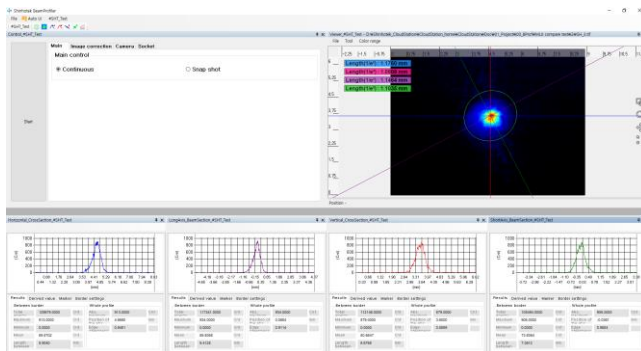




# Lumosity

## Software (based on .net Framework) Interface Manual



ExempleXMLInterface

Connect Disconnect Get info (Refresh) Status : Connected Version : 1.3.23.0

Evaluation Option  
☐ Continuous Interval : 1 Show preview Start / Stop Start Stop

Remote System  
Main Image correction Cam Sections ROI

Main control  
☒ Continuous ☐ Snap shot  
☐ External triggered

Image process  
☐ Floating average average over 2 Frames  
☐ Number restriction 2 Frames

Evaluation Available items

Name	Description
<input type="checkbox"/> FRAME_DATE	Date of the acquisition of the frame
<input type="checkbox"/> FRAME_TIME	Time of the acquisition of the frame
<input type="checkbox"/> FRAME_NUMBER	Number of the acquisition of the frame
<input type="checkbox"/> WIDTH	Width of the frame
<input type="checkbox"/> HEIGHT	Height of the frame
<input type="checkbox"/> FRAME_COG_X	Centroid in x-direction of all intensities of the frame
<input type="checkbox"/> FRAME_COG_Y	Centroid in y-direction of all intensities of the frame
<input type="checkbox"/> FRAME_BEAMWID...	Beam width long is the diameter of the Longaxis
<input type="checkbox"/> FRAME_BEAMWID...	Beam width short is the diameter of the Longaxis

Evaluation Use items

Name	Value
------	-------

Key Name : Value(unit) : Get

Open source

## 목차

1. 소개 .....	7
2. Lumosity 와 연결 .....	7
1) Server 동작 .....	7
2) Client 동작 .....	7
3) .net Framework Class library DLL .....	8
3. Lumosity xml interface class 구조 .....	9
1) Enum / Struct .....	9
⊙ struct EvaluationDataSet .....	9
⊙ enum GrabMode .....	9
⊙ enum FrameROI Mode .....	9
⊙ enum FrameROI Shape .....	9
⊙ enum FrameSectionBorderMode .....	9
⊙ enum ModeAutoControl .....	10
⊙ enum ModeBackgroundImageCorrection .....	10
⊙ enum GetFrameFormat .....	10
2) Event .....	10
⊙ event EventHandler FrameEvaluations .....	10
⊙ event EventHandler Disconnected .....	10
⊙ event EventHandler ErrorOccurred .....	10
3) Property .....	10
⊙ string[] AvailableEvaluations {get} .....	11
⊙ string[] UseEvaluations {get} .....	11
⊙ bool IsConnected {get} .....	11
⊙ string ConnectionStatus {get} .....	11
⊙ int SocketTimeOut {get, set} .....	11
⊙ string Version {get} .....	11
⊙ GrabMode GrabModeType {get, set} .....	11
⊙ bool FloatingAverageEnable {get, set} .....	11

