

Tachyon RunInstruction command-line tool

Tachyon RunInstruction command-line tool is a stand-alone command line (CLI) tool that supports submission of a Tachyon instruction, question or action to a set of computers.

RunInstruction does not track an instruction once it's submitted, so you'll need to use the Tachyon Explorer or other tools to track an instruction's progress and results.

Read [Installation and configuration](#) for how to install the Tachyon Toolkit and register a consumer.

On this page:

- [Running an Instruction](#)
 - [RunInstruction parameters](#)
 - [Instruction parameters](#)
- [Examples](#)
 - [Simple example](#)
 - [Example of Instruction parameters](#)
 - [Example: offloading the results of an instruction](#)
- [Installation and configuration](#)
 - [Prerequisites](#)
 - [Registering a consumer](#)
 - [Files](#)
 - [Tool configuration](#)
- [Diagnostics](#)

Running an Instruction

Use RunInstruction with the following parameters:

```
Tachyon.RunInstruction -I instruction_name [-P param1 param2 ...] -C consumer_API_URL [-U username password] [-V] [-N ConsumerName] [-O] -D device1 [device2 ...] | -F devices_file
```

If a parameter contains spaces, wrap it in double-quotes.

Option letters are not case-sensitive, and forward slashes can be used instead of hyphens for the options if desired. The options (followed by their associated values) can be in any order, but cannot be repeated.

RunInstruction parameters

Parameter	Description
<code>-I instruction_name</code>	Is the name of the Tachyon instruction to run, as defined in its product pack. The product pack must have been imported into Tachyon. Some instructions consist of a single word, others contain a number of space-separated words (where the value should be wrapped in quotes).
<code>-C consumer_API_URL</code>	Is the URL of the Tachyon Consumer API in the form <code>https://server/Consumer</code> or (from Tachyon v2.1) just <code>https://server</code> (although the <code>/Consumer</code> suffix can still be given).
<code>-P param1 param2 . . .</code>	A list of the parameters (if any) the instruction requires. Some instructions do not take parameters. Read Instruction Parameters for details.
<code>-U username</code>	An optional account username to access the Tachyon Consumer API. If it's not specified the account of the user running the tool is used.
<code>password</code>	An optional account password to access the Tachyon Consumer API. <div data-bbox="297 1564 1463 1675" style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"><p> Note</p><p>Give both a username and password or give neither.</p></div>
<code>-V</code>	An option specifying that device names should be verified against DNS and giving a warning for every device unrecognized by DNS. This is optional because it can be slow.
<code>-N ConsumerName</code>	The name that's passed to the server in the X-Tachyon-Consumer HTTP header. The default is "TachyonRunInstruction". Any custom consumer name must be prefixed with "TachyonRunInstruction".
<code>-O</code>	Specifies that results of the instruction should be Offloaded according to the offloading configuration set for this Consumer on the Tachyon server.

<code>-D device1 device2 ...</code>	A list of one or more devices identified by their FQDNs (fully-qualified domain names). The (single) Tachyon instruction will only be sent to these devices.
<code>-F device s_file</code>	A file containing the list of devices identified by their FQDNs. The file can be either a text file with devices separated by whitespace (for example split per line or space- or tab-separated) or a .csv file with an FQDN column (case is not significant). Duplicates are filtered out. The (single) Tachyon instruction is only sent to these devices. This is an alternative to listing the devices on the command line.

Instruction parameters

All instruction parameters consist of a name and a value. Name matching is case-insensitive, but the case of each value is preserved. Every parameter name defined in the instruction's manifest in the `<ParameterJson>` section must be present with a value. Missing parameters and extra parameters are treated as an error. This tool does not validate the parameter value (for example, in terms of type or a pattern that can be defined in the manifest), only that a required value is present.

Each parameter has the format: `name:value`

The value can contain colon characters – The first colon marks the end of the `name` part. The value can be empty, for example `tagvalue` specifies an empty string value.

Examples



If you copy and paste the following examples, beware of hyphen characters getting lost. You may paste what appears to be a hyphen, but is not.

Simple example

This example involves an instruction that takes no parameters.

```
Tachyon.RunInstruction.exe I 1E-Explorer-TachyonCore-InstalledMemoryDetails C https://tachyon.inv02.local
/Consumer V D inv02cm01.inv02.local inv02cm02.inv02.local
```

This submits the `1E-Explorer-TachyonCore-InstalledMemoryDetails` instruction which gets details of RAM chips. The Tachyon Consumer API server is `tachyon.inv02.local`. The 2 devices on which the instruction runs are `inv02-cm01.inv02.local` and `inv02-cm02.inv02.local` and there is a check that those FQDNs are known hosts. This instruction takes no parameters.

If submission is successful - the resulting instruction ID is displayed on the CLI and logged. The ID can then be used to monitor the progress of the instruction using the Tachyon Explorer.

If submission fails - (for example, the instruction is not known to Tachyon because its product pack has not been imported, or the Consumer API URL is not correct), error messages are displayed and logged.

