

Vicon DataStream SDK 1.1.0

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

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:
 - "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:
 - "ViconDataStreamSDK DotNET.dll"
 - "ViconDataStreamSDK_CPP.dll"
 - o "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 1.1.0

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

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 4.7, CentOS 5.3, Ubuntu 8.04, Ubuntu 9.04, Fedora 9, and Fedora 11. It should also work on any platform supporting glibc 2.3.4 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:

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



List of all SDK Functions

Contents

Construction and Destruction	6
Result	
GetVersion	10
Connect	11
ConnectToMulticast	12
Disconnect	13
IsConnected	14
StartTransmittingMulticast	15
StopTransmittingMulticast	
EnableSegmentData	17
EnableMarkerData	18
EnableUnlabeledMarkerData	
EnableDeviceData	20
DisableSegmentData	21
DisableMarkerData	22
DisableUnlabeledMarkerData	23
DisableDeviceData	24
IsSegmentDataEnabled	25
IsMarkerDataEnabled	26
IsUnlabeledMarkerDataEnabled	27
IsDeviceDataEnabled	28
SetStreamMode	29
SetAxisMapping	
GetAxisMapping	33
GetFrameGetFrame	34
GetFrameNumber	35
GetLatencyTotal	36
GetLatencySampleCount	37
GetLatencySampleName	38
GetLatencySampleValue	40
GetTimecode	41
GetSubjectCount	43
GetSubjectName	45
GetSubjectRootSegmentName	47
GetSegmentCount	49
GetSegmentName	51
GetSegmentParentName	54
GetSegmentChildCount	56
GetSegmentChildName	58
GetSegmentStaticTranslation	61
GetSegmentStaticRotationHelical	63
GetSegmentStaticRotationMatrix	65
GetSegmentStaticRotationQuaternion	67
GetSegmentStaticRotationEulerXYZ	69
GetSegmentGlobalTranslation	71
GetSegmentGlobalRotationHelical	73
GetSegmentGlobalRotationMatrix	75



GetSegmentGlobalRotationQuaternion	
GetSegmentGlobalRotationEulerXYZ	79
GetSegmentLocalTranslation	81
GetSegmentLocalRotationHelical	83
GetSegmentLocalRotationMatrix	85
GetSegmentLocalRotationQuaternion	87
GetSegmentLocalRotationEulerXYZ	89
GetMarkerCount	
GetMarkerName	
GetMarkerParentName	96
GetMarkerGlobalTranslation	98
GetUnlabeledMarkerCount	100
GetUnlabeledMarkerGlobalTranslation	102
GetDeviceCount	104
GetDeviceName	
GetDeviceOutputCount	109
GetDeviceOutputName	111
GetDeviceOutputValue	113

Construction and Destruction

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

```
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.
                ViconDataStreamSDK.DotNET.Client pHeapClient =
```

new ViconDataStreamSDK.DotNET.Client();

Output SomeFunction Output = pHeapClient.SomeFunction(InputParam);

// Signal to the garbage collector that it can clean up

pHeapClient.Disconnect(); pHeapClient = null;

The Result co	de 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 ViconDataStream { namespace CPP	

```
namespace Result
                enum Enum
                  Unknown,
                  NotImplemented,
                  Success,
                  InvalidHostName,
                  InvalidMulticastIP,
                  ClientAlreadyConnected,
                  ClientConnectionFailed,
                  ServerAlreadyTransmittingMulticast,
                  ServerNotTransmittingMulticast,
                  NotConnected,
                  NoFrame,
                  InvalidIndex,
                  InvalidSubjectName,
                  InvalidSegmentName,
                  InvalidMarkerName,
                  InvalidDeviceName,
                  InvalidDeviceOutputName,
                  InvalidLatencySampleName,
                  CoLinearAxes,
                  LeftHandedAxes
                 };
MATLAB
               classdef Result
                 properties (Constant = true)
                                                    = 0;
                  Unknown
                  NotImplemented
                                                   = 1;
                                                    = 2;
                  Success
                  InvalidHostName
                                                    = 3;
                  InvalidMulticastIP
                                                   = 4;
                  ClientAlreadyConnected
                                                   = 6;
                                                   = 7;
                  ClientConnectionFailed
                  ServerAlreadyTransmittingMulticast = 8;
                  ServerNotTransmittingMulticast = 9;
                  NotConnected
                                                    = 10;
                  NoFrame
                                                    = 11;
                  InvalidIndex
                                                    = 12;
                  InvalidSubjectName
                                                   = 13;
                                                   = 14;
                  InvalidSegmentName
                   InvalidMarkerName
                                                    = 15;
                  InvalidDeviceName
                                                    = 16;
                  InvalidDeviceOutputName
                                                   = 17;
                  InvalidLatencySampleName
                                                   = 18;
                  CoLinearAxes
                                                    = 19;
                                                    = 20;
                  LeftHandedAxes
                 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,
                   {\tt ServerAlreadyTransmittingMulticast,}
                   ServerNotTransmittingMulticast,
                   NotConnected,
                   NoFrame,
                   InvalidIndex,
                   InvalidSubjectName,
                   InvalidSegmentName,
                   InvalidMarkerName,
                   InvalidDeviceName,
                   InvalidDeviceOutputName,
                   InvalidLatencySampleName,
                   CoLinearAxes,
                   LeftHandedAxes
                 };
               } // End of namespace DotNET
               } // End of namespace ViconDataStreamSDK
```

0.4\/	-:			
GetVer				
	on of the Vicon Data	Stream SDK	1	
Input Output	Major	unsigned int	The major version number.	
Cutput	Major	and grida int	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++	<pre>// { // public: // unsigne // unsigne // unsigne // }; // // Output_Ge ViconDataStr</pre>	<pre>// public: // unsigned int Major; // unsigned int Minor; // unsigned int Point; // };</pre>		
MATLAB		rsion Output = MyClient.GetV	Version();	
IVIATLAD	MyClient = C	<pre>% [Output] = GetVersion() MyClient = Client(); Output = MyClient.GetVersion();</pre>		
.NET	// { // public // public // public // public // }; // // Output_Ge	uint Minor;	nt =	
	. 100115464561	=	ataStreamSDK.DotNET.Client();	
	Output GetVe	rsion Output = MyClient.GetV		

Connect Establish a dedicated connection to a Vicon DataStream Server See Also: ConnectToMulticast, Disconnect, 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 Result.ClientConnectionFailed C++ // class Output Connect // { // public: // Result::Enum Result; // }; 11 // Output Connect Connect(const String & HostName); ViconDataStreamSDK::CPP::Client MyClient; Output Connect Output = MyClient.Connect("localhost:801"); MATLAB % [Output] = Connect() MyClient = Client(); Output = MyClient.Connect('locahost:801'); .NET // class Output Connect // { // public Result Result; // }; // // Output Connect Connect(string HostName); ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client(); Output Connect Output = MyClient.Connect("localhost:801");

ConnectToMulticast

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

See Also : Co	onnect, Disconnect, IsC	onnected, Sta	rtTransmittingMulticast, StopTransmittingMulticast	
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"	
	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"	
Output	Result	Result	Result.Success Result.InvalidHostName Result.InvalidMulticastIP Result.ClientAlreadyConnected Result.ClientConnectionFailed	
C++	<pre>// { // public: // Result::En // }; // // Output_Conne // ConnectToMu // ViconDataStream Output_ConnectT</pre>	<pre>// class Output_ConnectToMulticast // { // public: // Result::Enum Result; // }; // // Output_ConnectToMulticast // ConnectToMulticast (const String & HostName,</pre>		
MATLAB	% [Output] = Co. MyClient = Clie	<pre>MyClient.ConnectToMulticast("localhost", "224.0.0.0"); % [Output] = ConnectToMulticast() MyClient = Client(); Output = MyClient.ConnectToMulticast('locahost', '224.0.0.0');</pre>		
.NET	// class Output // { // public Res	<pre>// class Output_ConnectToMulticast // { // public Result Result; // }; // Output_ConnectToMulticast ConnectToMulticast (string HostName,</pre>		
	ViconDataStream	SDK.DotNET.	Client MyClient =	
	Output Connect	<pre>new ViconDataStreamSDK.DotNET.Client(); Output ConnectToMulticast Output =</pre>		
			ast("localhost", "224.0.0.0");	

Disconnect

Disconnect from the Vicon DataStream Server.

Input					
Output	Result	Result	Result.Success Result.NotConnected		
C++	// class Out	out_Disconnect			
	// {				
	// public:				
	// Result	// Result::Enum Result;			
	// };				
	//				
	// Output_Dis	sconnect Disconnect();			
	ViconDataStreamSDK::CPP::Client MyClient;				
	MyClient.Con	nect("localhost");			
	Output Discor	nnect Output = MyClient.Di	sconnect();		
MATLAB	% [Output] =	Connect()			
	MyClient = C	lient();			
	<pre>MyClient.Connect("localhost");</pre>				
	Output = MyC	ient.Disconnect();			
.NET	// public cla	ass Output_Disconnect			
	// {				
	// public Result;				
	// };	// };			
	//				
	// Output_Dis	// Output_Disconnect Disconnect()			
	ViconDataStre	eamSDK.DotNET.Client MyCli	ent =		
		<pre>new ViconDataStreamSDK.DotNET.Client();</pre>			
	MyClient.Con	<pre>MyClient.Connect("localhost");</pre>			
	Output Discor	nnect Output = MyClient.Di	sconnect();		

IsConnected

Discover whether client is connected to the Vicon DataStream Server.

	nnect, Disconnect					
Input						
Output	Connected	boolean	True if you are connected to the stream, otherwise false.			
C++	// class Output	IsConnected	· ·			
	// {					
	// public:					
	// bool Connec	ted;				
	// };					
	//					
	// Output_IsConn	ected IsConnected()	const;			
	ViconDataStreamS	ViconDataStreamSDK::CPP::Client MyClient;				
	Output_IsConnect	ed Output = MyClient	.IsConnected()			
			<pre>// Output.Connected == false</pre>			
	MyClient.Connect	<pre>MyClient.Connect("localhost");</pre>				
	Output_IsConnect	Output_IsConnected Output = MyClient.IsConnected()				
			<pre>// Output.Connected == true</pre>			
			<pre>// (assuming localhost is serving)</pre>			
MATLAB	% [Output] = IsC	onnected()				
	MyClient = Clien	t();				
	Output = MyClien MyClient.Connect		<pre>// Output.Connected == false</pre>			
	Output = MyClien		// Output.Connected == true			
	1.7.2.2		<pre>// (assuming localhost is serving)</pre>			
.NET	// public class	Output IsConnected	37			
	// {	- =				
	// public bool	Connected;				
	// };					
	//					
	// Output_IsConn	// Output_IsConnected IsConnected();				
	ViconDataStreamSDK.DotNET.Client MyClient =					
	<pre>new ViconDataStreamSDK.DotNET.Client();</pre>					
	Output_IsConnect	Output_IsConnected Output = MyClient.IsConnected()				
			<pre>// Output.Connected == false</pre>			
	MyClient.Connect	<pre>MyClient.Connect("localhost");</pre>				
	Output_IsConnect	Output IsConnected Output = MyClient.IsConnected()				
			<pre>// Output.Connected == true</pre>			
			<pre>// (assuming localhost is serving)</pre>			

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.

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

Innut	ServerIP		The IP Address on the server that the		
Input	Servenia	string			
			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	StartTransmittir	ngMulticast		
	// {	_			
	// public:				
	// Result::En	um Result;			
	// };				
	//				
		// // Output StartTransmittingMulticast			
		// StartTransmittingMulticast (const String & ServerIP,			
		// const String & MulticastIP) const;			
		const string a marticastri / const,			
	ViconDataStream	ViconDataStreamSDK::CPP::CPP::Client MyClient;			
		MyClient.Connect("localhost");			
	=		east("10.0.0.1", "224.0.0.0");		
MATLAB					
	o [odopdo]	<pre>% [Output] = StartTransmittingMulticast ()</pre>			
	MyClient = Clie	nt.();			
	-	t("localhost");			
	-		cast('10.0.0.1', '224.0.0.0');		
.NET					
	// {	<pre>// public class Output_StartTransmittingMulticast // {</pre>			
	*	// public Result Result;			
	// };	_			
	//				
		// // Output StartTransmittingMulticast			
	_	// Output_startframsmittingMulticast // StartTransmittingMulticast(string ServerIP, string MulticastIP);			
	 ViconDataStream	ViconDataStreamSDK.DotNET.Client MyClient =			
	<pre>new ViconDataStreamSDK.DotNET.Client();</pre>				
	<pre>MyClient.Connect("localhost");</pre>				
		<pre>MyClient.StartTransmittingMulticast("10.0.0.1", "224.0.0.0");</pre>			

${\bf Stop Transmitting Multicast}$

Ask the DataStream Server to stop transmitting the data you are receiving directly to a Multicast address as well. You must previously have started a transmission via StartTransmittingMulticast.

