

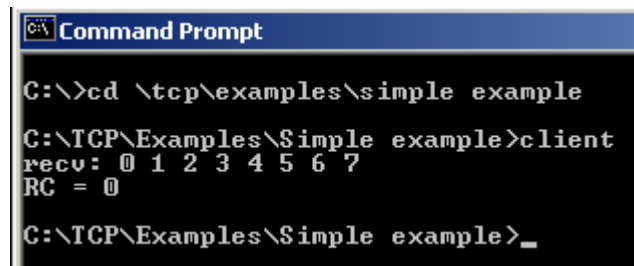
Thank you for your interest in our TCP/IP module. This document will show you step-by-step how to load and start the demo programs on your camera and PC. Before continuing please make sure, that you have a VC Ethernet camera VC20xxE connected to your PC. The default camera's IP address is 192.168.0.65. This is the default address, used in the respective PC client executables. If you have camera with different IP address, you should specify IP address when you start the PC demo programs, for example:

```
client 192.168.0.66
```

1. Simple example

1. Start the supplied terminal program – attoTerminal.
2. Choose the appropriate connection method and fill in the required details.
3. Click “**Connect**” and wait for the terminal to connect to the camera.
4. Click “**Upload MSF**”
5. Navigate to the “*Simple example*” folder inside the “*Examples*” folder and select SERVER.MSF
6. Click “OK” and wait for the upload to finish.
7. Open a command prompt window and navigate to the “*Simple example*” folder
8. Using the terminal start the server program on the camera – type server and hit “**Enter**” in the terminal window.
9. Now you have 9 seconds to establish connection with the camera. Go to the command prompt window and start client.exe. If you have changed the default IP address of the camera, you should start the client program by “client **new_ip_address**”, where the new IP address is in string format xxx.xxx.xxx.xxx.

The output from the server program should be “RC = 0”. The client program on the PC should receive a data dump of 0,1,...,7 and return code RC=0. On the command prompt window you should see something similar to the image below.



```
C:\>cd \tcp\examples\simple example
C:\TCP\Examples\Simple example>client
recv: 0 1 2 3 4 5 6 7
RC = 0
C:\TCP\Examples\Simple example>
```

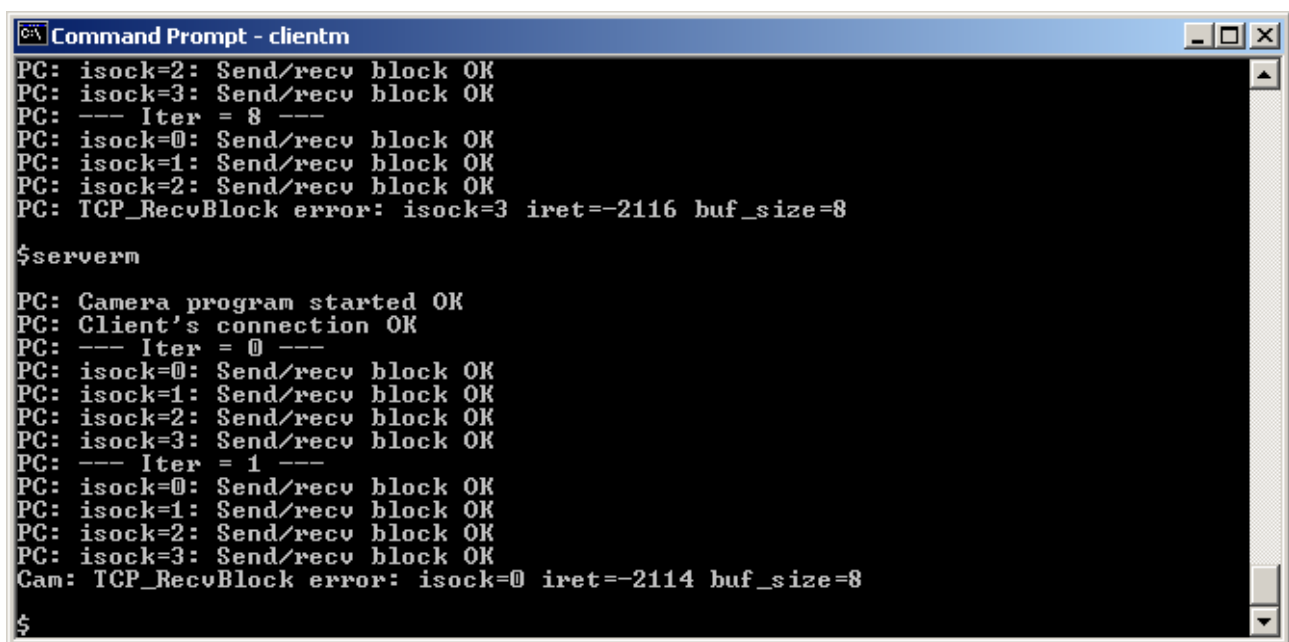
Files used by this example:

server.c	Server example source code
client.c	Client example source code
server.msf	Server example executable module for camera
client.exe	Client example executable module for PC

