Vicon RealTime SDK

Product Guide—Foundation Notes

This document describes the RealTime SDK, which enables you to create a client-server application to obtain the raw motion data acquired by Vicon cameras as the RealTime Engine streams it to Vicon application software. You can use the acquired real-time data in Vicon or third-party data visualization, analysis, or manipulation software.

This version of the RealTime SDK is based on the Tarsus RealTime Engine available in Vicon iQ and Workstation. There is a separate RealTime SDK based on the CG Stream available in Vicon Nexus; however, you can still use this Tarsus version with Vicon Nexus.

Contents

Vicon RealTime SDK Product Guide—Foundation Notes Revision 1.0 July 2008

For use with RealTime SDK Version 1.1 running with Vicon iQ, Vicon Nexus, Vicon Workstation under Vicon MX systems (including T-Series, F-series, MX+, and MX cameras and units) or V-series systems.

Vicon Motion Systems is an OMG Plc company.

Email: support@vicon.com

Web: http://www.vicon.com

Foundation Notes

This section of the Product Guide provides essential product information, which you need to get started using the current release of the RealTime SDK based on the Tarsus RealTime Engine. This version of the RealTime SDK runs with Vicon application software (Vicon iQ 2.x and Vicon Workstation 5.x) under Vicon MX systems (including T-Series, F-series, MX+, and MX cameras and units) or V-series systems. It runs with Nexus 1.x under Vicon MX systems for backward compatibility only. This document describes the main features and functionality; lists prerequisites and any limitations; and provides instructions on installing and using the RealTime SDK files.

Important

This document assumes that you are familiar with standard motion capture, data processing, and data management performed in the Vicon application software you are using. If you are not, see the documentation supplied with your Vicon application software for information on its features and functionality (for details, see *Product Documentation* on page 21).

About the RealTime SDK

The RealTime SDK is a set of sample software and documentation files provided to enable you to write your own client-server application to obtain the raw motion data acquired by Vicon cameras as the RealTime Engine streams it through Vicon application software. You can then use the data components you are interested in (e.g. labeled trajectories, additional angles, etc.) in data visualization, analysis, or manipulation software such as Vicon Polygon or third-party computer graphics software (for example, life sciences programs such as MATLAB or LabVIEW, or animation packages such as 3ds Max or Maya).

The Vicon RealTime Engine (RTE) produces 3D data based on the raw 2D motion data acquired by Vicon cameras. The RTE acquires data from the cameras and then reconstructs, labels, and optionally kinematically fits the data and then streams it in real time to your chosen visualization software.

For Vicon iQ and Workstation, the RTE is a separate application. You manage the RealTime data acquisition and processing within Vicon iQ and Workstation using the RealTime Engine Control, as shown in Figure 2-1. The RTE runs in a separate Command Prompt window, and you view the streaming data in Vicon iQ, Workstation, or in separate visualization software (such as Vicon Polygon or third-party computer graphics software).

Figure 2-1: RealTime Engine Control (Vicon iQ)

In Vicon Nexus, the RTE is integrated, so data streaming and visualization is all done within the Nexus (in the System resources pane and 3D Perspective view pane).

This *Vicon RealTime SDK Product Guide* explains the type of motion data you can obtain from the RealTime Engine data stream and provides guidance on writing a client-server application to obtain this motion data. The supplied sample files provide working examples of such code to assist you in writing your own client-server application. The supplied examples are written in C++, but you can use any software coding language.

Important

This document assumes that you are an experienced software programmer and are able to read and understand C++ code.
Before using the RealTime SDK, you are advised to read the *Disclaimer* below.

Disclaimer

The RealTime SDK is not covered under normal warranty or support agreements and future releases are not guaranteed. However, if you experience problems using this documentation or sample files, please inform Vicon Motion Systems Ltd.

Installation Requirements

This section describes the what you need to do to prepare for and install the RealTime SDK.

For details on the hardware requirements and the system architecture for setting up Vicon iQ or Workstation and the RealTime Engine, see the documentation supplied with your Vicon software (for details, see *Product Documentation* on page 21). For example, you might implement a distributed architecture where you install Vicon application software on the Vicon host PC, the RealTime Engine software on a remote PC, and the visualization/analysis software on another remote PC. In this architecture, each PC requires its own Ethernet card for client-server data communications (this is in addition to the Vicon Ethernet card on the host

PC). Alternatively, you may implement a standalone architecture with Vicon application software, the RealTime Engine, any visualization software, and any other Vicon application software installed on the same host PC with a dual processor.