See Also: Connect, ConnectToMulticast, Disconnect, StartTransmittingMulticast

Input	Tirrect, Cormect romanica	,			
Output	Result	Result	Result.Success		
Output	rtoodit	rtoodit	Result.NotConnected		
			Result.ServerNotTransmittingMulticast		
C++	// alaga Output C	t andranamittinaN			
Стт	_	// class Output_StopTransmittingMulticast			
	<pre>// { // public:</pre>				
	// Result::Enum	Result;			
	// };				
	//	11			
	// Output_StopTra				
	// StopTransmitt	ingMulticast ()	const;		
	ViconDataStreamSD	K::CPP::CPP::Cli	ent MyClient;		
	MyClient.Connect("localhost");			
	MyClient.StartTra	nsmittingMultica	st("10.0.0.1", "224.0.0.0");		
	// Do some stuff				
	MyClient.StopTran	smittingMulticas	t();		
MATLAB	% [Output] = Stop	TransmittingMult	icast ()		
	MyClient = Client	<pre>MyClient = Client();</pre>			
	MyClient.Connect("localhost");			
	MyClient.StartTra	nsmittingMultica	st('10.0.0.1', '224.0.0.0');		
	% Do some stuff				
	MyClient.StopTran	smittingMulticas	t();		
.NET	// public class 0	output_StopTransm	ittingMulticast		
	// {				
	// public Result	Result;			
	// };				
	//				
	// Output StopTra	nsmittingMultica	st		
	// StopTransmitt	ingMulticast();			
	ViconDataStreamSD	K.DotNET.Client	MyClient =		
			<pre>ViconDataStreamSDK.DotNET.Client();</pre>		
	MyClient.Connect("localhost");			
	MyClient.StartTra	nsmittingMultica	st("10.0.0.1", "224.0.0.0");		
	// Do some stuff				
	MyClient.StopTran	smittingMulticas	t();		

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++	<pre>// class Output_EnableSegmentData // { // public:</pre>				
	// Result::	Inum Result;			
	// };				
	//				
	// Output_Enabl	eSegmentData EnableSeg	gmentData();		
		nSDK::CPP::Client MyCli	ient;		
	-	ct("localhost");			
	_		Client.EnableSegmentData();		
MATLAB	% [Output] = Er	% [Output] = EnableSegmentData()			
	-	<pre>MyClient = Client();</pre>			
	<pre>MyClient.Connect("localhost");</pre>				
.NET		<pre>Output = MyClient.EnableSegmentData();</pre>			
.INE I	_	S Output_EnableSegmentI	Data		
	// {				
	=	sult Result;			
	// }; //				
	* *	eSegmentData EnableSec	rmon+Do+o().		
	// Output_Enab	esegmentibata Enablesec	gmenchata();		
	ViconDataStream	nSDK.DotNET.Client MyCl	lient =		
		new Vico	onDataStreamSDK.DotNET.Client();		
	MyClient.Connec	ct("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					
Output	Result	Result	Result.NotConnected Result.Success		
C++	// class Outp	ıt EnableMarkerData	110000000000000000000000000000000000000		
	// {	_			
	// public:				
	// Result:	:Enum Result;			
	// };				
	//				
	// Output_Enal	oleMarkerData EnableMaı	rkerData();		
		amSDK::CPP::Client MyCl	lient;		
	_	ect("localhost");			
144TI 45			Client.EnableMarkerData();		
MATLAB	% [Output] = EnableMarkerData()				
	<pre>MyClient = Client();</pre>				
	MyClient.Connect("localhost");				
.NET	Output = MyClient.EnableMarkerData(); // public class Output EnableMarkerData				
	// public clas	ss Output_mableMarker	oala		
	// public Re	esult Result:			
	// };	obaro nobaro,			
	//				
	// Output Enal	// Output EnableMarkerData EnableMarkerData();			
	- =	,, saspas			
	ViconDataStre	amSDK.DotNET.Client MyC	Client =		
		new Vic	conDataStreamSDK.DotNET.Client();		
	<pre>MyClient.Connect("localhost");</pre>				
	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						
Output	Result	Result	Result.NotConnected			
•			Result.Success			
C++	// class Out	// class Output EnableUnlabeledMarkerData				
	// {	_				
	// public:	// public:				
	// Result	::Enum Result;				
	// };					
	//					
	// Output_En	// Output_EnableUnlabeledMarkerData EnableUnlabeledMarkerData();				
	ViconDataStr	ViconDataStreamSDK::CPP::Client MyClient;				
	MyClient.Connect("localhost");					
	Output EnableUnlabeledMarkerData Output =					
	MyClient.EnableUnlabeledMarkerData();					
MATLAB	% [Output] = EnableUnlabeledMarkerData()					
	· · · · · · · · · · · · · · · · · · ·					
	<pre>MyClient = Client();</pre>					
	MyClient.Con	<pre>MyClient.Connect("localhost");</pre>				
	Output = MyClient.EnableUnlabeledMarkerData();					
.NET	// public class Output_EnableUnlabeledMarkerData					
	<u></u>					
	// public Result;					
	// };	// };				
	//	//				
	// Output_EnableUnlabeledMarkerData EnableUnlabeledMarkerData();					
	ViconDataStr	ViconDataStreamSDK.DotNET.Client MyClient =				
		new Vic	onDataStreamSDK.DotNET.Client();			
	MyClient.Con	nect("localhost");				
	Output_Enable	Output EnableUnlabeledMarkerData Output =				
		Му	Client.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 EnableDeviceData				
	// {				
	// public:				
	// Result::Enum	n Result;			
	// };				
	//				
	// Output_EnableDe	eviceData EnableDeviceData();			
		<pre>X::CPP::Client MyClient;</pre>			
	MyClient.Connect(
MATLAB	Output EnableDeviceData Output = MyClient.EnableDeviceData();		eDeviceData();		
IVIATLAD	% [Output] = EnableDeviceData()				
	<pre>MyClient = Client();</pre>				
	-				
	<pre>MyClient.Connect("localhost"); Output = MyClient.EnableDeviceData();</pre>				
.NET	// public class Output EnableDeviceData				
	// {				
	// public Result;				
	// };				
	11				
	// Output_EnableDeviceData EnableDeviceData();				
	ViconDataStreamSDF	<pre>C.DotNET.Client MyClient =</pre>			
	<pre>new ViconDataStreamSDK.DotNET.Client();</pre>				
	<pre>MyClient.Connect("localhost");</pre>				
	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_DisableSegmentData				
	// {				
	// public:				
	// Result::Enu	m Result;			
	// };				
	//				
	// Output_DisableSegmentData DisableSegmentData();				
	ViconDataStreamSDK::CPP::Client MyClient;				
	MyClient.Connect("localhost");				
MATLAB	Output DisableSegmentData Output = MyClient.DisableSegmentData();				
WATEAD	% [Output] = DisableSegmentData()				
	<pre>MyClient = Client();</pre>				
	<pre>MyClient.Connect("localhost");</pre>				
	Output = MyClient.DisableSegmentData();				
.NET	// public class Output DisableSegmentData				
	// {				
	// public Result;				
	// };				
	//				
	<pre>// Output_DisableSegmentData DisableSegmentData();</pre>				
	ViconDataStreamSDK.DotNET.Client MyClient =				
	<pre>new ViconDataStreamSDK.DotNET.Client();</pre>				
	<pre>MyClient.Connect("localhost");</pre>				
	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 Result.Success		
C++	// class Output_Di	// class Output DisableMarkerData			
	// public:				
	// Result::Enur	// Result::Enum Result;			
	// };				
	//				
	// Output_DisableM	MarkerData DisableMarkerData();			
		K::CPP::Client MyClient;			
	MyClient.Connect(
MATI AD	Output DisableMarkerData Output = MyClient.DisableMarkerData();		oleMarkerData();		
MATLAB	MATLAB % [Output] = DisableMarkerData()				
	Magaliant galiant				
	<pre>MyClient = Client(); MyClient.Connect("localhost");</pre>				
	= ·				
.NET	Output = MyClient.DisableMarkerData(); // public class Output DisableMarkerData				
	// public class output_bisablemarkerbata // {				
	// public Result Result;				
	// public Result Result,				
	// //				
	// Output DisableMarkerData DisableMarkerData();				
	ViconDataStreamSDF	K.DotNET.Client MyClient =			
	<pre>new ViconDataStreamSDK.DotNET.Client();</pre>				
	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			
Output	Result	Result	Result.NotConnected
			Result.Success
C++	// class Output Di		
	// {		
	// public:		
	// Result::Enum	n Result;	
	// };		
	//		
	// Output_Disable	JnlabeledMarkerData DisableUnla	abeledMarkerData();
	ViconDataStreamSDF	K::CPP::Client MyClient;	
	MyClient.Connect("localhost");	
	Output_DisableUnla	abeledMarkerData Output =	
		MyClient.Disak	oleUnlabeledMarkerData();
MATLAB	% [Output] = DisableUnlabeledMarkerData()		
	MyClient = Client	();	
	<pre>MyClient.Connect("localhost");</pre>		
	Output = MyClient.DisableUnlabeledMarkerData();		
.NET	// public class Ou	tput_DisableUnlabeledMarkerDat	za – – – – – – – – – – – – – – – – – – –
	// {		
	// public Result	Result;	
	// };		
	//		
	// Output_Disable(JnlabeledMarkerData DisableUnla	abeledMarkerData();
	ViconDataStreamSDF	<pre>C.DotNET.Client MyClient =</pre>	
			nSDK.DotNET.Client();
	MyClient.Connect(
	Output_DisableUnla	abeledMarkerData Output =	
		MyClient.Disak	oleUnlabeledMarkerData();

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					
Output	Result	Result	Result.NotConnected		
			Result.Success		
C++	// class Outpu	// class Output_DisableDeviceData			
	// {				
	// public:				
	// Result::	// Result::Enum Result;			
	// };	// };			
	//	//			
	// Output_Disa	bleDeviceData Disable	DeviceData();		
	ViconDataStreamSDK::CPP::Client MyClient;				
	<pre>MyClient.Connect("localhost");</pre>				
MAATI AD			/Client.DisableDeviceData();		
MATLAB	% [Output] = DisableDeviceData()				
	<pre>MyClient = Client();</pre>				
	<pre>MyClient.Connect("localhost");</pre>				
.NET	Output = MyClient.DisableDeviceData();				
.INE I	// public class Output_DisableDeviceData				
	// {				
	// public Result;				
	// }; //				
	<pre>// Output_DisableDeviceData DisableDeviceData();</pre>				
	ViconDataStreamSDK.DotNET.Client MyClient =				
	<pre>new ViconDataStreamSDK.DotNET.Client();</pre>				
	<pre>MyClient.Connect("localhost");</pre>				
	Output DisableDeviceData Output = MyClient.DisableDeviceData();				

IsSegmentDataEnabled

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

See Also: EnableSegmentData, DisableSegmentData, IsMarkerDataEnabled. IsUnlabeledMarkerDataEnabled, IsDeviceDataEnabled

