

# Vicon DataStream SDK 1.2.0

## **Developers Manual**

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 allow users to connect to and request data from the Vicon DataStream. The following combinations of platforms and technologies are supported:

	Windows x86 (32-	Windows x64 (64-bit)	Linux x86 (32-	Linux x64 (64-
	bit)		bit)	bit)
C++	✓	✓	✓	✓
.NET	✓	✓		
MATLAB	✓	✓ *		

<sup>\* -</sup> requires Microsoft Professional compiler

#### Important Notes:

- Not all function calls contained within the SDK will return data when connected to certain Vicon Applications. For example, Vicon Blade does not support analog devices, and therefore will not output device information into the DataStream.
- The current DataStream format is supported by Vicon Nexus 1.4+, Vicon Blade 1.6+, and Tracker 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 co-ordinate 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 – DLLs on Windows and SOs on Linux. The shared libraries and supporting files are required to 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

## **Installing on Linux**

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



### Windows - C++

Your application should

- #include "Client.h"
- Link against "ViconDataStreamSDK\_CPP.lib"
- Redistribute:
  - o "ViconDataStreamSDK CPP.dll"
  - o "Microsoft.VC8.CRT" (x86) or "Microsoft.VC9.CRT" (x64).

#### Windows - .NET

Your application should

- Link against the assembly "ViconDataStreamSDK\_DotNET.dll".
- Redistribute:
  - o "ViconDataStreamSDK\_DotNET.dll"
  - "ViconDataStreamSDK CPP.dll"
  - "Microsoft.VC8.CRT" (x86) or "Microsoft.VC9.CRT" (x64).
- Have the .NET Framework 2.0 or later installed.

The managed code in this assembly requires the unmanaged code in the C++ SDK

### Windows - MATLAB

Your application M file should be in the same directory as

- "Client.m"
- "DeviceType.m"
- "Direction.m"
- "Result.m"
- "StreamMode.m"
- "TimecodeStandard.m"
- "Unit.m"
- "ViconDataStreamSDK\_MATLAB.dll"
- "ViconDataStreamSDK\_MATLAB.h"
- "Microsoft.VC80.CRT"

### Linux - C++

Your application should

- #include "Client.h"
- Link against "libViconDataStreamSDK CPP.so"
- Redistribute "libViconDataStreamSDK CPP.so"



### What's New in Version 1.2.0

- 1. Added C++ Linux x64 support
- 2. Added support for .NET under Windows x64
- 3. New function calls:
  - i. GetForcePlateCount
  - ii. GetGlobalForceVector
  - iii. GetGlobalMomentVector
  - iv. GetGlobalCentreOfPressure

## Requirements

- A compatible licensed version of Vicon Blade, Vicon Nexus, or Vicon Tracker must be present.
- LabVIEW will make use of 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 7 and 8.
- The SDK has not been designed to allow access from Simulink.
- The Linux SDK has been specifically verified on CentOS 5.5, Ubuntu 8.04, Ubuntu 9.04, Fedora 9, and Fedora 11. 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 useful for debugging purposes.

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

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



# **List of all SDK Functions**

## **Contents**

Developers Manual	
Installing on Windows	
Installing on Linux	
Windows – C++	2
Windows – .NET	2
Windows – MATLAB	
Linux – C++	
What's New in Version 1.2.0	
Requirements	
Function Result Return Values	
List of all SDK Functions	
Construction and Destruction	
Result	
GetVersion	
Connect	
ConnectToMulticast	
Disconnect	
IsConnected	
StartTransmittingMulticast	
StopTransmittingMulticast	
EnableSegmentData	
EnableMarkerData	
EnableUnlabeledMarkerData	
EnableDeviceData	
DisableSegmentData	21
DisableMarkerData	
DisableUnlabeledMarkerData	
DisableDeviceData	24
IsSegmentDataEnabled	
IsMarkerDataEnabled	
IsUnlabeledMarkerDataEnabled	
IsDeviceDataEnabled	
SetStreamMode	
SetAxisMapping	
GetAxisMapping	
GetFrame	
Get rune Get FrameNumber	
GetLatencyTotal	
GetLatencySampleCount	
GetLatencySampleCount GetLatencySampleName	
GetLatencySampleValue	
GetSubjectCount	
GetSubjectName	
GetSubjectRootSegmentName	
GetSegmentCount	
GetSegmentName	
GetSegmentParentName	54



GetSegmentChildCount	56
GetSegmentChildName	58
GetSegmentStaticTranslation	61
GetSegmentStaticRotationHelical	63
GetSegmentStaticRotationMatrix	65
GetSegmentStaticRotationQuaternion	67
GetSegmentStaticRotationEulerXYZ	69
GetSegmentGlobalTranslation	71
GetSegmentGlobalRotationHelical	
GetSegmentGlobalRotationMatrix	
GetSegmentGlobalRotationQuaternion	77
GetSegmentGlobalRotationEulerXYZ	
GetSegmentLocalTranslation	81
GetSegmentLocalRotationHelical	83
GetSegmentLocalRotationMatrix	85
GetSegmentLocalRotationQuaternion	87
GetSegmentLocalRotationEulerXYZ	89
GetMarkerCount	91
GetMarkerName	93
GetMarkerParentName	96
GetMarkerGlobalTranslation	98
GetUnlabeledMarkerCount	
GetUnlabeledMarkerGlobalTranslation	
GetDeviceCount	104
GetDeviceName	106
GetDeviceOutputCount	109
GetDeviceOutputName	111
GetDeviceOutputValue	113
GetForcePlateCount	
GetGlobalForceVector	117
GetGlobalMomentVector	119
GetGlobalCentreOfPressure	121
Appendix A – What's New	
What's New in Version 1.0	
What's New in Version 1.0.1	
What's New in Version 1.1.0	
What's New in Version 1.2.0	124

## **Construction and Destruction**

You can create many instances of the Vicon DataStream Client which can connect to multiple Vicon DataStream Servers.

DataStream S	ervers.			
C++	C++ 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			
	ViconDataStreamSDK::CPP::Client * pHeapClient =			
	new ViconDataStreamSDK::CPP::Client();			
	Output_SomeFunction Output = pHeapClient->SomeFunction(Input);			
	delete pHeapClient;			
MATLAB	The MATLAB SDK is object oriented, and needs to be explicitly loaded and			
	unloaded.			
	Client.LoadViconDataStreamSDK();			
	pHeapClient = Client();			
	Output = pHeapClient.SomeFunction( Input );			
	Client.UnloadViconDataStreamSDK();			
.NET	.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.			
	C#			
	ViconDataStreamSDK.DotNET.Client pHeapClient = new ViconDataStreamSDK.DotNET.Client();			
	Output_SomeFunction Output = pHeapClient.SomeFunction( InputParam );			
	Output_oomer unction output = prieapolient.oomerunction( inputralam ),			
	// Signal to the garbage collector that it can clean up			
	pHeapClient.Disconnect();			
	pHeapClient = null;			

The Result o	code indicates the success or failure	of a function.
	Unknown	The result is unknown. Treat it as a failure.
	NotImplemented	The function called has not been implemented in this version of the SDK.
	Success	The function call succeeded.
	InvalidHostName	The "HostName" parameter passed to Connect() is invalid.
	InvalidMulticastIP	The "MulticastIP" parameter was not in the range "224.0.0.0" – "239.255.255.255"
	ClientAlreadyConnected	Connect() was called whilst already connected to a DataStream.
	ClientConnectionFailed	Connect() could not establish a connection to the DataStream server.
	ServerAlreadyTransmittin gMultcast	StartTransmittingMulticast() was called when the current DataStream server was already transmitting multicast on behalf of this client.
	ServerNotTransmittingMu Iticast	StopTransmittingMulticast() was called when the current DataStream server was not transmitting multicasr on behalf of this client.
	NotConnected	You have called a function which requires a connection to the DataStream server, but do not have a connection.
	NoFrame	You have called a function which requires a frame to be fetched from the DataStream server, but do not have a frame.
	InvalidIndex	An index you have passed to a function is out of range.
	InvalidSubjectName	The Subject Name you passed to a function is invalid in this frame.
	InvalidSegmentName	The Segment Name you passed to a function is invalid in this frame.
	InvalidMarkerName	The Marker Name you passed to a function is invalid in this frame.
	InvalidDeviceName	The Device Name you passed to a function is invalid in this frame.
	InvalidDeviceOutputNam e	The Device Output Name you passed to a function is invalid in this frame.
	InvalidLatencySampleNa me	The Latency Sample Name you passed to a function is invalid in this frame.
	CoLinearAxes	The directions passed to SetAxisMapping() contain input which would cause two or more axis to lie along the same line, e.g. "Up" and "Down" are on the same line.
	LeftHandedAxes	The directions passed to SetAxisMapping() would result in a left handed co-ordinate system. This is not supported in the SDK.
C++	namespace ViconDataStreamSDI { namespace CPP	supported in the SDK.

```
namespace Result
                    enum Enum
                     Unknown,
                     NotImplemented.
                     Success,
                     InvalidHostName,
                     InvalidMulticastIP,
                     ClientAlreadyConnected,
                     ClientConnectionFailed,
                     Server Already Transmitting Multicast,\\
                     ServerNotTransmittingMulticast,
                     NotConnected,
                     NoFrame,
                     InvalidIndex,
                     InvalidSubjectName,
                     InvalidSegmentName,
                     InvalidMarkerName,
                     InvalidDeviceName,
                     InvalidDeviceOutputName,
                     InvalidLatencySampleName,
                     CoLinearAxes,
                     LeftHandedAxes
                   };
MATLAB
                  classdef Result
                    properties (Constant = true)
                     Unknown
                                             = 0;
                     NotImplemented
                                                = 1;
                     Success
                                            = 2;
                     InvalidHostName
                                               = 3;
                     InvalidMulticastIP
                                              = 4;
                     ClientAlreadyConnected
                                                  = 6;
                     ClientConnectionFailed
                                                 = 7;
                     ServerAlreadyTransmittingMulticast = 8;
                     ServerNotTransmittingMulticast = 9;
                     NotConnected
                                               = 10;
                     NoFrame
                                             = 11:
                     InvalidIndex
                                            = 12;
                     InvalidSubjectName
                                                = 13;
                     InvalidSegmentName
                                                 = 14;
                     InvalidMarkerName
                                                = 15;
                     InvalidDeviceName
                                                = 16;
                                                   = 17;
                     InvalidDeviceOutputName
                     InvalidLatencySampleName
                                                    = 18;
                     CoLinearAxes
                                               = 19;
                     LeftHandedAxes
                                                = 20;
                    end
                    properties
                     Value
                    end
```

```
methods
                     function obj = Result( value )
                      obj.Value = value;
                     end% Constructor
                    end% methods
                  end% classdef
.NET
                  namespace ViconDataStreamSDK
                  namespace DotNET
                  public enum class Result
                     Unknown,
                     NotImplemented,
                     Success,
                     InvalidHostName,
                     InvalidMulticastIP,
                     ClientAlreadyConnected,
                     ClientConnectionFailed,
                     ServerAlreadyTransmittingMulticast,
                     Server Not Transmitting Multicast,\\
                     NotConnected,
                     NoFrame,
                     InvalidIndex,
                     InvalidSubjectName,
                     InvalidSegmentName,
                     InvalidMarkerName,
                     InvalidDeviceName,
                     InvalidDeviceOutputName,
                     InvalidLatencySampleName,
                     CoLinearAxes,
                     LeftHandedAxes
                   };
                  } // End of namespace DotNET
                  } // End of namespace ViconDataStreamSDK
```

GetVers	sion			
Get the versi	ion of the Vicon Data	Stream SDK		
Input				
Output	Major	unsigned int	The major version number. When this number increases we break backwards compatibility with previous major versions.	
	Minor	unsigned int	The minor version number. When this number increases we have probably added new functionality to the SDK without breaking backwards compatibility with previous versions.	
	Point	unsigned int	The point version number. When this number increases, we have introduced a bug fix or performance enhancement without breaking backwards compatibility with previous versions.	
C++	ViconDataStrear	Лаjor; Лinor;		
MATLAB	MyClient = Clien	% [Output] = GetVersion()  MyClient = Client(); Output = MyClient.GetVersion();		
.NET	// class Output_C // { // public uint Ma // public uint Mi // public uint Po // }; //	GetVersion .jor; nor;		
		nSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET on Output = MyClient.GetVersion();	C.Client();	

Connec	t			
Establish a d	ledicated connection t	o a Vicon DataStream Serve	er	
See Also : C	onnectToMulticast, Di	sconnect, IsConnected		
Input	Host Name	string	The DNS identifiable name, or IP address of the PC hosting the DataStream server. The function defaults to connecting on port 801. You can specify an alternate port number after a colon. "localhost"  "MyViconPC:804" "10.0.0.2"	
Output	Result	Result	Result.Success Result.InvalidHostName Result.ClientAlreadyConnecte d Result.ClientConnectionFailed	
C++	// { // public: // Result::Enum F // }; // // Output_Connect ViconDataStreams	// class Output_Connect // { // public: // Result::Enum Result; // };		
MATLAB	% [Output] = Conr MyClient = Client(	% [Output] = Connect()  MyClient = Client(); Output = MyClient.Connect( 'locahost:801' );		
.NET	// class Output_Co // { // public Result R // }; // // Output_Connect ViconDataStreams	// class Output_Connect // { // public Result Result; // };		

## **ConnectToMulticast**

Connect to a Vicon DataStream Server's Multicast stream. The stream content is managed by a client who calls StartTransmittingMulticast().

I I a a C N I a a a a	_	
Host Name	string	The DNS identifiable name, or IP address of the PC hosting the DataStream server. The function defaults to connecting on port 801. You can specify an alternate port number after a colon. "localhost" "MyViconPC:804" "10.0.0.2"
Multicast IP	string	The IP Address that the Multicast will be read from. The address should be in the range "224.0.0.0" – "239.255.255.255"
Result	Result	Result.Success Result.InvalidHostName Result.InvalidMulticastIP Result.ClientAlreadyConnected Result.ClientConnectionFailed
// { // public: // Result::Enum Resul // }; // Output_ConnectToM // ConnectToMulticast // const S  ViconDataStreamSDK: Output_ConnectToMulticast	ulticast ( const String string & Multic :CPP::Client I ticast Output :	astIP ); MyClient; =
% [Output] = ConnectToMulticast()  MyClient = Client();		
Output = MyClient.ConnectToMulticast( 'locahost', '224.0.0.0');  // class Output_ConnectToMulticast  // {  // public Result Result;  // };  //  // Output_ConnectToMulticast ConnectToMulticast ( string HostName,  // string MulticastIP );  ViconDataStreamSDK.DotNET.Client MyClient =  new ViconDataStreamSDK.DotNET.Client();  Output_ConnectToMulticast Output =		
	Multicast IP  Result  // class Output_Connect // { // public: // Result::Enum Result // }; // Output_ConnectToM // ConnectToMulticast // const S  ViconDataStreamSDK: Output_ConnectToMulticast MyClient.ConnectToMulticast MyClient.ConnectToMulticast // class Output_ConnectToMulticast // conput = Client(); Output = MyClient.Con // class Output_ConnectToMulticast // public Result Result // }; // Output_ConnectToMulticast // Soutput_ConnectToMulticast // Conductor ConnectToMulticast // Conductor ConnectToMulticast // Conductor ConnectToMulticast //	Multicast IP string  Result Result  // class Output_ConnectToMulticast // { // public: // Result::Enum Result; // }; // // Output_ConnectToMulticast // const String & Multic  ViconDataStreamSDK::CPP::Client Output_ConnectToMulticast Output MyClient.ConnectToMulticast( "local % [Output] = ConnectToMulticast()  MyClient = Client(); Output = MyClient.ConnectToMulticast // { // public Result Result; // }; // Output_ConnectToMulticast ConnectToMulticast // { // public Result Result; // }; // Output_ConnectToMulticast ConnectToMulticast // { // public Result Result; // }; // Output_ConnectToMulticast ConnectToMulticast // { // public Result Result; // }; // Output_ConnectToMulticast ConnectToMulticast // { // public Result Result; // }; // Output_ConnectToMulticast ConnectToMulticast // ViconDataStreamSDK.DotNET.Client // results.

