

## Vicon DataStream SDK 1.10.0 Developer's Guide

---

©2013-2020 Vicon Motion Systems Limited. All rights reserved.

Vicon DataStream SDK Developer's Guide April 2020  
For use with Vicon DataStream SDK 1.10.0 and later.

Vicon® is a registered trademark of Oxford Metrics plc. Vicon Blade™, Vicon Shōgun™, Vicon Nexus™, Vicon Tracker™, Vicon Evoke™, Bonita™, Vicon MX™, and Vicon MX T-Series™ are trademarks of Oxford Metrics plc. Other product and company names herein may be the trademarks of their respective owners. Vicon Motion Systems is an Oxford Metrics plc company.  
Email: [support@vicon.com](mailto:support@vicon.com). Web: [www.vicon.com](http://www.vicon.com).

## Contents

<b>Developer's Guide</b>	<b>1</b>
<b>Class Index</b>	<b>6</b>
Class List . . . . .	6
<b>Class Documentation</b>	<b>7</b>
Client Class Reference . . . . .	7
Detailed Description . . . . .	7
Constructor & Destructor Documentation . . . . .	14
Client . . . . .	14
~Client . . . . .	15
Member Function Documentation . . . . .	15
GetVersion . . . . .	15
Connect . . . . .	17
ConnectToMulticast . . . . .	18
Disconnect . . . . .	20
IsConnected . . . . .	21
StartTransmittingMulticast . . . . .	22
StopTransmittingMulticast . . . . .	23
EnableSegmentData . . . . .	24
EnableLightweightSegmentData . . . . .	25
EnableMarkerData . . . . .	26
EnableUnlabeledMarkerData . . . . .	27
EnableMarkerRayData . . . . .	28
EnableDeviceData . . . . .	29
EnableCentroidData . . . . .	30
EnableGreyscaleData . . . . .	31

EnableVideoData . . . . .	32
EnableDebugData . . . . .	33
DisableSegmentData . . . . .	34
DisableLightweightSegmentData . . . . .	35
DisableMarkerData . . . . .	36
DisableUnlabeledMarkerData . . . . .	37
DisableMarkerRayData . . . . .	38
DisableDeviceData . . . . .	39
DisableCentroidData . . . . .	40
DisableGreyscaleData . . . . .	41
DisableVideoData . . . . .	42
DisableDebugData . . . . .	43
IsSegmentDataEnabled . . . . .	44
IsLightweightSegmentDataEnabled . . . . .	45
IsMarkerDataEnabled . . . . .	46
IsUnlabeledMarkerDataEnabled . . . . .	47
IsMarkerRayDataEnabled . . . . .	48
IsDeviceDataEnabled . . . . .	49
IsCentroidDataEnabled . . . . .	50
IsGreyscaleDataEnabled . . . . .	51
IsVideoDataEnabled . . . . .	52
IsDebugDataEnabled . . . . .	53
SetBufferSize . . . . .	54
SetStreamMode . . . . .	55
SetApexDeviceFeedback . . . . .	57
SetAxisMapping . . . . .	58
GetAxisMapping . . . . .	59
GetFrame . . . . .	60
GetFrameNumber . . . . .	61
GetTimecode . . . . .	62
GetFrameRate . . . . .	63
GetLatencySampleCount . . . . .	64
GetLatencySampleName . . . . .	65
GetLatencySampleValue . . . . .	67
GetLatencyTotal . . . . .	69

GetHardwareFrameNumber . . . . .	70
GetFrameRateCount . . . . .	71
GetFrameRateName . . . . .	72
GetFrameRateValue . . . . .	73
GetSubjectCount . . . . .	74
GetSubjectName . . . . .	75
GetSubjectRootSegmentName . . . . .	77
GetSegmentCount . . . . .	79
GetSegmentName . . . . .	81
GetSegmentChildCount . . . . .	83
GetSegmentChildName . . . . .	85
GetSegmentParentName . . . . .	87
GetSegmentStaticTranslation . . . . .	89
GetSegmentStaticRotationHelical . . . . .	91
GetSegmentStaticRotationMatrix . . . . .	93
GetSegmentStaticRotationQuaternion . . . . .	95
GetSegmentStaticRotationEulerXYZ . . . . .	97
GetSegmentStaticScale . . . . .	99
GetSegmentGlobalTranslation . . . . .	101
GetSegmentGlobalRotationHelical . . . . .	103
GetSegmentGlobalRotationMatrix . . . . .	105
GetSegmentGlobalRotationQuaternion . . . . .	107
GetSegmentGlobalRotationEulerXYZ . . . . .	109
GetSegmentLocalTranslation . . . . .	111
GetSegmentLocalRotationHelical . . . . .	113
GetSegmentLocalRotationMatrix . . . . .	115
GetSegmentLocalRotationQuaternion . . . . .	117
GetSegmentLocalRotationEulerXYZ . . . . .	119
GetObjectQuality . . . . .	121
GetMarkerCount . . . . .	123
GetMarkerName . . . . .	125
GetMarkerParentName . . . . .	128
GetMarkerGlobalTranslation . . . . .	130
GetMarkerRayContributionCount . . . . .	132
GetMarkerRayContribution . . . . .	134

GetUnlabeledMarkerCount . . . . .	136
GetUnlabeledMarkerGlobalTranslation . . . . .	137
GetLabeledMarkerCount . . . . .	139
GetLabeledMarkerGlobalTranslation . . . . .	140
GetDeviceCount . . . . .	141
GetDeviceName . . . . .	142
GetDeviceOutputCount . . . . .	145
GetDeviceOutputName . . . . .	147
GetDeviceOutputComponentName . . . . .	150
GetDeviceOutputValue . . . . .	153
GetDeviceOutputValue . . . . .	155
GetDeviceOutputSubsamples . . . . .	157
GetDeviceOutputSubsamples . . . . .	159
GetDeviceOutputValue . . . . .	161
GetDeviceOutputValue . . . . .	163
GetForcePlateCount . . . . .	165
GetGlobalForceVector . . . . .	166
GetGlobalMomentVector . . . . .	168
GetGlobalCentreOfPressure . . . . .	170
GetForcePlateSubsamples . . . . .	172
GetGlobalForceVector . . . . .	174
GetGlobalMomentVector . . . . .	176
GetGlobalCentreOfPressure . . . . .	178
GetEyeTrackerCount . . . . .	180
GetEyeTrackerGlobalPosition . . . . .	181
GetEyeTrackerGlobalGazeVector . . . . .	183
GetCameraCount . . . . .	185
GetCameraName . . . . .	186
GetCameraId . . . . .	188
GetCameraUserId . . . . .	190
GetCameraType . . . . .	192
GetCameraDisplayName . . . . .	194
GetCameraResolution . . . . .	196
GetIsVideoCamera . . . . .	198
GetCentroidCount . . . . .	200

GetCentroidPosition . . . . .	202
GetCentroidWeight . . . . .	204
GetGreyscaleBlobCount . . . . .	206
GetGreyscaleBlob . . . . .	207
GetVideoFrame . . . . .	208
SetCameraFilter . . . . .	209
ClearSubjectFilter . . . . .	210
AddToSubjectFilter . . . . .	211
ConfigureWireless . . . . .	213
RetimingClient Class Reference . . . . .	213
Detailed Description . . . . .	213
Constructor & Destructor Documentation . . . . .	217
RetimingClient . . . . .	217
~RetimingClient . . . . .	218
Member Function Documentation . . . . .	218
GetVersion . . . . .	218
Connect . . . . .	220
Disconnect . . . . .	221
IsConnected . . . . .	222
EnableLightweightSegmentData . . . . .	223
DisableLightweightSegmentData . . . . .	224
IsLightweightSegmentDataEnabled . . . . .	225
SetAxisMapping . . . . .	226
GetAxisMapping . . . . .	227
UpdateFrame . . . . .	228
WaitForFrame . . . . .	229
GetSubjectCount . . . . .	230
GetSubjectName . . . . .	231
GetSubjectRootSegmentName . . . . .	233
GetSegmentCount . . . . .	235
GetSegmentName . . . . .	237
GetSegmentChildCount . . . . .	239
GetSegmentChildName . . . . .	241
GetSegmentParentName . . . . .	243
GetSegmentStaticTranslation . . . . .	245

GetSegmentStaticRotationHelical . . . . .	247
GetSegmentStaticRotationMatrix . . . . .	249
GetSegmentStaticRotationQuaternion . . . . .	251
GetSegmentStaticRotationEulerXYZ . . . . .	253
GetSegmentGlobalTranslation . . . . .	255
GetSegmentGlobalRotationHelical . . . . .	257
GetSegmentGlobalRotationMatrix . . . . .	259
GetSegmentGlobalRotationQuaternion . . . . .	261
GetSegmentGlobalRotationEulerXYZ . . . . .	263
GetSegmentLocalTranslation . . . . .	265
GetSegmentLocalRotationHelical . . . . .	267
GetSegmentLocalRotationMatrix . . . . .	269
GetSegmentLocalRotationQuaternion . . . . .	271
GetSegmentLocalRotationEulerXYZ . . . . .	273
SetMaximumPrediction . . . . .	275
MaximumPrediction . . . . .	275

# Developer's Guide

The Vicon DataStream Software Development Kit (SDK) allows easy programmable access to the information contained in the Vicon DataStream. The function calls within the SDK enables users to connect to and request data from the Vicon DataStream. The following combinations of platforms and technologies are distributed:

	Windows x86 (32-bit)	Windows x64 (64-bit)	Linux x64 (64-bit)	Mac OSX (64-bit)
C	YES	YES	YES	YES
C++	YES	YES	YES	YES
.NET	YES	YES		
MATLAB	Use .NET support	Uses .NET support		
Python	YES	YES		

For other platforms, source code is available and you can download it from our website.

Python documentation and usage examples can be found inline in the source as Docstrings.

## Important Notes

- Not all function calls contained within the SDK will return data when connected to certain Vicon applications. For example, Vicon Nexus does not support object quality metrics, and therefore will not output object quality information into the DataStream.
- The current DataStream format is supported by Vicon Nexus 1.4+, Vicon Shogun 1.0+, Vicon Blade 1.6+, and Tracker 1.0+, Evoke 1.0+. These applications may also output an additional stream in the legacy "Tarsus" format. This DataStream SDK only accesses the DataStream format.
- The current intention is that all future Vicon applications will support the DataStream format.
- Example files are supplied as *unsupported* examples only.
- The SDK only supports axis transformations into right-handed coordinate systems.
- The SDK is designed to allow multiple instances of a [Client](#) within a single process, which can connect to multiple DataStreams.
- The SDK is supplied as shared libraries: *DLL* on Windows, *dylib* on OSX and *so* on Linux. The shared libraries and supporting files must be copied alongside your client executable.



## Installing on Windows

There are separate installers for the 32-bit and 64-bit SDKs. The 64-bit installer will only work on a 64-bit version of Windows. The default install directories are:

### 64-bit Windows

- **32-bit SDK** - C:\Program Files (x86)\Vicon\DataStream SDK\Win32
- **64-bit SDK** - C:\Program Files\Vicon\DataStream SDK\Win64

### 32-bit Windows

- **32-bit SDK** - C:\Program Files\Vicon\DataStream SDK\Win32

## C++

Your application must:

- `#include "DataStreamClient.h"`
- Link against `ViconDataStreamSDK_CPP.lib`
- Redistribute:
  - `ViconDataStreamSDK_CPP.dll`
  - `Microsoft.VC141.CRT`
  - `boost_thread-vc140-mt-x{32|64}-1_68.dll`
  - `boost_system-vc140-mt-x{32|64}-1_68.dll`

## .NET

Your application must:

- Link against the assembly `ViconDataStreamSDK_DotNET.dll`.
- Redistribute:
  - `ViconDataStreamSDK_DotNET.dll`
  - `ViconDataStreamSDK_CPP.dll`
  - `Microsoft.VC141.CRT`
  - `boost_thread-vc140-mt-{32|64}-1_68.dll`
  - `boost_system-vc140-mt-{32|64}-1_68.dll`

The managed code in this assembly requires the unmanaged code in the C++ SDK. The .NET dll is built against .NET framework 4.0

## MATLAB

As of DataStream version 1.10, the native MATLAB support has been removed, and use of the .NET DLL from within MATLAB is now the only supported method.

This is valid for versions of MATLAB from 2009a

See `ViconDataStreamSDK_MATLABDotNETTest.m` for an example of how to use the .NET client from within MATLAB

The assembly is loaded with the command

```
% NET.addAssembly(which('ViconDataStreamSDK_DotNET.dll'));
```

Users should note that valid indexes into functions which take a count are 0 to Size-1, rather than the usual MATLAB concept of 1 to Size.

## Installing on Linux

The SDK is provided as a compressed archive. Extract the archive into a convenient location on your system.

## C++

Your application must:

- `#include "DataStreamClient.h"`
- Link against `libViconDataStreamSDK_CPP.so`
- Redistribute `libViconDataStreamSDK_CPP.so`

The 64-bit version of the SDK was compiled with gcc version 7.4 (Ubuntu 18.04).

## Installing on OSX

## C++

Requirements are:

- Intel 64- or 32-bit

Your application must:

- `#include "DataStreamClient.h"`
- Link against `libViconDataStreamSDK_CPP.dylib`

- Redistribute `libViconDataStreamSDK_CPP.dylib`

The SDK was compiled with gcc version 4.2.1 (Apple Inc. Build 5646) using flags:

```
-mmacosx-version-min=10.9 -isysroot /Developer/SDKs/MacOSX10.14.sdk -arch  
i386 -arch x86_64 -O2
```

## Requirements

A compatible licensed version of Vicon Blade, Vicon Shogun, Vicon Nexus, Vicon Tracker or Vicon Evoke must be present.

- LabVIEW uses the .NET dll, and has been found to function in versions 7.1 and 8.
- The MATLAB dll has been found to function in versions up to and including Matlab 2017.
- The MATLAB dll has been found to function in versions up to and including Matlab 2017.
- The SDK has not been designed to allow access from Simulink.
- The Linux SDK has been specifically verified on Ubuntu 16.4. It should also work on any platform supporting glibc 2.5 or later.

## Function Result Return Values

Every function returns a data structure containing elements specified in the 'Output' section of each method reference. Most functions return a 'Result' item, which indicates the success or cause of failure for the function and is useful for debugging purposes.

When a function has returned false, the output arguments are set to an appropriate default value:

- Booleans will be set to false.
- Integers will be set to zero.
- Doubles will be set to zero.
- Strings will be set to zero length.
- When the output argument is an array, all elements are set in this manner.

## Conventions

By default the global coordinate system matches the server application; Z-Up, Y-Left. This can be changed by using [Client::SetAxisMapping](#).

## Units

Positions are expressed in millimeters. Rotation is expressed in radians.

## Vectors and Matrices

Positions are passed as 3 elements corresponding to  $(x, y, z)$

$$\begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix}$$

A 3 matrix is passed row-wise as a vector of 9 elements:

$$\begin{pmatrix} a_0 & a_1 & a_2 \\ a_3 & a_4 & a_5 \\ a_6 & a_7 & a_8 \end{pmatrix}$$

Matrices are assumed to pre-multiply:

$$\mathbf{A} \times \mathbf{B} \times \mathbf{C} = \mathbf{A} \times (\mathbf{B} \times \mathbf{C})$$

## Euler Angles

When used, an XYZ Euler angle  $(x, y, z)$  is constructed:

$$\mathbf{R}_x \times \mathbf{R}_y \times \mathbf{R}_z$$

$$\mathbf{R}_x \times (\mathbf{R}_y \times \mathbf{R}_z)$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos x & -\sin x \\ 0 & \sin x & \cos x \end{pmatrix} \begin{pmatrix} \cos y & 0 & \sin y \\ 0 & 1 & 0 \\ -\sin y & 0 & \cos y \end{pmatrix} \begin{pmatrix} \cos z & -\sin z & 0 \\ \sin z & \cos z & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} \cos y \cos z & -\cos y \sin z & \sin y \\ \cos x \sin z + \sin x \sin y \cos z & \cos x \cos z - \sin x \sin y \sin z & -\sin x \cos y \\ \sin x \sin z - \cos x \sin y \cos z & \sin x \cos z + \cos x \sin y \sin z & \cos x \cos y \end{pmatrix}$$

## Class Index

### Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">Client</a>	Vicon DataStream SDK client . . . . .	7
<a href="#">RetimingClient</a>	The re-timing client class for C++ . . . . .	213

# Class Documentation

## Client Class Reference

### Detailed Description

Vicon DataStream SDK client.

The core client class for C++.

### Public Member Functions

- [Client](#) ()  
*Construction.*
- virtual [~Client](#) ()  
*Destruction.*
- Output\_GetVersion [GetVersion](#) () const  
*Get the version of the Vicon DataStream SDK.*
- Output\_Connect [Connect](#) (const String &HostName)  
*Establish a dedicated connection to a Vicon DataStream Server.*
- Output\_ConnectToMulticast [ConnectToMulticast](#) (const String &LocalIP, const String &MulticastIP)  
*Connect to a Vicon DataStream Server's Multicast stream.*
- Output\_Disconnect [Disconnect](#) ()  
*Disconnect from the Vicon DataStream Server.*
- Output\_IsConnected [IsConnected](#) () const  
*Discover whether client is connected to the Vicon DataStream Server.*
- Output\_StartTransmittingMulticast [StartTransmittingMulticast](#) (const String &ServerIP, const String &MulticastIP)  
*Ask the DataStream Server to start transmitting the data you are receiving directly to a Multicast address as well.*
- Output\_StopTransmittingMulticast [StopTransmittingMulticast](#) ()  
*Ask the DataStream Server to stop transmitting the data you are receiving directly to a Multicast address as well.*
- Output\_EnableSegmentData [EnableSegmentData](#) ()  
*Enable kinematic segment data in the Vicon DataStream.*

- `Output_EnableLightweightSegmentData` [EnableLightweightSegmentData](#) ()  
*Enable a lightweight transmission protocol for kinematic segment data in the Vicon DataStream.*
- `Output_EnableMarkerData` [EnableMarkerData](#) ()  
*Enable labeled reconstructed marker data in the Vicon DataStream.*
- `Output_EnableUnlabeledMarkerData` [EnableUnlabeledMarkerData](#) ()  
*Enable unlabeled reconstructed marker data in the Vicon DataStream.*
- `Output_EnableMarkerRayData` [EnableMarkerRayData](#) ()  
*Enable information about the rays contributing to each labeled marker in the Vicon DataStream.*
- `Output_EnableDeviceData` [EnableDeviceData](#) ()  
*Enable force plate, EMG, and other device data in the Vicon DataStream.*
- `Output_EnableCentroidData` [EnableCentroidData](#) ()  
*Enable centroid data in the Vicon DataStream.*
- `Output_EnableGreyscaleData` [EnableGreyscaleData](#) ()  
*Enable greyscale data in the Vicon DataStream.*
- `Output_EnableVideoData` [EnableVideoData](#) ()  
*Enable video data in the Vicon DataStream.*
- `Output_EnableDebugData` [EnableDebugData](#) ()  
*Enable debug data in the Vicon DataStream.*
- `Output_DisableSegmentData` [DisableSegmentData](#) ()  
*Disable kinematic segment data in the Vicon DataStream.*
- `Output_DisableLightweightSegmentData` [DisableLightweightSegmentData](#) ()  
*Disable the lightweight output mode for kinematic segment data in the Vicon DataStream.*
- `Output_DisableMarkerData` [DisableMarkerData](#) ()  
*Disable labeled reconstructed marker data in the Vicon DataStream.*
- `Output_DisableUnlabeledMarkerData` [DisableUnlabeledMarkerData](#) ()  
*Disable unlabeled reconstructed marker data in the Vicon DataStream.*
- `Output_DisableMarkerRayData` [DisableMarkerRayData](#) ()  
*Disable ray contribution data for markers in the Vicon DataStream.*
- `Output_DisableDeviceData` [DisableDeviceData](#) ()  
*Disable force plate, EMG, and other device data in the Vicon DataStream.*
- `Output_DisableCentroidData` [DisableCentroidData](#) ()  
*Disable centroid data in the Vicon DataStream.*
- `Output_DisableGreyscaleData` [DisableGreyscaleData](#) ()  
*Disable greyscale data in the Vicon DataStream.*
- `Output_DisableVideoData` [DisableVideoData](#) ()  
*Disable video data in the Vicon DataStream.*
- `Output_DisableDebugData` [DisableDebugData](#) ()  
*Disable debug data in the Vicon DataStream.*
- `Output_IsSegmentDataEnabled` [IsSegmentDataEnabled](#) () const  
*Return whether kinematic segment data is enabled in the Vicon DataStream.*
- `Output_IsLightweightSegmentDataEnabled` [IsLightweightSegmentDataEnabled](#) () const  
*Return whether the lightweight transport mode for kinematic segment data is enabled in the Vicon DataStream.*
- `Output_IsMarkerDataEnabled` [IsMarkerDataEnabled](#) () const  
*Return whether labeled reconstructed marker data is enabled in the DataStream.*

- Output\_IsUnlabeledMarkerDataEnabled [IsUnlabeledMarkerDataEnabled](#) () const  
*Return whether unlabeled marker data is enabled in the DataStream.*
- Output\_IsMarkerRayDataEnabled [IsMarkerRayDataEnabled](#) () const  
*Return whether marker ray data is enabled in the DataStream.*
- Output\_IsDeviceDataEnabled [IsDeviceDataEnabled](#) () const  
*Return whether force plate, EMG, and other device data is enabled in the DataStream.*
- Output\_IsCentroidDataEnabled [IsCentroidDataEnabled](#) () const  
*Return whether Centroid data is enabled in the DataStream.*
- Output\_IsGreyscaleDataEnabled [IsGreyscaleDataEnabled](#) () const  
*Return whether greyscale data is enabled in the DataStream.*
- Output\_IsVideoDataEnabled [IsVideoDataEnabled](#) () const  
*Return whether video data is enabled in the DataStream.*
- Output\_IsDebugEnabled [IsDebugEnabled](#) () const  
*Return whether debug data is enabled in the DataStream.*
- void [SetBufferSize](#) (unsigned int BufferSize)  
*Set the number of frames that the client should buffer.*
- Output\_SetStreamMode [SetStreamMode](#) (const **StreamMode::Enum** Mode)  
*There are three modes that the SDK can operate in.*
- Output\_SetApexDeviceFeedback [SetApexDeviceFeedback](#) (const String &DeviceName, bool i\_b-On)  
*Enable haptic feedback for the selected Apex device.*
- Output\_SetAxisMapping [SetAxisMapping](#) (const **Direction::Enum** XAxis, const **Direction::Enum** YAxis, const **Direction::Enum** ZAxis)  
*Remaps the 3D axis.*
- Output\_GetAxisMapping [GetAxisMapping](#) () const  
*Get the current Axis mapping.*
- Output\_GetFrame [GetFrame](#) ()  
*Request a new frame to be fetched from the Vicon DataStream Server.*
- Output\_GetFrameNumber [GetFrameNumber](#) () const  
*Return the number of the last frame retrieved from the DataStream.*
- Output\_GetTimecode [GetTimecode](#) () const  
*Return the timecode information for the last frame retrieved from the DataStream.*
- Output\_GetFrameRate [GetFrameRate](#) () const  
*Return the Vicon camera system frame rate (in Hz) at the time of the last frame retrieved from the DataStream.*
- Output\_GetLatencySampleCount [GetLatencySampleCount](#) () const  
*Return the number of latency measurements that were taken at various stages of the real-time pipeline.*
- Output\_GetLatencySampleName [GetLatencySampleName](#) (const unsigned int LatencySample-Index) const  
*Return the name of a latency sample.*
- Output\_GetLatencySampleValue [GetLatencySampleValue](#) (const String &LatencySampleName) const  
*Return the duration of a named latency sample in seconds.*
- Output\_GetLatencyTotal [GetLatencyTotal](#) () const  
*Return the total latency in seconds introduced at various stages of the real-time pipeline.*



- Output\_GetHardwareFrameNumber [GetHardwareFrameNumber](#) () const  
*Returns the hardware frame number as used by the cameras.*
- Output\_GetFrameRateCount [GetFrameRateCount](#) () const  
*Get the number of frame rate types that the server application reports.*
- Output\_GetFrameRateName [GetFrameRateName](#) (const unsigned int FrameRateIndex) const  
*Get the name of a frame rate type at the specified index.*
- Output\_GetFrameRateValue [GetFrameRateValue](#) (const String &FrameRateName) const  
*Get the current value of the specified frame rate type.*
- Output\_GetSubjectCount [GetSubjectCount](#) () const  
*Return the number of subjects in the DataStream.*
- Output\_GetSubjectName [GetSubjectName](#) (const unsigned int SubjectIndex) const  
*Return the name of a subject.*
- Output\_GetSubjectRootSegmentName [GetSubjectRootSegmentName](#) (const String &SubjectName) const  
*Return the name of the root segment for a specified subject.*
- Output\_GetSegmentCount [GetSegmentCount](#) (const String &SubjectName) const  
*Return the number of segments for a specified subject in the DataStream.*
- Output\_GetSegmentName [GetSegmentName](#) (const String &SubjectName, const unsigned int SegmentIndex) const  
*Return the name of a subject segment specified by index.*
- Output\_GetSegmentChildCount [GetSegmentChildCount](#) (const String &SubjectName, const String &SegmentName) const  
*Return the number of child segments for a specified subject segment.*
- Output\_GetSegmentChildName [GetSegmentChildName](#) (const String &SubjectName, const String &SegmentName, const unsigned int SegmentIndex) const  
*Return the name of the child segment for a specified subject segment and index.*
- Output\_GetSegmentParentName [GetSegmentParentName](#) (const String &SubjectName, const String &SegmentName) const  
*Return the name of the parent segment for a specified subject segment.*
- Output\_GetSegmentStaticTranslation [GetSegmentStaticTranslation](#) (const String &SubjectName, const String &SegmentName) const  
*Return the static pose translation of a subject segment.*
- Output\_GetSegmentStaticRotationHelical [GetSegmentStaticRotationHelical](#) (const String &SubjectName, const String &SegmentName) const  
*Return the static pose rotation of a subject segment in helical coordinates.*
- Output\_GetSegmentStaticRotationMatrix [GetSegmentStaticRotationMatrix](#) (const String &SubjectName, const String &SegmentName) const  
*Return the static pose rotation of a subject segment as a 3x3 row-major matrix.*
- Output\_GetSegmentStaticRotationQuaternion [GetSegmentStaticRotationQuaternion](#) (const String &SubjectName, const String &SegmentName) const  
*Return the static pose rotation of a subject segment in quaternion coordinates.*
- Output\_GetSegmentStaticRotationEulerXYZ [GetSegmentStaticRotationEulerXYZ](#) (const String &SubjectName, const String &SegmentName) const  
*Return the static pose rotation of a subject segment in Euler XYZ coordinates.*
- Output\_GetSegmentStaticScale [GetSegmentStaticScale](#) (const String &SubjectName, const String &SegmentName) const

*Return a 3D Scale of a subject segment if present.*

- Output\_GetSegmentGlobalTranslation [GetSegmentGlobalTranslation](#) (const String &SubjectName, const String &SegmentName) const

*Return the translation of a subject segment in global coordinates.*

- Output\_GetSegmentGlobalRotationHelical [GetSegmentGlobalRotationHelical](#) (const String &SubjectName, const String &SegmentName) const

*Return the rotation of a subject segment in global helical coordinates.*

- Output\_GetSegmentGlobalRotationMatrix [GetSegmentGlobalRotationMatrix](#) (const String &SubjectName, const String &SegmentName) const

*Return the rotation of a subject segment as a 3x3 row-major matrix in global coordinates.*

- Output\_GetSegmentGlobalRotationQuaternion [GetSegmentGlobalRotationQuaternion](#) (const String &SubjectName, const String &SegmentName) const

*Return the rotation of a subject segment in global quaternion coordinates.*

- Output\_GetSegmentGlobalRotationEulerXYZ [GetSegmentGlobalRotationEulerXYZ](#) (const String &SubjectName, const String &SegmentName) const

*Return the rotation of a subject segment in global Euler XYZ coordinates.*

- Output\_GetSegmentLocalTranslation [GetSegmentLocalTranslation](#) (const String &SubjectName, const String &SegmentName) const

*Return the translation of a subject segment in local coordinates relative to its parent segment.*

- Output\_GetSegmentLocalRotationHelical [GetSegmentLocalRotationHelical](#) (const String &SubjectName, const String &SegmentName) const

*Return the rotation of a subject segment in local helical coordinates relative to its parent segment.*

- Output\_GetSegmentLocalRotationMatrix [GetSegmentLocalRotationMatrix](#) (const String &SubjectName, const String &SegmentName) const

*Return the rotation row-major matrix of a subject segment in local coordinates relative to its parent segment.*

- Output\_GetSegmentLocalRotationQuaternion [GetSegmentLocalRotationQuaternion](#) (const String &SubjectName, const String &SegmentName) const

*Return the rotation of a subject segment in local quaternion coordinates relative to its parent segment.*

- Output\_GetSegmentLocalRotationEulerXYZ [GetSegmentLocalRotationEulerXYZ](#) (const String &SubjectName, const String &SegmentName) const

*Return the rotation of a subject segment in local Euler XYZ coordinates relative to its parent segment.*

- Output\_GetObjectQuality [GetObjectQuality](#) (const String &ObjectName) const

*Return the quality score for a specified Object (Subject).*

- Output\_GetMarkerCount [GetMarkerCount](#) (const String &SubjectName) const

*Return the number of markers for a specified subject in the DataStream.*

- Output\_GetMarkerName [GetMarkerName](#) (const String &SubjectName, const unsigned int MarkerIndex) const

*Return the name of a marker for a specified subject.*

- Output\_GetMarkerParentName [GetMarkerParentName](#) (const String &SubjectName, const String &MarkerName) const

*Return the name of the segment that is the parent of this marker.*

- Output\_GetMarkerGlobalTranslation [GetMarkerGlobalTranslation](#) (const String &SubjectName, const String &MarkerName) const

*Return the translation of a subject marker in global coordinates.*

- Output\_GetMarkerRayContributionCount [GetMarkerRayContributionCount](#) (const String &SubjectName, const String &MarkerName) const

*Return the number of rays that are contributing to a labeled marker in the DataStream.*

- Output\_GetMarkerRayContribution [GetMarkerRayContribution](#) (const String &SubjectName, const String &MarkerName, unsigned int MarkerRayContributionIndex) const

*Return the camera ID for an indexed ray that is contributing to a labeled marker in the DataStream.*

- Output\_GetUnlabeledMarkerCount [GetUnlabeledMarkerCount](#) () const

*Return the number of unlabeled markers in the DataStream.*

- Output\_GetUnlabeledMarkerGlobalTranslation [GetUnlabeledMarkerGlobalTranslation](#) (const unsigned int MarkerIndex) const

*Return the translation of an unlabeled marker in global coordinates.*

- Output\_GetLabeledMarkerCount [GetLabeledMarkerCount](#) () const

*Returns the number of all labeled markers in the datastream across all subjects.*

- Output\_GetLabeledMarkerGlobalTranslation [GetLabeledMarkerGlobalTranslation](#) (const unsigned int MarkerIndex) const

*Return the translation of a labeled marker in global coordinates.*

- Output\_GetDeviceCount [GetDeviceCount](#) () const

*Return the number of force plates, EMGs, and other devices in the DataStream.*

- Output\_GetDeviceName [GetDeviceName](#) (const unsigned int DeviceIndex) const

*Return the name and type of a device.*

- Output\_GetDeviceOutputCount [GetDeviceOutputCount](#) (const String &DeviceName) const

*Return the number of outputs for a device in the DataStream.*

- Output\_GetDeviceOutputName [GetDeviceOutputName](#) (const String &DeviceName, const unsigned int DeviceOutputIndex) const

*Return the name and SI unit of a device output.*

- Output\_GetDeviceOutputComponentName [GetDeviceOutputComponentName](#) (const String &DeviceName, const unsigned int DeviceOutputIndex) const

*Return the name of the output and component and SI unit of a device output.*

- Output\_GetDeviceOutputValue [GetDeviceOutputValue](#) (const String &DeviceName, const String &DeviceOutputComponentName) const

*Return the value of a device output.*

- Output\_GetDeviceOutputValue [GetDeviceOutputValue](#) (const String &DeviceName, const String &DeviceOutputName, const String &DeviceOutputComponentName) const

*Return the value of a device output.*

- Output\_GetDeviceOutputSubsamples [GetDeviceOutputSubsamples](#) (const String &DeviceName, const String &DeviceOutputName) const

*Return the number of samples available for the specified device at the current frame.*

- Output\_GetDeviceOutputSubsamples [GetDeviceOutputSubsamples](#) (const String &DeviceName, const String &DeviceOutputName, const String &DeviceOutputComponentName) const

*Return the number of samples available for the specified device at the current frame.*

- Output\_GetDeviceOutputValue [GetDeviceOutputValue](#) (const String &DeviceName, const String &DeviceOutputName, const unsigned int Subsample) const

*Return the value of a device output.*

- Output\_GetDeviceOutputValue [GetDeviceOutputValue](#) (const String &DeviceName, const String &DeviceOutputName, const String &DeviceOutputComponentName, const unsigned int Subsample) const

*Return the value of a device output.*

- Output\_GetForcePlateCount [GetForcePlateCount](#) () const

*Return the number of force plates available in the DataStream.*

- Output\_GetGlobalForceVector [GetGlobalForceVector](#) (const unsigned int ForcePlateIndex) const

*Return the force vector for the force plate in global coordinates.*

- Output\_GetGlobalMomentVector [GetGlobalMomentVector](#) (const unsigned int ForcePlateIndex) const

*Return the moment vector for the force plate in global coordinates.*

- Output\_GetGlobalCentreOfPressure [GetGlobalCentreOfPressure](#) (const unsigned int ForcePlateIndex) const

*Return the center of pressure for the force plate in global coordinates.*

- Output\_GetForcePlateSubsamples [GetForcePlateSubsamples](#) (const unsigned int ForcePlateIndex) const

*Return the number of subsamples available for a specified force plate in the current frame.*

- Output\_GetGlobalForceVector [GetGlobalForceVector](#) (const unsigned int ForcePlateIndex, const unsigned int Subsample) const

*Return the force vector for the force plate in global coordinates.*

- Output\_GetGlobalMomentVector [GetGlobalMomentVector](#) (const unsigned int ForcePlateIndex, const unsigned int Subsample) const

*Return the moment vector for the force plate in global coordinates.*

- Output\_GetGlobalCentreOfPressure [GetGlobalCentreOfPressure](#) (const unsigned int ForcePlateIndex, const unsigned int Subsample) const

*Return the center of pressure for the force plate in global coordinates.*

- Output\_GetEyeTrackerCount [GetEyeTrackerCount](#) () const

*Return the number of eye trackers available in the DataStream.*

- Output\_GetEyeTrackerGlobalPosition [GetEyeTrackerGlobalPosition](#) (const unsigned int EyeTrackerIndex) const

*Return the location of the eye.*

- Output\_GetEyeTrackerGlobalGazeVector [GetEyeTrackerGlobalGazeVector](#) (const unsigned int EyeTrackerIndex) const

*Return the gaze direction as a unit vector in global coordinates.*

- Output\_GetCameraCount [GetCameraCount](#) () const

*Return the number of cameras available in the DataStream.*

- Output\_GetCameraName [GetCameraName](#) (unsigned int CameraIndex) const

*Return the name of a camera.*

- Output\_GetCameraId [GetCameraId](#) (const std::string &CameraName) const

*Returns the internal ID of the camera with the specified name.*

- Output\_GetCameraUserId [GetCameraUserId](#) (const std::string &CameraName) const

*Returns the user-assigned ID of the camera with the specified name.*

- Output\_GetCameraType [GetCameraType](#) (const std::string &CameraName) const

*Returns the type of the camera with the specified name.*

- Output\_GetCameraDisplayName [GetCameraDisplayName](#) (const std::string &CameraName) const

*Returns the name of of the camera type as a string suitable for display to a user.*

- Output\_GetCameraResolution [GetCameraResolution](#) (const std::string &CameraName) const

*Returns the sensor resolution of the camera with the specified name.*

- Output\_GetIsVideoCamera [GetIsVideoCamera](#) (const std::string &CameraName) const

*Returns whether the camera with the specified name is a video camera.*

- Output\_GetCentroidCount [GetCentroidCount](#) (const std::string &CameraName) const

*Return the number of centroids reported by a named camera.*

- Output\_GetCentroidPosition [GetCentroidPosition](#) (const std::string &CameraName, const unsigned int CentroidIndex) const

*Return the position and radius of the centroid in camera coordinates.*

- Output\_GetCentroidWeight [GetCentroidWeight](#) (const std::string &CameraName, const unsigned int CentroidIndex) const

*Return the weight of the centroid.*

- Output\_GetGreyscaleBlobCount [GetGreyscaleBlobCount](#) (const std::string &CameraName) const

*Obtain the number of greyscale blobs that are available for the specified camera.*

- Output\_GetGreyscaleBlob [GetGreyscaleBlob](#) (const std::string &CameraName, const unsigned int i\_BlobIndex) const

*Obtains greyscale blob data for the specified camera and blob index.*

- Output\_GetVideoFrame [GetVideoFrame](#) (const std::string &CameraName) const

*Obtains video data for the specified camera.*

- Output\_SetCameraFilter [SetCameraFilter](#) (const std::vector< unsigned int > &CameraIdsForCentroids, const std::vector< unsigned int > &CameraIdsForBlobs, const std::vector< unsigned int > &CameraIdsForVideo)

*Add a filter to allow centroid, blob or video data to be transmitted for the specified cameras only.*

- Output\_ClearSubjectFilter [ClearSubjectFilter](#) ()

*Clear the subject filter.*

- Output\_AddToSubjectFilter [AddToSubjectFilter](#) (const String &SubjectName)

*Add a subject name to the subject filter.*

- virtual Output\_ConfigureWireless [ConfigureWireless](#) ()

*Request that the wireless adapters will be optimally configured for streaming data.*

## Constructor & Destructor Documentation

### Client ( )

Construction.

You can create many instances of the Vicon DataStream [Client](#) which can connect to multiple Vicon DataStream Servers.

C example

```
// The C version uses explicit creation methods

CClient * pClient = ClientCreate();
// C Client functions take the client as a parameter
CBool ok = Client_SomeFunction( pClient, Args );
// The C client needs to be explicitly destroyed
Client_Destroy( pClient );
```

C++ example

## Class Documentation

```
// C++ version of the SDK is object oriented, so use the class constructor.

ViconDataStreamSDK::CPP::Client StackClient;
Output_SomeFunction Output = StackClient.SomeFunction();
// Client is implicitly destroyed as it goes out of scope.

// Alternatively the Client can be made on the heap.

ViconDataStreamSDK::CPP::Client * pHeapClient
= new ViconDataStreamSDK::CPP::Client();
Output_SomeFunction Output = pHeapClient->SomeFunction( Input );
delete pHeapClient;
```

### MATLAB example

```
% MATLAB uses the .NET SDK.

dssdkAssembly = which('ViconDataStreamSDK_DotNET.dll');
NET.addAssembly(dssdkAssembly);
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.SomeFunction( Input );

%% There is no method to unload the assembly. Restart MATLAB to free up.
%% https://uk.mathworks.com/matlabcentral/answers/71198-net-assembly-unload-conundrum
```

### .NET example

```
///.NET is object oriented, so use the class constructor. Because objects are
// lazily garbage collected, your instance may outlive the last reference to it
// for some time.If the instance is pre-fetching frame data for you, then it
// can still use CPU and network bandwidth.Consider explicitly disconnecting
// prior to destruction.

ViconDataStreamSDK.DotNET.Client pHeapClient = new ViconDataStreamSDK.DotNET.Client();
Output_SomeFunction Output = pHeapClient.SomeFunction(InputParam);
// Signal to the garbage collector that it can clean up pHeapClient.Disconnect();
pHeapClient = null;
```

## ~Client ( ) [virtual]

Destruction.

Destruction will Disconnect if required.

See [Client::Client](#) for an example.

## Member Function Documentation

### Output\_GetVersion GetVersion ( ) const

Get the version of the Vicon DataStream SDK.

- **Major** When this number increases, we break backward compatibility with previous major versions.
- **Minor** When this number increases, we have probably added new functionality to the SDK without breaking backward compatibility with previous versions.

- **Point** When this number increases, we have introduced a bug fix or performance enhancement without breaking backward compatibility with previous versions.

The function can be called without the client being connected.

### C example

```
CClient * pClient = Client_Create();
COutput_GetVersion Output = Client_GetVersion( pClient );
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
Output_GetVersion Output = MyClient.GetVersion();
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
Output = MyClient.GetVersion();
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
Output_GetVersion Output = MyClient.GetVersion();
```

### Returns

Output\_GetVersion class containing the version information.



## Output\_Connect Connect ( const String & *HostName* )

Establish a dedicated connection to a Vicon DataStream Server.

See Also: [ConnectToMulticast\(\)](#), [Disconnect\(\)](#), [IsConnected\(\)](#).

The function defaults to connecting on port 801. You can specify an alternate port number after a colon. This is for future compatibility: current products serve data on port 801 only.

Additional clients can be added separated with a semicolon ';'. These are used in combination to reduce temporal jitter.

### C example

```
CClient * pClient = Client_Create();
COutput_Connect Output = Client_Connect( pClient, "localhost");
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
Output_Connect Output = MyClient.Connect( "localhost" );
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
Output = MyClient.Connect('localhost:801');
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
Output_Connect Output = MyClient.Connect("localhost:801");
```

### Parameters

<i>HostName</i>	The DNS-identifiable name, or IP address of the PC hosting the DataStream server. For example: <ul style="list-style-type: none"> <li>"localhost"</li> <li>"MyViconPC:801"</li> <li>"10.0.0.2"</li> </ul>
-----------------	--

### Returns

An Output\_Connect class containing the result of the connect operation.

- The Result will be:
  - Success
  - InvalidHostName
  - ClientAlreadyConnected
  - ClientConnectionFailed



## Output\_ConnectToMulticast ConnectToMulticast ( const String & *LocalIP*, const String & *MulticastIP* )

Connect to a Vicon DataStream Server's Multicast stream.

The stream content is managed by a client who calls [StartTransmittingMulticast\(\)](#).

See Also: [Connect\(\)](#), [Disconnect\(\)](#), [IsConnected\(\)](#), [StartTransmittingMulticast\(\)](#), [StopTransmittingMulticast\(\)](#)

```
// class Output_ConnectToMulticast
// {
//     public:
//         Result::Enum Result;
// };
```

### C example

```
CClient * pClient = Client_Create();
COutput_Connect Output = Client_ConnectToMulticast( pClient, "localhost", "224.0.0.0" );
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
Output_ConnectToMulticast Output = MyClient.ConnectToMulticast( "localhost", "224.0.0.0" );
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
Output = MyClient.ConnectToMulticast('localhost', '224.0.0.0');
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
Output_ConnectToMulticast Output = MyClient.ConnectToMulticast("localhost", "224.0.0.0");
```

### Parameters

<i>LocalIP</i>	The DNS-identifiable name, or IP address of the local Ethernet interface on which you wish to receive multicast data. Do not specify a port (any port specified will be ignored). For example: <ul style="list-style-type: none"> <li>"localhost"</li> <li>"10.0.0.2"</li> </ul>
<i>MulticastIP</i>	The IP Address of the Multicast group on which data will be received. The address must be in the range 224.0.0.0-239.255.255.255 You may also specify a port by appending it to the end of the IP Address after a colon, e.g. 224.0.0.0:30001. If you do not specify a port it will default to 44801.

## Returns

An Output\_ConnectToMulticast class containing the result of the connect operation.

- The Result will be:
  - Success
  - InvalidHostName
  - InvalidMulticastIP
  - ClientAlreadyConnected
  - ClientConnectionFailed

## Output\_Disconnect Disconnect ( )

Disconnect from the Vicon DataStream Server.