⊙ int FloatingAverageValue {get, set}.....	11
⊙ int FloatingAverageValueMax {get} .....	11
⊙ bool NumberRestrictionEnable {get, set}.....	11
⊙ int NumberRestrictionValue {get, set}.....	11
⊙ int NumberRestrictionValueMax {get} .....	12
⊙ string CamID {get}.....	12
⊙ int CamOffsetX {get, set}.....	12
⊙ int CamOffsetY {get, set} .....	12
⊙ int CamWidth {get, set}.....	12
⊙ int CamHeight {get, set} .....	12
⊙ int CamWidthMax {get}.....	12
⊙ int CamHeightMax {get} .....	12
⊙ int CamWidthStep {get}.....	12
⊙ int CamHeightStep {get}.....	12
⊙ int CamDepth {get}.....	12
⊙ double CamPixelSizeWidth {get} .....	13
⊙ double CamPixelSizeHeight {get} .....	13
⊙ bool CamAvailableGain {get} .....	13
⊙ double CamGainMax {get}.....	13
⊙ double CamGainMin {get}.....	13
⊙ double CamGain {get, set}.....	13
⊙ double CamExposureTimeMax {get} .....	13
⊙ double CamExposureTimeMin {get} .....	13
⊙ double CamExposureTime {get, set}.....	13
⊙ double CamTriggerDelayMax {get}.....	13
⊙ double CamTriggerDelayMin {get} .....	13
⊙ double CamTriggerDelay {get, set}.....	14
⊙ ModeAutoControl CamAutoControl {get, set}.....	14

⊙ bool IsEvaluationContinuous {get, set}.....	14
⊙ int EvaluationInterval {get, set}.....	14
⊙ bool FrameROIActive {get, set} .....	14
⊙ FrameROIMode FrameRoiMode {get, set} .....	14
⊙ FrameROIShape FrameRoiMode {get, set}.....	14
⊙ int FrameROILeft {get, set}.....	14
⊙ int FrameROITop {get, set}.....	14
⊙ int FrameROIWidth {get, set} .....	14
⊙ int FrameROIHeight {get, set}.....	14
⊙ bool FrameCrossSectionActive {get, set} .....	15
⊙ int FrameCrossSectionRow {get, set}.....	15
⊙ int FrameCrossSectionCol {get, set}.....	15
⊙ bool FrameCrossSectionAuto {get, set}.....	15
⊙ bool FrameBeamSectionActive {get, set}.....	15
⊙ int FrameBeamSectionRow {get, set} .....	15
⊙ int FrameBeamSectionCol {get, set} .....	15
⊙ bool FrameBeamSectionAuto {get, set}.....	15
⊙ double FrameBeamSectionAngle {get, set}.....	15
⊙ FrameSectionBorderMode FrameCrossSectionHorizontalBorderType {get, set}	15
⊙ double FrameCrossSectionHorizontalReference {get, set}.....	15
⊙ int FrameCrossSectionHorizontalOffset {get, set}.....	15
⊙ double FrameCrossSectionHorizontalLeft {get, set} .....	16
⊙ double FrameCrossSectionHorizontalRight {get, set} .....	16
⊙ FrameSectionBorderMode FrameCrossSectionVerticalBorderType {get, set}.16	
⊙ double FrameCrossSectionVerticalReference {get, set}.....	16
⊙ int FrameCrossSectionVerticalOffset {get, set}.....	16
⊙ double FrameCrossSectionVerticalLeft {get, set}.....	16

⊙	double FrameCrossSectionVerticalRight {get, set}.....	16
⊙	FrameSectionBorderMode FrameCrossSectionLongBorderType {get, set} .....	16
⊙	double FrameCrossSectionLongReference {get, set}.....	16
⊙	int FrameCrossSectionLongOffset {get, set}.....	16
⊙	double FrameCrossSectionLongLeft {get, set}.....	16
⊙	double FrameCrossSectionLongRight {get, set} .....	16
⊙	FrameSectionBorderMode FrameCrossSectionShortBorderType {get, set} .....	17
⊙	double FrameCrossSectionShortReference {get, set} .....	17
⊙	int FrameCrossSectionShortOffset {get, set}.....	17
⊙	double FrameCrossSectionShortLeft {get, set} .....	17
⊙	double FrameCrossSectionShortRight {get, set} .....	17
⊙	ModeBackgroundImageCorrection BackgroundMode {get, set} .....	17
⊙	bool BackgroundImageEnable {get, set}.....	17
⊙	int BackgroundAcquireFrames {get, set}.....	17
⊙	bool BackgroundCornerEnable {get, set}.....	17
⊙	int BackgroundCornerSpan {get, set}.....	17
⊙	bool BackgroundCornerOverlay {get, set} .....	17
⊙	bool BackgroundContantEnable {get, set} .....	17
⊙	double BackgroundConstantValue {get, set}.....	18
⊙	bool EvaluationGetFrameActive {get, set} .....	18
⊙	int EvaluationGetFrameScale {get, set}.....	18
⊙	GetFrameFormat EvaluationGetFrameFormat {get, set} .....	18
⊙	bool EvaluationGetFrameSucceeded {get}.....	18
⊙	int EvaluationGetFrameBitmapWidth {get}.....	18
⊙	int EvaluationGetFrameBitmapHeight {get} .....	18
⊙	Bitmap EvaluationGetFrameBitmap {get}.....	18
⊙	string ErrorDetail {get} .....	18
4)	Method.....	19

