Accessing Vicon Tracker data from Simulink

If you are working in an environment that supports the use of the Vicon DataStream SDK (where TCP/IP is supported), you can use the SDK from within a Simulink block (S-function) to access data streamed from Vicon Tracker.

In addition, Vicon Tracker 3 includes a UDP stream that contains object translation and rotation data. If you are working in an environment that supports only UDP and therefore cannot use the Vicon DataStream SDK to access Tracker object data from Simulink, you can access Tracker positional data from the UDP stream.

Important

The UDP stream contains only a small subset of the data that is available via the Vicon DataStream SDK, so if possible, use the Vicon DataStream SDK in preference to the UDP stream.

To help you access Tracker data from Simulink, examples of both types of access are installed with Tracker. They can be found in the following default location:

C:\Program Files\Vicon\Tracker3.#\Simulink

For more information, see:

Prerequisites for using Simulink with Vicon Tracker

To use Simulink with Tracker, ensure the following requirements are met:

- You are familiar with Simulink.
- For compiled S-functions, access to and proper configuration of a compiler in MATLAB.

  Tip

  You can use configurations other than compiled S-functions (such as using MATLAB code within the Simulink block) but there may be a performance disadvantage to using interpreted code. Alternative configurations have not been investigated or tested by Vicon.

- A properly installed C or C++ compiler. Microsoft Visual Studio 2013 was used during the development of the examples.
- Installed and licensed Instrument Control Toolbox. This toolbox is licensed separately (i.e. it is not part of the Simulink license). It is needed for receiving UDP packets only.

Back to Top

About the UDP stream

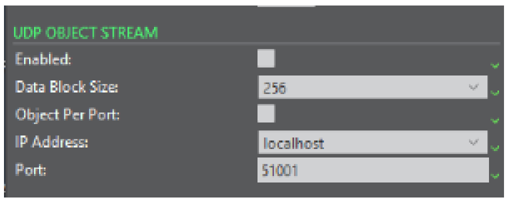
The UDP stream outputs translation and rotation information for active objects in Vicon Tracker.

To access data from this stream you must write a 'client' to access the stream and parse the data block to access its contents. The example clients provided with Tracker illustrate the block parsing and some possible configurations for block outputs.

For each frame in Tracker, one or more data blocks are sent. The number of blocks per frame sent is dependent on:

- The data block size setting
- The number of active objects
- The object-per-port setting

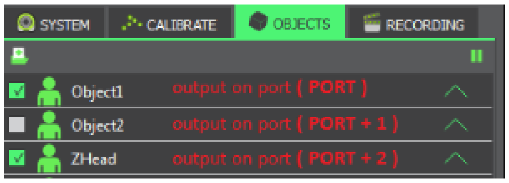
The UDP Stream parameters are stored in the *.system* file:



UDP OBJECT STREAM

Enabled:	☐
Data Block Size:	256
Object Per Port:	☐
IP Address:	localhost
Port:	51001

On the **System** tab, when you click on the **Local Vicon System** node, the following settings are available in the **UDP Object Stream** section of the **Proper ties** pane.

Setting	Description
Enabled	If selected, starts the UDP streaming of data. Unlike the data stream, the UDP stream does not maintain client connection information. If selected, data is output whether or not there are any connected clients.
Data Block Size	The size of the UDP datagrams (data blocks). Ensure the value selected matches the expected value for the datagram size in the client program. Options are 256, 512, and 1024.
Object Per Port	If cleared, all objects are output on the same port. If selected, each object is output on its own UDP port. Port assignments are made whether or not the object is active. The following image shows how port numbers are assigned:  SYSTEM CALIBRATE OBJECTS RECORDING Object1 output on port (PORT) Object2 output on port (PORT + 1) ZHead output on port (PORT + 2)
IP Address	The network address used to broadcast the data.
Port	The starting port for UDP streaming. If **Object Per Port** is selected, this is the starting port number. If **Object Per Port** is cleared, this is the output port for all objects.