### Disconnect Disconnect from the Vicon DataStream Server. See Also: Connect, IsConnected Input Output Result Result Result.Success Result.NotConnected C++ // class Output\_Disconnect // { // public: // Result::Enum Result; // }; // Output\_Disconnect Disconnect(); ViconDataStreamSDK::CPP::Client MyClient; MyClient.Connect( "localhost" ); Output\_Disconnect Output = MyClient.Disconnect(); **MATLAB** % [Output] = Connect() MyClient = Client(); MyClient.Connect( "localhost" ); Output = MyClient.Disconnect(); .NET // public class Output\_Disconnect // { // public Result Result; // }; // Output\_Disconnect Disconnect() ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client(); MyClient.Connect( "localhost" );

Output\_Disconnect Output = MyClient.Disconnect();

### **IsConnected** Discover whether client is connected to the Vicon DataStream Server. See Also: Connect, Disconnect Input Output Connected boolean True if you are connected to the stream, otherwise false. C++// class Output\_IsConnected // { // public: // bool Connected; // }; // Output\_IsConnected IsConnected() const; ViconDataStreamSDK::CPP::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** % [Output] = IsConnected() MyClient = Client(); Output = MyClient.IsConnected() // Output.Connected == false MyClient.Connect( "localhost" ); Output = MyClient.IsConnected() // Output.Connected == true // (assuming localhost is serving) .NET // public class Output\_IsConnected // public bool Connected: // }; // Output\_IsConnected IsConnected(); 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)

## **StartTransmittingMulticast**

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

366 AI30 . CO	Tirrect, Corriect romanic	asi, Disconneci, Sic		
Input	ServerIP	string	The IP Address on the server that the	
			Multicast will be sent from. This is required	
			because the server PC might need to	
			transmit on a separate NIC.	
	MulticastIP	string	The IP Address that the Multicast will be	
			sent to. The address should be in the	
			range "224.0.0.0" – "239.255.255.255"	
Output	Result	Result	Result.Success	
			Result.NotConnected	
			Result.InvalidMulticastIP	
			Result.ServerAlreadyTransmittingMulticast	
C++	// class Output_Start	FransmittingMulticast	Trooding of Your Missay Transmitting Manager	
011	// class output_otart	Transmitting Matticast		
	// public:			
	// Result::Enum Res	ult.		
	// ResultEnum Res	uit,		
	// }, //			
	1 1	or teet or and April Character		
	// Output_StartTransi		0. O a mare dD	
	_	lulticast ( const String a		
	//	// const String & MulticastIP ) const;		
	ViconDataStreamSD	ViconDataStreamSDK::CPP::CPP::Client MyClient;		
	MyClient.Connect( "lo	ocalhost");		
	MyClient.StartTransn	nittingMulticast( "10.0.0	0.1", "224.0.0.0" );	
MATLAB	% [Output] = StartTransmittingMulticast ()			
	MyClient = Client();			
	MyClient.Connect( "lo	ocalhost"):		
		nittingMulticast( '10.0.0	).1', '224.0.0.0' ):	
.NET		_StartTransmittingMult		
	// {			
	// public Result Resu	lt·		
	// };	,		
	// Output_StartTransi	mittinaMulticast		
	. –	lulticast( string ServerI	P string MulticastIP ):	
	// Start Fransi intilityiv	idilioasi( stillig Serveri	i , suring mullicastii <i>)</i> ,	
	ViconDataStreamSD	K.DotNET.Client MyCli	ent =	
		w ViconDataStreamSD		
	MyClient.Connect( "lo		V.	
		nittingMulticast("10.0.0	.1", "224.0.0.0" );	
	, ,	5	, , , , , , , , , , , , , , , , , , , ,	

# 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.

 $See \ Also: Connect, \ Connect To Multicast, \ Disconnect, \ Start Transmitting Multicast$ 

1 1	,	Ι΄				
Input						
Output	Result	Result	Result.Success			
			Result.NotConnected			
			Result.ServerNotTransmittingMulticast			
C++	// class Output_StopTra	// class Output_StopTransmittingMulticast				
	//{					
	// public:					
	// Result::Enum Result;					
	//};					
	// Output_StopTransmit	tingMulticast				
	// StopTransmittingMul	-				
		,				
	ViconDataStreamSDK::	CPP::CPP::Client MyClie	ent:			
	MyClient.Connect( "loca	-	,			
	MyClient.StartTransmittingMulticast( "10.0.0.1", "224.0.0.0" );					
	// Do some stuff					
		MyClient.StopTransmittingMulticast();				
MATLAB	% [Output] = StopTransmittingMulticast ()					
	70 [Output] - Otop Hallomitalliginalacact ()					
	MyClient = Client();					
	. "	MyClient.Connect( "localhost" );				
		MyClient.StartTransmittingMulticast( '10.0.0.1', '224.0.0.0' );				
	% Do some stuff					
	MyClient.StopTransmittingMulticast();					
.NET		topTransmittingMulticast				
	// {					
	// public Result Result;					
	// };					
	// // Output_StopTransmittingMulticast					
	// StopTransmittingMulticast();					
	" Otop Handinkungmakloast(),					
	ViconDataStreamSDK.DotNET.Client MyClient =					
	new ViconDataStreamSDK.DotNET.Client();					
		new viconDataStreamSDK.DotNET.Client();  MyClient.Connect( "localhost" );				
	MyClient.StartTransmittingMulticast( "10.0.0.1", "224.0.0.0" );					
	// Do some stuff					
	MyClient.StopTransmittingMulticast();					
	iviyoneni.Stop Hansiilit	ii igiviullicast(),				

## **EnableSegmentData**

Enable kinematic segment data in the Vicon DataStream. You should call this function on startup, after connecting to the server, and before trying to read local or global segment data.

See Also: IsSegmetnDataEnabled, DisableSegmentData, EnableMarkerData, EnableUnlabelledMarkerData, EnableDeviceData, GetSegmentCount, GetSegmentName, GetSegmentGlobalTranslation, GetSegmentLocalTranslation, GetSegmentLocalRotationXXX

Input					
Output	Result	Result	Result.NotConnected		
			Result.Success		
C++	// class Output_Enab	// class Output_EnableSegmentData			
	// {				
	// public:				
	// Result::Enum Re	sult;			
	// };				
	//				
	// Output_EnableSeg	mentData EnableSegmentData	a();		
		K::CPP::Client MyClient;			
	MyClient.Connect( "lo	•			
MATLAB	Output_EnableSegmentData Output = MyClient.EnableSegmentData();				
IVIATEAD	% [Output] = EnableSegmentData()				
	MyClient = Client();				
	MyClient.Connect( "localhost" );				
	Output = MyClient.EnableSegmentData();				
.NET	// public class Output_EnableSegmentData				
	// {				
	// public Result Result;				
	// };				
	// Output_EnableSegmentData EnableSegmentData();				
	ViconDataStreamSDK.DotNET.Client MyClient =				
	new ViconDataStreamSDK.DotNET.Client();				
	MyClient.Connect( "localhost" );				
	Output_EnableSegmentData Output = MyClient.EnableSegmentData();				

### **EnableMarkerData**

Enable labeled reconstructed marker data in the Vicon DataStream. You should call this function on startup, after connecting to the server, and before trying to read labeled marker data.

See Also: IsMarkerDataEnabled, DisableMarkerData, EnableSegmentData, EnableUnlabelledMarkerData, EnableDeviceData, GetMarkerCount, GetMarkerName, GetMarkerGlobalTranslation

Input		dirit, Octiviarite interior, Octiviari			
Output	Result	Result	Result.NotConnected Result.Success		
C++	// class Output_l	// class Output_EnableMarkerData			
	// {				
	// public:				
	// Result::Enum Result;				
	// };				
	//				
	// Output_Enable	eMarkerData EnableMarkerData();			
	ViconDataStreamSDK::CPP::Client MyClient;				
	MyClient.Connect( "localhost" );				
MATLAB	Output_EnableMarkerData Output = MyClient.EnableMarkerData();				
IVIATLAD	% [Output] = EnableMarkerData()				
	MyClient = Client();				
	MyClient.Connect( "localhost" );				
	Output = MyClient.EnableMarkerData();				
.NET	// public class Output_EnableMarkerData				
	// public Result Result;				
	// };				
	// Output_EnableMarkerData EnableMarkerData();				
	ViconDataStreamSDK.DotNET.Client MyClient =				
	new ViconDataStreamSDK.DotNET.Client();				
	MyClient.Connect( "localhost" );				
	Output_EnableMarkerData Output = MyClient.EnableMarkerData();				

### **EnableUnlabeledMarkerData**

Enable unlabeled reconstructed marker data in the Vicon DataStream. You should 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, GetUnlabeledMarkerGlobalTranslation

Input	Pata, EnableBeviet		ourn, Gotornaboroarnamor Grobarrianolation		
Output	Result	Result	Result.NotConnected		
-			Result.Success		
C++	// class Output_E	// class Output_EnableUnlabeledMarkerData			
	// {				
	// public:				
	// Result::Enum	// Result::Enum Result;			
	// };				
	//				
	// Output_Enable	UnlabeledMarkerData EnableUnl	labeledMarkerData();		
	ViconDataStrean	ViconDataStreamSDK::CPP::Client MyClient;			
	MyClient.Connect( "localhost" );				
	Output_EnableUnlabeledMarkerData Output =				
		MyClient.EnableUnlabeledM	arkerData();		
MATLAB	% [Output] = EnableUnlabeledMarkerData()				
	MyClient = Client();				
	MyClient.Connect( "localhost" );				
	Output = MyClient.EnableUnlabeledMarkerData();				
.NET	// public class Output_EnableUnlabeledMarkerData				
	// {				
	// public Result Result;				
	// Output_EnableUnlabeledMarkerData EnableUnlabeledMarkerData();				
	ViconDataStreamSDK.DotNET.Client MyClient =				
		new ViconDataStreamSDK.DotNET.Client();			
	MyClient.Connec	MyClient.Connect( "localhost" );			
	Output_EnableUnlabeledMarkerData Output =				
	MyClient.EnableUnlabeledMarkerData();				

### **EnableDeviceData**

Enable ForcePlate, EMG, and other device data in the Vicon DataStream. You should 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

Input					
Output	Result	Result	Result.NotConnected		
			Result.Success		
C++	// class Output_Enable	DeviceData			
	// {	// {			
	// public:				
		// Result::Enum Result;			
	// };				
	//				
	// Output_EnableDevice	eData EnableDeviceData();			
	ViconDataStreamSDK:	·CDD··Cliant MyCliant·			
	MyClient.Connect( "loc	•			
	,	oata Output = MyClient.EnableDeviceData	a():		
MATLAB	% [Output] = EnableDe		S(),		
	// [culput] = LitableBovioeBata()				
	MyClient = Client();				
	MyClient.Connect( "loc	alhost" );			
	Output = MyClient.Ena	bleDeviceData();			
.NET	// public class Output_E	EnableDeviceData			
	// {				
	// public Result Result	•			
	// };	// <b>}</b> ;			
	H				
	// Output_EnableDeviceData EnableDeviceData();				
	ViconDataStreamSDK.DotNET.Client MyClient =				
	new ViconDataStreamSDK.DotNET.Client();				
	MyClient.Connect( "localhost" );  Output FoobleDoviceDate Output = MyClient FoobleDoviceDate();				
	Output_EnableDeviceData Output = MyClient.EnableDeviceData();				

## DisableSegmentData

Disable kinematic segment data in the Vicon DataStream.

See Also: IsSegmetnDataEnabled, EnableSegmentData, EnableMarkerData, EnableUnlabelledMarkerData, EnableDeviceData, GetSegmentCount, GetSegmentName, GetSegmentGlobalTranslation, GetSegmentLocalTranslation, GetSegmentLocalRotationXXX

Input					
Output	Result	Result	Result.NotConnected		
			Result.Success		
C++	// class Output_D	isableSegmentData			
	// {	// {			
	// public:				
	// Result::Enum	Result;			
	// };				
	//				
	// Output_Disable	SegmentData DisableSegmentD	ata();		
	ViconDataStreamSDK::CPP::Client MyClient;				
	MyClient.Connect( "localhost" );				
MATLAB		Output_DisableSegmentData Output = MyClient.DisableSegmentData();			
IVIATEAD	% [Output] = DisableSegmentData()				
	MyClient = Client();				
	MyClient.Connect( "localhost" );				
	Output = MyClient.DisableSegmentData();				
.NET	// public class Output_DisableSegmentData				
	// {				
	// public Result Result;				
	// };				
	// Output_DisableSegmentData DisableSegmentData();				
	ViconDataStreamSDK.DotNET.Client MyClient =				
	new ViconDataStreamSDK.DotNET.Client();				
	MyClient.Connect( "localhost" );				
	Output_DisableSegmentData Output = MyClient.DisableSegmentData();				

### DisableMarkerData

Disable labeled reconstructed marker data in the Vicon DataStream.

See Also: IsMarkerDataEnabled, EnableMarkerData, EnableSegmentData, EnableUnlabelledMarkerData, EnableDeviceData, GetMarkerCount, GetMarkerName, GetMarkerGlobalTranslation

Input				
Output	Result	Result	Result NotConnected	
C			Result.Success	
C++	// class Output_Disablel	MarkerData		
	// { // publica			
	// public:			
	// Result::Enum Resul	ι,		
		// }; //		
	**	rData DisableMarkerData();		
	// Output_bloableMarke	Tata Bisabisivarior Bata(),		
	ViconDataStreamSDK::	CPP::Client MvClient:		
	MyClient.Connect( "loca	•		
	Output_DisableMarkerD	oata Output = MyClient.DisableMarkerDa	ta();	
MATLAB	% [Output] = DisableMa	rkerData()		
	MyClient = Client();			
	MyClient.Connect( "loca	•		
	Output = MyClient.Disal			
.NET	// public class Output_D	isableMarkerData		
	// {			
	// public Result Result;			
	// };			
	// // Output DischleMedianDete DischleMedianDete()			
	// Output_DisableMarkerData DisableMarkerData();			
	ViconDataStreamSDK.I	OotNET.Client MyClient =		
	new ViconDataStreamSDK.DotNET.Client();			
	MyClient.Connect( "localhost" );			
	Output_DisableMarkerData Output = MyClient.DisableMarkerData();			

### DisableUnlabeledMarkerData

Disable unlabeled reconstructed marker data in the Vicon DataStream.

See Also: IsUnlabeledMarkerDataEnabled, EnableUnlabeledMarkerData, EnableSegmentData, EnableMarkerData, EnableDeviceData, GetUnlabeledMarkerCount, GetUnlabeledMarkerGlobalTranslation

Input	ribata, Eriabiebevice.	Data, GetOrilabelediviarker Co	ount, GetUnlabeledMarkerGlobalTranslation		
Output	Result	Result	Result.NotConnected Result.Success		
C++	// class Output_Di	sableUnlabeledMarkerData	•		
	// {	·			
	// public:	// public:			
	// Result::Enum	Result;			
	// };				
	//				
	// Output_Disable	UnlabeledMarkerData DisableUn	labeledMarkerData();		
	ViconDataStream	ViconDataStreamSDK::CPP::Client MyClient;			
	MyClient.Connect( "localhost" );				
	Output_DisableUnlabeledMarkerData Output =				
	MyClient.DisableUnlabeledMarkerData();				
MATLAB	% [Output] = DisableUnlabeledMarkerData()				
	MyClient = Client();				
	MyClient.Connect( "localhost" );				
	Output = MyClient.DisableUnlabeledMarkerData();				
.NET	// public class Output_DisableUnlabeledMarkerData				
	// {				
	// public Result Result;				
	// };				
	// Output_DisableUnlabeledMarkerData DisableUnlabeledMarkerData();				
	ViconDataStreamSDK.DotNET.Client MyClient =				
		new ViconDataStreamSDK.DotNET.Client();			
		MyClient.Connect( "localhost" );			
	Output_DisableUnlabeledMarkerData Output =				
		MyClient.DisableUnlabeledMarkerData();			

### DisableDeviceData

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

See Also: IsDeviceDataEnabled, EnableDeviceData, EnableSegmentData, EnableMarkerData, EnableUnlabeledMarkerData, GetDeviceCount, GetDeviceName, GetDeviceOutputCount, GetDeviceOutputName,GetDeviceOutputValue

Input Input

Input					
Output	Result	Result	Result.NotConnected		
			Result.Success		
C++	// class Output_DisableDeviceData				
	// {	{			
	// public:	// public:			
	// Result::Enum Res	ult;			
	// };				
	//				
	// Output_DisableDevi	ceData DisableDeviceData();			
	ViconDataStreamSDK::CPP::Client MyClient;				
	MyClient.Connect( "localhost" );				
	Output_DisableDeviceData Output = MyClient.DisableDeviceData();				
MATLAB	% [Output] = DisableDeviceData()				
	MyClient = Client();				
	MyClient.Connect( "localhost" );				
	Output = MyClient.DisableDeviceData();				
.NET	// public class Output_DisableDeviceData				
	// {				
	// public Result Result;				
	// };				
	//				
	// Output_DisableDeviceData DisableDeviceData();				
	ViconDataStreamSDK.DotNET.Client MyClient =				
	new ViconDataStreamSDK.DotNET.Client();				
	MyClient.Connect( "localhost" );				
	Output_DisableDeviceData Output = MyClient.DisableDeviceData();				

## IsSegmentDataEnabled

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

See Also: EnableSegmentData, DisableSegmentData, IsMarkerDataEnabled.

IsUnlabeledM	arkerDataEnabled, Isl	DeviceDataEnabled		
Input				
Output	Enabled	boolean	Whether the data is enabled.	
C++	// class Output_IsS	egmentDataEnabled		
	// {			
	// public:			
	// bool Enabled;			
	// };			
	//			
	// Output_IsSegme	ntDataEnabled IsSegmentData	Enabled() const;	
	ViconDataStreamS	DK::CPP::Client MyClient;		
	MyClient.Connect(	"localhost" );		
	Output_IsSegment	DataEnabled Output = MyClien	t.IsSegmentDataEnabled();	
		// Output.Enabled == fa	alse	
	MyClient.EnableSegmentData();			
	Output_IsSegment	DataEnabled Output = MyClien	t.IsSegmentDataEnabled();	
		// Output.Enabled == ti	ue	
MATLAB	% [Output] = IsSeg	mentDataEnabled()		
	MyClient = Client();			
	MyClient.Connect( "localhost" );			
	Output = MyClient.IsSegmentDataEnabled(); % Output.Enabled == false			
	MyClient.EnableSegmentData();			
	Output = MyClient.IsSegmentDataEnabled(); % Output.Enabled == true			
.NET	// public class Outp	ut_IsSegmentDataEnabled		
	// {			
	// public bool Enabled;			
	// };			
	//			
	// Output_IsSegmentDataEnabled IsSegmentDataEnabled();			
	ViconDataStreamSDK.DotNET.Client MyClient =			
	new ViconDataStreamSDK.DotNET.Client();			
	MyClient.Connect(			
	Output_IsSegmentDataEnabled Output = MyClient.IsSegmentDataEnabled();			
	// Output.Enabled == false			
	MyClient.EnableSegmentData();			
	Output_IsSegmentDataEnabled Output = MyClient.IsSegmentDataEnabled();			
	// Output.Enabled == true			

### IsMarkerDataEnabled

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

See Also : EnableMarkerData, DisableMarkerData, IsSegmentDataEnabled. IsUnlabeledMarkerDataEnabled, IsDeviceDataEnabled

Input	Habica				
Output	Enabled	boolean	Whether the data is enabled.		
C++	// class Output_IsMark	•			
	// {				
	// public:				
	// bool Enabled;				
	// };				
	//				
	// Output_IsMarkerDat	aEnabled IsMarkerDataEnabled() const;			
		::CPP::Client MyClient;			
	MyClient.Connect( "loo				
	Output_IsMarkerDatal	Enabled Output = MyClient.lsMarkerData	Enabled();		
	// Output.Enabled == false				
	MyClient.EnableMarkerData();				
	Output_IsMarkerDatai	Enabled Output = MyClient.IsMarkerData	Enabled();		
MATLAB	% [Output] = IsMarker	// Output.Enabled == true			
WINTER CO	70 [Output] = Isiviarker	DataEriabled()			
	MyClient = Client();				
	MyClient.Connect( "localhost" );				
	Output = MyClient.IsMarkerDataEnabled(); % Output.Enabled == false				
	MyClient.EnableMarkerData();				
	Output = MyClient.IsMarkerDataEnabled(); % Output.Enabled == true				
.NET	// public class Output_	IsMarkerDataEnabled			
	// {				
	// public bool Enabled;				
	// };				
	//				
	// Output_IsMarkerDataEnabled IsMarkerDataEnabled();				
	ViconDataStreamSDK.DotNET.Client MyClient =				
	new ViconDataStreamSDK.DotNET.Client();				
	MyClient.Connect( "loo				
	Output_IsMarkerDataEnabled Output = MyClient.IsMarkerDataEnabled();  // Output.Enabled == false				
	MyClient.EnableMarke	•			
	Output_IsMarkerDataEnabled Output = MyClient.IsMarkerDataEnabled();				
		// Output.Enabled == true			

### IsUnlabeledMarkerDataEnabled

Return whether unlabeled marker data is enabled in the DataStream.