Input					
Output	Enabled	boolean	Whether the data is enabled.		
C++	// class Output I	sSegmentDataEnabled			
	// {				
	// public:				
	// bool Enabled	;			
	// };				
	//				
	// Output_IsSegme	ntDataEnabled IsSegmentDataEna	bled() const;		
		K::CPP::Client MyClient;			
	MyClient.Connect(
	Output_IsSegmentD	ataEnabled Output = MyClient.I			
			Output.Enabled == false		
	MyClient.EnableSe	=			
	Output_IsSegmentDataEnabled Output = MyClient.IsSegmentDataEnabled();				
MATLAB			Output.Enabled == true		
MATLAB	% [Output] = IsSe	gmentDataEnabled()			
	24 02 1 02 1				
	<pre>MyClient = Client();</pre>				
	MyClient.Connect("localhost");				
	Output = MyClient.IsSegmentDataEnabled(); % Output.Enabled == false				
	<pre>MyClient.EnableSegmentData(); Output = MyClient.IsSegmentDataEnabled(); % Output.Enabled == true</pre>				
.NET		utput IsSegmentDataEnabled	put.Enabled tide		
	// public class C	dcput_issegmentDataEnabled			
	// t // public bool Enabled;				
	// public bool	Enabled,			
	//				
	// // Output IsSegmentDataEnabled IsSegmentDataEnabled();				
	_				
	ViconDataStreamSDK.DotNET.Client MyClient =				
		new ViconDataStrea	mSDK.DotNET.Client();		
	MyClient.Connect("localhost");			
	Output_IsSegmentD	Output_IsSegmentDataEnabled Output = MyClient.IsSegmentDataEnabled();			
	// Output.Enabled == false				
	<pre>MyClient.EnableSegmentData();</pre>				
	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					
Output	Enabled	boolean	Whether the data is enabled.		
C++	// class Output	IsMarkerDataEnabled			
	// {	-			
	// public:				
	// bool Enable	ed;			
	// };				
	//				
	// Output_IsMark	erDataEnabled IsMarkerDataEnabl	ed() const;		
		DK::CPP::Client MyClient;			
	-	("localhost");			
	Output_IsMarkerI	ataEnabled Output = MyClient.Is			
			Output.Enabled == false		
	MyClient.EnableM				
	Output_IsMarkerD	<pre>Output_IsMarkerDataEnabled Output = MyClient.IsMarkerDataEnabled();</pre>			
144TI 4D			Output.Enabled == true		
MATLAB	% [Output] = IsM	<pre>IarkerDataEnabled()</pre>			
	<pre>MyClient = Client();</pre>				
	<pre>MyClient.Connect("localhost");</pre>				
	Output = MyClient.IsMarkerDataEnabled(); % Output.Enabled == false				
	<pre>MyClient.EnableMarkerData(); Output = MyClient.IsMarkerDataEnabled(); % Output.Enabled == true</pre>				
.NET			ut.Enabled == true		
.INE I	=	Output_IsMarkerDataEnabled			
	// {				
	// public bool Enabled;				
	// };				
	<pre>// Output_IsMarkerDataEnabled IsMarkerDataEnabled();</pre>				
	ViconDataStreamSDK.DotNET.Client MyClient =				
		new ViconDataStrea	mSDK.DotNET.Client();		
	MyClient.Connect	("localhost");			
	Output_IsMarkerD	ataEnabled Output = MyClient.Is	MarkerDataEnabled();		
	// Output.Enabled == false				
	<pre>MyClient.EnableMarkerData();</pre>				
	Output_IsMarkerDataEnabled Output = MyClient.IsMarkerDataEnabled();				
		//	Output.Enabled == true		

IsUnlabeledMarkerDataEnabled

Return whether unlabeled marker data is enabled in the DataStream.

See Also: EnableUnlabeledMarkerData, DisableUnlabeledMarkerData, IsSegmentDataEnabled. IsMarkerDataEnabled, IsDeviceDataEnabled

Input				
Output	Enabled	boolean	Whether the data is enabled.	
C++	// class Output_Is	sUnlabeledMarkerDataEnabled		
	// {			
	// public:			
	// bool Enabled;			
	// };			
	//			
	// Output_IsUnlabe	eledMarkerDataEnabled		
	//	<pre>IsUnlabeledMarkerDataEnabled()</pre>	const;	
	ViconDataStreamSDF	<pre>K::CPP::Client MyClient;</pre>		
	MyClient.Connect("localhost");		
	Output IsUnlabeled	dMarkerDataEnabled Output =		
	MyClient.Ist	JnlabeledMarkerDataEnabled(); /	// Output.Enabled == false	
	MyClient.EnableUnl	abeledMarkerData();		
	Output_IsUnlabeled	dMarkerDataEnabled Output =		
	MyClient.IsU	JnlabeledMarkerDataEnabled(); /	// Output.Enabled == true	
MATLAB	% [Output] = IsUn]	abeledMarkerDataEnabled()		
	<pre>MyClient = Client();</pre>			
	MyClient.Connect("localhost");		
	Output = MyClient.IsUnlabeledMarkerDataEnabled(); % Output.Enabled == f			
	MyClient.EnableUnl	labeledMarkerData();		
	Output = MyClient.	.IsUnlabeledMarkerDataEnabled()	; % Output.Enabled == true	
.NET	// public class Ou	tput_IsUnlabeledMarkerDataEnak	oled	
	// {			
	// public bool H	Enabled;		
	// };			
	//			
	// Output_IsUnlabeledMarkerDataEnabled IsUnlabeledMarkerDataEnabled();			
	ViconDataStreamSDK.DotNET.Client MyClient =			
		new ViconDataStream	nSDK.DotNET.Client();	
	MyClient.Connect("localhost");		
	Output_IsUnlabeled	MarkerDataEnabled Output =		
	MyCl	lient.IsMarkerDataEnabled(); /	Output.Enabled == false	
	MyClient.EnableUnlabeledMarkerData();			
	Output_IsUnlabeledMarkerDataEnabled Output =			
	MyClient.IsUr	nlabeledMarkerDataEnabled(); /,	Output.Enabled == true	

IsDeviceDataEnabled

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

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

Input			
Output	Enabled	boolean	Whether the data is enabled.
C++	// class Output Is	SDeviceDataEnabled	
	// {		
	// public:		
	// bool Enabled;		
	// };		
	//		
	// Output_IsDevice	eDataEnabled IsDeviceDataEnable	ed() const;
	ViconDataStreamSDF	<pre>K::CPP::Client MyClient;</pre>	
	MyClient.Connect("localhost");	
	Output_IsDeviceDat	aEnabled Output = MyClient.Isl	DeviceDataEnabled();
		// (Output.Enabled == false
	MyClient.EnableDev		
	Output_IsDeviceDat	aEnabled Output = MyClient.Isl	DeviceDataEnabled();
=		// (Output.Enabled == true
MATLAB	% [Output] = IsDeviceDataEnabled()		
	MyClient = Client		
	MyClient.Connect(
		<pre>IsDeviceDataEnabled(); % Outpr</pre>	ut.Enabled == false
	MyClient.EnableDev		17 1
NICT	Output = MyClient.IsDeviceDataEnabled(); % Output.Enabled == true		it.Enabled == true
.NET	// public class Output_IsDeviceDataEnabled		
	// {		
	// public bool E	mabled;	
	// }; //		
	· ·	DataEnabled JaponicaDataEnable	ad () •
	// Output_IsDevice	eDataEnabled IsDeviceDataEnable	=4();
	ViconDataStreamSDF	<pre>C.DotNET.Client MyClient =</pre>	
		=	mSDK.DotNET.Client();
	MyClient.Connect(
	_	caEnabled Output = MyClient.Isl	DeviceDataEnabled();
			Output.Enabled == false
	MyClient.EnableDev		-
	_	caEnabled Output = MyClient.Isl	DeviceDataEnabled();
	- =		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

Input	Mode	StreamMode	StreamMode.ServerPush		
			StreamMode.ClientPull		
			StreamMode.ClientPullPreFetch		
Output	Result	Result	Result.Success		
			Result.NotConnected		
C++	// class Output S	// class Output SetStreamMode			
	// {				
	// public:				
	// Result::Enum	Result;			
	// };				
	//				
		amMode SetStreamMod	le(const StreamMode::Enum Mode);		
	, , <u>_</u>	// Output_setstreammode setstreammode(const streammodemmm Mode),			
	ViconDataStreamSD	K::CPP::Client MyCl	ient;		
	MyClient.Connect("localhost");			
	MyClient.SetStrea	mMode(ViconDataStr	reamSDK::CPP::StreamMode::ServerPush);		
	MyClient.SetStrea	mMode(ViconDataStr	reamSDK::CPP::StreamMode::ClientPull);		
	MyClient.SetStrea	mMode(
	ViconDataStreamSD	K::CPP::StreamMode:	:ClientPullPreFetch);		
MATLAB	% [Output] = SetS	treamMode(Mode);			
	MyClient = Client	();			
	MyClient.Connect(`localhost');			
	MyClient SetStrea	MvClient.SetStreamMode(StreamMode.ServerPush):			

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 Outpu	t SetAxisMapping	<u> </u>		
	// {	_			
	// public:				
	// Result::Enum Result;				
	// };				
	//				
	// Output_SetAxisMapping SetAxisMapping(const Direction::Enum XAxis,				
	// const Direction::Enum YAxis,				
	// const Direction::Enum ZAxis)				
	Winner Debeg Channer OPP China M. China				
	<pre>ViconDataStreamSDK::CPP::Client MyClient; MyClient.SetAxisMapping(ViconDataStreamSDK::CPP::Direction::Forward,</pre>				
	ViconDataStreamSDK::CPP::Direction::Left,				
			eamSDK::CPP::Direction::Up);		
MATLAB	% [Output] = SetAxisMapping(XAxis,				
	% YAxis,				
	% ZAxis)				
	<pre>MyClient = Client();</pre>				
	MyClient.SetAxisMapping(Direction.Forward,				
	Direction.Left,				
	Direction.Up);				
.NET	// public clas	s Output_SetAxisMapping			
	// {				
	// public Resu	ılt Result;			
	// };				
	//				
		// Output_SetAxisMapping SetAxisMapping(Direction XAxis,			
	//		Direction YAxis,		
	//		Direction ZAxis);		

ViconDataStreamSDK.DotNET.Client MyClient =
<pre>new ViconDataStreamSDK.DotNET.Client();</pre>
MyClient.SetAxisMapping(ViconDataStreamSDK.DotNET.Direction.Forward,
ViconDataStreamSDK.DotNET.Direction.Left,
<pre>ViconDataStreamSDK.DotNET.Direction.Up);</pre>

Vicon Motion Systems © November 2009 Page 32 of 115

GetAxisMapping Get the current Axis mapping. See Also: SetAxisMapping Input XAxis Output Direction YAxis Direction ZAxis Direction C++ // class Output GetAxisMapping // { // public: // Direction::Enum XAxis; // Direction::Enum YAxis; // Direction::Enum ZAxis; // }; // // Output GetAxisMapping GetAxisMapping() const; ViconDataStreamSDK::CPP::Client MyClient; Output GetAxisMapping Output = MyClient.GetAxisMapping(); // Output.XAxis == ViconDataStreamSDK::CPP::Direction::Forward // Output.YAxis == ViconDataStreamSDK::CPP::Direction::Left // Output.ZAxis == ViconDataStreamSDK::CPP::Direction::Up MATLAB % [Output] = GetAxisMapping() MyClient = Client(); 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 YAxis; // public Direction ZAxis; // }; 11 // 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: SetStreamMode

	tStreamMode					
Input						
Output	Result	Result	Result.Success			
			Result.NotConnected			
C++	// class Output_GetFrame					
	// {	// {				
	// public:					
	// Result::Enum Result;					
	// };					
	//					
	// Output_GetF	<pre>// Output_GetFrame GetFrame();</pre>				
		Wissenberg Character CDD CDI in the Michigan				
		ViconDataStreamSDK::CPP::Client MyClient;				
	_	Output_GetFrame Output;				
		<pre>Output = MyClient.GetFrame(); // Output.Result == NotConnected</pre>				
	=	<pre>MyClient.Connect("localhost");</pre>				
	Output = MyCli	Output = MyClient.GetFrame(); // Output.Result == Success				
MATLAB	% [Output] = GetFrame()					
	MyClient = Cli	<pre>MyClient = Client();</pre>				
	Output = MyClient.GetFrame(); // Output.Result == NotConnected					
	<pre>MyClient.Connect("localhost");</pre>					
	Output = MyClient.GetFrame(); // Output.Result == Success					
.NET	// public class Output_GetFrame					
	// public Result;					
	// };	// };				
	//					
	// Output_GetF	// Output_GetFrame GetFrame();				
	ViconDataStrea:	ViconDataStreamSDK.DotNET.Client MyClient =				
		<pre>new ViconDataStreamSDK.DotNET.Client();</pre>				
	_	Output_GetFrame Output;				
		Output = MyClient.GetFrame(); // Output.Result == NotConnected				
	<pre>MyClient.Connect("localhost");</pre>					
	Output = MyClient.GetFrame(); // Output.Result == Success					