Example UDP Packet contents table

The following UDP Packet contents table provides more technical detail about the layout and content of the UDP stream. You may find this useful if, for example, you want to use the UDP stream, but do not want to use it with Simulink. It is taken from *DataBlock.h*, which is one of the Simulink example files:

Byte offset	Content	Comment
0-3	Frame Number	nnnn
4	ItemsInBlock	2
5	ItemHeader:ItemID	0 (0 for object data. Other object types not currently supported.)
6-7	ItemHeader:ItemDataSize	72
8-31	TrackerObject:ItemName	'O''b''j''e''c''t''1'00000000000000000
32-39	TrackerObject:TransX	
40-47	TrackerObject:TransY	
48-55	TrackerObject:TransZ	
56-63	TrackerObject:RotX	
64-71	TrackerObject:RotY	
72-79	TrackerObject:RotZ	
80	ItemHeader:ItemID	0 (0 for object data. Other object types not currently supported.)
81-82	ItemHeader:ItemDataSize	72
83-106	TrackerObject:ItemName	'O''b''j''e''c''t''2'00000000000000000
107-114	TrackerObject:TransX	
115-122	TrackerObject:TransY	
123-130	TrackerObject:TransZ	
131-138	TrackerObject:RotX	

139-146	TrackerObject:RotY	
147-154	TrackerObject:RotZ	

Object data output by the UDP stream

Data that is output matches the Vicon DataStream SDK values for the same frame. The UDP stream contains no axis mapping options.
For each object six values are output:

- Translation X, Y, and Z
  The values match the values received from GetSegmentGlobalTranslation through the Vicon DataStream SDK.

- Rotation X, Y, and Z
  The values match the values received from GetSegmentGlobalRotationEulerXYZ through the Vicon DataStream SDK.

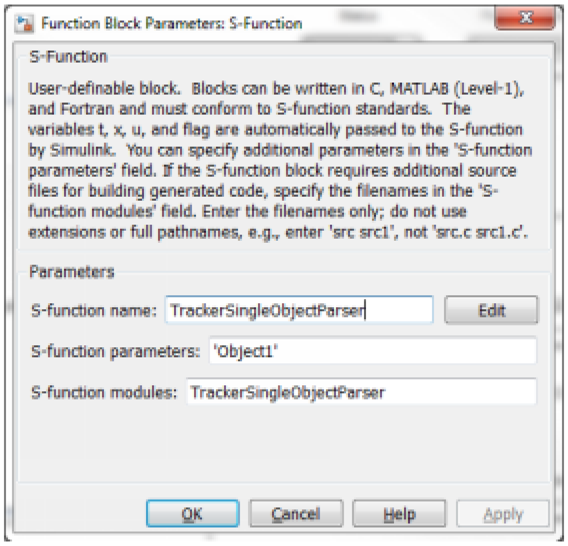
Back to Top

About the Simulink examples provided with Vicon Tracker

All of the examples consist of an S-function along with a Simulink model showing a block using the custom S-function in a simulation. Note the following points:

- Simulink models are stored in files with a .mdl extension
- Examples were developed and tested using 64-bit versions of MATLAB/Simulink/Vicon DataStream SDK. (You cannot mix 32- and 64-bit code.)
- Single object and multiple object examples on the same port are provided using the Vicon DataStream SDK as well as the UDP stream. The only difference in these examples is the method of data access.
- An additional example for the UDP stream is provided, which illustrates the Object Per Port functionality. This functionality is not available using the Vicon DataStream SDK.
- The examples use block parameters to specify the object names to be output.
  - String parameters are surrounded by single quotes.
  - Multiple parameters are separated by commas.
  - To access block parameters, double-click on the block in the model.

Single block parameter:



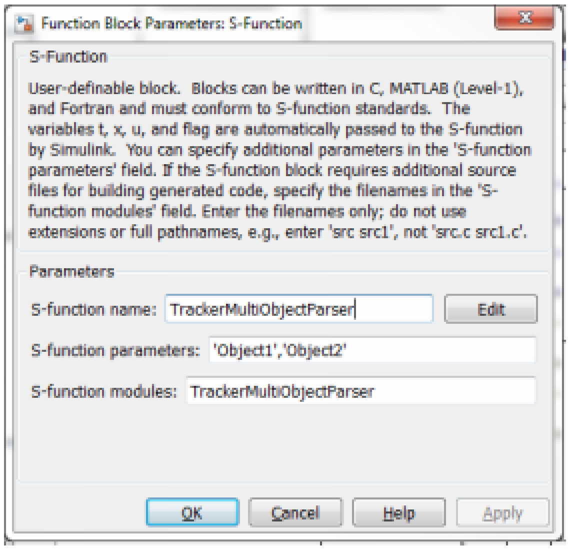
User-definable block. Blocks can be written in C, MATLAB (Level-1), and Fortran and must conform to S-function standards. The variables t, x, u, and flag are automatically passed to the S-function by Simulink. You can specify additional parameters in the 'S-function parameters' field. If the S-function block requires additional source files for building generated code, specify the filenames in the 'S-function modules' field. Enter the filenames only; do not use extensions or full pathnames, e.g., enter 'src src1', not 'src.c src1.c'.