 $See \ Also: Enable Unlabeled Marker Data, Disable Unlabeled Marker Data, \ Is Segment Data Enabled.$ 

IsMarkerDataEnabled, IsDeviceDataEnabled

IsMarkerData	Enabled, IsDeviceDa	taEnabled			
Input					
Output	Enabled	boolean	Whether the data is enabled.		
C++	// class Output_Isl	JnlabeledMarkerDataEnabled			
	// {				
	// public:				
	// bool Enabled;				
	// };				
	//				
	· ·	eledMarkerDataEnabled			
	// IsUnlai	beledMarkerDataEnabled() cons			
		SDK::CPP::Client MyClient;			
	MyClient.Connect	( "localhost" );			
	Output_IsUnlabele	edMarkerDataEnabled Output =			
	MyClient.IsUn	MyClient.IsUnlabeledMarkerDataEnabled(); // Output.Enabled == false			
	MyClient.EnableUnlabeledMarkerData();				
	Output_IsUnlabeledMarkerDataEnabled Output =				
MATLAD		labeledMarkerDataEnabled(); // 0	Output.Enabled == true		
MATLAB	% [Output] = IsUn	labeledMarkerDataEnabled()			
	MyClient = Client();				
	MyClient.Connect( "localhost" );				
	Output = MyClient	Output = MyClient.IsUnlabeledMarkerDataEnabled(); % Output.Enabled == false			
		MyClient.EnableUnlabeledMarkerData();			
	Output = MyClient.lsUnlabeledMarkerDataEnabled(); % Output.Enabled == true				
.NET	·	put_IsUnlabeledMarkerDataEnal	oled		
		// {			
	// public bool Enabled;				
	// <b>}</b> ;				
	//				
	// Output_lsUnlabeledMarkerDataEnabled IsUnlabeledMarkerDataEnabled();				
	ViconDataStreamSDK.DotNET.Client MyClient =				
		new ViconDataStreamSDK.DotN	IET.Client();		
	-	MyClient.Connect( "localhost" );			
	Output_IsUnlabeledMarkerDataEnabled Output =				
	•	IsMarkerDataEnabled(); // Outpu	t.Enabled == false		
	-	MyClient.EnableUnlabeledMarkerData();			
	Output_IsUnlabeledMarkerDataEnabled Output =				
	MyClient.IsUnlabeledMarkerDataEnabled(); // Output.Enabled == true				

### IsDeviceDataEnabled

Return whether ForcePlate, EMG, and other device data is enabled in the data stream.

See Also: EnableDeviceData, DisableDeviceData, IsSegmentDataEnabled. IsMarkerDataEnabled, IsIInlabeledMarkerDataEnabled

IsUnlabeledMar	kerDataEnabled	<del>_</del>			
Input					
Output	Enabled	boolean	Whether the data is enabled.		
C++	// class Output_IsDev	viceDataEnabled			
	// {				
	// public:				
	// bool Enabled;				
	// };				
	//				
	// Output_IsDeviceDa	ataEnabled IsDeviceDataEnabled() const;			
	ViconDataStreamSD	K::CPP::Client MyClient;			
	MyClient.Connect( "le				
	Output_IsDeviceData	aEnabled Output = MyClient.IsDeviceDatal	Enabled();		
		// Output.Enabled == false			
		MyClient.EnableDeviceData();			
	Output_IsDeviceDataEnabled Output = MyClient.IsDeviceDataEnabled();				
NAATI AD		// Output.Enabled == true			
MATLAB	% [Output] = IsDevic	eDataEnabled()			
	MacCliant Cliant()				
	MyClient = Client(); MyClient.Connect( "localhost" );				
	Output = MyClient.IsDeviceDataEnabled(); % Output.Enabled == false				
	Output = MyClient.isDeviceDataEnabled(); % Output.Enabled == false  MyClient.EnableDeviceData();				
	Output = MyClient.lsDeviceDataEnabled(); % Output.Enabled == true				
.NET	// public class Output_IsDeviceDataEnabled				
	// public class Output_isDeviceDataEnabled // {				
	// public bool Enabled;				
	// };	,			
	//				
	// Output_IsDeviceDataEnabled IsDeviceDataEnabled();				
	ViconDataStreamSDK.DotNET.Client MyClient =				
	new ViconDataStreamSDK.DotNET.Client();				
	MyClient.Connect( "le	ocalhost");			
	Output_IsDeviceDataEnabled Output = MyClient.IsDeviceDataEnabled();				
	// Output.Enabled == false				
	MyClient.EnableDeviceData();				
	Output_IsDeviceDataEnabled Output = MyClient.IsDeviceDataEnabled();				
		// Output.Enabled == true			

### **SetStreamMode**

There are three modes that the SDK can operate in. Each mode has a different impact on the Client, Server, and network resources used.

- 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 we can achieve. 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. Once all buffers have filled up then frames may be dropped at the Server and the performance of the Server may be affected.
- In "ClientPull" mode, the Client waits for a call to GetFrame(), and then request the latest frame of data from the Server. This increases latency, because we need to send a request over the network to the Server, the Server has to prepare the frame of data for the Client, and then we need to send the data back over the network. Network bandwidth is kept to a minimum, because the Server only sends what you need. We are very unlikely to fill up our buffers, and Server performance is unlikely to be affected. The GetFrame() method blocks the calling thread until the frame has been received.
- "ClientPullPreFetch" is an enhancement to "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 which we had cached in memory. GetFrame() does not need to block the calling thread. As with normal "ClientPull", buffers are unlikely to fill up, Server performance is unlikely to be affected. Latency is slightly reduced, but network traffic may increase if we 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, then try "ClientPullPreFetch" followed by "ServerPush".

See Also: GetFrame, GetLatencyTotal

00071100.00	our ramo, ColLatorioy roll	*·	<del>_</del>		
Input	Mode	StreamMode	StreamMode.ServerPush		
			StreamMode.ClientPull		
			StreamMode.ClientPullPreFetch		
Output	Result	Result	Result.Success		
-			Result.NotConnected		
C++	// class Output_SetStr	eamMode			
	// {				
	// public:				
	// Result::Enum Res	ult;			
	// };				
	//				
	// Output_SetStreamMode SetStreamMode( const StreamMode::Enum Mode );				
	ViconDataStreamSDK::CPP::Client MyClient;				
	MyClient.Connect( "localhost" );				
	MyClient.SetStreamM	MyClient.SetStreamMode( ViconDataStreamSDK::CPP::StreamMode::ServerPush );			
	MyClient.SetStreamM	MyClient.SetStreamMode( ViconDataStreamSDK::CPP::StreamMode::ClientPull );			
	MyClient.SetStreamM	MyClient.SetStreamMode( ViconDataStreamSDK::CPP::StreamMode::ClientPullPreFetch );			
MATLAB	% [Output] = SetStrea	mMode( Mode );			
	MyClient = Client();				
	MyClient.Connect( 'lo	MyClient.Connect( 'localhost' );			
	MyClient.SetStreamM	lode( StreamMode.Serve	rPush );		
	MyClient.SetStreamM	MyClient.SetStreamMode( StreamMode.ClientPull );			
	MyClient.SetStreamM	lode( StreamMode.Client	PullPreFetch);		
.NET	// class Output_SetStr	eamMode			

## **SetAxisMapping**

Remaps the 3D axis.

Vicon Data uses a right handed co-ordinate system, with +X forward, +Y left, and +Z up. Other systems use different co-ordinate systems. The SDK can transform its data into any valid right-handed co-ordinate system by re-mapping each axis.

Specify the direction of your X, Y, and Z axis relative to yourself as the observer. 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 : Ge	etAxisMapping				
Input	XAxis	Direction			
	YAxis	Direction			
	ZAxis	Direction			
Output	Result	Result	Result.Success Result.CoLinearAxes Result.LeftHandedAxes		
C++	// class Output_Se	etAxisMapping	1 222 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2		
	// {				
	// public:				
	// Result::Enum I	Result;			
	// };				
	//				
	// Output_SetAxis	Mapping SetAxisMapping( const D	Direction::Enum XAxis,		
	// const Direction::Enum YAxis,				
	// const Direction::Enum ZAxis )				
	Vices Data Streem SDI/u CDDu Client Mu Client				
	ViconDataStreamSDK::CPP::Client MyClient; MyClient.SetAxisMapping( ViconDataStreamSDK::CPP::Direction::Forward,				
	ViconDataStreamSDK::CPP::Direction::Left,				
	ViconDataStreamSDK::CPP::Direction::Leit, ViconDataStreamSDK::CPP::Direction::Up );				
MATLAB	% [Output] = SetAxisMapping( XAxis,				
	% YAxis,				
	% ZAxis )				
	MyClient = Client();				
	MyClient.SetAxisMapping( Direction.Forward,				
	Direction.Left,				
	Direction.Up );				
.NET	// public class Output_SetAxisMapping				
	// {				
	// public Result Result;				
	// );				
	H				
	// Output_SetAxisMapping SetAxisMapping( Direction XAxis,				
	// Direction YAxis,				
	// Direction ZAxis );				

	Mapping  nt Axis mapping.					
	etAxisMapping					
Input						
Output	XAxis	Direction				
	YAxis	Direction				
	ZAxis	Direction				
C++	// class Output_Ge	etAxisMapping				
	// {					
	// public:					
	// Direction::Enur	n XAxis;				
	// Direction::Enur	ı YAxis;				
	// Direction::Enur	n ZAxis;				
	// };					
	//					
	// Output_GetAxis	// Output_GetAxisMapping GetAxisMapping() const;				
	ViconDataStream	ViconDataStreamSDK::CPP::Client MyClient;				
	Output_GetAxisM	Output_GetAxisMapping Output = MyClient.GetAxisMapping();				
	// Output.XAxis =	// Output.XAxis == ViconDataStreamSDK::CPP::Direction::Forward				
	// Output.YAxis == ViconDataStreamSDK::CPP::Direction::Left					
	// Output.ZAxis == ViconDataStreamSDK::CPP::Direction::Up					
MATLAB	% [Output] = GetAxisMapping()					
	MyClient = Client();					
	Output = MyClient	Output = MyClient.GetAxisMapping();				
	% Output.XAxis == Direction.Forward					
	% Output.YAxis == Direction.Left					
	% Output.ZAxis == Direction.Up					
.NET	// public class Output_GetAxisMapping					
	// {					
	// public Direction XAxis;					
	// public Direction	// public Direction YAxis;				
	// public Direction ZAxis;					
	// };					
	//					
	// Output_GetAxisMapping GetAxisMapping();					
	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					

### GetFrame

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

See Also : Se	tStreamMode				
Input					
Output	Result	Result	Result.Success		
			Result.NotConnected		
C++	// class Output_GetFrame				
	// {				
	// public:				
	// Result::Enum Result;				
	// };				
	//				
	// Output_GetFrame GetFrame();				
	ViconDataStreamSDK::CPP::Client MyClient;				
	Output_GetFrame Output;				
	Output = MyClient.GetFrame(); // Output.Result == NotConnected				
	MyClient.Connect( "localhost" );				
	Output = MyClient.GetFrame(); // Output.Result == Success				
MATLAB	% [Output] = GetFrame()				
	MyClient = Client();				
	Output = MyClient.GetFrame(); // Output.Result == NotConnected				
	MyClient.Connect( "localhost" );				
	Output = MyClient.GetFrame(); // Output.Result == Success				
.NET	// public class Ou	tput_GetFrame			
	// {				
	// public Result Result;				
	// };				
	//				
	// Output_GetFrame GetFrame();				
	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				

GetFram	neNumber					
Return the nui	mber of the last frame re	trieved from the DataStream				
See Also : Ge	tFrame, GetTimecode					
Input						
Output	Result	Result	Result.Success Result.NotConnected Result.NoFrame			
	Frame Number	unsigned integer	The frame number			
C++	// class Output_GetFrameNumber // { // public: // Result::Enum Result; // unsigned int FrameNumber; // }; // //					
	// Output_GetFrameNo	umber GetFrameNumber() const	t;			
	ViconDataStreamSDK	::CPP::Client MyClient;				
	MyClient.Connect( "loo	calhost");				
	Output_GetFrameNun	nber Output;				
	Output = MyClient.GetFrameNumber(); // Output.Result == NoFrame					
		// Output.FrameNumber == 0				
	MyClient.GetFrame();					
	Output = MyClient.GetFrameNumber(); // Output.Result == Success					
	// Output.FrameNumber >= 1					
MATLAB	% [Output] = GetFrameNumber()					
	MyClient = 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	// class Output_GetFra	meNumber				
	// { // public Result Result; // public uint FrameNumber; // }; // Output_GetFrameNumber GetFrameNumber();					
	ViconDataStreamSDK.DotNET.Client MyClient =  new ViconDataStreamSDK.DotNET.Client();  MyClient.Connect( "localhost" );  Output: GotFrameNumber Output:					
	Output_GetFrameNumber Output;					
	Output = MyClient.GetFrameNumber(); // Output.Result == NoFrame					
	// Output.FrameNumber == 0					
	MyClient.GetFrame(); Output = MyClient CetFrameNumber(): // Output Begult Suggest					
	Output = MyClient.GetFrameNumber(); // Output.Result == Success					
	// Output.FrameNumber >= 1					

# GetLatencyTotal

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

GetLatencySa	ample Value				
Input					
Output	Result	Result	Result.Success Result.NotConnected Result.NoFrame		
	Total	double	The total latency in seconds made from summing the other latencies.		
C++	// class Output_GetLatencyTotal // { // public: // Result::Enum Result; // double Total; // }; // // Output_GetLatencyTotal GetLatencyTotal() const;  ViconDataStreamSDK::CPP::Client MyClient; MyClient.Connect( "localhost" ); MyClient.GetFrame();				
MATLAB	Output_GetLatencyTotal Output = MyClient.GetLatencyTotal();  % [Output] = GetLatencyTotal()  MyClient = Client(); MyClient.Connect( 'localhost' ); MyClient.GetFrame(); Output = MyClient.GetLatencyTotal();				
.NET	// class Output_GetLatencyTotal  // {  // public Result Result;  // public double Total;  // };  //  // Output_GetLatencyTotal GetLatencyTotal();  ViconDataStreamSDK.DotNET.Client MyClient =  new ViconDataStreamSDK.DotNET.Client();  MyClient.Connect( "localhost" );				
	MyClient.GetFrame(); Output_GetLatencyTotal Output = MyClient.GetLatencyTotal();				

#### **GetLatencySampleCount**

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

	Tarrie, Gerriniecou	e, deitatericy rotal, deitateric	y Sample value			
Input						
Output	Result	Result	Result.Success			
			Result.NotConnected			
			Result.NoFrame			
	Count	unsigned int	The number of samples			
			taken.			
C++	// class Output_Ge	etLatencySampleCount	<u> </u>			
	// {					
	// public:					
	// Result::Enum	Result:				
	// unsigned int C					
	// };	·				
	//					
	// Output_GetLate	ncySampleCount GetLatencySamp	leCount() const;			
	, –	" Super_Sociationsysamplessount Sociationsysamplessount() const,				
	ViconDataStreamSDK::CPP::Client MyClient;					
	MyClient.Connect( "localhost" );					
	MyClient.GetFrame();					
	Output_GetLatencySampleCount Output = MyClient.GetLatencySampleCount();					
MATLAB	% [Output] = GetLatencySampleCount()					
	MyClient = Client();					
	MyClient.Connect( 'localhost' );					
	MyClient.GetFrame();					
		.GetLatencySampleCount();				
.NET	// class Output_GetLatencySampleCount					
	//{ //					
	// public Result Result;					
	// public uint Count;					
	// };					
	// Output_GetLatencySampleCount GetLatencySampleCount();					
	ViconDataStreamSDK.DotNET.Client MyClient =					
	new ViconDataStreamSDK.DotNET.Client();					
	MyClient.Connect( "localhost" );					
	MyClient.GetFrame();					
	Output_GetLatencySampleCount Output = MyClient.GetLatencySampleCount();					

#### **GetLatencySampleName** Return the name of a latency sample. This value can be passed into GetLatencySampleValue(). See Also: GetFrame, GetTimecode, GetLatencyTotal, GetLatencySampleCount, GetLatencySampleValue LatencySampleIn Unsigned int The index of the name. Input dex Output Result Result Result.Success Result.NotConnected Result.NoFrame Result.InvalidIndex Name string The name of the latency sample. C++ A valid Latency Sample Index is between 0 and GetLatencySampleCount()-1 // class Output\_GetLatencySampleName // { // public: // Result::Enum Result; // String Name; // }; // Output\_GetLatencySampleName // GetLatencySampleName( const unsigned int LatencySampleIndex ) const; ViconDataStreamSDK::CPP::Client MyClient; MyClient.Connect( "localhost" ); MyClient.GetFrame(); Output\_GetLatencySampleName Output = MyClient.GetLatencySampleName( 0 ); // Output.Name == "Data Collected" **MATLAB** A valid Latency Sample Index is between 1 and GetLatencySampleCount() % [Output] = GetLatencySampleName() MyClient = Client(); MyClient.Connect( 'localhost'); MyClient.GetFrame(); Output = MyClient.GetLatencySampleName( 1 ); % Output.Name == 'Data Collected' .NET A valid Latency Sample Index is between 0 and GetLatencySampleCount()-1 // class Output\_GetLatencySampleName // { // public Result Result; // public string Name; // }; // Output\_GetLatencySampleName // GetLatencySampleName( uint LatencySampleIndex ); ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();

MyClient.Connect( "localhost" ); MyClient.GetFrame();
Output_GetLatencySampleName Output = MyClient.GetLatencySampleName( 0 ); // Output.Name == "Data Collected"

# **GetLatencySampleValue**

Return the duration of a named latency sample in seconds. This value can be passed into GetLatencySampleValue().