For Vicon Nexus, the RTE is integrated in the product software, so no additional RTE set up is required.

There are no special software requirements in addition to those for Vicon application software (for details, see the documentation supplied with your release of Vicon application software). If you want to compile the sample application supplied with the RealTime SDK, you will need Microsoft C++ version 6 or later.

Table 2-1 lists and describes all the files included in the RealTime SDK

Table 2-1: RealTime SDK Files

RealTime SDK Component	File Name	Description
Documentation	RealTimeSDK_Tarsus _ProductGuide _Foundation.pdf	This document
ExampleClientSource		Uncompiled C++ source files that can be used as examples for coding the client portion of your application. Note that this sample code is not related to the TestClient.exe application.
	ClientCodes.cpp	Source file that initializes some of the structures in ClientCodes.h.
	ClientCodes.h	Header file containing tokens and structures used by the data stream.
	ExampleClient.cpp	Source file illustrating the use of the data stream.
	ExampleClient.dsp	Microsoft Developer Studio Project file with the settings for the example.

Table 2-1: RealTime SDK Files

RealTime SDK Component	File Name	Description
RTViewer	RTViewer.exe	RealTime Viewer application
Meshes	CumnorBones.obj	Object file that defines mesh segments corresponding to names and segments in an associated *.vst* file or modeled data.
	CumnorBones.vamp	Map file that describes where the viewer should attach the various segments of a mesh to the bodies from the RT.
	Skeleton.def	Skeleton file. It describes a general skeleton so that the viewer can infer a skeleton from the RT channels. Also contains some scaling information. Can be edited with a text editor.
TestClient	TestClient.exe	Test Client application

Installation Instructions

This section provides instructions for installing the RealTime SDK files on the Vicon host PC along with your Vicon application software.

You can install the RealTime SDK files in either of the following ways:

- As part of an installation of some Vicon application software

  For information on the default file locations, see Table 2-1 on page 4. For instructions on installing Vicon application software, see *Vicon Nexus Product Guide—Foundation Notes*, *New Features in Vicon iQ 2.5*, or *New Features in Vicon Life Sciences Products 2006* as appropriate.

- As a separately installable utility available from the **Downloads** section of Vicon Online Support

  If you installed an earlier version of the RealTime SDK with your Vicon application software, you can obtain the latest version from this **Downloads** section.

Important

Before installing the RealTime SDK, close any open Vicon applications.

To download and install RealTime SDK from Vicon Online Support:

1. From the **Downloads** section of Vicon Online Support (*http://www.vicon.com/support/downloads.php*), navigate to the **Animation** area (Vicon iQ users) or the **Life Sciences** area (Workstation or Nexus users), then click **Tools and Utilities**, and then **RealTimeSDK**.
2. In the displayed **File Download** dialog box, click **Save**.
3. In the displayed **Save As** dialog box, in the **Save In** field, navigate to a location on your PC to download the file *RealTimeSDK.zip* and click **Save**.
4. Use a file extractor utility, such as WinZip, to extract the files from *RealTimeSDK.zip* to the RealTime SDK folder (by default, *C:\Program Files\Vicon\SDKs\RealTime*), ensuring that you select the option to **Use folder names**.

   Table 2-2 shows where the files are placed when the are extracted from the *.zip* file.

Table 2-2: RealTime SDK file locations

Location (under C:\Program Files\Vicon\SDKs\RealTime)	File Name
\Documentation	*RealTimeSDK_ProductGuide_Foundation.pdf*
\ExampleClientSource	*ClientCodes.cpp* *ClientCodes.h* *ExampleClient.cpp* *ExampleClient.dsp*
\RTViewer	*RTViewer.rtf* *RTViewer.exe*
\Meshes	*CumnorBones.obj*
\Meshes	*CumnorBones.vamp*
\Meshes	*Skeleton.def*
\TestClient	TestClient.exe

Tip

If you want to remove this or an earlier version of the RealTime SDK, from Windows Explorer you can manually delete the folders or files under the RealTime SDK program folder.