S-function name: TrackerSingleObjectParser

S-function parameters: 'Object1'

S-function modules: TrackerSingleObjectParser

Multiple block parameters:


Function Block Parameters: S-Function

S-Function

User-definable block. Blocks can be written in C, MATLAB (Level-1), and Fortran and must conform to S-function standards. The variables t, x, u, and flag are automatically passed to the S-function by Simulink. You can specify additional parameters in the 'S-function parameters' field. If the S-function block requires additional source files for building generated code, specify the filenames in the 'S-function modules' field. Enter the filenames only; do not use extensions or full pathnames, e.g., enter 'src src1', not 'src.c src1.c'.

Parameters

S-function name: TrackerMultiObjectParser

S-function parameters: 'Object1','Object2'

S-function modules: TrackerMultiObjectParser

Back to Top

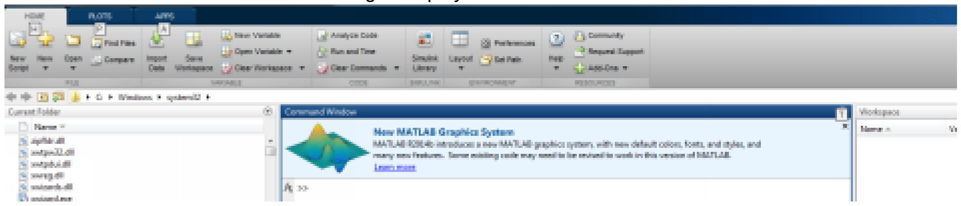
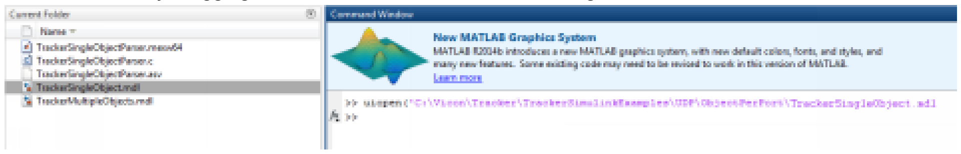
How to run the Simulink examples

The following steps described how to run one of the Simulink examples provided with Vicon Tracker, which demonstrate how to obtain Vicon Tracker data from Simulink.

Important

When you compile the code for the custom blocks, files are created in the same folder as your source file. It is recommended that you copy the example files to a folder other than the Tracker installation folder before compiling, running, or modifying the example files.

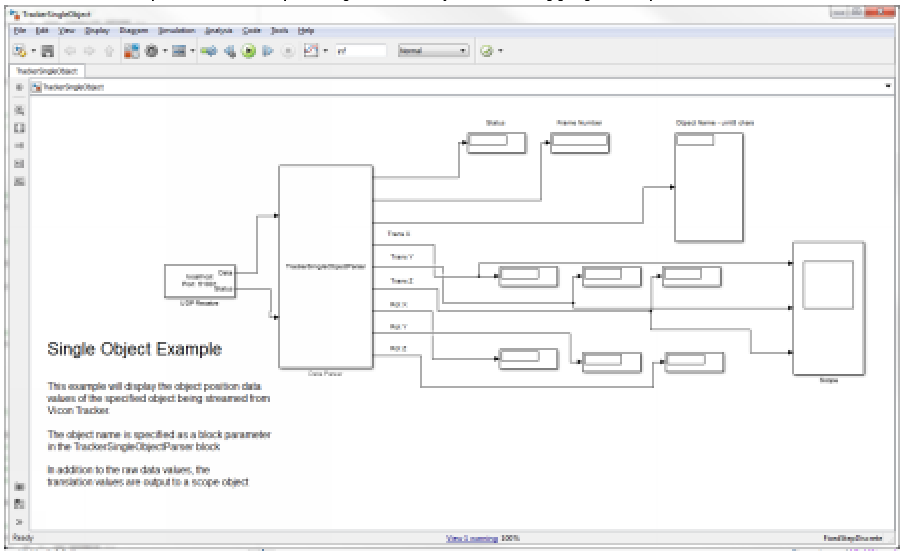
To run an example:

1. Ensure Tracker is running and streaming data.
2. The examples reference objects named Object1 and/or Object2. If you need to change the objects that are displayed, modify the block parameters to reference the desired object(s).
3. Open MATLAB.
   A MATLAB window similar to the following is displayed:
   
   New MATLAB Graphics System
   MATLAB R2014b introduces a new MATLAB graphics system, with new default colors, fonts, and styles, and many new features. Some existing code may need to be revised to work in this version of MATLAB.
   Learn more
4. Change your current folder to one containing the example you want to run.
5. Compile the example file. For more information, see the specific compile information below. Vicon DataStream SDK examples need to link in the proper Vicon DataStream SDK file as well.
6. Load the model by dragging the desired .mdl file from the file listing to the command window.
   
   Current Folder
   TrackerSingleObjectParser.mexw64
   TrackerSingleObjectParser.c
   TrackerSingleObjectParser.asv
   TrackerSingleObject.mdl
   TrackerMultipleObjects.mdl

   New MATLAB Graphics System
   MATLAB R2014b introduces a new MATLAB graphics system, with new default colors, fonts, and styles, and many new features. Some existing code may need to be revised to work in this version of MATLAB.
   Learn more

   ```
   >> uiopen('C:\Vicon\Tracker\TrackerSimulinkExamples\UDP\ObjectPerPort\TrackerSingleObject.mdl
   >>
   ```

This issues a uiopen command passing in the file you are dragging and opens the model window.



Single Object Example

This example will display the object position data values of the specified object being streamed from Vicon Tracker.

The object name is specified as a block parameter in the TrackerSingleObjectParser block.

In addition to the raw data values, the translation values are output to a scope object.

7. In the model window click the Play button to run the simulation.

> **Tip**
>
> If the model contains a scope block to draw a graph of the data, you must double-click on the scope to open it – it does not open automatically.

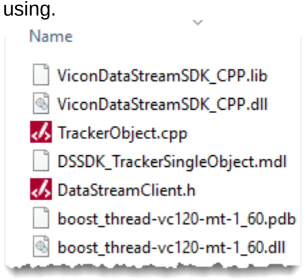
Back to Top

Vicon DataStream SDK examples

Before you use any of the Vicon DataStream SDK examples, they must be compiled and linked with the Vicon DataStream SDK.

To compile and link the Vicon DataStream SDK examples:

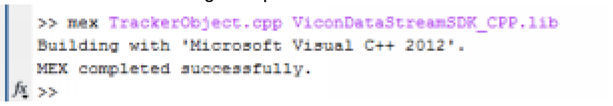
1. Copy the CPP files from the Vicon DataStream SDK installation folder to the folder containing the example files.
   Your file names might be slightly different from those in the following illustration, depending on the version of the Vicon DataStream SDK you are using.



Name

ViconDataStreamSDK_CPP.lib
ViconDataStreamSDK_CPP.dll
TrackerObject.cpp
DSSDK_TrackerSingleObject.mdl
DataStreamClient.h
boost_thread-vc120-mt-1_60.pdb
boost_thread-vc120-mt-1_60.dll

2. To compile and link C++ code, use the mex command. You need to compile the *.cpp* file and then link it with the Vicon DataStream SDK .lib file. You can do this in a single step that looks like this

