



User Manual

SDK Feature 사용

Rev 2.01

TS 팀

CREVIS

[illegible]

Table of Contents

1. Feature 찾기.....	4
1.1 소스코드 적용 방법.....	5
1.1.1 CvsCam.....	5
1.1.2 MCAM40.....	6
2. cvsFeatureGenerator.....	7
2.1. 사용 방법.....	7
2.1. 생성 예시.....	9

1. Feature 찾기

CvsCam SDK에서는 예제에서 제공하고 있지 않은 Feature를 찾아 사용 할 수 있습니다.

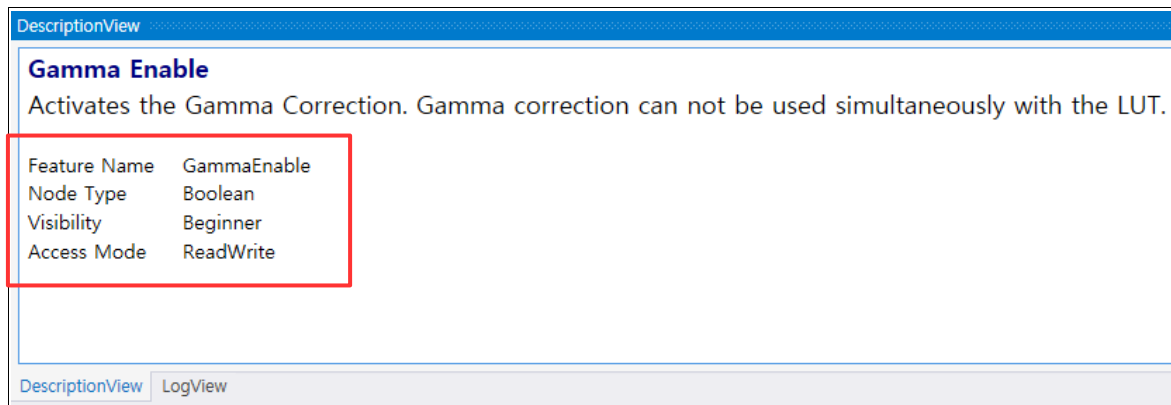
1. CamGuide에서 사용하고자 하는 Feature 선택

ex) Gamma Enable을 선택합니다.

Gamma Enable	False
Gamma	1.000000
Noise Reduction Filter	False

2. Description View 확인

- Feature Name : 함수에서 사용 할 Feature 이름을 확인 할 수 있습니다.
- Node Type : 해당 Feature를 조작하기 위해 어떤 타입의 함수를 사용해야 하는지 확인 할 수 있습니다.
- Access Mode : 해당 Feature의 권한을 확인 할 수 있습니다.



1.1 소스코드 적용 방법

1. Description View의 NodeType을 보고 원하는 자료형의 함수를 사용합니다.
2. 함수 내의 Parameter값에 FeatureName을 직접 입력합니다.

1.1.1 CvsCam

VirtualFG40, CvsCamCtrl, CvsCamDotnet API가 있으며 종류는 다르지만 사용 방법은 유사합니다.

아래 예시는 VirtualFG40 기준으로 작성되었습니다.

C#

```
// Boolean Featrue
status = s_virtualFG.SetBoolReg(hDevice, "GammaEnable", true);
if (status != VirtualFG40Library.MCAM_ERR_SUCCESS)
{
    throw new Exception(string.Format("Write Register failed : {0}", status));
}
```

C++

```
// Boolean Featrue
status = ST_SetBoolReg(hDevice, "GammaEnable", true);
CVS_CHECK(status);
```

1.1.2 MCAM40

VirtualFG40 API만 존재하며, 사용 방법은 유사합니다.