Product Update and Patch Information

The information in this document is correct at the time of release. You can obtain latest firmware and other software patches, models and scripts, and product documentation from the Vicon Online Support knowledge base on the Web (www.vicon.com.support/).

Creating a Client-Server Application to Acquire Vicon RealTime Data

The Vicon RealTime Engine produces 3D data based on the raw motion data acquired by Vicon MX (including the new T-Series) and Vicon V-series system cameras. The RealTime Engine reconstructs, labels, and optionally kinematically fits the data and then streams it in real time to your chosen visualization software.

To obtain Vicon real-time data from Vicon application software to incorporate into other Vicon or third-party animation packages (such as Max or Maya)/ life sciences programs (such as MATLAB or LabVIEW), you use the RealTime Engine Software Developers Kit (SDK) to create your own client-server application.

In this application, the RealTime Engine acts as the server; you write one or more clients to obtain the Vicon motion capture data and to identify and use the desired information. Communication takes places using TCP/IP (stream sockets).

The following sections describe these steps in creating a client-server application:

- *Creating a Client To Acquire Data from the RTE Server*
- *Creating a Client to Use Data Packets Returned by the RTE* on page 8

Creating a Client To Acquire Data from the RTE Server

You must write a client to obtain data from the RTE server. Your client can obtain data from the RTE through either of the following methods:

- frame-by-frame polling (request-reply-request-reply...)
- continuous streaming (request-reply-reply-reply...)

To acquire RTE data through frame-by-frame polling:

1. Your client connects to Vicon RealTime Engine (RTE) on port 800 and requests an Info packet.
2. The Vicon RTE returns an Info packet containing the names of available channels within the real-time data stream.

3. Your client sends a request for a Data packet.
4. The Vicon RTE returns a Data packet containing an array of real numbers (doubles), which identify what data is contained in each of the channels described in the Info packet.
5. Your application repeats steps 3-5 for each required frame of data.

To acquire RTE data through continuous streaming:

1. Your client connects to the Vicon RealTime Engine (RTE) on port 800 and requests an Info packet.
2. The Vicon RTE returns an Info packet containing the names of available channels within the real-time data stream.
3. Your client sends a Streaming On packet.
4. The Vicon RTE continuously returns a Data packet for each sample that it processes.
5. Your client requests either a Streaming Off or a Close connection packet.

For details on the protocols to send requests and receive packets from the RTE server, see *Client-Server Communications Protocol Specifications* on page 9.

Once your client has required the desired real-time data, it (or another client you create) must identify and analyze particular points of interest in the channels within the returned Info and Data packets. This is described in the following section.

Creating a Client to Use Data Packets Returned by the RTE

You must create a client to identify and analyze particular points of interest in the channels within the Info and Data packets returned by the RTE.

To identify and use the data acquired from the RTE:

1. Your client searches an Info packet for the information of interest and records the indices of those channels.
2. Your client uses the indices it created of the channels in the Info packet to look up information in Data packets.

   The position of each value in the array of real numbers (doubles) indicates which channel it corresponds to (i.e. the nth value in the Data packet corresponds to the nth string in the Info packet).

   To enable your client to understand the supplied data, additional information is encoded in channel names:

- Time
- Timecode (not supported by Nexus 1.x)
- Marker
- Global Body (not supported by Nexus 1.x)
- Local Body
- Free Joint
- Ball Joint
- 2DOF Joint
- Hinge Joint

For details on the content and format of these channels, see *Real-time Data Channel Specifications* on page 11)

Client-Server Communications Protocol Specifications

This section shows the format for your client requests the RTE, and the format of the data the RTE returns. Your code must use Intel byte order and double formats.

Communications for client requests and RTE returns use the following format:

```
| Type | Packet | Body |
```

where:

Type is a long int that takes either of the following values:

0 - Request
1 - Reply

Packet is a long int that takes one of the following values:

0 - Close
1 - Info
2 - Data
3 - Streaming On
4 - Streaming Off

Body is the content of the client request or the packet returned by the RTE:

- No body
- Info packet
- Data packet

For details, see *Body Contents*.

Body Contents

This section describes the format of the content of a client request or of a packet returned by the RTE.

