Connecting to a TapeTrack Framework Server

Each TapeTrack Framework Server accepts incoming communications on a single TCP/IP port (by default this port is 5000).

These incoming communications are firstly compressed with gzip level 9 compression, and then encrypted with AES 256 bit encryption.

To allow these connections the TapeTrack Framework Server must have firewall access to accept incoming connections, and the connecting clients must have outbound firewall access to establish a single TCP/IP stateful connection.

In the event the connection is terminated, either by a connection fault or the server timing out the connection, a new connection must be established.

In addition to direct TCP/IP connections, the TapeTrack communications API also supports connections via a HTTP Proxy.

Testing your TCP/IP connection

The simplest was to test if you are able to establish a connection to a TapeTrack Framework Server is to use the TMSS10Ping command line utility.



The TMSS10Ping utility simply sends an empty TapeTrack packet and receives a response. It is available for all supported platforms, requires no login credentials, and has no other dependencies.

Presuming your TapeTrack Framework Server at address **tapetrack.gazillabyte.com**, on port **5000** a successful test would look like this:

```
[root@documentation-us conf]# TMSS10Ping -P 5000 tapetrack.gazillabyte.com
44 bytes from 192.241.211.124: seq=1 time=0.00 ms
44 bytes from 192.241.211.124: seq=2 time=0.00 ms
44 bytes from 192.241.211.124: seq=3 time=0.00 ms
44 bytes from 192.241.211.124: seq=4 time=0.00 ms

--- 192.241.211.124 TMSS10Ping statistics ---
4 packets transmitted
round-trip min/avg/max = 0.00/0.00/0.00 ms
```

Connecting via TapeTrack Software

Command Line Programs

All TapeTrack Command Line programs that communicate with the TapeTrack Framework Server will have a `-S argument`.

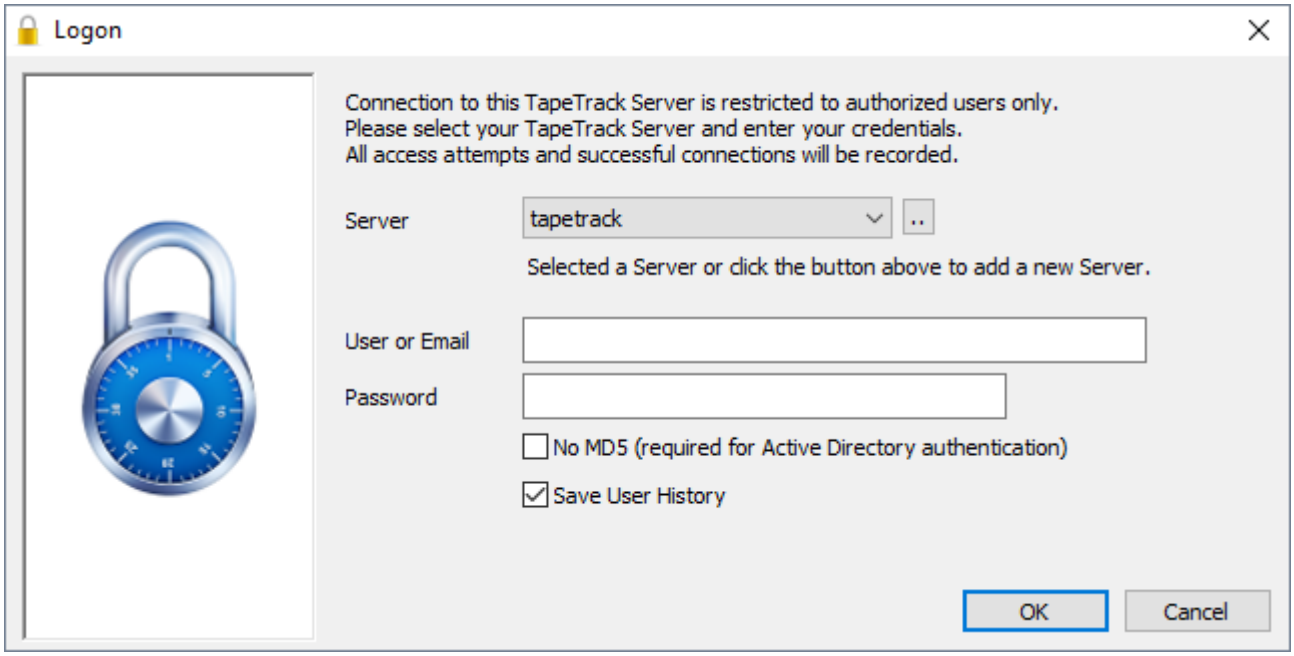
An example for connecting as user **fred**, with password **hackme** is:

```
-S fred:-hackme@tapetrack.gazillabyte.com
```

