specifies the bit pattern for payload generation. This mode is used for generating signal with known test patterns. The default value is 0. Given below are the possible values

- 0 – All 1s
- 1 – All 0s
- 2 – 10101010
- 3 – 11110000

➢ PayloadMode – Specifies the desired payload source. The default value is 0. Given below are the possible values

- 0 – PN Sequence

Specifies **Payload PN Order** and **Payload PN Seed** properties. The toolkit generates pseudo random sequence based on the **PN Order** and **Seed value.** The generated bit sequence is used as a payload for generating the signal. Use this mode for testing the receiver performance for random payload values. When the number of super frames is more than 1 then the toolkit maintains payload continuity across the super frames.

- 1 – User Defined Bits

Specifies desired bit pattern in the **Payload User Defined Bits** property. The generator repeats the entered bit pattern till the number of bits required for the frame, for the given configuration, is met

- 2 – Test Pattern

Specifies the test bit pattern. This mode is used for generating signal with known test patterns.

- 3 – Test File

Specifies the path of the test file. This mode is used for generating signal with the binary data from the file.

- 4 – Input Wave File (*.wav)

Specifies the path of the audio file. The inbuilt AAC encoder compresses the wave audio to AAC format compatible with DRM. MaxEye recommends wav file of type mono with sampling frequency 24 kHz.

- 5 – DRM Modulated AAC

Specifies the path of the DRM Modulated AAC file. In this mode configure the Audio File Path property and Num. of bytes in HPP (applicable only if Error Protection Type is Unequal Error Protection) and the toolkit ignores other properties in the Digital Audio Payload Control. The input AAC audio should be encoded as per DRM standard

- AudioFilePath – Specifies the file path of wav audio/ file path of test file/ file path of DRM specific AAC file when the payload mode is Input Wave File(*.wav)/ Test File/ DRM Modulated AAC.
- NumOfBytesInHPP – Specifies the number of bytes to be in higher protected part when the Error Protection Type is Unequal Error Protection.
- PayloadPNSeed – Specifies the initial seed for the PN generator
- PayloadPNOrder – Specifies the order for the PN generator
- PayloadUserDefinedBits – Specifies the custom bit pattern when the payload mode is User Defined Bits.

- ErrorCodeIn **–** Specifies the error code. The ErrorCodeIn can accept error information from previously called C API. Use this information to decide if any functionality should be bypassed in the event of errors from other C APIs.

OUTPUT PARAMETERS

- ConnectionIDOut **–** Returns the TCP connection reference. Connection ID Out is a network connection reference that uniquely identifies the TCP connection. Use this value to refer to this connection in subsequent function calls.
- ErrorCodeOut **–** Returns the error code**,** passes error or warning information out of an API to be used by other C APIs.

DEPENDENCIES

- Header **–** DRM Generation.h, extcode.h, fundtype.h, platdefines.h, lv_epilog.h
- Library **–** DRM Generation.lib, labview.lib
- DLL **–** DRM Generation.dll

3.11 MaxEye DRM SG Remote Set Image File Path

NAME MaxEye_DRM_SG_Remote_Set_ImageFilePath

DESCRIPTION

Configures the file paths of the images added to the SlideShow data feature for each carrier to the Client DRM SFP Application through TCP Network Connection

FUNCTION PROTOTYPE

```
void __cdecl MaxEye_DRM_SG_Remote_Set_ImageFilePath
(
        LVRefNum      *ConnectionIDIn,
        int32_t       CarrierIndex,
        int32_t       ServiceIndex,
        int32_t       ImagePathIndex,
        char          ImageFilePath[],
        int32_t       ErrorCodeIn,
        LVRefNum      *ConnectionIDOut,
        int32_t       *ErrorCodeOut
)
```

INPUT PARAMETERS

- ConnectionIDIn – Specifies the TCP connection reference. Connection ID In is a network connection reference that uniquely identifies the TCP connection.
- CarrierIndex – Specifies the index value of the selected carrier. The default value of the Carrier Index is 0 which corresponds to the first carrier. For generating multi carrier signal, configure the parameters for each carrier index.
- ServiceIndex – Specifies the index value of the selected service. The default value of the Service Index is 0 which corresponds to the first service. For generating multiple services, configure the parameters for each service index. Maximum 4 services can be configured per carrier.
- ImagePathIndex – Specifies the index value to keep a track on the current image under processing.
- ImageFilePath – Specifies the paths of the images intended to add to the SlideShow feature in array format. Each array element shall be one image path.
- ErrorCodeIn – Specifies the error code. The ErrorCodeIn can accept error information from previously called C API. Use this information to decide if any functionality should be bypassed in the event of errors from other C APIs.