See Also: [Connect\(\)](#), [IsConnected\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
COutput_Disconnect Output = Client_Disconnect( pClient );
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Output_Disconnect Output = MyClient.Disconnect();
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect("localhost");
Output = MyClient.Disconnect()
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect("localhost");
Output_Disconnect Output = MyClient.Disconnect();
```

### Returns

An Output\_Disconnect class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected

## Output\_IsConnected IsConnected ( ) const

Discover whether client is connected to the Vicon DataStream Server.

See Also: [Connect\(\)](#), [Disconnect\(\)](#)

### C example

```
CClient * pClient = Client_Create();
CBool Output = Client_IsConnected( pClient );
// Output == 0
Client_Connect( pClient, "localhost" );
Output = Client_IsConnected( pClient );
// Output == 1
COutput_Disconnect Output = Client_Disconnect( pClient );
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
Output_IsConnected Output = MyClient.IsConnected()
// Output.Connected == false
MyClient.Connect( "localhost" );
Output_IsConnected Output = MyClient.IsConnected()
// Output.Connected == true
// (assuming localhost is serving)
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
Output_IsConnected Output = MyClient.IsConnected()
// Output.Connected == false
MyClient.Connect( "localhost" );
Output_IsConnected Output = MyClient.IsConnected()
// Output.Connected == true
// (assuming localhost is serving)
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
Output_IsConnected Output = MyClient.IsConnected()
// Output.Connected == false
MyClient.Connect( "localhost" );
Output_IsConnected Output = MyClient.IsConnected()
// Output.Connected == true
// (assuming localhost is serving)
```

### Returns

An `Output_IsConnected` class containing a true value for `Connected` if you are connected to the stream, otherwise false.

## Output\_StartTransmittingMulticast StartTransmittingMulticast ( const String & ServerIP, const String & MulticastIP )

Ask the DataStream Server to start transmitting the data you are receiving directly to a Multicast address as well.

This allows multiple clients to connect to your stream (via [ConnectToMulticast\(\)](#) ) whilst minimizing network bandwidth use and frame delivery latency.

See Also: [Connect\(\)](#), [ConnectToMulticast\(\)](#), [Disconnect\(\)](#), [StopTransmittingMulticast\(\)](#) C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_StartTransmittingMulticast( pClient, "10.0.0.1", "224.0.0.0" );
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.StartTransmittingMulticast( "10.0.0.1", "224.0.0.0" );
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.StartTransmittingMulticast( "10.0.0.1", "224.0.0.0" );
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.StartTransmittingMulticast( "10.0.0.1", "224.0.0.0" );
```

### Parameters

<i>ServerIP</i>	The IP Address of the server Ethernet interface from which the Multicast data will be sent. Do not specify a port number (any port number specified will be ignored).
<i>MulticastIP</i>	The IP Address of the Multicast group to which Multicast data will be sent. The address must be in the range 224.0.0.0-239.255.255.255. You may also specify the port the data will be sent to by appending it to the IP Address after a colon, e.g. 224.0.0.0:30001. If you do not specify a port it will default to 44801.

### Returns

An Output\_StartTransmittingMulticast class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected
  - InvalidMulticastIP
  - ServerAlreadyTransmittingMulticast

## Output\_StopTransmittingMulticast StopTransmittingMulticast ( )

Ask the DataStream Server to stop transmitting the data you are receiving directly to a Multicast address as well.

You must previously have started a transmission via StartTransmittingMulticast.

See Also: [Connect\(\)](#), [ConnectToMulticast\(\)](#), [Disconnect\(\)](#), [StartTransmittingMulticast\(\)](#) C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_StartTransmittingMulticast( pClient, "10.0.0.1", "224.0.0.0" );
// Do some stuff
Client_StopTransmittingMulticast( pClient );
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.StartTransmittingMulticast( "10.0.0.1", "224.0.0.0" );
// Do some stuff
MyClient.StopTransmittingMulticast();
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.StartTransmittingMulticast( "10.0.0.1", "224.0.0.0" );
// Do some stuff
MyClient.StopTransmittingMulticast();
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.StartTransmittingMulticast( "10.0.0.1", "224.0.0.0" );
// Do some stuff
MyClient.StopTransmittingMulticast();
```

### Returns

An Output\_StopTransmittingMulticast class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected
  - ServerNotTransmittingMulticast

## Output\_EnableSegmentData EnableSegmentData ( )

Enable kinematic segment data in the Vicon DataStream.

Call this function on startup, after connecting to the server, and before trying to read local or global segment data.

See Also: [IsSegmentDataEnabled\(\)](#), [DisableSegmentData\(\)](#), [EnableMarkerData\(\)](#), [EnableUnlabeled-MarkerData\(\)](#), [EnableDeviceData\(\)](#), [GetSegmentCount\(\)](#), [GetSegmentName\(\)](#), [GetSegmentGlobal-Translation\(\)](#), [GetSegmentGlobalRotationEulerXYZ\(\)](#), [GetSegmentLocalTranslation\(\)](#), [GetSegment-LocalRotationEulerXYZ\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableSegmentData();
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Output_EnableSegmentData Output = MyClient.EnableSegmentData();
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output_EnableSegmentData Output = MyClient.EnableSegmentData();
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output_EnableSegmentData Output = MyClient.EnableSegmentData();
```

### Returns

An `Output_EnableSegmentData` class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected

## Output\_EnableLightweightSegmentData EnableLightweightSegmentData ( )

Enable a lightweight transmission protocol for kinematic segment data in the Vicon DataStream.

This will reduce the network bandwidth required to transmit segment data to approximately a quarter of that required by the previous method, at the expense of a small amount of precision. Use the existing methods such as [GetSegmentGlobalTranslation\(\)](#) and [GetSegmentGlobalRotationMatrix\(\)](#) as usual to obtain the segment data. Calling this method will automatically disable all other configurable output types. These may be re-enabled after the call if required.

Call this function on startup, after connecting to the server, and before trying to read local or global segment data.

See Also: [IsSegmentDataEnabled\(\)](#), [DisableSegmentData\(\)](#), [EnableMarkerData\(\)](#), [EnableUnlabeledMarkerData\(\)](#), [EnableDeviceData\(\)](#), [GetSegmentCount\(\)](#), [GetSegmentName\(\)](#), [GetSegmentGlobalTranslation\(\)](#), [GetSegmentGlobalRotationEulerXYZ\(\)](#), [GetSegmentLocalTranslation\(\)](#), [GetSegmentLocalRotationEulerXYZ\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableLightweightSegmentData();
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Output_EnableLightweightSegmentData Output = MyClient.EnableLightweightSegmentData();
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output_EnableLightweightSegmentData Output = MyClient.EnableLightweightSegmentData();
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output_EnableLightweightSegmentData Output = MyClient.EnableLightweightSegmentData();
```

### Returns

An [Output\\_EnableSegmentData](#) class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected



## Output\_EnableMarkerData EnableMarkerData ( )

Enable labeled reconstructed marker data in the Vicon DataStream.

Call this function on startup, after connecting to the server, and before trying to read labeled marker data.

See Also: [IsMarkerDataEnabled\(\)](#), [DisableMarkerData\(\)](#), [EnableSegmentData\(\)](#), [EnableUnlabeled-MarkerData\(\)](#), [EnableDeviceData\(\)](#), [GetMarkerCount\(\)](#), [GetMarkerName\(\)](#), [GetMarkerGlobal-Translation\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableMarkerData();
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Output_EnableMarkerData Output = MyClient.EnableMarkerData();
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output_EnableMarkerData Output = MyClient.EnableMarkerData();
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output_EnableMarkerData Output = MyClient.EnableMarkerData();
```

### Returns

An Output\_EnableMarkerData class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected

## Output\_EnableUnlabeledMarkerData EnableUnlabeledMarkerData ( )

Enable unlabeled reconstructed marker data in the Vicon DataStream.

Call this function on startup, after connecting to the server, and before trying to read global unlabeled marker data.

See Also: [IsUnlabeledMarkerDataEnabled\(\)](#), [DisableUnlabeledMarkerData\(\)](#), [EnableSegmentData\(\)](#), [EnableMarkerData\(\)](#), [EnableDeviceData\(\)](#), [GetUnlabeledMarkerCount\(\)](#), [GetUnlabeledMarkerGlobal-Translation\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableUnlabeledMarkerData();
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Output_EnableUnlabeledMarkerData Output = MyClient.EnableUnlabeledMarkerData();
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output_EnableUnlabeledMarkerData Output = MyClient.EnableUnlabeledMarkerData();
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output_EnableUnlabeledMarkerData Output = MyClient.EnableUnlabeledMarkerData();
```

### Returns

An Output\_UnlabeledMarkerData class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected

## ViconDataStreamSDK::CPP::Output\_EnableMarkerRayData EnableMarkerRayData ( )

Enable information about the rays contributing to each labeled marker in the Vicon DataStream.

Call this function on startup, after connecting to the server, and before trying to read global unlabeled marker data.

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableMarkerRayData();
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Output_EnableMarkerRayData Output = MyClient.EnableMarkerRayData();
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output = MyClient.EnableMarkerRayData();
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output_EnableMarkerRayData Output = MyClient.EnableMarkerRayData();
```

See Also: [IsMarkerRayDataEnabled\(\)](#), [DisableMarkerRayData\(\)](#), [EnableSegmentData\(\)](#), [EnableMarkerData\(\)](#), [EnableDeviceData\(\)](#), [GetUnlabeledMarkerCount\(\)](#), [GetUnlabeledMarkerGlobalTranslation\(\)](#)

### Returns

An Output\_EnableMarkerRayData class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected

## Output\_EnableDeviceData EnableDeviceData ( )

Enable force plate, EMG, and other device data in the Vicon DataStream.

Call this function on startup, after connecting to the server, and before trying to read device information.

See Also: [IsDeviceDataEnabled\(\)](#), [DisableDeviceData\(\)](#), [EnableSegmentData\(\)](#), [EnableMarkerData\(\)](#), [EnableUnlabeledMarkerData\(\)](#), [GetDeviceCount\(\)](#), [GetDeviceName\(\)](#), [GetDeviceOutputCount\(\)](#), [GetDeviceOutputName\(\)](#), [GetDeviceOutputValue\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableDeviceData();
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Output_EnableDeviceData Output = MyClient.EnableDeviceData();
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output = MyClient.EnableDeviceData();
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output_EnableDeviceData Output = MyClient.EnableDeviceData();
```

### Returns

An `Output_EnableDeviceData` class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected

## Output\_EnableCentroidData EnableCentroidData ( )

Enable centroid data in the Vicon DataStream.

Call this function on startup, after connecting to the server, and before trying to read centroid information.

See Also: [IsCentroidDataEnabled\(\)](#), [DisableCentroidData\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableCentroidData();
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Output_EnableCentroidData Output = MyClient.EnableCentroidData();
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output = MyClient.EnableCentroidData();
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output_EnableCentroidData Output = MyClient.EnableCentroidData();
```

### Returns

An `Output_EnableCentroidData` class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected

## Output\_EnableGreyscaleData EnableGreyscaleData ( )

Enable greyscale data in the Vicon DataStream.

Call this function on startup, after connecting to the server, and before trying to read greyscale information.

See Also: [IsGreyscaleDataEnabled\(\)](#), [DisableGreyscaleData\(\)](#)

C example

Not implemented

C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;  
MyClient.Connect( "localhost" );  
Output_EnableGreyscaleData Output = MyClient.EnableGreyscaleData();
```

MATLAB example

Not implemented

.NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();  
MyClient.Connect( "localhost" );  
Output_EnableGreyscaleData Output = MyClient.EnableGreyscaleData ();
```

### Returns

An Output\_EnableGreyscaleData class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected

## Output\_EnableVideoData EnableVideoData ( )

Enable video data in the Vicon DataStream.

Call this function on startup, after connecting to the server, and before trying to read video information.

See Also: [IsVideoDataEnabled\(\)](#), [DisableVideoData\(\)](#)

C example

Not implemented

C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;  
MyClient.Connect( "localhost" );  
Output_EnableVideoData Output = MyClient.EnableVideoData();
```

MATLAB example

Not implemented

.NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();  
MyClient.Connect( "localhost" );  
Output_EnableVideoData Output = MyClient.EnableVideoData ();
```

### Returns

An Output\_EnableVideoData class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected

## Output\_EnableDebugData EnableDebugData ( )

Enable debug data in the Vicon DataStream.

In order to receive debug data, call this function on startup, after connecting to the server.

See Also: [IsDebugDataEnabled\(\)](#), [DisableDebugData\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableDebugData();
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Output_EnableDebugData Output = MyClient.EnableDebugData();
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output = MyClient.EnableDebugData ();
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output_EnableDebugData Output = MyClient.EnableDebugData ();
```

### Returns

An `Output_EnableDebugData` class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected



## Output\_DisableSegmentData DisableSegmentData ( )

Disable kinematic segment data in the Vicon DataStream.

See Also: [IsSegmentDataEnabled\(\)](#), [EnableSegmentData\(\)](#), [EnableMarkerData\(\)](#), [EnableUnlabeled-MarkerData\(\)](#), [EnableDeviceData\(\)](#), [GetSegmentCount\(\)](#), [GetSegmentName\(\)](#), [GetSegmentGlobal-Translation\(\)](#), [GetSegmentGlobalRotationEulerXYZ\(\)](#), [GetSegmentLocalTranslation\(\)](#), [GetSegment-LocalRotationEulerXYZ\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_DisableSegmentData();
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Output_DisableSegmentData Output = MyClient.DisableSegmentData();
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output = MyClient.DisableSegmentData();
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output_DisableSegmentData Output = MyClient.DisableSegmentData();
```

### Returns

An `Output_DisableSegmentData` class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected

## Output\_DisableLightweightSegmentData DisableLightweightSegmentData ( )

Disable the lightweight output mode for kinematic segment data in the Vicon DataStream.

Calling this mode does not automatically enable any other data types.

See Also: [IsSegmentDataEnabled\(\)](#), [EnableSegmentData\(\)](#), [EnableMarkerData\(\)](#), [EnableUnlabeled-MarkerData\(\)](#), [EnableDeviceData\(\)](#), [GetSegmentCount\(\)](#), [GetSegmentName\(\)](#), [GetSegmentGlobal-Translation\(\)](#), [GetSegmentGlobalRotationEulerXYZ\(\)](#), [GetSegmentLocalTranslation\(\)](#), [GetSegment-LocalRotationEulerXYZ\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_DisableLightweightSegmentData();
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Output_DisableLightweightSegmentData Output = MyClient.DisableLightweightSegmentData();
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output = MyClient.DisableLightweightSegmentData ();
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output_DisableLightweightSegmentData Output = MyClient.DisableLightweightSegmentData ();
```

### Returns

An `Output_DisableLightweightSegmentData` class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected

## Output\_DisableMarkerData DisableMarkerData ( )

Disable labeled reconstructed marker data in the Vicon DataStream.

See Also: [IsMarkerDataEnabled\(\)](#), [EnableMarkerData\(\)](#), [EnableSegmentData\(\)](#), [EnableUnlabeledMarkerData\(\)](#), [EnableDeviceData\(\)](#), [GetMarkerCount\(\)](#), [GetMarkerName\(\)](#), [GetMarkerGlobalTranslation\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_DisableMarkerData();
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Output_DisableMarkerData Output = MyClient.DisableMarkerData();
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output = MyClient.DisableMarkerData ();
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output_DisableMarkerData Output = MyClient.DisableMarkerData ();
```

### Returns

An `Output_DisableMarkerData` class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected

## Output\_DisableUnlabeledMarkerData DisableUnlabeledMarkerData ( )

Disable unlabeled reconstructed marker data in the Vicon DataStream.

See Also: [IsUnlabeledMarkerDataEnabled\(\)](#), [EnableUnlabeledMarkerData\(\)](#), [EnableSegmentData\(\)](#), [EnableMarkerData\(\)](#), [EnableDeviceData\(\)](#), [GetUnlabeledMarkerCount\(\)](#), [GetUnlabeledMarkerGlobal-Translation\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_DisableUnlabeledMarkerData();
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Output_DisableUnlabeledMarkerData Output = MyClient.DisableUnlabeledMarkerData();
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output = MyClient.DisableUnlabeledMarkerData ();
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output_DisableUnlabeledMarkerData Output = MyClient.DisableUnlabeledMarkerData ();
```

### Returns

An `Output_DisableUnlabeledMarkerData` class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected

## ViconDataStreamSDK::CPP::Output\_DisableMarkerRayData DisableMarkerRayData ( )

Disable ray contribution data for markers in the Vicon DataStream.

See Also: [IsMarkerRayDataEnabled\(\)](#), [EnableMarkerRayData\(\)](#), [EnableSegmentData\(\)](#), [EnableMarkerData\(\)](#), [EnableDeviceData\(\)](#), [GetUnlabeledMarkerCount\(\)](#), [GetUnlabeledMarkerGlobalTranslation\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_DisableMarkerRayData();
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Output_DisableMarkerRayData Output = MyClient.DisableMarkerRayData();
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output = MyClient.DisableMarkerRayData ();
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output_DisableMarkerRayData Output = MyClient.DisableMarkerRayData ();
```

### Returns

An `Output_DisableMarkerRayData` class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected

## Output\_DisableDeviceData DisableDeviceData ( )

Disable force plate, EMG, and other device data in the Vicon DataStream.

See Also: [IsDeviceDataEnabled\(\)](#), [EnableDeviceData\(\)](#), [EnableSegmentData\(\)](#), [EnableMarkerData\(\)](#), [EnableUnlabeledMarkerData\(\)](#), [GetDeviceCount\(\)](#), [GetDeviceName\(\)](#), [GetDeviceOutputCount\(\)](#), [GetDeviceOutputName\(\)](#), [GetDeviceOutputValue\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_DisableDeviceData();
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Output_DisableDeviceData Output = MyClient.DisableDeviceData();
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output = MyClient.DisableDeviceData ();
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output_DisableDeviceData Output = MyClient.DisableDeviceData ();
```

### Returns

An `Output_DisableDeviceData` class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected

## Output\_DisableCentroidData DisableCentroidData ( )

Disable centroid data in the Vicon DataStream.

See Also: [IsCentroidDataEnabled\(\)](#), [EnableCentroidData\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_DisableCentroidData();
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Output_DisableCentroidData Output = MyClient.DisableCentroidData();
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output = MyClient.DisableCentroidData ();
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output_DisableCentroidData Output = MyClient.DisableCentroidData ();
```

### Returns

An `Output_DisableCentroidData` class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected

## Output\_DisableGreyscaleData DisableGreyscaleData ( )

Disable greyscale data in the Vicon DataStream.

See Also: [IsGreyscaleDataEnabled\(\)](#), [EnableGreyscaleData\(\)](#)

C example

Not implemented

C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;  
MyClient.Connect( "localhost" );  
Output_DisableGreyscaleData Output = MyClient.DisableGreyscaleData();
```

MATLAB example

Not implemented

.NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();  
MyClient.Connect( "localhost" );  
Output_DisableGreyscaleData Output = MyClient.DisableGreyscaleData ();
```

### Returns

An `Output_DisableGreyscaleData` class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected



## Output\_DisableVideoData DisableVideoData ( )

Disable video data in the Vicon DataStream.

See Also: [IsVideoDataEnabled\(\)](#), [EnableVideoData\(\)](#)

C example

Not implemented

C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;  
MyClient.Connect( "localhost" );  
Output_DisableVideoData Output = MyClient.DisableVideoData();
```

MATLAB example

Not implemented

.NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();  
MyClient.Connect( "localhost" );  
Output_DisableVideoData Output = MyClient.DisableVideoData ();
```

### Returns

An Output\_DisableVideoData class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected

## Output\_DisableDebugData DisableDebugData ( )

Disable debug data in the Vicon DataStream.

See Also: [IsDebugEnabled\(\)](#), [EnableDebugData\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_DisableDebugData();
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Output_DisableDebugData Output = MyClient.DisableDebugData();
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output = MyClient.DisableDebugData ();
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output_DisableDebugData Output = MyClient.DisableDebugData ();
```

### Returns

An `Output_DisableDebugData` class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected

## Output\_IsSegmentDataEnabled IsSegmentDataEnabled ( ) const

Return whether kinematic segment data is enabled in the Vicon DataStream.

See Also: [EnableSegmentData\(\)](#), [DisableSegmentData\(\)](#), [IsMarkerDataEnabled\(\)](#), [IsUnlabeledMarkerDataEnabled\(\)](#), [IsDeviceDataEnabled\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
CBool Output = Client_IsSegmentDataEnabled( pClient )
// Output == 0
Client_EnabledSegmentData( pClient );
CBool Output = Client_IsSegmentDataEnabled( pClient )
// Output == 1
Client_EnableUnlabeledMarkerData( pClient );
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Output_IsSegmentDataEnabled Output = MyClient.IsSegmentDataEnabled();
// Output.Enabled == false
MyClient.EnableSegmentData();
Output_IsSegmentDataEnabled Output = MyClient.IsSegmentDataEnabled();
// Output.Enabled == true
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output = MyClient.IsSegmentDataEnabled(); % Output.Enabled == false
MyClient.EnableSegmentData();
Output = MyClient.IsSegmentDataEnabled(); % Output.Enabled == true
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output_IsSegmentDataEnabled Output = MyClient.IsSegmentDataEnabled();
// Output.Enabled == false
MyClient.EnableSegmentData();
Output_IsSegmentDataEnabled Output = MyClient.IsSegmentDataEnabled();
// Output.Enabled == true
```

### Returns

An `Output_IsSegmentDataEnabled` class containing the result of the operation.

- The Result will be:
  - Whether the data is enabled

## Output\_IsLightweightSegmentDataEnabled IsLightweightSegmentDataEnabled ( ) const

Return whether the lightweight transport mode for kinematic segment data is enabled in the Vicon Data-Stream.

See Also: [EnableSegmentData\(\)](#), [DisableSegmentData\(\)](#), [IsMarkerDataEnabled\(\)](#), [IsUnlabeledMarkerDataEnabled\(\)](#), [IsDeviceDataEnabled\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
CBool Output = Client_IsLightweightSegmentDataEnabled( pClient )
// Output == 0
Client_EnabledSegmentData( pClient );
CBool Output = Client_IsLightweightSegmentDataEnabled( pClient )
// Output == 1
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Output_IsLightweightSegmentDataEnabled Output = MyClient.IsLightweightSegmentDataEnabled();
// Output.Enabled == false
MyClient.EnableSegmentData();
Output_IsLightweightSegmentDataEnabled Output = MyClient.IsLightweightSegmentDataEnabled();
// Output.Enabled == true
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output = MyClient.IsLightweightSegmentDataEnabled(); % Output.Enabled == false
MyClient.EnableSegmentData();
Output = MyClient.IsLightweightSegmentDataEnabled(); % Output.Enabled == true
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output_IsLightweightSegmentDataEnabled Output = MyClient.IsLightweightSegmentDataEnabled();
// Output.Enabled == false
MyClient.EnableSegmentData();
Output_IsLightweightSegmentDataEnabled Output = MyClient.IsLightweightSegmentDataEnabled();
// Output.Enabled == true
```

### Returns

An `Output_IsLightweightSegmentDataEnabled` class containing the result of the operation.

- The Result will be:
  - Whether the data is enabled

## Output\_IsMarkerDataEnabled IsMarkerDataEnabled ( ) const

Return whether labeled reconstructed marker data is enabled in the DataStream.

See Also: [EnableMarkerData\(\)](#), [DisableMarkerData\(\)](#), [IsSegmentDataEnabled\(\)](#), [IsUnlabeledMarkerDataEnabled\(\)](#), [IsDeviceDataEnabled\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
CBool Output = Client_IsMarkerDataEnabled( pClient );
// Output = 0;
Client_EnableMarkerData( pClient );
CBool Output = Client_IsMarkerDataEnabled( pClient );
// Output = 1;
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Output_IsMarkerDataEnabled Output = MyClient.IsMarkerDataEnabled();
// Output.Enabled == false
MyClient.EnableMarkerData();
Output_IsMarkerDataEnabled Output = MyClient.IsMarkerDataEnabled();
// Output.Enabled == true
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output = MyClient.IsMarkerDataEnabled(); % Output.Enabled == false
MyClient.EnableMarkerData();
Output = MyClient.IsMarkerDataEnabled(); % Output.Enabled == true
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new
ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output_IsMarkerDataEnabled Output = MyClient.IsMarkerDataEnabled();
// Output.Enabled == false
MyClient.EnableMarkerData();
Output_IsMarkerDataEnabled Output = MyClient.IsMarkerDataEnabled();
// Output.Enabled == true
```

### Returns

An Output\_IsMarkerDataEnabled class containing the result of the operation.

- The Result will be:
  - Whether the data is enabled

## Output\_IsUnlabeledMarkerDataEnabled IsUnlabeledMarkerDataEnabled ( ) const

Return whether unlabeled marker data is enabled in the DataStream.

See Also: [EnableUnlabeledMarkerData\(\)](#), [DisableUnlabeledMarkerData\(\)](#), [IsSegmentDataEnabled\(\)](#), [IsMarkerDataEnabled\(\)](#), [IsDeviceDataEnabled\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
CBool Output = Client_IsUnlabeledMarkerDataEnabled( pClient );
// Output = 0;
Client_EnableUnlabeledMarkerData( pClient );
CBool Output = Client_IsUnlabeledMarkerDataEnabled( pClient );
// Output = 1;
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Output_IsUnlabeledMarkerDataEnabled Output = MyClient.IsUnlabeledMarkerDataEnabled();
// Output.Enabled == false
MyClient.EnableUnlabeledMarkerData();
Output_IsUnlabeledMarkerDataEnabled Output = MyClient.IsUnlabeledMarkerDataEnabled();
// Output.Enabled == true
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output = MyClient.IsUnlabeledMarkerDataEnabled(); % Output.Enabled == false
MyClient.EnableUnlabeledMarkerData();
Output = MyClient.IsUnlabeledMarkerDataEnabled(); % Output.Enabled == true
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output_IsUnlabeledMarkerDataEnabled Output = MyClient.IsMarkerDataEnabled();
// Output.Enabled == false
MyClient.EnableUnlabeledMarkerData();
Output_IsUnlabeledMarkerDataEnabled Output = MyClient.IsUnlabeledMarkerDataEnabled();
// Output.Enabled == true
```

### Returns

An `Output_IsUnlabeledMarkerDataEnabled` class containing the result of the operation.

- The Result will be:
  - Whether the data is enabled

## ViconDataStreamSDK::CPP::Output\_IsMarkerRayDataEnabled IsMarkerRayDataEnabled ( ) const

Return whether marker ray data is enabled in the DataStream.

See Also: [EnableMarkerRayData\(\)](#), [DisableMarkerRayData\(\)](#), [IsSegmentDataEnabled\(\)](#), [IsMarkerDataEnabled\(\)](#), [IsDeviceDataEnabled\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
CBool Output = Client_IsMarkerRayDataEnabled( pClient );
// Output = 0;
Client_EnableMarkerRayData( pClient );
CBool Output = Client_IsMarkerRayDataEnabled( pClient );
// Output = 1;
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Output_IsMarkerRayDataEnabled Output = MyClient.IsMarkerRayDataEnabled();
// Output.Enabled == false
MyClient.EnableMarkerRayData();
Output_IsMarkerRayDataEnabled Output = MyClient.IsMarkerRayDataEnabled();
// Output.Enabled == true
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output = MyClient.IsMarkerRayDataEnabled(); % Output.Enabled == false
MyClient.EnableMarkerRayData();
Output = MyClient.IsMarkerRayDataEnabled(); % Output.Enabled == true
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new
ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output_IsMarkerRayDataEnabled Output = MyClient.IsMarkerDataEnabled();
// Output.Enabled == false
MyClient.EnableMarkerRayData();
Output_IsMarkerRayDataEnabled Output = MyClient.IsMarkerRayDataEnabled();
// Output.Enabled == true
```

### Returns

An `Output_IsMarkerRayDataEnabled` class containing the result of the operation.

- The Result will be:
  - Whether the data is enabled

## Output\_IsDeviceDataEnabled IsDeviceDataEnabled ( ) const

Return whether force plate, EMG, and other device data is enabled in the DataStream.

See Also: [EnableDeviceData\(\)](#), [DisableDeviceData\(\)](#), [IsSegmentDataEnabled\(\)](#), [IsMarkerDataEnabled\(\)](#), [IsUnlabeledMarkerDataEnabled\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
CBool Output = Client_IsDeviceDataDataEnabled( pClient );
// Output = 0;
Client_EnableDeviceDataData( pClient );
CBool Output = Client_IsDeviceDataDataEnabled( pClient );
// Output = 1;
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Output_IsDeviceDataEnabled Output = MyClient.IsDeviceDataEnabled();
// Output.Enabled == false
MyClient.EnableDeviceData();
Output_IsDeviceDataEnabled Output = MyClient.IsDeviceDataEnabled();
Output.Enabled == true
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output = MyClient.IsDeviceDataEnabled(); % Output.Enabled == false
MyClient.EnableDeviceData();
Output = MyClient.IsDeviceDataEnabled(); % Output.Enabled == true
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output_IsDeviceDataEnabled Output = MyClient.IsDeviceDataEnabled();
// Output.Enabled == false
MyClient.EnableDeviceData();
Output_IsDeviceDataEnabled Output = MyClient.IsDeviceDataEnabled();
// Output.Enabled == true
```

### Returns

An Output\_IsDeviceDataEnabled class containing the result of the operation.

- The Result will be:
  - Whether the data is enabled



## Output\_IsCentroidDataEnabled IsCentroidDataEnabled ( ) const

Return whether Centroid data is enabled in the DataStream.

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
CBool Output = Client_IsCentroidDataDataEnabled( pClient );
// Output = 0;
Client_EnableCentroidDataData( pClient );
CBool Output = Client_IsCentroidDataDataEnabled( pClient );
// Output = 1;
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Output_IsCentroidDataEnabled Output = MyClient.IsCentroidDataEnabled ();
// Output.Enabled == false
MyClient.EnableCentroidData();
Output_IsCentroidDataEnabled Output = MyClient.IsCentroidDataEnabled ();
// Output.Enabled == true
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output = MyClient.IsCentroidDataEnabled (); % Output.Enabled == false
MyClient.EnableCentroidData();
Output = MyClient.IsCentroidDataEnabled (); % Output.Enabled == true
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output_IsCentroidDataEnabled Output = MyClient.IsCentroidDataEnabled ();
// Output.Enabled == false
MyClient.EnableCentroidData();
Output_IsCentroidDataEnabled Output = MyClient.IsCentroidDataEnabled ();
// Output.Enabled == true
```

See Also: [EnableCentroidData\(\)](#), [DisableCentroidData\(\)](#)

### Returns

An `Output_IsCentroidDataEnabled` class containing the result of the operation.

- The Result will be:
  - Whether the data is enabled

## Output\_IsGreyscaleDataEnabled IsGreyscaleDataEnabled ( ) const

Return whether greyscale data is enabled in the DataStream.

See Also: [EnableGreyscaleData\(\)](#), [DisableGreyscaleData\(\)](#)

C example

Not implemented

C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Output_IsGreyscaleDataEnabled Output = MyClient.IsGreyscaleDataEnabled ();
// Output.Enabled == false
MyClient.EnableGreyscaleData();
Output_IsGreyscaleDataEnabled Output = MyClient.IsGreyscaleDataEnabled ();
// Output.Enabled == true
```

MATLAB example

.NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output_IsGreyscaleDataEnabled Output = MyClient.IsGreyscaleDataEnabled ();
// Output.Enabled == false
MyClient.EnableGreyscaleData();
Output_IsGreyscaleDataEnabled Output = MyClient.IsGreyscaleDataEnabled ();
// Output.Enabled == true
```

### Returns

An `Output_IsGreyscaleDataEnabled` class containing the result of the operation.

- The Result will be:
  - Whether the data is enabled

## Output\_IsVideoDataEnabled IsVideoDataEnabled ( ) const

Return whether video data is enabled in the DataStream.

See Also: [EnableVideoData\(\)](#), [DisableVideoData\(\)](#)

C example

Not implemented

C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Output_IsVideoEnabled Output = MyClient.IsVideoDataEnabled ();
// Output.Enabled == false
MyClient.EnableVideoData();
Output_IsVideoDataEnabled Output = MyClient.IsVideoDataEnabled ();
// Output.Enabled == true
```

MATLAB example

.NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output_IsVideoEnabled Output = MyClient.IsVideoDataEnabled ();
// Output.Enabled == false
MyClient.EnableVideoData();
Output_IsVideoDataEnabled Output = MyClient.IsVideoDataEnabled ();
// Output.Enabled == true
```

### Returns

An Output\_IsVideoDataEnabled class containing the result of the operation.

- The Result will be:
  - Whether the data is enabled

## Output\_IsDebugDataEnabled IsDebugDataEnabled ( ) const

Return whether debug data is enabled in the DataStream.

See Also: [EnableDebugData\(\)](#), [DisableDebugData\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
CBool Output = Client_IsDebugDataEnabled( pClient );
// Output = 0;
Client_EnableDebugData( pClient );
CBool Output = Client_IsDebugDataEnabled( pClient );
// Output = 1;
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Output_IsDebugDataEnabled Output = MyClient.IsDebugDataEnabled ();
// Output.Enabled == false
MyClient.EnableDebugData();
Output_IsDebugDataEnabled Output = MyClient.IsDebugDataEnabled ();
// Output.Enabled == true
```

### MATLAB example

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output_IsDebugDataEnabled Output = MyClient.IsDebugDataEnabled ();
// Output.Enabled == false
MyClient.EnableDebugData();
Output_IsDebugDataEnabled Output = MyClient.IsDebugDataEnabled ();
// Output.Enabled == true
```

### Returns

An Output\_IsDebugDataEnabled class containing the result of the operation.

- The Result will be:
  - Whether the data is enabled

## void SetBufferSize ( unsigned int *BufferSize* )

Set the number of frames that the client should buffer.

The default value is 1, which always supplies the latest frame. Choose higher values to reduce the risk of missing frames between calls.

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_SetBufferSize( 5 );
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.SetBufferSize( 5 );
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.SetBufferSize( 5 );
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.SetBufferSize( 5 );
```

See Also: [GetFrame\(\)](#)

### Parameters

<i>BufferSize</i>	The maximum number of frames to buffer.
-------------------	---

### Returns

Nothing

## Output\_SetStreamMode SetStreamMode ( const **StreamMode::Enum** Mode )

There are three modes that the SDK can operate in.

Each mode has a different impact on the [Client](#), Server, and network resources used.

- **ServerPush** In "ServerPush" mode, the Server pushes every new frame of data over the network to the [Client](#). The Server will try not to drop any frames. This results in the lowest latency that can be achieved. If the [Client](#) is unable to read data at the rate it is being sent, then it is buffered, firstly in the [Client](#), then on the TCP/IP connection, and then at the Server. When all the buffers are full then frames may be dropped at the Server and the performance of the Server may be affected. The [GetFrame\(\)](#) method returns the most recently received frame if available, or blocks the calling thread if the most recently received frame has already been processed.
- **ClientPull** In "ClientPull" mode, the [Client](#) waits for a call to [GetFrame\(\)](#), and then requests the latest frame of data from the Server. This increases latency, because a request must be sent over the network to the Server, the Server has to prepare the frame of data for the [Client](#), and then the data must be sent back over the network. Network bandwidth is kept to a minimum, because the Server only sends what you need. The buffers are very unlikely to be filled, and Server performance is unlikely to be affected. The [GetFrame\(\)](#) method blocks the calling thread until the frame has been received.
- **ClientPullPreFetch** "ClientPullPreFetch" is an enhancement to the "ClientPull" mode. A thread in the SDK continuously and preemptively does a "ClientPull" on your behalf, storing the latest requested frame in memory. When you next call [GetFrame\(\)](#), the SDK returns the last requested frame that was cached in memory. [GetFrame\(\)](#) does not need to block the calling thread. As with normal "ClientPull", buffers are unlikely to fill up, and Server performance is unlikely to be affected. Latency is slightly reduced, but network traffic may increase if you request frames on behalf of the [Client](#) which are never used. The stream defaults to "ClientPull" mode as this is considered the safest option. If performance is a problem, try "ClientPullPreFetch" followed by "ServerPush".

See Also: [GetFrame\(\)](#), [GetLatencyTotal\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_SetStreamMode( pClient, ServerPush );
Client_SetStreamMode( pClient, ClientPull );
Client_SetStreamMode( pClient, ClientPullPreFetch );
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.SetStreamMode( ViconDataStreamSDK::CPP::StreamMode::ServerPush );
MyClient.SetStreamMode( ViconDataStreamSDK::CPP::StreamMode::ClientPull );
MyClient.SetStreamMode( ViconDataStreamSDK::CPP::StreamMode::ClientPullPreFetch );
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( 'localhost' );
MyClient.SetStreamMode( ViconDataStreamSDK.DotNET.StreamMode.ServerPush );
MyClient.SetStreamMode( ViconDataStreamSDK.DotNET.StreamMode.ClientPull );
MyClient.SetStreamMode( ViconDataStreamSDK.DotNET.StreamMode.ClientPullPreFetch );
```

## .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.SetStreamMode( ViconDataStreamSDK.DotNET.StreamMode.ServerPush );
MyClient.SetStreamMode( ViconDataStreamSDK.DotNET.StreamMode.ClientPull );
MyClient.SetStreamMode( ViconDataStreamSDK.DotNET.StreamMode.ClientPullPreFetch );
```

## Parameters

<i>Mode</i>	Stream modes that the SDK can operate in <ul style="list-style-type: none"> <li>• StreamMode.ServerPush</li> <li>• StreamMode.ClientPull</li> <li>• StreamMode.ClientPullPreFetch</li> </ul>
-------------	--

## Returns

An Output\_SetStreamMode class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected

## ViconDataStreamSDK::CPP::Output\_SetApexDeviceFeedback SetApexDeviceFeedback ( const String & *DeviceName*, bool *i\_bOn* )

Enable haptic feedback for the selected Apex device.

Apex device names may be obtained using GetDeviceCount, GetDeviceName

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
Client_SetApexDeviceFeedback( pClient, "ViconApex_01", true );
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.GetFrame()
Output_GetDeviceName DeviceName MyClient.GetDeviceName( 0 );
MyClient.SetApexDeviceFeedback( DeviceName.DeviceName, true );
```

### MATLAB example

### .NET example

### Returns

An Output\_SetApexDeviceFeedback class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidDeviceName
  - NullClient
  - HapticAlreadySet



## Output\_SetAxisMapping SetAxisMapping ( const **Direction::Enum** XAxis, const **Direction::Enum** YAxis, const **Direction::Enum** ZAxis )

Remaps the 3D axis.

Vicon Data uses a right-handed coordinate system, with +X forward, +Y left, and +Z up. Other systems use different coordinate systems. The SDK can transform its data into any valid right-handed coordinate system by re-mapping each axis. Valid directions are "Up", "Down", "Left", "Right", "Forward", and "-Backward". Note that "Forward" means moving away from you, and "Backward" is moving towards you. Common usages are Z-up: SetAxisMapping( Forward, Left, Up ) Y-up: SetAxisMapping( Forward, Up, Right )

See Also: [GetAxisMapping\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_SetAxisMapping(pClient, Forward, Left, Up); // Z-up
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.SetAxisMapping( ViconDataStreamSDK::CPP::Direction::Forward,
ViconDataStreamSDK::CPP::Direction::Left,
ViconDataStreamSDK::CPP::Direction::Up );
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.SetAxisMapping( ViconDataStreamSDK.DotNET.Direction.Forward, ViconDataStreamSDK.DotNET.Direction.Left,
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.SetAxisMapping( ViconDataStreamSDK.DotNET.Direction.Forward,
ViconDataStreamSDK.DotNET.Direction.Left,
ViconDataStreamSDK.DotNET.Direction.Up );
```

### Parameters

XAxis	Specify the direction of your X axis relative to yourself as the observer.
YAxis	Specify the direction of your Y axis relative to yourself as the observer.
ZAxis	Specify the direction of your Z axis relative to yourself as the observer.

### Returns

An Output\_SetAxisMapping class containing the result of the operation.

- The Result will be:
  - Success
  - CoLinearAxes
  - LeftHandedAxes

## Output\_GetAxisMapping GetAxisMapping ( ) const

Get the current Axis mapping.

See Also: [SetAxisMapping\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
COutput_GetAxisMapping _Output_GetAxisMapping;
Client_GetAxisMapping( pClient, &_Output_GetAxisMapping );
// _Output_GetAxisMapping.XAxis == Forward
// _Output_GetAxisMapping.YAxis == Left
// _Output_GetAxisMapping.ZAxis == Up
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
Output_GetAxisMapping Output = MyClient.GetAxisMapping();
// Output.XAxis == ViconDataStreamSDK::CPP::Direction::Forward
// Output.YAxis == ViconDataStreamSDK::CPP::Direction::Left
// Output.ZAxis == ViconDataStreamSDK::CPP::Direction::Up
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
Output = MyClient.GetAxisMapping();
% Output.XAxis == ViconDataStreamSDK.DotNET.Direction.Forward
% Output.YAxis == ViconDataStreamSDK.DotNET.Direction.Left
% Output.ZAxis == ViconDataStreamSDK.DotNET.Direction.Up
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
Output_GetAxisMapping Output = MyClient.GetAxisMapping();
// Output.XAxis == ViconDataStreamSDK.DotNET.Direction.Forward
// Output.YAxis == ViconDataStreamSDK.DotNET.Direction.Left
// Output.ZAxis == ViconDataStreamSDK.DotNET.Direction.Up
```

### Returns

An Output\_GetAxisMapping class containing the result of the operation.

- The Result will be:
  - XAxis, YAxis, ZAxis

## Output\_GetFrame GetFrame ( )

Request a new frame to be fetched from the Vicon DataStream Server.

See Also: [SetStreamMode\(\)](#)

### C example

```
CClient * pClient = Client_Create();
CEnum Output = Client_GetFrame( pClient ); // Output == NotConnected
Client_Connect( pClient, "localhost" );
Output = Client_GetFrame( pClient ); // Output == Success
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
Output_GetFrame Output;
Output = MyClient.GetFrame(); // Output.Result == NotConnected
MyClient.Connect( "localhost" );
Output = MyClient.GetFrame(); // Output.Result == Success
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
Output = MyClient.GetFrame(); // Output.Result == NotConnected
MyClient.Connect( "localhost" );
Output = MyClient.GetFrame(); // Output.Result == Success
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
Output_GetFrame Output;
Output = MyClient.GetFrame(); // Output.Result == NotConnected
MyClient.Connect( "localhost" );
Output = MyClient.GetFrame(); // Output.Result == Success
```

### Returns

An Output\_GetFrame class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected

## Output\_GetFrameNumber GetFrameNumber ( ) const

Return the number of the last frame retrieved from the DataStream.

See Also: [GetFrame\(\)](#), [GetTimecode\(\)](#)

### C example

```
CClient * pClient = Client_Create();
COutput_GetFrameNumber _Output_GetFrameNumber;
Client_GetFrameNumber(pClient, &_Output_GetFrameNumber); // _Output_GetFrameNumber.FrameNumber == 0;
Client_Connect( pClient, "localhost" );
Client_GetFrameNumber(pClient, &_Output_GetFrameNumber); // _Output_GetFrameNumber.FrameNumber > 1;
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Output_GetFrameNumber Output;
Output = MyClient.GetFrameNumber(); // Output.Result == NoFrame
// Output.FrameNumber == 0
MyClient.GetFrame();
Output = MyClient.GetFrameNumber(); // Output.Result == Success
// Output.FrameNumber >= 1
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output = MyClient.GetFrameNumber(); % Output.Result == NoFrame
% Output.FrameNumber == 0
MyClient.GetFrame();
Output = MyClient.GetFrameNumber(); % Output.Result == Success
% Output.FrameNumber >= 1
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output_GetFrameNumber Output;
Output = MyClient.GetFrameNumber(); // Output.Result == NoFrame
// Output.FrameNumber == 0
MyClient.GetFrame();
Output = MyClient.GetFrameNumber(); // Output.Result == Success
// Output.FrameNumber >= 1
```

### Returns

An Output\_GetFrameNumber class containing the result of the operation and the frame number.