GetFrameNumber Return the number of the last frame retrieved from the DataStream. See Also: GetFrame, GetTimecode Input Result.Success Output Result Result Result.NotConnected Result.NoFrame Frame Number unsigned integer The frame number C++ // class Output GetFrameNumber // { // public: Result::Enum Result; unsigned int FrameNumber; // }; // // Output GetFrameNumber GetFrameNumber() const; ViconDataStreamSDK::CPP::Client MyClient; MyClient.Connect("localhost"); Output GetFrameNumber Output; Output = MyClient.GetFrameNumber(); // Output.Result == NoFrame // Output.FrameNumber == 0 MyClient.GetFrame(); Output = MyClient.GetFrameNumber(); // Output.Result == Success // Output.FrameNumber >= 1 **MATLAB** % [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 GetFrameNumber // { // public Result Result; // public uint FrameNumber; // }; // Output GetFrameNumber GetFrameNumber(); ViconDataStreamSDK.DotNET.Client MyClient = new ViconDataStreamSDK.DotNET.Client(); MyClient.Connect("localhost"); Output GetFrameNumber Output; Output = MyClient.GetFrameNumber(); // Output.Result == NoFrame // Output.FrameNumber == 0 MyClient.GetFrame(); Output = MyClient.GetFrameNumber(); // Output.Result == Success // Output.FrameNumber >= 1

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

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

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

Input	l l	de, detection for the first detections	cySampleName, GetLatencySampleValue	
Output	Result	Result	Result.Success Result.NotConnected Result.NoFrame	
	Count	unsigned int	The number of samples taken.	
C++	<pre>// class Output_GetLatencySampleCount // { // public: // Result::Enum Result; // unsigned int Count; // }; // // Output_GetLatencySampleCount GetLatencySampleCount() const; ViconDataStreamSDK::CPP::Client MyClient; MyClient.Connect("localhost"); MyClient.GetFrame();</pre>			
MATLAB		<pre>Output_GetLatencySampleCount Output = MyClient.GetLatencySampleCount(); % [Output] = GetLatencySampleCount()</pre>		
	<pre>MyClient = Client(); MyClient.Connect('localhost'); MyClient.GetFrame();</pre>			
.NET	<pre>// class Out; // { // public // public // }; // // Output_Ge ViconDataStree</pre>		encySampleCount();	
	<pre>MyClient.Connect("localhost"); MyClient.GetFrame(); Output_GetLatencySampleCount Output = MyClient.GetLatencySampleCount();</pre>			

GetLatencySampleName Return the name of a latency sample. This value can be passed into GetLatencySampleValue(). See Also: GetFrame, GetTimecode, GetLatencyTotal, GetLatencySampleCount, GetLatencySampleValue Input LatencySampleIn Unsigned int The index of the name. dex Result Result.Success Output Result Result.NotConnected Result.NoFrame Result.InvalidIndex The name of the latency Name string 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().

See Also: GetFrame, GetTimecode, GetLatencyTotal, GetLatencySampleCount, GetLatencySampleValue

Input	LatencySampleN	string	The name of the latency sample.		
	ame				
Output	Result	Result	Result.Success		
			Result.NotConnected		
			Result.NoFrame		
			Result.InvalidLatencySampleName		
	Value	double	The duration of the latency in seconds.		
C++	// class Output G	etLatencySampleV	·		
	// {				
	// public:				
	// Result::Enu	m Result:			
	// double	Value;			
	// };	,4140,			
	//				
	// Output GetLate	ncugamnleValue			
	_		String & LatencySampleName) const;		
	// Gethatencysal	mpievaiue(consc	String & Latency Samprename / Const,		
	ViconDataStreamSD	VCDDClion+ M	incliant.		
	MyClient.Connect(yciienc,		
	MyClient.GetFrame				
	Output GetLatency				
	_				
	_		"Data Collected");		
MATLAB		<pre>// Output.Value == 0.1 % [Output] = GetLatencySampleValue()</pre>			
WAILAD	% [Output] - Getta	acencysampievaiu	e ()		
	MyClient = Client	() •			
	MyClient.Connect(
	MyClient.GetFrame				
	-		oWalne (Nata Callestad))		
	% Output.Value ==		eValue('Data Collected');		
.NET			alue.		
.INL I	<pre>// class Output_G // {</pre>	ethatencysampiev	atue		
	// public Resu	1+ Pacul+•			
	// public doub				
	// public doub.	ie vaiue,			
	// };				
		n arrCompleTtolue			
	// Output_GetLate:		LatencySampleName);		
		=			
	ViconDataStreamSD		_		
	MyCliont Comment		<pre>ViconDataStreamSDK.DotNET.Client();</pre>		
	MyClient.Connect(
	MyClient.GetFrame				
	Output_GetLatency	=			
			"Data Collected");		
	// Output.Value ==	// Output.Value == 0.1			

GetTimecode

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

See Also : G	etFrame, GetFrameNumber	•	
Input	,		
Output	Result	Result	Result.Success Result.NotConnected Result.NoFrame
	Hours	Unsigned integer	
	Minutes	Unsigned integer	
	Seconds	Unsigned integer	
	Frames	Unsigned integer	
	SubFrame	Unsigned integer	
	FieldFlag	Boolean	
	Standard	TimecodeStandard	None PAL NTSC NTSCDrop Film
	SubFramesPerFra	Unsigned integer	
	me		
	UserBits	Unsigned integer	
C++	<pre>// unsigned int // unsigned int // }; //</pre>	Result; Hours; Minutes; Seconds; Frames; SubFrame; FieldFlag; ard::Enum Standard; SubFramesPerFrame; UserBits;	
	MyClient.Connect(MyClient.GetFrame()		e();
MATLAB	<pre>% [Output] = GetTin MyClient = Client() MyClient.Connect() MyClient.GetFrame()</pre>	mecode()); "localhost"););	- 1/ /
	Output = MyClient.		
.NET	// class Output Ge	tTimecode	<u></u>

```
// {
// public Result
                         Result;
// public uint
                         Hours;
// public uint
// public uint
                         Minutes;
                         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

Input	etSubjectName				
Output	Result	Result	Result.Success Result.NotConnected Result.NoFrame		
	Subject Count	unsigned integer	The number of subjects		
C++	// { // public: // Result::Enu	<pre>// public: // Result::Enum Result; // unsigned int SubjectCount; // }; //</pre>			
	,, ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	,,			
	ViconDataStreamS	DK::CPP::Client MyClient;			
	MyClient.Connect	("localhost");			
	Output_GetSubjectCount Output;				
	Output = MyClien	Output = MyClient.GetSubjectCount(); // Output.Result == NoFrame // Ooutput.SubjectCount == 0			
	<pre>MyClient.GetFrame();</pre>				
	Output = MyClien	Output = MyClient.GetSubjectCount(); // Output.Result == Success			
		// Output.SubjectCount >= 0			
MATLAB	% [Output] = GetSubjectCount()				
	MyClient = Clien	-()•			
	<pre>MyClient = Client(); MyClient.Connect('localhost');</pre>				
	Output = MyClient.GetSubjectCount(); % Output.Result == NoFrame				
	% Ooutput.SubjectCount == 0				
	MyClient.GetFram		pae.subjecteoune		
	Output = MyClient.GetSubjectCount(); % Output.Result == Success				
	odopao nyorion	% Output.SubjectCount >= 0			
.NET	// class Output_				
	// {	L. B			
	=	// public Result;			
	_	// public uint SubjectCount;			
	// }; //				
		<pre>// // Output GetSubjectCount GetSubjectCount();</pre>			
	//	_			
	// Output_GetSubjectCount GetSubjectCount()				
	ViconDataStreamS	<pre>ViconDataStreamSDK.DotNET.Client MyClient =</pre>			
			<pre>treamSDK.DotNET.Client();</pre>		
	MyClient.Connect				
	Output_GetSubjec	Count Output;			

GetSubjectName

Return the name of a subject. This can be passed into segment and marker functions.

See Also: GetSubjectCount

Input	Subject Index	unsigned integer	The index of the subject.
Output	Result	Result	Result.Success
			Result.NotConnected
			Result.NoFrame
			Result.InvalidIndex
	Subject Name	string	The name of the subject
C++	A valid Subject Ind	ex is between 0 and GetSubject0	Count()-1

```
// class Output_GetSubjectName
// {
   // public:
   // Result::Enum Result;
   // String SubjectName;
   // };
   //
   // Output_GetSubjectName GetSubjectName(
   // const unsigned int SubjectIndex ) const;

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

MATLAB A valid Subject Index is between 1 and GetSubjectCount()

Output GetSubjectCount OutputGSC;

```
% [Output] = GetSubjectName( SubjectIndex )

MyClient = Client;
MyClient.Connect( 'localhost' );
MyClient.GetFrame();

OutputGSC = MyClient.GetSubjectCount(); % OutputGSC.Result == Success
% OutputGSC.SubjectCount == 2
OutputGSN = MyClient.GetSubjectName(1); % OutputGSN.Result == Success
```

```
% OutputGSN.SubjectName == 'Al'
               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.

Input	Subject Name	string	nentChildCount, GetSegmentChildName The name of the subject		
Output	Result	Result	Result.Success		
o anp an		11000.11	Result.NotConnected		
			Result.NoFrame		
			Result.InvalidSubjectName		
	Segment Name	string	The name of the root		
	ocginent Name	Stillig	segment		
C++	// class Output G	etSubjectRootSeamentNam	· ·		
	<pre>// class Output_GetSubjectRootSegmentName // {</pre>				
	// { // public:				
	// Result::Enum	Result:			
	// String	SegmentName;			
	// };	beginerrervame,			
	//				
		ectRootSegmentName GetS	ubjectRootSegmentName(
	// const String & SubjectName) const				
	ViconDataStreamSDK::CPP::Client MyClient;				
	<pre>MyClient.Connect("localhost");</pre>				
	MyClient.EnableSegmentData();				
	<pre>MyClient.GetFrame();</pre>				
	Output_GetSubjectRootSegmentName Output;				
	<pre>Output = MyClient.GetSubjectRootSegmentName("Bob");</pre>				
	// Output.Result == Success				
		// Output.Seg	mentName == "Pelvis"		
MATLAB	% [Output] = GetSubjectRootSegmentName(SubjectName)				
	<pre>MyClient = Client();</pre>				
	<pre>MyClient.Connect("localhost");</pre>				
	<pre>MyClient.EnableSegmentData();</pre>				
	<pre>MyClient.GetFrame();</pre>				
	Outrot - Muchant Cat Culticat Dact Comment Name / NDah//)				
	Output = MyClient.GetSubjectRootSegmentName("Bob");				
	<pre>% Output.Result == Success % Output.SegmentName == "Pelvis"</pre>				
.NET	// mublic class C				
	<pre>// public class Output_GetSubjectRootSegmentName // {</pre>				
	// { // public Result Result;				
	<pre>// public Result; // public string SegmentName;</pre>				
	//	// }; //			
	// // Output GetSubjectRootSegmentName GetSubjectRootSegmentName(
	// Output_GetSubjectRootSegmentName GetSubjectRootSegmentName(// string SubjectName);				
	,, string subjectivance ,,				
	ViconDataStreamSDK.DotNET.Client MyClient =				

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

Input	Subjectivame, GetSegme	string	The name of the subject		
Output	Result	Result	Result.Success		
Output	IXESUIT	Nesuit	Result.NotConnected		
			Result.NoFrame		
			Result.InvalidSubjectName		
	Segment Count	unsigned integer	The number of segments		
C++	// class Output Ge		The number of segments		
011	// {	sesegmenteoune			
	// public:				
	// Result::Enum	Result:			
	// unsigned int				
	// };	oogerroodarro,			
	//				
		entCount GetSegmentCount(
	//	const String & Subject	ctName) const;		
		5	•		
	ViconDataStreamSDF	<pre>K::CPP::Client MyClient;</pre>			
		MyClient.EnableSegmentData();			
	MyClient.Connect(<pre>MyClient.Connect("localhost");</pre>			
		2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,			
	Output_GetSegment(Count Output;			
	Output = MyClient.	.GetSegmentCount("Bob"); /	// Output.Result == NoFrame		
		/	// Output.SegmentCount == 0		
	<pre>MyClient.GetFrame();</pre>				
	Output = MyClient.	.GetSegmentCount("Al"); /	// Output.Result ==		
		/	// InvalidSubjectName		
		/	// Output.SegmentCount == 0		
	Output = MyClient.	.GetSegmentCount("Bob"); /			
			// Output.SegmentCount >= 0		
MATLAB	% [Output] = GetSe	egmentCount(SubjectName)			
	MyClient = Client				
	MyClient.EnableSec				
	MyClient.Connect("localhost");			
	Outrook Magailant	Cat Cannot Canat (NP ala //)	Output Baselle Waller		
	Output = MyClient.	.GetSegmentCount("Bob"); %			
	MyClient.GetFrame		S Output.SegmentCount == 0		
	myclient.Getriame	(),			
	Output = MarCliont	.GetSegmentCount("Al"); %	: Output Result		
	Jucput - MycIIelit.	. Gecseymentcount (AI); 1	InvalidSubjectName		
		7	S Output.SegmentCount == 0		
		3	, output.beginemedount 0		
	Output = MyClient	.GetSegmentCount("Bob"); {	Control Result == Success		
	ouchas Hightelis.	. cccccgmcrrcccarre(bob /,	, cacpac.nebaic baccess		

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

GetSegmentName Return the name of a segment for a specified subject. This can be passed into segment functions. See Also: GetSegmentCount Subject Name The name of the subject Input string Segment Index unsigned integer The index of the segment. Result.Success Output Result Result Result.NotConnected Result.NoFrame Result.InvalidSubjectName Result.InvalidIndex Segment Name The name of the segment string C++ A valid Segment Index is between 0 and GetSegmentCount()-1 // class Output GetSegmentName // { // public: // Result::Enum Result; // String SegmentName; // }; // // Output GetSegmentName GetSegmentName(const String & SubjectName, // const unsigned int SegmentIndex) const ViconDataStreamSDK::CPP::Client MyClient; 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) **MATLAB** A valid Segment Index is between 1 and GetSegmentCount()