OUTPUT PARAMETERS

- ConnectionIDOut – Returns the TCP connection reference. Connection ID Out is a network connection reference that uniquely identifies the TCP connection. Use this value to refer to this connection in subsequent function calls.
- ErrorCodeOut – Returns the error code**,** passes error or warning information out of an API to be used by other C APIs.

DEPENDENCIES

- Header – DRM Generation.h, extcode.h, fundtype.h, platdefines.h, lv_epilog.h

- Library – DRM Generation.lib
- DLL – DRM Generation.dll

3.12 MaxEye DRM SG Remote Set Output Waveform File Path

NAME MaxEye_DRM_SG_Remote_Set_Ouptut_Waveform_File_Path

DESCRIPTION Configures the path to save the generated waveform to the Client DRM SFP Application through TCP Network Connection

FUNCTION PROTOTYPE

```
void __cdecl MaxEye_DRM_SG_Remote_Set_Output_Waveform_File_Path
(
        LVRefNum      *ConnectionIDIn,
        char          WavfeormFilePath[],
        int32_t       ErrorCodeIn,
        LVRefNum      *ConnectionIDOut,
        int32_t       *ErrorCodeOut
)
```

INPUT PARAMETERS

- ConnectionIDIn – Specifies the TCP connection reference. Connection ID In is a network connection reference that uniquely identifies the TCP connection.
- WavfeormFilePath – Specifies the file location where the generated IQ baseband waveform is stored.
- ErrorCodeIn – Specifies the error code. The ErrorCodeIn can accept error information from previously called C API. Use this information to decide if any functionality should be bypassed in the event of errors from other C APIs.

OUTPUT PARAMETERS

- ConnectionIDOut – Returns the TCP connection reference. Connection ID Out is a network connection reference that uniquely identifies the TCP connection. Use this value to refer to this connection in subsequent function calls.
- ErrorCodeOut – Returns the error code**,** passes error or warning information out of an API to be used by other C APIs.

DEPENDENCIES

- Header – DRM Generation.h, extcode.h, fundtype.h, platdefines.h, lv_epilog.h
- Library – DRM Generation.lib
- DLL – DRM Generation.dll

3.13 MaxEye DRM SG Remote Set Play Waveform Settings

NAME MaxEye_DRM_SG_Remote_Set_Play_Waveform_Settings

DESCRIPTION Configures the DRM Play Waveform from File Settings to the Client DRM SFP Application through TCP Network Connection

FUNCTION PROTOTYPE

```
void __cdecl MaxEye_DRM_SG_Remote_Set_Play_Waveform_Settings
(
        LVRefNum        *ConnectionIDIn,
        double          CenterFrequency,
        int32_t         WriteBlockSize,
        uint16_t        SampleWidth,
        char            WaveformFilePath[],
        int32_t         ErrorCodeIn,
        LVRefNum        *ConnectionIDOut,
        int32_t         *ErrorCodeOut
)
```

INPUT PARAMETERS

- ConnectionIDIn – Specifies the TCP connection reference. Connection ID In is a network connection reference that uniquely identifies the TCP connection.
- CenterFrequency – Specifies the center frequency of the DRM signal in Hz.
- WriteBlockSizeSamples – Specifies the size of the block in samples. The waveform is written in the hardware as blocks.
- SampleWidth – Specifies the sample width value. Use the same sample width value used for saving the waveform in the file.
- WaveformFilePath – Specifies the waveform file path to play the waveform.
- ErrorCodeIn – Specifies the error code. The ErrorCodeIn can accept error information from previously called C API. Use this information to decide if any functionality should be bypassed in the event of errors from other C APIs.

OUTPUT PARAMETERS

- ConnectionIDOut – Returns the TCP connection reference. Connection ID Out is a network connection reference that uniquely identifies the TCP connection. Use this value to refer to this connection in subsequent function calls.
- ErrorCodeOut – Returns the error code**,** passes error or warning information out of an API to be used by other C APIs.