- The Result will be:
  - Success
  - NotConnected

## Output\_GetTimecode GetTimecode ( ) const

Return the timecode information for the last frame retrieved from the DataStream.

If the stream is valid but timecode is not available, the Output will be Result.Success and the Standard will be None.

See Also: [GetFrame\(\)](#), [GetFrameNumber\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
COutput_GetTimecode _Output_Timecode;
Client_GetTimecode( pClient, &_Output_Timecode );
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetTimecode Output = MyClient.GetTimecode();
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output = MyClient.GetTimecode();
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetTimecode Output = MyClient.GetTimecode();
```

### Returns

An Output\_GetTimecode class containing the result of the operation and

- Hours
- Minutes
- Seconds
- Frames
- SubFrame
- FieldFlag
- Standard
- SubFramesPerFrame
- UserBits
- The Result will be:
  - Success
  - NotConnected
  - NoFrame

## Output\_GetFrameRate GetFrameRate ( ) const

Return the Vicon camera system frame rate (in Hz) at the time of the last frame retrieved from the DataStream.

See Also: [GetFrame\(\)](#), [GetFrameNumber\(\)](#), [GetTimecode\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
COutput_GetFrameRate Rate;
Client_GetFrameRate(pClient, &Rate);
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetFrameRate Output = MyClient.GetFrameRate ();
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output = MyClient.GetFrameRate ();
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetFrameRate Output = MyClient.GetFrameRate ();
```

### Returns

An Output\_GetFrameRate class containing the result of the operation and the frame rate in hz.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame

## Output\_GetLatencySampleCount GetLatencySampleCount ( ) const

Return the number of latency measurements that were taken at various stages of the real-time pipeline.

This value can be passed into [GetLatencySampleName\(\)](#).

See Also: [GetFrame\(\)](#), [GetTimecode\(\)](#), [GetLatencyTotal\(\)](#), [GetLatencySampleName\(\)](#), [GetLatencySampleValue\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
COutput_GetLatencySampleCount _Output_LatencySampleCount;
Client_GetLatencySampleCount( pClient, &_Output_LatencySampleCount );
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetLatencySampleCount Output = MyClient.GetLatencySampleCount();
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( 'localhost' );
MyClient.GetFrame();
Output = MyClient.GetLatencySampleCount();
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetLatencySampleCount Output = MyClient.GetLatencySampleCount();
```

### Returns

An [Output\\_GetLatencySampleCount](#) class containing the result of the operation and the number of samples taken.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame

## Output\_GetLatencySampleName GetLatencySampleName ( const unsigned int *LatencySampleIndex* ) const

Return the name of a latency sample.

This value can be passed into [GetLatencySampleValue\(\)](#).

See Also: [GetFrame\(\)](#), [GetTimecode\(\)](#), [GetLatencyTotal\(\)](#), [GetLatencySampleCount\(\)](#), [GetLatencySampleValue\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
char SampleName[128];
Client_GetLatencySampleName(pClient, 0, 128, SampleName);
// SampleName = "Data Collected"
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetLatencySampleName Output = MyClient.GetLatencySampleName( 0 );
// Output.Name == "Data Collected"
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( 'localhost' );
MyClient.GetFrame();
Output = MyClient.GetLatencySampleName( 0 );
% Output.Name == 'Data Collected'
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetLatencySampleName Output = MyClient.GetLatencySampleName( 0 );
Output.Name == "Data Collected"
```

### Parameters

<i>Latency-SampleIndex</i>	The index of the name.
----------------------------	------------------------

### Returns

An [Output\\_GetLatencySampleName](#) class containing the result of the operation and the name of the latency sample.

- The Result will be:
  - Success
  - NotConnected



- NoFrame
- InvalidIndex

## Class Documentation

### Output\_GetLatencySampleValue GetLatencySampleValue ( const String & LatencySampleName ) const

Return the duration of a named latency sample in seconds.

This value can be passed into [GetLatencySampleValue\(\)](#).

See Also: [GetFrame\(\)](#), [GetTimecode\(\)](#), [GetLatencyTotal\(\)](#), [GetLatencySampleCount\(\)](#), [GetLatencySampleValue\(\)](#)

#### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
COutput_GetLatencySampleValue _Output_GetLatencySampleValue ;
Client_GetLatencySampleValue ( pClient, "Data Collected", &_Output_GetLatencySampleValue );
// _Output_GetLatencySampleValue.Value = 0.1
Client_Destroy( pClient );
```

#### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetLatencySampleValue Output =
MyClient.GetLatencySampleValue( "Data Collected" );
// Output.Value == 0.1
```

#### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( 'localhost' );
MyClient.GetFrame();
Output = MyClient.GetLatencySampleValue( 'Data Collected' );
% Output.Value == 0.1
```

#### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetLatencySampleName Output =
MyClient.GetLatencySampleValue( "Data Collected" );
// Output.Value == 0.1
```

#### Parameters

<i>Latency-SampleName</i>	The name of the latency sample
---------------------------	--------------------------------

#### Returns

An [Output\\_GetLatencySampleValue](#) class containing the result of the operation and the duration of the latency in seconds.

- The Result will be:

- Success
- NotConnected
- NoFrame
- InvalidIndex

## Output\_GetLatencyTotal GetLatencyTotal ( ) const

Return the total latency in seconds introduced at various stages of the real-time pipeline.

If no latency information is available then all latencies will be reported as 0.0.

See Also: [GetFrame\(\)](#), [GetTimecode\(\)](#), [GetLatencySampleCount\(\)](#), [GetLatencySampleName\(\)](#), [GetLatencySampleValue\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
COutput_GetLatencyTotal _Output_GetLatencyTotal ;
Client_GetLatencyTotal ( pClient, &_Output_GetLatencyTotal );
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetLatencyTotal Output = MyClient.GetLatencyTotal();
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( 'localhost' );
MyClient.GetFrame();
Output = MyClient.GetLatencyTotal();
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetLatencyTotal Output = MyClient.GetLatencyTotal();
```

### Returns

An Output\_GetLatencyTotal class containing the result of the operation and the total latency in seconds made from summing the other latencies.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame

## Output\_GetHardwareFrameNumber GetHardwareFrameNumber ( ) const

Returns the hardware frame number as used by the cameras.

This is not reset on synchronization.

See Also: [GetFrameNumber\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
COutput_GetHardwareFrameNumber _Output_GetHardwareFrameNumber ;
Client_GetHardwareFrameNumber ( pClient, &_Output_GetHardwareFrameNumber );
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetHardwareFrameNumber Output = MyClient.GetHardwareFrameNumber();
```

### MATLAB example

Not implemented

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetHardwareFrameNumber Output = MyClient.GetHardwareFrameNumber();
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetHardwareFrameNumber Output = MyClient.GetHardwareFrameNumber();
```

### Returns

An Output\_GetHardwareFrameNumber class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected

## Output\_GetFrameRateCount GetFrameRateCount ( ) const

Get the number of frame rate types that the server application reports.

See Also: [GetFrameRateName\(\)](#), [GetFrameRateValue\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
COutput_GetFrameRateCount _Output_GetFrameRateCount;
Client_GetFrameRateCount ( pClient, &_Output_GetFrameRateCount );
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetFrameRateCount Output = MyClient.GetFrameRateCount();
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( 'localhost' );
MyClient.GetFrame();
Output = MyClient.GetFrameRateCount();
% Output.Count = 3
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetFrameRateCount Output = MyClient.GetFrameRateCount();
```

### Returns

An Output\_GetFrameRateCount class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected

## Output\_GetFrameRateName GetFrameRateName ( const unsigned int *FrameRateIndex* ) const

Get the name of a frame rate type at the specified index.

See Also: [GetFrameRateCount\(\)](#), [GetFrameRateValue\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
char FramerateName[128];
Client_GetFrameRateName(pClient, 0, 128, FramerateName);
Client_Destroy( pClient );
```

### C++ example

```
A valid index is between 0 and GetFrameRateCount() - 1
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetFrameRateCount Output = MyClient.GetFrameRateCount();
if( Output.Count > 0 )
{
    Output_GetFrameRateName NameOutput = MyClient.GetFrameRateIndex( 0 );
}
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( 'localhost' );
MyClient.GetFrame();
Output = MyClient.GetFrameRateName( 0 );
% Output.Name = 'name'
```

### .NET example

```
A valid index is between 0 and GetFrameRateCount() - 1
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetFrameRateCount Output = MyClient.GetFrameRateCount();
if( Output.Count > 0 )
{
    Output_GetFrameRateName NameOutput = MyClient.GetFrameRateIndex( 0 );
}
```

### Returns

An Output\_GetFrameRateName class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected
  - InvalidIndex

## Output\_GetFrameRateValue GetFrameRateValue ( const String & *FrameRateName* ) const

Get the current value of the specified frame rate type.

See Also: [GetFrameRateCount\(\)](#), [GetFrameRateName\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
char FramerateName[128];
Client_GetFrameRateName(pClient, 0, 128, FramerateName);
Output_GetFrameRateValue FramerateValue;
Client_GetFrameRateValue(pClient, FramerateName, &FramerateValue);
Client_Destroy( pClient );
```

### C++ example

```
A valid name is obtained from GetFrameRateName
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetFrameRateCount Output = MyClient.GetFrameRateCount();
Output_GetFrameRateName NameOutput = MyClient.GetFrameRateIndex( 0 );
Output_GetFrameRateValue ValueOutput = MyClient.GetFrameRateValue( NameOutput.Name );
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( 'localhost' );
MyClient.GetFrame();
Output = MyClient.GetFrameRateName( 0 );
ValueOutput = MyClient.GetFrameRateValue( Output.Name );
% Output.Value = '200'
```

### .NET example

```
A valid name is obtained from GetFrameRateName
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetFrameRateCount Output = MyClient.GetFrameRateCount();
Output_GetFrameRateName NameOutput = MyClient.GetFrameRateIndex( 0 );
Output_GetFrameRateValue ValueOutput = MyClient.GetFrameRateValue( NameOutput.Name );
```

### Returns

An `Output_GetFrameRateValue` class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected
  - InvalidFrameRateName



## Output\_GetSubjectCount GetSubjectCount ( ) const

Return the number of subjects in the DataStream.

This information can be used in conjunction with GetSubjectName.

See Also: [GetSubjectName\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
COutput_GetSubjectCount SubjectCount;
Client_GetSubjectCount(pClient, &SubjectCount); // SubjectCount.Result == NoFrame
// SubjectCount.SubjectCount == 0;
Client_GetFrame( pClient );
Client_GetSubjectCount(pClient, &SubjectCount); // SubjectCount.Result == Success;
// SubjectCount.SubjectCount == 0;
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Output_GetSubjectCount Output;
Output = MyClient.GetSubjectCount(); // Output.Result == NoFrame
// Output.SubjectCount == 0
MyClient.GetFrame();
Output = MyClient.GetSubjectCount(); // Output.Result == Success
// Output.SubjectCount >= 0
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( 'localhost' );
Output = MyClient.GetSubjectCount(); % Output.Result == NoFrame
% Output.SubjectCount == 0
MyClient.GetFrame();
Output = MyClient.GetSubjectCount(); % Output.Result == Success
% Output.SubjectCount >= 0
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output_GetSubjectCount Output;
Output = MyClient.GetSubjectCount(); // Output.Result == NoFrame
// Output.SubjectCount == 0
MyClient.GetFrame();
Output = MyClient.GetSubjectCount(); // Output.Result == Success
// Output.SubjectCount >= 0
```

### Returns

An Output\_GetSubjectCount class containing the result of the operation and the number of subjects.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame

## Output\_GetSubjectName GetSubjectName ( const unsigned int *SubjectIndex* ) const

Return the name of a subject.

This can be passed into segment and marker functions.

See Also: [GetSubjectCount\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
char SubjectName[128];
CEnum Output = Client_GetSubjectName(pClient, 0, 128, SubjectName);
// Output == Success
// SubjectName == "AI"
Output = Client_GetSubjectName(pClient, 1, 128, SubjectName);
// Output == Success
// SubjectName == "Bob"
Output = Client_GetSubjectName(pClient, 2, 128, SubjectName);
// Output == InvalidIndex
// SubjectName == ""

Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetSubjectCount OutputGSC;
OutputGSC = MyClient.GetSubjectCount(); // OutputGSC.Result == Success
// OutputGSC.SubjectCount == 2
Output_GetSubjectName OutputGSN;
OutputGSN = MyClient.GetSubjectName(0); // OutputGSN.Result == Success
// OutputGSN.SubjectName == "AI"
OutputGSN = MyClient.GetSubjectName(1); // OutputGSN.Result == Success
// OutputGSN .SubjectName == "Bob"
OutputGSN = MyClient.GetSubjectName(2); // OutputGSN.Result == InvalidIndex
// OutputGSN.SubjectName == ""
```

### MATLAB example

```
MyClient = Client;
MyClient.Connect( 'localhost' );
MyClient.GetFrame();
OutputGSC = MyClient.GetSubjectCount(); % OutputGSC.Result == Success
% OutputGSC.SubjectCount == 2
OutputGSN = MyClient.GetSubjectName(0); % OutputGSN.Result == Success
% OutputGSN.SubjectName == 'AI'
OutputGSN = MyClient.GetSubjectName(1); % OutputGSN.Result == Success
% OutputGSN .SubjectName == 'Bob'
OutputGSN = MyClient.GetSubjectName(2); % OutputGSN.Result == InvalidIndex
% OutputGSN.SubjectName == ''
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient =
new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
```

## Class Documentation

```
Output_GetSubjectCount OutputGSC;
OutputGSC = MyClient.GetSubjectCount(); // OutputGSC.Result == Success
// OutputGSC.SubjectCount == 2
Output_GetSubjectName OutputGSN;
OutputGSN = MyClient.GetSubjectName(0); // OutputGSN.Result == Success
// OutputGSN.SubjectName == "Al"
OutputGSN = MyClient.GetSubjectName(1); // OutputGSN.Result == Success
// OutputGSN .SubjectName == "Bob"
OutputGSN = MyClient.GetSubjectName(2); // OutputGSN.Result == InvalidIndex
// OutputGSN.SubjectName == ""
```

### Parameters

<i>SubjectIndex</i>	The index of the subject. A valid Subject Index is between 0 and <a href="#">GetSubjectCount()</a> -1. Matlab: A valid Subject Index is between 1 and <a href="#">GetSubjectCount()</a> .
---------------------	---

### Returns

An `Output_GetSubjectName` `GetSubjectName` class containing the result of the operation and the name of the subject.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidIndex

## Output\_GetSubjectRootSegmentName GetSubjectRootSegmentName ( const String & *SubjectName* ) const

Return the name of the root segment for a specified subject.

This can be passed into segment functions. The root segment is the ancestor of all other segments in the subject.

See Also: [GetSegmentCount\(\)](#), [GetSegmentParentName\(\)](#), [GetSegmentChildCount\(\)](#), [GetSegmentChildName\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableSegmentData( pClient );
Client_GetFrame( pClient );
char RootSegment[128];
CEnum Result = Client_GetSubjectRootSegmentName(pClient, "Bob", 128, RootSegment);
// Result == Success
// RootSegment == "Pelvis"
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableSegmentData();
MyClient.GetFrame();
Output_GetSubjectRootSegmentName Output;
Output = MyClient.GetSubjectRootSegmentName( "Bob" );
// Output.Result == Success
// Output.SegmentName == "Pelvis"
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableSegmentData();
MyClient.GetFrame();
Output = MyClient.GetSubjectRootSegmentName( "Bob" );
% Output.Result == Success
% Output.SegmentName == "Pelvis"
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableSegmentData();
MyClient.GetFrame();
Output_GetSubjectRootSegmentName Output;
Output = MyClient.GetSubjectRootSegmentName( "Bob" );
// Output.Result == Success
// Output.SegmentName == "Pelvis"
```

### Parameters

<i>SubjectName</i>	The name of the subject
--------------------	-------------------------

## Returns

An `Output_GetSubjectRootSegmentName` class containing the result of the operation and the name of the root segment.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidIndex

## Output\_GetSegmentCount GetSegmentCount ( const String & *SubjectName* ) const

Return the number of segments for a specified subject in the DataStream.

This information can be used in conjunction with GetSegmentName.

See Also: [GetSubjectName\(\)](#), [GetSegmentName\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
COutput_GetSegmentCount SegmentCount;
Client_GetSegmentCount( pClient, "Bob", &SegmentCount );
// SegmentCount.Result == NOFrame
// SegmentCount.Value == 0
Client_GetFrame( pClient );
Client_GetSegmentCount( pClient, "AI", &SegmentCount );
// SegmentCount.Result == InvalidSubjectName
// SegmentCount.Value == 0
Client_GetSegmentCount( pClient, "Bob", &SegmentCount );
// SegmentCount.Result == Success
// SegmentCount.Value >= 0

Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.EnableSegmentData();
MyClient.Connect( "localhost" );
Output_GetSegmentCount Output;
Output = MyClient.GetSegmentCount( "Bob" ); // Output.Result == NoFrame
// Output.SegmentCount == 0
MyClient.GetFrame();
Output = MyClient.GetSegmentCount( "AI" ); // Output.Result ==
// InvalidSubjectName
// Output.SegmentCount == 0
Output = MyClient.GetSegmentCount( "Bob" ); // Output.Result == Success
// Output.SegmentCount >= 0
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.EnableSegmentData();
MyClient.Connect( "localhost" );
Output = MyClient.GetSegmentCount( "Bob" ); % Output.Result == NoFrame
% Output.SegmentCount == 0
MyClient.GetFrame();
Output = MyClient.GetSegmentCount( "AI" ); % Output.Result ==
% InvalidSubjectName
% Output.SegmentCount == 0
Output = MyClient.GetSegmentCount( "Bob" ); % Output.Result == Success
% Output.SegmentCount >= 0
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.EnableSegmentData();
MyClient.Connect( "localhost" );
Output_GetSegmentCount Output;
Output = MyClient.GetSegmentCount( "Bob" ); // Output.Result == NoFrame
```

## Class Documentation

```
// Output.SegmentCount == 0
MyClient.GetFrame();
Output = MyClient.GetSegmentCount( "Al" ); // Output.Result ==
// InvalidSubjectName
// Output.SegmentCount == 0
Output = MyClient.GetSegmentCount( "Bob" ); // Output.Result == Success
// Output.SegmentCount >= 0
```

### Parameters

<i>SubjectName</i>	The name of the subject.
--------------------	--------------------------

### Returns

An `Output_GetSegmentCount` class containing the result of the operation and the number of segments.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidIndex

## Output\_GetSegmentName GetSegmentName ( const String & *SubjectName*, const unsigned int *SegmentIndex* ) const

Return the name of a subject segment specified by index.

See Also: [GetSegmentCount\(\)](#), [GetSegmentChildCount\(\)](#), [GetSegmentChildName\(\)](#), [GetSubjectRootSegmentName\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
char SegmentName[128];
// SegmentIndex must be between 0 and GetSegmentCount() - 1
Client_GetSegmentName(pClient, "Bob", 0, 128, SegmentName);
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableSegmentData();
MyClient.GetFrame();
Output_GetSegmentName Output;
// SegmentIndex must be between 0 and GetSegmentCount() - 1
Output = MyClient.GetSegmentName( "Bob", 0 );
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableSegmentData();
MyClient.GetFrame();
% SegmentIndex must be between 0 and GetSegmentCount() - 1
Output = MyClient.GetSegmentName( "Bob", 0 );
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableSegmentData();
MyClient.GetFrame();
Output_GetSegmentParentName Output;
// SegmentIndex must be between 0 and GetSegmentCount() - 1
Output = MyClient.GetSegmentName( "Bob", 0 );
```

### Parameters

<i>SubjectName</i>	The name of the subject
<i>SegmentIndex</i>	The index of the segment

### Returns

An Output\_GetSegmentName class containing the result of the operation and the name of the parent segment or an empty string if it is the root segment.



- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidIndex
  - InvalidSubjectName

## Output\_GetSegmentChildCount GetSegmentChildCount ( const String & *SubjectName*, const String & *SegmentName* ) const

Return the number of child segments for a specified subject segment.

This can be passed into segment functions.

See Also: [GetSegmentCount\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
COutput_GetSegmentChildCount ChildCount;
Client_GetSegmentChildCount(pClient, "Bob", "Pelvis", &ChildCount);
// ChildCount.Result == Success
// ChildCount.SegmentCount == 2
Client_GetSegmentChildCount(pClient, "Alice", "Pelvis", &ChildCount);
// ChildCount.Result == InvalidSubjectName
// ChildCount.SegmentCount == 0

char SegmentName[128];
Client_GetSegmentName(pClient, "Bob", , 128, SegmentName);
Client_GetSegmentName(pClient, "Bob", &SegmentName);
// ChildCount.Result == Success
// ChildCount.SegmentCount == 2
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableSegmentData();
MyClient.GetFrame();
Output_GetSegmentChildCount OutputGSCC;
OutputGSCC = MyClient.GetSegmentChildCount( "Bob", "Pelvis" );
// OutputGSCC.Result == Success
// OutputGSCC.SegmentCount == 2
Output_GetSegmentChildName OutputGSCN;
OutputGSCN = MyClient.GetSegmentName( "Alice", 0 );
// OutputGSCN.Result == InvalidSubjectName
// OutputGSCN.SegmentName == ""
OutputGSCN = MyClient.GetSegmentName( "Bob", "Pelvis", 0 );
// OutputGSCN.Result == Success
// OutputGSCN.SegmentName == "LFemur"
OutputGSCN = MyClient.GetSegmentName( "Bob", "Pelvis", 1 );
// OutputGSCN.Result == Success
// OutputGSCN.SegmentName == "RFemur"
OutputGSCN = MyClient.GetSegmentName( "Bob", "Pelvis", 2 );
// OutputGSCN.Result == InvalidIndex
// OutputGSCN.SegmentName == ""
// (no third segment)
```

### MATLAB example

```
A valid Segment Index is between 1 and GetSegmentChildCount()
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableSegmentData();
MyClient.GetFrame();
OutputGSCC = MyClient.GetSegmentChildCount( "Bob", "Pelvis" );
```

## Class Documentation

```
% OutputGSCC.Result == Success
% OutputGSCC.SegmentCount == 2
OutputGSCN = MyClient.GetSegmentChildName( "Alice", "Pelvis", 0 );
% OutputGSCN.Result == InvalidSubjectName
% OutputGSCN.SegmentName == ""
OutputGSCN = MyClient.GetSegmentChildName( "Bob", "Pelvis", 0 );
% OutputGSCN.Result == Success
% OutputGSCN.SegmentName == "LFemur"
OutputGSCN = MyClient.GetSegmentChildName( "Bob", "Pelvis", 1 );
% OutputGSCN.Result == Success
% OutputGSCN.SegmentName == "RFemur"
OutputGSCN = MyClient.GetSegmentChildName( "Bob", "Pelvis", 2 );
% OutputGSCN.Result == InvalidIndex
% OutputGSCN.SegmentName == ""
% (no third segment)
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableSegmentData();
MyClient.GetFrame();
Output_GetSegmentChildCount OutputGSCC;
OutputGSCC = MyClient.GetSegmentChildCount( "Bob", "Pelvis" );
// OutputGSCC.Result == Success
// OutputGSCC.SegmentCount == 2
Output_GetSegmentChildName OutputGSCN;
OutputGSCN = MyClient.GetSegmentChildName( "Alice", "Pelvis", 0 );
// OutputGSCN.Result == InvalidSubjectName
// OutputGSCN.SegmentName == ""
OutputGSCN = MyClient.GetSegmentChildName( "Bob", "Pelvis", 0 );
// OutputGSCN.Result == Success
// OutputGSCN.SegmentName == "LFemur"
OutputGSCN = MyClient.GetSegmentChildName( "Bob", "Pelvis", 1 );
// OutputGSCN.Result == Success
// OutputGSCN.SegmentName == "RFemur"
OutputGSCN = MyClient.GetSegmentChildName( "Bob", "Pelvis", 2 );
// OutputGSCN.Result == InvalidIndex
// OutputGSCN.SegmentName == ""
// (no third segment)
```

### Parameters

<i>SubjectName</i>	The name of the subject
<i>SegmentName</i>	The name of the segment

### Returns

An `Output_GetSegmentChildCount` class containing the result of the operation and the number of child segments.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidIndex
  - InvalidSubjectName
  - InvalidSegmentName

## Class Documentation

### Output\_GetSegmentChildName GetSegmentChildName ( const String & *SubjectName*, const String & *SegmentName*, const unsigned int *SegmentIndex* ) const

Return the name of the child segment for a specified subject segment and index.

See Also: [GetSegmentCount\(\)](#), [GetSegmentChildCount\(\)](#), [GetSegmentChildName\(\)](#), [GetSubjectRootSegmentName\(\)](#)

#### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableSegmentData( pClient );
Client_GetFrame( pClient );
char SegmentChildName[128];
// Segment index must be between 0 and Client_GetSegmentChildCount() - 1
Client_GetSegmentChildName( pClient, "Bob", "Pelvis", 0, 128, SegmentChildName );
Client_Destroy( pClient );
```

#### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableSegmentData();
MyClient.GetFrame();
Output_GetSegmentChildName Output;
// Segment index must be between 0 and GetSegmentChildCount() - 1
Output = MyClient.GetSegmentChildName( "Bob", "Pelvis", 0 );
```

#### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableSegmentData();
MyClient.GetFrame();
// Segment index must be between 0 and GetSegmentChildCount()
Output = MyClient.GetSegmentChildName( "Bob", "Pelvis", 0 );
```

#### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableSegmentData();
MyClient.GetFrame();
Output_GetSegmentChildName Output;
// Segment index must be between 0 and GetSegmentChildCount() - 1
Output = MyClient.GetSegmentChildName( "Bob", "Pelvis", 0 );
```

#### Parameters

<i>SubjectName</i>	The name of the subject
<i>SegmentName</i>	The name of the segment
<i>SegmentIndex</i>	The index of the child segment. A valid Segment Index is between 0 and <a href="#">GetSegmentChildCount()</a> -1.

## Returns

An `Output_GetSegmentChildName` class containing the result of the operation and the name of the child segment.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidIndex
  - InvalidSubjectName
  - InvalidSegmentName

## Output\_GetSegmentParentName GetSegmentParentName ( const String & SubjectName, const String & SegmentName ) const

Return the name of the parent segment for a specified subject segment.

If the specified segment is the root segment of the subject then it will return an empty string.

See Also: [GetSegmentCount\(\)](#), [GetSegmentChildCount\(\)](#), [GetSegmentChildName\(\)](#), [GetSubjectRootSegmentName\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
char SegmentParentName[128];
CEnum Result = Client_GetSegmentParentName(pClient, "Bob", "Pelvis", 128, SegmentParentName);
// Result == Success
// SegmentParentName = ""
// This is the root segment
Result = Client_GetSegmentParentName(pClient, "Bob", "LFemur", 128, SegmentParentName);
// Result == Success
// SegmentParentName = "Pelvis"
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableSegmentData();
MyClient.GetFrame();
Output_GetSegmentParentName Output;
Output = MyClient.GetSegmentParentName( "Bob", "Pelvis" );
// Output.Result == Success
// Output.SegmentName == ""
// This is the root segment
Output = MyClient.GetSegmentParentName( "Bob", "LFemur" );
// Output.Result == Success
// Output.SegmentName == "Pelvis"
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableSegmentData();
MyClient.GetFrame();
Output = MyClient.GetSegmentParentName( "Bob", "Pelvis" );
% Output.Result == Success
% Output.SegmentCount == ""
% This is the root segment
Output = MyClient.GetSegmentParentName( "Bob", "LFemur" );
% Output.Result == Success
% Output.SegmentCount == "Pelvis"
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableSegmentData();
MyClient.GetFrame();
Output_GetSegmentParentName Output;
```

## Class Documentation

```
Output = MyClient.GetSegmentParentName( "Bob", "Pelvis" );
// Output.Result == Success
// Output.SegmentName == ""
// This is the root segment
Output = MyClient.GetSegmentParentName( "Bob", "LFemur" );
// Output.Result == Success
// Output.SegmentName == "Pelvis"
```

### Parameters

<i>SubjectName</i>	The name of the subject
<i>SegmentName</i>	The name of the segment

### Returns

An `Output_GetSegmentParentName` class containing the result of the operation and the name of the parent segment or an empty string if it is the root segment.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidSubjectName
  - InvalidSegmentName

## Output\_GetSegmentStaticTranslation GetSegmentStaticTranslation ( const String & *SubjectName*, const String & *SegmentName* ) const

Return the static pose translation of a subject segment.

The static translation of the segment corresponds to the PRE-POSITION element of the segment in the subject vsk. It is the base position of the segment, and is included in the value returned by GetLocalTranslation. If you are required to calculate the amount a segment has moved from its base position, subtract this value from the local translation.

See Also: [GetSegmentStaticRotationHelical\(\)](#), [GetSegmentStaticRotationMatrix\(\)](#), [GetSegmentStaticRotationQuaternion\(\)](#), [GetSegmentStaticRotationEulerXYZ\(\)](#), [GetSegmentLocalTranslation\(\)](#), [GetSegmentLocalRotationHelical\(\)](#), [GetSegmentLocalRotationMatrix\(\)](#), [GetSegmentLocalRotationQuaternion\(\)](#), [GetSegmentLocalRotationEulerXYZ\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
COutput_GetSegmentStaticTranslation _Output_GetSegmentStaticTranslation;
Client_GetSegmentStaticTranslation(pClient, "Alice", "Pelvis", &_Output_GetSegmentStaticTranslation);
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableSegmentData();
MyClient.GetFrame();
Output_GetSegmentStaticTranslation Output =
MyClient.GetSegmentStaticTranslation( "Alice", "Pelvis" );
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableSegmentData();
MyClient.GetFrame();
Output = MyClient.GetSegmentStaticTranslation( "Alice", "Pelvis" );
```

### .NET example

```
ViconDataStramSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableSegmentData();
MyClient.GetFrame();
Output_GetSegmentStaticTranslation Output =
MyClient.GetSegmentStaticTranslations( "Alice", "Pelvis" );
```

### Parameters

<i>SubjectName</i>	The name of the subject
<i>SegmentName</i>	The name of the segment



## Returns

An `Output_GetSegmentStaticTranslation` class containing the result of the operation and the translation of the segment.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidSubjectName
  - InvalidSegmentName

## Output\_GetSegmentStaticRotationHelical GetSegmentStaticRotationHelical ( const String & *SubjectName*, const String & *SegmentName* ) const

Return the static pose rotation of a subject segment in helical coordinates.

The helical coordinates represent a vector whose length is the amount of rotation in radians, and the direction is the axis about which to rotate.

The static rotation of the segment corresponds to the PRE-ORIENTATION element of the segment in the subject vsk. It is the base rotation of the segment, and is included in the value returned by GetLocalRotation\*. If you are required to calculate the amount a segment has rotated from its base position, subtract this value from the local rotation.

See Also: [GetSegmentStaticTranslation\(\)](#), [GetSegmentStaticRotationMatrix\(\)](#), [GetSegmentStaticRotationQuaternion\(\)](#), [GetSegmentStaticRotationEulerXYZ\(\)](#), [GetSegmentLocalTranslation](#), [GetSegmentLocalRotationHelical](#), [GetSegmentLocalRotationMatrix](#), [GetSegmentLocalRotationQuaternion](#), [GetSegmentLocalRotationEulerXYZ](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
COutput_GetSegmentStaticRotationHelical _Output_GetSegmentStaticRotationHelical;
Client_GetSegmentStaticRotationHelical(
    pClient, "Alice", "Pelvis", &_Output_GetSegmentStaticRotationHelical);
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetSegmentStaticRotationHelical Output =
MyClient.GetSegmentStaticRotationHelical( "Alice", "Pelvis" );
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output = MyClient.GetSegmentStaticRotationHelical( "Alice", "Pelvis" );
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetSegmentStaticRotationHelical Output =
MyClient.GetSegmentStaticRotationHelical( "Alice", "Pelvis" );
```

### Parameters

<i>SubjectName</i>	The name of the subject
<i>SegmentName</i>	The name of the segment

## Returns

An `Output_GetSegmentStaticRotationHelical` class containing the result of the operation and the rotation of the segment.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidSubjectName
  - InvalidSegmentName

## Output\_GetSegmentStaticRotationMatrix GetSegmentStaticRotationMatrix ( const String & *SubjectName*, const String & *SegmentName* ) const

Return the static pose rotation of a subject segment as a 3x3 row-major matrix.

The static rotation of the segment corresponds to the PRE-ORIENTATION element of the segment in the subject vsk. It is the base rotation of the segment, and is included in the value returned by GetLocalRotation\*. If you are required to calculate the amount a segment has rotated from its base position, subtract this value from the local rotation.

See Also: [GetSegmentStaticTranslation\(\)](#), [GetSegmentStaticRotationHelical\(\)](#), [GetSegmentStaticRotationQuaternion\(\)](#), [GetSegmentStaticRotationEulerXYZ\(\)](#), [GetSegmentLocalTranslation\(\)](#), [GetSegmentLocalRotationHelical\(\)](#), [GetSegmentLocalRotationQuaternion\(\)](#), [GetSegmentLocalRotationEulerXYZ\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
COutput_GetSegmentStaticRotationMatrix _Output_GetSegmentStaticRotationMatrix;
Client_GetSegmentStaticRotationMatrix(pClient, "Alice", "Pelvis", &_Output_GetSegmentStaticRotationMatrix);
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetSegmentStaticRotationMatrix Output =
MyClient.GetSegmentStaticRotationMatrix( "Alice", "Pelvis" );
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output = MyClient.GetSegmentStaticRotationMatrix( "Alice", "Pelvis" );
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetSegmentStaticRotationMatrix Output =
MyClient.GetSegmentStaticRotationMatrix( "Alice", "Pelvis" );
```

### Parameters

<i>SubjectName</i>	The name of the subject
<i>SegmentName</i>	The name of the segment

## Returns

An `Output_GetSegmentStaticRotationMatrix` class containing the result of the operation and the rotation of the segment.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidSubjectName
  - InvalidSegmentName

## Output\_GetSegmentStaticRotationQuaternion GetSegmentStaticRotationQuaternion ( const String & *SubjectName*, const String & *SegmentName* ) const

Return the static pose rotation of a subject segment in quaternion coordinates.

The quaternion is of the form (x, y, z, w) where w is the real component and x, y and z are the imaginary components. N.B. This is different from that used in many other applications, which use (w, x, y, z).

The static rotation of the segment corresponds to the PRE-ORIENTATION element of the segment in the subject vsk. It is the base rotation of the segment, and is included in the value returned by GetLocalRotation\*. If you are required to calculate the amount a segment has rotated from its base position, subtract this value from the local rotation.

See Also: [GetSegmentStaticTranslation\(\)](#), [GetSegmentStaticRotationHelical\(\)](#), [GetSegmentStaticRotationMatrix\(\)](#), [GetSegmentStaticRotationEulerXYZ\(\)](#), [GetSegmentLocalTranslation\(\)](#), [GetSegmentLocalRotationHelical\(\)](#), [GetSegmentLocalRotationMatrix\(\)](#), [GetSegmentLocalRotationQuaternion\(\)](#), [GetSegmentLocalRotationEulerXYZ\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
COutput_GetSegmentStaticRotationQuaternion _Output_GetSegmentStaticRotationQuaternion;
Client_GetSegmentStaticRotationQuaternion(
    pClient, "Alice", "Pelvis", &_Output_GetSegmentStaticRotationQuaternion);
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetSegmentStaticRotationQuaternion Output =
MyClient.GetSegmentStaticRotationQuaternion( "Alice", "Pelvis" );
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output = MyClient.GetSegmentStaticRotationQuaternion( "Alice", "Pelvis" );
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetSegmentStaticRotationQuaternion Output =
MyClient.GetSegmentStaticRotationQuaternion( "Alice", "Pelvis" );
```

### Parameters

<i>SubjectName</i>	The name of the subject
<i>SegmentName</i>	The name of the segment

## Returns

An `Output_GetSegmentStaticRotationQuaternion` class containing the result of the operation and the rotation of the segment.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidSubjectName
  - InvalidSegmentName

## Output\_GetSegmentStaticRotationEulerXYZ GetSegmentStaticRotationEulerXYZ ( const String & *SubjectName*, const String & *SegmentName* ) const

Return the static pose rotation of a subject segment in Euler XYZ coordinates.

The static rotation of the segment corresponds to the PRE-ORIENTATION element of the segment in the subject vsk. It is the base rotation of the segment, and is included in the value returned by GetLocalRotation\*. If you are required to calculate the amount a segment has rotated from its base position, subtract this value from the local rotation.

See Also: [GetSegmentStaticTranslation\(\)](#), [GetSegmentStaticRotationHelical\(\)](#), [GetSegmentStaticRotationMatrix\(\)](#), [GetSegmentStaticRotationQuaternion\(\)](#), [GetSegmentLocalTranslation\(\)](#), [GetSegmentLocalRotationHelical\(\)](#), [GetSegmentLocalRotationMatrix\(\)](#), [GetSegmentLocalRotationQuaternion\(\)](#), [GetSegmentLocalRotationEulerXYZ\(\)](#).

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
COutput_GetSegmentStaticRotationEulerXYZ _Output_GetSegmentStaticRotationEulerXYZ;
Client_GetSegmentStaticRotationEulerXYZ(
    pClient, "Alice", "Pelvis", &_Output_GetSegmentStaticRotationEulerXYZ);
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetSegmentStaticRotationEulerXYZ Output;
Output = MyClient.GetSegmentStaticRotationEulerXYZ( "Alice", "Pelvis" );
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output = MyClient.GetSegmentStaticRotationEulerXYZ( "Alice", "Pelvis" );
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetSegmentStaticRotationEulerXYZ Output =
MyClient.GetSegmentStaticRotationEulerXYZ( "Alice", "Pelvis" );
```

### Parameters

<i>SubjectName</i>	The name of the subject.
<i>SegmentName</i>	The name of the segment.



## Returns

An `Output_GetSegmentStaticRotationEulerXYZ` class containing the result of the request and the rotation of the segment  $(x,y,z)$ .

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidSubjectName
  - InvalidSegmentName

## Output\_GetSegmentStaticScale GetSegmentStaticScale ( const String & *SubjectName*, const String & *SegmentName* ) const

Return a 3D Scale of a subject segment if present.

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
COutput_GetSegmentStaticScale _Output_GetSegmentStaticScale;
Client_GetSegmentStaticScale(pClient, "Alice", "Pelvis", &_Output_GetSegmentStaticScale);
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.EnableSegmentData();
MyClient.GetFrame();
Output_GetSegmentStaticScale Output =
MyClient.GetSegmentStaticScale( "Alice", "Pelvis" );
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.EnableSegmentData();
MyClient.GetFrame();
Output = MyClient.GetSegmentStaticScale( "Alice", "Pelvis" );
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient =
new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.EnableSegmentData();
MyClient.GetFrame();
Output_GetSegmentStaticScale Output =
MyClient.GetSegmentStaticScale( "Alice", "Pelvis" );
```

### Parameters

<i>SubjectName</i>	The name of the subject.
<i>SegmentName</i>	The name of the segment.

### Returns

An Output\_GetSegmentStaticScale class containing the result of the operation, the scale of the segment.

- The Result will be:
  - Success

- NotConnected
- NoFrame
- InvalidSubjectName
- InvalidSegmentName
- NotSupported
- NotPresent

## Output\_GetSegmentGlobalTranslation GetSegmentGlobalTranslation ( const String & *SubjectName*, const String & *SegmentName* ) const

Return the translation of a subject segment in global coordinates.

The translation is of the form ( x, y, z ) where x, y and z are in millimeters with respect to the global origin.

See Also: [GetSegmentGlobalRotationHelical\(\)](#), [GetSegmentGlobalRotationMatrix\(\)](#), [GetSegmentGlobalRotationQuaternion\(\)](#), [GetSegmentGlobalRotationEulerXYZ\(\)](#), [GetSegmentLocalTranslation\(\)](#), [GetSegmentLocalRotationHelical\(\)](#), [GetSegmentLocalRotationMatrix\(\)](#), [GetSegmentLocalRotationQuaternion\(\)](#), [GetSegmentLocalRotationEulerXYZ\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
COutput_GetSegmentGlobalTranslation _Output_GetSegmentGlobalTranslation;
Client_GetSegmentGlobalTranslation(pClient, "Alice", "Pelvis", &_Output_GetSegmentGlobalTranslation);
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.EnableSegmentData();
MyClient.GetFrame();
Output_GetSegmentGlobalTranslation Output =
MyClient.GetSegmentGlobalTranslation( "Alice", "Pelvis" );
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.EnableSegmentData();
MyClient.GetFrame();
Output = MyClient.GetSegmentGlobalTranslation( "Alice", "Pelvis" );
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient =
new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.EnableSegmentData();
MyClient.GetFrame();
Output_GetSegmentGlobalTranslation Output =
MyClient.GetSegmentGlobalTranslations( "Alice", "Pelvis" );
```

### Parameters

<i>SubjectName</i>	The name of the subject.
<i>SegmentName</i>	The name of the segment.

## Returns

An `Output_GetSegmentGlobalTranslation` class containing the result of the operation, the translation of the segment, and whether the segment is occluded.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidSubjectName
  - InvalidSegmentName
- Occluded will be True if the segment was absent at this frame. In this case the translation will be [0,0,0].

## Output\_GetSegmentGlobalRotationHelical GetSegmentGlobalRotationHelical ( const String & *SubjectName*, const String & *SegmentName* ) const

Return the rotation of a subject segment in global helical coordinates.

See Also: [GetSegmentGlobalTranslation\(\)](#), [GetSegmentGlobalRotationMatrix\(\)](#), [GetSegmentGlobalRotationQuaternion\(\)](#), [GetSegmentGlobalRotationEulerXYZ\(\)](#), [GetSegmentLocalTranslation\(\)](#), [GetSegmentLocalRotationHelical\(\)](#), [GetSegmentLocalRotationMatrix\(\)](#), [GetSegmentLocalRotationQuaternion\(\)](#), [GetSegmentLocalRotationEulerXYZ\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
COutput_GetSegmentGlobalRotationHelical _Output_GetSegmentGlobalRotationHelical;
Client_GetSegmentGlobalRotationHelical(
    pClient, "Alice", "Pelvis", &_Output_GetSegmentGlobalRotationHelical);
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.GetFrame();
Output_GetSegmentGlobalRotationHelical Output =
MyClient.GetSegmentGlobalRotationHelical( "Alice", "Pelvis" );
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.GetFrame();
Output = MyClient.GetSegmentGlobalRotationHelical( "Alice", "Pelvis" );
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.GetFrame();
Output_GetSegmentGlobalRotationHelical Output =
MyClient.GetSegmentGlobalRotationHelical( "Alice", "Pelvis" );
```

### Parameters

<i>SubjectName</i>	The name of the subject.
<i>SegmentName</i>	The name of the segment.

### Returns

An Output\_GetSegmentGlobalRotationHelical class containing the result of the operation, the rotation of the segment, and whether the segment is occluded.

- The Result will be:

- Success
- NotConnected
- NoFrame
- InvalidSubjectName
- InvalidSegmentName
- Occluded will be True if the segment was absent at this frame. In this case, the rotation will be [0,0,0].

## Output\_GetSegmentGlobalRotationMatrix GetSegmentGlobalRotationMatrix ( const String & *SubjectName*, const String & *SegmentName* ) const

Return the rotation of a subject segment as a 3x3 row-major matrix in global coordinates.

See Also: [GetSegmentGlobalTranslation\(\)](#), [GetSegmentGlobalRotationHelical\(\)](#), [GetSegmentGlobalRotationQuaternion\(\)](#), [GetSegmentGlobalRotationEulerXYZ\(\)](#), [GetSegmentLocalTranslation\(\)](#), [GetSegmentLocalRotationHelical\(\)](#), [GetSegmentLocalRotationQuaternion\(\)](#), [GetSegmentLocalRotationEulerXYZ\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
COutput_GetSegmentGlobalRotationMatrix _Output_GetSegmentGlobalRotationMatrix;
Client_GetSegmentGlobalRotationMatrix(pClient, "Alice", "Pelvis", &_Output_GetSegmentGlobalRotationMatrix);
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.GetFrame();
Output_GetSegmentGlobalRotationMatrix Output =
MyClient.GetSegmentGlobalRotationMatrix( "Alice", "Pelvis" );
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.GetFrame();
Output = MyClient.GetSegmentGlobalRotationMatrix( "Alice", "Pelvis" );
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.GetFrame();
Output_GetSegmentGlobalRotationMatrix Output =
MyClient.GetSegmentGlobalRotationMatrix( "Alice", "Pelvis" );
```

### Parameters

<i>SubjectName</i>	The name of the subject.
<i>SegmentName</i>	The name of the segment.