Client Request

A client request contains no body.

An RTE Reply to Client Close Request

A reply by the RTE to a client close request contains no body.

Data Packet

The format for the body of a Data packet is:

```
| Count | Value | Value | Value | ...
```

where:

`Count` is a long int representing the number of values to receive.

`Value` a double representing the channel value (for details,*Channel Name Descriptions* on page 12).

Info Packet

The format for the body of an Info packet is:

```
| Count | String | String | String | ...
```

where:

`Count` is a long int representing the number of values to receive.

`String` is the channel name without a terminating '\0' in the following format:

```
| Letters | Char | Char | Char | ...
```

where:

`Letters` is a long int representing the number of characters to read.

`Char` is a single-byte atomic C data type.

Real-time Data Channel Specifications

As described in the previous sections, in response to client requests, the Vicon RTE returns an Info packet containing the names of available channels of the real-time data and Data packets containing an array of real numbers (doubles) identifying the actual data contained in each of these channels.

This section lists the available channels and describes their encoded contents.

Channel Specifications

The format for Channel Name is:

`Name <Code>`

where:

`Name` is the name of a segment or a marker. A name is case sensitive. It may have an optional Subject prefix, for example:

```
SubjectName:SegmentName
SubjectName:MarkerName
```

The Name must not contain the special characters ' ', '-', '<', or '>'.

Important: The same name can be used more than once with different codes.

`<Code>` is additional semantic information about the channel type encoded in the channel name to enable the client to understand the meaning of the data.

For details of the available channels and their code descriptions, see *Channel Name Descriptions* on page 12.

Channel Name Descriptions

Table 2-3 lists the available channel names, gives the channel type codes for each, and describes the information supplied by that code.

Table 2-3: RTE Channel Name Descriptions

Channel Name	Channel Type Code	Description
Time X fps	The frame rate in frames per second (fps) where x:	
	<F>	frame rate for the system
Timecode †	The timecode for the current frame:	
	<TC-V>	timecode validity
	<TC-R>	timecode rate
	<TC-H>	timecode hours
	<TC-M>	timecode minutes
	<TC-S>	timecode seconds
	<TC-MS>	timecode milliseconds
	<TC-F>	timecode frames
	<TC-OFF>	timecode offset (sub-frames)
Marker	The positions of the reconstructed markers for the given sample:	
	<P-X>	X axis position
	<P-Y>	Y axis position
	<P-Z>	Z axis position
	<P-O>	Occlusion state: 1.0=occluded 0.0=visible
Global Body †	The kinematic state of a body in the global coordinate system (rotation for this 3 DOF body is expressed as an angle-axis triplet; for details, see *Channel Code Specifications* on page 14):	
	<A-X>	X angle-axis rotation
	<A-Y>	Y angle-axis rotation
	<A-Z>	Z angle-axis rotation
	<T-X>	X translation
	<T-Y>	Y translation
	<T-Z>	Z translation

Table 2-3: RTE Channel Name Descriptions

Channel Name	Channel Type Code	Description
Local Body	The kinematic state of a body in the local coordinate system (rotation for this 3 DOF body is expressed as an angle-axis triplet; for details, see *Channel Code Specifications* on page 14):	
	<ba-X>	X angle-axis rotation
	<ba-Y>	Y angle-axis rotation
	<ba-Z>	Z angle-axis rotation
	<bt-X>	X translation
	<bt-Y>	Y translation
	<bt-Z>	Z translation
Free Joint	The kinematic state of a free joint, such as the pelvis, in the local coordinate system (rotation for this 3 DOF body is expressed as an angle-axis triplet; for details, see *Channel Code Specifications* on page 14):	
	<r-X>	X angle-axis rotation
	<r-Y>	Y angle-axis rotation
	<r-Z>	Z angle-axis rotation
	<t-X>	X translation
	<t-Y>	Y translation
	<t-Z>	Z translation
Ball Joint	The kinematic state of a ball joint in the local coordinate system (rotation for this 3 DOF body is expressed as an angle-axis triplet; for details, see *Channel Code Specifications* on page 14):	
	<r-X>	rotation around X axis
	<r-Y>	rotation around Y axis
	<r-Z>	rotation around Z axis
2DOF Joint	The kinematic state of a 2-degree of freedom joint in the local coordinate system:	
	<r-A>	rotation around the primary axis
	<r-B>	rotation around the secondary axis
Hinge Joint	The kinematic state of a hinge joint in the local coordinate system:	
	<r-H>	rotation around the hinge axis

†Not supported by Nexus 1.x.