Input	LatencySampleN	string	The name of the latency sample.			
·	ame		, i			
Output	Result	Result	Result.Success			
•			Result.NotConnected			
			Result.NoFrame			
			Result.InvalidLatencySampleName			
	Value	double	The duration of the latency in seconds.			
C++	// class Output_GetLate	encySampleValue				
	// {					
	// public:					
	// Result::Enum Resu	lt;				
	// double Value;					
	// };					
	//					
	// Output_GetLatencyS	ampleValue				
			& LatencySampleName ) const;			
	" 23.23.37 Campio raido ( cons. camig a Latorio) campio raino ) const,					
	ViconDataStreamSDK::CPP::Client MyClient;					
	MyClient.Connect( "localhost" );					
	MyClient.GetFrame();					
	Output_GetLatencySar	Output_GetLatencySampleValue Output =				
	MyClient.GetLatencySampleValue( "Data Collected" );					
	// Output.Value == 0.1					
MATLAB	% [Output] = GetLatencySampleValue()					
	MyClient = Client();					
	MyClient.Connect( 'localhost' );					
	MyClient.GetFrame();					
	Output = MyClient.GetLatencySampleValue( 'Data Collected' );					
NET	% Output.Value == 0.1					
.NET	// class Output_GetLate	encySampleValue				
	// { // public Pocult Pocult:					
	// public Result Result;					
	// public double Value;					
	// }; //					
	// Custout Cott stancy Console Value					
	// Output_GetLatencySampleValue					
	-	// GetLatencySampleValue( string LatencySampleName );				
	ViconDataStreamSDK.DotNET.Client MyClient =					
	new ViconDataStreamSDK.DotNET.Client();					
	MyClient.Connect( "localhost" );					
	MyClient.GetFrame();					
	Output_GetLatencySampleName Output =					
	MyClient.GetLatencySampleValue( "Data Collected" );					
	// Output.Value == 0.1					

#### **GetTimecode**

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

Ho   Mii   See   Fra   Su   Fie   Sta   Sta	esult  ours inutes econds ames ubFrame eldFlag andard  ubFramesPerFra e serBits	Result  Unsigned integer Unsigned integer Unsigned integer Unsigned integer Unsigned integer Boolean TimecodeStandard	Result.Success Result.NotConnected Result.NoFrame  None PAL NTSC NTSCDrop Film
Ho   Mii   See   Fra   Su   Fie   Sta   Sta	ours inutes econds ames ubFrame eldFlag andard  ubFramesPerFra e	Unsigned integer Unsigned integer Unsigned integer Unsigned integer Unsigned integer Boolean TimecodeStandard	Result.NotConnected Result.NoFrame  None PAL NTSC NTSCDrop
Mii   Se   Fra   Su   Fie   Sta   Sta	inutes econds ames ubFrame eldFlag andard ubFramesPerFra	Unsigned integer Unsigned integer Unsigned integer Unsigned integer Boolean TimecodeStandard	PAL NTSC NTSCDrop
Se   Fra   Su   Fie   Sta   Sta	econds rames ubFrame eldFlag andard ubFramesPerFra e	Unsigned integer Unsigned integer Unsigned integer Boolean TimecodeStandard	PAL NTSC NTSCDrop
Fra   Su   Fie   Sta   Sta	ames ubFrame eldFlag andard ubFramesPerFra e	Unsigned integer Unsigned integer Boolean TimecodeStandard	PAL NTSC NTSCDrop
Su   Fie   Sta   Sta	ubFrame eldFlag andard ubFramesPerFra e	Unsigned integer Boolean TimecodeStandard	PAL NTSC NTSCDrop
Fie   Sta	eldFlag andard ubFramesPerFra e	Boolean TimecodeStandard	PAL NTSC NTSCDrop
Sta  Su me Us  C++	andard ubFramesPerFra e	TimecodeStandard	PAL NTSC NTSCDrop
Su me Us C++	ubFramesPerFra e		PAL NTSC NTSCDrop
C++	е	I Indiana al interne	1
C++	serBits	Unsigned integer	
// {     // p     // // // // // // // // // // // //		Unsigned integer	
My My Ou	Result::Enum Runsigned int Unsigned int Unsi	Result; purs; nutes; conds; ames; bFrame; Flag; num Standard; bFramesPerFrame; erBits;  GetTimecode() const;  CPP::Client MyClient;	
Myr Myr Myr	[Output] = GetTimecoo /Client = Client();	host" );	
.NET // c	/Client.Connect( "local /Client.GetFrame(); /tput = MyClient.GetTir	*	

```
// {
// public Result
                     Result;
// public uint
                    Hours;
// public uint
                    Minutes;
// public uint
                    Seconds;
// public uint
                    Frames;
// public uint
                    SubFrame;
// public bool
                    FieldFlag;
// public TimecodeStandard Standard;
// public uint
                    SubFramesPerFrame;
// public uint
                    UserBits;
// };
//
// Output_GetTimecode GetTimecode();
ViconDataStreamSDK.DotNET.Client MyClient =
                 new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.GetFrame();
Output_GetTimecode Output = MyClient.GetTimecode();
```

### **GetSubjectCount**

Return the number of subjects in the DataStream. This information can be used in conjunction with GetSubjectName

See Also : GetSubjectName

See Also : Ge	tSubjectName				
Input					
Output	Result	Result	Result.Success Result.NotConnected Result.NoFrame		
	Subject Count	unsigned integer	The number of subjects		
C++	// { // public: // Result::Enum Resul // unsigned int Subjec // }; // // Output_GetSubjectCo	// public: // Result::Enum Result; // unsigned int SubjectCount; // }; // // Output_GetSubjectCount GetSubjectCount() const;  ViconDataStreamSDK::CPP::Client MyClient;			
	Output = MyClient.GetS	Output_GetSubjectCount(); // Output.Result == NoFrame			
	// Ooutput.SubjectCount == 0 MyClient.GetFrame(); Output = MyClient.GetSubjectCount(); // Output.Result == Success				
MATLAB	// Output.SubjectCount >= 0 % [Output] = GetSubjectCount()				
	MyClient.GetFrame();	alhost'); SubjectCount(); % Output.Result == Noi % Ooutput.SubjectCount == 0 SubjectCount(); % Output.Result == Suc % Output.SubjectCount >= 0			
.NET	// // Output_GetSubjectCo	; Count;  Dunt GetSubjectCount();  Dunt GetSubjectCount()			
	ViconDataStreamSDK.DotNET.Client MyClient =				

// Ooutput.SubjectCount == 0
MyClient.GetFrame();
Output = MyClient.GetSubjectCount(); // Output.Result == Success
// Output.SubjectCount >= 0

GetSubj	ectName					
	-	an be passed into segment and	I marker functions.			
	tSubjectCount		1=			
Input						
Output	Result	Result	Result.Success Result.NotConnected Result.NoFrame Result.InvalidIndex			
	Subject Name	etring	The name of the subject			
C++		string Index is between 0 and GetS				
	// class Output_Get // { // public: // Result::Enum Re // String Subject // }; // // Output_GetSubject // cor  ViconDataStreamSl MyClient.Connect( ' MyClient.GetFrame  Output_GetSubject	SubjectName esult; etName; etName GetSubjectName( est unsigned int SubjectIndex ) const DK::CPP::Client MyClient; clocalhost"); ();	t; GC.Result == Success			
	OutputGSN = MyCl OutputGSN = MyCl	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 == ""				
MATLAB	A valid Subject	Index is between 1 and GetS				
		% [Output] = GetSubjectName( SubjectIndex )				
	,	MyClient = Client; MyClient.Connect( 'localhost' ); MyClient.GetFrame();				
	OutputGSC = MyCI	OutputGSC = MyClient.GetSubjectCount(); % OutputGSC.Result == Success % OutputGSC.SubjectCount == 2				
		OutputGSN = MyClient.GetSubjectName(1); % OutputGSN.Result == Success % OutputGSN.SubjectName == 'Al'				
	OutputGSN = MyCl	OutputGSN = MyClient.GetSubjectName(2); % OutputGSN.Result == Success  % OutputGSN .SubjectName == 'Bob'				

	OutputGSN = MyClient.GetSubjectName(3); % OutputGSN.Result == InvalidIndex					
	// OutputGSN.SubjectName == "					
.NET	A valid Subject Index is between 0 and GetSubjectCount()-1					
	// public class Output_GetSubjectName					
	// public Result Result;					
	// public string SubjectName;					
	//					
	// Output_GetSubjectName GetSubjectName( uint SubjectIndex );					
	ViconDataStreamSDK.DotNET.Client MyClient =					
	new ViconDataStreamSDK.DotNET.Client();					
	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 == "Al"					
	OutputGSN = MyClient.GetSubjectName(1);// OutputGSN.Result == Success					
	// OutputGSN .SubjectName == "Bob"					
	OutputGSN = MyClient.GetSubjectName(2);// OutputGSN.Result == InvalidIndex					
	// OutputGSN.SubjectName == ""					

### **GetSubjectRootSegmentName**

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.

 $See \ Also: Get Segment Count, \ Get Segment Parent Name, \ Get Segment Child Name$ 

Input	Subject Name	string	The name of the subject			
Output	Result	Result	Result.Success			
Output	Nesuit	Nesuit	Result.NotConnected			
			Result NoFrame			
	0	-100	Result.InvalidSubjectName			
	Segment Name	string	The name of the root			
0			segment			
C++	· ·	jectRootSegmentName				
	// {					
	// public:					
	// Result::Enum Resul	t;				
	// String Segment	Name;				
	// };					
	//					
	// Output_GetSubjectR	ootSegmentName GetSubjectR	tootSegmentName(			
	// const S	ring & SubjectName ) const				
	ViconDataStreamSDK:	ViconDataStreamSDK::CPP::Client MyClient;				
	MyClient.Connect( "loc	MyClient.Connect( "localhost" );				
	MyClient.EnableSegmentData();					
	MyClient.GetFrame();					
	Output_GetSubjectRootSegmentName Output;					
	Output = MyClient.GetSubjectRootSegmentName( "Bob" );					
	// Output.Result == Success					
		out.SegmentName == "Pelvis"				
MATLAB	% [Output] = GetSubjectRootSegmentName( SubjectName )					
		,	,			
	MyClient = Client();					
	MyClient.Connect( "localhost" );					
		MyClient.EnableSegmentData();				
		MyClient.GetFrame();				
	MyGlicht.Gett fame(),					
	Output = MyClient.GetSubjectRootSegmentName( "Bob" );					
	% Output.Result == Success					
		put.SegmentName == "Pelvis"				
.NET						
	// public class Output_GetSubjectRootSegmentName					
		// { // public Result Result;				
	// public Result; // public string SegmentName;					
	// }; //					
		// Catanat Carl Carl Carl Carl Carl Carl Carl Carl				
		// Output_GetSubjectRootSegmentName GetSubjectRootSegmentName(				
	"	// string SubjectName );				
	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"

# **GetSegmentCount**

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

See Also : Ge	tSubjectName, GetSegm	entiname				
Input	Subject Name	string	The name of the subject			
Output	Result	Result	Result.Success			
•			Result.NotConnected			
			Result.NoFrame			
			Result.InvalidSubjectName			
	Segment Count	unsigned integer	The number of segments			
C++	// class Output_GetSeg		The manner of deginerite			
011		gmentcount				
	// { // public:	// { // code				
	· ·	14.				
	// Result::Enum Resu					
	// unsigned int Segme	entCount;				
	// }; //					
	//					
		Count GetSegmentCount(				
	// const St	ring & SubjectName ) const;				
	ViconDataStreamSDK	::CPP::Client MyClient;				
	MyClient.EnableSegm	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	% [Output] = GetSegm	entCount( SubjectName )				
I	MyClient = Client();					
	MyClient.EnableSegmentData();					
	MyClient.Connect( "localhost" );					
	injunition ),					
	Output = MyClient.GetSegmentCount( "Bob" ); % Output.Result == NoFrame					
	% Output.SegmentCount == 0					
	MyClient.GetFrame();					
	Output = Maclicat Cat	Output - Ma Olient CatCommontCount (AP) v (/ Output Double				
	Output - Mychent.Get	Output = MyClient.GetSegmentCount( "Al" ); % Output.Result ==				
		% InvalidSubjectName				
		% Output.SegmentCount == 0				
	Output = MyClient.GetSegmentCount( "Bob" ); % Output.Result == Success					
	Output - Myonent.Get	% Output.SegmentCount >= 0				
		/o Output.SegmentCount >= 0	1			

```
.NET
                  // public class Output_GetSegmentCount
                  // {
                  // public Result Result;
                  // public uint SegmentCount;
                  // };
                  // Output_GetSegmentCount GetSegmentCount( string SubjectName );
                  ViconDataStreamSDK.DotNET.Client MyClient =
                                   new ViconDataStreamSDK.DotNET.Client();
                  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
```

GetSegmentName				
Return the name See Also : GetSe	-	ecified subject. This can be passed in	nto segment functions.	
Input	Subject Name	string	The name of the subject	
Прис	Segment Index	unsigned integer	The index of the segment.	
Output	Result	Result	Result.Success	
Output	resuit	result	Result.NotConnected Result.NoFrame Result.InvalidSubjectName Result.InvalidIndex	
	Segment Name	string	The name of the segment	
C++	A valid Segment In  // class Output_GetSegn // { // public: // Result::Enum Result // String SegmentN // };  // Output_GetSegmentN // const Str // const un  ViconDataStreamSDK:: MyClient.Connect( "loca MyClient.EnableSegme MyClient.GetFrame();  Output_GetSegmentCo OutputGSC = MyClient. // Outp // Outp Output_GetSegmentNa OutputGSN = MyClient. // Outp // Outp // Outp // Outp	dex is between 0 and GetSegmentName  ; lame GetSegmentName( ring & SubjectName, signed int SegmentIndex) const  CPP::Client MyClient; alhost"); ntData();  unt OutputGSC; GetSegmentCount("Bob"); utGSC.Result == Success utGSC.SegmentCount == 2		
	// Outp OutputGSN = MyClient. // Outp	utGSN.Result == Success utGSN.SegmentName == "Head" GetSegmentName( "Bob", 1 ); utGSN.Result == Success utGSN.SegmentName == "Pelvis"		
	// Outp // Outp	GetSegmentName( "Bob", 2 ); utGSN.Result == InvalidIndex utGSN.SegmentName == "" hird segment)		
MATLAB		dex is between 1 and GetSegme	entCount()	

```
% [Output] = GetSegmentName( SubjectName, SegmentIndex )
                  MyClient = Client();
                  MyClient.Connect( "localhost" );
                  MyClient.EnableSegmentData();
                  MyClient.GetFrame();
                  OutputGSC = MyClient.GetSegmentCount( "Bob" );
                                % OutputGSC.Result == Success
                                % OutputGSC.SegmentCount == 2
                  OutputGSN = MyClient.GetSegmentName( "Alice", 1 );
                                % OutputGSN.Result == InvalidSubjectName
                                 % OutputGSN.SegmentName == ""
                  OutputGSN = MyClient.GetSegmentName( "Bob", 1 );
                                % OutputGSN.Result == Success
                                 % OutputGSN.SegmentName == "Head"
                  OutputGSN = MyClient.GetSegmentName( "Bob", 2 );
                                % OutputGSN.Result == Success
                                % OutputGSN.SegmentName == "Pelvis"
                  OutputGSN = MyClient.GetSegmentName( "Bob", 3 );
                                % OutputGSN.Result == InvalidIndex
                                % OutputGSN.SegmentName == ""
                                % (no third segment)
.NET
                  A valid Segment Index is between 0 and GetSegmentCount()-1
                 // public class Output_GetSegmentName
                 // {
                 // public Result Result;
                 // public string SegmentName;
                 // };
                  // Output_GetSegmentName GetSegmentName( string SubjectName,
                                        uint SegmentIndex);
                  ViconDataStreamSDK.DotNET.Client MyClient =
                                  new ViconDataStreamSDK.DotNET.Client();
                  MyClient.Connect( "localhost" );
                  MyClient.EnableSegmentData();
                  MyClient.GetFrame();
                  Output_GetSegmentCount OutputGSC;
                  OutputGSC = MyClient.GetSegmentCount("Bob");
                                // OutputGSC.Result == Success
                                // OutputGSC.SegmentCount == 2
                  Output_GetSegmentName OutputGSN;
                  OutputGSN = MyClient.GetSegmentName( "Alice", 0 );
                                // OutputGSN.Result == InvalidSubjectName
                                // OutputGSN.SegmentName == ""
                  OutputGSN = MyClient.GetSegmentName( "Bob", 0 );
```

// OutputGSN.Result == Success // OutputGSN.SegmentName == "Head"
OutputGSN = MyClient.GetSegmentName( "Bob", 1 );
// OutputGSN.Result == Success
// OutputGSN.SegmentName == "Pelvis"
OutputGSN = MyClient.GetSegmentName( "Bob", 2 );
// OutputGSN.Result == InvalidIndex
// OutputGSN.SegmentName == ""
// (no third segment)

# **GetSegmentParentName**

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.

 $See \ \textit{Also}: Get Segment Count, \ \ \textit{Get Segment Child Count}, \ \ \textit{Get Segment Child Name},$ 

GetSubjectRootSegmentName

	ootSegmentName	Τ .	Ι
Input	Subject Name	string	The name of the subject
	Segment Name	string	The name of the segment
Output	Result	Result	Result.Success
			Result.NotConnected
			Result.NoFrame
			Result.InvalidSubjectName
			Result.InvalidSegmentName
	Segment Name	string	The name of the parent
			segment or an empty string if
			it is the root segment.
C++	// class Output_GetSeg	mentParentName	
	// {		
	// public:		
	// Result::Enum Result	··	
	// String Segment		
	// String Segment	vairie,	
	// }, //		
		Paranthama CatCagmantDaranthama/	
		ParentName GetSegmentParentName(	
		ring & SubjectName,	
	// const St	ring & SegmentName ) const	
	ViconDataStreamSDK::	CPP::Client MyClient;	
	MyClient.Connect( "loca	-	
	MyClient.EnableSegme		
	MyClient.GetFrame();	· ·	
	Output_GetSegmentPa	rentName Output:	
	· · · · · · · · · · · · · · · · · · ·	SegmentParentName( "Bob", "Pelvis" );	
		out.Result == Success	
		out.SegmentName == ""	
	· ·	is the root segment	
		SegmentParentName( "Bob", "LFemur" );	
		out.Result == Success	
	·	out.SegmentName == "Pelvis"	
MATLAB		entParentName( SubjectName, Segment	Name )
	MyClient = Client();		
	MyClient.Connect( "loca	alhost");	
	MyClient.EnableSegme	entData();	
	MyClient.GetFrame();		
	Output = MyClient.GetS	SegmentParentName( "Bob", "Pelvis" );	
		put.Result == Success	
		_	
		S .	
	Output = MyClient.GetS % Out % Out % This	-	

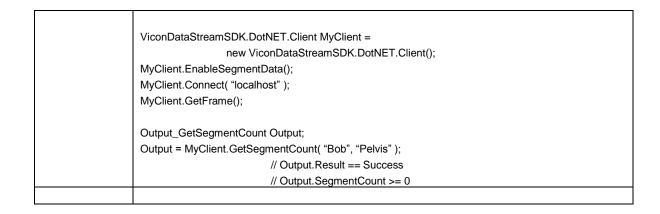
	% Output.Result == Success
	% Output.SegmentCount == "Pelvis"
.NET	// public class Output_GetSegmentParentName
	// public Result Result;
	// public string SegmentName;
	// };
	//
	// Output_GetSegmentParentName GetSegmentParentName(
	// string SubjectName,
	// string SegmentName );
	ViconDataStreamSDK.DotNET.Client MyClient =
	new ViconDataStreamSDK.DotNET.Client();
	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"

# **GetSegmentChildCount**

Return the number of descendant segments for a specified subject segment in the DataStream. This information can be used in conjunction with GetSegmentChildName.