### Returns

An Output\_GetSegmentGlobalRotationMatrix Class containing the result of the operation, the rotation of the segment, and whether the segment is occluded.

- The Result will be:



- Success
- NotConnected
- NoFrame
- InvalidSubjectName
- InvalidSegmentName
- Occluded will be True if the segment was absent at this frame.

## Output\_GetSegmentGlobalRotationQuaternion GetSegmentGlobalRotationQuaternion ( const String & *SubjectName*, const String & *SegmentName* ) const

Return the rotation of a subject segment in global quaternion coordinates.

The quaternion is of the form (x, y, z, w) where w is the real component and x, y and z are the imaginary components. N.B. This is different from that used in many other applications, which use (w, x, y, z).

See Also: [GetSegmentGlobalTranslation\(\)](#), [GetSegmentGlobalRotationHelical\(\)](#), [GetSegmentGlobalRotationMatrix\(\)](#), [GetSegmentGlobalRotationEulerXYZ\(\)](#), [GetSegmentLocalTranslation\(\)](#), [GetSegmentLocalRotationHelical\(\)](#), [GetSegmentLocalRotationMatrix\(\)](#), [GetSegmentLocalRotationQuaternion\(\)](#), [GetSegmentLocalRotationEulerXYZ\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
COutput_GetSegmentGlobalRotationQuaternion _Output_GetSegmentGlobalRotationQuaternion;
Client_GetSegmentGlobalRotationQuaternion(
    pClient, "Alice", "Pelvis", &_Output_GetSegmentGlobalRotationQuaternion);
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.GetFrame();
Output_GetSegmentGlobalRotationQuaternion Output =
MyClient.GetSegmentGlobalRotationQuaternion( "Alice", "Pelvis" );
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.GetFrame();
Output = MyClient.GetSegmentGlobalRotationQuaternion( "Alice", "Pelvis" );
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.GetFrame();
Output_GetSegmentGlobalRotationQuaternion Output =
MyClient.GetSegmentGlobalRotationQuaternion( "Alice", "Pelvis" );
```

### Parameters

<i>SubjectName</i>	The name of the subject.
<i>SegmentName</i>	The name of the segment.

## Returns

An `Output_GetSegmentGlobalRotationQuaternion` class containing the result of the operation, the rotation of the segment, and whether the segment is occluded.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidSubjectName
  - InvalidSegmentName
- Occluded will be True if the segment was absent at this frame. In this case the Rotation will be [1,0,0,0].

## Output\_GetSegmentGlobalRotationEulerXYZ GetSegmentGlobalRotationEulerXYZ ( const String & *SubjectName*, const String & *SegmentName* ) const

Return the rotation of a subject segment in global Euler XYZ coordinates.

See Also: [GetSegmentGlobalTranslation\(\)](#), [GetSegmentGlobalRotationHelical\(\)](#), [GetSegmentGlobalRotationMatrix\(\)](#), [GetSegmentGlobalRotationQuaternion\(\)](#), [GetSegmentLocalTranslation\(\)](#), [GetSegmentLocalRotationHelical\(\)](#), [GetSegmentLocalRotationMatrix\(\)](#), [GetSegmentLocalRotationQuaternion\(\)](#), [GetSegmentLocalRotationEulerXYZ\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
COutput_GetSegmentGlobalRotationEulerXYZ _Output_GetSegmentGlobalRotationEulerXYZ;
Client_GetSegmentGlobalRotationEulerXYZ(
    pClient, "Alice", "Pelvis", &_Output_GetSegmentGlobalRotationEulerXYZ);
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.GetFrame();
Output_GetSegmentGlobalRotationEulerXYZ Output =
MyClient.GetSegmentGlobalRotationEulerXYZ( "Alice", "Pelvis" );
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.GetFrame();
Output = MyClient.GetSegmentGlobalRotationEulerXYZ( "Alice", "Pelvis" );
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.GetFrame();
Output_GetSegmentGlobalRotationEulerXYZ Output =
MyClient.GetSegmentGlobalRotationEulerXYZ( "Alice", "Pelvis" );
```

### Parameters

<i>SubjectName</i>	The name of the subject.
<i>SegmentName</i>	The name of the segment.

### Returns

An Output\_GetSegmentGlobalRotationEulerXYZ class containing the result of the operation, the rotation of the segment, and whether the segment is occluded.

- The Result will be:

- Success
- NotConnected
- NoFrame
- InvalidSubjectName
- InvalidSegmentName
- Occluded will be True if the segment was absent at this frame. In this case the rotation will be [0,0,0].

## Output\_GetSegmentLocalTranslation GetSegmentLocalTranslation ( const String & *SubjectName*, const String & *SegmentName* ) const

Return the translation of a subject segment in local coordinates relative to its parent segment.

See Also: [GetSegmentLocalRotationHelical\(\)](#), [GetSegmentLocalRotationMatrix\(\)](#), [GetSegmentLocalRotationQuaternion\(\)](#), [GetSegmentLocalRotationEulerXYZ\(\)](#), [GetSegmentGlobalTranslation\(\)](#), [GetSegmentGlobalRotationHelical\(\)](#), [GetSegmentGlobalRotationMatrix\(\)](#), [GetSegmentGlobalRotationQuaternion\(\)](#), [GetSegmentGlobalRotationEulerXYZ\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
COutput_GetSegmentLocalTranslation _Output_GetSegmentLocalTranslation;
Client_GetSegmentLocalTranslation(pClient, "Alice", "Pelvis", &_Output_GetSegmentLocalTranslation);
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.EnableSegmentData();
MyClient.GetFrame();
Output_GetSegmentLocalTranslation Output =
MyClient.GetSegmentLocalTranslation( "Alice", "Pelvis" );
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.EnableSegmentData();
MyClient.GetFrame();
Output = MyClient.GetSegmentLocalTranslation( "Alice", "Pelvis" );
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.EnableSegmentData();
MyClient.GetFrame();
Output_GetSegmentLocalTranslation Output =
MyClient.GetSegmentLocalTranslations( "Alice", "Pelvis" );
```

### Parameters

<i>SubjectName</i>	The name of the subject.
<i>SegmentName</i>	The name of the segment.

## Returns

An `Output_GetSegmentLocalTranslation` class containing the result of the operation, the translation of the segment, and whether the segment is occluded.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidSubjectName
  - InvalidSegmentName
- Occluded will be True if the segment was absent at this frame. In this case the translation will be [0,0,0].

## Output\_GetSegmentLocalRotationHelical GetSegmentLocalRotationHelical ( const String & *SubjectName*, const String & *SegmentName* ) const

Return the rotation of a subject segment in local helical coordinates relative to its parent segment.

See Also: [GetSegmentLocalTranslation\(\)](#), [GetSegmentLocalRotationMatrix\(\)](#), [GetSegmentLocalRotationQuaternion\(\)](#), [GetSegmentLocalRotationEulerXYZ\(\)](#), [GetSegmentGlobalTranslation\(\)](#), [GetSegmentGlobalRotationHelical\(\)](#), [GetSegmentGlobalRotationMatrix\(\)](#), [GetSegmentGlobalRotationQuaternion\(\)](#), [GetSegmentGlobalRotationEulerXYZ\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
COutput_GetSegmentLocalRotationHelical _Output_GetSegmentLocalRotationHelical;
Client_GetSegmentLocalRotationHelical(
    pClient, "Alice", "Pelvis", &_Output_GetSegmentLocalRotationHelical);
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.GetFrame();
Output_GetSegmentLocalRotationHelical Output =
MyClient.GetSegmentLocalRotationHelical( "Alice", "Pelvis" );
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.GetFrame();
Output = MyClient.GetSegmentLocalRotationHelical( "Alice", "Pelvis" );
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.GetFrame();
Output_GetSegmentLocalRotationHelical Output =
MyClient.GetSegmentLocalRotationHelical( "Alice", "Pelvis" );
```

### Parameters

<i>SubjectName</i>	The name of the subject.
<i>SegmentName</i>	The name of the segment.

### Returns

An Output\_GetSegmentLocalRotationHelical class containing the result of the operation, the rotation of the segment, and whether the segment is occluded.

- The Result will be:



- Success
- NotConnected
- NoFrame
- InvalidSubjectName
- InvalidSegmentName
- Occluded will be True if the segment was absent at this frame. In this case the Rotation will be [0,0,0].

## Output\_GetSegmentLocalRotationMatrix GetSegmentLocalRotationMatrix ( const String & *SubjectName*, const String & *SegmentName* ) const

Return the rotation row-major matrix of a subject segment in local coordinates relative to its parent segment.

See Also: [GetSegmentLocalTranslation\(\)](#), [GetSegmentLocalRotationQuaternion\(\)](#), [GetSegmentLocalRotationEulerXYZ\(\)](#), [GetSegmentGlobalTranslation\(\)](#), [GetSegmentGlobalRotationHelical\(\)](#), [GetSegmentGlobalRotationMatrix\(\)](#) , [GetSegmentGlobalRotationQuaternion\(\)](#), [GetSegmentGlobalRotationEulerXYZ\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
COutput_GetSegmentLocalRotationMatrix _Output_GetSegmentLocalRotationMatrix;
Client_GetSegmentLocalRotationMatrix(
    pClient, "Alice", "Pelvis", &_Output_GetSegmentLocalRotationMatrix);
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.GetFrame();
Output_GetSegmentLocalRotationMatrix Output =
MyClient.GetSegmentLocalRotationMatrix( "Alice", "Pelvis" );
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.GetFrame();
Output = MyClient.GetSegmentLocalRotationMatrix( "Alice", "Pelvis" );
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.GetFrame();
Output_GetSegmentLocalRotationMatrix Output =
MyClient.GetSegmentLocalRotationMatrix( "Alice", "Pelvis" );
```

### Parameters

<i>SubjectName</i>	The name of the subject.
<i>SegmentName</i>	The name of the segment.

### Returns

An Output\_GetSegmentLocalRotationMatrix class containing the result of the operation, the rotation of the segment, and whether the segment is occluded.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidSubjectName
  - InvalidSegmentName
- Occluded will be True if the segment was absent at this frame.

## Output\_GetSegmentLocalRotationQuaternion GetSegmentLocalRotationQuaternion ( const String & *SubjectName*, const String & *SegmentName* ) const

Return the rotation of a subject segment in local quaternion coordinates relative to its parent segment.

The quaternion is of the form (x, y, z, w) where w is the real component and x, y and z are the imaginary components. N.B. This is different from that used in many other applications, which use (w, x, y, z).

See Also: [GetSegmentLocalTranslation\(\)](#), [GetSegmentLocalRotationHelical\(\)](#), [GetSegmentLocalRotationMatrix\(\)](#), [GetSegmentLocalRotationEulerXYZ\(\)](#), [GetSegmentGlobalTranslation\(\)](#), [GetSegmentGlobalRotationHelical\(\)](#), [GetSegmentGlobalRotationMatrix\(\)](#), [GetSegmentGlobalRotationQuaternion\(\)](#), [GetSegmentGlobalRotationEulerXYZ\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
COutput_GetSegmentLocalRotationQuaternion _Output_GetSegmentLocalRotationQuaternion;
Client_GetSegmentLocalRotationQuaternion(
    pClient, "Alice", "Pelvis", &_Output_GetSegmentLocalRotationQuaternion);
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.GetFrame();
Output_GetSegmentLocalRotationQuaternion Output =
MyClient.GetSegmentLocalRotationQuaternion( "Alice", "Pelvis" );
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.GetFrame();
Output = MyClient.GetSegmentLocalRotationQuaternion( "Alice", "Pelvis" );
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.GetFrame();
Output_GetSegmentLocalRotationQuaternion Output =
MyClient.GetSegmentLocalRotationQuaternion( "Alice", "Pelvis" );
```

### Parameters

<i>SubjectName</i>	The name of the subject.
<i>SegmentName</i>	The name of the segment.

## Returns

An `Output_GetSegmentLocalRotationQuaternion` class containing the result of the operation, the rotation of the segment, and whether the segment is occluded.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidSubjectName
  - InvalidSegmentName
- Occluded will be True if the segment was absent at this frame. In this case the rotation will be [1,0,0,0].

## Output\_GetSegmentLocalRotationEulerXYZ GetSegmentLocalRotationEulerXYZ ( const String & *SubjectName*, const String & *SegmentName* ) const

Return the rotation of a subject segment in local Euler XYZ coordinates relative to its parent segment.

See Also: [GetSegmentLocalTranslation\(\)](#), [GetSegmentLocalRotationHelical\(\)](#), [GetSegmentLocalRotationMatrix\(\)](#), [GetSegmentLocalRotationQuaternion\(\)](#), [GetSegmentGlobalTranslation\(\)](#), [GetSegmentGlobalRotationHelical\(\)](#), [GetSegmentGlobalRotationMatrix\(\)](#) , [GetSegmentGlobalRotationQuaternion\(\)](#), [GetSegmentGlobalRotationEulerXYZ\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
COutput_GetSegmentLocalRotationEulerXYZ _Output_GetSegmentLocalRotationEulerXYZ;
Client_GetSegmentLocalRotationEulerXYZ(
    pClient, "Alice", "Pelvis", &_Output_GetSegmentLocalRotationEulerXYZ);
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.GetFrame();
Output_GetSegmentLocalRotationEulerXYZ Output =
MyClient.GetSegmentLocalRotationEulerXYZ( "Alice", "Pelvis" );
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.GetFrame();
Output = MyClient.GetSegmentLocalRotationEulerXYZ( "Alice", "Pelvis" );
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.GetFrame();
Output_GetSegmentLocalRotationEulerXYZ Output =
MyClient.GetSegmentLocalRotationEulerXYZ( "Alice", "Pelvis" );
```

### Parameters

<i>SubjectName</i>	The name of the subject.
<i>SegmentName</i>	The name of the segment.

### Returns

An Output\_GetSegmentLocalRotationEulerXYZ class containing the result of the operation, the rotation of the segment, and whether the segment is occluded.

- The Result will be:

- Success
- NotConnected
- NoFrame
- InvalidSubjectName
- InvalidSegmentName
- Occluded will be True if the segment was absent at this frame. In this case the rotation will be [0,0,0].

## Output\_GetObjectQuality GetObjectQuality ( const String & *ObjectName* ) const

Return the quality score for a specified Object (Subject).

This is only implemented by applications that use an object tracking graph such as Evoke and Tracker.

See Also: [GetSubjectCount\(\)](#), [GetSubjectName\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_EnableSegmentData( pClient );
Client_Connect( pClient, "localhost" );
COutput_GetObjectQuality _Output_GetObjectQuality;
Client_GetObjectQuality(pClient, "Object", &_Output_GetObjectQuality);
// _output_GetObjectQuality.Result = NoFrame
// _output_GetObjectQuality.Quality = 0
Client_GetFrame( pClient );
Client_GetObjectQuality(pClient, "Object", &_Output_GetObjectQuality);
// _output_GetObjectQuality.Result = Success
// _output_GetObjectQuality.Quality >= 0.0 && _output_GetObjectQuality.Quality <= 1.0
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.EnableSegmentData();
MyClient.Connect( "localhost" );
Output_GetObjectQuality Output;
Output = MyClient.GetObjectQuality( "Object" );
// Output.Result == NoFrame
// Output.Quality == 0
MyClient.GetFrame();
Output = MyClient.GetObjectQuality( "Camera" );
// Output.Result == InvalidSubjectName
// Output.Quality == 0
// (no "Camera")
Output = MyClient.GetObjectQuality( "Object" );
// Output.Result == Success
// Output.Quality >= 0.0 && Output.Quality <= 1.0
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.EnableSegmentData ();
MyClient.Connect( "localhost" );
Output = MyClient.GetObjectQuality( "Object" );
% Output.Result == NoFrame
% Output.Quality == 0
MyClient.GetFrame();
Output = MyClient.GetObjectQuality( "Camera" );
% Output.Result == InvalidSubjectName
% Output.Quality == 0
% (no "Camera")
Output = MyClient.GetObjectQuality( "Object" );
% Output.Result == Success
% Output.Quality >= 0 && Output.Quality >= 1.0
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.EnableSegmentData ();
```



## Class Documentation

```
MyClient.Connect( "localhost" );
Output_GetMarkerCount Output;
Output = MyClient.GetObjectQuality( "Object" );
// Output.Result == NoFrame
// Output.Quality == 0
MyClient.GetFrame();
Output = MyClient.GetObjectQuality( "Camera" );
// Output.Result == InvalidSubjectName
// Output.Quality == 0
// (no "Camera")
Output = MyClient.GetObjectQuality( "Object" );
// Output.Result == Success
// Output.Quality >= 0 && Output.Quality >= 1.0
```

### Parameters

<i>ObjectName</i>	The name of the subject.
-------------------	--------------------------

### Returns

An `Output_GetObjectQuality` class containing the result of the operation and the quality score of the object.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidSubjectName

## Output\_GetMarkerCount GetMarkerCount ( const String & *SubjectName* ) const

Return the number of markers for a specified subject in the DataStream.

This information can be used in conjunction with GetMarkerName.

See Also: [GetSubjectName\(\)](#), [GetMarkerName\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableMarkerData( pClient );
COutput_GetMarkerCount MarkerCount;
Client_GetMarkerCount(pClient, "Bob", &MarkerCount);
// Output.Result = NoFrame
// Output.MarkerCount = 0
Client_GetFrame( pClient );
Client_GetMarkerCount(pClient, "Bob", &MarkerCount);
// Output.Result = Success
// Output.MarkerCount >= 0
Client_GetMarkerCount(pClient, "Alice", &MarkerCount);
// (no "Alice")
// Output.Result = InvalidSubjectName
// Output.MarkerCount == 0
Client_Destroy( pClient );
```

### C++ example

```
CPP::Client MyClient;
MyClient.EnableMarkerData();
MyClient.Connect( "localhost" );
Output_GetMarkerCount Output;
Output = MyClient.GetMarkerCount( "Bob" ); // Output.Result == NoFrame
// Output.MarkerCount == 0
MyClient.GetFrame();
Output = MyClient.GetMarkerCount( "Alice" );
// Output.Result == InvalidSubjectName
// Output.MarkerCount == 0
// (no "Alice")
Output = MyClient.GetMarkerCount( "Bob" ); // Output.Result == Success
// Output.MarkerCount >= 0
```

### MATLAB example

```
// MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.EnableMarkerData();
MyClient.Connect( "localhost" );
Output = MyClient.GetMarkerCount( "Bob" ); % Output.Result == NoFrame
% Output.MarkerCount == 0
MyClient.GetFrame();
Output = MyClient.GetMarkerCount( "Alice" );
% Output.Result == InvalidSubjectName
% Output.MarkerCount == 0
% (no "Alice")
Output = MyClient.GetMarkerCount( "Bob" ); % Output.Result == Success
% Output.MarkerCount >= 0
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient =
new ViconDataStreamSDK.DotNET.Client();
```

## Class Documentation

```
MyClient.EnableMarkerData();
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
Output_GetMarkerCount Output;
Output = MyClient.GetMarkerCount( "Bob" ); // Output.Result == NoFrame
// Output.MarkerCount == 0
MyClient.GetFrame();
Output = MyClient.GetMarkerCount( "Alice" );
// Output.Result == InvalidSubjectName
// Output.MarkerCount == 0
// (no "Alice")
Output = MyClient.GetMarkerCount( "Bob" ); // Output.Result == Success
// Output.MarkerCount >= 0
```

### Parameters

<i>SubjectName</i>	The name of the subject.
--------------------	--------------------------

### Returns

An Output\_GetMarkerCount class containing the result of the operation, and the number of markers.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidSubjectName

## Output\_GetMarkerName GetMarkerName ( const String & *SubjectName*, const unsigned int *MarkerIndex* ) const

Return the name of a marker for a specified subject.

This can be passed into GetMarkerGlobalTranslation.

See Also: [GetMarkerCount\(\)](#), [GetMarkerGlobalTranslation\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableMarkerData( pClient );
Client_GetFrame( pClient );
COutput_GetMarkerCount MarkerCount;
Client_GetMarkerCount(pClient, "Bob", &MarkerCount);
// MarkerCount.Result == Success
// MarkerCount.MarkerCount == 2
A valid Marker Index is between 0 and GetMarkerCount()-1
char MarkerName[128];
Client_GetMarkerName(pClient, "Alice", 0, 128, MarkerName);
// MarkerName.Result == InvalidSubjectName
// MarkerName.MarkerName == ""
// (no "Alice")
Client_GetMarkerName(pClient, "Bob", 0, 128, MarkerName);
// MarkerName.Result == Success
// MarkerName.MarkerName == "LASI"
Client_GetMarkerName(pClient, "Bob", 1, 128, MarkerName);
// MarkerName.Result == Success
// MarkerName.MarkerName == "RASI"
Client_GetMarkerName(pClient, "Bob", 2, 128, MarkerName);
// MarkerName.Result == InvalidIndex
// MarkerName.MarkerName == ""
// (no third marker)
Client_Destroy( pClient );
```

### C++ example

```
A valid Marker Index is between 0 and GetMarkerCount()-1
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableMarkerData();
MyClient.GetFrame();
Output_GetMarkerCount OutputGMC;
OutputGMC = MyClient.GetMarkerCount( "Bob" );
// OutputGMC.Result == Success
// OutputGMC.MarkerCount == 2
Output_GetMarkerName OutputGMN;
OutputGMN = MyClient.GetMarkerName( "Alice", 0 );
// OutputGMN.Result == InvalidSubjectName
// OutputGMN.MarkerName == ""
// (no "Alice")
OutputGMN = MyClient.GetMarkerName( "Bob", 0 );
// OutputGMN.Result == Success
// OutputGMN.MarkerName == "LASI"
OutputGMN = MyClient.GetMarkerName( "Bob", 1 );
// OutputGMN.Result == Success
// OutputGMN.MarkerName == "RASI"
OutputGMN = MyClient.GetMarkerName( "Bob", 2 );
// OutputGMN.Result == InvalidIndex
// OutputGMN.MarkerName == ""
// (no third marker)
```

## Class Documentation

### MATLAB example

```
A valid Marker Index is between 1 and GetMarkerCount()
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableMarkerData();
MyClient.GetFrame();
OutputGMC = MyClient.GetMarkerCount( "Bob" );
// OutputGMC.Result == Success
// OutputGMC.MarkerCount == 2
OutputGMN = MyClient.GetMarkerName( "Alice", 0 );
// OutputGMN.Result == InvalidSubjectName
// OutputGMN.MarkerName == ""
// (no "Alice")
OutputGMN = MyClient.GetMarkerName( "Bob", 0 );
// OutputGMN.Result == Success
// OutputGMN.MarkerName == "LASI"
OutputGMN = MyClient.GetMarkerName( "Bob", 1 );
// OutputGMN.Result == Success
// OutputGMN.MarkerName == "RASI"
OutputGMN = MyClient.GetMarkerName( "Bob", 2 );
// OutputGMN.Result == InvalidIndex
// OutputGMN.MarkerName == ""
// (no third marker)
```

### .NET example

```
A valid Marker Index is between 0 and GetMarkerCount()-1
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableMarkerData();
MyClient.GetFrame();
Output_GetMarkerCount OutputGMC;
OutputGMC = MyClient.GetMarkerCount( "Bob" );
// OutputGMC.Result == Success
// OutputGMC.MarkerCount == 2
Output_GetMarkerName OutputGMN;
OutputGMN = MyClient.GetMarkerName( "Alice", 0 );
// OutputGMN.Result == InvalidSubjectName
// OutputGMN.MarkerName == ""
// (no "Alice")
OutputGMN = MyClient.GetMarkerName( "Bob", 0 );
// OutputGMN.Result == Success
// OutputGMN.MarkerName == "LASI"
OutputGMN = MyClient.GetMarkerName( "Bob", 1 );
// OutputGMN.Result == Success
// OutputGMN.MarkerName == "RASI"
OutputGMN = MyClient.GetMarkerName( "Bob", 2 );
// OutputGMN.Result == InvalidIndex
// OutputGMN.MarkerName == ""
// (no third marker)
```

### Parameters

<i>SubjectName</i>	The name of the subject.
<i>MarkerIndex</i>	The index of the marker.

## Returns

An Output\_GetMarkerName class containing the result of the operation and the name of the marker.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidSubjectName
  - InvalidIndex

## Output\_GetMarkerParentName GetMarkerParentName ( const String & SubjectName, const String & MarkerName ) const

Return the name of the segment that is the parent of this marker.

See Also: [GetMarkerCount\(\)](#), [GetMarkerName\(\)](#), [GetMarkerGlobalTranslation\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableMarkerData( pClient );
Client_GetFrame( pClient );
char MarkerParentName[128];
CEnum Result = Client_GetMarkerParentName(pClient, "Bob", "LFHD", 128, MarkerParentName);
// Result == Success
// MarkerParentName == "Head"
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableMarkerData();
MyClient.GetFrame();
Output_GetMarkerParentName Output;
Output = MyClient.GetMarkerParentName( "Bob", "LFHD" );
// Output.Result == Success
// Output.SegmentName == "Head"
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableMarkerData();
MyClient.GetFrame();
Output = MyClient.GetMarkerParentName( "Bob", "LFHD" );
// Output.Result == Success
// Output.SegmentName == "Head"
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableMarkerData();
MyClient.GetFrame();
Output_GetMarkerParentName Output;
Output = MyClient.GetMarkerParentName( "Bob", "LFHD" );
// Output.Result == Success
// Output.SegmentName == "Head"
```

### Parameters

<i>SubjectName</i>	The name of the subject.
<i>MarkerName</i>	The name of the marker.

## Returns

An `Output_GetMarkerParentName` class containing the result of the operation and the name of the parent segment.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidSubjectName
  - InvalidMarkerName



## Output\_GetMarkerGlobalTranslation GetMarkerGlobalTranslation ( const String & *SubjectName*, const String & *MarkerName* ) const

Return the translation of a subject marker in global coordinates.

The Translation is of the form ( x, y, z ) where x, y and z are in millimeters with respect to the global origin.

See Also: [GetMarkerName\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableMarkerData( pClient );
Client_GetFrame( pClient );
COutput_GetMarkerGlobalTranslation _Output_GetMarkerGlobalTranslation;
Client_GetMarkerGlobalTranslation(pClient, "Alice", "LASI", &_Output_GetMarkerGlobalTranslation);
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableMarkerData();
MyClient.GetFrame();
Output_GetMarkerGlobalTranslation Output =
MyClient.GetMarkerGlobalTranslation( "Alice", "LASI" );
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableMarkerData();
MyClient.GetFrame();
Output = MyClient.GetMarkerGlobalTranslation( "Alice", "LASI" );
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableMarkerData();
MyClient.GetFrame();
Output_GetMarkerGlobalTranslation Output =
MyClient.GetMarkerGlobalTranslation( "Alice", "LASI" );
```

### Parameters

<i>SubjectName</i>	The name of the subject.
<i>MarkerName</i>	The name of the marker.

### Returns

An Output\_GetMarkerGlobalTranslation class containing the result of the operation, the translation of the marker, and whether the marker is occluded.

- The Result will be:

- Success
- NotConnected
- NoFrame
- InvalidSubjectName
- InvalidMarkerName
- Occluded will be true if the marker was absent at this frame. In this case the Translation will be [0,0,0].

## Class Documentation

### ViconDataStreamSDK::CPP::Output\_GetMarkerRayContributionCount GetMarkerRayContributionCount ( const String & *SubjectName*, const String & *MarkerName* ) const

Return the number of rays that are contributing to a labeled marker in the DataStream.

This information can be used in conjunction with GetMarkerRayContribution.

See Also: [GetMarkerRayContribution\(\)](#), [EnableMarkerRayData\(\)](#), [DisableMarkerRayData\(\)](#), [IsMarker-RayDataEnabled\(\)](#)

#### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableMarkerRayData( pClient );
Client_GetFrame( pClient );
COutput_GetMarkerRayContributionCount _Output_GetMarkerRayContributionCount;
Client_GetMarkerRayContributionCount( pClient, "Alice", "LASI", &_Output_GetMarkerRayContributionCount);
Client_Destroy( pClient );
```

#### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableMarkerRayData();
MyClient.GetFrame();
Output_GetMarkerRayContributionCount Output =
MyClient.GetMarkerRayContributionCount ( "Alice", "LASI" );
```

#### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableMarkerRayData();
MyClient.GetFrame();
Output = MyClient.GetMarkerRayContributionCount ( "Alice", "LASI" );
```

#### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableMarkerRayData();
MyClient.GetFrame();
Output_GetMarkerRayContributionCount Output =
MyClient.GetMarkerRayContributionCount( "Alice", "LASI" );
```

#### Parameters

<i>SubjectName</i>	The name of the subject.
<i>MarkerName</i>	The name of the marker.

#### Returns

An Output\_GetMarkerRayContributionCount class containing the result of the operation and the number of rays.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidSubjectName
  - InvalidMarkerName

## Class Documentation

**ViconDataStreamSDK::CPP::Output\_GetMarkerRayContribution** GetMarkerRayContribution ( const String & *SubjectName*, const String & *MarkerName*, unsigned int *MarkerRayContributionIndex* ) const

Return the camera ID for an indexed ray that is contributing to a labeled marker in the DataStream.

This information can be used in conjunction with GetMarkerRayContributionCount.

See Also: [GetMarkerRayContributionCount\(\)](#), [EnableMarkerRayData\(\)](#), [DisableMarkerRayData\(\)](#), [IsMarkerRayDataEnabled\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableMarkerData( pClient );
Client_GetFrame( pClient );
Output_GetMarkerRayContribution _Output_GetMarkerRayContribution;
Client_GetMarkerRayContribution(pClient, "Alice", "LASI", 0, &_Output_GetMarkerRayContribution);
Client_Destroy( pClient );
```

### C++ example

```
A valid Ray Index is between 0 and GetMarkerRayContributionCount()-1
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.EnableMarkerRayData();
MyClient.GetFrame();
Output_GetMarkerRayContribution Output =
MyClient.GetMarkerRayContribution( "Alice", "LASI", 0 );
```

### MATLAB example

```
A valid Ray Index is between 0 and GetMarkerRayContributionCount() -1
MarkerRayContributionIndex )
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.EnableMarkerRayData();
MyClient.GetFrame();
Output = MyClient.GetMarkerRayContribution ( "Alice", "LASI", 0 );
```

### .NET example

```
A valid Ray Index is between 0 and GetMarkerRayContributionCount()-1
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.EnableMarkerRayData();
MyClient.GetFrame();
Output_GetMarkerRayContribution Output =
MyClient.GetMarkerRayContribution( "Alice", "LASI", 0 );
```

### Parameters

<i>SubjectName</i>	The name of the subject.
<i>MarkerName</i>	The name of the marker.

<i>MarkerRay-Contribution-Index</i>	The index of the ray required.
-------------------------------------	--------------------------------

## Returns

An Output\_GetMarkerRayContribution class containing the result of the operation, the camera ID of the camera producing the ray and the index of the centroid resulting from the ray.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidSubjectName
  - InvalidMarkerName

## Output\_GetUnlabeledMarkerCount GetUnlabeledMarkerCount ( ) const

Return the number of unlabeled markers in the DataStream.

This information can be used in conjunction with GetGlobalUnlabeledMarkerTranslation

See Also: GetGlobalUnlabeledMarkerTranslation()

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableUnlabeledMarkerData( pClient );
Client_GetFrame( pClient );
COutput_GetUnlabeledMarkerCount UnlabeledMarkerCount;
Client_GetUnlabeledMarkerCount(pClient, &UnlabeledMarkerCount);
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.EnableUnlabeledMarkerData();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetUnlabeledMarkerCount Output =
MyClient.GetUnlabeledMarkerCount(); // Output.Result == Success
// Output.MarkerCount >= 0
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.EnableUnlabeledMarkerData();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output = MyClient.GetUnlabeledMarkerCount(); // Output.Result == Success
// Output.MarkerCount >= 0
```

### .NET example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.EnableUnlabeledMarkerData();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output = MyClient.GetUnlabeledMarkerCount(); // Output.Result == Success
// Output.MarkerCount >= 0
```

### Returns

An Output\_GetUnlabeledMarkerCount class containing the result of the operation and the number of markers.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame

## Output\_GetUnlabeledMarkerGlobalTranslation GetUnlabeledMarkerGlobalTranslation ( const unsigned int *MarkerIndex* ) const

Return the translation of an unlabeled marker in global coordinates.

The Translation is of the form ( x, y, z ) where x, y and z are in millimeters with respect to the global origin.

See Also: [GetUnlabeledMarkerCount\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableUnlabeledMarkerData( pClient );
Client_GetFrame( pClient );
COutput_GetUnlabeledMarkerGlobalTranslation _Output_GetUnlabeledMarkerGlobalTranslation;
Client_GetUnlabeledMarkerGlobalTranslation( pClient, 0, &_Output_GetUnlabeledMarkerGlobalTranslation );
Client_Destroy( pClient );
```

### C++ example

```
A valid Marker Index is between 0 and GetUnlabeledMarkerCount()-1
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableUnlabeledMarkerData();
MyClient.GetFrame();
Output_GetUnlabeledMarkerGlobalTranslation Output =
MyClient.GetUnlabeledMarkerGlobalTranslation( 0 );
```

### MATLAB example

```
A valid Marker Index is between 0 and GetUnlabeledMarkerCount() - 1
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableUnlabeledMarkerData();
MyClient.GetFrame();
Output = MyClient.GetUnlabeledMarkerGlobalTranslation( 0 );
```

### .NET example

```
A valid Marker Index is between 0 and GetUnlabeledMarkerCount()-1
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableUnlabeledMarkerData();
MyClient.GetFrame();
Output_GetUnlabeledMarkerGlobalTranslation Output =
MyClient.GetUnlabeledMarkerGlobalTranslation( 0 );
```

### Parameters

<i>MarkerIndex</i>	The index of the marker
--------------------	-------------------------

### Returns

An Output\_GetUnlabeledMarkerGlobalTranslation class containing the result of the operation and the translation of the marker.



- The Result will be:
  - Success
  - NotConnected
  - NoFrame

## Output\_GetLabeledMarkerCount GetLabeledMarkerCount ( ) const

Returns the number of all labeled markers in the datastream across all subjects.

This may be used to determine marker index range for use with [GetLabeledMarkerGlobalTranslation\(\)](#).

See Also: [GetLabeledMarkerGlobalTranslation\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableMarkerData( pClient );
Client_GetFrame( pClient );
COutput_GetLabeledMarkerCount LabeledMarkerCount;
Client_GetLabeledMarkerCount( pClient, &LabeledMarkerCount );
// LabeledMarkerCount.Result == Success
// LabeledMarkerCount.Markercount >= 0
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.EnableMarkerData();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetLabeledMarkerCount Output =
MyClient.GetLabeledMarkerCount();
// Output.Result == Success
// Output.MarkerCount >= 0
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.EnableMarkerData();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output = MyClient.GetLabeledMarkerCount(); // Output.Result == Success
// Output.MarkerCount >= 0
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.EnableMarkerData();
MyClient.Connect("localhost");
MyClient.GetFrame();
Output_GetLabeledMarkerCount Output = MyClient.GetLabeledMarkerCount();
// Output.Result == Success
// Output.MarkerCount >= 0
```

### Returns

An `Output_GetLabeledMarkerCount` class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame

## Output\_GetLabeledMarkerGlobalTranslation GetLabeledMarkerGlobalTranslation ( const unsigned int *MarkerIndex* ) const

Return the translation of a labeled marker in global coordinates.

The Translation is of the form ( x, y, z ) where x, y and z are in millimeters with respect to the global origin.

See Also: [GetLabeledMarkerCount\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableMarkerData( pClient );
Client_GetFrame( pClient );
COutput_GetLabeledMarkerGlobalTranslation LabeledMarkerGlobalTranslation;
Client_GetLabeledMarkerGlobalTranslation( pClient, &LabeledMarkerGlobalTranslation );
Client_Destroy( pClient );
```

### C++ example

```
A valid Marker Index is between 0 and GetLabeledMarkerCount()-1
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableMarkerData();
MyClient.GetFrame();
Output_GetLabeledMarkerGlobalTranslation Output =
MyClient.GetLabeledMarkerGlobalTranslation( 0 );
```

### MATLAB example

```
A valid Marker Index is between 0 and GetUnlabeledMarkerCount() - 1
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableMarkerData();
MyClient.GetFrame();
Output = MyClient.GetLabeledMarkerGlobalTranslation( 0 );    ///
```

### .NET example

```
A valid Marker Index is between 0 and GetLabeledMarkerCount()-1
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableLabeledMarkerData();
MyClient.GetFrame();
Output_GetLabeledMarkerGlobalTranslation Output =
MyClient.GetLabeledMarkerGlobalTranslation( 0 );
```

### Returns

An Output\_GetLabeledMarkerGlobalTranslation class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidIndex

## Output\_GetDeviceCount GetDeviceCount ( ) const

Return the number of force plates, EMGs, and other devices in the DataStream.

This information can be used in conjunction with GetDeviceName.

See Also: [GetDeviceName\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
COutput_GetDeviceCount DeviceCount;
Client_GetDeviceCount( pClient, &DeviceCount );
// DeviceCount.Result == Success
// DeviceCount.DeviceCount >= 0
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.EnableDeviceData();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetDeviceCount Output = MyClient.GetDeviceCount();
// Output.Result == Success
// Output.DeviceCount >= 0
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.EnableDeviceData();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output = MyClient.GetDeviceCount(); // Output.Result == Success
// Output.DeviceCount >= 0
```

### .NET example

```
ViconDataStreamSDK::DotNET::Client MyClient;
MyClient.EnableDeviceData();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetDeviceCount Output = MyClient.GetDeviceCount();
// Output.Result == Success
// Output.DeviceCount >= 0
```

### Returns

An Output\_GetDeviceCount class containing the result of the operation and the number of devices.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame

## Output\_GetDeviceName GetDeviceName ( const unsigned int *DeviceIndex* ) const

Return the name and type of a device.

This name can be passed into device functions.

See Also: [GetDeviceCount\(\)](#), [GetDeviceOutputCount\(\)](#), [GetDeviceOutputValue\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableDeviceData( pClient );
Client_GetFrame( pClient );
COutput_GetDeviceCount DeviceCount;
Client_GetDeviceCount( pClient, &DeviceCount );
// DeviceCount.Result == Success
// DeviceCount.DeviceCount == 2
char DeviceName[128];
CEnum DeviceType;
CEnum Result = Client_GetDeviceName( pClient, 0, 128, DeviceName, &DeviceType );
// Result == Success
// DeviceName == "ZeroWire"
// DeviceType == Unknown
Result = Client_GetDeviceName( pClient, 1, 128, DeviceName, &DeviceType );
// Result == Success
// DeviceName == "AMTI #1"
// DeviceType == ForcePlate
Result = Client_GetDeviceName( pClient, 2, 128, DeviceName, &DeviceType );
// Result == InvalidIndex
// DeviceName == ""
// DeviceType == Unknown
Client_Destroy( pClient );
```

### C++ example

```
A valid Device Index is between 0 and GetDeviceCount()-1
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData();
MyClient.GetFrame();
Output_GetDeviceCount OutputGDC;
OutputGDC = MyClient.GetDeviceCount( DeviceCount );
// OutputGDC.Result == Success
// OutputGDC.DeviceCount == 2
Output_GetDeviceName OutputGDN;
OutputGDN = MyClient.GetDeviceName( 0 );
// OutputGDN.Result == Success
// OutputGDN.DeviceName == "ZeroWire"
// OutputGDN.DeviceType == Unknown
OutputGDN = MyClient.GetDeviceName( 1 );
// OutputGDN.Result == Success
// OutputGDN.DeviceName == "AMTI #1"
// OutputGDN.DeviceType == ForcePlate
OutputGDN = MyClient.GetDeviceName( 2 );
// OutputGDN.Result == InvalidIndex
// OutputGDN.DeviceName == ""
// OutputGDN.DeviceType == Unknown
```

### MATLAB example

```
A valid Device Index is between 0 and GetDeviceCount() - 1
MyClient = ViconDataStreamSDK.DotNET.Client();
```

## Class Documentation

```
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData();
MyClient.GetFrame();
OutputGDC = MyClient.GetDeviceCount( DeviceCount );
% OutputGDC.Result == Success
% OutputGDC.DeviceCount == 2
OutputGDN = MyClient.GetDeviceName( 0 );
% OutputGDN.Result == Success
% OutputGDN.DeviceName == "ZeroWire"
% OutputGDN.DeviceType == Unknown
OutputGDN = MyClient.GetDeviceName( 1 );
% OutputGDN.Result == Success
% OutputGDN.DeviceName == "AMTI #1"
% OutputGDN.DeviceType == ForcePlate
OutputGDN = MyClient.GetDeviceName( 2 );
% OutputGDN.Result == InvalidIndex
% OutputGDN.DeviceName == ""
% OutputGDN.DeviceType == Unknown
```

### .NET example

```
A valid Device Index is between 0 and GetDeviceCount()-1
ViconDataStreamSDK.DotNET.Client MyClient =
new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData();
MyClient.GetFrame();
Output_GetDeviceCount OutputGDC;
OutputGDC = MyClient.GetDeviceCount( DeviceCount );
// OutputGDC.Result == Success
// OutputGDC.DeviceCount == 2
Output_GetDeviceName OutputGDN;
OutputGDN = MyClient.GetDeviceName( 0 );
// OutputGDN.Result == Success
// OutputGDN.DeviceName == "ZeroWire"
// OutputGDN.DeviceType == Unknown
OutputGDN = MyClient.GetDeviceName( 1 );
// OutputGDN.Result == Success
// OutputGDN.DeviceName == "AMTI #1"
// OutputGDN.DeviceType == ForcePlate
OutputGDN = MyClient.GetDeviceName( 2 );
// OutputGDN.Result == InvalidIndex
// OutputGDN.DeviceName == ""
// OutputGDN.DeviceType == Unknown
```

### Parameters

<i>DeviceIndex</i>	The index of the device.
--------------------	--------------------------

### Returns

An `Output_GetDeviceName` class containing the result of the operation, the name of the device, and the device type.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidIndex The Device Type will be:

- Unknown
- ForcePlate

## Class Documentation

### Output\_GetDeviceOutputCount GetDeviceOutputCount ( const String & DeviceName ) const

Return the number of outputs for a device in the DataStream.

This information can be used in conjunction with GetDeviceOutputName.

