

# ContentDistribution.StageAndExecute

<b>Method</b>	<b>StageAndExecute</b>
<b>Module</b>	ContentDistribution
<b>Library</b>	Core
<b>Action</b>	Similar to <a href="#">Stage</a> (including optional Nomad integration), downloading content from a web server, but also executes a command on successful download.
<b>Parameters</b>	<p><b>LszUrl</b> (string): The URL of a Nomad-style LSZ file that acts as a manifest for the content.</p> <p><b>DirectoryUrl</b> (string): The URL of the top-level directory of the content. The directory can contain files and an arbitrary level of subdirectories. Only files listed in the LSZ manifest will be downloaded, but the LSZ file should have been generated to use the entire content.</p> <p><b>CommandLine</b> (string): A command to run if download is successful.</p>
<b>Return values</b>	<p><b>Downloaded</b> (boolean): Whether download was successful or not.</p> <p><b>NomadStatus</b> (string): One of</p> <ul style="list-style-type: none"><li>• "Nomad not supported" (for non-Windows)</li><li>• "Unknown" (Windows, but did not get a chance to use Nomad)</li><li>• "Nomad successful" (for download)</li><li>• "Nomad failed" (Nomad not installed or suitable, or download with Nomad attempted but failed, but subsequent HTTP download may have succeeded)</li></ul> <p><b>ExitCode</b> (int): The exit code from the command (or the attempt to run the command if the command is not defined).</p> <div style="border: 1px solid red; padding: 5px;"> Many Windows programs set a zero exit code even on failure.</div> <p><b>Output</b> (string): The output that the command sends to standard output and standard error, combined.</p>
<b>Example</b>	<pre>ContentDistribution.Stage(LszUrl: "http://fileserver.domain.local/Content/Tachyon_1d82db90-39ba-47c3-98b1-4483d20ac8f7_1.LSZ",     DirectoryUrl: "http://fileserver.domain.local/Content/Tachyon_1d82db90-39ba-47c3-98b1-4483d20ac8f7",     CommandLine: "msiexec /i installer.msi /qn /l*v c:\hiddenshare\logs\program12\installation.log");</pre>
<b>Platforms</b>	<ul style="list-style-type: none"><li>• Windows (also supports Nomad)</li><li>• Linux</li><li>• MacOS</li><li>• Solaris Intel</li><li>• Solaris Sparc</li><li>• Android - no</li></ul>

## Notes

This action is typically used to download and then install software. The downloaded content is stored in a temporary location which is deleted on completion.

If you just want to execute a script or an install command for pre-existing content then use [Scripting.Run](#) or [NativeServices.RunCommand](#) methods instead.

Just before the `CommandLine` is executed, the working directory is (temporarily) set to where the content was downloaded to (which is a temporary directory). So, for example, if the downloaded content includes an installer file named "app.msi", then the `CommandLine` "`msiexec /i app.msi ...`" will work, i.e. you do not have to worry about deriving a full path.

### Warning: Space for temporary files

Because `StageAndExecute` downloads content to a temporary area, there can be a problem with large content (e.g. GB in size) on non-Windows platforms if the default temporary directory, usually `/tmp`, is in a filesystem that is not big enough. For example, on a Fedora system with 80 GB of disc, a default allocation for the `tmpfs` filesystem which houses `/tmp` is only 500 MB. If this problem occurs, you will see an error like this in the Tachyon agent log:

```
INFO - Downloading "<remote_URL>" to file "/tmp/tachyon<aaaaaa>/<local_file>"
...
ERROR - Error while trying to write to "/tmp/tachyon<aaaaaa>/<local_file>": (28) No
space left on device
```

The selection order for temporary file and directory locations on non-Windows platforms is as follows:

1. Environment variable `TMPDIR`, if set
2. The standard Linux/Unix location defined by `P_tmpdir`, usually, `/tmp`
3. `/tmp`
4. `/var/tmp`

The Tachyon agent selects the first of these that is an existing directory.

Hence, if the error occurs (typically with a `/tmp` that is too small), environment variable `TMPDIR` can be set to `/var/tmp`, for example, which is usually on a filesystem with much more space assigned to it. However, since the agent runs as root daemon/service, it is not recommended to make any changes that would affect all root actions. Instead, change the way the daemon runs as follows. This describes Linux systems that run daemons using the `systemd` subsystem. As usual, use `sudo` commands to perform these actions.

First, after installation of the Tachyon Agent, edit `/usr/lib/systemd/system/tachyon.agent.service` to add this line in the `[Service]` section:

```
Environment="TMPDIR=/var/tmp"
```

(or specify some other suitable directory for large temporary files).

Then run these commands to pick up the change and restart the Tachyon agent:

```
systemctl daemon-reload
systemctl restart tachyon.agent.service
```