See Also: GetSegmentChildName, GetSegmentParentName

_	10 1: AN	T .	T. (4) 1.1 (		
Input	Subject Name	string	The name of the subject		
	Segment Name	string	The name of the segment		
Output	Result	Result	Result.Success		
			Result.NotConnected		
			Result.NoFrame		
			Result.InvalidSubjectName		
			Result.InvalidSegmentName		
	Segment Count	unsigned integer	The number of segments		
C++	// class Output_GetSeg				
011	// {	gmentermacount			
	// public:				
	· ·	١4.			
	// Result::Enum Resu	•			
	// unsigned int Segme	entCount;			
	// }; //				
	//				
		ChildCount GetSegmentChildCount	t(		
		ring & SubjectName,			
	// const St	// const String & SegmentName ) const;			
	ViconDataStreamSDK	:CPP::Client MyClient;			
	MyClient.EnableSegmentData();				
	MyClient.Connect( "localhost" );				
	MyClient.GetFrame();				
	Output_GetSegmentC	nildCount Output;			
	Output = MyClient.Get	SegmentCount( "Bob", "Pelvis" );			
		// Output.Result == Success			
		// Output.SegmentCount >= 0			
MATLAB	% [Output] = GetSegm	entChildCount( SubjectName, Segr	mentName)		
	MyClient = Client();				
	MyClient.EnableSegm	entData();			
	MyClient.Connect( "loc	alhost");			
	MyClient.GetFrame();	,			
	Output = MvClient.Get	SegmentChildCount( "Bob", "Pelvis"	'):		
	, , , , , , , , , , , , , , , , , , , ,	% Output.Result == Success	,,		
		% Output.SegmentCount >= 0			
.NET	// public class Output	GetSegmentChildCount			
	// {				
	// public Result Resul	<del>.</del>			
	// public result result // public uint Segme				
		noount,			
	// }; //				
		ChildCount CotCoomontChildCount	t/ string SubjectName		
		ChildCount GetSegmentChildCount	it stillig Subjectivame,		
	//	string SegmentName );			



# **GetSegmentChildName**

Return the name of a child segment for a specified subject segment. This can be passed into segment functions.

See Also : G	etSegmentCount					
Input	Subject Name	string	The name of the subject			
	Segment Name	string	The name of the parent			
			segment.			
	Segment Index	unsigned integer	The index of the child			
			segment.			
Output	Result	Result	Result.Success			
			Result.NotConnected			
			Result.NoFrame			
			Result.InvalidSubjectName			
			Result.InvalidSegmentName			
			Result.InvalidIndex			
	Segment Name	string	The name of the child			
			segment			
C++	A valid Segment I	ndex is between 0 and Get	SegmentChildCount()-1			
	// class Output_GetSeg	gmentChildName				
	// {					
	// public:					
	// Result::Enum Resu	lt;				
	// String Segment	Name;				
	// };	// };				
	//					
	// Output_GetSegmentChildName GetSegmentName(					
	// const S	•				
	// const S					
	// const u	nsigned int SegmentIndex) cons	st			
	ViconDataStreamSDK	::CPP::Client MyClient;				
	MyClient.Connect( "loc	-				
	MyClient.EnableSegm					
	MyClient.GetFrame();	eritData(),				
	wiyolient.Geti fame(),					
	Output_GetSegmentC	hildCount OutputGSCC;				
	OutputGSCC = MyClie	ent.GetSegmentChildCount( "Bob"	", "Pelvis" );			
	// Out	putGSCC.Result == Success				
	// Out	putGSCC.SegmentCount == 2				
	Output_GetSegmentC	hildName OutputGSCN;				
	OutputGSCN = MyClie	ent.GetSegmentName( "Alice", 0 )	;			
	// Out	putGSN.Result == InvalidSubject	Name			
	// Out	putGSN.SegmentName == ""				
	OutputGSCN = MyClie	ent.GetSegmentName( "Bob", "Pe	elvis", 0 );			
		putGSCN.Result == Success	**			
		putGSCN.SegmentName == "LFe	emur"			
		nt.GetSegmentName( "Bob", "Pe				
	· · ·	putGSCN.Result == Success	•			
		putGSCN.SegmentName == "RF	emur"			

```
OutputGSCN = MyClient.GetSegmentName( "Bob", "Pelvis", 2);
                                 // OutputGSCN.Result == InvalidIndex
                                 // OutputGSCN.SegmentName == ""
                                 // (no third segment)
MATLAB
                  A valid Segment Index is between 1 and GetSegmentChildCount()
                  % [Output] = GetSegmentChildName( SubjectName, SegmentName, SegmentIndex )
                  MyClient = Client();
                  MyClient.Connect( "localhost" );
                  MyClient.EnableSegmentData();
                  MyClient.GetFrame();
                  OutputGSCC = MyClient.GetSegmentChildCount( "Bob", "Pelvis" );
                                 % OutputGSCC.Result == Success
                                 % OutputGSCC.SegmentCount == 2
                  OutputGSCN = MyClient.GetSegmentChildName( "Alice", "Pelvis", 1 );
                                 % OutputGSCN.Result == InvalidSubjectName
                                 % OutputGSCN.SegmentName == ""
                  OutputGSCN = MyClient.GetSegmentChildName( "Bob", "Pelvis", 1);
                                 % OutputGSCN.Result == Success
                                 % OutputGSCN.SegmentName == "LFemur"
                  OutputGSCN = MyClient.GetSegmentChildName( "Bob", "Pelvis", 2);
                                 % OutputGSCN.Result == Success
                                 % OutputGSCN.SegmentName == "RFemur"
                  OutputGSCN = MyClient.GetSegmentChildName("Bob", "Pelvis", 3);
                                 % OutputGSCN.Result == InvalidIndex
                                 % OutputGSCN.SegmentName == ""
                                 % (no third segment)
.NET
                  A valid Segment Index is between 0 and GetSegmentChildCount()-1
                  // public class Output_GetSegmentChildName
                  // {
                  // public Result Result;
                  // public string SegmentName;
                  // };
                  //
                  // Output_GetSegmentChildName GetSegmentChildName( string SubjectName,
                                              string SegmentName,
                  //
                                              uint SegmentIndex);
                  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.Result == Success
// OutputGSCN.SegmentName == "RFemur"

OutputGSCN = MyClient.GetSegmentChildName( "Bob", "Pelvis", 2 );

// OutputGSCN.Result == InvalidIndex
// OutputGSCN.SegmentName == ""
// (no third segment)
```

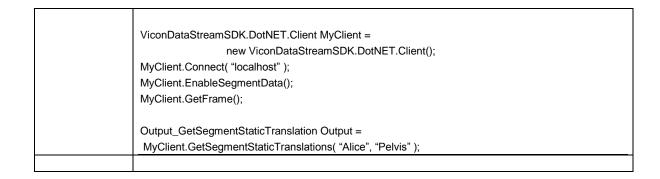
### **GetSegmentStaticTranslation**

Return the static pose translation of a subject segment.

 $See \ Also: Get Segment Static Rotation Helical, \ Get Segment Static Rotation Matrix,$ 

 $Get Segment Static Rotation Quaternion, \ Get Segment Static Rotation Euler XYZ, \ Get Segment Local Rotation, Get Segment Local Rotation Matrix, \ Get Segment Local Rotation Quaternion, \ Get Segment Local Rotation \ Get Segment Local Rotation \ Get Segment Local Rotation \ Get Segment \ Get S$ 

Input	Subject Name	string	The name of the subject	
прис		-	-	
0 1 1	Segment Name	string	The name of the segment.	
Output	Result	Result	Result.Success	
			Result.NotConnected	
			Result.NoFrame	
			Result.InvalidSubjectName	
			Result.InvalidSegmentName	
	Translation	double[3]	The translation of the	
			segment	
C++	// class Output_GetSeg	mentStaticTranslation		
	// {			
	// public:			
	// Result::Enum Resul	t:		
	// double Translati			
	// };			
	//			
		StaticTranslation GetSegmentStaticTrar	nslation(	
	// const String & St	_	iolation (	
	_	egmentName ) const;		
	// Const String & St	sgmentivame / const,		
	ViconDataStreamSDK:	ViconDataStreamSDK::CPP::Client MyClient;		
	MyClient.Connect( "localhost" );			
	MyClient.EnableSegme	entData();		
	MyClient.GetFrame();			
	0.10.10.10.10.10.10.10.10.10.10.10.10.10	atta Tanan alatta a Oudand		
		aticTranslation Output =		
MATLAB		StaticTranslation( "Alice", "Pelvis" );	manthlama)	
IVIATEAD	% [Output] = GetSegm	entStaticTranslation( SubjectName, Seg	gmenuvame)	
	MyClient = Client();			
	MyClient.Connect( "loc	alhost" ):		
	MyClient.EnableSegme			
	MyClient.GetFrame();	moduly,		
	wyoneni.Geti fame(),			
	Output = MvClient Get	SegmentStaticTranslation( "Alice", "Pelv	is"):	
.NET		GetSegmentStaticTranslation		
	// {	<u> </u>		
	// public Result Resu	lt;		
	// public double[] Tran			
	// };			
		StaticTranslation GetSegmentStaticTrar	nslation(	
	// string SubjectNa	_		
	// string Segment			
	in string ocgineriti	iaiio /,		



### **GetSegmentStaticRotationHelical**

Return the static pose rotation of a subject segment in helical co-ordinates.

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

 $See \ Also: Get Segment Static Translation, \ Get Segment Static Rotation Matrix,$ 

 $Get Segment Static Rotation Quaternion, \ Get Segment Static Rotation Euler XYZ, \ Get Segment Local Translation, \ Get Segment Local Rotation Helical, \ Get Segment Local Rotation Quaternion, \ Get Segment Local Rotation \ Get Segment Local Rotation \ Get Segment Local Rotation \ Get Segment \ Get Segmen$ 

GetSegmentLocalRotationEulerXYZ				
Input	Subject Name	string	The name of the subject	
	Segment Name	string	The name of the segment.	
Output	Result	Result	Result.Success	
			Result.NotConnected	
			Result.NoFrame	
			Result.InvalidSubjectName	
			Result.InvalidSegmentName	
	Rotation	double[3]	The rotation of the segment	
C++	// class Output_GetSegr	mentStaticRotationHelical		
	// {			
	// public:			
	// Result::Enum Result			
	// double Rotation[	3];		
	// };			
	//			
	// Output_GetSegmentS			
	// GetSegmentStaticRo	,		
	// const String & Su			
	// const String & SegmentName ) const			
	Visan Data Stream SDV	CDDuClient MuClient		
	ViconDataStreamSDK::CPP::Client MyClient; MyClient.Connect( "localhost" );			
	MyClient.GetFrame();	iiilost ),		
	wycheni.Getrianie(),			
	Output GetSegmentSta	nticRotationHelical Output =		
		StaticRotationHelical( "Alice", "Pelvis" );		
MATLAB		entStaticRotationHelical(SubjectName, S	SegmentName )	
	, tempon	( <b>,</b> -	,	
	MyClient = Client();			
	MyClient.Connect( "loca	ılhost");		
	MyClient.GetFrame();	·		
	Output = MyClient.GetS	egmentStaticRotationHelical( "Alice", "Pe	elvis");	
.NET	// public class Output_G	etSegmentStaticRotationHelical		
	// {			
	// public Result Result	t;		
	// public double[] Rotat	ion;		
	// };			
	//			
	// Output_GetSegmentS	staticRotationHelical		
	_	otationHelical( string SubjectName,		
	//	string SegmentName );		

#### **GetSegmentStaticRotationMatrix**

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

 $See \ Also: Get Segment Static Translation, \ Get Segment Static Rotation Helical,$ 

GetSegmentStaticRotationQuaternion, GetSegmentStaticRotationEulerXYZ, GetSegmentLocalTranslation, GetSegmentLocalRotationHelical, GetSegmentLocalRotationQuaternion, GetSegmentLocalRotationEulerXY.

GetSegmentL		SegmentLocalRotationQuaternion	n, GetSegmentLocalRotationEulerXYZ			
Input	Subject Name	string	The name of the subject			
	Segment Name	string	The name of the segment.			
Output	Success	Result	Result.Success			
			Result.NotConnected			
			Result.NoFrame			
			Result.InvalidSubjectName			
			Result.InvalidSegmentName			
	Rotation	double[9]	The rotation of the segment			
C++	· ·	gmentStaticRotationMatrix				
	// {					
	// public:	16.				
	// Result::Enum Resu					
	// double Rotation	[ 9 ];				
	// }; //					
	// // Output_GetSegment	StaticRotationMatrix				
	// GetSegmentStaticF					
	// const String & S	· ·				
	_	egmentName ) const;				
		, solici daling a doginaria and y dolici,				
	ViconDataStreamSDK	ViconDataStreamSDK::CPP::Client MyClient;				
	MyClient.Connect( "localhost" ); MyClient.GetFrame();					
	Output CotSogmontS	tatioBatationMatrix Output -				
	•	taticRotationMatrix Output = tStaticRotationMatrix( "Alice", "Pelvis"	" )•			
MATLAB		entStaticRotationMatrix( SubjectNam				
	70 [Gatpat] = Gotoogii	onetalion totalionivialis (Cabjeet tain	io, oogmona amo /			
	MyClient = Client();					
	MyClient.Connect( "loc	ealhost");				
	MyClient.GetFrame();					
	Output = MyClient.Get	SegmentStaticRotationMatrix( "Alice"	, "Pelvis" );			
.NET	// public class Output_	GetSegmentStaticRotationMatrix				
	// {					
	// public Result Resu					
	// public double[] Rota	ation;				
	// }; 					
	//	0 D M				
	// Output_GetSegment					
		RotationMatrix( string SubjectName,				
	//	string SegmentName );				
	ViconDataStreamSDK	.DotNET.Client MyClient =				
		ViconDataStreamSDK.DotNET.Clien	nt();			
	•					

MyClient.Connect( "localhost" ); MyClient.GetFrame();
Output_GetSegmentStaticRotationMatrix Output =
MyClient.GetSegmentStaticRotationMatrix( "Alice", "Pelvis" );

#### **GetSegmentStaticRotationQuaternion**

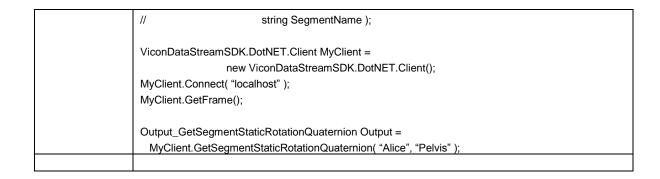
Return the static pose rotation of a subject segment in quaternion co-ordinates.

The quaterion is of the form (x, y, x, w) where w is the real component and x, y & 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,

Colocyment	alRotationEulerX YZ		T
Input	Subject Name	string	The name of the subject
	Segment Name	string	The name of the segment.
Output	Result	Result	Result.Success
			Result.NotConnected
			Result.NoFrame
			Result.InvalidSubjectName
			Result.InvalidSegmentName
	Rotation	double[4]	The rotation of the segment
C++	// class Output_GetSegr	mentStaticRotationQuaternion	
	// {		
	// public:		
	// Result::Enum Result	· •	
	// double Rotation[	4];	
	// };		
	//		
	// Output_GetSegmentS	staticRotationQuaternion	
	// GetSegmentStaticR	otationQuaternion(	
	// const String & Su	bjectName,	
	// const String & SegmentName ) const		
	ViconDataStreamSDK::	· · · · · · · · · · · · · · · · · · ·	
	MyClient.Connect( "loca	ılhost");	
	MyClient.GetFrame();		
	O tra t	tis Deteties Overtonnies Outrout	
		nticRotationQuaternion Output =	~" \.
MATLAB		StaticRotationQuaternion( "Alice", "Pelvi	•
IVIATLAD	% [Output] = GetSegme	entStaticRotationQuaternion(SubjectNan	ne, segmentivame )
	MyClient = Client();		
	MyClient.Connect( "loca	ilhost"):	
	MyClient.GetFrame();	,,	
	my onorm. Sour ramo(),		
	Output = MvClient.GetS	egmentStaticRotationQuaternion( "Alice"	". "Pelvis" ):
.NET		etSegmentStaticRotationQuaternion	, D
	// {		
	// public Result Result	<b>t</b> :	
	// public double[] Rotat		
	// };	,	
	// //		
	// Output_GetSegmentS	staticRotationQuaternion	
		otationQuaternion( string SubjectName,	



#### **GetSegmentStaticRotationEulerXYZ**

Return the static pose rotation of a subject segment in EulerXYZ co-ordinates.

See Also: GetSegmentStaticTranslation, GetSegmentStaticRotationHelical,

 $Get Segment Static Rotation Matrix,\ Get Segment Static Rotation Quaternion,\ Get Segment Local Translation,$ GetSegmentLocalRotationHelical, GetSegmentLocalRotationMatrix, GetSegmentLocalRotationQuaternion,

GetSegmentL	ocalRotationEulerXYZ		
Input	Subject Name	string	The name of the subject
	Segment Name	string	The name of the segment.
Output	Result	Result	Result.Success Result.NotConnected Result.NoFrame Result.InvalidSubjectName Result.InvalidSegmentName
	Rotation	double[3]	The rotation of the segment
C++	// {     // public:     // Result::Enum Resu     // double Rotation     // };     //     // Output_GetSegment     // GetSegmentStaticF     // const String & S     // const String & S     // viconDataStreamSDK     MyClient.Connect( "loc     MyClient.GetFrame();  Output_GetSegmentSt	StaticRotationEulerXYZ totationEulerXYZ( ubjectName, egmentName ) const :CPP::Client MyClient;	vis" ):
MATLAB	MyClient = Client(); MyClient.Connect( "loc MyClient.GetFrame();	entStaticRotationEulerXYZ( SubjectNa alhost" ); SegmentStaticRotationEulerXYZ( "Alic	
.NET	// public class Output_( // { // public Result Result // public double[] Rota // }; // Output_GetSegment	GetSegmentStaticRotationEulerXYZ	

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

Page 70 of 124

# **GetSegmentGlobalTranslation**

Return the translation of a subject segment in global co-ordinates.

The Translation is of the form (x, y, z) where x, y & z are in Millimeters with respect to the global origin.