See Also: [GetDeviceName\(\)](#), [GetDeviceOutputName\(\)](#)

#### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
client_EnableDeviceData( pClient );
Client_GetFrame( pClient );
COutput_GetDeviceOutputCount DeviceOutputCount;
Client_GetDeviceOutputCount(pClient, "DataGlove", &DeviceOutputCount);
// DeviceOutputCount.Result == InvalidDeviceName
// DeviceOutputCount.DeviceOutputCount == 0
// (no "DataGlove" device)
Client_GetDeviceOutputCount(pClient, "ZeroWire", &DeviceOutputCount);
// DeviceOutputCount.Result == Success
// DeviceOutputCount.DeviceOutputCount == 6
Client_Destroy( pClient );
```

#### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData();
MyClient.GetFrame();
Output_GetDeviceOutputCount Output;
Output = MyClient.GetDeviceOutputCount( "DataGlove" );
// Output.Result == InvalidDeviceName
// Output.DeviceOutputCount == 0
// (no "DataGlove" device)
Output = MyClient.GetDeviceOutputCount( "ZeroWire" );
// Output.Result == Success
// Output.DeviceOutputCount == 6
```

#### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData();
MyClient.GetFrame();
Output = MyClient.GetDeviceOutputCount( "DataGlove" );
// Output.Result == InvalidDeviceName
// Output.DeviceOutputCount == 0
// (no "DataGlove" device)
Output = MyClient.GetDeviceOutputCount( "ZeroWire" );
// Output.Result == Success
// Output.DeviceOutputCount == 6
```

#### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData();
MyClient.GetFrame();
Output_GetDeviceOutputCount Output;
Output = MyClient.GetDeviceOutputCount( "DataGlove" );
```



## Class Documentation

```
// Output.Result == InvalidDeviceName
// Output.DeviceOutputCount == 0
// (no "DataGlove" device)
Output = MyClient.GetDeviceOutputCount( "ZeroWire" );
// Output.Result == Success
// Output.DeviceOutputCount == 6
```

### Parameters

<i>DeviceName</i>	The device name
-------------------	-----------------

### Returns

An Output\_GetDeviceOutputCount class containing the result of the operation and the number of device outputs.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidDeviceName

## Class Documentation

[Output\\_GetDeviceOutputName](#) `GetDeviceOutputName ( const String & DeviceName, const unsigned int DeviceOutputIndex ) const`

Return the name and **SI** unit of a device output.

This name can be passed into `GetDeviceOutputValue`.

See Also: [GetDeviceCount\(\)](#), [GetDeviceOutputCount\(\)](#), [GetDeviceOutputValue\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableDeviceData();
Client_GetFrame( pClient );
char DeviceOutputName[128];
CEnum DeviceOutputUnit;
CEnum Result = Client_GetDeviceOutputName(pClient, "AMTI", 0, 128, DeviceOutputName, &DeviceOutputUnit);
// Result == Success
// DeviceOutputName == "Fx"
// DeviceOutputUnit == Newton
Client_Destroy( pClient );
```

### C++ example

```
A valid Device Output Index is between 0 and GetDeviceOutputCount()-1
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.EnableDeviceData();
MyClient.GetFrame();
Output_GetDeviceOutputName Output =
MyClient.GetDeviceOutputName( "AMTI", 0 );
// Output.Result == Success
// Output.DeviceOutputName == "Fx"
// Output.DeviceOutputUnit == Newton
```

### MATLAB example

```
A valid Device Output Index is between 0 and GetDeviceOutputCount() - 1
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.EnableDeviceData();
MyClient.GetFrame();
Output = MyClient.GetDeviceOutputName( "AMTI", 0 );
% Output.Result == Success
% Output.DeviceOutputName == "Fx"
% Output.DeviceOutputUnit == Newton
```

### .NET example

```
A valid Device Output Index is between 0 and GetDeviceOutputCount()-1
ViconDataStreamSDK.DotNET.Client MyClient =
new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.EnableDeviceData();
MyClient.GetFrame();
Output_GetDeviceOutputName Output =
MyClient.GetDeviceOutputName( "AMTI", 0 );
// Output.Result == Success
```

## Class Documentation

```
// Output.DeviceOutputName == "Fx"
// Output.DeviceOutputUnit == Newton
```

### Parameters

<i>DeviceName</i>	The device name
<i>DeviceOutput-Index</i>	The index of the device output

### Returns

An `Output_GetDeviceOutputName` class containing the result of the operation, the name of the device output and the unit of the device output.

- The Result will be:
  - Success
  - NotConnected
- The DeviceOutputName could be:
  - "Fx" - Force X
  - "Fy" - Force Y
  - "Fz" - Force Z
  - "Mx" - Moment X
  - "My" - Moment Y
  - "Mz" - Moment Z
  - "Cx" - Center Of Pressure X
  - "Cy" - Center Of Pressure Y
  - "Cz" - Center Of Pressure Z
  - "Pin1" - Analog Input 1
  - "Pin2" - Analog Input 2 - The Device Output Unit will be:
    - Unit.Unknown
    - Unit.Volt
    - Unit.Newton
    - Unit.NewtonMeter
    - Unit.Meter
    - Unit.Kilogram
    - Unit.Second
    - Unit.Ampere
    - Unit.Kelvin
    - Unit.Mole
    - Unit.Candela
    - Unit.Radian

- Unit.Steradian
- Unit.MeterSquared
- Unit.MeterCubed
- Unit.MeterPerSecond
- Unit.MeterPerSecondSquared
- Unit.RadianPerSecond
- Unit.RadianPerSecondSquared
- Unit.Hertz
- Unit.Joule
- Unit.Watt
- Unit.Pascal
- Unit.Lumen
- Unit.Lux
- Unit.Coulomb
- Unit.Ohm
- Unit.Farad
- Unit.Weber
- Unit.Tesla
- Unit.Henry
- Unit.Siemens
- Unit.Becquerel
- Unit.Gray
- Unit.Sievert
- Unit.Katal

## Output\_GetDeviceOutputComponentName GetDeviceOutputComponentName ( const String & DeviceName, const unsigned int DeviceOutputIndex ) const

Return the name of the output and component and **SI** unit of a device output.

This name can be passed into GetDeviceOutputValue.

See Also: [GetDeviceCount\(\)](#), [GetDeviceOutputCount\(\)](#), [GetDeviceOutputValue\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableDeviceData();
Client_GetFrame( pClient );
char DeviceOutputName[128];
CEnum DeviceOutputUnit;
CEnum Result = Client_GetDeviceOutputComponentName(pClient, "AMTI", 0, 128, DeviceOutputName, &DeviceOutputUnit);
// Result == Success
// DeviceOutputName == "Force"
// DeviceOutputComponentName == "Fx"
// DeviceOutputUnit == Newton
Client_Destroy( pClient );
```

### C++ example

```
A valid Device Output Index is between 0 and GetDeviceOutputCount()-1
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.EnableDeviceData();
MyClient.GetFrame();
Output_GetDeviceOutputComponentName Output =
MyClient.GetDeviceOutputComponentName( "AMTI", 0 );
// Output.Result == Success
// Output.DeviceOutputName == "Force"
// Output.DeviceOutputComponentName == "Fx"
// Output.DeviceOutputUnit == Newton
```

### MATLAB example

```
A valid Device Output Index is between 0 and GetDeviceOutputCount() - 1
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.EnableDeviceData();
MyClient.GetFrame();
Output = MyClient.GetDeviceOutputComponentName( "AMTI", 0 );
% Output.Result == Success
% Output.DeviceOutputName == "Force"
% Output.DeviceOutputComponentName == "Fx"
% Output.DeviceOutputUnit == Newton
```

### .NET example

```
A valid Device Output Index is between 0 and GetDeviceOutputCount()-1
ViconDataStreamSDK.DotNET.Client MyClient =
new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.EnableDeviceData();
MyClient.GetFrame();
```

## Class Documentation

```
Output_GetDeviceOutputComponentName Output =
MyClient.GetDeviceOutputComponentName( "AMTI", 0 );
// Output.Result == Success
// Output.DeviceOutputName == "Force"
// Output.DeviceOutputComponentName == "Fx"
// Output.DeviceOutputUnit == Newton
```

### Parameters

<i>DeviceName</i>	The device name
<i>DeviceOutput-Index</i>	The index of the device output

### Returns

An `Output_GetDeviceOutputName` class containing the result of the operation, the name of the device output and component and the unit of the device output.

- The Result will be:
  - Success
  - NotConnected
- The DeviceOutputName could be:
  - "Fx" - Force X
  - "Fy" - Force Y
  - "Fz" - Force Z
  - "Mx" - Moment X
  - "My" - Moment Y
  - "Mz" - Moment Z
  - "Cx" - Center Of Pressure X
  - "Cy" - Center Of Pressure Y
  - "Cz" - Center Of Pressure Z
  - "Pin1" - Analog Input 1
  - "Pin2" - Analog Input 2
  - Custom text if output has been renamed by the user in the application - The Device Output Unit will be:
    - Unit.Unknown
    - Unit.Volt
    - Unit.Newton
    - Unit.NewtonMeter
    - Unit.Meter
    - Unit.Kilogram
    - Unit.Second
    - Unit.Ampere

- Unit.Kelvin
- Unit.Mole
- Unit.Candela
- Unit.Radian
- Unit.Steradian
- Unit.MeterSquared
- Unit.MeterCubed
- Unit.MeterPerSecond
- Unit.MeterPerSecondSquared
- Unit.RadianPerSecond
- Unit.RadianPerSecondSquared
- Unit.Hertz
- Unit.Joule
- Unit.Watt
- Unit.Pascal
- Unit.Lumen
- Unit.Lux
- Unit.Coulomb
- Unit.Ohm
- Unit.Farad
- Unit.Weber
- Unit.Tesla
- Unit.Henry
- Unit.Siemens
- Unit.Becquerel
- Unit.Gray
- Unit.Sievert
- Unit.Katal

## Output\_GetDeviceOutputValue GetDeviceOutputValue ( const String & DeviceName, const String & DeviceOutputComponentName ) const

Return the value of a device output.

If there are multiple samples for a frame, then the first sample is returned. The force plate data provided in the individual device channels is in a coordinate system local to the force plate aligned Z upwards, Y towards the front of the force plate. This coordinate system is located at the center of the top surface of the force plate. Any plate origin offset has been accounted for in the moment data. These are forces not reactions.

See Also: [GetDeviceCount\(\)](#), [GetDeviceOutputCount\(\)](#), [GetDeviceOutputName\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableDeviceData( pClient );
Client_GetFrame( pClient );
COutput_GetDeviceOutputValue _Output_GetDeviceOutputValue;
Client_GetDeviceOutputComponentValue( pClient, "AMTI", "Fx", &_Output_GetDeviceOutputValue );
// _OutputGetDeviceOutputValue.Result == Success
// _OutputGetDeviceOutputValue.Value == ?
// _OutputGetDeviceOutputValue.Value.Occluded = ?
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData();
MyClient.GetFrame();
Output_GetDeviceOutputValue Output =
MyClient.GetDeviceOutputValue( "AMTI", "Fx" );
// Output.Result == Success
// Output.Value == ?
// Output.Occluded = ?
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData();
MyClient.GetFrame();
Output = MyClient.GetDeviceOutputValue( "AMTI", "Fx" );
// Output.Result == Success
// Output.Value == ?
// Output.Occluded = ?
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient =
new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData();
MyClient.GetFrame();
Output_GetDeviceOutputValue Output =
MyClient.GetDeviceOutputValue( "AMTI", "Fx" );
// Output.Result == Success
// Output.Value == ?
// Output.Occluded = ?
```



## Parameters

<i>DeviceName</i>	The device name
<i>DeviceOutput-Component-Name</i>	The name of the device output - This is the component name, not the output name

## Returns

An Output\_GetDeviceOutputValue class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidDeviceName
  - InvalidDeviceOutputName

## Output\_GetDeviceOutputValue GetDeviceOutputValue ( const String & DeviceName, const String & DeviceOutputName, const String & DeviceOutputComponentName ) const

Return the value of a device output.

If there are multiple samples for a frame, then the first sample is returned. The force plate data provided in the individual device channels is in a coordinate system local to the force plate aligned Z upwards, Y towards the front of the force plate. This coordinate system is located at the center of the top surface of the force plate. Any plate origin offset has been accounted for in the moment data. These are forces not reactions.

See Also: [GetDeviceCount\(\)](#), [GetDeviceOutputCount\(\)](#), [GetDeviceOutputName\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableDeviceData( pClient );
Client_GetFrame( pClient );
COutput_GetDeviceOutputValue _Output_GetDeviceOutputValue;
Client_GetDeviceOutputComponentValue( pClient, "AMTI", "Force", "Fx", &_Output_GetDeviceOutputValue );
// _OutputGetDeviceOutputValue.Result == Success
// _OutputGetDeviceOutputValue.Value == ?
// _OutputGetDeviceOutputValue.Value.Occluded = ?
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData();
MyClient.GetFrame();
Output_GetDeviceOutputValue Output =
MyClient.GetDeviceOutputValue( "AMTI", "Force", "Fx" );
// Output.Result == Success
// Output.Value == ?
// Output.Occluded = ?
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData();
MyClient.GetFrame();
Output = MyClient.GetDeviceOutputValue( "AMTI", "Force", "Fx" );
// Output.Result == Success
// Output.Value == ?
// Output.Occluded = ?
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient =
new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData();
MyClient.GetFrame();
Output_GetDeviceOutputValue Output =
MyClient.GetDeviceOutputValue( "AMTI", "Force", "Fx" );
// Output.Result == Success
// Output.Value == ?
// Output.Occluded = ?
```

## Parameters

<i>DeviceName</i>	The device name
<i>DeviceOutput-Name</i>	The name of the device output
<i>DeviceOutput-Component-Name</i>	The name of the device output component

## Returns

An `Output_GetDeviceOutputValue` class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidDeviceName
  - InvalidDeviceOutputName

## Output\_GetDeviceOutputSubsamples GetDeviceOutputSubsamples ( const String & DeviceName, const String & DeviceOutputName ) const

Return the number of samples available for the specified device at the current frame.

If an analog device is sampling at 1000 Hz and the system is running at 100 Hz then this function will return 10. The samples can be accessed by supplying the subsample index to GetDeviceOutputValue. See below.

See Also: [GetDeviceOutputCount\(\)](#), [GetDeviceOutputValue\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableDeviceData( pClient );
Client_GetFrame( pClient );
COutput_GetDeviceOutputSubsamples DeviceOutputSubsamples;
Client_GetDeviceOutputSubsamples( pClient, "AMTI", "Fx", &DeviceOutputSubsamples );
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData();
MyClient.GetFrame();
Output_GetDeviceOutputSubsamples Output =
MyClient.GetDeviceOutputSubsamples ( "AMTI", "Fx" );
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData();
MyClient.GetFrame();
Output = MyClient.GetDeviceOutputSubsamples ( "AMTI", "Fx" );
// Output.Result == Success
// Output.DeviceOutputSubsamples == ?
// Output.Occluded = ?
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient =
new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData();
MyClient.GetFrame();
Output_GetDeviceOutputSubsamples Output =
MyClient.GetDeviceOutputSubsamples( "AMTI", "Fx" );
// Output.Result == Success
// Output.DeviceOutputSubsamples == ?
// Output.Occluded = ?
```

### Parameters

<i>DeviceName</i>	The device name
<i>DeviceOutput-Name</i>	The name of the device output - This is the component name, not the output name

## Returns

An `Output_GetDeviceOutputSubsamples` class containing the result of the operation, the number of subsamples for this device output, and whether the device is occluded.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidDeviceName
  - InvalidDeviceOutputName
- Occluded will be true if the value was absent at this frame. In this case the value will be 0.

[Output\\_GetDeviceOutputSubsamples](#) `GetDeviceOutputSubsamples ( const String & DeviceName, const String & DeviceOutputName, const String & DeviceOutputComponentName ) const`

Return the number of samples available for the specified device at the current frame.

If an analog device is sampling at 1000 Hz and the system is running at 100 Hz then this function will return 10. The samples can be accessed by supplying the subsample index to `GetDeviceOutputValue`. See below.

See Also: [GetDeviceOutputCount\(\)](#), [GetDeviceOutputValue\(\)](#)

#### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableDeviceData( pClient );
Client_GetFrame( pClient );
COutput_GetDeviceOutputSubsamples DeviceOutputSubsamples;
Client_GetDeviceOutputComponentSubsamples( pClient, "AMTI", "Force", "Fx", &DeviceOutputSubsamples );
Client_Destroy( pClient );
```

#### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData();
MyClient.GetFrame();
Output_GetDeviceOutputSubsamples Output =
MyClient.GetDeviceOutputSubsamples ( "AMTI", "Force", "Fx" );
```

#### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData();
MyClient.GetFrame();
Output = MyClient.GetDeviceOutputSubsamples ( "AMTI", "Force", "Fx" );
// Output.Result == Success
// Output.DeviceOutputSubsamples == ?
// Output.Occluded = ?
```

#### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient =
new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData();
MyClient.GetFrame();
Output_GetDeviceOutputSubsamples Output =
MyClient.GetDeviceOutputSubsamples( "AMTI", "Force", "Fx" );
// Output.Result == Success
// Output.DeviceOutputSubsamples == ?
// Output.Occluded = ?
```

#### Parameters

<i>DeviceName</i>	The device name
-------------------	-----------------

<i>DeviceOutput-Name</i>	The name of the device output
<i>DeviceOutput-Component-Name</i>	The name of the device output component

## Returns

An Output\_GetDeviceOutputSubsamples class containing the result of the operation, the number of subsamples for this device output, and whether the device is occluded.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidDeviceName
  - InvalidDeviceOutputName
- Occluded will be true if the value was absent at this frame. In this case the value will be 0.

## Class Documentation

**Output\_GetDeviceOutputValue** GetDeviceOutputValue ( const String & *DeviceName*, const String & *DeviceOutputName*, const unsigned int *Subsample* ) const

Return the value of a device output.

This override allows access to the individual subsamples for the current frame of data. See [GetDeviceOutputValue](#) for information about the meaning of the force plate channels.

See Also: [GetDeviceOutputSubsamples\(\)](#), [GetDeviceOutputValue\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableDeviceData( pClient );
Client_GetFrame( pClient );
COutput_GetDeviceOutputValue _Output_GetDeviceOutputValue;
Client_GetDeviceOutputValueForSubsample( pClient, "AMTI", "Fx", 6, &_Output_GetDeviceOutputValue);
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData();
MyClient.GetFrame();
Output_GetDeviceOutputValue Output =
MyClient.GetDeviceOutputValue( "AMTI", "Fx", 6 );
// Output.Result == Success
// Output.Value == ?
// Output.Occluded = ?
```

### MATLAB example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData();
MyClient.GetFrame();
Output_GetDeviceOutputValue Output =
MyClient.GetDeviceOutputValue( "AMTI", "Fx", 6 );
// Output.Result == Success
// Output.Value == ?
// Output.Occluded = ?
// Output.Value == ?
// Output.Occluded = ?
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData();
MyClient.GetFrame();
Output_GetDeviceOutputValue Output =
MyClient.GetDeviceOutputValue( "AMTI", "Fx", 6 );
// Output.Result == Success
// Output.Value == ?
// Output.Occluded = ?
```

### Parameters



<i>DeviceName</i>	The device name
<i>DeviceOutput-Name</i>	The name of the device output - This is the component name, not the output name.
<i>Subsample</i>	The subsamples to access

## Returns

An `Output_GetDeviceOutputValue` class containing the result of the operation, the value of the device output, and whether the device is occluded.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidDeviceName
  - InvalidDeviceIndex
  - InvalidDeviceOutputName

## Class Documentation

**Output\_GetDeviceOutputValue** GetDeviceOutputValue ( const String & *DeviceName*, const String & *DeviceOutputName*, const String & *DeviceOutputComponentName*, const unsigned int *Subsample* ) const

Return the value of a device output.

This override allows access to the individual subsamples for the current frame of data. See [GetDeviceOutputValue](#) for information about the meaning of the force plate channels.

See Also: [GetDeviceOutputSubsamples\(\)](#), [GetDeviceOutputValue\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableDeviceData( pClient );
Client_GetFrame( pClient );
COutput_GetDeviceOutputValue _Output_GetDeviceOutputValue;
Client_GetDeviceOutputComponentValueForSubsample( pClient,
                                                    "AMTI", "Force", "Fx",
                                                    6, &_Output_GetDeviceOutputValue);

Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData();
MyClient.GetFrame();
Output_GetDeviceOutputValue Output =
MyClient.GetDeviceOutputValue( "AMTI", "Force", "Fx", 6 );
// Output.Result == Success
// Output.Value == ?
// Output.Occluded = ?
```

### MATLAB example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData();
MyClient.GetFrame();
Output_GetDeviceOutputValue Output =
MyClient.GetDeviceOutputValue( "AMTI", "Force", "Fx", 6 );
// Output.Result == Success
// Output.Value == ?
// Output.Occluded = ?
// Output.Value == ?
// Output.Occluded = ?
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData();
MyClient.GetFrame();
Output_GetDeviceOutputValue Output =
MyClient.GetDeviceOutputValue( "AMTI", "Force", "Fx", 6 );
// Output.Result == Success
// Output.Value == ?
// Output.Occluded = ?
```

## Parameters

<i>DeviceName</i>	The device name
<i>DeviceOutput-Name</i>	The name of the device output
<i>DeviceOutput-Component-Name</i>	The name of the device output component
<i>Subsample</i>	The subsamples to access

## Returns

An `Output_GetDeviceOutputValue` class containing the result of the operation, the value of the device output, and whether the device is occluded.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidDeviceName
  - InvalidDeviceIndex
  - InvalidDeviceOutputName

## Output\_GetForcePlateCount GetForcePlateCount ( ) const

Return the number of force plates available in the DataStream.

See Also: [GetGlobalForceVector\(\)](#), [GetGlobalMomentVector\(\)](#), [GetGlobalCentreOfPressure\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_EnableDeviceData( pClient );
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
COutput_GetForcePlateCount ForcePlateCount;
Client_GetForcePlateCount(pClient, &ForcePlateCount);
// ForcePlateCount.Result == Success
// ForcePlateCount.ForcePlateCount >= 0
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.EnableDeviceData();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetForcePlateCount Output = MyClient.GetForcePlateCount ();
// Output.Result == Success
// Output.ForcePlateCount >= 0
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.EnableDeviceData();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output = MyClient.GetForcePlateCount(); // Output.Result == Success
// Output.ForcePlateCount >= 0
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.EnableDeviceData();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetForcePlateCount Output = MyClient.GetForcePlateCount();
// Output.Result == Success
// Output.ForcePlateCount >= 0
```

### Returns

An `Output_GetForcePlateCount` class containing the result of the operation and the number of force plates.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame

## Output\_GetGlobalForceVector GetGlobalForceVector ( const unsigned int ForcePlateIndex ) const

Return the force vector for the force plate in global coordinates.

The vector is in Newtons and is with respect to the global coordinate system regardless of the orientation of the force plate. The vector represents the force exerted upon the force plate, not the reaction force. If multiple subsamples are available, this function returns the first subsample. See the alternate version of this function to access all of the analog data.

See Also: [GetGlobalMomentVector\(\)](#), [GetGlobalCentreOfPressure\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableDeviceData ( pClient );
Client_GetFrame( pClient );
COutput_GetGlobalForceVector _Output_GetForceVector;
Client_GetGlobalForceVector( pClient, 0, &_Output_GetForceVector);
Client_Destroy( pClient );
```

### C++ example

```
A valid ForcePlateIndex is between 0 and GetForcePlateCount()-1
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData ();
MyClient.GetFrame();
Output_GetGlobalForceVector Output = MyClient.GetGlobalForceVector( 0 );
```

### MATLAB example

```
A valid ForcePlateIndex is between 0 and GetForcePlateCount() - 1
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData ();
MyClient.GetFrame();
Output = MyClient.GetGlobalForceVector( 0 );
```

### .NET example

```
A valid ForcePlateIndex is between 0 and GetForcePlateCount() - 1
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableUnlabeledMarkerData();
MyClient.GetFrame();
Output_GetGlobalForceVector Output = MyClient.GetGlobalForceVector( 0 );
```

### Parameters

<i>ForcePlate-Index</i>	The index of the force plate
-------------------------	------------------------------

## Returns

An `Output_GetGlobalForceVector` class containing the result of the operation and the force on the force plate

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidIndex

## Output\_GetGlobalMomentVector GetGlobalMomentVector ( const unsigned int ForcePlateIndex ) const

Return the moment vector for the force plate in global coordinates.

The vector is in Newton-meters and is with respect to the global coordinate system regardless of the orientation of the force plate. The vector represents the moment exerted upon the force plate, not the reaction moment. Any force plate origin offset is accounted for in the moments so they are acting about the exact center of the top surface of the force plate. If multiple subsamples are available, this function returns the first subsample. See the alternate version of this function to access all of the analog data.

See Also: [GetGlobalForceVector\(\)](#), [GetGlobalCentreOfPressure\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableDeviceData ( pClient );
Client_GetFrame( pClient );
COutput_GetGlobalMomentVector _Output_GetMomentVector;
Client_GetGlobalMomentVector( pClient, 0, &_Output_GetMomentVector );
Client_Destroy( pClient );
```

### C++ example

```
A valid ForcePlateIndex is between 0 and GetForcePlateCount()-1
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData ();
MyClient.GetFrame();
Output_GetGlobalMomentVector Output = MyClient.GetGlobalMomentVector( 0 );
```

### MATLAB example

```
A valid ForcePlateIndex is between 0 and GetForcePlateCount() - 1
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData ();
MyClient.GetFrame();
Output = MyClient.GetGlobalMomentVector( 0 );
```

### .NET example

```
A valid ForcePlateIndex is between 0 and GetForcePlateCount() - 1
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData ();
MyClient.GetFrame();
Output_GetGlobalMomentVector Output = MyClient.GetGlobalMomentVector( 0 );
```

### Parameters

<i>ForcePlate-Index</i>	The index of the force plate
-------------------------	------------------------------

## Returns

An `Output_GetGlobalMomentVector` class containing the result of the operation and the moment exerted on the force plate

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidIndex



## Output\_GetGlobalCentreOfPressure GetGlobalCentreOfPressure ( const unsigned int *ForcePlateIndex* ) const

Return the center of pressure for the force plate in global coordinates.

The position is in millimeters and is with respect to the global coordinate system. If multiple subsamples are available this function returns the first subsample. See the alternate version of this function to access all of the analog data.

See Also: [GetGlobalForceVector\(\)](#), [GetGlobalMomentVector\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_EnableDeviceData ( pClient );
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
COutput_GetGlobalCentreOfPressure _Output_GetCentreOfPressure;
Client_GetGlobalCentreOfPressure( pClient, 0, &_Output_GetCentreOfPressure );
Client_Destroy( pClient );
```

### C++ example

```
A valid ForcePlateIndex is between 0 and GetForcePlateCount()-1
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData ();
MyClient.GetFrame();
Output_GetGlobalCentreOfPressure Output = MyClient.GetGlobalCentreOfPressure( 0 );
```

### MATLAB example

```
A valid ForcePlateIndex is between 0 and GetForcePlateCount() - 1
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData ();
MyClient.GetFrame();
Output = MyClient.GetGlobalCentreOfPressure( 0 );
```

### .NET example

```
A valid ForcePlateIndex is between 0 and GetForcePlateCount() - 1
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData ();
MyClient.GetFrame();
Output_GetGlobalCentreOfPressure Output = MyClient.GetGlobalCentreOfPressure( 0 );
```

### Parameters

<i>ForcePlate-Index</i>	The index of the force plate
-------------------------	------------------------------

### Returns

An Output\_GetGlobalCentreOfPressure class containing the result of the operation and the center of pressure.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidIndex

## Output\_GetForcePlateSubsamples GetForcePlateSubsamples ( const unsigned int ForcePlateIndex ) const

Return the number of subsamples available for a specified force plate in the current frame.

Additional versions of GetGlobalForceVector, GetGlobalMomentVector, and GetGlobalCentreOfPressure take the subsample index to allow access to all the force plate data.

See Also: [GetGlobalForceVector\(\)](#), [GetGlobalMomentVector\(\)](#), [GetGlobalCentreOfPressure\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableDeviceData ( pClient );
Client_GetFrame( pClient );
COutput_GetForcePlateSubsamples ForcePlateSubsamples;
Client_GetForcePlateSubsamples( pClient, 0, &ForcePlateSubsamples );
// ForcePlateSubsamples.Result == Success
// ForcePlateSubsamples.ForcePlateSubsamples >= 0
Client_Destroy( pClient );
```

### C++ example

```
A valid ForcePlateIndex is between 0 and GetForcePlateCount()-1
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.EnableDeviceData();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetForcePlateSubsamples Output = MyClient.GetForcePlateSubsamples ( 0 );
// Output.Result == Success
// Output.ForcePlateSubsamples >= 0
```

### MATLAB example

```
A valid ForcePlateIndex is between 0 and GetForcePlateCount() - 1
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.EnableDeviceData();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output = MyClient.GetForcePlateSubsamples( 0 );
// Output.Result == Success
// Output.ForcePlateSubsamples >= 0
```

### .NET example

```
A valid ForcePlateIndex is between 0 and GetForcePlateCount()-1
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.EnableDeviceData();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetForcePlateSubsamples Output = MyClient.GetForcePlateSubsamples ( 0 );
// Output.Result == Success
// Output.ForcePlateSubsamples >= 0
```

### Parameters

<i>ForcePlate-Index</i>	The index of the force plate
-------------------------	------------------------------

## Returns

An `Output_GetForcePlateSubsamples` class containing the result of the operation and the number of subsamples.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidIndex

## Output\_GetGlobalForceVector GetGlobalForceVector ( const unsigned int ForcePlateIndex, const unsigned int Subsample ) const

Return the force vector for the force plate in global coordinates.

This version takes a subsample index that allows access to all of the force information. The vector is in Newtons and is with respect to the global coordinate system, regardless of the orientation of the plate. The vector represents the force exerted upon the force plate, not the reaction force.

See Also: [GetGlobalMomentVector\(\)](#), [GetGlobalCentreOfPressure\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableDeviceData ( pClient );
Client_GetFrame( pClient );
unsigned int Index(0);
COutput_GetForcePlateSubsamples ForcePlateSubsamples;
Client_GetForcePlateSubsamples( pClient, Index, &ForcePlateSubsamples );
for ( unsigned int ForcePlateSubsample = 0;
      ForcePlateSubsample < ForcePlateSubsamples.ForcePlateSubsamples; ++ForcePlateSubsample)
{
    COutput_GetGlobalForceVector _Output_GetForceVector;
    Client_GetGlobalForceVectorForSubsample(
        pClient, Index, ForcePlateSubsample, &_Output_GetForceVector );
}
Client_Destroy( pClient );
```

### C++ example

```
A valid ForcePlateIndex is between 0 and GetForcePlateCount()-1
A valid Subsample is between 0 and GetForcePlateSubsamples()-1
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData ();
MyClient.GetFrame();
const unsigned int Index(0);
const unsigned int Samples = MyClient.GetForcePlateSubsamples( index ).ForcePlateSubsamples;
for( unsigned int Sample = 0; Sample < Samples; ++ Sample)
{
    Output_GetGlobalForceVector Output = MyClient.GetGlobalForceVector( Index, Sample );
}
```

### MATLAB example

```
A valid ForcePlateIndex is between 0 and GetForcePlateCount() - 1
A valid Subsample is between 0 and GetForcePlateSubsamples() - 1
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData ();
MyClient.GetFrame();
Index = 0;
Output_GetForcePlateSubsamples = MyClient.GetForcePlateSubsamples(Index );
for Sample = 1:Output_GetForcePlateSubsamples.ForcePlateSubsamples
    Output = MyClient.GetGlobalForceVector( Index, Sample );
end
```

### .NET example

```
A valid ForcePlateIndex is between 0 and GetForcePlateCount()-1
```

## Class Documentation

```
A valid Subsample is between 0 and GetForcePlateSubsamples()-1
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableUnlabeledMarkerData();
MyClient.GetFrame();
uint Index = 0;
uint Samples = MyClient.GetForcePlateSubsamples(ForcePlateIndex).ForcePlateSubsamples;
for (uint Sample = 0; Sample < Samples; ++ Sample)
{
    Output_GetGlobalForceVector Output = MyClient.GetGlobalForceVector( Index, Sample );
}
```

### Parameters

<i>ForcePlate-Index</i>	The index of the force plate
<i>Subsample</i>	The subsample to access

### Returns

An Output\_GetGlobalForceVector class containing the result of the operation and the force on the forceplate.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidIndex

## Output\_GetGlobalMomentVector GetGlobalMomentVector ( const unsigned int ForcePlateIndex, const unsigned int Subsample ) const

Return the moment vector for the force plate in global coordinates.

This version takes a subsample index that allows access to all of the force information. The vector is in Newton-meters and is with respect to the global coordinate system, regardless of the orientation of the plate. The vector represents the moment exerted upon the force plate, not the reaction moment. Any force plate origin offset is accounted for in the moments so they are acting about the exact center of the top surface of the force plate.

See Also: [GetGlobalForceVector\(\)](#), [GetGlobalCentreOfPressure\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_EnableDeviceData ( pClient );
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
unsigned int Index(0);
COutput_GetForcePlateSubsamples ForcePlateSubsamples;
Client_GetForcePlateSubsamples( pClient, Index, &ForcePlateSubsamples );
for ( unsigned int ForcePlateSubsample = 0;
      ForcePlateSubsample < ForcePlateSubsamples.ForcePlateSubsamples; ++ForcePlateSubsample)
{
    COutput_GetGlobalMomentVector _Output_GetMomentVector;
    Client_GetGlobalMomentVectorForSubsample(
        pClient, Index, ForcePlateSubsample, &_Output_GetMomentVector);
}
Client_Destroy( pClient );
```

### C++ example

```
A valid ForcePlateIndex is between 0 and GetForcePlateCount()-1
A valid Subsample is between 0 and GetForcePlateSubsamples()-1
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData ();
MyClient.GetFrame();
const unsigned int Index(0);
const unsigned int Samples = MyClient.GetForcePlateSubsamples( index ).ForcePlateSubsamples;
for( unsigned int Sample = 0; Sample < Samples; ++ Sample)
{
    Output_GetGlobalMomentVector Output = MyClient.GetGlobalMomentVector( Index, Sample );
}
```

### MATLAB example

```
A valid ForcePlateIndex is between 0 and GetForcePlateCount() - 1
A valid Subsample is between 0 and GetForcePlateSubsamples() - 1
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData ();
MyClient.GetFrame();
Index = 0;
Output_GetForcePlateSubsamples = MyClient.GetForcePlateSubsamples( Index );
for Sample = 1:Output_GetForcePlateSubsamples.ForcePlateSubsamples
    Output = MyClient.GetGlobalMomentVector ( Index, Sample );
end
```

### .NET example

## Class Documentation

```
A valid ForcePlateIndex is between 0 and GetForcePlateCount() - 1
A valid Subsample is between 0 and GetForcePlateSubsamples()-1
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData ();
MyClient.GetFrame();
uint Index = 0;
uint Samples = MyClient.GetForcePlateSubsamples(ForcePlateIndex).ForcePlateSubsamples;
for (uint Sample = 0; Sample < Samples; ++ Sample)
{
    Output_GetGlobalMomentVector Output = MyClient.GetGlobalMomentVector( Index, Sample );
}
```

### Parameters

<i>ForcePlate-Index</i>	The index of the force plate
<i>Subsample</i>	The subsample to access

### Returns

An `Output_GetGlobalMomentVector` class containing the result of the operation and the moment exerted on the force plate.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidIndex



## Output\_GetGlobalCentreOfPressure GetGlobalCentreOfPressure ( const unsigned int *ForcePlateIndex*, const unsigned int *Subsample* ) const

Return the center of pressure for the force plate in global coordinates.

This version takes a subsample index that allows access to all of the force information. The position is in millimeters and is with respect to the global coordinate system.

See Also: [GetGlobalForceVector\(\)](#), [GetGlobalMomentVector\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableDeviceData ( pClient );
Client_GetFrame( pClient );
unsigned int Index(0);
COutput_GetForcePlateSubsamples ForcePlateSubsamples;
Client_GetForcePlateSubsamples( pClient, Index, &ForcePlateSubsamples );
for ( unsigned int ForcePlateSubsample = 0;
      ForcePlateSubsample < ForcePlateSubsamples.ForcePlateSubsamples; ++ForcePlateSubsample)
{
    COutput_GetGlobalCentreOfPressure _Output_GetCentreOfPressure;
    Client_GetGlobalCentreOfPressureForSubsample(
        pClient, Index, ForcePlateSubsample, &_Output_GetCentreOfPressure);
}
Client_Destroy( pClient );
```

### C++ example

```
A valid ForcePlateIndex is between 0 and GetForcePlateCount()-1
A valid Subsample is between 0 and GetForcePlateSubsamples()-1
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData ();
MyClient.GetFrame();
const unsigned int Index(0);
const unsigned int Samples = MyClient.GetForcePlateSubsamples( index ).ForcePlateSubsamples;
for( unsigned int Sample = 0; Sample < Samples; ++ Sample)
{
    Output_GetGlobalCentreOfPressure Output = MyClient.GetGlobalCentreOfPressure(Index,Sample);
}
```

### MATLAB example

```
A valid ForcePlateIndex is between 0 and GetForcePlateCount() - 1
A valid Subsample is between 0 and GetForcePlateSubsamples() - 1
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Client_GetFrame( pClient );
MyClient.EnableDeviceData ();
MyClient.GetFrame();
Index = 0;
Output_GetForcePlateSubsamples = MyClient.GetForcePlateSubsamples( Index );
for Sample = 1:Output_GetForcePlateSubsamples.ForcePlateSubsamples
    Output = MyClient.GetGlobalCentreOfPressure( Index, Sample );
end
```

### .NET example

```
A valid ForcePlateIndex is between 0 and GetForcePlateCount()-1
```

## Class Documentation

```
A valid Subsample is between 0 and GetForcePlateSubsamples()-1
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableDeviceData ();
MyClient.GetFrame();
uint Index = 0;
uint Samples = MyClient.GetForcePlateSubsamples(ForcePlateIndex).ForcePlateSubsamples;
for (uint Sample = 0; Sample < Samples; ++ Sample)
{
    Output_GetGlobalCentreOfPressure Output = MyClient.GetGlobalCentreOfPressure (Index,Sample);
}
```

### Parameters

<i>ForcePlate-Index</i>	The index of the force plate
<i>Subsample</i>	The subsample to access

### Returns

An `Output_GetGlobalCentreOfPressure` class containing the result of the operation the center of pressure

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidIndex

## Output\_GetEyeTrackerCount GetEyeTrackerCount ( ) const

Return the number of eye trackers available in the DataStream.

See Also: [GetEyeTrackerGlobalGazeVector\(\)](#), [GetEyeTrackerGlobalGazeVector\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_EnableDeviceData( pClient );
Client_Connect( pClient, "localhost" );
Client_GetFrame( pClient );
COutput_GetEyeTrackerCount EyeTrackerCount;
Client_GetEyeTrackerCount(pClient, &EyeTrackerCount);
// EyeTrackerCount.Result == Success
// EyeTrackerCount.EyeTrackerCount >= 0
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.EnableDeviceData();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetDeviceCount Output = MyClient.GetEyeTrackerCount ();
// Output.Result == Success
// Output.EyeTrackerCount >= 0
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.EnableDeviceData();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output = MyClient.GetEyeTrackerCount();
// Output.Result == Success
// Output.EyeTrackerCount >= 0
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.EnableDeviceData();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetEyeTrackerCount Output = MyClient.GetEyeTrackerCount();
// Output.Result == Success
// Output.EyeTrackerCount >= 0
```

### Returns

An `Output_GetEyeTrackerCount` class containing the result of the operation and the number of eye trackers.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame

## Output\_GetEyeTrackerGlobalPosition GetEyeTrackerGlobalPosition ( const unsigned int *EyeTrackerIndex* ) const

Return the location of the eye.

The position is in millimeters with respect to the global origin. The segment and device data need to be enabled to get the position.

See Also: [GetEyeTrackerCount\(\)](#), [GetEyeTrackerGlobalGazeVector\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableDeviceData( pClient );
Client_GetFrame( pClient );
COutput_GetEyeTrackerGlobalPosition _Output_GetEyeTrackerGlobalPosition;
Client_GetEyeTrackerGlobalPosition(pClient, 0, &_Output_GetEyeTrackerGlobalPosition);
Client_Destroy( pClient );
```

### C++ example

```
A valid EyeTrackerIndex is between 0 and GetEyeTrackerCount()-1
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableSegmentData ();
MyClient.EnableDeviceData ();
MyClient.GetFrame();
Output_GetEyeTrackerGlobalPosition Output = MyClient.GetEyeTrackerGlobalPosition ( 0 );
```

### MATLAB example

```
A valid EyeTrackerIndex is between 0 and GetEyeTrackerCount() - 1
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableSegmentData ();
MyClient.EnableDeviceData ();
MyClient.GetFrame();
Output = MyClient.GetEyeTrackerGlobalPosition ( 0 );
```

### .NET example

```
A valid EyeTrackerIndex is between 0 and GetEyeTrackerCount() - 1
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableSegmentData ();
MyClient.EnableDeviceData ();
MyClient.GetFrame();
Output_GetEyeTrackerGlobalPosition Output = MyClient.GetEyeTrackerGlobalPosition ( 0 );
```

### Parameters

<i>EyeTrackerIndex</i>	The index of the eye tracker
------------------------	------------------------------

## Returns

An `Output_GetEyeTrackerGlobalPosition` class containing the result of the operation, the eye position and whether the eye tracker is occluded.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidIndex
- Occluded will be true if the segment that has the eye tracker attached is not visible. If true the position will be (0,0,0).

## Output\_GetEyeTrackerGlobalGazeVector GetEyeTrackerGlobalGazeVector ( const unsigned int *EyeTrackerIndex* ) const

Return the gaze direction as a unit vector in global coordinates.

The gaze vector will be marked as occluded if the segment that has the eye tracker attached is not visible, the eye tracker is not calibrated or the pupil is not found. The segment and device data need to be enabled to get the gaze vector.

See Also: [GetEyeTrackerCount\(\)](#), [GetEyeTrackerGlobalPosition\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableDeviceData( pClient );
Client_GetFrame( pClient );
COutput_GetEyeTrackerGlobalGazeVector _Output_GetEyeTrackerGlobalGazeVector;
Client_GetEyeTrackerGlobalGazeVector(pClient, 0, &_Output_GetEyeTrackerGlobalGazeVector);
Client_Destroy( pClient );
```

### C++ example

```
A valid EyeTrackerIndex is between 0 and GetEyeTrackerCount()-1
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableSegmentData ();
MyClient.EnableDeviceData ();
MyClient.GetFrame();
Output_GetEyeTrackerGlobalPosition Output = MyClient.GetEyeTrackerGlobalGazeVector ( 0 );
```

### MATLAB example

```
A valid EyeTrackerIndex is between 0 and GetEyeTrackerCount() - 1
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableSegmentData ();
MyClient.EnableDeviceData ();
MyClient.GetFrame();
Output = MyClient.GetEyeTrackerGlobalGazeVector ( 0 );
```

### .NET example

```
A valid EyeTrackerIndex is between 0 and GetEyeTrackerCount() - 1
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableSegmentData ();
MyClient.EnableDeviceData ();
MyClient.GetFrame();
Output_GetEyeTrackerGlobalPosition Output = MyClient.GetEyeTrackerGlobalPosition ( 0 );
```

### Parameters

<i>EyeTracker-Index</i>	The index of the eye tracker
-------------------------	------------------------------

## Returns

An `Output_GetEyeTrackerGlobalGazeVector` class containing the result of the operation, the gaze direction vector, and whether the eye tracker is occluded.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidIndex
- Occluded will be true if the gaze vector could not be calculated. If true, the position will be (0,0,0).