⦿	<code>bool Connect(string ip, int port, int timeout = 7000)</code>	19
⦿	<code>void Disconnect()</code>	19
⦿	<code>bool CmdGetInfomation()</code>	19
⦿	<code>bool Start()</code>	19
⦿	<code>bool Stop()</code>	19
⦿	<code>bool StartBackground()</code>	19
⦿	<code>void AddUseEvaluation(string evaluationName)</code>	19
⦿	<code>void RemoveUseEvaluation (string evaluationName)</code>	20
⦿	<code>void ClearUseEvaluations()</code>	20
⦿	<code>string GetEvaluationDescription(string evaluationName)</code>	20

## 1. 소개

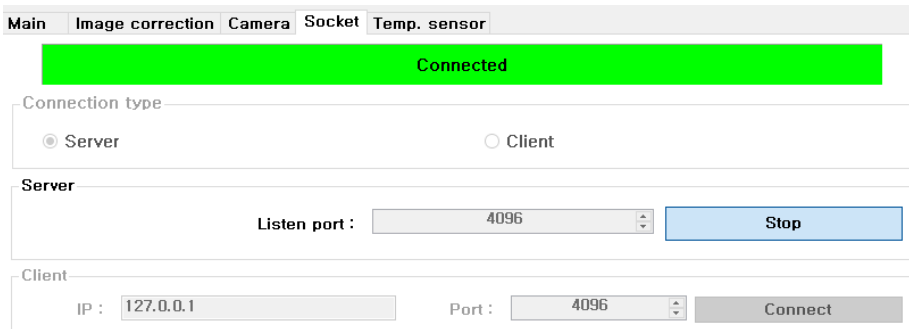
Lumosity XML protocol 을 조금 더 쉽고 직관적으로 사용할 수 있도록 Open source 로 제작된 .net Framework 기반의 Class library DLL 입니다. (Sample code 포함) 코드를 수정이 가능하며, 기존에 제공되고 있던 XML interface 를 Socket 에 연결하고, Data 를 주고받는 통신에 대한 구현을 시간 단축하고자 제작되었습니다.

이 코드는 [GitHub](https://github.com/Shinhotek/LumositySWInterface.git) 에 수정가능 하도록 공개되어 있고, 코드의 사용/수정에는 제한이 없습니다. 다만, Commit 을 할 때에는 당사에 알려주셔야 합니다. (설치한 Lumosity Software 에도 포함되어 있습니다.)

 GitHub Link : <https://github.com/Shinhotek/LumositySWInterface.git>

## 2. LUMOSITY 와 연결

XML 인터페이스를 제공하는 Lumosity software 는 두 가지 (Server, Client) 기능을 제공합니다. 이 설정은 Lumosity software 의 Control tab 의 Socket 탭에서 설정과 제어를 할 수 있습니다.



### 1) Server 동작

서버를 선택하면 Lumosity software 에서 단일 클라이언트 연결을 위한 서버 소켓을 제공합니다. 서버의 Port 번호는 서버의 동작을 멈춘 상태에서 바꿀 수 있습니다. 서버의 Start 버튼으로 Listening 상태를 만들 수 있고 이 상태에서 외부 클라이언트 Software 와 연결이 가능합니다.

### 2) Client 동작

클라이언트를 선택하면 Lumosity software 에서 외부서버의 Software 와 연결이 가능합니다. IP 가 127.0.0.1 로 되어 있는 경우 한 PC 에서 연결이 가능합니다. 다른 PC 와 연결할 경우 IP 를 다른 PC 의 IP 를 확인한 후 해당 IP 주소와 PORT 를 입력하여 연결이 가능합니다. Connect 버튼으로 연결하고 연결 시 Server 가 준비되어 있지 않다면, Connecting...상태를 일정시간 재시도 하다가 실패 시 None 상태로 변경됩니다.