Channel Code Specifications

Table 2-4 describes how values are expressed for different types of channel codes.

Table 2-4: RTE Channel Code Specifications

Value Type	Unit of Measure	Description
Angle	radians	
Rotation		

3 DOF Joints	angle-axis triplets (also known as Finite Axis of Rotation and Exponential Maps)	The normalized vector is the axis of rotation. The magnitude of the vector is the amount of rotation (in radians).
		For example, given the angle axis parameters:
		`(ax,ay,az)`
		Rotation about axis is:
		`theta = sqrt(ax*ax+ay*ay+az*az)`
		Axis of rotation is:
		`A = (ax,ay,az) / theta`
		You can convert these values to Euler angles if desired. For details on doing this, see *Converting an Angle-Axis Triplet to a Rotation Matrix*.

Translation	millimeters	X,Y,Z components are as defined in the Vicon global coordinate system.

Converting an Angle-Axis Triplet to a Rotation Matrix

Use the following code snippet to convert an angle-axis triplet (ax,ay,az) to a rotation matrix (M):

```
double theta;
double c, s, x, y, z;
double M[3][3];
theta = sqrt( ax*ax + ay*ay + az*az );

if (theta < 1e-15)
{
   M[0][0] = M[1][1] = M[2][2] = 1.0;
   M[0][1] = M[0][2] = M[1][0] = M[1][2] = M[2][0] = M[2][1] = 0.0;
}
else
{
    x = data[iBody->RX]/len;
    y = data[iBody->RY]/len;
    z = data[iBody->RZ]/len;

    c = cos(len);
    s = sin(len);

    M[0][0] = c + (1-c)*x*x;
    M[0][1] =     (1-c)*x*y + s*(-z);
    M[0][2] =     (1-c)*x*z + s*y;
    M[1][0] =     (1-c)*y*x + s*z;
    M[1][1] = c + (1-c)*y*y;
    M[1][2] =     (1-c)*y*z + s*(-x);
    M[2][0] =     (1-c)*z*x + s*(-y);
    M[2][1] =     (1-c)*z*y + s*x;
    M[2][2] = c + (1-c)*z*z;
}
```

For further explanations of rotation parameterization, see *References* on page 22. If necessary, you can convert this rotation matrix to an Euler angled as described in *Converting a Rotation Matrix to an Euler Angle* on page 16.

Converting a Rotation Matrix to an Euler Angle

Use the following code snippet to convert a rotation matrix (M) to an Euler angle:

```
double eulerX, eulerY, eulerZ

assert(fabs(M[0][2]) <= 1);
eulerY = asin(-M[2][0]);

if(fabs(cos(y)) > std::numeric_limits<double>::epsilon() ) //
cos(y) != 0 Gimbal-Lock
{
  eulerX = atan2(M[2][1], M[2][2]);
  eulerZ = atan2(M[1][0], M[0][0]);
}
else
{
  eulerZ = 0;
  eulerX = atan2(M[0][1], M[1][1]);
}
```

Using the Sample Code

The supplied sample code, *ExampleClient.cpp*,contains simple examples of the various elements you will need to use the Vicon RealTime stream:

Table 2-5: Sample Code Contents

Section	Illustrates
1	Initializing socket
2	Requesting and receiving the Info packet
3	Parsing the Info packet to determine what data is available
4	Getting the data
5	Looking up the parts of the data that interest you
6	Interpreting the data
7	Closing down the sockets

Typically, section 1 will be performed on start up, sections 2 and 3 will be performed on connection. sections 4-6 are performed repetitively while the program is connected, and section 7 is performed during shut down.

Using the Test Client

You can use the supplied test client application, *TestClient.exe*, to view the real-time data stream. Seeing the type of information contained in the Info and Data packets can help you to understand the client-server communications protocol, and so guide you in writing your application.