DEPENDENCIES

- Header – DRM Generation.h, extcode.h, fundtype.h, platdefines.h, lv_epilog.h
- Library – DRM Generation.lib
- DLL – DRM Generation.dll

3.14 MaxEye DRM SG Remote Set Impairments

NAME MaxEye_DRM_SG_Remote_Set_Impairments

DESCRIPTION Configures the Impairment properties for each carrier to the Client DRM SFP Application through TCP Network Connection

FUNCTION PROTOTYPE

```
void __cdecl MaxEye_DRM_SG_Remote_Set_Impairments
(
        LVRefNum        *ConnectionIDIn,
        int32_t         CarrierIndex
        Impairment      *Impairments,
        int32_t         ErrorCodeIn,
        LVRefNum        *ConnectionIDOut,
        int32_t         *ErrorCodeOut
)
```

INPUT PARAMETERS

- ConnectionIDIn – Specifies the TCP connection reference. Connection ID In is a network connection reference that uniquely identifies the TCP connection.
- CarrierIndex – Specifies the index value of the selected carrier. The default value of the Carrier Index is 0 which corresponds to the first carrier. For generating multi carrier signal, configure the parameters for each carrier index.
- Impairments – Specifies the Impairment Configuration parameters

```
typedef struct
{
        uint16_t            ImpairmentsEnabled;
        uint16_t            AWGNEnabled;
        double              CarrierToNoiseRatioDB;
        double              FrequencyOffsetHz;
        double              ClockOffsetPPM;
        IQ_Impairments      IQImpairments;
} Impairment
```

- ImpairmentsEnabled – Specifies whether the impairment addition is enabled or not. If this property is set to True then the toolkit adds the impairments to the generated signal as per the user configuration for the supported impairments. The default value is 0 (False). Given below are the possible values
  - 0 – False
  - 1 – True
- AWGNEnabled – Specifies if the AWGN noise addition is enabled or not. If this property is set to True then the toolkit adds Additive White Gaussian Noise (AWGN) to the created waveform based on the value configured in the Carrier to Noise Ratio property. The default value is 0 (False). Given below are the possible values
  - 0 – False
  - 1 – True
- CarrierToNoiseRatio – Specifies the Carrier to Noise ratio of the generated signal. The default value is 0 dB.

- FrequencyOffsetHz – Specifies the frequency offset in Hz. The toolkit applies frequency offset to the created waveform based on the value configured in this property. The applied frequency offset is relative to the signal generator's carrier frequency. The default value is 0.
- ClockOffsetPPM – Specifies the clock offset in parts per million (ppm). The toolkit applies the clock offset to the generated waveform based on this value. The applied clock offset is relative to the clock frequency of the signal generator. The default value is 0.
- IQImpairments – Specifies IQ Impairment Configuration parameters

```
typedef struct
{
	double IDCOffset;
	double QDCOffset;
	double IQGainImbalanceDB;
	double QuadratureSkewDeg;
} IQ_Impairments
```

- IDCOffset – Specifies the In-phase DC offset value. The toolkit adds the DC offset to the in-phase signal component (I) of the complex waveform as a percentage of the root mean square magnitude of the unaltered I signal. The default value is 0.
- QDCOffset – Specifies the Quadrature DC offset value. The toolkit adds the DC offset to the quadrature-phase signal component (Q) of the complex waveform as a percentage of the root mean square magnitude of the unaltered Q signal. The default value is 0.
- IQGainImbalanceDB – Specifies the ratio, in dB, of the mean amplitude of the in-phase (I) signal to the mean amplitude of the quadrature-phase (Q) signal. The default value is 0.
- QuadratureSkewDeg – Specifies the deviation in angle from 90 degrees between the in-phase (I) and quadrature-phase (Q) signals. The default value for the Quadrature Skew is 0.

- ErrorCodeIn **–** Specifies the error code. The ErrorCodeIn can accept error information from previously called C API. Use this information to decide if any functionality should be bypassed in the event of errors from other C APIs.

OUTPUT PARAMETERS

- ConnectionIDOut **–** Returns the TCP connection reference. Connection ID Out is a network connection reference that uniquely identifies the TCP connection. Use this value to refer to this connection in subsequent function calls.
- ErrorCodeOut **–** Returns the error code**,** passes error or warning information out of an API to be used by other C APIs.