 $See \ Also: Get Segment Global Rotation Helical, \ Get Segment Global Rotation Matrix,$ 

 $Get Segment Global Rotation Quaternion, \ Get Segment Global Rotation Euler XYZ, \ Get Segment Local Translation, \\ Get Segment Local Rotation Helical, \ Get Segment Local Rotation Matrix, \ Get Segment Local Rotation Quaternion, \\ Get Segment Local Rotation Helical, \ Get Segment Local Rotation Matrix, \ Get Segment Local Rotation Quaternion, \\ Get Segment Local Rotation Matrix, \ Get Segment Local Rotation Matr$ 

GetSegmentL	ocalRotationEulerXYZ		
Input	Subject Name	string	The name of the subject
	Segment Name	string	The name of the segment.
Output	Result	Result	Result.Success
			Result.NotConnected
			Result.NoFrame
			Result.InvalidSubjectName
			Result.InvalidSegmentName
	Translation	double[3]	The translation of the
			segment
	Occluded	boolean	True if the segment was
			present at this frame. If false,
			then Translation will be
			[0,0,0]
C++	// class Output_GetSegmentGlobalTranslation		
	// {		
	// public:		
	// Result::Enum Result;		
	// double Translation[ 3 ];		
	// bool Occluded;		
	// };		
	//		
	// Output_GetSegmentGlobalTranslation GetSegmentGlobalTranslation(		
	// const String & SubjectName,		
	// const String & SegmentName ) const;		
	ViconDataStreamSDK::CPP::Client MyClient;		
	MyClient.Connect( "localhost" );		
	MyClient.EnableSegmentData();		
	MyClient.GetFrame();		
	Output_GetSegmentGlobalTranslation Output =		
	MyClient.GetSegmentGlobalTranslation( "Alice", "Pelvis" );		
MATLAB	% [Output] = GetSegmentGlobalTranslation( SubjectName, SegmentName )		
	MyClient = Client();		
	MyClient.Connect( "localhost" );		
	MyClient.EnableSegmentData();		
	MyClient.GetFrame();		
	Output = MyClient.GetSegmentGlobalTranslation( "Alice", "Pelvis" );		
.NET	// public class Output_GetSegmentGlobalTranslation		
	// {		

# **GetSegmentGlobalRotationHelical**

Return the rotation of a subject segment in global helical co-ordinates.

See Also: GetSegmentGlobalTranslation, GetSegmentGlobalRotationMatrix,

GetSegmentGlobalRotationQuaternion, GetSegmentGlobalRotationEulerXYZ, GetSegmentLocalTranslation, GetSegmentLocalRotationHelical, GetSegmentLocalRotationMatrix, GetSegmentLocalRotationQuaternion,

GetSegmentLocalRotationEulerXYZ

GetSegmentLoc	alRotationEulerXYZ		<del>_</del>
Input	Subject Name	string	The name of the subject
	Segment Name	string	The name of the segment.
Output	Result	Result	Result.Success
			Result.NotConnected
			Result.NoFrame
			Result.InvalidSubjectName
			Result.InvalidSegmentName
	Rotation	double[3]	The rotation of the segment
	Occluded	boolean	True if the segment was
			present at this frame. If false,
			then Rotation will be [0,0,0]
C++	// class Output_GetSeg	mentGlobalRotationHelical	
	// {		
	// public:		
	// Result::Enum Resul	;	
	// double Rotation[	3 ];	
	// bool Occluded;		
	// };		
	// Output_GetSegment0		
	// GetSegmentGlobalF	•	
	// const String & SubjectName, // const String & SegmentName ) const  ViconDataStreamSDK::CPP::Client MyClient;		
	MyClient.Connect( "loca	-	
	MyClient.GetFrame();	,, , , , , , , , , , , , , , , , , , ,	
	, 55		
	Output_GetSegmentGl	obalRotationHelical Output =	
		GlobalRotationHelical( "Alice", "Pelvis" );	
MATLAB	% [Output] = GetSegme	entGlobalRotationHelical(SubjectName,	SegmentName )
	MyCliont - Cliont():		
	MyClient = Client(); MyClient.Connect( "local	alhoet" )·	
	MyClient.GetFrame();	aniost j,	
	wycheni.Geti rame(),		
	Output = MyClient.GetS	SegmentGlobalRotationHelical( "Alice", "F	Pelvis" );
.NET	// public class Output_0	SetSegmentGlobalRotationHelical	
	// {		
	// public Result Resu	lt;	
	// public double[] Rota	tion;	
	// public bool Occlud	ded;	
	// };		

### **GetSegmentGlobalRotationMatrix**

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

 $See\ Also: Get Segment Global Translation,\ Get Segment Global Rotation Helical,$ 

GetSegmentGlobalRotationQuaternion, GetSegmentGlobalRotationEulerXYZ, GetSegmentLocalRotationHelical, GetSegmentLocalRotationQuaternion, GetSegmentLocalRotationEulerXYZ

Input	Subject Name	string	The name of the subject
mpat	Segment Name	string	The name of the segment.
Output	Success	Result	Result.Success
Output	Success	Result	Result.NotConnected
			Result.NoFrame
			Result.InvalidSubjectName
			Result.InvalidSegmentName
	Rotation	double[9]	The rotation of the segment
	Occluded	boolean	True if the segment was
			present at this frame. If false,
			then Rotation will be all 0.
C++	// class Output_GetSeg	mentGlobalRotationMatrix	
	// {		
	// public:		
	// Result::Enum Result		
	// double Rotation[	9 1:	
	// bool Occluded;	•	
	// };		
	// Output_GetSegment@	HohalRotationMatrix	
	// GetSegmentGlobalR		
	=		
	// const String & SubjectName, // const String & SegmentName ) const;		
	ViconDataStreamSDK::		
	MyClient.Connect( "loca	alhost");	
	MyClient.GetFrame();		
	Output_GetSegmentGlobalRotationMatrix Output =		
	MyClient.GetSegment	GlobalRotationMatrix( "Alice", "Pelvis" );	
MATLAB	% [Output] = GetSegme	entGlobalRotationMatrix( SubjectName, S	SegmentName)
	MyClient = Client();		
	MyClient.Connect( "loca	alhost" ):	
	MyClient.GetFrame();	,,	
	Output = MyClient.GetS	egmentGlobalRotationMatrix( "Alice", "P	elvis");
.NET	'	etSegmentGlobalRotationMatrix	
	// {	<b>.</b>	
	// public Result Result		
	// public double[] Rotat		
	// public bool Occlud	iea;	
	// };		
	//		

# **GetSegmentGlobalRotationQuaternion**

Return the rotation of a subject segment in global quaternion co-ordinates.

The quaterion is of the form (x, y, x, w) where w is the real component and x, y & 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

GetsegmentLo	cainotationEulerx 12		
Input	Subject Name	string	The name of the subject
	Segment Name	string	The name of the segment.
Output	Result	Result	Result.Success
			Result.NotConnected
			Result.NoFrame
			Result.InvalidSubjectName
			Result.InvalidSegmentName
	Rotation	double[4]	The rotation of the segment
	Occluded	boolean	True if the segment was
			present at this frame. If false,
			then Rotation will be [0,0,0,0]
C++	// class Output_GetSeg	mentGlobalRotationQuaternion	• • • • •
	// {		
	// public:		
	// Result::Enum Resul	<del>;</del> ,	
	// double Rotation[	4];	
	// bool Occluded;		
	// };		
	// // // Output_GetSegmentGlobalRotationQuaternion // GetSegmentGlobalRotationQuaternion(		
	// const String & Su	ıbjectName,	
	// const String & Se	egmentName ) const	
		,	
	ViconDataStreamSDK:	CPP::Client MyClient;	
	MyClient.Connect( "local		
	MyClient.GetFrame();	,	
	7 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 -		
	Output_GetSegmentGlobalRotationQuaternion Output =		
	MyClient.GetSegmen	tGlobalRotationQuaternion( "Alice", "Pel	vis" );
MATLAB	% [Output] = GetSegme	entGlobalRotationQuaternion( SubjectNa	ame, SegmentName )
	MyClient = Client();		
	MyClient.Connect( "local	alhost");	
	MyClient.GetFrame();		
	Output = MyClient.GetS	SegmentGlobalRotationQuaternion( "Alic	e", "Pelvis" );
.NET	// public class Output_0	GetSegmentGlobalRotationQuaternion	
	// {		
	// public Result Resu	lt;	
	// public double[] Rota	tion;	

### **GetSegmentGlobalRotationEulerXYZ**

Return the rotation of a subject segment in global EulerXYZ co-ordinates.

See Also: GetSegmentGlobalTranslation, GetSegmentGlobalRotationHelical,

GetSegmentGlobalRotationMatrix, GetSegmentGlobalRotationQuaternion, GetSegmentLocalTranslation, GetSegmentLocalRotationHelical, GetSegmentLocalRotationMatrix, GetSegmentLocalRotationQuaternion,

GetSegmentLocalRotationEulerXYZ

GetSegmentLo	ocalRotationEulerXYZ			
Input	Subject Name	string	The name of the subject	
	Segment Name	string	The name of the segment.	
Output	Result	Result	Result.Success	
			Result.NotConnected	
			Result.NoFrame	
			Result.InvalidSubjectName	
			Result.InvalidSegmentName	
	Rotation	double[3]	The rotation of the segment	
	Occluded	boolean	True if the segment was	
			present at this frame. If false,	
			then Rotation will be [0,0,0]	
C++	// class Output_GetSeg	mentGlobalRotationEulerXYZ		
	// {			
	// public:			
	// Result::Enum Resul	t;		
	// double Rotation			
		// bool Occluded;		
	// };			
	// Output_GetSegmentGlobalRotationEulerXYZ			
	// GetSegmentGlobalRotationEulerXYZ(			
	// const String & SubjectName,			
	// const String & SegmentName ) const			
	Vices Data Stream SDV ·· CDD ·· Client MuClient			
	ViconDataStreamSDK::CPP::Client MyClient; MyClient.Connect( "localhost" ); MyClient.GetFrame();  Output_GetSegmentGlobalRotationEulerXYZ Output =			
	_	tGlobalRotationEulerXYZ( "Alice", "Pelv	/is" ):	
MATLAB		entGlobalRotationEulerXYZ( SubjectNa		
 	70 [Output] = Cotcogiii		arro, Goginoria tarro /	
	MyClient = Client();			
	MyClient.Connect( "loc	alhost"):		
	MyClient.GetFrame();	,		
	,			
	Output = MyClient.Get	SegmentGlobalRotationEulerXYZ( "Alic	e", "Pelvis" );	
.NET		GetSegmentGlobalRotationEulerXYZ	·	
	//{	-		
	// public Result Resu	ılt;		
	// public double[] Rota			
	// public bool Occlu	ded;		
	// };			

# **GetSegmentLocalTranslation**

Return the translation of a subject segment in local co-ordinates relative to its parent segment.

See Also: GetSegmentLocalRotationHelical, GetSegmentLocalRotationMatrix,

GetSegmentLocalRotationQuaternion, GetSegmentLocalRotationEulerXYZ,

 $Get Segment Global Translation, Get Segment Global Rotation Helical, \ Get Segment Global Rotation Matrix,$ 

GetSegmentGlobalRotationQuaternion, GetSegmentGlobalRotationEulerXYZ

_		string	
Input	Subject Name	string	The name of the subject
	Segment Name	string	The name of the segment.
Output	Result	Result	Result.Success
			Result.NotConnected
			Result.NoFrame
			Result.InvalidSubjectName
			Result.InvalidSegmentName
	Translation	double[3]	The translation of the
			segment
	Occluded	boolean	True if the segment was
			present at this frame. If false,
			then Translation will be
			[0,0,0]
C++	// class Output_GetSeg	mentLocalTranslation	
	// {		
	// public:		
	// Result::Enum Resul	t;	
	// double Translation		
	// bool Occluded;	• •	
	// };		
	//		
	// // Output_GetSegmentLocalTranslation GetSegmentLocalTranslation( // const String & SubjectName,		
	_	egmentName ) const;	
	" const offing & oc	ginentivanie / const,	
	ViconDataStreamSDK:	CPP: Client MvClient:	
	MyClient.Connect( "local	-	
	MyClient.EnableSegme		
	MyClient.GetFrame();	media(),	
	inyoneni. Geti raine(),		
	Output GetSeamentlin	calTranslation Output =	
		LocalTranslation( "Alice", "Pelvis" );	
MATLAB		entLocalTranslation( SubjectName, Segn	nentName )
	, , , , , , , , , , , , , , , , , , ,		,
	MyClient = Client();		
	MyClient.Connect( "local	alhost" ):	
	MyClient.EnableSegme		
	MyClient.GetFrame();	· · · · · · · · · · · · · · · · · · ·	
	, 5		
	Output = MyClient.GetS	SegmentLocalTranslation( "Alice", "Pelvis	");
.NET	// public class Output_0	SetSegmentLocalTranslation	
	// {		
	// public Result Resu		
	// public double[] Trans	slation;	

# **GetSegmentLocalRotationHelical**

Return the rotation of a subject segment in local helical co-ordinates relative to its parent segment.

See Also: GetSegmentLocalTranslation, GetSegmentLocalRotationMatrix, GetSegmentLocalRotationQuaternion, GetSegmentLocalRotationEulerXYZ,

 $Get Segment Global Translation, Get Segment Global Rotation Helical, \ Get Segment Global Rotation Matrix,$ 

GetSegmentGlobalRotationQuaternion, GetSegmentGlobalRotationEulerXYZ

Input	Subject Name	string	The name of the subject	
πραι			-	
Output	Segment Name Result	string	The name of the segment.	
Output	Result	Result	Result.Success Result.NotConnected	
			Result.NoFrame	
			Result InvalidSubjectName	
	Pototion	doublo[2]	Result.InvalidSegmentName	
	Rotation Occluded	double[3] boolean	The rotation of the segment	
	Occiuded	boolean	True if the segment was present at this frame. If false,	
			then Rotation will be [0,0,0]	
C++	// class Output GetSed	mentLocalRotationHelical	alon Rolation will be [6,6,6]	
	// {			
	// public:			
	// Result::Enum Resul	t;		
	// double Rotation[			
	// bool Occluded;			
	// };			
	//			
	// Output_GetSegmentLocalRotationHelical			
	// GetSegmentLocalRe	// GetSegmentLocalRotationHelical(		
	// const String & Su	• • •		
	// const String & Se	// const String & SegmentName ) const		
	ViconDataStreamSDK:	:CPP::Client MyClient;		
	MyClient.Connect( "local	alhost");		
	MyClient.GetFrame();	MyClient.GetFrame();		
	Output_GetSegmentLo	Output_GetSegmentLocalRotationHelical Output =		
	MyClient.GetSegment	:LocalRotationHelical( "Alice", "Pelvis" );		
MATLAB	% [Output] = GetSegmentLocalRotationHelical( SubjectName, SegmentName )			
	MyClient = Client();			
	MyClient.Connect( "local	alhost");		
	MyClient.GetFrame();			
	Output = MacOlient Out	Compost coolDetetical alical "Alica" "D	olvio" ):	
.NET		SegmentLocalRotationHelical( "Alice", "P	eivis j;	
.INL I		SetSegmentLocalRotationHelical		
	// {	14.		
	// public Result Resu			
	// public double[] Rota // public bool Occlude			
	· ·	u <del>c</del> u,		
	// };			

# **GetSegmentLocalRotationMatrix**

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

See Also: GetSegmentLocalTranslation, GetSegmentLocalRotationQuaternion, GetSegmentLocalRotationEulerXYZ, GetSegmentGlobalTranslation,GetSegmentGlobalRotationHelical, GetSegmentGlobalRotationMatrix, GetSegmentGlobalRotationQuaternion,

GetSegmentGlobalRotationEulerXYZ

Input	Subject Name	string	The name of the subject	
	Segment Name	string	The name of the segment.	
Output	Result	Result	Result.Success	
			Result.NotConnected	
			Result.NoFrame	
			Result.InvalidSubjectName	
			Result.InvalidSegmentName	
	Rotation	double[9]	The rotation of the segment	
	Occluded	boolean	True if the segment was	
			present at this frame. If false,	
			then Rotation will be all 0	
C++	// class Output_GetSeg	mentLocalRotationMatrix		
	// {			
	// public:			
	// Result::Enum Resul	t;		
	// double Rotation	9];		
	// bool Occluded;			
	// };			
	//			
	// Output_GetSegmentI	_ocalRotationMatrix		
	// GetSegmentLocalRe	otationMatrix(		
	// const String & St	ıbjectName,		
	// const String & SegmentName ) const;			
	ViconDataStreamSDK:	:CPP::Client MyClient;		
	MyClient.Connect( "loc	alhost");		
	MyClient.GetFrame();	MyClient.GetFrame(); Output_GetSegmentLocalRotationMatrix Output =		
	Output GetSegmentlio			
		LocalRotationMatrix( "Alice", "Pelvis" );		
MATLAB	% [Output] = GetSegmentLocalRotationMatrix( SubjectName, SegmentName )			
	MyClient = Client();			
	MyClient.Connect( "loc	alhost");		
	MyClient.GetFrame();			
	Output = MyClient.GetS	SegmentLocalRotationMatrix( "Alice", "Po	elvis" );	
.NET	// public class Output_0	SetSegmentLocalRotationMatrix		
	// {			
	// public Result Resu	lt;		
	// public double[] Rota	tion;		
	// public bool Occlu	ded;		
	// <del>}</del> ;			

### **GetSegmentLocalRotationQuaternion**

Return the rotation of a subject segment in local quaternion co-ordinates relative to its parent segment. The quaterion is of the form (x, y, x, w) where w is the real component and x, y & 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

detdegmentd	IODAINOIAIIONEUIEIX IZ		_	
Input	Subject Name	string	The name of the subject	
	Segment Name	string	The name of the segment.	
Output	Result	Result	Result.Success	
			Result.NotConnected	
			Result.NoFrame	
			Result.InvalidSubjectName	
			Result.InvalidSegmentName	
	Rotation	double[4]	The rotation of the segment	
	Occluded	boolean	True if the segment was	
			present at this frame. If false,	
			then Rotation will be [0,0,0,0]	
C++	// class Output_GetSeg	mentLocalRotationQuaternion		
	// {			
	// public:			
	// Result::Enum Resul	t;		
	// double Rotation	[ 4 ];		
	// bool Occluded;			
	// };			
	//			
	// Output_GetSegmentI	_ocalRotationQuaternion		
	// GetSegmentLocalR	•		
	// const String & St	ubjectName,		
	// const String & Se	ViconDataStreamSDK::CPP::Client MyClient;		
	ViconDataStreamSDK:			
	MyClient.Connect( "localhost" );			
	MyClient.GetFrame();			
	Output GetSegmentlin	calRotationQuaternion Output =		
	·	atLocalRotationQuaternion( "Alice", "Pelvi	s"):	
MATLAB		entLocalRotationQuaternion( SubjectNar	-	
	MyClient = Client();			
	MyClient.Connect( "loc	alhost");		
	MyClient.GetFrame();			
	Output = MyClient.GetS	SegmentLocalRotationQuaternion( "Alice	", "Pelvis" );	
.NET		GetSegmentLocalRotationQuaternion		
	// {			
	// public Result Resu	lt;		

### **GetSegmentLocalRotationEulerXYZ**

Return the rotation of a subject segment in local EulerXYZ co-ordinates relative to its parent segment.

