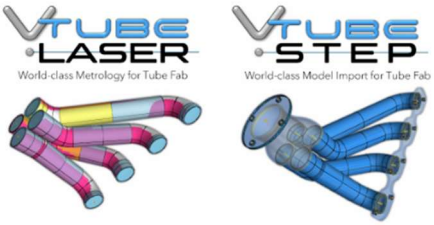
Command Server Tutorial



Date: Added in January 2019
Version: Added in VTube-STEP or VTube-LASER 2.9.18
Author: Michael Cone
Current Version: V4.5 B1961

Please note that this document is considered confidential. Each page footer also indicates that this information is confidential.

All versions of VTube now include a command server that watches for external commands to perform STEP and IGES model conversion tasks remotely and automatically.

The Command Server was a request from a CNC bender OEM that needed the ability to allow VTube-STEP to automatically import clean[1] STEP or IGES files and then send the converted data the CNC bender control automatically.

With the Command Server, customers can develop client programs that call out to and use the Command Server in order to automate STEP or IGES file conversion.

The server responds to five commands from the client software

The commands that VTube will respond to are:

1. Acknowledge VTube ready
2. Convert STEP/IGES model file to bender data
3. Minimize VTube
4. Maximize VTube
5. Close VTube

[1] "clean" means that there is nothing in the model except the tube. There are no brackets, end forms, holes, etc.

When does the server run?

The server runs whenever VTube-STEP or VTube-LASER is running. It takes a few seconds for the server to start watching for commands.

How does the server communicate?

To keep the design as straight forward and easy as possible, we created a system that communicates using files.

The three files are:

- Send file A client software process sends commands to VTube in this file.
- Response file VTube responds to the command in this file.
- Data file VTube puts converted data in this file.

The Send File

1. The server watches for a command file that your software creates with the fixed name: "**send_to_vtube_command_server.vtcom**".
2. Always save the send file to the fixed VTube configuration path location: "**c:\users\public\documents\vtube**".
3. One benefit of using this path is that it is easy to share the public folder over the network. Also, public folders are read/write accessible to all processes – so there are no read/write issues with this location.

What goes into the Send file?

What goes into the send file depends on what you want VTube to do.

Example Send File - To convert a step file, you would create a file with three lines like this:

```
linear unit = i
input file = C:\Users\Public\Documents\VTube\Demo Files\VTube-LASER Demo 5.stp
command = Autobuild Parametrics from Model File
```

- The first line shows what linear units you want VTube to respond in. The two parameters are "**i**" for inches or "**m**" for millimeters. The "**linear unit**" is a fixed label that must be spelled exactly that way with the space between the words for VTube to recognize it. Spaces between the equals sign and the parameter is ignored. Capitalization is ignored.

- The second line stores the complete path and filename of the STEP or IGES file to convert. The "**input file**" is a fixed label that VTube recognizes. Spaces between the equals sign and the parameter are ignored. Capitalization is not ignored. The complete filename that you indicate in this line can be anywhere that VTube also has access to. VTube will only be reading this file – so write access is not necessary for this model file.

- The third line gives the command. The word "**command**" followed by an equals sign followed by a parameter always means "do this now" to VTube. Therefore, it should be inserted on the third and last line only in the send file.

  In this case the command is "**Autobuild Parametrics from Model File**". You must use this wording exactly with spaces intact. Capitalization does not matter.

You can use this same form to convert an IGES file like this:

```
linear unit = m
input file = C:\Users\Public\Documents\VTube\Demo Files\VTube-LASER Demo 5.igs
command = Autobuild Parametrics from Model File
```

VTube knows which kind of model type to convert based on the filename *extension*. Valid extensions are STEP, STP, IGES, or IGS.

What the VTube server returns – the Response file

When a command is sent to the VTube command server, VTube will respond with one or two files, depending on the command. In all cases, VTube should send a **response** file.

File 1 - The Response File

1. The response filename is named "**response_from_vtube_command_server.vtcom**"
2. When you save a "**send_to_vtube_command_server.vtcom**" file in the VTube public folder, then VTube will load the file and attempt to interpret what to do, and then…
3. VTube should always answer – even if it does not understand the command. The answer will always be in the **response** file.

Example Response File - The format of the response file is similar for all commands. It can have one or two lines.

```
success = 1
output file = C:\Users\Public\Documents\VTube\output_data_from_vtube.vtcom
```

If you command VTube to convert a STEP or IGES file, and the conversion was successful, then you will see two lines in the response file like the example shown above.

- The first line starts with "**success =** ". Spaces and capitalization are ignored. The value after the equals sign is always either 1 or 0. "**1**" means true. "**0**" means false.

- The second line will be the path and filename of the Supravision file for your application to load. The file given in the second line points to a file that contains the bender data. For now, the standard **data** filename is **output_data_from_vtube.vtcom**. But that may change in the future, so it's good to parse in that actual filename VTube responds with.

  The Supravision file format is a legacy tube fabrication format that has been the standard for a few decades. It's not too hard to parse through for humans or computers. See the next page for a sample Supravision file.

The contents of a sample data file

This is a sample of the contents of a Supravision data file returned by VTube.