2. Multiple sockets example

1. Start the supplied terminal program – attoTerminal.
2. Choose the appropriate connection method and fill in the required details.
3. Click “**Connect**” and wait for the terminal to connect to the camera.
4. Click “**Upload MSF**”
5. Navigate to the “*Multiple sockets example*” folder inside the “*Examples*” folder and select SERVER.MSF
6. Click “OK” and wait for the upload to finish.

7. Disconnect or close the terminal program. If you don't disconnect the terminal from the camera you will not be able to establish connection with the CLIENTM application and will not be able to continue with this example.
8. Open a command prompt window and navigate to the "*Multiple sockets example*" folder
9. Start the client application CLIENTM.EXE. If you have changed the default IP address of the camera, you should start the client program by "`clientm new_ip_address`", where the new IP address is in string format `xxx.xxx.xxx.xxx`.
10. Press Enter several times to see the camera shell prompt '\$'.
11. Press F1 to start the test. This command starts the camera program serverm and then executes PC client-connection code. The camera and PC will connect exchange data through 4 sockets. You should receive the following messages in an endless loop:
PC: --- Iter = xxx ---
PC: isock=0: Send/rcv block OK
PC: isock=1: Send/rcv block OK
PC: isock=2: Send/rcv block OK
PC: isock=3: Send/rcv block OK
12. Press Q to disconnect camera or any other key to disconnect PC and return to the camera shell. Depending on the disconnection type, the PC or the camera will display an error message in the following format:
PC:
Cam:
13. Press Esc or F10 to exit the PC program or press F1 to restart the test



```
Command Prompt - clientm
PC: isock=2: Send/rcv block OK
PC: isock=3: Send/rcv block OK
PC: --- Iter = 8 ---
PC: isock=0: Send/rcv block OK
PC: isock=1: Send/rcv block OK
PC: isock=2: Send/rcv block OK
PC: TCP_RecvBlock error: isock=3 iret=-2116 buf_size=8

$serverm
PC: Camera program started OK
PC: Client's connection OK
PC: --- Iter = 0 ---
PC: isock=0: Send/rcv block OK
PC: isock=1: Send/rcv block OK
PC: isock=2: Send/rcv block OK
PC: isock=3: Send/rcv block OK
PC: --- Iter = 1 ---
PC: isock=0: Send/rcv block OK
PC: isock=1: Send/rcv block OK
PC: isock=2: Send/rcv block OK
PC: isock=3: Send/rcv block OK
Cam: TCP_RecvBlock error: isock=0 iret=-2114 buf_size=8

$
```

Files used by this example:

- | | |
|-------------|---|
| serverm.c | Server example source code - multiple sockets |
| clientm.c | Client example source code - multiple sockets |
| conn_mul.c | Multiple socket connection functions - common source code (included in serverm.c and clientm.c) |
| serverm.msf | Server example executable module for camera - multiple sockets |
| clientm.exe | Client example executable module for PC - multiple sockets |