```
% [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 Also: GetSegmentCount, GetSegmentChildCount, GetSegmentChildName,
GetSubjectRootSegmentName

Input Subject Name string The

Input	Subject Name	string	The name of the subject	
nip at	Segment Name	string	The name of the segment	
Output	Result	Result	Result.Success	
Output	Nesuit	Nesuit	Result.NotConnected	
			Result.NoFrame	
			Result.InvalidSubjectName	
	0 (1)		Result.InvalidSegmentName	
	Segment Name	string	The name of the parent	
			segment or an empty string if	
			it is the root segment.	
C++	_	etSegmentParentName		
	// {			
	// public:			
	// Result::Enum	Result;		
	// String	SegmentName;		
	// };			
	//			
	// Output_GetSegme	// Output_GetSegmentParentName GetSegmentParentName(
	//	const String & SubjectN	jectName,	
	// const String & SegmentName) const			
	ViconDataStreamSDK::CPP::Client MyClient;			
	<pre>MyClient.Connect("localhost");</pre>			
	<pre>MyClient.EnableSegmentData();</pre>			
	<pre>MyClient.GetFrame();</pre>			
	Output GetSegmentParentName Output;			
	Output = MyClient.GetSegmentParentName("Bob", "Pelvis");			
		// Output.Result ==		
		// Output.SegmentNam		
		// This is the root		
	Output = MvClient	.GetSegmentParentName("Bob",	-	
		// Output.Result ==		
		// Output.SegmentNam		
MATLAB	% [Output] = GetSe	egmentParentName(SubjectName,		
	Mireliant Cli	() .		
	MyClient = Client			
	MyClient.Connect(
	MyClient.EnableSec			
	MyClient.GetFrame	();		
	Output = MyClient	.GetSegmentParentName("Bob",	"Pelvis");	
		% Output.Result == S	uccess	
		% Output.SegmentCoun	t == ""	
		% This is the root s	egment	

```
Output = MyClient.GetSegmentParentName( "Bob", "LFemur");
                                         % Output.Result == Success
                                         % Output.SegmentCount == "Pelvis"
.NET
               // public class Output GetSegmentParentName
               // {
               // public Result Result;
               // public string SegmentName;
               // };
               11
               // 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

_	tSegmentUniidName, Get		The second of the second		
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_Ge	etSegmentChildCount			
	// {				
	// public:				
	// Result::Enum	Result;			
	// unsigned int	SegmentCount;			
	// };				
	//				
	// Output_GetSegme	entChildCount GetSegmentChildC	ount(
	//	const String & SubjectNa	me,		
	//				
	ViconDataStreamSDK::CPP::Client MyClient;				
	<pre>MyClient.EnableSegmentData();</pre>				
	<pre>MyClient.Connect("localhost");</pre>				
	<pre>MyClient.GetFrame();</pre>				
	Output_GetSegmentChildCount Output;				
	Output = MyClient.GetSegmentCount("Bob", "Pelvis");				
			Output.Result == Success		
MATLAD			Output.SegmentCount >= 0		
MATLAB	% [Output] = GetSe	egmentChildCount(SubjectName,	SegmentName)		
	Marchiant - Oliant	() -			
	MyClient = Client				
	MyClient.EnableSegmentData();				
	-	<pre>MyClient.Connect("localhost");</pre>			
	<pre>MyClient.GetFrame();</pre>				
	Output = MyClient.GetSegmentChildCount("Bob", "Pelvis");				
	Output - Myclient	-			
			utput.Result == Success utput.SegmentCount >= 0		
.NET	//	itput GetSegmentChildCount	utput.segmentcount >= 0		
.INL I	// public class of	rcput_Getsegmentchildcount			
		Decult.			
	// public Result				
	// public uint	SegmentCount;			
	// };				
	//				
	// Output_GetSegme	entChildCount GetSegmentChildC			
	//		string SegmentName);		

GetSegmentChildName

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

See Also: GetSegmentCount

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 Index is between 0 and GetSegmentChildCount()-1

```
// class Output GetSegmentChildName
// {
// public:
// Result::Enum Result;
// String SegmentName;
// };
//
// Output GetSegmentChildName GetSegmentName(
//
                     const String & SubjectName,
//
                      const String
                                       & SegmentName,
                       const unsigned int SegmentIndex ) const
11
ViconDataStreamSDK::CPP::Client MyClient;
MyClient.Connect( "localhost" );
MyClient.EnableSegmentData();
MyClient.GetFrame();
Output GetSegmentChildCount OutputGSCC;
OutputGSCC = MyClient.GetSegmentChildCount( "Bob", "Pelvis" );
                          // OutputGSCC.Result == Success
                          // OutputGSCC.SegmentCount == 2
Output GetSegmentChildName OutputGSCN;
OutputGSCN = MyClient.GetSegmentName( "Alice", 0 );
                          // OutputGSN.Result == InvalidSubjectName
                          // OutputGSN.SegmentName == ""
OutputGSCN = MyClient.GetSegmentName( "Bob", "Pelvis", 0 );
                          // OutputGSCN.Result == Success
                          // OutputGSCN.SegmentName == "LFemur"
OutputGSCN = MyClient.GetSegmentName( "Bob", "Pelvis", 1 );
                          // OutputGSCN.Result == Success
```

```
// OutputGSCN.SegmentName == "RFemur"
               OutputGSCN = MyClient.GetSegmentName( "Bob", "Pelvis", 2 );
                                          // OutputGSCN.Result == InvalidIndex
                                          // OutputGSCN.SegmentName == ""
                                          // (no third segment)
MATLAB
               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
```

GetSegmentStaticTranslation

Return the static pose translation of a subject segment.

See Also: GetSegmentStaticRotationHelical, GetSegmentStaticRotationMatrix,
GetSegmentStaticRotationQuaternion, GetSegmentStaticRotationEulerXYZ, GetSegmentLocalTranslation,
GetSegmentLocalRotationHelical, GetSegmentLocalRotationMatrix, GetSegmentLocalRotationQuaternion,
GetSegmentLocalRotationEulerXYZ

Input	Subject Name	string	The name of the subject		
прис	<u> </u>		-		
Output	Segment Name	string	The name of the segment.		
Output	Result	Result	Result NotConnected		
			Result.NotConnected		
			Result.NoFrame		
			Result.InvalidSubjectName		
			Result.InvalidSegmentName		
	Translation	double[3]	The translation of the		
			segment		
C++	// class Output_0	etSegmentStaticTranslation			
	// {				
	// public:				
	// Result::Enum	// Result::Enum Result;			
	// double	<pre>// double Translation[3];</pre>			
	// };				
	//				
	// Output_GetSegmentStaticTranslation GetSegmentStaticTranslation(
	// const String & SubjectName,				
	// const String & SegmentName) const;				
	ViconDataStreamSDK::CPP::Client MyClient;				
	<pre>MyClient.Connect("localhost");</pre>				
	MyClient.EnableSegmentData();				
	MyClient.GetFrame	e();			
	Outrook Cat Carrier	Chat's Turner lating Output			
		StaticTranslation Output =	// NDalada//) .		
MATLAB		mentStaticTranslation("Alice'			
IVIATLAD	% [Output] = Gets	egmentStaticTranslation(Sub	jectName, SegmentName)		
	<pre>MyClient = Client();</pre>				
	_				
	MyClient.Connect				
	<pre>MyClient.EnableSegmentData();</pre>				
	<pre>MyClient.GetFrame();</pre>				
	Output = MyClient	GetSegmentStaticTranslation	("Alice", "Pelvis");		
.NET		Output GetSegmentStaticTransla			
	// {				
	// public Resul	t Result;			
	=	e[] Translation;			
	// };	·			
	// //				
	// Output GetSegmentStaticTranslation GetSegmentStaticTranslation(
	// string SubjectName,				
	// string SegmentName);				

GetSegmentStaticRotationHelical

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

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

GetSegmentStaticRotationQuaternion, GetSegmentStaticRotationEulerXYZ, GetSegmentLocalTranslation, GetSegmentLocalRotationHelical, GetSegmentLocalRotationMatrix, GetSegmentLocalRotationQuaternion,

GetSegmentLocalRotationEulerXYZ

	ocalRotationEulerXYZ	т.	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[3]	The rotation of the segment		
C++	// class Output_G	etSegmentStaticRotationHelica	1		
	// {				
	// public:				
	// Result::Enum	Result;			
	// double	<pre>Rotation[3];</pre>			
	// };	,,			
	//				
	// Output_GetSegm	entStaticRotationHelical			
	// GetSegmentStaticRotationHelical(
	// const String & SubjectName,				
	// const String & SegmentName) const				
	ViconDataStreamSDK::CPP::Client MyClient;				
	<pre>MyClient.Connect("localhost");</pre>				
	<pre>MyClient.GetFrame();</pre>				
	-	StaticRotationHelical Output			
MATLAB		mentStaticRotationHelical("A			
IVIATLAD	% [Output] = GetSegmentStaticRotationHelical(SubjectName, SegmentName)				
	<pre>MyClient = Client();</pre>				
	MyClient.Connect(
	MyClient.GetFrame				
	mycrient.Getriame	(),			
	Output = MvClient	.GetSegmentStaticRotationHeli	cal("Alice", "Pelvis"):		
.NET		utput GetSegmentStaticRotatio			
	// public class 0				
	·	t Result;			
	-				
	// };	<pre>// public double[] Rotation; // }:</pre>			
	//				
	// Output_GetSegmentStaticKotationHelical // GetSegmentStaticRotationHelical(string SubjectName,				
	// GetsegmentstaticRotationHeildai (String SubjectName,);				
	,, stilling beginning and				
	ViconDataStreamSD	K.DotNET.Client MyClient =			
		-			

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

Output_GetSegmentStaticRotationHelical Output =
    MyClient.GetSegmentStaticRotationHelical( "Alice", "Pelvis" );
```