## Output\_GetCameraCount GetCameraCount ( ) const

Return the number of cameras available in the DataStream.

See Also: [GetCameraName\(\)](#), [GetCentroidCount\(\)](#), [GetCentroidPosition\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableCentroidData( pClient );
COutput_GetCameraCount CameraCount;
Client_GetCameraCount( pClient, &CameraCount );
// CameraCount.Result == Success
// CameraCount.CameraCount >= 0
Client_GetFrame( pClient );
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableCentroidData ();
MyClient.GetFrame();
Output_GetCameraCount Output = MyClient.GetCameraCount();
// Output.Result == Success
// Output.CameraCount >= 0
```

### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableCentroidData();
MyClient.GetFrame();
Output = MyClient.GetCameraCount();
% Output.Result == Success, Output.CameraCount >= 0
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableCentroidData ();
MyClient.GetFrame();
Output_GetCameraCount Output = MyClient.GetCameraCount();
// Output.Result == Success
// Output.CameraCount >= 0
```

### Returns

An `Output_GetCameraCount` class containing the result of the operation and the number of cameras.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame



## Output\_GetCameraName GetCameraName ( unsigned int *CameraIndex* ) const

Return the name of a camera.

This name can be passed into centroid functions.

See Also: [GetCameraCount\(\)](#), [GetCentroidCount\(\)](#), [GetCentroidPosition\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableCentroidData ( pClient );
Client_GetFrame( pClient );
COutput_GetCameraCount CameraCount;
Client_GetCameraCount(pClient, &CameraCount);
// CamerCount.Result == Success
// CameraCount.CameraCount == 1
char CameraName[128];
Client_GetCameraName(pClient, 0, 128, CameraName);
Client_Destroy( pClient );
```

### C++ example

```
A valid CameraIndex is between 0 and GetCameraCount()-1
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableCentroidData ();
MyClient.GetFrame();
Output_GetCameraCount OutputGCC = MyClient.GetCameraCount();
// OutputGCC.Result == Success
// OutputGCC.CameraCount == 1
Output_GetCameraName OutputGCN;
OutputGCN = MyClient.GetCameraName( 0 )
```

### MATLAB example

```
A valid CameraIndex is between 0 and GetCameraCount() - 1
% [Output] = GetCameraName ( CameraIndex )
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableCentroidData ();
MyClient.GetFrame();
OutputGCC = MyClient.GetCameraCount ( 0 );
% OutputGCC.Result == Success
% OutputGCC.CameraCount == 1
OutputGCN = MyClient.GetCameraName( 0 );
```

### .NET example

```
A valid CameraIndex is between 0 and GetCameraCount() - 1
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableCentroidData ();
MyClient.GetFrame();
Output_GetCameraCount OutputGCC = MyClient.GetCameraCount();
// OutputGCC.Result == Success
// OutputGCC.CameraCount == 1
Output_GetCameraName OutputGCN;
OutputGCN = MyClient.GetCameraName( 0 )
```

## Parameters

<i>CameraIndex</i>	The index of the camera
--------------------	-------------------------

## Returns

An `Output_GetCameraName` class containing the result of the operation and the name of the camera.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidIndex

## Output\_GetCameraId GetCameraId ( const std::string & CameraName ) const

Returns the internal ID of the camera with the specified name.

See Also: [GetCameraName\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableCentroidData ( pClient );
COutput_GetCameraCount CameraCount;
Client_GetCameraCount(pClient, &CameraCount);
if( CameraCount.Result == Success && CameraCount.CameraCount > 0 )
{
    char CameraName[128];
    CEnum Output = Client_GetCameraName(pClient, 0, 128, CameraName);
    if ( Output == Success )
    {
        COutput_GetCameraId CameraId;
        Client_GetCameraId(pClient, CameraName, &CameraId );
    }
}
Client_GetFrame( pClient );
Client_Destroy( pClient );
```

### C++ example

```
A valid CameraName may be obtained from GetCameraName()
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableCentroidData ();
MyClient.GetFrame();
Output_GetCameraCount OutputGCC = MyClient.GetCameraCount();
if( OutputGCC.Result == Success && OutputGCC.CameraCount > 0 )
{
    Output_GetCameraName OutputGCN;
    OutputGCN = MyClient.GetCameraName( 0 );
    if( OutputGCN.Result == Success )
    {
        Output_GetCameraId Output_GCI = MyClient.GetCameraId( OutputGCN.CameraName );
    }
}
```

### MATLAB example

Not implemented

### .NET example

```
A valid CameraName may be obtained from GetCameraName()
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableCentroidData ();
MyClient.GetFrame();
Output_GetCameraCount OutputGCC = MyClient.GetCameraCount();
if( OutputGCC.Result == Success && OutputGCC.CameraCount > 0 )
{
    Output_GetCameraName OutputGCN;
    OutputGCN = MyClient.GetCameraName( 0 );
    if( OutputGCN.Result == Success )
    {
        Output_GetCameraId Output_GCI = MyClient.GetCameraId( OutputGCN.CameraName );
    }
}
```

```
}  
}
```

## Returns

An Output\_GetCameraId class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected
  - InvalidCameraName

## Output\_GetCameraUserId GetCameraUserId ( const std::string & CameraName ) const

Returns the user-assigned ID of the camera with the specified name.

See Also: [GetCameraName\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableCentroidData ( pClient );
Client_GetFrame( pClient );
COutput_GetCameraCount CameraCount;
Client_GetCameraCount(pClient, &CameraCount);
if( CameraCount.Result == Success && CameraCount.CameraCount > 0 )
{
    char CameraName[128];
    CEnum Output = Client_GetCameraName(pClient, 0, 128, CameraName);
    if ( Output == Success )
    {
        COutput_GetCameraUserId CameraId;
        Client_GetCameraUserId(pClient, CameraName, &CameraUserId );
    }
}
Client_Destroy( pClient );
```

### C++ example

```
A valid CameraName may be obtained from GetCameraName()
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableCentroidData ();
MyClient.GetFrame();
Output_GetCameraCount OutputGCC = MyClient.GetCameraCount();
if( OutputGCC.Result == Success && OutputGCC.CameraCount > 0 )
{
    Output_GetCameraName OutputGCN;
    OutputGCN = MyClient.GetCameraName( 0 );
    if( OutputGCN.Result == Success )
    {
        Output_GetCameraUserId Output_GCI = MyClient.GetCameraUserId( OutputGCN.CameraName );
    }
}
```

### MATLAB example

Not implemented

### .NET example

```
A valid CameraName may be obtained from GetCameraName()
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableCentroidData ();
MyClient.GetFrame();
Output_GetCameraCount OutputGCC = MyClient.GetCameraCount();
if( OutputGCC.Result == Success && OutputGCC.CameraCount > 0 )
{
    Output_GetCameraName OutputGCN;
    OutputGCN = MyClient.GetCameraName( 0 );
    if( OutputGCN.Result == Success )
    {

```

```
Output_GetCameraUserId Output_GCI = MyClient.GetCameraUserId( OutputGCN.CameraName );  
}  
}
```

### Returns

An Output\_GetCameraUserId class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected
  - InvalidCameraName

## Output\_GetCameraType GetCameraType ( const std::string & CameraName ) const

Returns the type of the camera with the specified name.

The type returned is an internal type string.

See Also: [GetCameraName\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableCentroidData ( pClient );
Client_GetFrame( pClient );
COutput_GetCameraCount CameraCount;
Client_GetCameraCount(pClient, &CameraCount);
if( CameraCount.Result == Success && CameraCount.CameraCount > 0 )
{
    char CameraName[128];
    CEnum Output = Client_GetCameraName(pClient, 0, 128, CameraName);
    if ( Output == Success )
    {
        char CameraType[128];
        CEnum Result = Client_GetCameraType( pClient, CameraName, 128, CameraType );
    }
}
Client_Destroy( pClient );
```

### C++ example

```
A valid CameraName may be obtained from GetCameraName()
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableCentroidData ();
MyClient.GetFrame();
Output_GetCameraCount OutputGCC = MyClient.GetCameraCount();
if( OutputGCC.Result == Success && OutputGCC.CameraCount > 0 )
{
    Output_GetCameraName OutputGCN;
    OutputGCN = MyClient.GetCameraName( 0 );
    if( OutputGCN.Result == Success )
    {
        Output_GetCameraType Output_GCT = MyClient.GetCameraType( OutputGCN.CameraName );
    }
}
```

### MATLAB example

Not implemented

### .NET example

```
A valid CameraName may be obtained from GetCameraName()
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableCentroidData ();
MyClient.GetFrame();
Output_GetCameraCount OutputGCC = MyClient.GetCameraCount();
if( OutputGCC.Result == Success && OutputGCC.CameraCount > 0 )
{
    Output_GetCameraName OutputGCN;
    OutputGCN = MyClient.GetCameraName( 0 );
    if( OutputGCN.Result == Success )
```

## Class Documentation

```
{  
    Output_GetCameraType Output_GCT = MyClient.GetCameraType( OutputGCN.CameraName );  
}
```

### Returns

An Output\_ class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected
  - InvalidCameraName



## Output\_GetCameraDisplayName GetCameraDisplayName ( const std::string & CameraName ) const

Returns the name of of the camera type as a string suitable for display to a user.

See Also: [GetCameraName\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableCentroidData ( pClient );
Client_GetFrame( pClient );
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableCentroidData ( pClient );
Client_GetFrame( pClient );
COutput_GetCameraCount CameraCount;
Client_GetCameraCount(pClient, &CameraCount);
if( CameraCount.Result == Success && CameraCount.CameraCount > 0 )
{
    char CameraName[128];
    CEnum Output = Client_GetCameraName(pClient, 0, 128, CameraName);
    if ( Output == Success )
    {
        char CameraDisplayName[128];
        CEnum Result = Client_GetCameraDisplayName( pClient, CameraName, 128, CameraDisplayName );
    }
}
Client_Destroy( pClient );
```

### C++ example

```
A valid CameraName may be obtained from GetCameraName()
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableCentroidData ();
MyClient.GetFrame();
Output_GetCameraCount OutputGCC = MyClient.GetCameraCount();
if( Output_GCC.Result == Success && OutputGCC.CameraCount > 0 )
{
    Output_GetCameraName OutputGCN;
    OutputGCN = MyClient.GetCameraName( 0 );
    if( OutputGCN.Result == Success )
    {
        Output_GetCameraDisplayName Output_GCD = MyClient.GetCameraDisplayName( OutputGCN.CameraName );
    }
}
```

### MATLAB example

Not implemented

### .NET example

```
A valid CameraName may be obtained from GetCameraName()
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableCentroidData ();
MyClient.GetFrame();
Output_GetCameraCount OutputGCC = MyClient.GetCameraCount();
if( Output_GCC.Result == Success && OutputGCC.CameraCount > 0 )
{
```

## Class Documentation

```
Output_GetCameraName OutputGCN;  
OutputGCN = MyClient.GetCameraName( 0 );  
if( OutputGCN.Result == Success )  
{  
    Output_GetCameraDisplayName Output_GCD = MyClient.GetCameraDisplayName( OutputGCN.CameraName );  
}  
}
```

### Returns

An Output\_ class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected
  - InvalidCameraName

## Output\_GetCameraResolution GetCameraResolution ( const std::string & CameraName ) const

Returns the sensor resolution of the camera with the specified name.

See Also: [GetCameraName\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableCentroidData ( pClient );
Client_GetFrame( pClient );
COutput_GetCameraCount CameraCount;
Client_GetCameraCount(pClient, &CameraCount);
if( CameraCount.Result == Success && CameraCount.CameraCount > 0 )
{
    char CameraName[128];
    CEnum Output = Client_GetCameraName(pClient, 0, 128, CameraName);
    if ( Output == Success )
    {
        COutput_GetCameraResolution CameraResolution;
        Client_GetCameraResolution(pClient, CameraName, &CameraResolution );
    }
}
Client_Destroy( pClient );
```

### C++ example

```
A valid CameraName may be obtained from GetCameraName()
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableCentroidData ();
MyClient.GetFrame();
Output_GetCameraCount OutputGCC = MyClient.GetCameraCount();
if( OutputGCC.Result == Success && OutputGCC.CameraCount > 0 )
{
    Output_GetCameraName OutputGCN;
    OutputGCN = MyClient.GetCameraName( 0 );
    if( OutputGCN.Result == Success )
    {
        Output_GetCameraResolution Output_GCR = MyClient.GetCameraResolution( OutputGCN.CameraName );
    }
}
```

### MATLAB example

Not implemented

### .NET example

```
A valid CameraName may be obtained from GetCameraName()
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableCentroidData ();
MyClient.GetFrame();
Output_GetCameraCount OutputGCC = MyClient.GetCameraCount();
if( OutputGCC.Result == Success && OutputGCC.CameraCount > 0 )
{
    Output_GetCameraName OutputGCN;
    OutputGCN = MyClient.GetCameraName( 0 );
    if( OutputGCN.Result == Success )
    {

```

```
Output_GetCameraResolution Output_GCR = MyClient.GetCameraResolution( OutputGCN.CameraName );  
}  
}
```

### Returns

An Output\_ class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected
  - InvalidCameraName

## Output\_GetIsVideoCamera GetIsVideoCamera ( const std::string & CameraName ) const

Returns whether the camera with the specified name is a video camera.

See Also: [GetCameraName\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableCentroidData ( pClient );
Client_GetFrame( pClient );
COutput_GetCameraCount CameraCount;
Client_GetCameraCount(pClient, &CameraCount);
if( CameraCount.Result == Success && CameraCount.CameraCount > 0 )
{
    char CameraName[128];
    CEnum Output = Client_GetCameraName(pClient, 0, 128, CameraName);
    if ( Output == Success )
    {
        COutput_GetIsVideoCamera IsVideoCamera;
        Client_GetIsVideoCamera(pClient, CameraName, &IsVideoCamera );
    }
}
Client_Destroy( pClient );
```

### C++ example

```
A valid CameraName may be obtained from GetCameraName()
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableCentroidData ();
MyClient.GetFrame();
Output_GetCameraCount OutputGCC = MyClient.GetCameraCount();
if( OutputGCC.Result == Success && OutputGCC.CameraCount > 0 )
{
    Output_GetCameraName OutputGCN;
    OutputGCN = MyClient.GetCameraName( 0 );
    if( OutputGCN.Result == Success )
    {
        Output_GetIsVideoCamera Output_GCV = MyClient.GetIsVideoCamera( OutputGCN.CameraName );
    }
}
```

### MATLAB example

Not implemented

### .NET example

```
A valid CameraName may be obtained from GetCameraName()
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableCentroidData ();
MyClient.GetFrame();
Output_GetCameraCount OutputGCC = MyClient.GetCameraCount();
if( OutputGCC.Result == Success && OutputGCC.CameraCount > 0 )
{
    Output_GetCameraName OutputGCN;
    OutputGCN = MyClient.GetCameraName( 0 );
    if( OutputGCN.Result == Success )
    {

```

## Class Documentation

```
Output_GetIsVideoCamera Output_GCV = MyClient.GetIsVideoCamera ( OutputGCN.CameraName );  
}  
}
```

### Returns

An Output\_ class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected
  - InvalidCameraName

## Class Documentation

### Output\_GetCentroidCount GetCentroidCount ( const std::string & CameraName ) const

Return the number of centroids reported by a named camera.

The centroid data needs to be enabled to get the number of centroids.

See Also: [GetCameraCount\(\)](#), [GetCameraName\(\)](#), [GetCentroidPosition\(\)](#)

#### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableCentroidData ( pClient );
Client_GetFrame( pClient );
COutput_GetCameraCount CameraCount;
Client_GetCameraCount(pClient, &CameraCount);
if( CameraCount.Result == Success && CameraCount.CameraCount > 0 )
{
    char CameraName[128];
    CEnum Output = Client_GetCameraName(pClient, 0, 128, CameraName);
    if ( Output == Success )
    {
        COutput_GetCentroidCount CentroidCount;
        Client_GetCentroidCount(pClient, CameraName, &CentroidCount );
    }
}
Client_Destroy( pClient );
```

#### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableCentroidData ();
MyClient.GetFrame();
Output_GetCameraCount OutputGCC = MyClient.GetCameraCount();
for( unsigned int CameraIndex = 0; CameraIndex < OutputGCC.CameraCount; ++CameraIndex )
{
    Output_GetCameraName OutputGCN = MyClient.GetCameraName( CameraIndex );
    Output_GetCentroidCount OutputGCeC = MyClient.GetCentroidCount( OutputGCN.CameraName );
    // OutputGCeC.Result == Success
    // OutputGCeC.CentroidCount >= 0
}
```

#### MATLAB example

```
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableCentroidData();
MyClient.GetFrame();
OutputGCC = MyClient.GetCameraCount();
for CameraIndex = 0:OutputGCC.CameraCount - 1
    OutputGCN = MyClient.GetCameraName( CameraIndex );
    OutputGCeC = MyClient.GetCentroidCount( OutputGCN.CameraName )
    % OutputGCeC.Result == Success
    % OutputGCeC.CentroidCount >= 0
End
```

#### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
```

## Class Documentation

```
MyClient.EnableCentroidData ();
MyClient.GetFrame();
Output_GetCameraCount OutputGCC = MyClient.GetCameraCount();
for( unsigned int CameraIndex = 0; CameraIndex < OutputGCC.CameraCount; ++CameraIndex )
{
    OutputGCN = MyClient.GetCameraName( CameraIndex );
    OutputGCeC = MyClient.GetCentroidCount( OutputGCN.CameraName )
    // OutputGCeC.Result == Success
    // OutputGCeC.CentroidCount >= 0
}
```

### Parameters

<i>CameraName</i>	The name of the camera.
-------------------	-------------------------

### Returns

An `Output_GetCentroidCount` class containing the result of the operation and the number of centroids.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidCameraName



## Output\_GetCentroidPosition GetCentroidPosition ( const std::string & *CameraName*, const unsigned int *CentroidIndex* ) const

Return the position and radius of the centroid in camera coordinates.

The centroid data needs to be enabled to get the centroid position and radius.

See Also: [GetCameraCount\(\)](#), [GetCameraName\(\)](#), [GetCentroidCount\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableCentroidData ( pClient );
Client_GetFrame( pClient );
char CameraName[128];
CEnum Output = Client_GetCameraName(pClient, 0, 128, CameraName);
COutput_GetCentroidPosition CentroidPosition;
Client_GetCentroidPosition(pClient, CameraName, 0, &CentroidPosition );
Client_Destroy( pClient );
```

### C++ example

```
A valid CameraName is obtained from GetCameraName( CameraIndex )
A valid CentroidIndex is between 0 and GetCentroidCount( CameraName )-1
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableCentroidData ();
MyClient.GetFrame();
Output_GetCameraName OutputGCN = MyClient.GetCameraName( 0 );
Output_GetCentroidPosition Output = MyClient.GetCentroidPosition( OutputGCN.CameraName, 0 );
```

### MATLAB example

```
A valid CameraName is obtained from GetCameraName( CameraIndex )
A valid CentroidIndex is between 0 and GetCentroidCount( CameraName ) - 1
% [Output] = GetCentroidPosition( CameraName, CentroidIndex )
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableCentroidData ();
MyClient.GetFrame();
OutputGCN = MyClient.GetCameraName( 0 );
Output = MyClient.GetCentroidPosition( OutputGCN.CameraName, 0 );
```

### .NET example

```
A valid CameraName is obtained from GetCameraName( CameraIndex )
A valid CentroidIndex is between 0 and GetCentroidCount( CameraName )-1
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableCentroidData ();
MyClient.GetFrame();
Output_GetCameraName OutputGCN = MyClient.GetCameraName( 0 );
Output_GetCentroidPosition Output = MyClient.GetCentroidPosition( OutputGCN.CameraName, 0 );
```

### Parameters

<i>CameraName</i>	The name of the camera.
<i>CentroidIndex</i>	The index of the centroid.

## Returns

An `Output_GetCentroidPosition` class containing the result of the operation, the position of the centroid and the radius of the centroid.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidCameraName
  - InvalidIndex

## Output\_GetCentroidWeight GetCentroidWeight ( const std::string & *CameraName*, const unsigned int *CentroidIndex* ) const

Return the weight of the centroid.

The centroid data needs to be enabled to get the centroid weight. Only supported by Tracker - weights will be 1.0 for all centroids if Low Jitter mode is not enabled.

See Also: [GetCameraCount\(\)](#), [GetCameraName\(\)](#), [GetCentroidCount\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableCentroidData ( pClient );
Client_GetFrame( pClient );
char CameraName[128];
CEnum Output = Client_GetCameraName(pClient, 0, 128, CameraName);
COutput_GetCentroidWeight CentroidWeight;
Client_GetCentroidWeight(pClient, CameraName, 0, &CentroidWeight );
Client_Destroy( pClient );
```

### C++ example

```
A valid CameraName is obtained from GetCameraName( CameraIndex )
A valid CentroidIndex is between 0 and GetCentroidCount( CameraName ) -1
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableCentroidData ();
MyClient.GetFrame();
Output_GetCameraName OutputGCN = MyClient.GetCameraName( 0 );
Output_GetCentroidWeight Output = MyClient.GetCentroidWeight( OutputGCN.CameraName, 0 );
```

### MATLAB example

```
A valid CameraName is obtained from GetCameraName( CameraIndex )
A valid CentroidIndex is between 0 and GetCentroidCount( CameraName ) - 1
% [Output] = GetCentroidWeight( CameraName, CentroidIndex )
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableCentroidData ();
MyClient.GetFrame();
OutputGCN = MyClient.GetCameraName( 1 );
Output = MyClient.GetCentroidWeight( OutputGCN.CameraName, 0 );
```

### .NET example

```
A valid CameraName is obtained from GetCameraName( CameraIndex )
A valid CentroidIndex is between 0 and GetCentroidCount( CameraName ) -1
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableCentroidData ();
MyClient.GetFrame();
Output_GetCameraName OutputGCN = MyClient.GetCameraName( 0 );
Output_GetCentroidWeight Output = MyClient.GetCentroidWeight( OutputGCN.CameraName, 0 );
```

### Parameters

<i>CameraName</i>	The name of the camera.
<i>CentroidIndex</i>	The index of the centroid.

## Returns

An `Output_GetCentroidWeight` class containing the result of the operation and the weight of the centroid.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidCameraName
  - InvalidIndex

## Class Documentation

### Output\_GetGreyscaleBlobCount GetGreyscaleBlobCount ( const std::string & CameraName ) const

Obtain the number of greyscale blobs that are available for the specified camera.

See Also: [GetGreyscaleBlob\(\)](#), [EnableGreyscaleData\(\)](#)

C example

Not implemented

C++ example

```
A valid camera name may be obtained from GetCameraName( CameraIndex )
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableGreyscaleData ();
MyClient.GetFrame();
Output_GetCameraName CameraName = MyClient.GetCameraName( 0 );
Output_GetGreyscaleBlobCount Output = MyClient.GetGreyscaleBlobCount( CameraName.CameraName );
```

MATLAB example

Not implemented

.NET example

```
A valid camera name may be obtained from GetCameraName( CameraIndex )
A valid blob index is between 0 and GetGreyscaleBlobCount() - 1
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableGreyscaleData ();
MyClient.GetFrame();
Output_GetCameraName CameraName = MyClient.GetCameraName( 0 );
Output_GetGreyscaleBlob GreyscaleData = MyClient.GetGreyscaleBlob( CameraName.CameraName, 0 );
```

### Returns

An Output\_GetGreyscaleBlobCount class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected
  - InvalidCameraName

## Class Documentation

`Output_GetGreyscaleBlob GetGreyscaleBlob ( const std::string & CameraName, const unsigned int i_BlobIndex ) const`

Obtains greyscale blob data for the specified camera and blob index.

See Also: [GetGreyscaleBlobCount\(\)](#), [EnableGreyscaleData\(\)](#)

C example

Not implemented

C++ example

```
A valid camera name may be obtained from GetCameraName( CameraIndex )
A valid blob index is between 0 and GetGreyscaleBlobCount() - 1
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableGreyscaleData ();
MyClient.GetFrame();
Output_GetCameraName CameraName = MyClient.GetCameraName( 0 );
Output_GetGreyscaleBlob GreyscaleData = MyClient.GetGreyscaleBlob( CameraName.CameraName, 0 );
```

MATLAB example

Not implemented

.NET example

```
A valid camera name may be obtained from GetCameraName( CameraIndex )
A valid blob index is between 0 and GetGreyscaleBlobCount() - 1
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableGreyscaleData ();
MyClient.GetFrame();
Output_GetCameraName CameraName = MyClient.GetCameraName( 0 );
Output_GetGreyscaleBlob GreyscaleData = MyClient.GetGreyscaleBlob( CameraName.CameraName, 0 );
```

## Returns

An `Output_GetGreyscaleBlob` class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected
  - InvalidCameraName
  - InvalidIndex

## Output\_GetVideoFrame GetVideoFrame ( const std::string & *CameraName* ) const

Obtains video data for the specified camera.

See Also: -

C example

Not implemented

C++ example

```
A valid camera name may be obtained from GetCameraName( CameraIndex )
A valid blob index is between 0 and GetGreyscaleBlobCount() - 1
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableVideoData ();
MyClient.GetFrame();
Output_GetCameraName CameraName = MyClient.GetCameraName( 0 );
Output_GetVideoFrame VideoData = MyClient.GetVideoFrame( CameraName.CameraName );
```

MATLAB example

Not implemented

.NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableVideoData ();
MyClient.GetFrame();
Output_GetCameraName CameraName = MyClient.GetCameraName( 0 );
Output_GetVideoFrame VideoData = MyClient.GetVideoFrame( CameraName.CameraName ); /// -----
```

### Returns

An Output\_GetVideoFrame class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected
  - InvalidCameraName

## Class Documentation

**Output\_SetCameraFilter** SetCameraFilter ( const std::vector< unsigned int > & *CameraIdsForCentroids*, const std::vector< unsigned int > & *CameraIdsForBlobs*, const std::vector< unsigned int > & *CameraIdsForVideo* )

Add a filter to allow centroid, blob or video data to be transmitted for the specified cameras only.

See Also: [GetGreyscaleBlobCount\(\)](#), [GetGreyscaleBlob\(\)](#), [GetCentroidCount\(\)](#), [GetCentroidPosition\(\)](#), [GetCentroidWeight\(\)](#)

### C example

Not implemented

### C++ example

```
A valid camera name may be obtained from GetCameraName( CameraIndex )
A valid camera id may be obtained from GetCameraId( CameraName )
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetCameraName CameraName = MyClient.GetCameraName( 0 );
Output_GetCameraId CameraId = MyClient.GetCameraName( CameraName.CameraName );
std::vector< unsigned int > ReceiveCentroids;
ReceiveCentroids.push_back( CameraId.CameraId );
std::vector< unsigned int > ReceiveBlobs;
ReceiveBlobs.push_back( CameraId.CameraId );
std::vector< unsigned int > ReceiveVideo;
ReceiveVideo.push_back( CameraId.CameraId );
Output_SetCameraFilter FilterResults =
    MyClient.SetCameraFilter( ReceiveCentroids, ReceiveBlobs, ReceiveVideo );
```

### MATLAB example

Not implemented

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetCameraName CameraName = MyClient.GetCameraName( 0 );
Output_GetCameraId CameraId = MyClient.GetCameraName( CameraName.CameraName );
List< unsigned int >^ ReceiveCentroids = gcnew List< unsigned int >();
ReceiveCentroids.Add( CameraId.CameraId );
List< unsigned int >^ ReceiveBlobs = gcnew List< unsigned int >();
ReceiveBlobs.Add( CameraId.CameraId );
List< unsigned int >^ ReceiveVideo = gcnew List< unsigned int >();
ReceiveVideo.Add( CameraId.CameraId );
Output_SetCameraFilter FilterResults =
    MyClient.SetCameraFilter( ReceiveCentroids, ReceiveBlobs, ReceiveVideo );    /// -----
```

### Returns

An **Output\_SetCameraFilter** class containing the result of the operation.

- The Result will be:
  - Success



## Output\_ClearSubjectFilter ClearSubjectFilter ( )

Clear the subject filter.

This will result in all subjects being sent.

See Also: [AddToSubjectFilter\(\)](#)

### Returns

An Output\_ClearSubjectFilter class containing the result of the operation.

- The Result will be:
  - Success

## Output\_AddToSubjectFilter AddToSubjectFilter ( const String & *SubjectName* )

Add a subject name to the subject filter.

Only subjects present in the subject filter will be sent and subjects not in the filter will be presented as absent/occluded. If no filtered subjects are present, all subjects will be sent.

See Also : [ClearSubjectFilter\(\)](#)

### C example

```
// assuming there are two subjects in the stream, "Subject1" and "Subject2"
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableSegmentData( pClient );
Client_GetFrame( pClient );
Client_ClearSubjectFilter();
Client_AddToSubjectFilter( "Subject1" );
Client_GetFrame( pClient );
COutput_GetSegmentGlobalTranslation _Output_Subject1;
COutput_GetSegmentGlobalTranslation _Output_Subject2;
Client_GetSegmentGlobalTranslation(pClient, "Subject1", "root", &_Output_Subject1);
Client_GetSegmentGlobalTranslation(pClient, "Subject2", "root", &_Output_Subject2);
// _Output_Subject1.Occluded == true
// _Output_Subject2.Occluded == true
Client_Destroy( pClient );
```

### C++ example

```
// assuming there are two subjects in the stream, "Subject1" and "Subject2"
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableSegmentData();
MyClient.GetFrame();
MyClient.ClearSubjectFilter();
Output_GetAddToSubjectFilter Output = MyClient.AddToSubjectFilter( "Subject1" );
// New frames now only contain the filtered subject(s) if subject is in the stream.
MyClient.GetFrame();
Output_GetSegmentGlobalTranslation Output_Sub1 = MyClient.GetSegmentGlobalTranslation("Subject1","root");
Output_GetSegmentGlobalTranslation Output_Sub2 = MyClient.GetSegmentGlobalTranslation("Subject2","root");
// Output_Sub1.Occluded == false
// Output_Sub2.Occluded == true
```

### MATLAB example

```
// assuming there are two subjects in the stream, "Subject1" and "Subject2"
MyClient = ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
MyClient.EnableSegmentData();
MyClient.ClearSubjectFilter();
MyClient.AddToSubjectFilter("Subject1");
MyClient.GetFrame();
Output_Subject1 = MyClient.GetSegmentGlobalTranslation( "Subject1", "root" );
Output_Subject2 = MyClient.GetSegmentGlobalTranslation( "Subject2", "root" );
// Output_Subject1.Occluded == false
// Output_Subject2.Occluded == true
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient =
new ViconDataStreamSDK.DotNET.Client();
```

## Class Documentation

```
MyClient.Connect( "localhost" );
MyClient.EnableSegmentData();
Client.GetFrame();
Client.ClearSubjectFilter();
Client.AddToSubjectFilter("Subject1")
MyClient.GetFrame();
Output_GetSegmentGlobalTranslation Output_Subject1 =
MyClient.GetSegmentGlobalTranslations( "Subject1", "root" );
Output_GetSegmentGlobalTranslation Output_Subject2 =
MyClient.GetSegmentGlobalTranslations( "Subject2", "root" );
// Output_Subject1.Occluded = false;
// Output_Subject2.Occluded = true;
```

### Parameters

<i>SubjectName</i>	The name of the subject.
--------------------	--------------------------

### Returns

An Output\_AddToSubjectFilter class containing the result of the operation.

- The Result will be:
  - Success
  - InvalidSubjectName

## Output\_ConfigureWireless ConfigureWireless ( ) [virtual]

Request that the wireless adapters will be optimally configured for streaming data.

On Windows this will disable background scan and enable streaming. The call does not need the client to be connected.

C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.ConfigureWireless();
```

### Returns

An Output\_ConfigureWireless class containing the result of the operation.

- The Result will be:
  - Success if the adapters are configured or there are no adapters to configure
  - NotSupported if the OS does not support this function
  - WirelessConfigurationFailed if the request failed
- The Error will provide additional information in the failure case

The documentation for this class was generated from the following files:

- DataStreamClient.h
- DataStreamClient.cpp

## RetimingClient Class Reference

### Detailed Description

The re-timing client class for C++.

### Vicon DataStream SDK Re-Timing Client

#### Intended uses

The Vicon DataStream re-timing client provides calls to obtain subject data from the DataStream with minimal latency and temporal jitter. When [UpdateFrame\(\)](#) is called, the client uses re-timed data that has been linearly interpolated from an internal buffer to predict the position of each segment to the current time.

The system and network latencies are used when determining the amount of prediction required. If additional prediction is required, for example, for use in a VR system where an additional latency is present due to rendering and display latency; this may be requested in the call to [UpdateFrame\(\)](#).

The user will call [UpdateFrame\(\)](#), which will update the current frame state to the time of calling and return immediately. This is intended for use in systems where you require subject data positions at times driven by an external clock.

If you do not have an external clock, and require behavior similar to that of the standard `DataStream` client running in `ServerPush` streaming mode, then the system may be configured to provide frame data at a consistent frame rate by providing a frame rate to the `Connect()` call. The user will then call `WaitForFrame()`, which will block in a similar method to `Client::GetFrame()`, but using retimed data in order to keep the frame period very consistent.

## Examples of use

If you are using the client in a situation where you need to obtain the position of subjects

```
ViconDataStreamSDK::CPP::RetimingClient _MyClient;
_MyClient.Connect( "localhost" );

// example render method. Real code would probably cache the subject and segment names and bind
// them to a model, so this code would iterate over the model and update the joint positions.
void render()
{
    _MyClient.UpdateFrame();

    Output_GetSubjectCount SubjectCount = _MyClient.GetSubjectCount();
    if (SubjectCount.Result == Result::Success)
    {
        for (unsigned int SubjectIndex = 0; SubjectIndex < SubjectCount.SubjectCount; ++SubjectIndex)
        {
            Output_GetSubjectName SubjectName = _MyClient.GetSubjectName(SubjectIndex);
            if (SubjectName.Result == Result::Success)
            {
                Output_GetSegmentCount SegmentCount = _MyClient.GetSegmentCount(SubjectName.SubjectName);
                if (SegmentCount.Result == Result::Success)
                {
                    for (unsigned int SegmentIndex = 0; SegmentIndex < SegmentCount.SegmentCount; ++SegmentIndex)
                    {
                        Output_GetSegmentName SegmentName =
                            _MyClient.GetSegmentName(SubjectName.SubjectName, SegmentIndex);
                        if (SegmentName.Result == Result::Success)
                        {
                            Output_GetSegmentGlobalRotationQuaternion SegmentRotation =
                                _MyClient.GetSegmentGlobalRotationQuaternion
                                    (SubjectName.SubjectName, SegmentName.SegmentName);
                            if (SegmentRotation.Result == Result::Success && !SegmentRotation.Occluded)
                            {
                                // use the segment rotation
                            }
                        }
                    }
                }
            }
        }
    }
}
```

If using the client where there is no render call and you require your own timing.

```
ViconDataStreamSDK::CPP::RetimingClient _MyClient;
// Request a retimed update frame rate of 90Hz.
_MyClient.Connect( "localhost", 90 );
while( _MyClient.IsConnected() )
{
    Output_WaitForFrame WaitOutput = _MyClient.WaitForFrame();
    if( WaitOutput.Result == Result::Success )
    {
    }
```

```
// iterate over subjects and segments and obtain the joint positions and rotations as above.
}
}
```

For a more detailed example, see the `ViconDataStreamSDK_CPPRetimingTest` example. The Simple-Viewer application also provides an example of re-timing client use in a practical context.

## Public Member Functions

- [RetimingClient](#) ()  
*Construction.*
- virtual [~RetimingClient](#) ()  
*Destruction.*
- Output\_GetVersion [GetVersion](#) () const  
*Get the version of the Vicon DataStream SDK.*
- Output\_Connect [Connect](#) (const String &HostName, double FrameRate=0.0)  
*Establish a dedicated connection to a Vicon DataStream Server.*
- Output\_Disconnect [Disconnect](#) ()  
*Disconnect from the Vicon DataStream Server.*
- Output\_IsConnected [IsConnected](#) () const  
*Discover whether client is connected to the Vicon DataStream Server.*
- Output\_EnableLightweightSegmentData [EnableLightweightSegmentData](#) ()  
*Enable a lightweight transmission protocol for kinematic segment data in the Vicon DataStream.*
- Output\_DisableLightweightSegmentData [DisableLightweightSegmentData](#) ()  
*Disable the lightweight output mode for kinematic segment data in the Vicon DataStream.*
- Output\_IsLightweightSegmentDataEnabled [IsLightweightSegmentDataEnabled](#) () const  
*Return whether the lightweight transport mode for kinematic segment data is enabled in the Vicon DataStream.*
- Output\_SetAxisMapping [SetAxisMapping](#) (const **Direction::Enum** XAxis, const **Direction::Enum** YAxis, const **Direction::Enum** ZAxis)  
*Remaps the 3D axis.*
- Output\_GetAxisMapping [GetAxisMapping](#) () const  
*Get the current Axis mapping.*
- Output\_UpdateFrame [UpdateFrame](#) (double Offset=0.0)  
*Update the current frame state to represent the position of all active subjects at the current time.*
- Output\_WaitForFrame [WaitForFrame](#) ()  
*Used when running the retiming client with a specified frame rate.*
- Output\_GetSubjectCount [GetSubjectCount](#) () const  
*Return the number of subjects in the DataStream.*
- Output\_GetSubjectName [GetSubjectName](#) (const unsigned int SubjectIndex) const  
*Return the name of a subject.*
- Output\_GetSubjectRootSegmentName [GetSubjectRootSegmentName](#) (const String &SubjectName) const  
*Return the name of the root segment for a specified subject.*
- Output\_GetSegmentCount [GetSegmentCount](#) (const String &SubjectName) const

*Return the number of segments for a specified subject in the DataStream.*

- Output\_GetSegmentName [GetSegmentName](#) (const String &SubjectName, const unsigned int SegmentIndex) const

*Return the name of a subject segment specified by index.*

- Output\_GetSegmentChildCount [GetSegmentChildCount](#) (const String &SubjectName, const String &SegmentName) const

*Return the number of child segments for a specified subject segment.*

- Output\_GetSegmentChildName [GetSegmentChildName](#) (const String &SubjectName, const String &SegmentName, const unsigned int SegmentIndex) const

*Return the name of the child segment for a specified subject segment and index.*

- Output\_GetSegmentParentName [GetSegmentParentName](#) (const String &SubjectName, const String &SegmentName) const

*Return the name of the parent segment for a specified subject segment.*

- Output\_GetSegmentStaticTranslation [GetSegmentStaticTranslation](#) (const String &SubjectName, const String &SegmentName) const

*Return the static pose translation of a subject segment.*

- Output\_GetSegmentStaticRotationHelical [GetSegmentStaticRotationHelical](#) (const String &SubjectName, const String &SegmentName) const

*Return the static pose rotation of a subject segment in helical coordinates.*

- Output\_GetSegmentStaticRotationMatrix [GetSegmentStaticRotationMatrix](#) (const String &SubjectName, const String &SegmentName) const

*Return the static pose rotation of a subject segment as a 3x3 row-major matrix.*

- Output\_GetSegmentStaticRotationQuaternion [GetSegmentStaticRotationQuaternion](#) (const String &SubjectName, const String &SegmentName) const

*Return the static pose rotation of a subject segment in quaternion coordinates.*

- Output\_GetSegmentStaticRotationEulerXYZ [GetSegmentStaticRotationEulerXYZ](#) (const String &SubjectName, const String &SegmentName) const

*Return the static pose rotation of a subject segment in Euler XYZ coordinates.*

- Output\_GetSegmentGlobalTranslation [GetSegmentGlobalTranslation](#) (const String &SubjectName, const String &SegmentName) const

*Return the translation of a subject segment in global coordinates.*

- Output\_GetSegmentGlobalRotationHelical [GetSegmentGlobalRotationHelical](#) (const String &SubjectName, const String &SegmentName) const

*Return the rotation of a subject segment in global helical coordinates.*

- Output\_GetSegmentGlobalRotationMatrix [GetSegmentGlobalRotationMatrix](#) (const String &SubjectName, const String &SegmentName) const

*Return the rotation of a subject segment as a 3x3 row-major matrix in global coordinates.*

- Output\_GetSegmentGlobalRotationQuaternion [GetSegmentGlobalRotationQuaternion](#) (const String &SubjectName, const String &SegmentName) const

*Return the rotation of a subject segment in global quaternion coordinates.*

- Output\_GetSegmentGlobalRotationEulerXYZ [GetSegmentGlobalRotationEulerXYZ](#) (const String &SubjectName, const String &SegmentName) const

*Return the rotation of a subject segment in global Euler XYZ coordinates.*

- Output\_GetSegmentLocalTranslation [GetSegmentLocalTranslation](#) (const String &SubjectName, const String &SegmentName) const

*Return the translation of a subject segment in local coordinates relative to its parent segment.*

- Output\_GetSegmentLocalRotationHelical [GetSegmentLocalRotationHelical](#) (const String &SubjectName, const String &SegmentName) const  
*Return the rotation of a subject segment in local helical coordinates relative to its parent segment.*
- Output\_GetSegmentLocalRotationMatrix [GetSegmentLocalRotationMatrix](#) (const String &SubjectName, const String &SegmentName) const  
*Return the rotation row-major matrix of a subject segment in local coordinates relative to its parent segment.*
- Output\_GetSegmentLocalRotationQuaternion [GetSegmentLocalRotationQuaternion](#) (const String &SubjectName, const String &SegmentName) const  
*Return the rotation of a subject segment in local quaternion coordinates relative to its parent segment.*
- Output\_GetSegmentLocalRotationEulerXYZ [GetSegmentLocalRotationEulerXYZ](#) (const String &SubjectName, const String &SegmentName) const  
*Return the rotation of a subject segment in local Euler XYZ coordinates relative to its parent segment.*
- void [SetMaximumPrediction](#) (double MaxPrediction)  
*Sets the maximum amount by which the interpolation engine will predict later than the latest received frame.*
- double [MaximumPrediction](#) () const  
*Returns the maximum prediction value currently in use.*

## Constructor & Destructor Documentation

### [RetimingClient](#) ( )

Construction.

Instances of the Vicon Data Stream [RetimingClient](#) create a [DataStreamClient](#) internally that manages the connection to the data stream.

The [RetimingClient](#) will set up the underlying client to receive the required data from the stream and to set the correct data delivery mode, so it is not necessary to set this up manually.

C example

```
// The C version uses explicit creation methods

CClient * pClient = RetimingClient_Create();
// C Client functions take the client as a parameter
CBool ok = RetimingClient_SomeFunction( pClient, Args );
// The C client needs to be explicitly destroyed
RetimingClient_Destroy( pClient );
```

C++ example

```
// The C++ version of the SDK is object oriented, so use the class constructor.

{
    ViconDataStreamSDK::CPP::RetimingClient StackRetimingClient;
    Output_SomeFunction Output = StackRetimingClient.SomeFunction();
    // ...
}
// Client is implicitly destroyed as it goes out of scope.

// Alternatively the Client can be made on the heap.
```



```
ViconDataStreamSDK::CPP::RetimingClient * pHeapRetimingClient
= new ViconDataStreamSDK::CPP::RetimingClient();
Output_SomeFunction Output = pHeapRetimingClient->SomeFunction( Input );
```

## MATLAB example

See .NET example

## .NET example

```
// .NET is object oriented, so use the class constructor.

// Because objects are lazily garbage collected, your instance may outlive the
// last reference to it for some time.

// If the instance is pre-fetching frame data for you, then it can still use CPU
// and network bandwidth.

// Consider explicitly disconnecting prior to destruction.