See Also: GetSegmentLocalTranslation, GetSegmentLocalRotationHelical, GetSegmentLocalRotationMatrix, GetSegmentLocalRotationQuaternion, GetSegmentGlobalTranslation,GetSegmentGlobalRotationHelical, GetSegmentGlobalRotationMatrix, GetSegmentGlobalRotationQuaternion,

GetSegmentGlobalRotationEulerXYZ

Input	Subject Name	string	The name of the subject
	Segment Name	string	The name of the segment.
Output	Result	Result	Result.Success Result.NotConnected Result.NoFrame Result.InvalidSubjectName Result.InvalidSegmentName
	Rotation	double[3]	The rotation of the segment
	Occluded	boolean	True if the segment was present at this frame. If false, then Rotation will be [0,0,0]
C++	// class Output_GetSegmentLocalRotationEulerXYZ // { // public: // Result::Enum Result; // double Rotation[ 3 ]; // bool Occluded; // }; // // Output_GetSegmentLocalRotationEulerXYZ // GetSegmentLocalRotationEulerXYZ( // const String & SubjectName, // const String & SegmentName ) const  ViconDataStreamSDK::CPP::Client MyClient; MyClient.Connect( "localhost" ); MyClient.GetFrame();		);
MATLAB	MyClient = Client(); MyClient.Connect( "local MyClient.GetFrame();		
NET			"Pelvis" );
.INE I	// { // public Result Result // public double[] Rotati	t; ion;	
	// public bool Occlud	led;	
MATLAB .NET	ViconDataStreamSDK::0 MyClient.Connect( "local MyClient.GetFrame();  Output_GetSegmentLoc MyClient.GetSegmentL % [Output] = GetSegment MyClient = Client(); MyClient.Connect( "local MyClient.GetFrame();  Output = MyClient.GetSe // public class Output_Ge // public Result Result // public double[] Rotati	CPP::Client MyClient; Ilhost");  calRotationEulerXYZ Output = _ocalRotationEulerXYZ( "Alice", "Pelvis" IntLocalRotationEulerXYZ( SubjectName Ilhost");  egmentLocalRotationEulerXYZ( "Alice", etSegmentLocalRotationEulerXYZ t; ion;	, SegmentName )

### **GetMarkerCount**

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

Subject Name	string	The name of the subject	
Result	Result	Result.Success	
		Result.NotConnected	
		Result.NoFrame	
		Result.InvalidSubjectName	
Marker Count	unsigned integer	The number of markers	
// class Output_GetMark	kerCount		
-			
// unsigned int Marker(	Count;		
=			
//			
// Output_GetMarkerCo	unt GetMarkerCount(		
· ·			
,			
ViconDataStreamSDK::	CPP::Client MyClient;		
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		
% [Output] = GetMarker	rCount( SubjectName )		
MyClient = Client();			
The state of the s			
MyClient.Connect( "localhost" );			
Output = MvClient.GetMarkerCount( "Bob" ): % Output Result == NoFrame			
MyClient.GetFrame();			
Output = MvClient GetMarkerCount( "Alice" ):			
	Result  Marker Count  // class Output_GetMark // { // public: // Result::Enum Result // unsigned int Marker( // }; // // Output_GetMarkerCo // const String & Subj  ViconDataStreamSDK:: MyClient.EnableMarker MyClient.Connect( "loca  Output_GetMarkerCour Output = MyClient.GetM  MyClient.GetFrame();  Output = MyClient.GetM  // //  Output = MyClient.GetM  % [Output] = GetMarker  MyClient = Client(); MyClient.EnableMarker MyClient.Connect( "loca  Output = MyClient.GetM  MyClient.Connect( "loca  Output = MyClient.GetM  MyClient.GetFrame();  Output = MyClient.GetM  MyClient.GetFrame();  Output = MyClient.GetM	Result   Result   Result	

```
Output = MyClient.GetMarkerCount( "Bob" ); % Output.Result == Success
                                            % Output.MarkerCount >= 0
.NET
                   // public class Output_GetMarkerCount
                   // {
                   // public Result Result;
                   // public uint MarkerCount;
                   // };
                   //
                   // Output_GetMarkerCount GetMarkerCount( string SubjectName );
                   ViconDataStreamSDK.DotNET.Client MyClient =
                                    new ViconDataStreamSDK.DotNET.Client();
                   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
```

#### **GetMarkerName** Return the name of a marker for a specified subject. This can be passed into GetMarkerGlobalTranslation. See Also: GetMarkerCount, GetMarkerGlobalTranslation Input Subject Name string The name of the subject Marker Index unsigned integer The index of the marker. Output Result Result Result.Success Result.NotConnected Result.NoFrame Result.InvalidSubjectName Result.InvalidIndex Marker Name string The name of the marker C++ A valid Marker Index is between 0 and GetMarkerCount()-1 // class Output\_GetMarkerName // { // public: // Result::Enum Result; // String MarkerName; // }; // // Output GetMarkerName GetMarkerName( // const String & SubjectName, // const unsigned int MarkerIndex ) const; 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) **MATLAB** A valid Marker Index is between 1 and GetMarkerCount()

```
% [Output] = GetMarkerName( SubjectName, MarkerIndex )
                  MyClient = Client();
                  MyClient.Connect( "localhost" );
                  MyClient.EnableMarkerData();
                  MyClient.GetFrame();
                  OutputGMC = MyClient.GetMarkerCount( "Bob" );
                                         // OutputGMC.Result == Success
                                         // OutputGMC.MarkerCount == 2
                  OutputGMN = MyClient.GetMarkerName( "Alice", 1 );
                                     // OutputGMN.Result == InvalidSubjectName
                                     // OutputGMN.MarkerName == ""
                                     // (no "Alice")
                  OutputGMN = MyClient.GetMarkerName( "Bob", 1 );
                                       // OutputGMN.Result == Success
                                       // OutputGMN.MarkerName == "LASI"
                  OutputGMN = MyClient.GetMarkerName( "Bob", 2 );
                                       // OutputGMN.Result == Success
                                       // OutputGMN.MarkerName == "RASI"
                  OutputGMN = MyClient.GetMarkerName( "Bob", 3 );
                                       // OutputGMN.Result == InvalidIndex
                                       // OutputGMN.MarkerName == ""
                                       // (no third marker)
.NET
                  A valid Marker Index is between 0 and GetMarkerCount()-1
                  // public class Output_GetMarkerName
                  // {
                  // public Result Result;
                  // public string MarkerName;
                  // };
                  // Output_GetMarkerName GetMarkerName( string SubjectName,
                                        uint MarkerIndex);
                  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)

### **GetMarkerParentName**

Return the name of the segment which is the parent of this marker.

Input	Subject Name	string	The name of the subject
	Marker Name	string	The name of the marker.
Output	Result	Result	Result.Success Result.NotConnected Result.NoFrame Result.InvalidSubjectName Result.InvalidMarkerName
	Segment Name	string	The name of the parent segment.
C++	// class Output_GetMarkerParentName // { // public: // Result::Enum Result; // String SegmentName; // }; // // Output_GetMarkerParentName GetMarkerParentName( // const String & SubjectName, // const String & MarkerName ) const;  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	% [Output] = GetMarke	rParentName( SubjectName, Mark	erName )
	1.		
.NET	// public class Output_0 // { // public Result Result // public string Segmen // };	GetMarkerParentName	

### **GetMarkerGlobalTranslation**

Return the translation of a subject marker in global co-ordinates.

The Translation is of the form (x, y, z) where x, y & z are in Millimeters with respect to the global origin.

See Also: GetMarkerName

See Also : Getil				
Input	Subject Name	string	The name of the subject	
	Marker Name	string	The name of the marker.	
Output	Result	Result	Result.Success	
			Result.NotConnected	
			Result.NoFrame	
			Result.InvalidSubjectName	
			Result.InvalidMarkerName	
	Translation	double[3]	The translation of the marker	
	Occluded	boolean	True if the marker was	
			present at this frame. If false,	
			then Translation will be	
			[0,0,0]	
C++	// class Output_GetMarl	kerGlobalTranslation		
	// {			
	// public:			
	// Result::Enum Result	;		
	// bool Occluded;			
	// };			
	//			
	// Output_GetMarkerGlobalTranslation GetMarkerGlobalTranslation(			
	// const String & Su	· · · · · · · · · · · · · · · · · · ·		
	_			
	Vican Data Stroom SDK···CDD···Client MvClient			
	ViconDataStreamSDK::CPP::Client MyClient; MyClient.Connect( "localhost" );			
	MyClient.Connect( localitost ),  MyClient.EnableMarkerData();  MyClient.GetFrame();			
	inyonent.oeti idille(),			
Output_GetMarkerGlobalTranslation Output =				
	-	obalTranslation( "Alice", "LASI" );		
MATLAB		GlobalTranslation( SubjectName, Market	erName )	
	MyClient = Client();			
	MyClient.Connect( "loca	ılhost" ):		
	MyClient.EnableMarker			
	MyClient.GetFrame();			
		larkerGlobalTranslation( "Alice", "LASI" )	);	
.NET	· -	etMarkerGlobalTranslation		
	// {			
	// public Result Resul			
	// public double[] Trans			
	// public bool Occlud	led;		
	// };			

### **GetUnlabeledMarkerCount**

Return the number of unlabeled markers in the data stream. This information can be used in conjunction with GetGlobalUnlabeledMarkerTranslation

See Also: GetGlobalUnlabeledMarkerTranslation

	etGlobalUnlabeledMarke	r i ransiation			
Input	D If	D II	D		
Output	Result	Result	Result.Success		
			Result.NotConnected		
	1		Result.NoFrame		
	MarkerCount	unsigned integer	The number of markers		
C++	// class Output_GetU	nlabeledMarkerCount			
	// {	// {			
	// public:	// public:			
	// Result::Enum Result;				
	// unsigned int Mark	// unsigned int MarkerCount;			
		<i> </i>			
	//				
	// Output_GetUnlabel	// Output_GetUnlabeledMarkerCount GetUnlabeledMarkerCount() const;			
	ViconDataStreamSD	K::CPP::Client MyClient;			
	MyClient.EnableUnla				
	MyClient.Connect( "lo				
	MyClient.GetFrame()	MyClient.GetFrame();			
	Output_GetUnlabeled	Output_GetUnlabeledMarkerCount Output =			
	MyClient.GetUnlabe	MyClient.GetUnlabeledMarkerCount(); // Output.Result == Success			
		// Output.MarkerCount >= 0			
MATLAB	% [Output] = GetUnla	% [Output] = GetUnlabeledMarkerCount();			
	MyClient = Client();				
	MyClient.EnableUnla	MyClient.EnableUnlabeledMarkerData();			
		MyClient.Connect( "localhost" );			
	MyClient.GetFrame()	MyClient.GetFrame();			
	Output = MyClient.Ge	Output = MyClient.GetUnlabeledMarkerCount(); // Output.Result == Success			
		// Output.MarkerCount >= 0			
.NET		_GetUnlabeledMarkerCount			
		<i>  </i>			
		// public Result;			
	· ·	// public uint MarkerCount;			
	// };	// <b>}</b> ;			
	//				
	// Output_GetUnlabel	// Output_GetUnlabeledMarkerCount GetUnlabeledMarkerCount();			
		ViconDataStreamSDK.DotNET.Client MyClient =			
	ne	new ViconDataStreamSDK.DotNET.Client();			
		MyClient.EnableUnlabeledMarkerData();			
	MyClient.Connect( "lo	MyClient.Connect( "localhost" );			
	MyClient.GetFrame()	MyClient.GetFrame();			
	Output_GetUnlabeled	MarkerCount Output =			

MyClient.GetUnlabeledMarkerCount(); // Output.Result == Success
// Output.MarkerCount >= 0

### GetUnlabeledMarkerGlobalTranslation

Return the translation of an unlabeled marker in global co-ordinates.

The Translation is of the form (x, y, z) where x, y & z are in Millimeters with respect to the global origin.

See Also: GetUnlabelledMarkerCount

See Also : Ge	tUnlabelledMarkerCount				
Input	Marker Index	unsigned integer	The index of the marker.		
Output	Result	Result	Result.Success		
			Result.NotConnected		
			Result.NoFrame		
			Result.InvalidIndex		
	Translation	double[3]	The translation of the marker		
C++	A valid Marker Index is between 0 and GetUnlabeledMarkerCount()-1				
-		A valid Warker index to between a did determabledwarker bount()			
	// class Output GetUn	// class Output_GetUnlabeledMarkerGlobalTranslation			
	// {	· · · · · · · · · · · · · · · · · · ·			
	// public:				
	// Result::Enum Resu				
	// double Translat	ion[ 3 ];			
		// };			
	**				
	•	// Output_GetUnlabeledMarkerGlobalTranslation			
		// GetUnlabeledMarkerGlobalTranslation(			
	// const unsigned in	// const unsigned int MarkerIndex ) const;			
	ViconDataStreamSDK	ViconDataStreamSDK::CPP::Client MyClient;			
		MyClient.Connect( "localhost" );			
	-	MyClient.EnableUnlabeledMarkerData();			
	MyClient.GetFrame();	(,			
	Output_GetUnlabeledMarkerGlobalTranslation Output =  MyClient.GetUnlabeledMarkerGlobalTranslation(0);				
MATLAB	A valid Marker Index is between 1 and GetUnlabeledMarkerCount()				
	% [Output] = GetUnlabeledMarkerGlobalTranslation( MarkerIndex )  MyClient = Client(); MyClient.Connect( "localhost" );				
MyClient.EnableUnlabeledMarkerData(); MyClient.GetFrame();					
	iviy Onoric. Oct. rainle(),				
	Output = MyClient.GetUnlabeledMarkerGlobalTranslation( 1 );				
.NET		A valid Marker Index is between 0 and GetUnlabeledMarkerCount()-1			
	7 Valla Markor Inc	7. Valid Markor mack is between a and Setoniabeledivializationality.			
	// public close Output Cott InloholodMarker Clohol Translation				
		// public class Output_GetUnlabeledMarkerGlobalTranslation // { // public Result Result;			
	·				
	// public double[] Translation;				
	// };				
	//				
	•	// Output_GetUnlabeledMarkerGlobalTranslation			
// GetUnlabeledMarkerGlobalTranslation( uint MarkerIndex ) const;			ndex ) const;		

ViconDataStreamSDK.DotNET.Client MyClient =
new ViconDataStreamSDK.DotNET.Client();
MyClient.Connect( "localhost" );
MyClient.EnableUnlabeledMarkerData();
MyClient.GetFrame();
Output_GetUnlabeledMarkerGlobalTranslation Output =
MyClient.GetUnlabeledMarkerGlobalTranslation( 0 );

### **GetDeviceCount**

Return the number of ForcePlates, EMGs, and other devices in the DataStream. This information can be used in conjunction with GetDeviceName

See Also: GetDeviceName

	tDeviceName			
Input				
Output	Result	Result	Result.Success	
			Result.NotConnected	
			Result.NoFrame	
	Device Count	unsigned integer	The number of devices	
C++	// class Output_GetDe	// class Output_GetDeviceCount		
	// {	// {		
	// public:	// public:		
	// Result::Enum Result;			
	// unsigned int Devic	// unsigned int DeviceCount;		
	// };			
	//			
	// Output_GetDeviceC	// Output_GetDeviceCount GetDeviceCount() const;		
	ViconDataStreamSDF	:::CPP::Client MyClient;		
	MyClient.EnableDevice	eData();		
	MyClient.Connect( "lo	calhost");		
	MyClient.GetFrame();	MyClient.GetFrame();		
	Output_GetDeviceCo	Output_GetDeviceCount Output = MyClient.GetDeviceCount();		
		// Output.Result == Success		
		// Output.DeviceCount >= 0		
MATLAB	% [Output] = GetDevi	% [Output] = GetDeviceCount()		
	MyClient = Client();	MyClient = Client();		
		MyClient.EnableDeviceData();		
		MyClient.Connect( "localhost" );		
	MyClient.GetFrame();	MyClient.GetFrame();		
	Output = MyClient.Ge	Output = MyClient.GetDeviceCount(); // Output.Result == Success		
	// Output.DeviceCount >= 0			
.NET	// public class Output_	GetDeviceCount		
	// {			
		// public Result Result;		
	·	// public uint DeviceCount;		
	// };	// <b>}</b> ;		
	//			
	// Output_GetDeviceC	// Output_GetDeviceCount GetDeviceCount();		
		ViconDataStreamSDK.DotNET.Client MyClient =		
	nev	new ViconDataStreamSDK.DotNET.Client();		
		MyClient.EnableDeviceData();		
	MyClient.Connect( "lo	MyClient.Connect( "localhost" );		
	MyClient.GetFrame();	MyClient.GetFrame();		
	Output_GetDeviceCount Output = MyClient.GetDeviceCount();			

	// Output.Result == Success
	// Output.DeviceCount >= 0

#### **GetDeviceName** Return the name and type of a device. This name can be passed into device functions. See Also: GetDeviceCount, GetDeviceOutputCount, GetDeviceOutputValue The index of the device. Input Device Index unsigned integer Output Result Result Result.Success Result.NotConnected Result.NoFrame Result.InvalidIndex **Device Name** The name of the device string Unknown Device Type DeviceType ForcePlate C++ A valid Device Index is between 0 and GetDeviceCount()-1 // class Output\_GetDeviceName // { // public: // Result::Enum Result; // String DeviceName: // DeviceType::Enum DeviceType; // }; // Output\_GetDeviceName // GetDeviceName( const unsigned int DeviceIndex ) const; 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 == ""

**MATLAB** 

// OutputGDN.DeviceType == Unknown

A valid Device Index is between 1 and GetDeviceCount()