To use the Test Client:

1. Start your Vicon application software and ensure it is connected to the Vicon system and in Live mode.
2. In Windows Explorer, navigate to the TestClient sub folder under your RealTime SDK folder (by default, *C:\Program Files\Vicon\SDK \RealTime\TestClient*) and double-click *TestClient.exe*.

   The TestClient window, shown in Figure 2-2, is displayed.

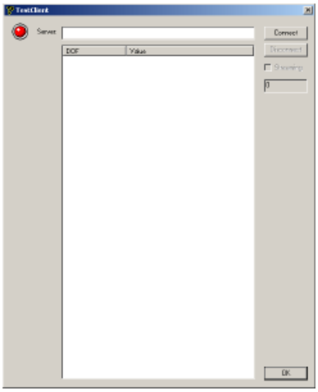

Figure 2-2: TestClient window

3. In the **Server** field, enter the machine name or the IP address of the Ethernet network card on the computer on which the RTE server is running.

   Tip

   If this is the same computer on which the TestClient application is running, you can leave this field blank and localhost will be assumed.

4. Click **Connect**.

   When the Test Client is successfully connected to the RTE server, the red status light turns green and the data in the real-time stream is displayed in the window.

5. In the **DOF** column, view the data in the Info packet, and in the **Value** column, view the data in the Data packets.
6. Select the **Streaming** check box to acquire data from the RTE through frame-by-frame polling; clear the check box to acquire data through continuous streaming (for details on these methods, see *Creating a Client To Acquire Data from the RTE Server* on page 7.
7. View the current frame delivery rate (in fps) in the field below the **Streaming** check box.
8. Click **Disconnect** to stop real-time data streaming.
9. Click **OK** to close the TestClient window.

Using the RealTime Viewer

If you are using Vicon iQ or Workstation in a distributed architecture, you can use the supplied RealTime Viewer application, *RTViewer.exe*, to view the real-time data streaming across the network. Seeing the RealTime data in 3D on a remote PC enables you to quickly confirm that real-time data is streaming as intended. You can display RealTime Data live from the Vicon cameras, or stream data from a previously saved *.x2d* file.

Important

If you are using Vicon Nexus, you can view the display of real-time data in a 3D Perspective view pane, so you do not need to use the separate RealTime Viewer application.

To use the RealTime Viewer:

1. Start your Vicon application software.
2. Open the desired session node in the motion capture database.
3. Select the fully calibrated Vicon Skeleton (*.vsk*) file for the subject whose motion you want to capture.
4. Configure the RealTime Engine to specify the:
   - Data streaming mode: Live or from File
   - Required level of output: At a minimum, reconstructed and labeled markers. Specify kinematics if you want to bones as well as markers.

5. Connect the Vicon application software to the real-time data stream:
   - Vicon iQ or Workstation: Click the **Connect System** button on the RealTime Engine Control Bar
     .
   - Vicon Nexus: Click the **Live** button on the **Resources** pane
     Live .
6. In Windows Explorer, navigate to the RTViewer sub folder under your RealTime SDK folder (by default, *C:\Program Files\Vicon\SDK\RealTime \RTViewer*) and double-click *RTViewer.exe*.

   The Vicon - Real-time Viewer window, shown in Figure 2-3, is displayed.

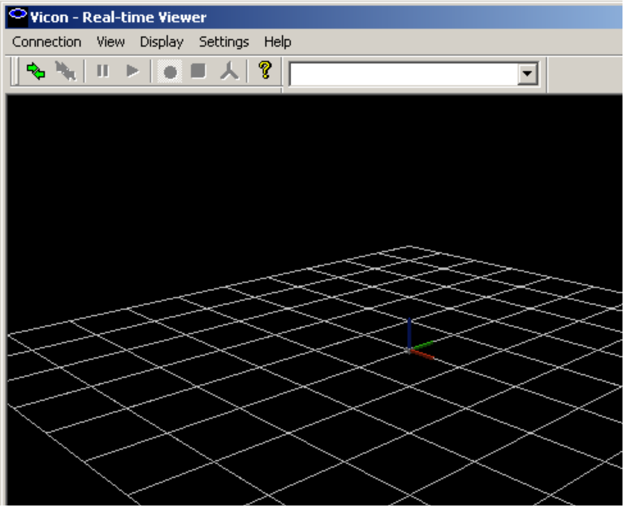

Figure 2-3: RTViewer window

7. From the tool bar at the top of the window, click the **Connect** button
   .
8. In the displayed **Connect to server** dialog box, shown in Figure 2-4, enter the machine name or the IP address of the Ethernet network card

on the computer on which the RTE server is running and then click **Connect**.