ViconDataStreamSDK.DotNET.RetimingClient pHeapClient
= new ViconDataStreamSDK.DotNET.RetimingClient();
Output_SomeFunction Output = pHeapClient.SomeFunction(InputParam);
// Signal to the garbage collector that it can clean up
pHeapClient.Disconnect();
pHeapClient = null;
```

## ~RetimingClient ( ) [virtual]

Destruction.

Destruction will Disconnect if required.

See [RetimingClient::RetimingClient](#) for an example.

## Member Function Documentation

### Output\_GetVersion GetVersion ( ) const

Get the version of the Vicon DataStream SDK.

- **Major** When this number increases, we break backward compatibility with previous major versions.
- **Minor** When this number increases, we have probably added new functionality to the SDK without breaking backward compatibility with previous versions.
- **Point** When this number increases, we have introduced a bug fix or performance enhancement without breaking backward compatibility with previous versions.

The function can be called without the client being connected.

C example

## Class Documentation

```
CRetimingClient * pClient = RetimingClient_Create();  
COutput_GetVersion Output = RetimingClient_GetVersion( pClient );  
RetimingClient_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::RetimingClient MyClient;  
Output_GetVersion Output = MyClient.GetVersion();
```

### MATLAB example

See .NET example

### .NET example

```
ViconDataStreamSDK.DotNET.RetimingClient MyClient = new ViconDataStreamSDK.DotNET.RetimingClient();  
Output_GetVersion Output = MyClient.GetVersion();
```

### Returns

Output\_GetVersion class containing the version information.

## Output\_Connect Connect ( const String & *HostName*, double *FrameRate* = 0.0 )

Establish a dedicated connection to a Vicon DataStream Server.

See Also: [Disconnect\(\)](#), [IsConnected\(\)](#).

The function defaults to connecting on port 801. You can specify an alternate port number after a colon. This is for future compatibility: current products serve data on port 801 only.

### C example

```
CRetimingClient * pRetimingClient = RetimingClient_Create();
COutput_Connect Output = RetimingClient_Connect( pRetimingClient, "localhost");
RetimingClient_Destroy( pRetimingClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::RetimingClient MyClient;
Output_Connect Output = MyClient.Connect( "localhost" );
```

### MATLAB example

See [.NET example](#)

### .NET example

```
ViconDataStreamSDK.DotNET.RetimingClient MyClient = new ViconDataStreamSDK.DotNET.RetimingClient();
Output_Connect Output = MyClient.Connect( "localhost" );
// Connect with alternative FrameRate parameter
Output_Connect Output = MyClient.Connect( "localhost", 90.0 );
```

### Parameters

<i>HostName</i>	The DNS-identifiable name, or IP address of the PC hosting the DataStream server. For example: <ul style="list-style-type: none"> <li>• "localhost"</li> <li>• "MyViconPC:801"</li> <li>• "10.0.0.2"</li> </ul>
<i>FrameRate</i>	An optional parameter - if specified, the re-timing client's internal frame output clock will be active. This is implemented by a separate overloaded method on .NET

### Returns

An Output\_Connect class containing the result of the connect operation.

- The Result will be:
  - Success
  - InvalidHostName
  - ClientAlreadyConnected
  - ClientConnectionFailed

## Output\_Disconnect Disconnect ( )

Disconnect from the Vicon DataStream Server.

See Also: [Connect\(\)](#), [IsConnected\(\)](#)

### C example

```
CRetimingClient * pRetimingClient = RetimingClient_Create();
RetimingClient_Connect( pRetimingClient, "localhost" );
COutput_Disconnect Output = RetimingClient_Disconnect( pRetimingClient );
RetimingClient_Destroy( pRetimingClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::RetimingClient MyClient;
MyClient.Connect( "localhost" );
Output_Disconnect Output = MyClient.Disconnect();
```

### MATLAB example

See [.NET example](#)

### .NET example

```
ViconDataStreamSDK.DotNET.RetimingClient MyClient = new ViconDataStreamSDK.DotNET.RetimingClient();
MyClient.Connect( "localhost" );
Output_Disconnect Output = MyClient.Disconnect();
```

### Returns

An `Output_Disconnect` class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected

## Output\_IsConnected IsConnected ( ) const

Discover whether client is connected to the Vicon DataStream Server.

See Also: [Connect\(\)](#), [Disconnect\(\)](#)

### C example

```
CRetimingClient * pRetimingClient = RetimingClient_Create();
CBool Output = RetimingClient_IsConnected( pRetimingClient );
// Output == 0
RetimingClient_Connect( pRetimingClient, "localhost" );
Output = RetimingClient_IsConnected( pRetimingClient );
// Output == 1
COutput_Disconnect Output = RetimingClient_Disconnect( pRetimingClient );
RetimingClient_Destroy( pRetimingClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
Output_IsConnected Output = MyClient.IsConnected()
// Output.Connected == false
MyClient.Connect( "localhost" );
Output_IsConnected Output = MyClient.IsConnected()
// Output.Connected == true
// (assuming localhost is serving)
```

### MATLAB example

See [.NET example](#)

### .NET example

```
ViconDataStreamSDK.DotNET.RetimingClient MyClient = new ViconDataStreamSDK.DotNET.RetimingClient();
Output_IsConnected Output = MyClient.IsConnected()
// Output.Connected == false
MyClient.Connect( "localhost" );
Output_IsConnected Output = MyClient.IsConnected()
// Output.Connected == true
// (assuming localhost is serving)
```

### Returns

An `Output_IsConnected` class containing a true value for `Connected` if you are connected to the stream, otherwise false.

## Output\_EnableLightweightSegmentData EnableLightweightSegmentData ( )

Enable a lightweight transmission protocol for kinematic segment data in the Vicon DataStream.

This will reduce the network bandwidth required to transmit segment data to approximately a quarter of that required by the previous method, at the expense of a small amount of precision. Use the existing methods such as [GetSegmentGlobalTranslation\(\)](#) and [GetSegmentGlobalRotationMatrix\(\)](#) as usual to obtain the segment data. Calling this method will automatically disable all other configurable output types. These may be re-enabled after the call if required.

Call this function on startup, after connecting to the server, and before trying to read local or global segment data.

See Also: [DisableLightweightSegmentData\(\)](#), [IsLightWeightSegmentDataEnabled\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_EnableLightweightSegmentData();
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Output_EnableLightweightSegmentData Output = MyClient.EnableLightweightSegmentData();
```

### MATLAB example

```
MyClient = Client();
MyClient.Connect( "localhost" );
Output_EnableLightweightSegmentData Output = MyClient.EnableLightweightSegmentData();
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output_EnableLightweightSegmentData Output = MyClient.EnableLightweightSegmentData();
```

### Returns

An [Output\\_EnableSegmentData](#) class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected

## Output\_DisableLightweightSegmentData DisableLightweightSegmentData ( )

Disable the lightweight output mode for kinematic segment data in the Vicon DataStream.

The implementation in this retiming client automatically enables normal segment data; this is distinct to the non retiming client where the user must do this themselves.

See Also: [EnableLightweightSegmentData\(\)](#), [IsLightWeightSegmentDataEnabled\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
Client_DisableLightweightSegmentData();
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Output_DisableLightweightSegmentData Output = MyClient.DisableLightweightSegmentData();
```

### MATLAB example

```
MyClient = Client();
MyClient.Connect( "localhost" );
Output = MyClient.DisableLightweightSegmentData ();
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output_DisableLightweightSegmentData Output = MyClient.DisableLightweightSegmentData ();
```

### Returns

An `Output_DisableLightweightSegmentData` class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected

## Output\_IsLightweightSegmentDataEnabled IsLightweightSegmentDataEnabled ( ) const

Return whether the lightweight transport mode for kinematic segment data is enabled in the Vicon Data-Stream.

See Also: [EnableLightweightSegmentData\(\)](#), [DisableLightWeightSegmentDataEnabled\(\)](#)

### C example

```
CClient * pClient = Client_Create();
Client_Connect( pClient, "localhost" );
CBool Output = Client_IsLightweightSegmentDataEnabled( pClient )
// Output == 0
Client_EnabledSegmentData( pClient );
CBool Output = Client_IsLightweightSegmentDataEnabled( pClient )
// Output == 1
Client_Destroy( pClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
Output_IsLightweightSegmentDataEnabled Output = MyClient.IsLightweightSegmentDataEnabled();
// Output.Enabled == false
MyClient.EnableSegmentData();
Output_IsLightweightSegmentDataEnabled Output = MyClient.IsLightweightSegmentDataEnabled();
// Output.Enabled == true
```

### MATLAB example

```
MyClient = Client();
MyClient.Connect( "localhost" );
Output = MyClient.IsLightweightSegmentDataEnabled(); % Output.Enabled == false
MyClient.EnableSegmentData();
Output = MyClient.IsLightweightSegmentDataEnabled(); % Output.Enabled == true
```

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
Output_IsLightweightSegmentDataEnabled Output = MyClient.IsLightweightSegmentDataEnabled();
// Output.Enabled == false
MyClient.EnableSegmentData();
Output_IsLightweightSegmentDataEnabled Output = MyClient.IsLightweightSegmentDataEnabled();
// Output.Enabled == true
```

### Returns

An `Output_IsLightweightSegmentDataEnabled` class containing the result of the operation.

- The Result will be:
  - Whether the data is enabled



## Output\_SetAxisMapping SetAxisMapping ( const **Direction::Enum** XAxis, const **Direction::Enum** YAxis, const **Direction::Enum** ZAxis )

Remaps the 3D axis.

Vicon Data uses a right-handed coordinate system, with +X forward, +Y left, and +Z up. Other systems use different coordinate systems. The SDK can transform its data into any valid right-handed coordinate system by re-mapping each axis. Valid directions are "Up", "Down", "Left", "Right", "Forward", and "-Backward". Note that "Forward" means moving away from you, and "Backward" is moving towards you. Common usages are Z-up: SetAxisMapping( Forward, Left, Up ) Y-up: SetAxisMapping( Forward, Up, Right )

See Also: [GetAxisMapping\(\)](#)

### C example

```
CRetimingClient * pRetimingClient = RetimingClient_Create();
RetimingClient_Connect( pRetimingClient, "localhost" );
RetimingClient_SetAxisMapping(pRetimingClient, Forward, Left, Up); // Z-up
RetimingClient_Destroy( pRetimingClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::RetimingClient MyClient;
MyClient.SetAxisMapping( ViconDataStreamSDK::CPP::Direction::Forward,
ViconDataStreamSDK::CPP::Direction::Left,
ViconDataStreamSDK::CPP::Direction::Up );
```

### MATLAB example

See [.NET example](#)

### .NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
MyClient.SetAxisMapping( ViconDataStreamSDK.DotNET.Direction.Forward,
ViconDataStreamSDK.DotNET.Direction.Left,
ViconDataStreamSDK.DotNET.Direction.Up );
```

### Parameters

XAxis	Specify the direction of your X axis relative to yourself as the observer.
YAxis	Specify the direction of your Y axis relative to yourself as the observer.
ZAxis	Specify the direction of your Z axis relative to yourself as the observer.

### Returns

An Output\_SetAxisMapping class containing the result of the operation.

- The Result will be:
  - Success
  - CoLinearAxes
  - LeftHandedAxes

## Output\_GetAxisMapping GetAxisMapping ( ) const

Get the current Axis mapping.

See Also: [SetAxisMapping\(\)](#)

C example

```
CRetimingClient * pRetimingClient = RetimingClient_Create();
RetimingClient_Connect( pRetimingClient, "localhost" );
RetimingClient_SetAxisMapping(pRetimingClient, Forward, Left, Up); // Z-up
RetimingClient_Destroy( pRetimingClient );
```

C++ example

```
ViconDataStreamSDK::CPP::RetimingClient MyClient;
Output_GetAxisMapping Output = MyClient.GetAxisMapping();
// Output.XAxis == ViconDataStreamSDK::CPP::Direction::Forward
// Output.YAxis == ViconDataStreamSDK::CPP::Direction::Left
// Output.ZAxis == ViconDataStreamSDK::CPP::Direction::Up
```

MATLAB example

See .NET example

.NET example

```
ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();
Output_GetAxisMapping Output = MyClient.GetAxisMapping();
// Output.XAxis == ViconDataStreamSDK.DotNET.Direction.Forward
// Output.YAxis == ViconDataStreamSDK.DotNET.Direction.Left
// Output.ZAxis == ViconDataStreamSDK.DotNET.Direction.Up
```

### Returns

An Output\_GetAxisMapping class containing the result of the operation.

- The Result will be:
  - XAxis, YAxis, ZAxis

## Output\_UpdateFrame UpdateFrame ( double *Offset* = 0.0 )

Update the current frame state to represent the position of all active subjects at the current time.

The position of each segment is estimated by predicting forwards from the most recent frames received from the DataStream, taking into account the latency reported by the system to determine the amount of prediction required.

The results of calls which return details about the current frame state such as [GetSubjectCount\(\)](#) and [GetSegmentGlobalRotationQuaternion\(\)](#) will all return the stream contents and position at the time that this call was made.

If no call to [UpdateFrame\(\)](#) is made, calls querying the stream state will return NoFrame.

### C example

```
CRetimingClient * pRetimingClient = RetimingClient_Create();
CEnum Output = RetimingClient_GetFrame(); // Output == NotConnected
RetimingClient_Connect( pRetimingClient, "localhost" );
Output = RetimingClient_UpdateFrame(); // Output == Success
RetimingClient_Destroy( pRetimingClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::RetimingClient MyClient;
Output_UpdateFrame Output;
Output = MyClient.UpdateFrame(); // Output.Result == NotConnected
MyClient.Connect( "localhost" );
Output = MyClient.UpdateFrame(); // Output.Result == Success
```

### MATLAB example

See .NET example

### .NET example

```
ViconDataStreamSDK.DotNET.RetimingClient MyClient = new ViconDataStreamSDK.DotNET.RetimingClient();
Output_UpdateFrame Output;
Output = MyClient.UpdateFrame(); // Output.Result == NotConnected
MyClient.Connect( "localhost" );
Output = MyClient.UpdateFrame(); // Output.Result == Success
Output = MyClient.UpdateFrame(20); // Output.Result == Success
```

### Parameters

<i>Offset</i>	An additional offset that will be applied to the time at which the predicted position is calculated. This may be used to compensate for additional delays that are in the user's system, such as render delay. This is implemented in a separate overloaded method in .NET.
---------------	---

### Returns

An Output\_UpdateFrame class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected

## Output\_WaitForFrame WaitForFrame ( )

Used when running the retiming client with a specified frame rate.

This call will block until the next frame is available, as driven by an internal clock running at the frame rate specified by Connect( Host, FrameRate). The frame data is re-timed to the correct time point.

### C example

```
CRetimingClient * pRetimingClient = RetimingClient_Create();
CEnum Output = RetimingClient_GetFrame(); // Output == NotConnected
RetimingClient_ConnectAndStart( pRetimingClient, "localhost", 200 );
Output = RetimingClient_WaitForFrame(); // Output == Success
RetimingClient_Destroy( pRetimingClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::RetimingClient MyClient;
MyClient.Connect( "localhost", 200 );
Output = MyClient.WaitForFrame(); // Output.Result == Success
```

### MATLAB example

See .NET example

### .NET example

```
ViconDataStreamSDK.DotNET.RetimingClient MyClient = new ViconDataStreamSDK.DotNET.RetimingClient();
MyClient.Connect( "localhost", 200 );
Output = MyClient.WaitForFrame(); // Output.Result == Success
```

### Returns

An Output\_WaitForFrame class containing the result of the operation.

- The Result will be:
  - Success
  - NotConnected

## Output\_GetSubjectCount GetSubjectCount ( ) const

Return the number of subjects in the DataStream.

This information can be used in conjunction with GetSubjectName.

See Also: [GetSubjectName\(\)](#)

### C example

```
CRetimingClient * pRetimingClient = RetimingClient_Create();
RetimingClient_Connect( pRetimingClient, "localhost" );
COutput_GetSubjectCount SubjectCount;
RetimingClient_GetSubjectCount( pRetimingClient, &SubjectCount); // SubjectCount.Result == NoFrame
// SubjectCount.SubjectCount == 0;
RetimingClient_GetFrame( pRetimingClient );
RetimingClient_GetSubjectCount( pRetimingClient, &SubjectCount); // SubjectCount.Result == Success;
// SubjectCount.SubjectCount == 0;
RetimingClient_Destroy( pRetimingClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::RetimingClient MyClient;
MyClient.Connect( "localhost" );
Output_GetSubjectCount Output;
Output = MyClient.GetSubjectCount(); // Output.Result == NoFrame
// Ooutput.SubjectCount == 0
MyClient.GetFrame();
Output = MyClient.GetSubjectCount(); // Output.Result == Success
// Output.SubjectCount >= 0
```

### MATLAB example

See [.NET example](#)

### .NET example

```
ViconDataStreamSDK.DotNET.RetimingClient MyClient = new ViconDataStreamSDK.DotNET.RetimingClient();
MyClient.Connect( "localhost" );
Output_GetSubjectCount Output;
Output = MyClient.GetSubjectCount(); // Output.Result == NoFrame
// Ooutput.SubjectCount == 0
MyClient.UpdateFrame();
Output = MyClient.GetSubjectCount(); // Output.Result == Success
// Output.SubjectCount >= 0
```

### Returns

An Output\_GetSubjectCount class containing the result of the operation and the number of subjects.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame

## Output\_GetSubjectName GetSubjectName ( const unsigned int *SubjectIndex* ) const

Return the name of a subject.

This can be passed into segment and marker functions.

See Also: [GetSubjectCount\(\)](#)

### C example

```
CRetimingClient * pRetimingClient = RetimingClient_Create();
RetimingClient_Connect( pRetimingClient, "localhost" );
RetimingClient_GetFrame( pRetimingClient );
char SubjectName[128];
CEnum Output = RetimingClient_GetSubjectName(pRetimingClient, 0, 128, SubjectName);
// Output == Success
// SubjectName == "AI"
Output = RetimingClient_GetSubjectName(pRetimingClient, 1, 128, SubjectName);
// Output == Success
// SubjectName == "Bob"
Output = RetimingClient_GetSubjectName(pRetimingClient, 2, 128, SubjectName);
// Output == InvalidIndex
// SubjectName == ""

RetimingClient_Destroy( pRetimingClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::RetimingClient MyClient;
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetSubjectCount OutputGSC;
OutputGSC = MyClient.GetSubjectCount(); // OutputGSC.Result == Success
// OutputGSC.SubjectCount == 2
Output_GetSubjectName OutputGSN;
OutputGSN = MyClient.GetSubjectName(0); // OutputGSN.Result == Success
// OutputGSN.SubjectName == "AI"
OutputGSN = MyClient.GetSubjectName(1); // OutputGSN.Result == Success
// OutputGSN.SubjectName == "Bob"
OutputGSN = MyClient.GetSubjectName(2); // OutputGSN.Result == InvalidIndex
// OutputGSN.SubjectName == ""
```

### MATLAB example

See .NET example

### .NET example

```
ViconDataStreamSDK.DotNET.RetimingClient MyClient = new ViconDataStreamSDK.DotNET.RetimingClient();
MyClient.Connect( "localhost" );
MyClient.UpdateFrame();
Output_GetSubjectCount OutputGSC;
OutputGSC = MyClient.GetSubjectCount(); // OutputGSC.Result == Success
// OutputGSC.SubjectCount == 2
Output_GetSubjectName OutputGSN;
OutputGSN = MyClient.GetSubjectName(0); // OutputGSN.Result == Success
// OutputGSN.SubjectName == "AI"
OutputGSN = MyClient.GetSubjectName(1); // OutputGSN.Result == Success
// OutputGSN.SubjectName == "Bob"
OutputGSN = MyClient.GetSubjectName(2); // OutputGSN.Result == InvalidIndex
// OutputGSN.SubjectName == ""
```

## Parameters

<i>SubjectIndex</i>	The index of the subject. A valid Subject Index is between 0 and <a href="#">GetSubjectCount()</a> -1. Matlab: A valid Subject Index is between 1 and <a href="#">GetSubjectCount()</a> .
---------------------	---

## Returns

An `Output_GetSubjectName` `GetSubjectName` class containing the result of the operation and the name of the subject.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidIndex

## Output\_GetSubjectRootSegmentName GetSubjectRootSegmentName ( const String & *SubjectName* ) const

Return the name of the root segment for a specified subject.

This can be passed into segment functions. The root segment is the ancestor of all other segments in the subject.

See Also: [GetSegmentCount\(\)](#), [GetSegmentParentName\(\)](#), [GetSegmentChildCount\(\)](#), [GetSegmentChildName\(\)](#)

### C example

```
CRetimingClient * pRetimingClient = RetimingClient_Create();
RetimingClient_Connect( pRetimingClient, "localhost" );
RetimingClient_EnableSegmentData( pRetimingClient );
RetimingClient_GetFrame( pRetimingClient );
char RootSegment[128];
CEnum Result = RetimingClient_GetSubjectRootSegmentName(pRetimingClient, "Bob", 128, RootSegment);
// Result == Success
// RootSegment == "Pelvis"
RetimingClient_Destroy( pRetimingClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::RetimingClient MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableSegmentData();
MyClient.GetFrame();
Output_GetSubjectRootSegmentName Output;
Output = MyClient.GetSubjectRootSegmentName( "Bob" );
// Output.Result == Success
// Output.SegmentName == "Pelvis"
```

### MATLAB example

See .NET example

### .NET example

```
ViconDataStreamSDK.DotNET.RetimingClient MyClient = new ViconDataStreamSDK.DotNET.RetimingClient();
MyClient.Connect( "localhost" );
MyClient.UpdateFrame();
Output_GetSubjectRootSegmentName Output;
Output = MyClient.GetSubjectRootSegmentName( "Bob" );
// Output.Result == Success
// Output.SegmentName == "Pelvis"
```

### Parameters

<i>SubjectName</i>	The name of the subject
--------------------	-------------------------

### Returns

An Output\_GetSubjectRootSegmentName class containing the result of the operation and the name of the root segment.

- The Result will be:



- Success
- NotConnected
- NoFrame
- InvalidIndex

## Output\_GetSegmentCount GetSegmentCount ( const String & *SubjectName* ) const

Return the number of segments for a specified subject in the DataStream.

This information can be used in conjunction with GetSegmentName.

See Also: [GetSubjectName\(\)](#), [GetSegmentName\(\)](#)

### C example

```
CRetimingClient * pRetimingClient = RetimingClient_Create();
RetimingClient_Connect( pRetimingClient, "localhost" );
COutput_GetSegmentCount SegmentCount;
RetimingClient_GetSegmentCount( pRetimingClient, "Bob", &SegmentCount );
// SegmentCount.Result == NOFrame
// SegmentCount.Value == 0
RetimingClient_GetFrame( pRetimingClient );
RetimingClient_GetSegmentCount( pRetimingClient, "Al", &SegmentCount );
// SegmentCount.Result == InvalidSubjectName
// SegmentCount.Value == 0
RetimingClient_GetSegmentCount( pRetimingClient, "Bob", &SegmentCount );
// SegmentCount.Result == Success
// SegmentCount.Value >= 0
RetimingClient_Destroy( pRetimingClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::RetimingClient MyClient;
MyClient.EnableSegmentData();
MyClient.Connect( "localhost" );
Output_GetSegmentCount Output;
Output = MyClient.GetSegmentCount( "Bob" ); // Output.Result == NoFrame
// Output.SegmentCount == 0
MyClient.GetFrame();
Output = MyClient.GetSegmentCount( "Al" ); // Output.Result ==
// InvalidSubjectName
// Output.SegmentCount == 0
Output = MyClient.GetSegmentCount( "Bob" ); // Output.Result == Success
// Output.SegmentCount >= 0
```

### MATLAB example

See .NET example

### .NET example

```
ViconDataStreamSDK.DotNET.RetimingClient MyClient = new ViconDataStreamSDK.DotNET.RetimingClient();
MyClient.Connect( "localhost" );
Output_GetSegmentCount Output;
Output = MyClient.GetSegmentCount( "Bob" ); // Output.Result == NoFrame
// Output.SegmentCount == 0
MyClient.UpdateFrame();
Output = MyClient.GetSegmentCount( "Al" ); // Output.Result ==
// InvalidSubjectName
// Output.SegmentCount == 0
Output = MyClient.GetSegmentCount( "Bob" ); // Output.Result == Success
// Output.SegmentCount >= 0
```

### Parameters

<i>SubjectName</i>	The name of the subject.
--------------------	--------------------------

## Returns

An `Output_GetSegmentCount` class containing the result of the operation and the number of segments.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidIndex

## Output\_GetSegmentName GetSegmentName ( const String & *SubjectName*, const unsigned int *SegmentIndex* ) const

Return the name of a subject segment specified by index.

See Also: [GetSegmentCount\(\)](#), [GetSegmentChildCount\(\)](#), [GetSegmentChildName\(\)](#), [GetSubjectRootSegmentName\(\)](#)

### C example

```
CRetimingClient * pRetimingClient = RetimingClient_Create();
RetimingClient_Connect( pRetimingClient, "localhost" );
COutput_GetSegmentCount SegmentCount;
RetimingClient_GetSegmentCount( pRetimingClient, "Bob", &SegmentCount );
// SegmentCount.Result == NOFrame
// SegmentCount.Value == 0
RetimingClient_GetFrame( pRetimingClient );
RetimingClient_GetSegmentCount( pRetimingClient, "AI", &SegmentCount );
// SegmentCount.Result == InvalidSubjectName
// SegmentCount.Value == 0
RetimingClient_GetSegmentCount( pRetimingClient, "Bob", &SegmentCount );
// SegmentCount.Result == Success
// SegmentCount.Value >= 0
RetimingClient_Destroy( pRetimingClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::RetimingClient MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableSegmentData();
MyClient.GetFrame();
Output_GetSegmentName Output;
// SegmentIndex must be between 0 and GetSegmentCount() - 1
Output = MyClient.GetSegmentName( "Bob", 0 );
```

### MATLAB example

See .NET example

### .NET example

```
ViconDataStreamSDK.DotNET.RetimingClient MyClient = new ViconDataStreamSDK.DotNET.RetimingClient();
MyClient.Connect( "localhost" );
MyClient.UpdateFrame();
Output_GetSegmentName Output;
// SegmentIndex must be between 0 and GetSegmentCount() - 1
Output = MyClient.GetSegmentName( "Bob", 0 );
```

### Parameters

<i>SubjectName</i>	The name of the subject
<i>SegmentIndex</i>	The index of the segment

## Returns

An `Output_GetSegmentName` class containing the result of the operation and the name of the parent segment or an empty string if it is the root segment.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidIndex
  - InvalidSubjectName

## Output\_GetSegmentChildCount GetSegmentChildCount ( const String & SubjectName, const String & SegmentName ) const

Return the number of child segments for a specified subject segment.

This can be passed into segment functions.

See Also: [GetSegmentCount\(\)](#)

### C example

```
CRetimingClient * pRetimingClient = RetimingClient_Create();
RetimingClient_Connect( pRetimingClient, "localhost" );
RetimingClient_GetFrame( pRetimingClient );
COutput_GetSegmentChildCount ChildCount;
RetimingClient_GetSegmentChildCount( pRetimingClient, "Bob", "Pelvis", &ChildCount);
// ChildCount.Result == Success
// ChildCount.SegmentCount == 2
RetimingClient_GetSegmentChildCount( pRetimingClient, "Alice", "Pelvis", &ChildCount);
// ChildCount.Result == InvalidSubjectName
// ChildCount.SegmentCount == 0
char SegmentName[128];
RetimingClient_GetSegmentName( pRetimingClient, "Bob", , 128, SegmentName);
RetimingClient_GetSegmentName( pRetimingClient, "Bob", &SegmentName);
// ChildCount.Result == Success
// ChildCount.SegmentCount == 2
RetimingClient_Destroy( pRetimingClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::RetimingClient MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableSegmentData();
MyClient.GetFrame();
Output_GetSegmentChildCount OutputGSCC;
OutputGSCC = MyClient.GetSegmentChildCount( "Bob", "Pelvis" );
// OutputGSCC.Result == Success
// OutputGSCC.SegmentCount == 2
Output_GetSegmentChildName OutputGSCN;
OutputGSCN = MyClient.GetSegmentName( "Alice", 0 );
// OutputGSCN.Result == InvalidSubjectName
// OutputGSCN.SegmentName == ""
OutputGSCN = MyClient.GetSegmentName( "Bob", "Pelvis", 0 );
// OutputGSCN.Result == Success
// OutputGSCN.SegmentName == "LFemur"
OutputGSCN = MyClient.GetSegmentName( "Bob", "Pelvis", 1 );
// OutputGSCN.Result == Success
// OutputGSCN.SegmentName == "RFemur"
OutputGSCN = MyClient.GetSegmentName( "Bob", "Pelvis", 2 );
// OutputGSCN.Result == InvalidIndex
// OutputGSCN.SegmentName == ""
// (no third segment)
```

### MATLAB example

See .NET example

### .NET example

```
ViconDataStreamSDK.DotNET.RetimingClient MyClient = new ViconDataStreamSDK.DotNET.RetimingClient();
MyClient.Connect( "localhost" );
```

## Class Documentation

```
MyClient.UpdateFrame();
Output_GetSegmentChildCount OutputGSCC;
OutputGSCC = MyClient.GetSegmentChildCount( "Bob", "Pelvis" );
// OutputGSCC.Result == Success
// OutputGSCC.SegmentCount == 2
Output_GetSegmentChildName OutputGSCN;
OutputGSCN = MyClient.GetSegmentName( "Alice", 0 );
// OutputGSCN.Result == InvalidSubjectName
// OutputGSCN.SegmentName == ""
OutputGSCN = MyClient.GetSegmentName( "Bob", "Pelvis", 0 );
// OutputGSCN.Result == Success
// OutputGSCN.SegmentName == "LFemur"
OutputGSCN = MyClient.GetSegmentName( "Bob", "Pelvis", 1 );
// OutputGSCN.Result == Success
// OutputGSCN.SegmentName == "RFemur"
OutputGSCN = MyClient.GetSegmentName( "Bob", "Pelvis", 2 );
// OutputGSCN.Result == InvalidIndex
// OutputGSCN.SegmentName == ""
// (no third segment)
```

### Parameters

<i>SubjectName</i>	The name of the subject
<i>SegmentName</i>	The name of the segment

### Returns

An `Output_GetSegmentChildCount` class containing the result of the operation and the number of child segments.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidIndex
  - InvalidSubjectName
  - InvalidSegmentName

## Output\_GetSegmentChildName GetSegmentChildName ( const String & *SubjectName*, const String & *SegmentName*, const unsigned int *SegmentIndex* ) const

Return the name of the child segment for a specified subject segment and index.

See Also: [GetSegmentCount\(\)](#), [GetSegmentChildCount\(\)](#), [GetSegmentChildName\(\)](#), [GetSubjectRootSegmentName\(\)](#)

### C example

```
CRetimingClient * pRetimingClient = RetimingClient_Create();
RetimingClient_Connect( pRetimingClient, "localhost" );
RetimingClient_EnableSegmentData( pRetimingClient );
RetimingClient_GetFrame( pRetimingClient );
char SegmentChildName[128];
// Segment index must be between 0 and RetimingClient_GetSegmentChildCount() - 1
RetimingClient_GetSegmentChildName( pRetimingClient, "Bob", "Pelvis", 0, 128, SegmentChildName );
RetimingClient_Destroy( pRetimingClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::RetimingClient MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableSegmentData();
MyClient.GetFrame();
Output_GetSegmentChildName Output;
// Segment index must be between 0 and GetSegmentChildCount() - 1
Output = MyClient.GetSegmentChildName( "Bob", "Pelvis", 0 );
```

### MATLAB example

See .NET example

### .NET example

```
ViconDataStreamSDK.DotNET.RetimingClient MyClient = new ViconDataStreamSDK.DotNET.RetimingClient();
MyClient.Connect( "localhost" );
MyClient.UpdateFrame();
Output_GetSegmentChildName Output;
// Segment index must be between 0 and GetSegmentChildCount() - 1
Output = MyClient.GetSegmentChildName( "Bob", "Pelvis", 0 );
```

### Parameters

<i>SubjectName</i>	The name of the subject
<i>SegmentName</i>	The name of the segment
<i>SegmentIndex</i>	The index of the child segment. A valid Segment Index is between 0 and <a href="#">GetSegmentChildCount()</a> -1.

### Returns

An Output\_GetSegmentChildName class containing the result of the operation and the name of the child segment.



- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidIndex
  - InvalidSubjectName
  - InvalidSegmentName

## Output\_GetSegmentParentName GetSegmentParentName ( const String & SubjectName, const String & SegmentName ) const

Return the name of the parent segment for a specified subject segment.

If the specified segment is the root segment of the subject then it will return an empty string.

See Also: [GetSegmentCount\(\)](#), [GetSegmentChildCount\(\)](#), [GetSegmentChildName\(\)](#), [GetSubjectRootSegmentName\(\)](#)

### C example

```
CRetimingClient * pRetimingClient = RetimingClient_Create();
RetimingClient_Connect( pRetimingClient, "localhost" );
RetimingClient_GetFrame( pRetimingClient );
char SegmentParentName[128];
CEnum Result = RetimingClient_GetSegmentParentName(
    pRetimingClient, "Bob", "Pelvis", 128, SegmentParentName);
// Result == Success
// SegmentParentName = ""
// This is the root segment
Result = RetimingClient_GetSegmentParentName(pRetimingClient, "Bob", "LFemur", 128, SegmentParentName);
// Result == Success
// SegmentParentName = "Pelvis"
RetimingClient_Destroy( pRetimingClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::RetimingClient MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableSegmentData();
MyClient.GetFrame();
Output_GetSegmentParentName Output;
Output = MyClient.GetSegmentParentName( "Bob", "Pelvis" );
// Output.Result == Success
// Output.SegmentName == ""
// This is the root segment
Output = MyClient.GetSegmentParentName( "Bob", "LFemur" );
// Output.Result == Success
// Output.SegmentName == "Pelvis"
```

### MATLAB example

See .NET example

### .NET example

```
ViconDataStreamSDK.DotNET.RetimingClient MyClient = new ViconDataStreamSDK.DotNET.RetimingClient();
MyClient.Connect( "localhost" );
MyClient.UpdateFrame();
Output_GetSegmentParentName Output;
Output = MyClient.GetSegmentParentName( "Bob", "Pelvis" );
// Output.Result == Success
// Output.SegmentName == ""
// This is the root segment
Output = MyClient.GetSegmentParentName( "Bob", "LFemur" );
// Output.Result == Success
// Output.SegmentName == "Pelvis"
```

## Parameters

<i>SubjectName</i>	The name of the subject
<i>SegmentName</i>	The name of the segment

## Returns

An `Output_GetSegmentParentName` class containing the result of the operation and the name of the parent segment or an empty string if it is the root segment.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame
  - InvalidSubjectName
  - InvalidSegmentName

## Output\_GetSegmentStaticTranslation GetSegmentStaticTranslation ( const String & *SubjectName*, const String & *SegmentName* ) const

Return the static pose translation of a subject segment.

See Also: [GetSegmentStaticRotationHelical\(\)](#), [GetSegmentStaticRotationMatrix\(\)](#), [GetSegmentStaticRotationQuaternion\(\)](#), [GetSegmentStaticRotationEulerXYZ\(\)](#), [GetSegmentLocalTranslation\(\)](#), [GetSegmentLocalRotationHelical\(\)](#), [GetSegmentLocalRotationMatrix\(\)](#), [GetSegmentLocalRotationQuaternion\(\)](#), [GetSegmentLocalRotationEulerXYZ\(\)](#)

### C example

```
CRetimingClient * pRetimingClient = RetimingClient_Create();
RetimingClient_Connect( pRetimingClient, "localhost" );
RetimingClient_GetFrame( pRetimingClient );
COutput_GetSegmentStaticTranslation _Output_GetSegmentStaticTranslation;
RetimingClient_GetSegmentStaticTranslation(
    pRetimingClient, "Alice", "Pelvis", &_Output_GetSegmentStaticTranslation);
RetimingClient_Destroy( pRetimingClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::RetimingClient MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableSegmentData();
MyClient.GetFrame();
Output_GetSegmentStaticTranslation Output =
MyClient.GetSegmentStaticTranslation( "Alice", "Pelvis" );
```

### MATLAB example

See .NET example

### .NET example

```
ViconDataStreamSDK.DotNET.RetimingClient MyClient = new ViconDataStreamSDK.DotNET.RetimingClient();
MyClient.Connect( "localhost" );
MyClient.UpdateFrame();
Output_GetSegmentStaticTranslation Output =
MyClient.GetSegmentStaticTranslation( "Alice", "Pelvis" );
```

### Parameters

<i>SubjectName</i>	The name of the subject
<i>SegmentName</i>	The name of the segment

### Returns

An Output\_GetSegmentStaticTranslation class containing the result of the operation and the translation of the segment.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame

- InvalidSubjectName
- InvalidSegmentName

## Output\_GetSegmentStaticRotationHelical GetSegmentStaticRotationHelical ( const String & *SubjectName*, const String & *SegmentName* ) const

Return the static pose rotation of a subject segment in helical coordinates.

The helical coordinates represent a vector whose length is the amount of rotation in radians, and the direction is the axis about which to rotate.

### C example

```
CRetimingClient * pRetimingClient = RetimingClient_Create();
RetimingClient_Connect( pRetimingClient, "localhost" );
RetimingClient_GetFrame( pRetimingClient );
COutput_GetSegmentStaticRotationHelical _Output_GetSegmentStaticRotationHelical;
RetimingClient_GetSegmentStaticRotationHelical(
    pRetimingClient, "Alice", "Pelvis", &_Output_GetSegmentStaticRotationHelical);
RetimingClient_Destroy( pRetimingClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::RetimingClient MyClient;
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetSegmentStaticRotationHelical Output =
MyClient.GetSegmentStaticRotationHelical( "Alice", "Pelvis" );
```

### MATLAB example

See .NET example

### .NET example

```
ViconDataStreamSDK.DotNET.RetimingClient MyClient = new ViconDataStreamSDK.DotNET.RetimingClient();
MyClient.Connect( "localhost" );
MyClient.UpdateFrame();
Output_GetSegmentStaticRotationHelical Output =
MyClient.GetSegmentStaticRotationHelical( "Alice", "Pelvis" );
```

See Also: [GetSegmentStaticTranslation\(\)](#), [GetSegmentStaticRotationMatrix\(\)](#), [GetSegmentStaticRotationQuaternion\(\)](#), [GetSegmentStaticRotationEulerXYZ\(\)](#), [GetSegmentLocalTranslation](#), [GetSegmentLocalRotationHelical](#), [GetSegmentLocalRotationMatrix](#), [GetSegmentLocalRotationQuaternion](#), [GetSegmentLocalRotationEulerXYZ](#)

### Parameters

<i>SubjectName</i>	The name of the subject
<i>SegmentName</i>	The name of the segment

### Returns

An Output\_GetSegmentStaticRotationHelical class containing the result of the operation and the rotation of the segment.

- The Result will be:
  - Success

- NotConnected
- NoFrame
- InvalidSubjectName
- InvalidSegmentName

## Output\_GetSegmentStaticRotationMatrix GetSegmentStaticRotationMatrix ( const String & *SubjectName*, const String & *SegmentName* ) const

Return the static pose rotation of a subject segment as a 3x3 row-major matrix.

See Also: [GetSegmentStaticTranslation\(\)](#), [GetSegmentStaticRotationHelical\(\)](#), [GetSegmentStaticRotationQuaternion\(\)](#), [GetSegmentStaticRotationEulerXYZ\(\)](#), [GetSegmentLocalTranslation\(\)](#), [GetSegmentLocalRotationHelical\(\)](#), [GetSegmentLocalRotationQuaternion\(\)](#), [GetSegmentLocalRotationEulerXYZ\(\)](#)

### C example

```
CRetimingClient * pRetimingClient = RetimingClient_Create();
RetimingClient_Connect( pRetimingClient, "localhost" );
RetimingClient_GetFrame( pRetimingClient );
COutput_GetSegmentStaticRotationMatrix _Output_GetSegmentStaticRotationMatrix;
RetimingClient_GetSegmentStaticRotationMatrix(
    pRetimingClient, "Alice", "Pelvis", &_Output_GetSegmentStaticRotationMatrix);
RetimingClient_Destroy( pRetimingClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::RetimingClient MyClient;
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetSegmentStaticRotationMatrix Output =
MyClient.GetSegmentStaticRotationMatrix( "Alice", "Pelvis" );
```

### MATLAB example

See [.NET example](#)

### .NET example

```
ViconDataStreamSDK.DotNET.RetimingClient MyClient = new ViconDataStreamSDK.DotNET.RetimingClient();
MyClient.Connect( "localhost" );
MyClient.UpdateFrame();
Output_GetSegmentStaticRotationMatrix Output =
MyClient.GetSegmentStaticRotationMatrix( "Alice", "Pelvis" );
```

### Parameters

<i>SubjectName</i>	The name of the subject
<i>SegmentName</i>	The name of the segment

### Returns

An `Output_GetSegmentStaticRotationMatrix` class containing the result of the operation and the rotation of the segment.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame



- InvalidSubjectName
- InvalidSegmentName

## Output\_GetSegmentStaticRotationQuaternion GetSegmentStaticRotationQuaternion ( const String & *SubjectName*, const String & *SegmentName* ) const

Return the static pose rotation of a subject segment in quaternion coordinates.

The quaternion is of the form (x, y, z, w) where w is the real component and x, y and z are the imaginary components. N.B. This is different from that used in many other applications, which use (w, x, y, z).

See Also: [GetSegmentStaticTranslation\(\)](#), [GetSegmentStaticRotationHelical\(\)](#), [GetSegmentStaticRotationMatrix\(\)](#), [GetSegmentStaticRotationEulerXYZ\(\)](#), [GetSegmentLocalTranslation\(\)](#), [GetSegmentLocalRotationHelical\(\)](#), [GetSegmentLocalRotationMatrix\(\)](#), [GetSegmentLocalRotationQuaternion\(\)](#), [GetSegmentLocalRotationEulerXYZ\(\)](#)

### C example

```
CRetimingClient * pRetimingClient = RetimingClient_Create();
RetimingClient_Connect( pRetimingClient, "localhost" );
RetimingClient_GetFrame( pRetimingClient );
COutput_GetSegmentStaticRotationQuaternion _Output_GetSegmentStaticRotationQuaternion;
RetimingClient_GetSegmentStaticRotationQuaternion(
    pRetimingClient, "Alice", "Pelvis", &_Output_GetSegmentStaticRotationQuaternion);
RetimingClient_Destroy( pRetimingClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::RetimingClient MyClient;
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetSegmentStaticRotationQuaternion Output =
MyClient.GetSegmentStaticRotationQuaternion( "Alice", "Pelvis" );
```

### MATLAB example

See .NET example

### .NET example

```
ViconDataStreamSDK.DotNET.RetimingClient MyClient = new ViconDataStreamSDK.DotNET.RetimingClient();
MyClient.Connect( "localhost" );
MyClient.UpdateFrame();
Output_GetSegmentStaticRotationQuaternion Output =
MyClient.GetSegmentStaticRotationQuaternion( "Alice", "Pelvis" );
```

### Parameters

<i>SubjectName</i>	The name of the subject
<i>SegmentName</i>	The name of the segment

### Returns

An Output\_GetSegmentStaticRotationQuaternion class containing the result of the operation and the rotation of the segment.

- The Result will be:
  - Success

- NotConnected
- NoFrame
- InvalidSubjectName
- InvalidSegmentName

## Output\_GetSegmentStaticRotationEulerXYZ GetSegmentStaticRotationEulerXYZ ( const String & *SubjectName*, const String & *SegmentName* ) const

Return the static pose rotation of a subject segment in Euler XYZ coordinates.

See Also: [GetSegmentStaticTranslation\(\)](#), [GetSegmentStaticRotationHelical\(\)](#), [GetSegmentStaticRotationMatrix\(\)](#), [GetSegmentStaticRotationQuaternion\(\)](#), [GetSegmentLocalTranslation\(\)](#), [GetSegmentLocalRotationHelical\(\)](#), [GetSegmentLocalRotationMatrix\(\)](#), [GetSegmentLocalRotationQuaternion\(\)](#), [GetSegmentLocalRotationEulerXYZ\(\)](#).

### C example