DEPENDENCIES

- Header **–** DRM Generation.h, extcode.h, fundtype.h, platdefines.h, lv_epilog.h
- Library **–** DRM Generation.lib
- DLL **–** DRM Generation.dll

3.15 MaxEye DRM SG Remote Command

NAME MaxEye_DRM_SG_Remote_Command

DESCRIPTION Initiates or stops the signal generation

FUNCTION PROTOTYPE

```
void __cdecl MaxEye_DRM_SG_Remote_Command
(
        LVRefNum        *ConnectionIDIn,
        uint16_t        Command,
        int32_t         ErrorCodeIn,
        LVRefNum        *ConnectionIDOut,
        int32_t         *ErrorCodeOut
)
```

INPUT PARAMETERS

- ConnectionIDIn – Specifies the TCP connection reference. Connection ID In is a network connection reference that uniquely identifies the TCP connection.
- Command – Specifies whether to initiate or stop signal generation. The default value is 0.
  - 0 – Start Generation
  - 1 – Stop Generation
  - 2 –Save Configuration
  - 3 –Load Configuration
- ErrorCodeIn – Specifies the error code. The ErrorCodeIn can accept error information from previously called C API. Use this information to decide if any functionality should be bypassed in the event of errors from other C APIs.

OUTPUT PARAMETERS

- ConnectionIDOut – Returns the TCP connection reference. Connection ID Out is a network connection reference that uniquely identifies the TCP connection. Use this value to refer to this connection in subsequent function calls.
- ErrorCodeOut – Returns the error code**,** passes error or warning information out of an API to be used by other C APIs.

DEPENDENCIES

- Header – DRM Generation.h, extcode.h, fundtype.h, platdefines.h, lv_epilog.h
- Library – DRM Generation.lib
- DLL – DRM Generation.dll

3.16 MaxEye DRM SG Remote Get Error Status

NAME MaxEye_DRM_SG_Remote_Get_Error_Status

DESCRIPTION Receives Error Message from the Remote SFP Application through TCP Network Connection

FUNCTION PROTOTYPE

```
void __cdecl MaxEye_DRM_SG_Remote_Get_Error_Status
(
        LVRefNum        *ConnectionIDIn,
        int32_t         ErrorCodeIn,
        LVRefNum        *ConnectionIDOut,
        DRM_Get_Generation_Parameters *GetGenerationParameters,
        int32_t         lengthOfError,
        char            ErrorStatus[],
        int32_t         *ErrorCodeOut
)
```

INPUT PARAMETERS

- ConnectionIDIn – Specifies the TCP connection reference. Connection ID In is a network connection reference that uniquely identifies the TCP connection.
- Length Of Error – Specifies the size of the ErrorStatus.
- ErrorCodeIn – Specifies the error code. The ErrorCodeIn can accept error information from previously called C API. Use this information to decide if any functionality should be bypassed in the event of errors from other C APIs.

OUTPUT PARAMETERS

- ConnectionIDOut – Returns the TCP connection reference. Connection ID Out is a network connection reference that uniquely identifies the TCP connection. Use this value to refer to this connection in subsequent function calls.
- Get Generation Parameters- specifies the following parameters

```
typedef struct
 {
    int32_t GenerationMode;
    double GeneratingFrames;
    double MaximumNumberOfFrames;
    LVBoolean GenerationStatus;
    LVBoolean GenerationCompleted;
    int32_t ErrorOrWarning;
} DRM_Get_Generation_Parameters;
```

- GenerationStatus – Returns the status of the generator.
- Generation Mode – Returns the generation mode.
- Generating frames– Returns the generated frames.
- Maximum number of frames– Returns the maximum number of frames to be generated.
- Error or warning – Specifies Error or warning occurred at the generator.
- GeneratingFrames – Returns the current frame number being generated, to the user.
- ErrorStatus – Returns the description of the error occurred.
- ErrorCodeOut – Returns the error code**,** passes error or warning information out of an API to be used by other C APIs.