Tip

If the RTE and your Vicon application software are running on the same computer, you can leave this field blank and localhost will be assumed

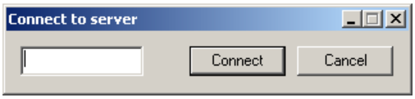

Figure 2-4: Connect to server dialog box

The RT Viewer requests all channels from the RTE and displays them as objects in the 3D perspective view pane in the **Real-time Viewer** window. By default, RT Viewer uses an inferred skeleton, so you should see a human figure moving in the window.

9. You can manage the display of real-time data using the following tool bar buttons:

⏸	Freeze the data display in RT Viewer
▶	Continue data streaming display in RT Viewer
●	Toggle the display of Markers on and off
🟥	Toggle the display of bodies (boxes representing bones) on and off
⅄	Toggle the display of the bone axes on and off
▾	Select an object from the list of those available in the real-time data stream

10. Click **Disconnect** to stop real-time data streaming ⊘.

User Assistance Materials

Vicon provide user assistance materials to support your use of our products:

- *Product Documentation*
- *Vicon Online Support* on page 22

Published research papers provide related information you may find useful:

- *References* on page 22

Product Documentation

The following product documentation is available to familiarize you with the features and functionality in RealTime SDK Version 1.1:

- **Vicon RealTime SDK Product Guide—Foundation Notes**

  This document, which provides the information you need to know about the current release of the RealTime SDK. It contains essential product information: describes the current release, provides prerequisites, installation, licensing, startup, and basic usage instructions; details new features; and identifies any known issues.

  To access it, from the Windows Explorer, navigate to *C:\Program Files \Vicon\SDK\RealTime\Documentation* and select *RealTimeSDK_Tarsus_ProductGuide_Foundation.pdf*.

- **Vicon iQ / Workstation RealTime Engine System Option**

  This book provides detailed information on the Vicon RealTime Engine, which is accessible from the Vicon application software motion capture and analysis software and whose output can be viewed and incorporated in third-party visualization software. The book describes the features and functionality of the RealTime Engine and provides instructions on its operation in Vicon application software.

  To access them, from the Windows **Start** menu, point to **All Programs**, then **Vicon**, then **Documentation**, then **Books**, then **System Option**, and select the desired book.

- **Vicon Nexus Information System**

  This online help system provides detailed information on the Vicon Nexus motion capture and analysis software. The help describes the features and functionality of the Vicon Nexus and provides instructions on its operation.

  To access it, from the Nexus **Help** menu, select **Contents**.

- **Vicon MX Hardware System Reference**

  This book provides detailed information on the features and functionality of each hardware component of Vicon MX, an integrated system for motion measurement and analysis. It also illustrates some common Vicon MX architecture models, showing you how you can combine Vicon MX and third-party elements to create an architecture that meets your application requirements.

  To access this book, from the Windows **Start** menu, point to **All Programs**, then **Vicon**, then **Documentation**, then **Books**, then **System Reference**, and select the book.

These documents are all installed in PDF format (requires Adobe Acrobat version 5.0 or later).

Vicon Online Support

If you are a licensed Vicon user and have a valid Vicon System Maintenance Agreement, you can access the Vicon Online Support knowledge base at www.vicon.com/support.

Vicon Online Support provides the following online support resources:

- **Downloads**

  Obtain latest firmware and other software patches, models and scripts, and product documentation.

- **FAQs**

  Locate topics providing answers to frequently asked questions about Vicon hardware, software, plug-ins, and licensing as well as third-party software.

- **Cases**

  Submit your own question or report a problem if you cannot locate the information you need in the FAQs, then track responses to your questions and updates to your problems.

References

These research publications provide supporting information on the scientific basis and validation of this RealTime SDK:

- GRASSIA, F. SEBASTIAN. *Practical Parameterisation of Rotations Using the Exponential Map*. Journal of Graphics Tools, (1998) 3(3):29-48.