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, 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++	// class Output Ge	etSegmentStaticRotationMatrix	-		
	// {				
	<pre>// public: // Result::Enum Result;</pre>				
	// double	Rotation[9];			
	// };				
	//				
	// Output_GetSegme	entStaticRotationMatrix			
	// GetSegmentStaticRotationMatrix(// const String & SubjectName, // const String & SegmentName) const;				
	<pre>ViconDataStreamSDK::CPP::Client MyClient; MyClient.Connect("localhost"); MyClient.GetFrame();</pre>				
	Output_GetSegmentStaticRotationMatrix Output =				
MATI AD	MyClient.GetSegmentStaticRotationMatrix("Alice", "Pelvis");				
MATLAB	% [Output] = GetSe	egmentStaticRotationMatrix(Su	bjectName, SegmentName)		
	<pre>MyClient = Client();</pre>				
	<pre>MyClient.Connect("localhost");</pre>				
	<pre>MyClient.GetFrame();</pre>				
	Outmut - MacClinat		/ Nalice// Nacleric// N		
.NET		.GetSegmentStaticRotationMatri			
.INL I		utput_GetSegmentStaticRotation	Matrix		
	// {	- Dogult.			
	<pre>// public Result Result; // public double[] Rotation;</pre>				
	_	e[] ROLACION;			
	// }; //				
		entStaticRotationMatrix			
	_		iectName		
	<pre>// GetSegmentStaticRotationMatrix(string SubjectName, // string SegmentName);</pre>				
	/ /	SCIIIIG Seg.	merrowanie),		
	ViconDataStreamSDK.DotNET.Client MyClient =				
	v recirbacas creatilis Di	_			
	<pre>new ViconDataStreamSDK.DotNET.Client();</pre>				

```
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.

See Also: GetSegmentStaticTranslation, GetSegmentStaticRotationHelical, GetSegmentStaticRotationMatrix, GetSegmentStaticRotationEulerXYZ, GetSegmentLocalTranslation, GetSegmentLocalRotationHelical, GetSegmentLocalRotationMatrix, GetSegmentLocalRotationQuaternion, 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[4]	The rotation of the segment			
C++	// class Output GetSegmentStaticRotationQuaternion					
	// {					
	// public:					
	// Result::Enum Result;					
	<pre>// double Rotation[4];</pre>					
	// };					
	// // Output_GetSegmentStaticRotationQuaternion // GetSegmentStaticRotationQuaternion(// const String & SubjectName, // const String & SegmentName) const					
		<pre>K::CPP::Client MyClient;</pre>				
	_	MyClient.Connect("localhost");				
	MyClient.GetFrame	<pre>MyClient.GetFrame();</pre>				
		StaticRotationQuaternion Outpu				
MATLAB		gmentStaticRotationQuaternion(
IVIATLAD	% [Output] = GetSe	egmentStaticRotationQuaternion	(SubjectName, SegmentName)			
	MyClient = Client	() •				
	MyClient.Connect(
	<pre>MyClient.GetFrame();</pre>					
	Output = MyClient	.GetSegmentStaticRotationQuate	rnion("Alice", "Pelvis");			
.NET		tput GetSegmentStaticRotation				
	// {		~			
	·					
	// public double[] Rotation;					
	// public double[] Notation, // };					
	// // Output_GetSegmentStaticRotationQuaternion					
	// GetSegmentStaticRotationQuaternion(string SubjectName,					
	//	string	<pre>SegmentName);</pre>			

GetSegmentStaticRotationEulerXYZ

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

See Also: GetSegmentStaticTranslation, GetSegmentStaticRotationHelical, GetSegmentStaticRotationMatrix, GetSegmentStaticRotationQuaternion, GetSegmentLocalRotationHelical, GetSegmentLocalRotationHelical, GetSegmentLocalRotationMatrix, GetSegmentLocalRotationQuaternion, 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++	<pre>// class Output_GetSegmentStaticRotationEulerXYZ // { // public: // Result::Enum Result; // double Rotation[3]; // }; // // Output_GetSegmentStaticRotationEulerXYZ // GetSegmentStaticRotationEulerXYZ(// const String & SubjectName, // const String & SegmentName) const ViconDataStreamSDK::CPP::Client MyClient; MyClient.Connect("localhost"); MyClient.GetFrame();</pre>			
	<pre>Output_GetSegmentStaticRotationEulerXYZ Output = MyClient.GetSegmentStaticRotationEulerXYZ("Alice", "Pelvis");</pre>			
MATLAB	<pre>% [Output] = GetSegmentStaticRotationEulerXYZ(SubjectName, SegmentName) MyClient = Client(); MyClient.Connect("localhost"); MyClient.GetFrame(); Output = MyClient.GetSegmentStaticRotationEulerXYZ("Alice", "Pelvis");</pre>			
.NET	<pre>// public class Output_GetSegmentStaticRotationEulerXYZ // { // public Result Result; // public double[] Rotation; // }; // // Output_GetSegmentStaticRotationEulerXYZ // GetSegmentStaticRotationEulerXYZ(string SubjectName, // string SegmentName);</pre>			

GetSegmentGlobalTranslation

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

See Also: GetSegmentGlobalRotationHelical, GetSegmentGlobalRotationMatrix, GetSegmentGlobalRotationQuaternion, GetSegmentGlobalRotationEulerXYZ, GetSegmentLocalTranslation, GetSegmentLocalRotationHelical, GetSegmentLocalRotationMatrix, GetSegmentLocalRotationQuaternion, GetSegmentLocalRotationEulerXYZ

	Subject Name	etring	The name of the subject			
Input	Subject Name	string	 			
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			
	Occluded	boolean	True if the segment was			
			present at this frame. If false,			
			then Translation will be			
			[0,0,0]			
C++	// class Output_G	// class Output GetSegmentGlobalTranslation				
	// {					
	// public:					
	// Result::Enum	// Result::Enum Result; // double Translation[3]; // bool Occluded;				
	// double					
	// bool					
	// };					
	//					
	// Output GetSegm	entGlobalTranslation GetSegmen	tGlobalTranslation(
		// const String & SubjectName,				
		,, conditional formation of the conditional formation of the conditional formation of the conditional formation of the condition of the conditional formation of the condition of the conditional formation of the conditio				
	ViconDataStreamSD:	ViconDataStreamSDK::CPP::Client MyClient;				
	MyClient.Connect("localhost");				
	MyClient.EnableSe	MyClient.EnableSegmentData();				
	MyClient.GetFrame	<pre>MyClient.GetFrame();</pre>				
	_	Output_GetSegmentGlobalTranslation Output =				
		MyClient.GetSegmentGlobalTranslation("Alice", "Pelvis");				
MATLAB	% [Output] = GetS	egmentGlobalTranslation(Subje	ctName, SegmentName)			
	MyClient = Client					
	MyClient.Connect(
	MyClient.EnableSe	MyClient.EnableSegmentData();				
	MyClient.GetFrame	<pre>MyClient.GetFrame();</pre>				
NET		Output = MyClient.GetSegmentGlobalTranslation("Alice", "Pelvis");				
.NET	// public class 0	// public class Output_GetSegmentGlobalTranslation				
	// {	// {				
	// public Resul	// public Result Result;				

GetSegmentGlobalRotationHelical

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

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

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

Input	Subject Name	string	The name of the subject		
прис	Segment Name	string	The name of the segment.		
Output	Result	Result	Result.Success		
Output	Result	Result	Result.NotConnected		
			Result.NoFrame		
			Result InvalidSubjectName		
	Detetion	doblo[2]	Result.InvalidSegmentName		
	Rotation Occluded	double[3] boolean	The rotation of the segment		
	Occidaea	boolean	True if the segment was		
			present at this frame. If false,		
C++	//] 0 1 1 0		then Rotation will be [0,0,0]		
C++	-	etSegmentGlobalRotationHelical			
	// {				
	// public:	5 1.			
	// Result::Enum				
		Rotation[3];			
		Occluded;			
	// };				
	//	and Clabel Debat described			
	_	entGlobalRotationHelical			
		obalRotationHelical(
	<pre>// const String & SubjectName, // const String & SegmentName) const ViconDataStreamSDK::CPP::Client MyClient;</pre>				
		<pre>MyClient.Connect("localhost");</pre>			
	MyClient.GetFrame				
	ily offerie. Geer rame	MyCITEHC.Georgame();			
	Output GetSeament	GlobalRotationHelical Output =			
	-	mentGlobalRotationHelical("Al			
MATLAB		egmentGlobalRotationHelical(S			
			, , ,		
	MyClient = Client	();			
	MyClient.Connect(
	MyClient.GetFrame	();			
	Output = MyClient	.GetSegmentGlobalRotationHelic	al("Alice", "Pelvis");		
.NET	// public class 0	utput_GetSegmentGlobalRotation	Helical		
	// {				
	// public Resul	t Result;			
	// public double	e[] Rotation;			
	// public bool	Occluded;			
	// };				

GetSegmentGlobalRotationMatrix

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

See Also: GetSegmentGlobalTranslation, GetSegmentGlobalRotationHelical,

GetSegmentGlobalRotationQuaternion, GetSegmentGlobalRotationEulerXYZ, GetSegmentLocalTranslation, GetSegmentLocalRotationHelical, GetSegmentLocalRotationQuaternion, GetSegmentLocalRotationEulerXYZ

		egmentLocalRotationQuaternion, G	
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
	Occluded	boolean	True if the segment was
			present at this frame. If false,
			then Rotation will be all 0.
C++	// class Output_Ge	etSegmentGlobalRotationMatrix	
	// {		
	// public:		
	// Result::Enum	Result;	
	// double	<pre>Rotation[9];</pre>	
	// bool	Occluded;	
	// };		
	//		
	// Output_GetSegme	entGlobalRotationMatrix	
	// GetSegmentGlo	obalRotationMatrix(
	// const St	tring & SubjectName,	
	// const String & SegmentName) const;		
	ViconDataStreamSDI	K::CPP::Client MyClient;	
	<pre>MyClient.Connect("localhost");</pre>		
	MyClient.GetFrame	();	
	_	GlobalRotationMatrix Output =	
MATIAD		mentGlobalRotationMatrix("Ali	
MATLAB	% [Output] = GetSe	egmentGlobalRotationMatrix(Su	bjectName, SegmentName)
	MyClient = Client	();	
	MyClient.Connect("localhost");	
	MyClient.GetFrame	();	
	Output = MvClient	.GetSegmentGlobalRotationMatri	x("Alice", "Pelvis");
.NET		utput GetSegmentGlobalRotation	
	// {		
	// public Result	Result;	
	// public double		
	// public bool	Occluded;	
	// };		
	//		

GetSegmentGlobalRotationQuaternion

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

See Also: GetSegmentGlobalTranslation, GetSegmentGlobalRotationHelical, GetSegmentGlobalRotationMatrix, GetSegmentGlobalRotationEulerXYZ, GetSegmentLocalTranslation, GetSegmentLocalRotationHelical, GetSegmentLocalRotationMatrix, GetSegmentLocalRotationQuaternion, GetSegmentLocalRotationEulerXYZ

	alRotationEulerXYZ		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
	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_Ge	etSegmentGlobalRotationQuatern	ion
	// {		
	// public:		
	// Result::Enum		
	// double	Rotation[4];	
		Occluded;	
	// };		
	//		
	<u> </u>	entGlobalRotationQuaternion	
	_	lobalRotationQuaternion(
	// const String & SubjectName,		
	// const St	cring & SegmentName) const	
		K::CPP::Client MyClient;	
	MyClient.Connect(
	MyClient.GetFrame	();	
	Output_GetSegment(GlobalRotationQuaternion Outpu	t =
		gmentGlobalRotationQuaternion(