### 3) .net Framework Class library DLL

Lumosity software 와 XML Protocol 으로 측정된 Data 를 가져오는 부분을 간단하게 Library 형태로 구현하였습니다. 프로토콜 레벨보다 편리하게 사용할 수 있도록 제작되었습니다.



### 3. LUMOSITY XML INTERFACE CLASS 구조

C# Class 로 아래와 같이 구현되어 있습니다.

#### 1) Enum / Struct

: LumosityXMLInterface 에 사용되는 옵션 및 Data 구조에 사용되는 struct 와 enum 입니다.

##### ⊙ **struct EvaluationDataSet**

: Frame 의 측정 항목들의 Data 를 표현하는 데이터 구조

Type	Name	Description
sting	val	Evaluation 의 값
string	unit	Evaluation 의 단위

##### ⊙ **enum GrabMode**

: Main control Mode 항목

Continuous	지속적인 이미지 취득
SnapShot	한번의 이미지 취득
ExternalTriggered	외부 트리거를 이용한 이미지 취득

##### ⊙ **enum FrameROIMode**

: Viewer 의 ROI Mode

MANUAL	Viewer ROI 수동으로 조작할 수 있는 모드
CENTROID	Viewer ROI 의 Size 만 조작할 수 있는 모드(위치는 자동 이동)
AUTORESIZE	Viewer ROI 의 위치와 Size 모두 자동으로 조절

##### ⊙ **enum FrameROIShape**

: Viewer 의 ROI Mode

RECTANGLE	Viewer ROI 사각모양
CIRCLE	Viewer ROI 원형모양

##### ⊙ **enum FrameSectionBorderMode**

: Sections border Mode

MANUAL	Border 를 마우스를 이용하여 수동으로 이동 가능
REFERENCE	Border 의 Percentage of reference 값을 기준으로 자동으로 Border 이동

⊙ **enum ModeAutoControl**

: Camera 의 자동조절 mode

AutoOff	모든 Camera property 를 수동으로 조절
AutoExposure	Camera 의 Exposure time 을 자동으로 조절

⊙ **enum ModeBackgroundImageCorrection**

: Background correction mode

IMAGE	Background image 기반 노이즈 제거 모드
CORNER	Corner 4 부분의 영역의 Intensity 를 노이즈로 기준하여 노이즈 제거 모드
CONSTANT	상수를 설정하여 노이즈를 상수 값만큼 제거하는 모드

⊙ **enum GetFrameFormat**

: Frame image format

BMP	BMP format
JPG	JPG format
PNG	PNG format

## 2) Event

: LumosityXMLInterface 를 사용할 때, 결과값 또는 Error 등 Frame 이 발생되어 측정된 값을 업데이트 하거나 Error 상황이 발생했을 때 이벤트들입니다.

⊙ **event EventHandler FrameEvaluations**

: Frame 의 측정 항목의 결과에 대한 Event

⊙ **event EventHandler Disconnected**

: Socket 의 연결이 끊어짐에 대한 Event

⊙ **event EventHandler ErrorOccurred**

: Error 발생에 대한 Event

## 3) Property

: LumosityXMLInterface 를 사용할 때 사용 가능한 값 또는 연결상태, Error 상태 등과 같은 속성을 가지고 있습니다.