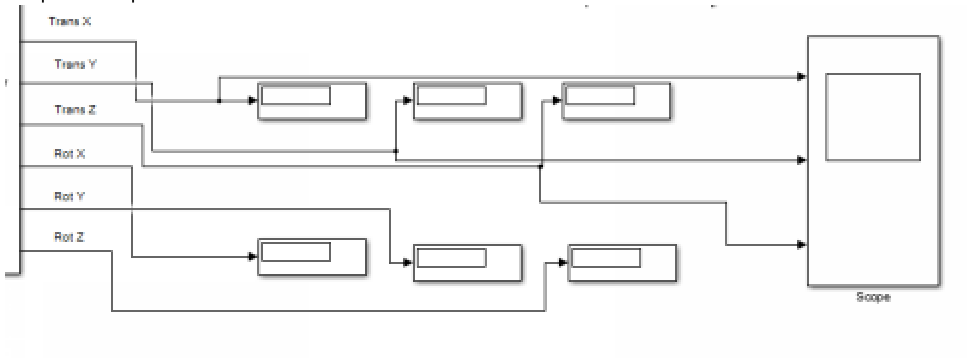


```
>> mex TrackerObject.cpp ViconDataStreamSDK_CPP.lib
Building with 'Microsoft Visual C++ 2012'.
MEX completed successfully.
>>
```

Successful compilation results in the creation of a file with a *.mexw64* extension.

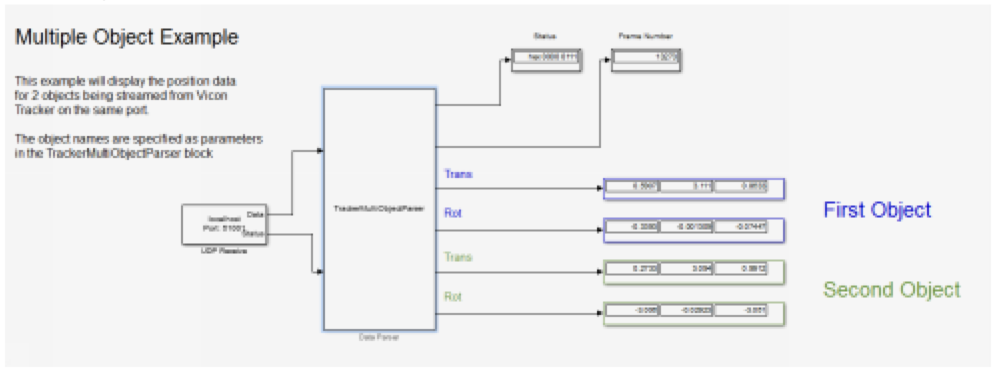
Vicon DataStream SDK SingleObject example

This example displays the positional information for an object (object name specified as block parameter). Each of the translation and rotation values is on a separate output.



Vicon DataStream SDK MultipleObjects example

This example displays the positional information for two objects (names specified as block parameters). Three values are provided for each of the four defined outputs.


Multiple Object Example

This example will display the position data for 2 objects being streamed from Vicon Tracker on the same port

The object names are specified as parameters in the TrackerMultiObjectParser block

Back to Top

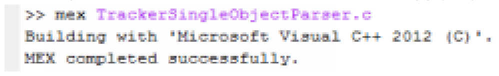
UDP stream examples

All examples use the Instrument Control Toolbox receiving UDP packets. This toolbox is licensed separately (i.e. it is not part of the Simulink license).

All of the example S-functions have been written in C. Before using any of these examples, they must be compiled.

To compile the examples:

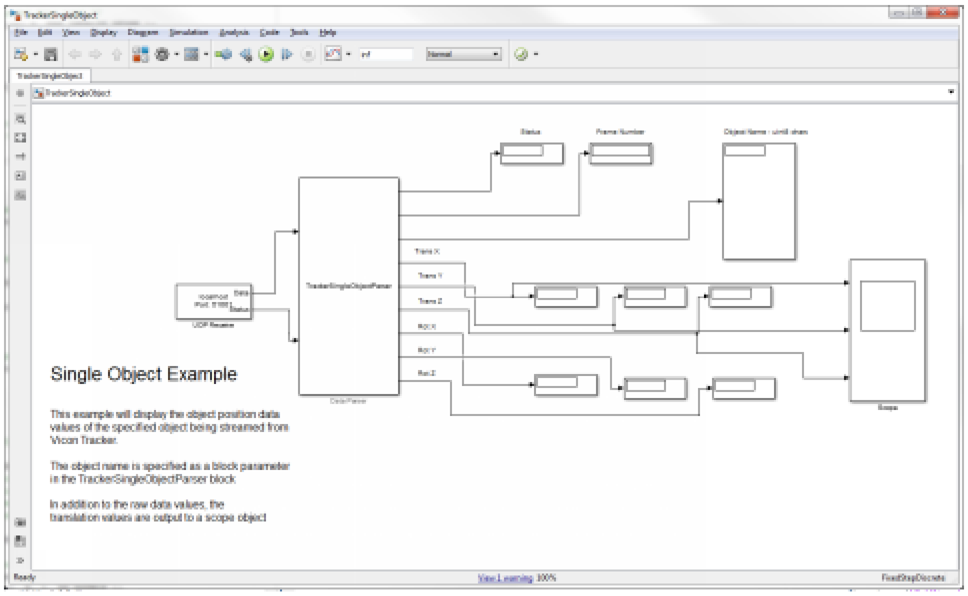
- To compile C code, use the mex command, supplying the name of the .c file as input, as shown in the following illustration:


```
>> mex TrackerSingleObjectParser.c
Building with 'Microsoft Visual C++ 2012 (C)'.
MEX completed successfully.
```

ObjectPerPort folder

This folder contains two models:
TrackerSingleObject.mdl and **TrackerMultipleObjects.mdl**

Both models use the same custom S-function, TrackerSingleObjectParser. The object parsing function looks for the object name passed in as a block parameter and displays the positional information for the object. Each of the translation and rotation values is on a separate output.



Single Object Example

This example will display the object position data values of the specified object being streamed from Vicon Tracker.

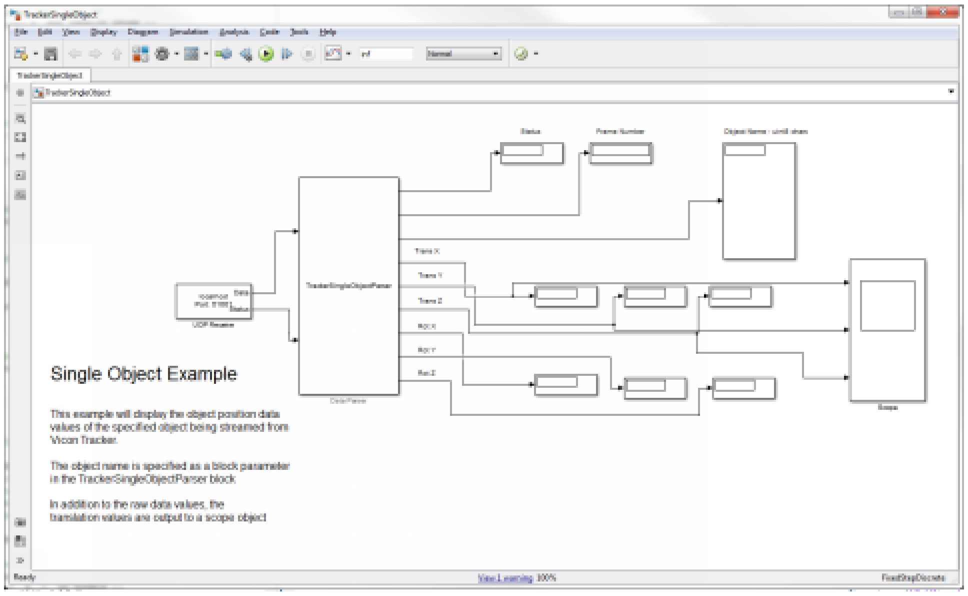
The object name is specified as a block parameter in the TrackerSingleObjectParser block.

In addition to the raw data values, the translation values are output to a scope object

You must configure the UDP Receive blocks in the models to match Tracker output with regard to data block size and port numbers.

UDP stream TrackerSingleObject example

This example displays the positional information for an object (object name specified as block parameter). Each of the translation and rotation values is on a separate output.