C#

```
// Boolean Feature
status = _virtualFG40.SetBoolReg(hDevice, "GammaEnable", true);
if (status != VirtualFG40Library.MCAM_ERR_SUCCESS)
{
    throw new Exception(String.Format("Write Register failed : {0}", status));
}
```

C++

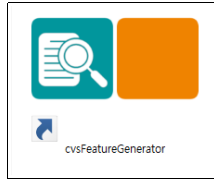
```
// Boolean Feature
status = ST_SetBoolReg(hDevice, "GammaEnable", true);
if(status != MCAM_ERR_SUCCESS)
{
    printf("Write Register failed : %d\n", status);
}
```

VB.NET

```
' Boolean Feature
status = _virtualFG40.SetBoolReg(hDevice, "GammaEnable", True)
If (status <> VirtualFG40Library.MCAM_ERR_SUCCESS) Then
    Throw New Exception(String.Format("Write Register failed : {0}", status))
End If
```

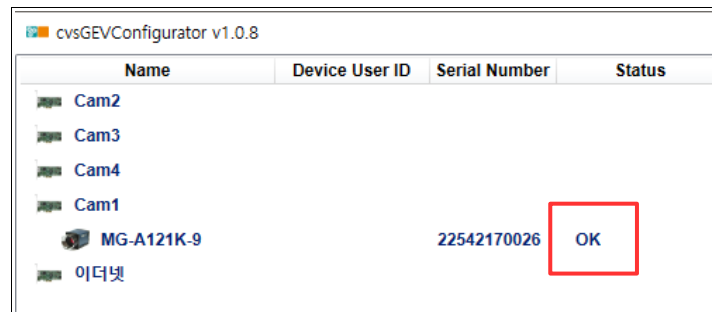
2. cvsFeatureGenerator

CvsCam SDK에서는 카메라의 Feature 클래스를 생성 할 수 있는 툴을 제공하고 있습니다.

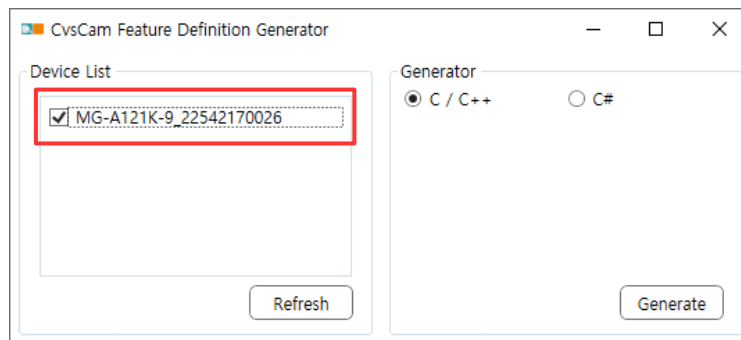


2.1. 사용 방법

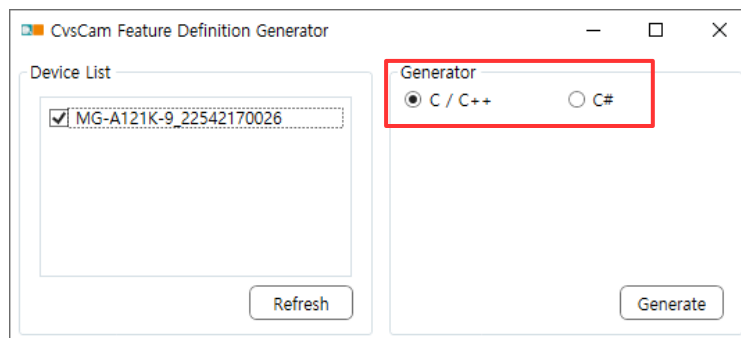
1. 생성하고자 하는 인터페이스의 카메라를 연결합니다.
GigE의 경우 cvsGEVConfigurator의 카메라 Status가 OK여야 합니다.



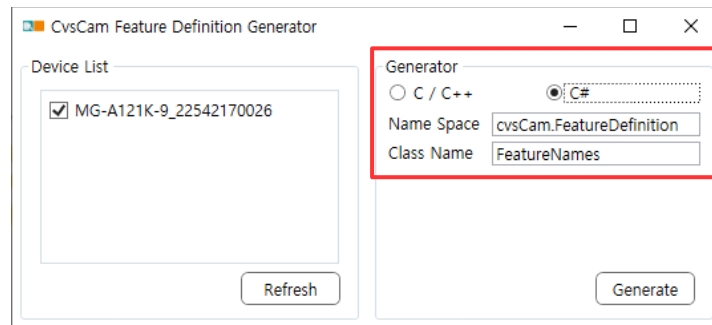
2. 생성하고자 하는 인터페이스의 카메라를 선택합니다.