- ⊙ ***string[] AvailableEvaluations {get}***  
: 사용 가능한 Evaluation 항목
- ⊙ ***string[] UseEvaluations {get}***  
: 사용중인 Evaluation 의 항목
- ⊙ ***bool IsConnected {get}***  
: Lumosity 와 Socket 연결/연결 끊김 상태
- ⊙ ***string ConnectionStatus {get}***  
: Lumosity 와 Socket 연결 상세 상태
- ⊙ ***int SocketTimeOut {get, set}***  
: Lumosity 와 연결 중 Data 지연 Timeout
- ⊙ ***string Version {get}***  
: 연결된 Lumosity 의 버전
- ⊙ ***GrabMode GrabModeType {get, set}***  
: Main control 의 Type mode
- ⊙ ***bool FloatingAverageEnable {get, set}***  
: Main control 의 Image process 에서 Float average 활성화/비활성
- ⊙ ***int FloatingAverageValue {get, set}***  
: Main control 의 Image process 에서 Float average Frame 수
- ⊙ ***int FloatingAverageValueMax {get}***  
: Main control 의 Image process 에서 Float average Frame 최대 설정가능 수
- ⊙ ***bool NumberRestrictionEnable {get, set}***  
: Main control 의 Image process 에서 Number restriction 활성화/비활성
- ⊙ ***int NumberRestrictionValue {get, set}***  
: Main control 의 Image process 에서 Number restriction Frame 수

- ⊙ ***int NumberRestrictionValueMax {get}***  
: Main control 의 Image process 에서 Number restriction 최대 설정가능 수
- ⊙ ***string CamID {get}***  
: 연결된 Lumosity Beam profiler 의 카메라 ID
- ⊙ ***int CamOffsetX {get, set}***  
: Camera offset X
- ⊙ ***int CamOffsetY {get, set}***  
: Camera offset Y
- ⊙ ***int CamWidth {get, set}***  
: Camera Width
- ⊙ ***int CamHeight {get, set}***  
: Camera Height
- ⊙ ***int CamWidthMax {get}***  
: Camera Width 최대 설정가능 값
- ⊙ ***int CamHeightMax {get}***  
: Camera Height 최대 설정가능 값
- ⊙ ***int CamWidthStep {get}***  
: Camera Width step
- ⊙ ***int CamHeightStep {get}***  
: Camera Height step
- ⊙ ***int CamDepth {get}***  
: Camera intensity depth (camera bit 수에 따른 밝기강도의 해상도 깊이)

- ⊙ ***double CamPixelSizeWidth {get}***  
: Camera Width Pixel size
- ⊙ ***double CamPixelSizeHeight {get}***  
: Camera Height Pixel size
- ⊙ ***bool CamAvailableGain {get}***  
: Camera Gain 조절가능 여부
- ⊙ ***double CamGainMax {get}***  
: Camera Gain 최대 설정가능 값
- ⊙ ***double CamGainMin {get}***  
: Camera Gain 최소 설정가능 값
- ⊙ ***double CamGain {get, set}***  
: Camera Gain
- ⊙ ***double CamExposureTimeMax {get}***  
: Camera Exposure time 최대 설정가능 값
- ⊙ ***double CamExposureTimeMin {get}***  
: Camera Exposure time 최소 설정가능 값
- ⊙ ***double CamExposureTime {get, set}***  
: Camera Exposure time
- ⊙ ***double CamTriggerDelayMax {get}***  
: Camera Trigger delay 최대 설정가능 값
- ⊙ ***double CamTriggerDelayMin {get}***  
: Camera Trigger delay 최소 설정가능 값

- ⊙ ***double CamTriggerDelay {get, set}***  
: Camera Trigger delay
- ⊙ ***ModeAutoControl CamAutoControl {get, set}***  
: Camera auto control mode
- ⊙ ***bool IsEvaluationContinuous {get, set}***  
: Evaluation Frame capture 마다 값을 받아올지 여부
- ⊙ ***int EvaluationInterval {get, set}***  
: Evaluation interval 설정
- ⊙ ***bool FrameROIActive {get, set}***  
: Frame ROI 기능 활성화/비활성
- ⊙ ***FrameROIMode FrameRoiMode {get, set}***  
: Viewer 의 Frame ROI mode 설정
- ⊙ ***FrameROIShape FrameRoiMode {get, set}***  
: Viewer 의 Frame ROI shape
- ⊙ ***int FrameROILeft {get, set}***  
: Viewer 의 Frame ROI 의 Left 위치 이동 (단위: Pixel)
- ⊙ ***int FrameROITop {get, set}***  
: Viewer 의 Frame ROI 의 Top 위치 이동 (단위: Pixel)
- ⊙ ***int FrameROIWidth {get, set}***  
: Viewer 의 Frame ROI 의 넓이 (단위: Pixel)
- ⊙ ***int FrameROIHeight {get, set}***  
: Viewer 의 Frame ROI 의 높이 (단위: Pixel)