Example of Instruction parameters

```
Tachyon.RunInstruction.exe I "1E-Explorer-TachyonCore-RegistryEnumerateValues" P "hive:HKLM" "subkey:
Software\1E\Client\Persist" C https://tachyon.inv02.local/Consumer U myusername mypassword D inv02cm01.
inv02.local inv02cm02.inv02.local
```

This submits the `1E-Explorer-TachyonCore-RegistryEnumerateValues` instruction (from the `1E-Explorer-TachyonCore` product pack, found under Classic product packs). As in the example above, the Tachyon Consumer API server is `tachyon.inv02.local`. The 2 devices on which the instruction is run are `inv02-cm01.inv02.local` and `inv02-cm02.inv02.local`.

In this case there's no check that those FQDNs are known hosts (because there is no `-v` option). The name and password of an account with access rights to submit an instruction have been specified. If this question is successful, the values under `HKLM\Software\1E\Client\Persist` in the registry of the specified devices are retrieved.

Example: offloading the results of an instruction

```
Tachyon.RunInstruction.exe -I 1E-Explorer-TachyonCore-BiosDetails -C http://localhost/Consumer -N
SyslogRelay -O -D client1.acme.local
```

This submits the `"1E-Explorer-TachyonCore-BiosDetails"` instruction. The Tachyon Consumer API server is `localhost`, which is the command-line tool from the same computer where Tachyon is installed. The `-N` parameter specifies the the Consumer Name `SyslogRelay` should be used. This consumer has to be previously configured from the Administration pages in Tachyon. Finally, `-O` indicates the results should be offloaded (to whatever address was configured for the consumer) instead of saved to the Tachyon Responses database.

Installation and configuration

The tool is installed as part of the Tachyon Toolkit. The default installation location is:

```
%ProgramFiles(x86)%\1E\Tachyon\Toolkit\TachyonRunInstruction.
```

You can then copy the TachyonRunInstruction folder to be used elsewhere.

Prerequisites

RunInstruction installation prerequisites:

Tachyon installed with its Consumer API web service - for example: <https://tachyon.inv02.local/Consumer>

In Tachyon make sure you have:

- Tachyon user(s) with a known username and password
- The `TachyonRunInstruction` consumer registered in Tachyon
- Imported relevant Tachyon instructions into Tachyon
- Permissioned imported instructions for the Tachyon user(s)
- For Tachyon 4.x permissioned for at least one Management Group, for example **All Devices**.
- Uploaded into Tachyon the instructions you wish to issue.

On the remote client you have:

- .NET Framework 4.5 or later
- A copy of the Tachyon Toolkit.

Registering a consumer

Register a consumer with **Name** `Tachyon RunInstruction` and **Enable** it. Its other properties can be left at their defaults.



If you use the `-N` option to specify a different consumer name at runtime, register and enable a different name prefixed with `Tachyon RunInstruction`.

If you use the `-O` option, specify the **Offload target URL**. For example <https://tachyon.inv02.local/OffloadingClient/api/Responses>

You can register multiple Consumers with variations of names and offloaders.

Registering a consumer in Tachyon 4.x

Go to the Tachyon Portal and navigate to **SettingsConfigurationConsumers**

Registering a consumer in Tachyon 3.x

Go to the Tachyon Explorer Portal and navigate to **AdministrationConsumers**

Add consumer

Name

Consumer URL (optional)

Maximum simultaneous instructions

Offload target URL (optional)

Offload timeout seconds (optional)

Use Windows Authentication

Enabled

Add

Cancel

Consumer settings

Name

Consumer URL

Maximum simultaneous instructions

Offload target URL

Offload timeout seconds

Use Windows Authentication

Enabled

Cancel ✕

Save 📁

Files

The Tachyon Toolkit installer places these files in %ProgramFiles(x86)%\1E\Tachyon\Toolkit\TachyonRunInstruction.

File	Description & Notes
log4net.dll	Provides logging.
Newtonsoft.Json.dll	JSON serialize deserialize library used to communicate with the Tachyon Consumer API.
System.Data.SQLite.dll	SQLite library.
Tachyon.ConsumerApi.Interface.dll	Library to access the Tachyon Consumer API.
Tachyon.RunInstruction.exe	The tool itself.
Tachyon.RunInstruction.exe.config	Configuration file for the tool, controlling logging etc.
Tachyon.SDK.Consumer.dll	Consumer API SDK library.

Tool configuration

Configuration is controlled by `Tachyon.RunInstruction.exe.config` - see above for its location. It controls logging and output, and should not need modification. Any output is sent both to the console and with more details to the logfile.

The default logfile path is:

```
C:\ProgramData\1E\Tachyon\Tachyon.RunInstruction.log
```

The default logging level is `INFO` (which also displays errors and warnings). Read <http://logging.apache.org/log4net/release/manual/configuration.html> for more details about log configuration.

Diagnostics

If the command fails with an error of the form `Consumer API returned BadRequest (Bad Request)` it may not have been registered as a Tachyon consumer (see Prerequisites).