DEPENDENCIES

- Header – Header – Header – DRM Generation.h, extcode.h, fundtype.h, platdefines.h
- Library – DRM Generation.lib
- DLL – DRM Generation.dll

3.17 MaxEye DRM SG Remote TCP Close Connection

NAME MaxEye_DRM_SG_Remote_TCP_Close_Connection

DESCRIPTION Closes TCP network connection between DRM SFP Client and Server applications

FUNCTION PROTOTYPE

```
void __cdecl MaxEye_DRM_SG_Remote_TCP_Close_Connection
(
        LVRefNum      *ConnectionIDIn,
        int32_t       ErrorCodeIn
        int32_t       *ErrorCodeOut
)
```

INPUT PARAMETERS

- ConnectionIDIn – Specifies the TCP connection reference. Connection ID In is a network connection reference that uniquely identifies the TCP connection.
- ErrorCodeIn – Specifies the error code. The ErrorCodeIn can accept error information from previously called C API. Use this information to decide if any functionality should be bypassed in the event of errors from other C APIs.

OUTPUT PARAMETERS

- ErrorCodeOut – Returns the error code**,** passes error or warning information out of an API to be used by other C APIs.

DEPENDENCIES

- Header – Header – Header – DRM Generation.h, extcode.h, fundtype.h, platdefines.h
- Library – DRM Generation.lib
- DLL – DRM Generation.dll

3.18 MaxEye DRM SG Remote Save Configuration

NAME MaxEye_DRM_SG_Remote_Save_Configuration

DESCRIPTION Configures the file path to save the configurations in file.

FUNCTION PROTOTYPE

```
void __cdecl MaxEye_DRM_SG_Remote_Save_Configuration
(
        LVRefNum      *ConnectionIDIn,
```

```
        char        SaveConfigurationFilePath[],
        int32_t     ErrorCodeIn,
        LVRefNum    *ConnectionIDOut,
        int32_t     *ErrorCodeOut
)
```

INPUT PARAMETERS

- ConnectionIDIn – Specifies the TCP connection reference. Connection ID In is a network connection reference that uniquely identifies the TCP connection.
- SaveConfigurationFilePath[] – Specifies the file path to save the configurations in file.
- ErrorCodeIn – Specifies the error code. The ErrorCodeIn can accept error information from previously called C API. Use this information to decide if any functionality should be bypassed in the event of errors from other C APIs.

OUTPUT PARAMETERS

- ConnectionIDOut – Returns the TCP connection reference. Connection ID Out is a network connection reference that uniquely identifies the TCP connection. Use this value to refer to this connection in subsequent API calls.
- ErrorCodeOut – Returns the error code**,** passes error or warning information out of an API to be used by other C APIs.

DEPENDENCIES

- Header – DRM Generation.h, extcode.h, fundtype.h, platdefines.h, lv_epilog.h
- Library – DRM Generation.lib
- DLL – DRM Generation.dll

3.19 MaxEye DVBT SG Remote Load Configuration

NAME MaxEye_DRM_SG_Remote_Load_Configuration

DESCRIPTION Configures the file path to load the saved configurations from file.

FUNCTION PROTOTYPE

```
void __cdecl MaxEye_DRM_SG_Remote_Load_Configuration
(
        LVRefNum    *ConnectionIDIn,
        char        LoadConfigurationFilePath[],
        int32_t     ErrorCodeIn,
        LVRefNum    *ConnectionIDOut,
        int32_t     *ErrorCodeOut
)
```

INPUT PARAMETERS

- ConnectionIDIn – Specifies the TCP connection reference. Connection ID In is a network connection reference that uniquely identifies the TCP connection.

- LoadConfigurationFilePath[] – Specifies the file path to load the saved configurations from file.
- ErrorCodeIn – Specifies the error code. The ErrorCodeIn can accept error information from previously called C API. Use this information to decide if any functionality should be bypassed in the event of errors from other C APIs.

OUTPUT PARAMETERS

- ConnectionIDOut – Returns the TCP connection reference. Connection ID Out is a network connection reference that uniquely identifies the TCP connection. Use this value to refer to this connection in subsequent API calls.
- ErrorCodeOut – Returns the error code**,** passes error or warning information out of an API to be used by other C APIs.

DEPENDENCIES

- Header – DRM Generation.h, extcode.h, fundtype.h, platdefines.h, lv_epilog.h
- Library – DRM Generation.lib
- DLL – DRM Generation.dll