MATLAB	% [Output] = GetSe	egmentGlobalRotationQuaternion	(SubjectName, SegmentName)
	MyClient = Client	();	
	MyClient.Connect("localhost");	
	MyClient.GetFrame	();	
	0.1	Cat Carrier to Claba 177 to 11 to 12	
.NET		GetSegmentGlobalRotationQuate	
	// public class of	utput_GetSegmentGlobalRotation	⊼aacetii±0ii
	// public Result	Result;	
	// public double		
	// public bool	Occluded;	
	// };		
	//		

GetSegmentGlobalRotationEulerXYZ

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

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

Input	Subject Name	string	The name of the subject		
прис	Segment Name	string	The name of the segment.		
Output	Result	Result	Result.Success		
Output	Result	Result	Result.NotConnected		
			Result.NoFrame		
			Result.InvalidSubjectName		
			Result.InvalidSegmentName		
	Rotation	double[3]			
	Occluded	boolean	The rotation of the segment True if the segment was		
	Occidaed	boolean	present at this frame. If false,		
			then Rotation will be [0,0,0]		
C++	// alaga Output C	o+Cogmon+ClobalDo+a+ionEulogVV			
OTT	// Class Output_G	etSegmentGlobalRotationEulerXY	Δ		
	// t // public:				
	// Result::Enum	Pagul+•			
		Rotation[3];			
		Occluded;			
	// };	occiuded,			
	//				
		entGlobalRotationEulerXYZ			
	-				
	// const String & SabjectName, // const String & SegmentName) const ViconDataStreamSDK::CPP::Client MyClient;				
	MyClient.Connect(<pre>MyClient.Connect("localhost");</pre>			
	MyClient.GetFrame	<pre>MyClient.GetFrame();</pre>			
	Output_GetSegment	GlobalRotationEulerXYZ Output	=		
	MyClient.GetSeg	mentGlobalRotationEulerXYZ("A	alice", "Pelvis");		
MATLAB	% [Output] = GetS	egmentGlobalRotationEulerXYZ(SubjectName, SegmentName)		
	MyClient = Client				
	MyClient.Connect(
	MyClient.GetFrame	();			
NET	Cacpae Hyerrene	.GetSegmentGlobalRotationEuler	1112 (111100 / 101110),		
.NET		utput_GetSegmentGlobalRotation	EulerXYZ		
	// {				
	// public Resul				
	// public double				
	// public bool	Occluded;			
<u> </u>	// };				

GetSegmentLocalTranslation

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

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

GetSegmentLocalRotationQuaternion, GetSegmentLocalRotationEulerXYZ,

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

GetSegmentGlobalRotationQuaternion, GetSegmentGlobalRotationEulerXYZ

GetSegment		, GetSegmentGlobalRotationEuler>			
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 G	etSegmentLocalTranslation			
	// {				
	// public:				
	// Result::Enum	Result;			
	// double	<pre>Translation[3];</pre>			
	// bool	Occluded;			
	// };				
	//				
	// Output GetSegmentLocalTranslation GetSegmentLocalTranslation(
		tring & SubjectName,			
	ViconDataStreamSD	K::CPP::Client MyClient;			
	MyClient.Connect(
	MyClient.EnableSe	gmentData();			
	MyClient.GetFrame	();			
	Output GetSegment	LocalTranslation Output =			
	_	entLocalTranslation("Alice",	"Pelvis");		
MATLAB		egmentLocalTranslation(Subje			
	MyClient = Client	();			
	MyClient.Connect("localhost");			
	MyClient.EnableSe	gmentData();			
	MyClient.GetFrame	();			
		.GetSegmentLocalTranslation(
.NET	<pre>// public class 0 // {</pre>	utput_GetSegmentLocalTranslat	ion		
	// public Resul	t Result;			
	// public doubl	e[] Translation;			

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	
Прис	Segment Name	string	The name of the segment.	
Output	Result	Result	Result.Success	
Culput	1769uit	INGOUIL	Result.NotConnected	
			Result.NoFrame	
			Result.InvalidSubjectName	
			Result.InvalidSegmentName	
	Rotation	double[3]	The rotation of the segment	
	Occluded	boolean	True if the segment was	
	Joolaaca	boolean	present at this frame. If false,	
			then Rotation will be [0,0,0]	
C++	// class Output G	etSegmentLocalRotationHelical	their retailer will be [e,e,e]	
	// {	00009011000011.0000101110011		
	// public:			
	// Result::Enum	Result;		
	// double	Rotation[3];		
	// bool	Occluded;		
	// };			
	//			
	// Output_GetSegm	entLocalRotationHelical		
	// GetSegmentLo	calRotationHelical(
	// const S	tring & SubjectName,		
	// const S	tring & SegmentName) const		
	ViconDataStreamSD	K::CPP::Client MyClient;		
	MyClient.Connect(-		
	MyClient.GetFrame			
	ing offeno.			
	Output_GetSegment	LocalRotationHelical Output =		
	MyClient.GetSeg	mentLocalRotationHelical("Ali	ce", "Pelvis");	
MATLAB	% [Output] = GetS	% [Output] = GetSegmentLocalRotationHelical(SubjectName, SegmentName)		
	MyClient = Client			
	MyClient.Connect(
	MyClient.GetFrame	();		
	Output = MyClient	.GetSegmentLocalRotationHelica	l("Alice", "Pelvis");	
.NET	// public class 0	${ t utput_GetSegmentLocalRotationH}$	elical	
	// {			
	// public Resul			
	// public doubl	e[] Rotation;		
	// public bool	Occluded;		
	// };			

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
Output	rtoount	result	Result.NotConnected
			Result.NoFrame
			Result.InvalidSubjectName
			Result.InvalidSegmentName
	Rotation	double[9]	The rotation of the segment
	Occluded	boolean	True if the segment was
	Occidada	boolcari	present at this frame. If false,
			then Rotation will be all 0
C++	// class Output G	etSegmentLocalRotationMatrix	mon redución win se un e
	// {	0000g0100u11100u010	
	// public:		
	// Result::Enum	Result;	
	// double		
	// bool		
	// };		
	//		
	// Output GetSegm	entLocalRotationMatrix	
	-	calRotationMatrix(
	// const S	tring & SubjectName,	
	// const S	tring & SegmentName) const;	
	ViconDataStreamSD	K::CPP::Client MyClient;	
	MyClient.Connect("localhost");	
	MyClient.GetFrame	();	
	_	LocalRotationMatrix Output =	
MAATI AD		mentLocalRotationMatrix("Alic	
MATLAB	% [Output] = GetS	egmentLocalRotationMatrix(Sub	ojectName, SegmentName)
	Marcliant - Cliant	() -	
	MyClient = Client MyClient.Connect(
	MyClient.GetFrame		
	mycrient.Getriame	(),	
	Output = MyClient	.GetSegmentLocalRotationMatri>	x("Alice", "Pelvis");
.NET		utput GetSegmentLocalRotationN	
	// {	<u> </u>	
	// public Resul	t Result;	
	=	e[] Rotation;	
		Occluded;	
	// };		

GetSegmentLocalRotationQuaternion

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

See Also: GetSegmentLocalTranslation, GetSegmentLocalRotationHelical, GetSegmentLocalRotationMatrix, 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
Output	result	Result	Result.NotConnected
			Result.NoFrame
			Result.InvalidSubjectName
			Result.InvalidSegmentName
	Rotation	double[4]	The rotation of the segment
	Occluded	boolean	True if the segment was
	Occidaed	boolean	present at this frame. If false,
			then Rotation will be [0,0,0,0]
C++	// -1 0		
OTT	// Class Output_G	etSegmentLocalRotationQuaterni	LOII
	// t // public:		
	// Result::Enum	Pacult.	
	// double		
		Occluded;	
	// 3;	occiuded,	
	//		
		entLocalRotationQuaternion	
	_	ocalRotationQuaternion(
	=	tring & SubjectName,	
		tring & SabjectName,	
	, , consc s	erring a begineriename , conoc	
	ViconDataStreamSD	K::CPP::Client MyClient;	
	MyClient.Connect("localhost");	
	MyClient.GetFrame	();	
	_	LocalRotationQuaternion Output	
		gmentLocalRotationQuaternion(
MATLAB	% [Output] = GetS	egmentLocalRotationQuaternion	(SubjectName, SegmentName)
	W 01 '		
	MyClient = Client		
	MyClient.Connect(
	MyClient.GetFrame	();	
	Output = MyClient	.GetSegmentLocalRotationQuate:	rnion("Alice", "Pelvis");
.NET	// public class 0	utput_GetSegmentLocalRotation(Quaternion
	// {		
	// public Resul	t Result;	
	// public doubl	e[] Rotation;	
	// public bool		
	// };		

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
Catpat	rtoount	rtosan	Result.NotConnected
			Result.NoFrame
			Result.InvalidSubjectName
			Result.InvalidSegmentName
	Rotation	double[3]	The rotation of the segment
	Occluded	boolean	True if the segment was
	Occidada	Boolean	present at this frame. If false,
			then Rotation will be [0,0,0]
C++	// class Output Ge	 etSegmentLocalRotationEulerXYZ	mon retation will be [e,e,e]
.	// {	obegment Ebotatio dation Earcini E	
	// public:		
	// Result::Enum	Result:	
	// double		
	// bool		
	// };	,	
	//		
	// Output GetSegme	entLocalRotationEulerXYZ	
	<u> </u>	calRotationEulerXYZ(
	// const St	tring & SubjectName,	
	// const St	tring & SegmentName) const	
	ViconDataStreamSDF	K::CPP::Client MyClient;	
	MyClient.Connect("localhost");	
	MyClient.GetFrame	();	
	_	LocalRotationEulerXYZ Output =	
	MyClient.GetSegr	nentLocalRotationEulerXYZ("Al	ice", "Pelvis");
MATLAB	% [Output] = GetSe	egmentLocalRotationEulerXYZ(S	ubjectName, SegmentName)
	MyClient = Client		
	MyClient.Connect(
	MyClient.GetFrame	();	
	0	Cot Comment I and I be to the court of the c	WE (Nalice // Nacional
.NET		.GetSegmentLocalRotationEulerX	
.INL I	// public class Ou	utput_GetSegmentLocalRotationE	ntervia
	// t // public Result	- Decul+•	
	// public double		
	// public double		
	// public bool // };	occiuded,	

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

Input	Subject Name	string	The name of the subject			
Output	Result	Result	Result.Success			
Output	Result	Result	Result.NotConnected			
			Result.NoFrame			
	Marker Count	unsigned integr	Result.InvalidSubjectNam	ie		
C		unsigned intege	The number of markers			
C++	// class Output_G	etMarkerCount				
	// {					
	<pre>// public: // Result::Enum</pre>	Dag. 14.				
	// Result::Enum					
	// unsigned int	MarkerCount;				
	// //					
	// Output GetMark	erCount GetMarke	rCount (
		ng & SubjectName				
	,, сопас аст	a babjectwame	, 3030,			
	ViconDataStreamSD	K::CPP::Client M	yClient;			
	MyClient.EnableMa		-			
	MyClient.Connect(<pre>MyClient.Connect("localhost");</pre>				
	Output_GetMarkerC	Output_GetMarkerCount Output;				
	Output = MyClient	Output = MyClient.GetMarkerCount("Bob"); // Output.Result == NoFrame				
		// Output.MarkerCount == 0				
	MyClient.GetFrame	<pre>MyClient.GetFrame();</pre>				
	Output = MvClient	Output = MyClient.GetMarkerCount("Alice");				
		// Output == InvalidSubjectName				
			// Output.MarkerCount == 0			
		// (no "Alice")				
	Output = MyClient	.GetMarkerCount("Bob"); // Output.Result == Success			
			<pre>// Output.MarkerCount >= 0</pre>			
MATLAB	% [Output] = GetM	arkerCount(Subj	ectName)			
	MyClient = Client					
	MyClient.EnableMa					
	MyClient.Connect("localhost");				
	Output = MyClient	Output = MyClient.GetMarkerCount("Bob"); % Output.Result == NoFrame				
			% Output.MarkerCount == 0			
	MyClient.GetFrame	();				
	Output = MyClient	.GetMarkerCount("Alice"):			
	Output = myclient	.Getmarkercount(<pre>% Output.Result == InvalidSubjectName % Output.MarkerCount == 0</pre>			