For detailed instructions on how to use the `-S argument` please see the Command Line Server Argument page.

TapeTrack Desktop Programs

Most users will connect to the TapeTrack Framework Server via one of the Desktop Applications.



Each of these applications will automatically launch the Desktop Login Window which allows the user to setup a server connection.

Server connections added using this Window will be stored in the user's Windows registry under the **HKEY_CURRENT_USER** section. In addition, connections can also be defined in the program's configuration file.

For detailed instructions on how to setup a connection please see:

- Adding a Desktop Server Connection
- Adding a Desktop Server Connection via the Configuration File.
- TapeTrack Server Connection via URL Protocol Handler.

Connecting via a HTTP Proxy

TapeTrack Desktop and Command Line programs allow TapeTrack Framework Server communications to be tunneled via HTTP proxy server.

Although the communications sent via the proxy are not HTTP/HTTPS requests, they are presented to the proxy as HTTPS (Encrypted HTTP) communications.

For details on how to tunnel TapeTrack communications via a HTTP Proxy please see:

- Desktop HTTP Proxy setup
- Command Line HTTP Proxy setup.

Connection Troubleshooting

All TapeTrack programs use an underlying low level communications API which allows users to trace connection and communications requests.

To enable logging, set the TMSSAPILOGDIR environmental variable to the path of an existing directory.



Care should be taken to not leave the TMSSAPILOGDIR set as it will slow all TapeTrack Software due to the overhead of logging.

```
set TMSSAPILOGDIR=c:\Us</note>ers\gnicol\Desktop\logs
TMSS10Ping localhost
```

Creates the file **TMSSAPILOG-005336.txt** with the contents:

```
17:39:25: TMSS10API_Init (API Version=Feb  2 2017 10:18:50)
17:39:25: TMSS10API_Connect: Server(localhost) Port(5000)
17:39:25: TMSS10API_Connect: Connection OK IP(127.0.0.1)
17:39:25: TMSS10API_Send: Function(Ping) Length(0) Flags(0) OptArgs(0)
17:39:25: TMSS10API_Send: RC(Request OK) Feedback(0) MiliSecs(0) Avail(0)
Count(0) Length(0)
17:39:25: TMSS10API_Send: Function(Ping) Length(0) Flags(0) OptArgs(0)
17:39:25: TMSS10API_Send: RC(Request OK) Feedback(0) MiliSecs(0) Avail(0)
Count(0) Length(0)
17:39:25: TMSS10API_Send: Function(Ping) Length(0) Flags(0) OptArgs(0)
17:39:25: TMSS10API_Send: RC(Request OK) Feedback(0) MiliSecs(0) Avail(0)
Count(0) Length(0)
17:39:25: TMSS10API_Send: Function(Ping) Length(0) Flags(0) OptArgs(0)
17:39:25: TMSS10API_Send: RC(Request OK) Feedback(0) MiliSecs(0) Avail(0)
Count(0) Length(0)
17:39:25: TMSS10API_Term
```

communication, environmental variables

From:
https://docs.tapetrack.com/ - **TapeTrack Documentation**

Permanent link:
https://docs.tapetrack.com/general/framework_connection?rev=1496857750

Last update: **2025/01/21 22:07**