- ⊙ ***bool FrameCrossSectionActive {get, set}***  
: Viewer 의 Cross section 활성화/비활성
- ⊙ ***int FrameCrossSectionRow {get, set}***  
: Viewer 의 Cross section 의 수평(Horizontal) 이동
- ⊙ ***int FrameCrossSectionCol {get, set}***  
: Viewer 의 Cross section 의 수직(Vertical) 이동
- ⊙ ***bool FrameCrossSectionAuto {get, set}***  
: Viewer 의 Cross section 의 자동 추적 활성화/비활성
- ⊙ ***bool FrameBeamSectionActive {get, set}***  
: Viewer 의 Cross section 의 자동 추적 활성화/비활성
- ⊙ ***int FrameBeamSectionRow {get, set}***  
: Viewer 의 Beam section 의 장축(Long axis) 이동
- ⊙ ***int FrameBeamSectionCol {get, set}***  
: Viewer 의 Beam section 의 단축(Short axis) 이동
- ⊙ ***bool FrameBeamSectionAuto {get, set}***  
: Viewer 의 Beam section 의 자동 추적 활성화/비활성
- ⊙ ***double FrameBeamSectionAngle {get, set}***  
: Viewer 의 Beam section 의 각도 조절
- ⊙ ***FrameSectionBorderMode FrameCrossSectionHorizontalBorderType {get, set}***  
: Cross section Horizontal Border mode
- ⊙ ***double FrameCrossSectionHorizontalReference {get, set}***  
: Cross section Horizontal Border percentage of Ref.
- ⊙ ***int FrameCrossSectionHorizontalOffset {get, set}***  
: Cross section Horizontal Border Percentage offset

- ⊙ *double* *FrameCrossSectionHorizontalLeft {get, set}*  
: Cross section Horizontal Border Left
- ⊙ *double* *FrameCrossSectionHorizontalRight {get, set}*  
: Cross section Horizontal Border Right
- ⊙ *FrameSectionBorderMode FrameCrossSectionVerticalBorderType {get, set}*  
: Cross section Vertical Border mode
- ⊙ *double* *FrameCrossSectionVerticalReference {get, set}*  
: Cross section Vertical Border percentage of Ref.
- ⊙ *int* *FrameCrossSectionVerticalOffset {get, set}*  
: Cross section Vertical Border Percentage offset
- ⊙ *double* *FrameCrossSectionVerticalLeft {get, set}*  
: Cross section Vertical Border Left
- ⊙ *double* *FrameCrossSectionVerticalRight {get, set}*  
: Cross section Vertical Border Right
- ⊙ *FrameSectionBorderMode FrameCrossSectionLongBorderType {get, set}*  
: Cross section Long axis Border mode
- ⊙ *double* *FrameCrossSectionLongReference {get, set}*  
: Cross section Long axis Border percentage of Ref.
- ⊙ *int* *FrameCrossSectionLongOffset {get, set}*  
: Cross section Long axis Border Percentage offset
- ⊙ *double* *FrameCrossSectionLongLeft {get, set}*  
: Cross section Long axis Border Left
- ⊙ *double* *FrameCrossSectionLongRight {get, set}*  
: Cross section Long axis Border Right



- ⊙ *FrameSectionBorderMode* *FrameCrossSectionShortBorderType {get, set}*  
: Cross section Short axis Border mode
- ⊙ *double FrameCrossSectionShortReference {get, set}*  
: Cross section Short axis Border percentage of Ref.
- ⊙ *int FrameCrossSectionShortOffset {get, set}*  
: Cross section Short axis Border Percentage offset
- ⊙ *double FrameCrossSectionShortLeft {get, set}*  
: Cross section Short axis Border Left
- ⊙ *double FrameCrossSectionShortRight {get, set}*  
: Cross section Short axis Border Right
- ⊙ *ModeBackgroundImageCorrection* *BackgourndMode {get, set}*  
: Image correction 의 Mode
- ⊙ *bool BackgroundImageEnable {get, set}*  
: Image correction 활성화/비활성화
- ⊙ *int BackgroundAcquireFrames {get, set}*  
: Image correction 의 Background mode 의 Acquire 할 이미지 Frame 수
- ⊙ *bool BackgroundCornerEnable {get, set}*  
: Image correction 의 Corner mode 의 활성화/비활성화
- ⊙ *int BackgroundCornerSpan {get, set}*  
: Image correction 의 Corner mode 의 Span
- ⊙ *bool BackgroundCornerOver lay {get, set}*  
: Image correction 의 Corner mode 에서 Viewer 에 표시 여부
- ⊙ *bool BackgroundContantEnable {get, set}*  
: Image correction 의 Constant 활성화/비활성

