755 Phoenix Drive, Ann Arbor MI, 48108, USA
Phone: 734 668-2930 • Fax: 734 668-2877 • Email: info@carsim.com
carsim.com

vs_sf VS Connect Server

Introduction 1
Prerequisites 1
Related Material 2
VehicleSim SDK 2
VehicleSim Dynamics Plugin for Unreal Engine 2
CarSim Example Dataset 2
VS Connect Server Configuration 2
Parsfile Keywords 2
Server Capabilities 4
Incoming Data (Imports) 4
Outgoing Data (Outputs) 5
Logging and Troubleshooting 6

This document describes the VS Connect Server features of the vs_sf Simulink s-function in VehicleSim version 2022.1.

Introduction

The vs_sf s-function has been enhanced to include VS Connect Server functionality. When this feature is enabled and properly configured, a VS Connect Client can connect to vs_sf while it is running within Simulink. Once connected, the Client can send data which are applied as Imports to the VS Solver, and request to receive Output data from the solver.

The **vs_sf VS Connect Server** functionality is distinct from, and complementary to, the stand-alone **VS Connect S-function** which was previously released. The **VS Connect S-function** is a generic VS Connect Client implemented as a Simulink block, and it may be configured to connect to (and exchange data with) any VS Connect Server, including the **vs_sf VS Connect Server** described in this document.

Prerequisites

The reader is expected to be familiar with VS Connect and its concepts such as Nodes, Links, Peers, Schemas, Fields, Records, and Contracts.

The reader should also be familiar with Imports and Outputs of the VehicleSim solver.

For information on these topics, refer to the items in the **Related Material** section below.

Related Material

VehicleSim SDK

The following documents contain additional information related to VS Connect and can be found in the VS SDK, which is available for download here: https://www.carsim.com/users/vs_sdk

- The VS Connect API Reference Manual (VS_Connect_API.pdf)
- The VS Connect S-function (vs_connect_s_function.pdf)

VehicleSim Dynamics Plugin for Unreal Engine

The **VehicleSim Dynamics Plugin for Unreal Engine** contains VS Connect based functionality. Its documentation contains additional information about VS Connect. The Plugin, its documentation, example projects, and tutorials (including some which utilize VS Connect) can be downloaded here: https://www.carsim.com/users/unreal_plugin

CarSim Example Dataset

CarSim version 2022.1 includes an example dataset which utilizes the vs_sf VS Connect Server functionality described in this document:

Category: External Control, Wrappers

Dataset: Unreal Engine Live Animation

VS Connect Server Configuration

The vs_sf VS Connect Server is configured via Parsfile keywords. These keywords can be placed in any generic yellow field within the VehicleSim Browser dataset.

Parsfile Keywords

Table 1. Keywords for configuring the vs_sf VS Connect Server.

Keyword	Value	Description
`VSC_OPT_ENABLE_SERVER`	0 or 1	Disable (0, the default) or enable (1) or the VS Connect server.
`VSC_HOST_ADDR`	*string*	Host name or IP address of server.
`VSC_HOST_PORT`	*number*	The UDP/IP port on which the server will listen for incoming connection requests. Default is 4367.
`VSC_SHUTDOWN_TIMEOUT_MS`	*number*	An integer specifying the maximum amount of time, in milliseconds, the server is allocated to cleanly shutdown.
`VSC_MAX_CONNECTIONS`	*number*	The maximum number of simultaneous connections allowed.
`VSC_VEHICLE_NAME`	*string*	The VS Connect Object name for this solver.

VSC_OPT_ENABLE_SERVER

Specify either 0 to disable the VS Connect Server feature, or 1 to enable it. By default, the VS Connect Server feature is disabled.

VSC_HOST_ADDR

Optionally specify the hostname or IP address to use for the VS Connect Server. This is useful to specify which specific Ethernet interface to use if your computer has more than one. If not specified, the VS Connect server will automatically use the default ethernet interface.

VSC_HOST_PORT

Optionally specify the UDP/IP port for the VS Connect Server to use. The default port that will be used if this keyword is not used is 4367. If multiple VS Connect servers are present on the same Ethernet interface, or on the same machine if it has only one Ethernet interface, they must each use different ports. You may also need to specify a custom port setting if another program on your computer is already using port 4367.

VSC_SHUTDOWN_TIMEOUT_MS

The VS Connect asynchronous communication system maintains stateful connections between connected Nodes (a.k.a. Peers). This means that the normal disconnection process entails back-and-forth communication between any two connected peers.

The amount of time it takes for this disconnection process to cleanly complete is dependent on many factors, including: the speed and bandwidth of the connection, the speed of the machines, and the way in which the two programs have been written to use VS Connect.

This keyword allows the user to limit the amount of time that this VS Connect Server instance is permitted to use to attempt to cleanly disconnect its connections and shut down the server.

The default is 1500 milliseconds (1.5 seconds).

> **Note** In most common situations (communicating between processes on the same machine or between multiple machines on a high-speed high-bandwidth LAN), the disconnection process takes only a few milliseconds.

VSC_MAX_CONNECTIONS

This limits the number of connections that the VS Connect Server will accept. The default is 3.