```
CRetimingClient * pRetimingClient = RetimingClient_Create();
RetimingClient_Connect( pRetimingClient, "localhost" );
RetimingClient_GetFrame( pRetimingClient );
COutput_GetSegmentStaticRotationEulerXYZ _Output_GetSegmentStaticRotationEulerXYZ;
RetimingClient_GetSegmentStaticRotationEulerXYZ(
    pRetimingClient, "Alice", "Pelvis", &_Output_GetSegmentStaticRotationEulerXYZ);
RetimingClient_Destroy( pRetimingClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::RetimingClient MyClient;
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetSegmentStaticRotationEulerXYZ Output;
Output = MyClient.GetSegmentStaticRotationEulerXYZ( "Alice", "Pelvis" );
```

### MATLAB example

See [.NET example](#)

### .NET example

```
ViconDataStreamSDK.DotNET.RetimingClient MyClient = new ViconDataStreamSDK.DotNET.RetimingClient();
MyClient.Connect( "localhost" );
MyClient.UpdateFrame();
Output_GetSegmentStaticRotationEulerXYZ Output;
Output = MyClient.GetSegmentStaticRotationEulerXYZ( "Alice", "Pelvis" );
```

### Parameters

<i>SubjectName</i>	The name of the subject.
<i>SegmentName</i>	The name of the segment.

### Returns

An `Output_GetSegmentStaticRotationEulerXYZ` class containing the result of the request and the rotation of the segment ( $x, y, z$ ).

- The Result will be:
  - Success
  - NotConnected
  - NoFrame

- InvalidSubjectName
- InvalidSegmentName

## Output\_GetSegmentGlobalTranslation GetSegmentGlobalTranslation ( const String & SubjectName, const String & SegmentName ) const

Return the translation of a subject segment in global coordinates.

The translation is of the form ( x, y, z ) where x, y and z are in millimeters with respect to the global origin.

See Also: [GetSegmentGlobalRotationHelical\(\)](#), [GetSegmentGlobalRotationMatrix\(\)](#), [GetSegmentGlobalRotationQuaternion\(\)](#), [GetSegmentGlobalRotationEulerXYZ\(\)](#), [GetSegmentLocalTranslation\(\)](#), [GetSegmentLocalRotationHelical\(\)](#), [GetSegmentLocalRotationMatrix\(\)](#), [GetSegmentLocalRotationQuaternion\(\)](#), [GetSegmentLocalRotationEulerXYZ\(\)](#)

### C example

```
CRetimingClient * pRetimingClient = RetimingClient_Create();
RetimingClient_Connect( pRetimingClient, "localhost" );
RetimingClient_GetFrame( pRetimingClient );
COutput_GetSegmentGlobalTranslation _Output_GetSegmentGlobalTranslation;
RetimingClient_GetSegmentGlobalTranslation(
    pRetimingClient, "Alice", "Pelvis", &_Output_GetSegmentGlobalTranslation);
RetimingClient_Destroy( pRetimingClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::RetimingClient MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableSegmentData();
MyClient.GetFrame();
Output_GetSegmentGlobalTranslation Output =
MyClient.GetSegmentGlobalTranslation( "Alice", "Pelvis" );
```

### MATLAB example

See .NET example

### .NET example

```
ViconDataStreamSDK.DotNET.RetimingClient MyClient = new ViconDataStreamSDK.DotNET.RetimingClient();
MyClient.Connect( "localhost" );
MyClient.UpdateFrame();
Output_GetSegmentGlobalTranslation Output =
MyClient.GetSegmentGlobalTranslation( "Alice", "Pelvis" );
```

### Parameters

<i>SubjectName</i>	The name of the subject.
<i>SegmentName</i>	The name of the segment.

### Returns

An Output\_GetSegmentGlobalTranslation class containing the result of the operation, the translation of the segment, and whether the segment is occluded.

- The Result will be:
  - Success

- NotConnected
  - NoFrame
  - InvalidSubjectName
  - InvalidSegmentName
- Occluded will be True if the segment was absent at this frame. In this case the translation will be [0,0,0].

## Output\_GetSegmentGlobalRotationHelical GetSegmentGlobalRotationHelical ( const String & *SubjectName*, const String & *SegmentName* ) const

Return the rotation of a subject segment in global helical coordinates.

See Also: [GetSegmentGlobalTranslation\(\)](#), [GetSegmentGlobalRotationMatrix\(\)](#), [GetSegmentGlobalRotationQuaternion\(\)](#), [GetSegmentGlobalRotationEulerXYZ\(\)](#), [GetSegmentLocalTranslation\(\)](#), [GetSegmentLocalRotationHelical\(\)](#), [GetSegmentLocalRotationMatrix\(\)](#), [GetSegmentLocalRotationQuaternion\(\)](#), [GetSegmentLocalRotationEulerXYZ\(\)](#)

### C example

```
CRetimingClient * pRetimingClient = RetimingClient_Create();
RetimingClient_Connect( pRetimingClient, "localhost" );
RetimingClient_GetFrame( pRetimingClient );
COutput_GetSegmentGlobalRotationHelical _Output_GetSegmentGlobalRotationHelical;
RetimingClient_GetSegmentGlobalRotationHelical(
    pRetimingClient, "Alice", "Pelvis", &_Output_GetSegmentGlobalRotationHelical);
RetimingClient_Destroy( pRetimingClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::RetimingClient MyClient;
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetSegmentGlobalRotationHelical Output =
MyClient.GetSegmentGlobalRotationHelical( "Alice", "Pelvis" );
```

### MATLAB example

See .NET example

### .NET example

```
ViconDataStreamSDK.DotNET.RetimingClient MyClient = new ViconDataStreamSDK.DotNET.RetimingClient();
MyClient.Connect( "localhost" );
MyClient.UpdateFrame();
Output_GetSegmentGlobalRotationHelical Output =
MyClient.GetSegmentGlobalRotationHelical( "Alice", "Pelvis" );
```

### Parameters

<i>SubjectName</i>	The name of the subject.
<i>SegmentName</i>	The name of the segment.

### Returns

An Output\_GetSegmentGlobalRotationHelical class containing the result of the operation, the rotation of the segment, and whether the segment is occluded.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame



- InvalidSubjectName
  - InvalidSegmentName
- Occluded will be True if the segment was absent at this frame. In this case, the rotation will be [0,0,0].

## Output\_GetSegmentGlobalRotationMatrix GetSegmentGlobalRotationMatrix ( const String & *SubjectName*, const String & *SegmentName* ) const

Return the rotation of a subject segment as a 3x3 row-major matrix in global coordinates.

See Also: [GetSegmentGlobalTranslation\(\)](#), [GetSegmentGlobalRotationHelical\(\)](#), [GetSegmentGlobalRotationQuaternion\(\)](#), [GetSegmentGlobalRotationEulerXYZ\(\)](#), [GetSegmentLocalTranslation\(\)](#), [GetSegmentLocalRotationHelical\(\)](#), [GetSegmentLocalRotationQuaternion\(\)](#), [GetSegmentLocalRotationEulerXYZ\(\)](#)

### C example

```
CRetimingClient * pRetimingClient = RetimingClient_Create();
RetimingClient_Connect( pRetimingClient, "localhost" );
RetimingClient_GetFrame( pRetimingClient );
COutput_GetSegmentGlobalRotationMatrix _Output_GetSegmentGlobalRotationMatrix;
RetimingClient_GetSegmentGlobalRotationMatrix(
    pRetimingClient, "Alice", "Pelvis", &_Output_GetSegmentGlobalRotationMatrix);
RetimingClient_Destroy( pRetimingClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::RetimingClient MyClient;
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetSegmentGlobalRotationMatrix Output =
MyClient.GetSegmentGlobalRotationMatrix( "Alice", "Pelvis" );
```

### MATLAB example

See .NET example

### .NET example

```
ViconDataStreamSDK.DotNET.RetimingClient MyClient = new ViconDataStreamSDK.DotNET.RetimingClient();
MyClient.Connect( "localhost" );
MyClient.UpdateFrame();
Output_GetSegmentGlobalRotationMatrix Output =
MyClient.GetSegmentGlobalRotationMatrix( "Alice", "Pelvis" );
```

### Parameters

<i>SubjectName</i>	The name of the subject.
<i>SegmentName</i>	The name of the segment.

### Returns

An Output\_GetSegmentGlobalRotationMatrix Class containing the result of the operation, the rotation of the segment, and whether the segment is occluded.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame

- InvalidSubjectName
- InvalidSegmentName
- Occluded will be True if the segment was absent at this frame.

## Output\_GetSegmentGlobalRotationQuaternion GetSegmentGlobalRotationQuaternion ( const String & *SubjectName*, const String & *SegmentName* ) const

Return the rotation of a subject segment in global quaternion coordinates.

The quaternion is of the form (x, y, z, w) where w is the real component and x, y and z are the imaginary components. N.B. This is different from that used in many other applications, which use (w, x, y, z).

See Also: [GetSegmentGlobalTranslation\(\)](#), [GetSegmentGlobalRotationHelical\(\)](#), [GetSegmentGlobalRotationMatrix\(\)](#), [GetSegmentGlobalRotationEulerXYZ\(\)](#), [GetSegmentLocalTranslation\(\)](#), [GetSegmentLocalRotationHelical\(\)](#), [GetSegmentLocalRotationMatrix\(\)](#), [GetSegmentLocalRotationQuaternion\(\)](#), [GetSegmentLocalRotationEulerXYZ\(\)](#)

### C example

```
CRetimingClient * pRetimingClient = RetimingClient_Create();
RetimingClient_Connect( pRetimingClient, "localhost" );
RetimingClient_GetFrame( pRetimingClient );
COutput_GetSegmentGlobalRotationQuaternion _Output_GetSegmentGlobalRotationQuaternion;
RetimingClient_GetSegmentGlobalRotationQuaternion(
    pRetimingClient, "Alice", "Pelvis", &_Output_GetSegmentGlobalRotationQuaternion);
RetimingClient_Destroy( pRetimingClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::RetimingClient MyClient;
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetSegmentGlobalRotationQuaternion Output =
MyClient.GetSegmentGlobalRotationQuaternion( "Alice", "Pelvis" );
```

### MATLAB example

See .NET example

### .NET example

```
ViconDataStreamSDK.DotNET.RetimingClient MyClient = new ViconDataStreamSDK.DotNET.RetimingClient();
MyClient.Connect( "localhost" );
MyClient.UpdateFrame();
Output_GetSegmentGlobalRotationQuaternion Output =
MyClient.GetSegmentGlobalRotationQuaternion( "Alice", "Pelvis" );
```

### Parameters

<i>SubjectName</i>	The name of the subject.
<i>SegmentName</i>	The name of the segment.

### Returns

An [Output\\_GetSegmentGlobalRotationQuaternion](#) class containing the result of the operation, the rotation of the segment, and whether the segment is occluded.

- The Result will be:

- Success
- NotConnected
- NoFrame
- InvalidSubjectName
- InvalidSegmentName
- Occluded will be True if the segment was absent at this frame. In this case the Rotation will be [1,0,0,0].

## Output\_GetSegmentGlobalRotationEulerXYZ GetSegmentGlobalRotationEulerXYZ ( const String & *SubjectName*, const String & *SegmentName* ) const

Return the rotation of a subject segment in global Euler XYZ coordinates.

See Also: [GetSegmentGlobalTranslation\(\)](#), [GetSegmentGlobalRotationHelical\(\)](#), [GetSegmentGlobalRotationMatrix\(\)](#), [GetSegmentGlobalRotationQuaternion\(\)](#), [GetSegmentLocalTranslation\(\)](#), [GetSegmentLocalRotationHelical\(\)](#), [GetSegmentLocalRotationMatrix\(\)](#), [GetSegmentLocalRotationQuaternion\(\)](#), [GetSegmentLocalRotationEulerXYZ\(\)](#)

### C example

```
CRetimingClient * pRetimingClient = RetimingClient_Create();
RetimingClient_Connect( pRetimingClient, "localhost" );
RetimingClient_GetFrame( pRetimingClient );
COutput_GetSegmentGlobalRotationEulerXYZ _Output_GetSegmentGlobalRotationEulerXYZ;
RetimingClient_GetSegmentGlobalRotationEulerXYZ(
    pRetimingClient, "Alice", "Pelvis", &_Output_GetSegmentGlobalRotationEulerXYZ);
RetimingClient_Destroy( pRetimingClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::RetimingClient MyClient;
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetSegmentGlobalRotationEulerXYZ Output =
MyClient.GetSegmentGlobalRotationEulerXYZ( "Alice", "Pelvis" );
```

### MATLAB example

See .NET example

### .NET example

```
ViconDataStreamSDK.DotNET.RetimingClient MyClient = new ViconDataStreamSDK.DotNET.RetimingClient();
MyClient.Connect( "localhost" );
MyClient.UpdateFrame();
Output_GetSegmentGlobalRotationEulerXYZ Output =
MyClient.GetSegmentGlobalRotationEulerXYZ( "Alice", "Pelvis" );
```

### Parameters

<i>SubjectName</i>	The name of the subject.
<i>SegmentName</i>	The name of the segment.

### Returns

An Output\_GetSegmentGlobalRotationEulerXYZ class containing the result of the operation, the rotation of the segment, and whether the segment is occluded.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame

- InvalidSubjectName
  - InvalidSegmentName
- Occluded will be True if the segment was absent at this frame. In this case the rotation will be [0,0,0].

## Output\_GetSegmentLocalTranslation GetSegmentLocalTranslation ( const String & SubjectName, const String & SegmentName ) const

Return the translation of a subject segment in local coordinates relative to its parent segment.

See Also: [GetSegmentLocalRotationHelical\(\)](#), [GetSegmentLocalRotationMatrix\(\)](#), [GetSegmentLocalRotationQuaternion\(\)](#), [GetSegmentLocalRotationEulerXYZ\(\)](#), [GetSegmentGlobalTranslation\(\)](#), [GetSegmentGlobalRotationHelical\(\)](#), [GetSegmentGlobalRotationMatrix\(\)](#), [GetSegmentGlobalRotationQuaternion\(\)](#), [GetSegmentGlobalRotationEulerXYZ\(\)](#)

### C example

```
CRetimingClient * pRetimingClient = RetimingClient_Create();
RetimingClient_Connect( pRetimingClient, "localhost" );
RetimingClient_GetFrame( pRetimingClient );
COutput_GetSegmentLocalTranslation _Output_GetSegmentLocalTranslation;
RetimingClient_GetSegmentLocalTranslation(
    pRetimingClient, "Alice", "Pelvis", &_Output_GetSegmentLocalTranslation);
RetimingClient_Destroy( pRetimingClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::RetimingClient MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableSegmentData();
MyClient.GetFrame();
Output_GetSegmentLocalTranslation Output =
MyClient.GetSegmentLocalTranslation( "Alice", "Pelvis" );
```

### MATLAB example

See .NET example

### .NET example

```
ViconDataStreamSDK.DotNET.RetimingClient MyClient = new ViconDataStreamSDK.DotNET.RetimingClient();
MyClient.Connect( "localhost" );
MyClient.UpdateFrame();
Output_GetSegmentLocalTranslation Output =
MyClient.GetSegmentLocalTranslation( "Alice", "Pelvis" );
```

### Parameters

<i>SubjectName</i>	The name of the subject.
<i>SegmentName</i>	The name of the segment.

### Returns

An Output\_GetSegmentLocalTranslation class containing the result of the operation, the translation of the segment, and whether the segment is occluded.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame



- InvalidSubjectName
  - InvalidSegmentName
- Occluded will be True if the segment was absent at this frame. In this case the translation will be [0,0,0].

## Output\_GetSegmentLocalRotationHelical GetSegmentLocalRotationHelical ( const String & *SubjectName*, const String & *SegmentName* ) const

Return the rotation of a subject segment in local helical coordinates relative to its parent segment.

See Also: [GetSegmentLocalTranslation\(\)](#), [GetSegmentLocalRotationMatrix\(\)](#), [GetSegmentLocalRotationQuaternion\(\)](#), [GetSegmentLocalRotationEulerXYZ\(\)](#), [GetSegmentGlobalTranslation\(\)](#), [GetSegmentGlobalRotationHelical\(\)](#), [GetSegmentGlobalRotationMatrix\(\)](#), [GetSegmentGlobalRotationQuaternion\(\)](#), [GetSegmentGlobalRotationEulerXYZ\(\)](#)

### C example

```
CRetimingClient * pRetimingClient = RetimingClient_Create();
RetimingClient_Connect( pRetimingClient, "localhost" );
RetimingClient_GetFrame( pRetimingClient );
COutput_GetSegmentLocalRotationHelical _Output_GetSegmentLocalRotationHelical;
RetimingClient_GetSegmentLocalRotationHelical(
    pRetimingClient, "Alice", "Pelvis", &_Output_GetSegmentLocalRotationHelical);
RetimingClient_Destroy( pRetimingClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::RetimingClient MyClient;
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetSegmentLocalRotationHelical Output =
MyClient.GetSegmentLocalRotationHelical( "Alice", "Pelvis" );
```

### MATLAB example

See .NET example

### .NET example

```
ViconDataStreamSDK.DotNET.RetimingClient MyClient = new ViconDataStreamSDK.DotNET.RetimingClient();
MyClient.Connect( "localhost" );
MyClient.UpdateFrame();
Output_GetSegmentLocalRotationHelical Output =
MyClient.GetSegmentLocalRotationHelical( "Alice", "Pelvis" );
```

### Parameters

<i>SubjectName</i>	The name of the subject.
<i>SegmentName</i>	The name of the segment.

### Returns

An Output\_GetSegmentLocalRotationHelical class containing the result of the operation, the rotation of the segment, and whether the segment is occluded.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame

- InvalidSubjectName
  - InvalidSegmentName
- Occluded will be True if the segment was absent at this frame. In this case the Rotation will be [0,0,0].

## Output\_GetSegmentLocalRotationMatrix GetSegmentLocalRotationMatrix ( const String & *SubjectName*, const String & *SegmentName* ) const

Return the rotation row-major matrix of a subject segment in local coordinates relative to its parent segment.

See Also: [GetSegmentLocalTranslation\(\)](#), [GetSegmentLocalRotationQuaternion\(\)](#), [GetSegmentLocalRotationEulerXYZ\(\)](#), [GetSegmentGlobalTranslation\(\)](#), [GetSegmentGlobalRotationHelical\(\)](#), [GetSegmentGlobalRotationMatrix\(\)](#) , [GetSegmentGlobalRotationQuaternion\(\)](#), [GetSegmentGlobalRotationEulerXYZ\(\)](#)

### C example

```
CRetimingClient * pRetimingClient = RetimingClient_Create();
RetimingClient_Connect( pRetimingClient, "localhost" );
RetimingClient_GetFrame( pRetimingClient );
COutput_GetSegmentLocalRotationMatrix _Output_GetSegmentLocalRotationMatrix;
RetimingClient_GetSegmentLocalRotationMatrix(
    pRetimingClient, "Alice", "Pelvis", &_Output_GetSegmentLocalRotationMatrix);
RetimingClient_Destroy( pRetimingClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::RetimingClient MyClient;
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetSegmentLocalRotationMatrix Output =
MyClient.GetSegmentLocalRotationMatrix( "Alice", "Pelvis" );
```

### MATLAB example

See [.NET example](#)

### .NET example

```
ViconDataStreamSDK.DotNET.RetimingClient MyClient = new ViconDataStreamSDK.DotNET.RetimingClient();
MyClient.Connect( "localhost" );
MyClient.UpdateFrame();
Output_GetSegmentLocalRotationMatrix Output =
MyClient.GetSegmentLocalRotationMatrix( "Alice", "Pelvis" );
```

### Parameters

<i>SubjectName</i>	The name of the subject.
<i>SegmentName</i>	The name of the segment.

### Returns

An `Output_GetSegmentLocalRotationMatrix` class containing the result of the operation, the rotation of the segment, and whether the segment is occluded.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame

- InvalidSubjectName
  - InvalidSegmentName
- Occluded will be True if the segment was absent at this frame.

## Output\_GetSegmentLocalRotationQuaternion GetSegmentLocalRotationQuaternion ( const String & *SubjectName*, const String & *SegmentName* ) const

Return the rotation of a subject segment in local quaternion coordinates relative to its parent segment.

The quaternion is of the form (x, y, z, w) where w is the real component and x, y and z are the imaginary components. N.B. This is different from that used in many other applications, which use (w, x, y, z).

See Also: [GetSegmentLocalTranslation\(\)](#), [GetSegmentLocalRotationHelical\(\)](#), [GetSegmentLocalRotationMatrix\(\)](#), [GetSegmentLocalRotationEulerXYZ\(\)](#), [GetSegmentGlobalTranslation\(\)](#), [GetSegmentGlobalRotationHelical\(\)](#), [GetSegmentGlobalRotationMatrix\(\)](#), [GetSegmentGlobalRotationQuaternion\(\)](#), [GetSegmentGlobalRotationEulerXYZ\(\)](#)

### C example

```
CRetimingClient * pRetimingClient = RetimingClient_Create();
RetimingClient_Connect( pRetimingClient, "localhost" );
RetimingClient_GetFrame( pRetimingClient );
COutput_GetSegmentLocalRotationQuaternion _Output_GetSegmentLocalRotationQuaternion;
RetimingClient_GetSegmentLocalRotationQuaternion(
    pRetimingClient, "Alice", "Pelvis", &_Output_GetSegmentLocalRotationQuaternion);
RetimingClient_Destroy( pRetimingClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::RetimingClient MyClient;
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetSegmentLocalRotationQuaternion Output =
MyClient.GetSegmentLocalRotationQuaternion( "Alice", "Pelvis" );
```

### MATLAB example

See .NET example

### .NET example

```
ViconDataStreamSDK.DotNET.RetimingClient MyClient = new ViconDataStreamSDK.DotNET.RetimingClient();
MyClient.Connect( "localhost" );
MyClient.UpdateFrame();
Output_GetSegmentLocalRotationQuaternion Output =
MyClient.GetSegmentLocalRotationQuaternion( "Alice", "Pelvis" );
```

### Parameters

<i>SubjectName</i>	The name of the subject.
<i>SegmentName</i>	The name of the segment.

### Returns

An Output\_GetSegmentLocalRotationQuaternion class containing the result of the operation, the rotation of the segment, and whether the segment is occluded.

- The Result will be:
  - Success

- NotConnected
  - NoFrame
  - InvalidSubjectName
  - InvalidSegmentName
- Occluded will be True if the segment was absent at this frame. In this case the rotation will be [1,0,0,0].

## Output\_GetSegmentLocalRotationEulerXYZ GetSegmentLocalRotationEulerXYZ ( const String & *SubjectName*, const String & *SegmentName* ) const

Return the rotation of a subject segment in local Euler XYZ coordinates relative to its parent segment.

See Also: [GetSegmentLocalTranslation\(\)](#), [GetSegmentLocalRotationHelical\(\)](#), [GetSegmentLocalRotationMatrix\(\)](#), [GetSegmentLocalRotationQuaternion\(\)](#), [GetSegmentGlobalTranslation\(\)](#), [GetSegmentGlobalRotationHelical\(\)](#), [GetSegmentGlobalRotationMatrix\(\)](#) , [GetSegmentGlobalRotationQuaternion\(\)](#), [GetSegmentGlobalRotationEulerXYZ\(\)](#)

### C example

```
CRetimingClient * pRetimingClient = RetimingClient_Create();
RetimingClient_Connect( pRetimingClient, "localhost" );
RetimingClient_GetFrame( pRetimingClient );
COutput_GetSegmentLocalRotationEulerXYZ _Output_GetSegmentLocalRotationEulerXYZ;
RetimingClient_GetSegmentLocalRotationEulerXYZ(
    pRetimingClient, "Alice", "Pelvis", &_Output_GetSegmentLocalRotationEulerXYZ);
RetimingClient_Destroy( pRetimingClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::RetimingClient MyClient;
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetSegmentLocalRotationEulerXYZ Output =
MyClient.GetSegmentLocalRotationEulerXYZ( "Alice", "Pelvis" );
```

### MATLAB example

See .NET example

### .NET example

```
ViconDataStreamSDK.DotNET.RetimingClient MyClient = new ViconDataStreamSDK.DotNET.RetimingClient();
MyClient.Connect( "localhost" );
MyClient.UpdateFrame();
Output_GetSegmentLocalRotationEulerXYZ Output =
MyClient.GetSegmentLocalRotationEulerXYZ( "Alice", "Pelvis" );
```

### Parameters

<i>SubjectName</i>	The name of the subject.
<i>SegmentName</i>	The name of the segment.

### Returns

An Output\_GetSegmentLocalRotationEulerXYZ class containing the result of the operation, the rotation of the segment, and whether the segment is occluded.

- The Result will be:
  - Success
  - NotConnected
  - NoFrame



- InvalidSubjectName
  - InvalidSegmentName
- Occluded will be True if the segment was absent at this frame. In this case the rotation will be [0,0,0].

## void SetMaximumPrediction ( double *MaxPrediction* )

Sets the maximum amount by which the interpolation engine will predict later than the latest received frame.

If required to predict by more than this amount, the result LateDataRequested will be returned.

### C example

```
CRetimingClient * pRetimingClient = RetimingClient_Create();
RetimingClient_SetMaximumPrediction( pRetimingClient, 30 );
RetimingClient_Connect( pRetimingClient, "localhost" );
RetimingClient_GetFrame( pRetimingClient );
RetimingClient_Destroy( pRetimingClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::RetimingClient MyClient;
MyClient.SetMaximumPrediction( 30 );
MyClient.Connect( "localhost" );
MyClient.GetFrame();
```

### MATLAB example

See .NET example

### .NET example

```
ViconDataStreamSDK.DotNET.RetimingClient MyClient = new ViconDataStreamSDK.DotNET.RetimingClient();
MyClient.SetMaximumPrediction( 30 );
MyClient.Connect( "localhost" );
MyClient.UpdateFrame();
```

### Parameters

<i>MaxPrediction</i>	The maximum amount of prediction required in milliseconds
----------------------	---

## double MaximumPrediction ( ) const

Returns the maximum prediction value currently in use.

The default value is 100 ms.

### C example

```
CRetimingClient * pRetimingClient = RetimingClient_Create();
RetimingClient_SetMaximumPrediction( pRetimingClient, 30 );
RetimingClient_MaximumPrediction( pRetimingClient ); // Returns 30
RetimingClient_Destroy( pRetimingClient );
```

### C++ example

```
ViconDataStreamSDK::CPP::RetimingClient MyClient;
MyClient.SetMaximumPrediction( 30 );
MyClient.MaximumPrediction(); // Returns 30
```

## MATLAB example

See .NET example

## .NET example

```
ViconDataStreamSDK.DotNET.RetimingClient MyClient = new ViconDataStreamSDK.DotNET.RetimingClient();  
MyClient.SetMaximumPrediction( 30 );  
MyClient.MaximumPrediction(); // Returns 30
```

## Returns

The maximum prediction allowed in milliseconds

The documentation for this class was generated from the following files:

- `DataStreamRetimingClient.h`
- `DataStreamRetimingClient.cpp`

## Index

- ~Client
  - Client, [15](#)
- ~RetimingClient
  - RetimingClient, [218](#)
- AddToSubjectFilter
  - Client, [211](#)
- ClearSubjectFilter
  - Client, [210](#)
- Client, [7](#)
  - ~Client, [15](#)
  - AddToSubjectFilter, [211](#)
  - ClearSubjectFilter, [210](#)
  - Client, [14](#)
  - ConfigureWireless, [213](#)
  - Connect, [17](#)
  - ConnectToMulticast, [18](#)
  - DisableCentroidData, [40](#)
  - DisableDebugData, [43](#)
  - DisableDeviceData, [39](#)
  - DisableGreyscaleData, [41](#)
  - DisableLightweightSegmentData, [35](#)
  - DisableMarkerData, [36](#)
  - DisableMarkerRayData, [38](#)
  - DisableSegmentData, [34](#)
  - DisableUnlabeledMarkerData, [37](#)
  - DisableVideoData, [42](#)
  - Disconnect, [20](#)
  - EnableCentroidData, [30](#)
  - EnableDebugData, [33](#)
  - EnableDeviceData, [29](#)
  - EnableGreyscaleData, [31](#)
  - EnableLightweightSegmentData, [25](#)
  - EnableMarkerData, [26](#)
  - EnableMarkerRayData, [28](#)
  - EnableSegmentData, [24](#)
  - EnableUnlabeledMarkerData, [27](#)
  - EnableVideoData, [32](#)
  - GetAxisMapping, [59](#)
  - GetCameraCount, [185](#)
  - GetCameraDisplayName, [194](#)
  - GetCameraId, [188](#)
  - GetCameraName, [186](#)
  - GetCameraResolution, [196](#)
  - GetCameraType, [192](#)
  - GetCameraUserId, [190](#)
  - GetCentroidCount, [200](#)
  - GetCentroidPosition, [202](#)
  - GetCentroidWeight, [204](#)
  - GetDeviceCount, [141](#)
  - GetDeviceName, [142](#)
  - GetDeviceOutputComponentName, [150](#)
  - GetDeviceOutputCount, [145](#)
  - GetDeviceOutputName, [147](#)
  - GetDeviceOutputSubsamples, [157, 159](#)
  - GetDeviceOutputValue, [153, 155, 161, 163](#)
  - GetEyeTrackerCount, [180](#)
  - GetEyeTrackerGlobalGazeVector, [183](#)
  - GetEyeTrackerGlobalPosition, [181](#)
  - GetForcePlateCount, [165](#)
  - GetForcePlateSubsamples, [172](#)
  - GetFrame, [60](#)
  - GetFrameNumber, [61](#)
  - GetFrameRate, [63](#)
  - GetFrameRateCount, [71](#)
  - GetFrameRateName, [72](#)
  - GetFrameRateValue, [73](#)
  - GetGlobalCentreOfPressure, [170, 178](#)
  - GetGlobalForceVector, [166, 174](#)
  - GetGlobalMomentVector, [168, 176](#)
  - GetGreyscaleBlob, [207](#)
  - GetGreyscaleBlobCount, [206](#)
  - GetHardwareFrameNumber, [70](#)
  - GetIsVideoCamera, [198](#)
  - GetLabeledMarkerCount, [139](#)
  - GetLabeledMarkerGlobalTranslation, [140](#)
  - GetLatencySampleCount, [64](#)
  - GetLatencySampleName, [65](#)
  - GetLatencySampleValue, [67](#)
  - GetLatencyTotal, [69](#)
  - GetMarkerCount, [123](#)
  - GetMarkerGlobalTranslation, [130](#)
  - GetMarkerName, [125](#)
  - GetMarkerParentName, [128](#)

- GetMarkerRayContribution, [134](#)
- GetMarkerRayContributionCount, [132](#)
- GetObjectQuality, [121](#)
- GetSegmentChildCount, [83](#)
- GetSegmentChildName, [85](#)
- GetSegmentCount, [79](#)
- GetSegmentGlobalRotationEulerXYZ, [109](#)
- GetSegmentGlobalRotationHelical, [103](#)
- GetSegmentGlobalRotationMatrix, [105](#)
- GetSegmentGlobalRotationQuaternion, [107](#)
- GetSegmentGlobalTranslation, [101](#)
- GetSegmentLocalRotationEulerXYZ, [119](#)
- GetSegmentLocalRotationHelical, [113](#)
- GetSegmentLocalRotationMatrix, [115](#)
- GetSegmentLocalRotationQuaternion, [117](#)
- GetSegmentLocalTranslation, [111](#)
- GetSegmentName, [81](#)
- GetSegmentParentName, [87](#)
- GetSegmentStaticRotationEulerXYZ, [97](#)
- GetSegmentStaticRotationHelical, [91](#)
- GetSegmentStaticRotationMatrix, [93](#)
- GetSegmentStaticRotationQuaternion, [95](#)
- GetSegmentStaticScale, [99](#)
- GetSegmentStaticTranslation, [89](#)
- GetSubjectCount, [74](#)
- GetSubjectName, [75](#)
- GetSubjectRootSegmentName, [77](#)
- GetTimecode, [62](#)
- GetUnlabeledMarkerCount, [136](#)
- GetUnlabeledMarkerGlobalTranslation, [137](#)
- GetVersion, [15](#)
- GetVideoFrame, [208](#)
- IsCentroidDataEnabled, [50](#)
- IsConnected, [21](#)
- IsDebugEnabled, [53](#)
- IsDeviceDataEnabled, [49](#)
- IsGreyscaleDataEnabled, [51](#)
- IsLightweightSegmentDataEnabled, [45](#)
- IsMarkerDataEnabled, [46](#)
- IsMarkerRayDataEnabled, [48](#)
- IsSegmentDataEnabled, [44](#)
- IsUnlabeledMarkerDataEnabled, [47](#)
- IsVideoDataEnabled, [52](#)
- SetApexDeviceFeedback, [57](#)
- SetAxisMapping, [58](#)
- SetBufferSize, [54](#)
- SetCameraFilter, [209](#)
- SetStreamMode, [55](#)
- StartTransmittingMulticast, [22](#)
- StopTransmittingMulticast, [23](#)
- ConfigureWireless
  - Client, [213](#)
- Connect
  - Client, [17](#)
  - RetimingClient, [220](#)
- ConnectToMulticast
  - Client, [18](#)
- DisableCentroidData
  - Client, [40](#)
- DisableDebugData
  - Client, [43](#)
- DisableDeviceData
  - Client, [39](#)
- DisableGreyscaleData
  - Client, [41](#)
- DisableLightweightSegmentData
  - Client, [35](#)
  - RetimingClient, [224](#)
- DisableMarkerData
  - Client, [36](#)
- DisableMarkerRayData
  - Client, [38](#)
- DisableSegmentData
  - Client, [34](#)
- DisableUnlabeledMarkerData
  - Client, [37](#)
- DisableVideoData
  - Client, [42](#)
- Disconnect
  - Client, [20](#)
  - RetimingClient, [221](#)
- EnableCentroidData
  - Client, [30](#)
- EnableDebugData
  - Client, [33](#)
- EnableDeviceData
  - Client, [29](#)
- EnableGreyscaleData
  - Client, [31](#)
- EnableLightweightSegmentData
  - Client, [25](#)
  - RetimingClient, [223](#)
- EnableMarkerData
  - Client, [26](#)
- EnableMarkerRayData
  - Client, [28](#)
- EnableSegmentData
  - Client, [24](#)
- EnableUnlabeledMarkerData
  - Client, [27](#)
- EnableVideoData

## INDEX

- Client, [32](#)
- GetAxisMapping
  - Client, [59](#)
  - RetimingClient, [227](#)
- GetCameraCount
  - Client, [185](#)
- GetCameraDisplayName
  - Client, [194](#)
- GetCamerald
  - Client, [188](#)
- GetCameraName
  - Client, [186](#)
- GetCameraResolution
  - Client, [196](#)
- GetCameraType
  - Client, [192](#)
- GetCameraUserId
  - Client, [190](#)
- GetCentroidCount
  - Client, [200](#)
- GetCentroidPosition
  - Client, [202](#)
- GetCentroidWeight
  - Client, [204](#)
- GetDeviceCount
  - Client, [141](#)
- GetDeviceName
  - Client, [142](#)
- GetDeviceOutputComponentName
  - Client, [150](#)
- GetDeviceOutputCount
  - Client, [145](#)
- GetDeviceOutputName
  - Client, [147](#)
- GetDeviceOutputSubsamples
  - Client, [157](#), [159](#)
- GetDeviceOutputValue
  - Client, [153](#), [155](#), [161](#), [163](#)
- GetEyeTrackerCount
  - Client, [180](#)
- GetEyeTrackerGlobalGazeVector
  - Client, [183](#)
- GetEyeTrackerGlobalPosition
  - Client, [181](#)
- GetForcePlateCount
  - Client, [165](#)
- GetForcePlateSubsamples
  - Client, [172](#)
- GetFrame
  - Client, [60](#)
- GetFrameNumber
  - Client, [61](#)
- GetFrameRate
  - Client, [63](#)
- GetFrameRateCount
  - Client, [71](#)
- GetFrameRateName
  - Client, [72](#)
- GetFrameRateValue
  - Client, [73](#)
- GetGlobalCentreOfPressure
  - Client, [170](#), [178](#)
- GetGlobalForceVector
  - Client, [166](#), [174](#)
- GetGlobalMomentVector
  - Client, [168](#), [176](#)
- GetGreyscaleBlob
  - Client, [207](#)
- GetGreyscaleBlobCount
  - Client, [206](#)
- GetHardwareFrameNumber
  - Client, [70](#)
- GetIsVideoCamera
  - Client, [198](#)
- GetLabeledMarkerCount
  - Client, [139](#)
- GetLabeledMarkerGlobalTranslation
  - Client, [140](#)
- GetLatencySampleCount
  - Client, [64](#)
- GetLatencySampleName
  - Client, [65](#)
- GetLatencySampleValue
  - Client, [67](#)
- GetLatencyTotal
  - Client, [69](#)
- GetMarkerCount
  - Client, [123](#)
- GetMarkerGlobalTranslation
  - Client, [130](#)
- GetMarkerName
  - Client, [125](#)
- GetMarkerParentName
  - Client, [128](#)
- GetMarkerRayContribution
  - Client, [134](#)
- GetMarkerRayContributionCount
  - Client, [132](#)
- GetObjectQuality
  - Client, [121](#)
- GetSegmentChildCount

## INDEX

- Client, [83](#)
- RetimingClient, [239](#)
- GetSegmentChildName
  - Client, [85](#)
  - RetimingClient, [241](#)
- GetSegmentCount
  - Client, [79](#)
  - RetimingClient, [235](#)
- GetSegmentGlobalRotationEulerXYZ
  - Client, [109](#)
  - RetimingClient, [263](#)
- GetSegmentGlobalRotationHelical
  - Client, [103](#)
  - RetimingClient, [257](#)
- GetSegmentGlobalRotationMatrix
  - Client, [105](#)
  - RetimingClient, [259](#)
- GetSegmentGlobalRotationQuaternion
  - Client, [107](#)
  - RetimingClient, [261](#)
- GetSegmentGlobalTranslation
  - Client, [101](#)
  - RetimingClient, [255](#)
- GetSegmentLocalRotationEulerXYZ
  - Client, [119](#)
  - RetimingClient, [273](#)
- GetSegmentLocalRotationHelical
  - Client, [113](#)
  - RetimingClient, [267](#)
- GetSegmentLocalRotationMatrix
  - Client, [115](#)
  - RetimingClient, [269](#)
- GetSegmentLocalRotationQuaternion
  - Client, [117](#)
  - RetimingClient, [271](#)
- GetSegmentLocalTranslation
  - Client, [111](#)
  - RetimingClient, [265](#)
- GetSegmentName
  - Client, [81](#)
  - RetimingClient, [237](#)
- GetSegmentParentName
  - Client, [87](#)
  - RetimingClient, [243](#)
- GetSegmentStaticRotationEulerXYZ
  - Client, [97](#)
  - RetimingClient, [253](#)
- GetSegmentStaticRotationHelical
  - Client, [91](#)
  - RetimingClient, [247](#)
- GetSegmentStaticRotationMatrix
  - Client, [93](#)
  - RetimingClient, [249](#)
- GetSegmentStaticRotationQuaternion
  - Client, [95](#)
  - RetimingClient, [251](#)
- GetSegmentStaticScale
  - Client, [99](#)
- GetSegmentStaticTranslation
  - Client, [89](#)
  - RetimingClient, [245](#)
- GetSubjectCount
  - Client, [74](#)
  - RetimingClient, [230](#)
- GetSubjectName
  - Client, [75](#)
  - RetimingClient, [231](#)
- GetSubjectRootSegmentName
  - Client, [77](#)
  - RetimingClient, [233](#)
- GetTimecode
  - Client, [62](#)
- GetUnlabeledMarkerCount
  - Client, [136](#)
- GetUnlabeledMarkerGlobalTranslation
  - Client, [137](#)
- GetVersion
  - Client, [15](#)
  - RetimingClient, [218](#)
- GetVideoFrame
  - Client, [208](#)
- IsCentroidDataEnabled
  - Client, [50](#)
- IsConnected
  - Client, [21](#)
  - RetimingClient, [222](#)
- IsDebugEnabled
  - Client, [53](#)
- IsDeviceDataEnabled
  - Client, [49](#)
- IsGreyscaleDataEnabled
  - Client, [51](#)
- IsLightweightSegmentDataEnabled
  - Client, [45](#)
  - RetimingClient, [225](#)
- IsMarkerDataEnabled
  - Client, [46](#)
- IsMarkerRayDataEnabled
  - Client, [48](#)
- IsSegmentDataEnabled
  - Client, [44](#)

## INDEX

- IsUnlabeledMarkerDataEnabled
  - Client, [47](#)
- IsVideoDataEnabled
  - Client, [52](#)
- MaximumPrediction
  - RetimingClient, [275](#)
- RetimingClient, [213](#)
  - ~RetimingClient, [218](#)
  - Connect, [220](#)
  - DisableLightweightSegmentData, [224](#)
  - Disconnect, [221](#)
  - EnableLightweightSegmentData, [223](#)
  - GetAxisMapping, [227](#)
  - GetSegmentChildCount, [239](#)
  - GetSegmentChildName, [241](#)
  - GetSegmentCount, [235](#)
  - GetSegmentGlobalRotationEulerXYZ, [263](#)
  - GetSegmentGlobalRotationHelical, [257](#)
  - GetSegmentGlobalRotationMatrix, [259](#)
  - GetSegmentGlobalRotationQuaternion, [261](#)
  - GetSegmentGlobalTranslation, [255](#)
  - GetSegmentLocalRotationEulerXYZ, [273](#)
  - GetSegmentLocalRotationHelical, [267](#)
  - GetSegmentLocalRotationMatrix, [269](#)
  - GetSegmentLocalRotationQuaternion, [271](#)
  - GetSegmentLocalTranslation, [265](#)
  - GetSegmentName, [237](#)
  - GetSegmentParentName, [243](#)
  - GetSegmentStaticRotationEulerXYZ, [253](#)
  - GetSegmentStaticRotationHelical, [247](#)
  - GetSegmentStaticRotationMatrix, [249](#)
  - GetSegmentStaticRotationQuaternion, [251](#)
  - GetSegmentStaticTranslation, [245](#)
  - GetSubjectCount, [230](#)
  - GetSubjectName, [231](#)
  - GetSubjectRootSegmentName, [233](#)
  - GetVersion, [218](#)
  - IsConnected, [222](#)
  - IsLightweightSegmentDataEnabled, [225](#)
  - MaximumPrediction, [275](#)
  - RetimingClient, [217](#)
  - RetimingClient, [217](#)
  - SetAxisMapping, [226](#)
  - SetMaximumPrediction, [275](#)
  - UpdateFrame, [228](#)
  - WaitForFrame, [229](#)
- Client, [58](#)
- RetimingClient, [226](#)
- SetBufferSize
  - Client, [54](#)
- SetCameraFilter
  - Client, [209](#)
- SetMaximumPrediction
  - RetimingClient, [275](#)
- SetStreamMode
  - Client, [55](#)
- StartTransmittingMulticast
  - Client, [22](#)
- StopTransmittingMulticast
  - Client, [23](#)
- UpdateFrame
  - RetimingClient, [228](#)
- WaitForFrame
  - RetimingClient, [229](#)
- SetApexDeviceFeedback
  - Client, [57](#)
- SetAxisMapping