- ⊙ *double BackgroundConstantValue {get, set}*  
: Image correction 의 Constant 값 (Intensity)
- ⊙ *bool EvaluationGetFrameActive {get, set}*  
: Frame 취득 시 Viewer 의 컬러 이미지 취득 활성/비활성
- ⊙ *int EvaluationGetFrameScale {get, set}*  
: Frame 취득 시 가져올 컬러 이미지의 Scale 설정 (성능의 영향을 줄 수 있음)
- ⊙ *GetFrameFormat EvaluationGetFrameFormat {get, set}*  
: Frame 취득시 가져올 컬러 이미지의 Format(BMP, JPG, PNG) 설정
- ⊙ *bool EvaluationGetFrameSucceeded {get}*  
: 취득된 Frame 이미지의 성공 여부 확인
- ⊙ *int EvaluationGetFrameBitmapWidth {get}*  
: 취득된 Frame 이미지의 Width
- ⊙ *int EvaluationGetFrameBitmapHeight {get}*  
: 취득된 Frame 이미지의 Height
- ⊙ *Bitmap EvaluationGetFrameBitmap {get}*  
: 취득된 Frame 이미지의 Bitmap data
- ⊙ *string ErrorDetail {get}*  
: 연결된 Lumosity Beam profiler interface Error 내용

#### 4) Method

: LumosityXMLInterface 를 사용할 때 연결 제어 및 Command 처리 등의 기능을 하는 Method 입니다.

⊙ **bool Connect(string ip, int port, int timeout = 7000)**

: 클라이언트 연결기능으로 Lumosity 를 서버로 구동하여 연결한다.

Parameter	Description
ip	Lumosity 의 IP
port	Lumosity 서버 포트
timeout	연결 및 데이터 통신시 timeout 설정
Return	연결 성공여부

⊙ **void Disconnect()**

: Lumosity 와의 연결을 끊는다.

⊙ **bool CmdGetInfomation()**

: 연결된 Lumosity 의 전체 정보(info tag)를 가져온다.

Return	연결 성공여부
--------	---------

⊙ **bool Start()**

: 카메라 Capture(Grab) 하여 Beam profiler 의 측정 구동을 시작

Return	Start 명령의 정상 동작 여부
--------	--------------------

⊙ **bool Stop()**

: 카메라 Capture(Grab) 하여 Beam profiler 의 측정 구동을 시작

Return	Stop 명령의 정상 동작 여부
--------	-------------------

⊙ **bool StartBackground()**

: Image correction 의 background mode 에 필요한 Image acquire(이미지취득)

Return	StartBackground 명령의 정상 동작 여부
--------	------------------------------

⊙ **void AddUseEvaluation(string evaluationName)**

: AvailableEvaluations(사용가능 Eevaluation) 항목에서 필요한 항목을 사용항목으로 추가

Parameter	Description
evaluationName	사용 가능 Evaluation 항목 이름 중에서 사용

⊙ *void RemoveUseEvaluation (string evaluationName)*

: UseEvaluations(사용중인 Evaluation) 항목에서 제거할 항목 지정하여 사용항목에서 제거

Parameter	Description
evaluationName	사용중인 Evaluation 에서 제거할 항목 이름

⊙ *void ClearUseEvaluations()*

: 사용중인 Evaluation 모두 제거

⊙ *string GetEvaluationDescription(string evaluationName)*

: AvailableEvaluations(사용가능 Evaluation)의 설명을 가져옴

Parameter	Description
evaluationName	사용 가능 Evaluation 항목 이름
Return	해당 Evaluation 의 설명