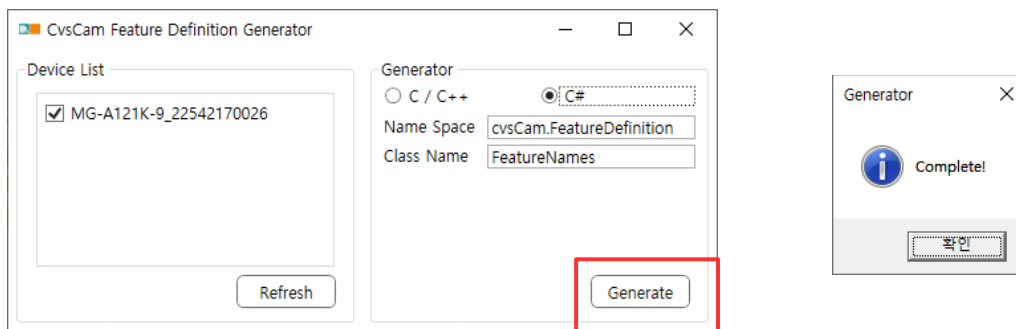
3. 생성하고자 하는 언어를 선택합니다.
 - C / C++은 .h파일이 생성됩니다.
 - C#은 .cs 파일이 생성됩니다.



4. C#의 경우 Name Space와 Class Name 지정이 가능합니다.



5. Generate 버튼 클릭 시 파일 저장이 가능하며, 성공하면 알람 메시지가 출력 됩니다.



2.1. 생성 예시

C / C++

```

1  /*--- Features for : MG-A121K-9 ---*/
2  // Feature Definitions
3  // DeviceControl
4  #define DEVICE_SCAN_TYPE      (char*)"DeviceScanType"      //Enum
5  #define DEVICE_VENDOR_NAME    (char*)"DeviceVendorName"    //String
6  #define DEVICE_MODEL_NAME     (char*)"DeviceModelName"     //String
7  #define DEVICE_MANUFACTURER_INFO (char*)"DeviceManufacturerInfo" //String
8  #define DEVICE_VERSION        (char*)"DeviceVersion"        //String
9  #define DEVICE_ID             (char*)"DeviceID"             //String
10 #define DEVICE_USER_ID        (char*)"DeviceUserID"         //String
11 #define DEVICE_MAX_THROUGHPUT  (char*)"DeviceMaxThroughput"  //Integer
12 #define DEVICE_TEMPERATURE_SELECTOR (char*)"DeviceTemperatureSelector" //Enum
13 #define DEVICE_TEMPERATUREC    (char*)"DeviceTemperature"    //Float
14 #define TEMPERATURE_UPDATE_FEATURE (char*)"TemperatureUpdateFeature" //Command

```

C#

```

namespace cvsCam.FeatureDefinition
{
    public static class FeatureNames
    {
        // --- Features for : MG-A121K-9 ---

        // Feature Definitions
        // DeviceControl
        public const string DEVICE_SCAN_TYPE = "DeviceScanType"; // Enum
        public const string DEVICE_VENDOR_NAME = "DeviceVendorName"; // String
        public const string DEVICE_MODEL_NAME = "DeviceModelName"; // String
        public const string DEVICE_MANUFACTURER_INFO = "DeviceManufacturerInfo"; // String
        public const string DEVICE_VERSION = "DeviceVersion"; // String
        public const string DEVICE_ID = "DeviceID"; // String
        public const string DEVICE_USER_ID = "DeviceUserID"; // String
        public const string DEVICE_MAX_THROUGHPUT = "DeviceMaxThroughput"; // Integer
        public const string DEVICE_TEMPERATURE_SELECTOR = "DeviceTemperatureSelector"; // Enum
        public const string DEVICE_TEMPERATUREC = "DeviceTemperature"; // Float
        public const string TEMPERATURE_UPDATE_FEATURE = "TemperatureUpdateFeature"; // Command
    }
}

```