```
% (no "Alice")
              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.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.

See Also : G	GetMarkerCount, GetMarke	erName, GetMarkerGlobalTransl	ation			
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_G	etMarkerParentName				
	// {					
	// public:					
	// Result::Enum	Result;				
	// String	SegmentName;				
	// };					
	//					
	// Output_GetMark	erParentName GetMarkerParer	ntName(
	// const Stri	ng & SubjectName,				
	// const Stri	// const String & MarkerName) const;				
	ViconDataStreamSD	ViconDataStreamSDK::CPP::Client MyClient;				
	<u> </u>	<pre>MyClient.Connect("localhost");</pre>				
	<u> </u>	<pre>MyClient.EnableMarkerData();</pre>				
	<pre>MyClient.GetFrame();</pre>					
	Output GetMarkerParentName Output;					
		arentname Output; .GetMarkerParentName("Bob'	/ WIFUD//) •			
	Output - Myclient		nt.Result == Success			
		-	nt.Result Success nt.SegmentName == "Head"			
		// Outpt	rc. Segmenthame nead			
MATLAB	% [Output] = GetM	arkerParentName(SubjectNam	me, MarkerName)			
		, ,	,			
	MyClient = Client	();				
	MyClient.Connect("localhost");				
	MyClient.EnableMa	rkerData();				
	MyClient.GetFrame	MyClient.GetFrame();				
	Output = MyClient	.GetMarkerParentName("Bob'				
		-	ut.Result == Success			
		// Outpi	nt.SegmentName == "Head"			
.NET	// public class 0	utput GetMarkerParentName				
	// public class 0	achac actual vertare including				
	// t // public Resul	+ Dacul+•				
	-					
		g SegmentName;				
	// };					
	//					

GetMarkerGlobalTranslation

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

See Also : G	etMarkerName					
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			[0,0,0]			
C++	_	GetMarkerGlobalTranslatio	n			
	// {					
	// public:	December 1				
	// Result::Enu					
		// double Translation[3];				
	<pre>// bool Occluded; //);</pre>					
	//					
	// Output GetMarkerGlobalTranslation GetMarkerGlobalTranslation(
	// const String & SubjectName,					
	// const String & MarkerName) const;					
	,, , , , , , , , , , , , , , , , , , , ,					
	ViconDataStreamSDK::CPP::Client MyClient;					
	<pre>MyClient.Connect("localhost");</pre>					
	<pre>MyClient.EnableMarkerData();</pre>					
	<pre>MyClient.GetFrame();</pre>					
	_	GlobalTranslation Output				
MAATI AD		rkerGlobalTranslation("A				
MATLAB	% [Output] = GetMarkerGlobalTranslation(SubjectName, MarkerName)					
	MucClient = Client():					
	<pre>MyClient = Client(); MyClient.Connect("localhost");</pre>					
	MyClient.EnableMarkerData();					
	<pre>MyClient.GetFrame();</pre>					
	ing offent. Occir I an	· · · · · · · · · · · · · · · · · · ·				
	Output = MyClier	nt.GetMarkerGlobalTranslat	ion("Alice", "LASI");			
.NET	// public class Output GetMarkerGlobalTranslation					
	// {					
	// public Result;					
	// public double[] Translation[];					
	// public bool Occluded;					
	// };					
	//					

GetUnlabeledMarkerCount

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

See Also: GetGlobalUnlabeledMarkerTranslation

Input	tGiobaiUniabelediviarker	Translation .		
Output	Result	Result	Result.Success Result.NotConnected Result.NoFrame	
	MarkerCount	unsigned integer	The number of markers	
C++	// { // public: // Result::Enur // unsigned int // }; //	·	MarkerCount() const;	
	ViconDataStreamSI	DK::CPP::Client MyClient;		
		nlabeledMarkerData();		
	MyClient.Connect			
	MyClient.GetFrame			
	Output_GetUnlabeledMarkerCount Output = MyClient.GetUnlabeledMarkerCount(); // Output.Result == Success // Output.MarkerCount >= 0			
MATLAB	% [Output] = GetUnlabeledMarkerCount();			
	<pre>MyClient = Client(); MyClient.EnableUnlabeledMarkerData(); MyClient.Connect("localhost"); MyClient.GetFrame();</pre>			
	Output = MyClient.GetUnlabeledMarkerCount(); // Output.Result == Success // Output.MarkerCount >= 0			
.NET	<pre>// { // public Result // public uint // }; // // Output_GetUnla ViconDataStreamSI MyClient.EnableUnd </pre>	MarkerCount; abeledMarkerCount GetUnlabeledMarkerCount GetUnlabeledMarkerDataStreament	MarkerCount(); amSDK.DotNET.Client();	
	<pre>MyClient.Connect("localhost"); MyClient.GetFrame();</pre>			

Output_GetUnlabeledMarkerCount Output =

MyClient.GetUnlabeledMarkerCount(); // Output.Result == Success

// Output.MarkerCount >= 0

GetUnlabeledMarkerGlobalTranslation

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

See Also: GetUnlabelledMarkerCount

See Also : Ge	etUnlabelledMarkerCour	nt			
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 In	dex is between 0 and GetUr	nlabeledMarkerCount()-1		
	// class Output	GetUnlabeledMarkerGlobalTr	ranslation		
	// {				
	// public:				
	// Result::Enu	ım Result;			
	// double	<pre>Translation[3];</pre>			
	// };				
	//				
	// Output GetUn	abeledMarkerGlobalTranslat	tion		
	// GetUnlabeledMarkerGlobalTranslation(
	// const unsigned int MarkerIndex) const;				
	ViconDataStreamSDK::CPP::Client MyClient;				
	<pre>MyClient.Connect("localhost");</pre>				
	<pre>MyClient.EnableUnlabeledData();</pre>				
	<pre>MyClient.GetFrame();</pre>				
	Output_GetUnlabeledMarkerGlobalTranslation Output =				
MyClient.GetUnlabeledMarkerGlobalTranslation(0);			ation(0);		
MATLAB	A valid Marker In	A valid Marker Index is between 1 and GetUnlabeledMarkerCount()			
	% [Output] = GetUnlabeledMarkerGlobalTranslation(MarkerIndex)				
	<pre>MyClient = Client();</pre>				
	MyClient.Connect	<pre>MyClient.Connect("localhost");</pre>			
	<pre>MyClient.EnableUnlabeledData();</pre>				
	<pre>MyClient.GetFrame();</pre>				
	Output = MyClient.GetUnlabeledMarkerGlobalTranslation(1);				
.NET	A valid Marker Index is between 0 and GetUnlabeledMarkerCount()-1				
	// public class Output_GetUnlabeledMarkerGlobalTranslation				
	// {	// {			
	// public Resu	// public Result;			
	// public doub	// public double[] Translation;			
	// };				
	//				
	// Output_GetUnlabeledMarkerGlobalTranslation				

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

Input	etDeviceivame			
Output	Result	Result	Result.Success Result.NotConnected Result.NoFrame	
	Device Count	unsigned integer	The number of devices	
C++	<pre>// class Output_GetDeviceCount // { // public: // Result::Enum Result; // unsigned int DeviceCount; // }; // // Output_GetDeviceCount GetDeviceCount() const; ViconDataStreamSDK::CPP::Client MyClient;</pre>			
	MyClient.EnableDe	viceData();		
	MyClient.Connect("localhost");		
	MyClient.GetFrame	<pre>MyClient.GetFrame();</pre>		
	<pre>Output_GetDeviceCount Output = MyClient.GetDeviceCount();</pre>			
MATLAB	// Output.DeviceCount >= 0		Output.DeviceCount >= 0	
NET		<pre>(); viceData(); "localhost"); (); .GetDeviceCount(); // Output.</pre>	Result == Success DeviceCount >= 0	
.NET	<pre>// { // public Resul // public uint // }; // // Output_GetDevi</pre>	<pre>DeviceCount; ceCount GetDeviceCount(); K.DotNET.Client MyClient =</pre>	mSDK.DotNET.Client();	

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

Input	Device Index	unsigned integer	The index of the device.
Output	Result	Result	Result.Success
			Result.NotConnected
			Result.NoFrame
			Result.InvalidIndex
	Device Name	string	The name of the device
	Device Type	DeviceType	Unknown
			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 == ""
                                  // OutputGDN.DeviceType == Unknown
```

```
MATLAB
               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;
                                   DeviceName;
               // public string
               // 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 : Ge	etDeviceName, GetDevic	eOutputName			
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_GetDeviceOutputCount				
	// {				
	// public:				
	// Result::Enu	m Result;			
	// unsigned in	t DeviceOutputCount;			
	// };				
	//				
	// Output GetDev	iceOutputCount GetDev	iceOutputCount(
		const String & Device	Name) const;		
		// const string a sevicename / const,			
	ViconDataStreamSDK::CPP::Client MyClient;				
	<pre>MyClient.Connect("localhost");</pre>				
	<pre>MyClient.EnableDeviceData();</pre>				
	<pre>MyClient.GetFrame();</pre>				
	Output_GetDeviceOutputCount Output;				
	Output = MyClien	Output = MyClient.GetDeviceOutputCount("DataGlove");			
	// Output.Result == InvalidDeviceName				
		//	Output.DeviceOutputCount == 0		
		//	(no "DataGlove" device)		
	<pre>Output = MyClient.GetDeviceOutputCount("ZeroWire");</pre>				
		//	Output.Result == Success		
		//	Output.DeviceOutputCount == 6		
MATLAB	% [Output] = Get	DeviceOutputCount(De	viceName)		
	MyClient = Clien	t();			
	<pre>MyClient.Connect("localhost");</pre>				
	MyClient.EnableDeviceData();				
	MyClient.GetFram	<pre>MyClient.GetFrame();</pre>			
	<pre>Output = MyClient.GetDeviceOutputCount("DataGlove");</pre>				
		//	Output.Result == InvalidDeviceName		
		//	Output.DeviceOutputCount == 0		
		//	(no "DataGlove" device)		
	Output = MyClient.GetDeviceOutputCount("ZeroWire");				
		-	Output.Result == Success		
			Output.DeviceOutputCount == 6		

```
.NET
               // public class Output_GetDeviceOutputCount
               // {
               // 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.

 $See \ Also: Get Device Count, \ Get Device Output Count, \ Get Device Output Value$

Input	Device Name	string	The device name	
put	Device Output	integer	The index of the device	
	Index	integer	output.	
Output	Result	Result	Result.Success	
Output	Result	Result	Result.NotConnected	
			Result.NoFrame	
			Result.InvalidDeviceName	
			Result.InvalidIndex	
	Device Output	string	The name of the device	
	Name		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			
	/ valid Device durpat mack to between a and design needs apared and			
	// class Output_GetDeviceOutputName			
	// Class output_GetDeviceOutputName			
	// { // public:			
ı	// DogultEnum	Dogul+.		
	// Result::Enum	•		
	// String	DeviceOutputName;		
	// String // Unit::Enum	•		
	// String // Unit::Enum // };	DeviceOutputName;		
	// String // Unit::Enum // }; //	<pre>DeviceOutputName; DeviceOutputUnit;</pre>		
	<pre>// String // Unit::Enum // }; // // Output_GetDevi</pre>	DeviceOutputName; DeviceOutputUnit; ceOutputName GetDeviceOutputN	ame (
	// String // Unit::Enum // }; //	DeviceOutputName; DeviceOutputUnit; ceOutputName GetDeviceOutputN	ame (
	<pre>// String // Unit::Enum // }; // // Output_GetDevi // const S</pre>	DeviceOutputName; DeviceOutputUnit; ceOutputName GetDeviceOutputN		
	<pre>// String // Unit::Enum // }; // // Output_GetDevi // const S // const u</pre>	DeviceOutputName; DeviceOutputUnit; ceOutputName GetDeviceOutputN tring & DeviceName, nsigned int DeviceOutputInde		
	<pre>// String // Unit::Enum // }; // // Output_GetDevi // const S // const u ViconDataStreamSD</pre>	DeviceOutputName; DeviceOutputUnit; ceOutputName GetDeviceOutputN tring & DeviceName, nsigned int DeviceOutputInde K::CPP::Client MyClient;		
	<pre>// String // Unit::Enum // }; // // Output_GetDevi // const S // const U ViconDataStreamSD MyClient.Connect(</pre>	DeviceOutputName; DeviceOutputUnit; ceOutputName GetDeviceOutputN tring & DeviceName, nsigned int DeviceOutputInde K::CPP::Client MyClient; "localhost");		
	<pre>// String // Unit::Enum // }; // // Output_GetDevi // const S // const u ViconDataStreamSD</pre>	DeviceOutputName; DeviceOutputUnit; ceOutputName GetDeviceOutputN tring & DeviceName, nsigned int DeviceOutputInde K::CPP::Client MyClient; "localhost"); viceData();		

```
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

		DeviceOutputCount, GetDeviceOutputName		
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	
	Gooladoa	200.00.1	this frame. If false, then Value will	
			be 0.	
C++	// class Output_G	etDeviceOutputValue	1 20 0.	
	// {			
	// public:			
	// Result::Enum	Result;		
	// double	Value;		
	// bool	Occluded;		
	// };			
	//			
	// Output_GetDevi	ceOutputValue		
	<pre>// GetDeviceOutputValue(// const String & DeviceName, // const String & DeviceOutputName) const; ViconDataStreamSDK::CPP::Client MyClient;</pre>			
		<pre>MyClient.Connect("localhost"); MyClient.EnableDeviceData(); MyClient.GetFrame();</pre>		
	-			
	Output GetDeviceO	utputValue Output =		
	-	iceOutputValue("AMTI",	"Fx");	
	-		Output.Result == Success	
			Output.Value == ?	
		// (Output.Occluded = ?	
MATLAB	// [Output] = Ge	tDeviceOutputValue(Dev	iceName, DeviceOutputName)	
	MyClient = Client	();		
	MyClient.Connect("localhost");		
	MyClient.EnableDe	viceData();		
	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 = ?
```

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

^{***} These functions will not work with Vicon Nexus 1.4 and Vicon Blade 1.6.