VSC_VEHICLE_NAME

This keyword specifies the VS Connect Object name for this vehicle.

When a VS Connect client requests data associated with this vehicle, it must specify this same string as the VS Connect Object.

Server Capabilities

The **VS Connect** data synchronization library is a generic system that enables multiple processes on multiple machines to communicate asynchronously. Each program/process that utilizes VS Connect is free to define what data it can accept from connected Peers, and what data it can provide to connected Peers.

This section describes the data which the vs_sf VS Connect Server can accept and provide, and how it processes this data.

The vs_sf VS Connect Server does not, as of version 2022.1, support VS Connect **Time Synchronization** requests for incoming data. Incoming data is applied to CarSim Imports immediately when it is received, and the Simulink model execution is never throttled or blocked to await incoming data.

The vs_sf VS Connect Server does not monitor the Simulink execution speed, and it assumes that the Simulink model is executing at approximately the same rate as the remote simulation. Undesirable behavior may result if the Simulink simulation executes much faster or slower than real-time, or communications are established with a remote simulation that is running at a significantly different rate than the Simulink model.

For example, network buffer overflows may occur if Simulink is running much faster than the remote client expects. This is because the rate at which data is sent (the VS Connect Contract update rate) is based on simulation time, not wall time. If the remote Peer requests a high update rate, such as 1 millisecond (1,000 Hz) and Simulink is running at 10x real-time, the VS Connect server will attempt to send data at 100 microsecond intervals (10,000 Hz), which may overflow network buffers, clog network traffic, and overwhelm the remote application. This can result in lost data, reduced performance, and warning or error messages in the Diagnostic Viewer.

Incoming Data (Imports)

Incoming Contract Requests from VS Connect clients are expected to specify one or more solver Import variables. The Schema of the Contract Request should contain one Field description for each solver Import it will send. Each field should specify the following information:

Table 2. Incoming Field Properties.

Field Property	**Value**
Data Type	VSC_DATATYPE_FLOAT
Data Element Size (bits)	64
Element Count	1
Object Name	The name specified with the `VSC_VEHICLE_NAME` keyword.
Property Name	The name of the VS Solver Import.
Parameters	Optional Import parameter string.

For example, if the server has been configured with "`VSC_VEHICLE_NAME MyCar`", and a remote VS Connect Peer wishes to send data to add pressure in units of megapascals to the vehicle's brake master cylinder, with an initial import value of 0.0, it would specify the following Field parameters:

Data Type:	VSC_DATATYPE_FLOAT
Data Element Size:	64
Element Count:	1
Object Name:	MyCar
Property Name:	IMP_PCON_BK
Parameters:	Add 0.0 ; MPa

The vs_sf Server will translate this to a **VS Commands** instruction for the VS Solver to activate an Import. For this example, the VS Commands instruction will be:

```
IMPORT IMP_PCON_BK Add 0.0 ; MPa
```

Note that if the specified Import variable was already an active Import for the solver, the parameters are ignored, and incoming data is applied to the existing Import.

> **Note** Although the VS Connect Server may activate additional solver Imports that were not originally specified in the Parsfile, which will expand the solver's Import Array, this will not have any effect on the number or order of the Simulink block input ports.

When a Record (data update) is received from a Client, the Record's values are applied to the solver's Import Array. The last value received from the Client will be used for subsequent iterations of the simulation until a new value is received from the Client.

If an Import variable was already active and receiving values from a Simulink block input, the value supplied by the Simulink model to the block will be ignored and overridden by the last value received from the remote VS Connect Peer.

Outgoing Data (Outputs)

Outgoing Contract Requests from VS Connect Clients are expected to specify one or more solver Output variables. The Schema of the Contract Request should contain one Field description for each solver Output to be sent. Each Field description should specify the following information:

Table 3. Outgoing Field properties.

Field Property	**Value**
Data Type	`VSC_DATATYPE_FLOAT`
Data Element Size (bits)	64
Element Count	1
Object Name	The name specified with the `VSC_VEHICLE_NAME` keyword.
Property Name	The name of the VS Solver Output.
Parameters	*<n/a>*

For example, if the server has been configured with "`VSC_VEHICLE_NAME MyCar`", and a remote VS Connect Peer wishes to receive the forward velocity of the vehicle, the Remote Peer will send an Outgoing Contract Request which contains a single Field with the following parameters:

Data Type:	VSC_DATATYPE_FLOAT
Data Element Size:	64
Element Count:	1
Object Name:	MyCar
Property Name:	Vx
Parameters:	*<empty>*

The vs_sf Server will transmit the requested data (in the default units for the given Output) using the Update Schedule specified in the Contract Request.

> **Note** The VS Connect Server does not utilize the VS Solver Export array to access the solver Outputs. Outgoing VS Connect data has no effect on the Simulink block output ports.

Logging and Troubleshooting

The VS Connect Server within the vs_sf s-function writes log entries which may be useful while developing Simulink models and diagnosing communication problems. Log entries are written using Simulink's `ssPrintf()` function (a.k.a. `mexPrintf()`), and can be viewed using Simulink's **Diagnostic Viewer.**