Single Object Example

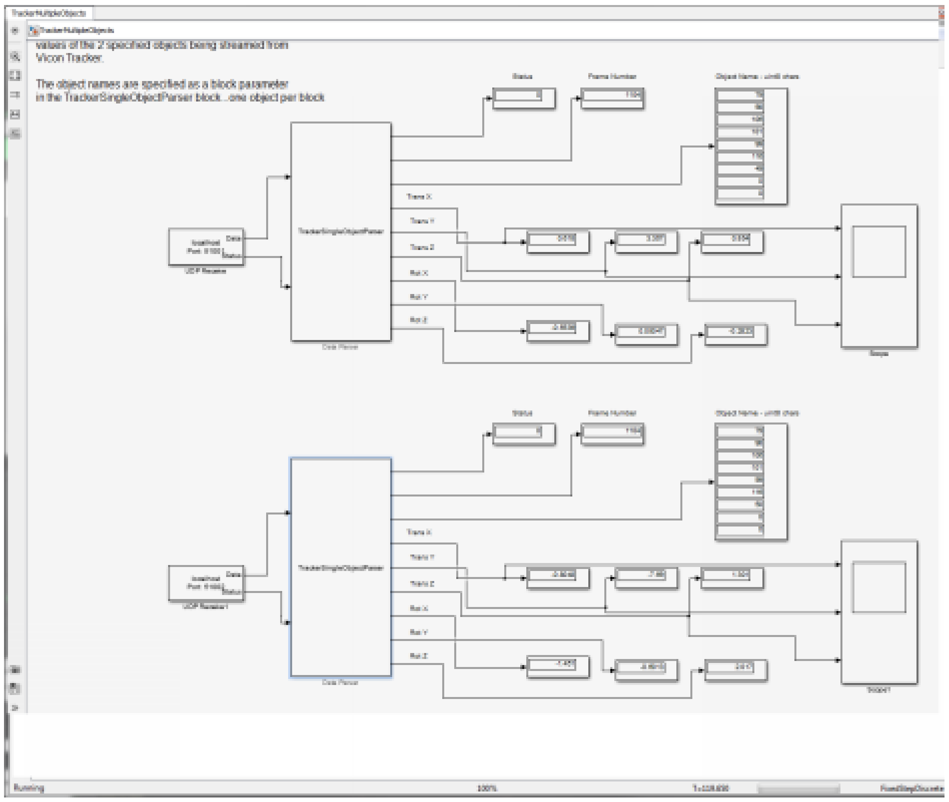
This example will display the object position data values of the specified object being streamed from Vicon Tracker.

The object name is specified as a block parameter in the TrackerSingleObjectParser block.

In addition to the raw data values, the translation values are output to a scope object

UDP stream TrackerMultipleObjects example

This model has two UDP receive blocks to receive data from Tracker, using the Object Per Port option.

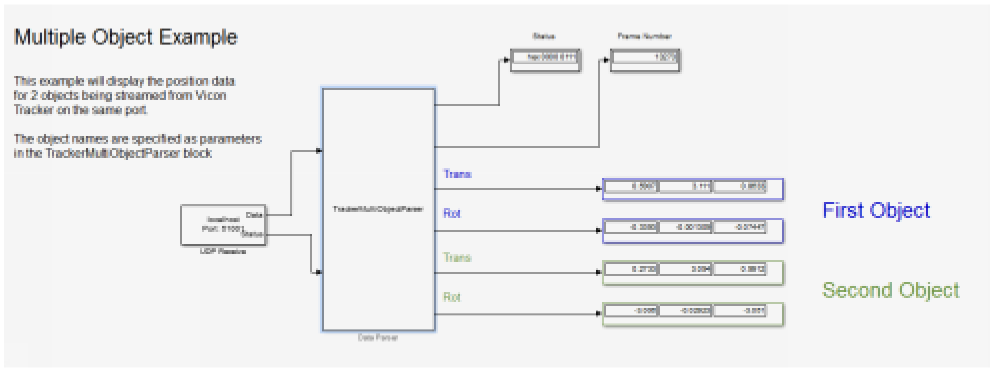


values of the 2 specified objects being streamed from Vicon Tracker.

The object names are specified as a block parameter in the TrackerSingleObjectParser block...one object per block

UDP stream MultipleObjectsSamePort example

This example displays the positional information for two objects (object names specified as block parameters). Three values are provided for each of the four defined outputs.



Multiple Object Example

This example will display the position data for 2 objects being streamed from Vicon Tracker on the same port.

The object names are specified as parameters in the TrackerMultiObjectParser block

Back to Top