3. VIMOS¹ example

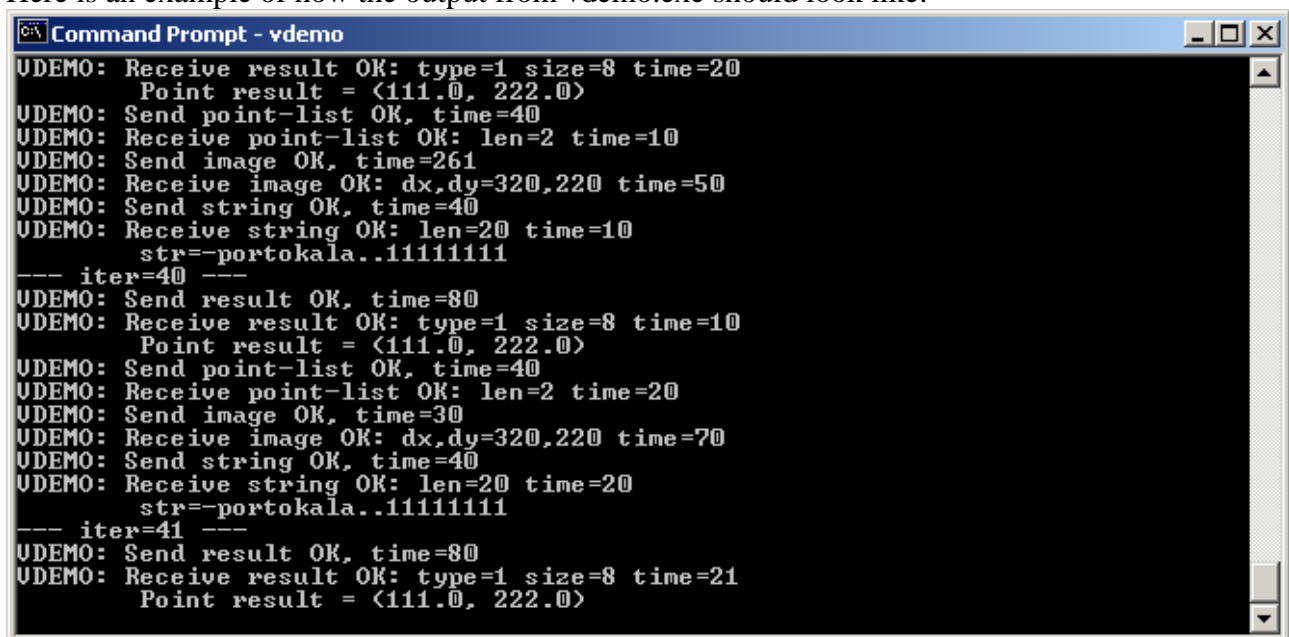
This example will show you how VIMOS can communicate with a PC application. VIMOS is our powerful image processing system. It has over 120 built-in tools which allow you to easily create portable and powerful customized image processing solutions. For more information about VIMOS visit <http://www.vimos.com>.

In order to run the example you must have VIMOS installed on your camera and on your PC. Make sure that you have monitor attached to your camera and that you can see it while working on your PC.

1. Open VDEMO.AEF in the VIMOS Editor.
2. Set your computer's IP address in the "Create server device" tool.
3. Export the program to the VIMOS Simulator and save it as file up0.vm.
4. Load up0.vm to camera.
5. Start VIMOS on camera with the program up0.vm.
6. Open a command-prompt window and start VDEMO on PC. If you have changed the default camera IP address 192.168.0.65, you should start the program by "vdemo **new_ip_address**", where the new IP address is in string format xxx.xxx.xxx.xxx.

You should receive in endless loop of OK messages for the tested I/O tools. Press a key to disconnect and terminate the VDEMO program. You can restart VDEMO after a little wait time - when the "Create server device" tool result, shown on the camera monitor, receives a nonzero error value.

Here is an example of how the output from vdemo.exe should look like:



```
Command Prompt - vdemo
UDEMO: Receive result OK: type=1 size=8 time=20
Point result = <111.0, 222.0>
UDEMO: Send point-list OK, time=40
UDEMO: Receive point-list OK: len=2 time=10
UDEMO: Send image OK, time=261
UDEMO: Receive image OK: dx,dy=320,220 time=50
UDEMO: Send string OK, time=40
UDEMO: Receive string OK: len=20 time=10
str=-portokala..1111111
--- iter=40 ---
UDEMO: Send result OK, time=80
UDEMO: Receive result OK: type=1 size=8 time=10
Point result = <111.0, 222.0>
UDEMO: Send point-list OK, time=40
UDEMO: Receive point-list OK: len=2 time=20
UDEMO: Send image OK, time=30
UDEMO: Receive image OK: dx,dy=320,220 time=70
UDEMO: Send string OK, time=40
UDEMO: Receive string OK: len=20 time=20
str=-portokala..1111111
--- iter=41 ---
UDEMO: Send result OK, time=80
UDEMO: Receive result OK: type=1 size=8 time=21
Point result = <111.0, 222.0>
```



INFORMATION. This example was created using VIMOS version 2.61. Since VIMOS is constantly evolving it is possible that this example requires changes to be made in order to run it under newer VIMOS versions. If you have downloaded a newer VIMOS version and this example is not running, please send us and [e-mail](mailto:info@atto-systems.com).

Alternatively, you can download VIMOS v2.61 [here](#):

¹ More information about VIMOS, VIMOS Editor, VIMOS Simulator, VIMOS tools etc. is available in VIMOS documentation. Please, visit the official VIMOS site <http://www.vimos.com> or contact us at info@atto-systems.com for more information.

Files used by this example:

vdemo.c	Client demo program for communication with the VIMOS program vdemo.aef
vdemo.aef	Server VIMOS program for communication with stand-alone program vdemo.c
vdemo.exe	PC executable module, compiled from vdemo.c