File 2 - Data File Example

```
$pnum VTube-LASER Demo 5

$unit E

$mxyz 7
13.79216702 0.99805199 20.24528337 0.00000000
15.11456470 -1.55112348 20.27990406 1.50000000
17.51219778 -0.62401898 23.00702714 1.50000000
20.49553014 -1.26072441 22.93158781 1.50000000
22.19442787 -0.80419622 20.50578403 1.50000000
25.10750698 -1.29884413 20.16784081 1.50000000
26.58721673 -0.71190426 21.72336617 0.00000000

$clra 5
1.49278595 0 0 0 0.00000000 0 0 0 85.19447398 0 0 0 1.50000000
1.56710854 0 0 0 -135.27252704 0 0 0 56.22935039 0 0 0 1.50000000
1.43314890 0 0 0 45.40730485 0 0 0 57.14444987 0 0 0 1.50000000
1.45550476 0 0 0 179.17530052 0 0 0 51.53892692 0 0 0 1.50000000
1.41613081 0 0 0 -44.16573753 0 0 0 58.13449525 0 0 0 1.50000000
1.39192402 0 0 0

$pvar 2
TBDI 0.75000000
TBWT 0.05000000
```

Please let us know if you need our document describing this data format.

What VTube returns if it cannot perform the command

For errors, the same response file format is used, but the "**success =**" parameter is set to "**0**".

Response File Example

```
success = 0
VTube-STEP could not load the STEP file
```

As is shown above, there will often be a text message on the second line that acts as an explanation for why the command failed.

About the other send commands

Here are the other commands to use in the **send** file. They are single-line commands that have no parameter lines above them.

Send File Example 1 - command = Acknowledge

The **response** is "**success = 1**" followed by "**VTube-STEP ready**" in the second line.

Send File Example 2 - command = Minimize

The **response** is "**success = 1**" followed by "**VTube-STEP window minimized**" in the second line.

Send File Example 3 - command = Maximize

The **response** is "**success = 1**" followed by "**VTube-STEP window maximized**" in the second line.

Send File Example 4 - command = Close

The **response** is "**success = 1**" followed by "**VTube-STEP is closing**" in the second line.

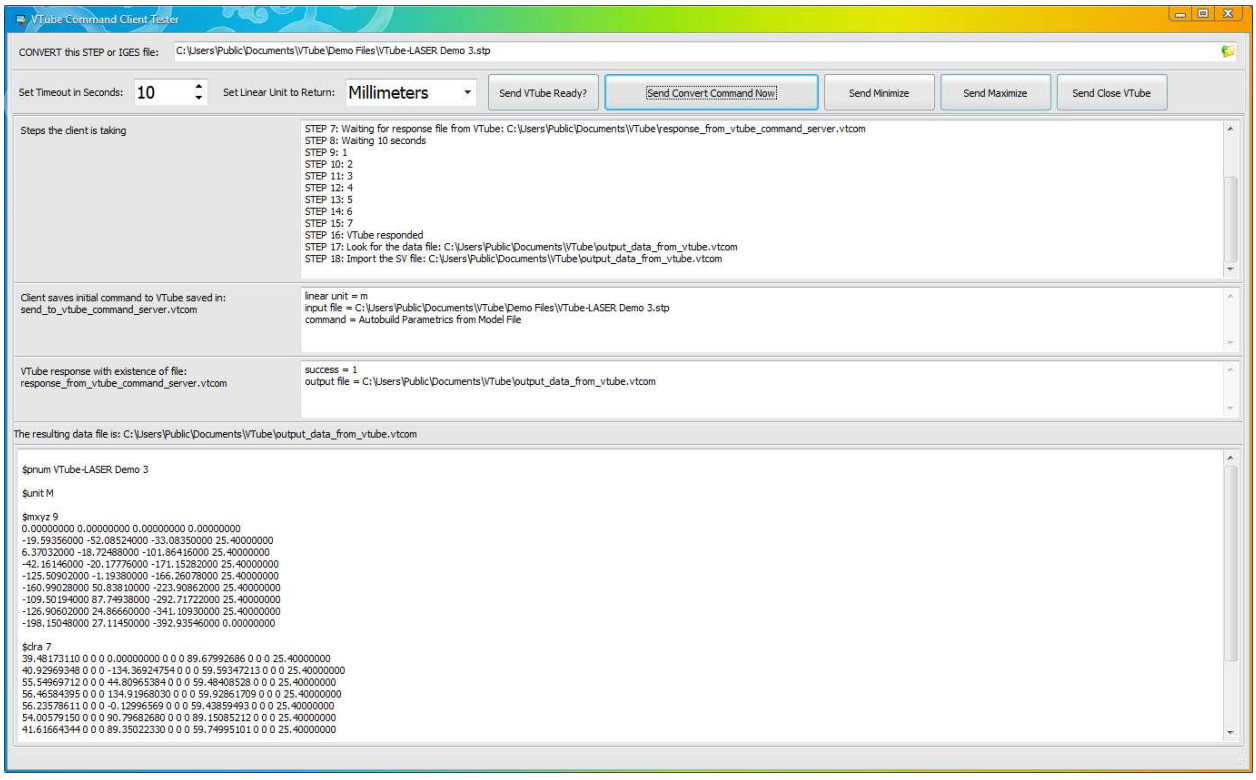
Suggested timeout

It's a good practice to calculate and use a timeout to stop waiting for data from VTube in case something goes wrong on the VTube side and VTube does not answer at all. A recommended minimum wait time is 10 seconds. Our test computer runs at this speed:

25 bends	0.75 seconds per bend for a total of 18 seconds
50 bends	0.86 seconds per bend for a total of 43 seconds
100 bends	1.25 seconds per bend for a total of 125 seconds

VTube Command Client for Testing

We include a VTube Command Client program that communicates with the Command Server so that you can experiment sending commands to the server. This program is included in the VTube program folder under the "c:\Program Files (x86)\VTube (version you are using)" folder.



VTube – commands to the server override everything

It is important to understand that using the Command Server will completely override whatever a VTube user is currently working on in VTube. Be careful to be sure you mean to send a command to a VTube instance that is being used by another operator.

For example, any data that is currently in VTube will be cleared automatically in response to the conversion commands sent by any client software.