```
% [Output] = GetDeviceName( DeviceIndex )
                  MyClient = Client();
                  MyClient.Connect( "localhost" );
                  MyClient.EnableDeviceData();
                  MyClient.GetFrame();
                  OutputGDC = MyClient.GetDeviceCount( DeviceCount );
                                     % OutputGDC.Result == Success
                                     % OutputGDC.DeviceCount == 2
                  OutputGDN = MyClient.GetDeviceName(1);
                                     % OutputGDN.Result == Success
                                     % OutputGDN.DeviceName == "ZeroWire"
                                     % OutputGDN.DeviceType == Unknown
                  OutputGDN = MyClient.GetDeviceName(2);
                                     % OutputGDN.Result == Success
                                     % OutputGDN.DeviceName == "AMTI #1"
                                     % OutputGDN.DeviceType == ForcePlate
                  OutputGDN = MyClient.GetDeviceName(3);
                                     % OutputGDN.Result == InvalidIndex
                                     % OutputGDN.DeviceName == ""
                                     % OutputGDN.DeviceType == Unknown
.NET
                  A valid Device Index is between 0 and GetDeviceCount()-1
                 // public class Output_GetDeviceName
                 // {
                 // public Result Result;
                 // public string DeviceName;
                 // public DeviceType DeviceType;
                 // };
                 // Output_GetDeviceName
                 // GetDeviceName( uint DeviceIndex );
                  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
```

# **GetDeviceOutputCount**

Return the number of outputs for a device in the data stream. This information can be used in conjunction with GetDeviceOutputName

See Also: GetDeviceName, GetDeviceOutputName

	etDeviceName, GetDevice	1	T		
Input	Device Name	string	The device name		
Output	Result	Result	Result.Success		
			Result.NotConnected		
			Result.NoFrame		
			Result.InvalidDeviceName		
	Device Output	unsigned integer	The number of device		
	Count		outputs		
C++	// class Output_GetDev	// class Output_GetDeviceOutputCount			
	// {	·			
	// public:				
	// Result::Enum Resul				
	// unsigned int Device	OutputCount;			
	// };	,			
	//				
		tputCount GetDeviceOutputCount(			
	· ·	& DeviceName ) const;			
	,,g	,			
	ViconDataStreamSDK:	ViconDataStreamSDK::CPP::Client MyClient;			
	MyClient.Connect( "localhost" );				
	MyClient.EnableDeviceData();				
	MyClient.ErlableDeviceData(),  MyClient.GetFrame();				
	wyonencoch rame(),				
	Output_GetDeviceOutputCount Output;				
	Output = MyClient.GetDeviceOutputCount( "DataGlove" );				
	// Output.Result == InvalidDeviceName				
	// Output.DeviceOutputCount == 0				
	// (no "DataGlove" device)				
	" (no batablove device)				
	Output = MyClient.GetDeviceOutputCount( "ZeroWire" );				
	// Output.Result == Success				
	// Output.DeviceOutputCount == 6				
MATLAB		eOutputCount( DeviceName )			
		,			
	MyClient = Client();				
	MyClient.Connect( "localhost" );				
		MyClient.EnableDeviceData();			
	MyClient.GetFrame();				
	,				
	Output = MyClient.GetDeviceOutputCount( "DataGlove" );				
	// Output.Result == InvalidDeviceName				
	// Output.DeviceOutputCount == 0				
	// (no "DataGlove" device)				
	" (no batableve device)				
	Output = MvClient.Get[	Output = MyClient.GetDeviceOutputCount( "ZeroWire" );			
	// Output.Result == Success				
		Output.DeviceOutputCount == 6			
.NET	// public class Output (				
	" pasie siace catput_c				

```
// {
// public Result Result;
// public uint DeviceOutputCount;
// };
// Output_GetDeviceOutputCount GetDeviceOutputCount( string DeviceName );
ViconDataStreamSDK.DotNET.Client MyClient =
                 new ViconDataStreamSDK.DotNET.Client();
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
```

# **GetDeviceOutputName**

Return the name and SI unit of a device output. This name can be passed into GetDeviceOutputValue.

See Also: GetDeviceCount, GetDeviceOutputCount, GetDeviceOutputValue

Input	Device Name	OutputCount, GetDeviceOutputVa string	The device name		
прис	Device Output	integer	The index of the device		
	Index	lineger	output.		
Output	Result	Result	Result.Success		
Catpat	rtoodit	result	Result.NotConnected		
			Result.NoFrame		
			Result.InvalidDeviceName		
			Result.InvalidIndex		
	Device Output	string	The name of the device		
	Name	ag	output, e.g.		
			"Fx" - Force X		
			"Fy" - Force Y		
			"Fz" - Force Z		
			"Mx" - Moment X		
			"My" - Moment Y		
			"Mz" - Moment Z		
			"Cx" - Centre Of Pressure X		
			"Cy" - Centre Of Pressure Y		
			"Cz" - Centre Of Pressure Z		
			"Pin1" - Analog Input 1		
			"Pin2" - Analog Input 2		
	Device Output	Unit	The unit of the device output.		
	Unit		Unit.Unknown		
			Unit.Volt		
			Unit.Newton		
			Unit.NewtonMeter		
			Unit.Meter		
C++	A valid Device Output Index is between 0 and GetDeviceOutputCount()-1				
	V				
	// class Output_GetDeviceOutputName				
	//{				
	// public:				
	// Result::Enum Result;				
	// String DeviceOutputName;				
	// Unit::Enum DeviceOutputUnit;				
	// };				
	// Output_GetDeviceOutputName GetDeviceOutputName(				
	// const String & DeviceName,				
	// const unsigned int DeviceOutputIndex ) const;				
	ViconDataStreamSDK::CPP::Client MyClient;				
	MyClient.Connect( "localhost" );				
	MyClient.EnableDeviceData();				
	MyClient.GetFrame();				
	Output_GetDeviceOutputName Output =				

	MyClient.GetDeviceOutputName( "AMTI", 0 );
	// Output.Result == Success
	// Output.DeviceOutputName == "Fx"
	// Output.DeviceOutputUnit == Newton
MATLAB	A valid Device Output Index is between 1 and GetDeviceOutputCount()
	% [Output] = GetDeviceOutputName( DeviceName, DeviceOutputIndex )
	MyClient = Client();
	MyClient.Connect( "localhost" );
	MyClient.EnableDeviceData();
	MyClient.GetFrame();
	Output = MyClient.GetDeviceOutputName( "AMTI", 0 );
	% Output.Result == Success
	% Output.DeviceOutputName == "Fx"
	% Output.DeviceOutputUnit == Newton
.NET	A valid Device Output Index is between 0 and GetDeviceOutputCount()-1
	// public class Output_GetDeviceOutputName
	// {
	// public Result Result;
	// public string DeviceOutputName;
	// public Unit DeviceOutputUnit;
	// };
	// Output_GetDeviceOutputName GetDeviceOutputName(
	// string DeviceName,
	// uint DeviceOutputIndex );
	ViconDataStreamSDK.DotNET.Client MyClient =
	new ViconDataStreamSDK.DotNET.Client();
	MyClient.Connect( "localhost" );
	MyClient.EnableDeviceData();
	MyClient.GetFrame();
	Output_GetDeviceOutputName Output =
	MyClient.GetDeviceOutputName( "AMTI", 0 );
	// Output.Result == Success
	// Output.DeviceOutputName == "Fx"
	// Output.DeviceOutputUnit == Newton

# GetDeviceOutputValue

Return the value of a device output. If there are multiple samples for a frame, then the first sample is returned.

See Also: GetDeviceCount, GetDeviceOutputCount, GetDeviceOutputName

See Also : Ge	tDeviceCount, GetDevi	ceOutputCount, GetDev	iceOutputName	
Input	Device Name	string	The device name	
	Device Output Name	string	The name of the device output.	
Output	Result	Result	Result.Success Result.NotConnected Result.NoFrame Result.InvalidDeviceName Result.InvalidDeviceOutputName	
	Value	double	The value of the device output	
	Occluded	boolean	True if the value was present at this frame. If false, then Value will be 0.	
C++	// class Output_GetDeviceOutputValue // { // public: // Result::Enum Result; // double Value; // bool Occluded; // }; // // Output_GetDeviceOutputValue // GetDeviceOutputValue // const String & DeviceName, // const String & DeviceOutputName ) const;  ViconDataStreamSDK::CPP::Client MyClient; MyClient.Connect( "localhost" ); MyClient.EnableDeviceData(); MyClient.GetFrame();  Output_GetDeviceOutputValue Output = MyClient.GetDeviceOutputValue ("AMTI", "Fx" ); // Output.Result == Success // Output.Value == ?			
MATLAB	// Output.Occluded = ?  // [Output] = GetDeviceOutputValue( DeviceName, DeviceOutputName )			
	MyClient = Client(); MyClient.Connect( "localhost" ); MyClient.EnableDeviceData(); MyClient.GetFrame();			
	Output = MyClient.GetDeviceOutputValue( "AMTI", "Fx" );  // Output.Result == Success  // Output.Value == ?  // Output.Occluded = ?			

```
.NET
                   // public class Output_GetDeviceOutputValue
                   // {
                  // public Result Result;
                   // public double Value;
                  // public bool Occluded;
                  // };
                   //
                   // Output_GetDeviceOutputValue
                   // GetDeviceOutputValue( string DeviceName,
                                   string DeviceOutputName );
                   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 = ?
```

### GetForcePlateCount Return the number of ForcePlates available in the DataStream. See Also: GetGlobalForceVector, GetGlobalMomentVector, GetGlobalCentreOfPressure Input Output Result Result Result.Success Result.NotConnected Result.NoFrame Force Plate The number of force plates unsigned integer Count C++ // class Output\_GetForcePlateCount // { // public: // Result::Enum Result; // unsigned int ForcePlateCount; // }; // Output\_GetForcePlateCount GetForcePlateCount() const; ViconDataStreamSDK::CPP::Client MyClient; MyClient.EnableDeviceData(); MyClient.Connect( "localhost" ); MyClient.GetFrame(); Output\_GetDeviceCount Output = MyClient. GetForcePlateCount (); // Output.Result == Success // Output. ForcePlateCount >= 0 **MATLAB** % [Output] = GetForcePlateCount() MyClient = Client(); MyClient.EnableDeviceData(); MyClient.Connect( "localhost" ); MyClient.GetFrame(); Output = MyClient.GetForcePlateCount(); // Output.Result == Success // Output.ForcePlateCount >= 0 .NET // public class Output\_GetForcePlateCount // { // public Result Result; // public uint ForcePlateCount; // }; // Output\_GetForcePlateCount GetForcePlateCount(); ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client(); MyClient.EnableDeviceData(); MyClient.Connect( "localhost" ); MyClient.GetFrame(); Output\_GetForcePlateCount Output = MyClient.GetForcePlateCount(); // Output.Result == Success

// Output Force Plate Count > _ 0
// Output.ForcePlateCount >= 0

### **GetGlobalForceVector**

Return the force vector for the plate in global co-ordinates.

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 plate, not the reaction force.

See Also: GetGlobalMomentVector, GetGlobalCentreOfPressure

		unsigned integer	The index of the plate		
Input	Force Plate Index	unsigned integer	The index of the plate		
Output	Result	Result	Result.Success		
			Result.NotConnected		
			Result.NoFrame		
	- ,, ,		Result.InvalidIndex		
	ForceVector	double[3]	The force on the plate		
C++	A valid ForcePlatel	A valid ForcePlateIndex is between 0 and GetForcePlateCount()-1			
		// class Output_GetGlobalForceVector			
	· ·	<b>//</b> {			
	// public:				
	// Result::Enum Result	•			
	// double ForceVed	ctor[ 3 ];			
	// };				
	//				
	// Output_GetGlobalFor				
	// GetGlobalForceVect				
	// const unsigned int ForcePlateIndex ) const;				
	ViconDataStreamSDK::CPP::Client MyClient;				
		MyClient.Connect( "localhost" );			
MyClient.EnableUnlabeledMarkerData();					
	MyClient.GetFrame();				
	Output_GetGlobalForceVector Output = MyClient.GetGlobalForceVector( 0 );				
MATLAB	A valid ForcePlateIndex is between 1 and GetForcePlateCount()				
	% [Output] = GetGlobalForceVector( ForcePlateIndex )				
	MyClient = Client();				
MyClient.Connect( "localhost" );					
	MyClient.EnableUnlabe	MyClient.EnableUnlabeledMarkerData();			
	MyClient.GetFrame();				
	Output = MyClient. Get				
.NET	A valid ForcePlateIndex is between 0 and GetForcePlateCount() - 1				
	// public class Output_ GetGlobalForceVector				
	// {				
	// public Result Result;				
// public double[] ForceVector; // };					
	H				
	// Output_GetGlobalFor	// Output_GetGlobalForceVector			

// GetGlobalForceVector( uint ForcePlateIndex ) const;

ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client();

MyClient.Connect( "localhost" );

MyClient.EnableUnlabeledMarkerData();

MyClient.GetFrame();

Output\_ GetGlobalForceVector Output = MyClient. GetGlobalForceVector( 0 );

## **GetGlobalMomentVector**

Return the moment vector for the plate in global co-ordinates.

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 plate, not the reaction moment.

See Also: GetGlobalForceVector, GetGlobalCentreOfPressure

_	Dieta Index		The index of the force of the		
Input	Plate Index	unsigned integer	The index of the force plate		
Output	Result	Result	Result.Success		
			Result.NotConnected		
			Result.NoFrame		
			Result.InvalidIndex		
	MomentVector	double[3]	The moment exterted on the		
			plate		
C++	A valid ForcePlate	Index is between 0 and G	etForcePlateCount()-1		
			V		
	// class Output_GetGlo	balMomentVector			
	// {				
	// public:				
	// Result::Enum Resul	<del>t·</del>			
		, /ector[ 3 ];			
	// };	, coto, [ o ],			
	// Output_GetGlobalMo	mont\/octor			
	// GetGlobalMomentV				
		,			
	// const unsigned int ForcePlateIndex ) const;				
	Wasan Data Olympia O DD O O'r at McO'r at				
	ViconDataStreamSDK::CPP::Client MyClient;				
	MyClient.Connect( "localhost" );				
	MyClient.EnableUnlabeledMarkerData();				
	MyClient.GetFrame();				
	Output_GetGlobalMomentVector Output = MyClient.GetGlobalMomentVector( 0 );				
MATLAB					
	A valid 1 order laterindex is between 1 and Geti order late-outhit()				
	% [Output] = GetGlobalMomentVector( ForcePlateIndex )				
	// [Output] = Octolobalivion(voctor( voctor laterilatox)				
	MyClient = Client();	MyClient = Client():			
		MyClient.Connect( "localhost" );			
		MyClient.EnableUnlabeledMarkerData();			
MyClient.GetFrame();					
	wyononi. Oeti raine(),	myonoria. God ramo(),			
	Output = MvClient Get	GlobalMomentVector( 1 );			
.NET			etForcePlateCount() - 1		
	7. 74.14 1 01001 1410	A valid ForcePlateIndex is between 0 and GetForcePlateCount() - 1			
	// nublic class Output	// public class Output_ GetGlobalMomentVector			
	// public class Output_				
		l <del>t</del> ·			
	•	// public Result Result; // public double[] MomentVector;			
	·	aoubie[] iviomentVector;			
	//				
	// Output_GetGlobalMomentVector				

## **GetGlobalCentreOfPressure**

Return the centre of pressure for the plate in global co-ordinates.

The vector is in Meters and is with respect to the global coordinate system regardless of the orientation of the plate.

See Also: GetGlobalForceVector, GetGlobalMomentVector

	Plate Index		The index of the force plate		
Input	Plate Index	unsigned integer	The index of the force plate		
Output	Result	Result	Result.Success		
			Result.NotConnected		
			Result.NoFrame		
			Result.InvalidIndex		
	CentreOfPressur	double[3]	The CoP.		
	е				
C++	A valid ForcePlatel	ndex is between 0 and G	SetForcePlateCount()-1		
	-	// class Output_GetGlobalCentreOfPressure			
	// {				
	// public:				
	// Result::Enum Result	•			
	// double CentreOf	Pressure[ 3 ];			
	// };				
	//				
	// Output_GetGlobalCer	// Output_GetGlobalCentreOfPressure			
	// GetGlobalCentreOfPressure (				
	// const unsigned int ForcePlateIndex ) const;				
	ViconDataStreamSDK::CPP::Client MyClient;				
	MyClient.Connect( "localhost" );				
	MyClient.EnableUnlabeledMarkerData();				
	MyClient.GetFrame();				
	Output_GetGlobalCentreOfPressure Output = MyClient.GetGlobalCentreOfPressure( 0 );				
MATLAB	A valid ForcePlateIndex is between 1 and GetForcePlateCount()				
	% [Output] = GetGlobalCentreOfPressure( ForcePlateIndex )				
	MyClient = Client();				
	MyClient.Connect( "localhost" );				
	MyClient.EnableUnlabeledMarkerData();				
	MyClient.GetFrame();				
	Output = MyClient. GetGlobalCentreOfPressure( 1 );				
.NET	A valid ForcePlateIndex is between 0 and GetForcePlateCount() - 1				
	// LIS 1 0 1 1 0 10 11 10 1 000				
	// public class Output_ GetGlobalCentreOfPressure				
	// {				
	// public Result Result;				
	// public double[] CentreOfPressure;				
	// };				
	<i>II</i>				
	// Output_GetGlobalCentreOfPressure				
	// GetGlobalCentreOfPressure( uint ForcePlateIndex ) const;				

## Appendix A – What's New

#### What's New in Version 1.0

- 1. Full access to analog device data in Nexus. This can be scaled data or raw voltages.
- 2. One SDK for all applications.
- Four segment rotation options: Quaternion, 3x3 row-major Matrix, Helical, and EulerXYZ format.
- 4. Support streaming, request, and pre-fetch modes.
- 5. Formats specific to C++, MATLAB and .NET.
- 6. Version control.
- 7. Result feedback for success criteria.

#### What's New in Version 1.0.1

- 1. C++ programs that access the DS-SDK dll files can now be complied in Debug mode.
- 2. New function calls for Vicon Tracker \*\*\*
  - i. ConnectToMulticast
  - ii. StartTransmittingMulticast
  - iii. StopTransmittingMulticast
  - iv. GetLatencyTotal
  - v. GetLatencySampleCount
  - vi. GetLatencySampleName
  - vii. GetLatencySampleValue

#### What's New in Version 1.1.0

- 1. Release of C++ and .NET SDKs on Windows x64.
- 2. Release of C++ SDK on Linux x86.
- 3. New function calls
  - i. DisableSegmentData
  - ii. DisableMarkerData
  - iii. DisableUnlabeledMarkerData
  - iv. DisableDeviceData
  - v. GetMarkerParentName
  - vi. GetSubjectRootSegmentName
  - vii. GetSegmentParentName
  - viii. GetSegmentChildCount
  - ix. GetSegmentChildName
  - x. GetSegmentStaticTranslation
  - xi. GetSegmentStaticRotationHelical
  - xii. GetSegmentStaticRotationMatrix
  - xiii. GetSegmentStaticRotationQuaternion
  - xiv. GetSegmentStaticRotationEulerXYZ
- 4. Corrected some units. The values given by the SDK have not changed they were incorrectly labeled in previous versions.
  - i. "NewtonMillimetre" has become "NewtonMeter"
  - ii. "Millimetre" has become "Meter"

<sup>\*\*\*</sup> These functions will not work with Vicon Nexus 1.4 and Vicon Blade 1.6.

- 5. Corrected segment rotations following calls to SetAxisMapping()6. Added command-line options for the Test programs to specify a host to connect to.

### What's New in Version 1.2.0

- 1. Added C++ Linux x64 support
- 2. Fix to support of .NET under Windows x64
- 3. New function calls:
  - GetForcePlateCount i.
  - ii. GetGlobalForceVector
  - GetGlobalMomentVector iii.
  - iv. GetGlobalCentreOfPressure