

Tatsoft.FrameworkX PowerShell Module Reference

Native PowerShell cmdlets for FrameworkX — lifting IT deployment from raw `.exe + /silent` calls to one-liner cmdlets with `Get-Help`, parameter validation, pipeline composition, and structured exit-code handling.

[Release Notes](#) [IT Deployment RunBook](#) Tatsoft.FrameworkX PowerShell Module Reference

Version 10.1.5+

Overview

Tatsoft.FrameworkX is a single PowerShell module that wraps three existing FrameworkX surfaces:

Surface	Wrapped CLI / endpoint	Audience
Product lifecycle	FrameworkX-Setup.exe, unins000.exe (Inno Setup)	IT / DevOps deploying FrameworkX
Service lifecycle	TManageServices.exe (per-service install / uninstall verbs)	IT / DevOps managing Windows Services
SolutionCenter API	https://<host>:10108/api/v1/... (TWebServices)	Fleet operators, OEM admin tooling, central monitoring

Compatible with **PowerShell 5.1** (Windows PowerShell, ships with Windows) and **PowerShell 7+** (cross-platform). The module is signed by Tatsoft and exports 33 cmdlets.

Install

```
# All-users (typical IT deployment, requires admin):
Copy-Item -Recurse '\\fileshare\Tatsoft.FrameworkX' 'C:\Program Files\WindowsPowerShell\Modules\Tatsoft.FrameworkX'
Import-Module Tatsoft.FrameworkX
```

The module is also dropped under `<InstallDir>\Tools\PowerShell\Tatsoft.FrameworkX\` by future FrameworkX installer revisions.

Quick start

Install on a fresh box

```
Install-FrameworkX -InstallerPath C:\Setup\FrameworkX-Setup.exe -InstallDir 'C:\Program Files\Tatsoft\FrameworkX'
Install-FrameworkXWebServices
Install-FrameworkXRuntime -SolutionPath 'C:\Solutions\Plant1.tproj' -Profile Production
```

Manage solutions remotely

```
Connect-FrameworkX -Endpoint https://plant1-srv07:10108 -ServiceAccountToken $env:FX_TOKEN
Get-FrameworkXSolution
Start-FrameworkXSolution -Name Plant1SCADA -Profile Production
Get-FrameworkXSolution | Where-Object status -eq running | Stop-FrameworkXSolution
Disconnect-FrameworkX
```

Cmdlets

Product lifecycle

Cmdlet	Wraps	Notes
--------	-------	-------

Install-FrameworkX - InstallerPath -InstallDir - LogPath	FrameworkX-Setup.exe /VERYSILENT /SUPPRESSMSGBOXES /NORESTART /DIR= /LOG=	Maps Inno exit codes to a structured result. Exit code 3010 = success + reboot required.
Uninstall-FrameworkX [- InstallDir]	unins000.exe /VERYSILENT	Auto-detects install directory from HKLM.

Service lifecycle (machine-local, requires admin)

Cmdlet	Wraps
Install-FrameworkXRuntime -SolutionPath -Profile	/installtstartup /solution:"..." /profile:n — installs TStartup- p-<SolutionName>
Uninstall-FrameworkXRuntime -SolutionName	/uninstalltstartup /solution:"..."
Install-FrameworkXWebServices / Uninstall-FrameworkXWebServices	/installtwebervices, /uninstalltwebervices
Install-FrameworkXSecureGateway / Uninstall-FrameworkXSecureGateway	/installtsecuregateway, /uninstalltsecuregateway
Install-FrameworkXMqttBroker / Uninstall-FrameworkXMqttBroker	/installtmqttbroker, /uninstalltmqttbroker
Install-FrameworkXHardkey / Uninstall-FrameworkXHardkey	/installthardkey, /uninstallthardkey
Install-FrameworkXRuntimeMcpHttp / Uninstall-FrameworkXRuntimeMcpHttp	/installruntimemcphttp, /uninstallruntimemcphttp
Remove-FrameworkXAllServices	/removeallservices — every FrameworkX service in one call

Session

Cmdlet	Calls
Connect-FrameworkX -Endpoint - ServiceAccountToken	(no HTTP — stores JWT and base URL in module state)
Connect-FrameworkX -Endpoint -OidcProvider	Opens browser to /api/v1/auth/oidc/login; captures JWT via localhost listener
Disconnect-FrameworkX	(no HTTP — clears module state, restores TLS validation)

Solutions

Cmdlet	Calls
Get-FrameworkXSolution	GET /api/v1/solutions/
Get-FrameworkXSolution -Name X	GET /api/v1/solutions/X/info (enriched with isRunning + runningProfiles)
Test-FrameworkXSolutionRunning -Name X	GET /api/v1/solutions/X/isrunning (returns [bool])
Start-FrameworkXSolution -Name -Profile	POST /api/v1/solutions/X/run
Stop-FrameworkXSolution -Name	POST /api/v1/solutions/X/stop
Set-FrameworkXSolutionAutoStart -Name - AutoStart	PUT /api/v1/solutions/X/autostart

License

Cmdlet	Calls
Get-FrameworkXLicenseInfo	GET /api/v1/license/info
Get-FrameworkXSiteCode	GET /api/v1/license/site- code

Set-FrameworkXLicenseKey -LicenseKey	PUT /api/v1/license/key
Invoke-FrameworkXLicenseActivation -ActivationCode [-SerialNumber] [-SiteCode] [-Upgrade]	POST /api/v1/license/activate
Invoke-FrameworkXLicenseDeactivation -ActivationCode	POST /api/v1/license/deactivate

Machine settings

Cmdlet	Calls
Get-FrameworkXMachineSetting	GET /api/v1/machine/settings (lists the allowlist)
Get-FrameworkXMachineSetting -Filename	GET /api/v1/machine/settings/{filename}
Set-FrameworkXMachineSetting -Filename -Content	PUT /api/v1/machine/settings/{filename} (requires installation:admin)

Server

Cmdlet	Calls
Get-FrameworkXServerInfo	GET /api/v1/server/info
Get-FrameworkXConnection	GET /api/v1/server/connections (requires installation:admin)
Disconnect-FrameworkXConnection -Guid	DELETE /api/v1/server/connections/{guid} (requires installation:admin)

Discovery

Cmdlet	Calls
Get-FrameworkXOpenApi [-OutFile]	GET /api/v1/openapi (anonymous, bypasses the activation gate)

Authentication

The SolutionCenter API requires a JWT Bearer token on every call (RFC 9068 at+jwt). Two issuance paths:

- **Service-account JWT** — pre-issued by your OEM admin tool, dropped on the box as a config file or held in an environment variable. Pass via `Connect-FrameworkX -ServiceAccountToken <jwt>`.
- **OIDC interactive** — `Connect-FrameworkX -OidcProvider <name>` opens the default browser to the configured IdP via `/api/v1/auth/oidc/login`, then captures the JWT redirect via a localhost `HttpListener`. The named provider must be configured in `MachineSettings/SolutionCenterApi-OidcProviders.json` on the target installation.

Scopes (assigned at issuance time): `installation:read`, `installation:control`, `installation:files`, `installation:license`, `installation:machine`, `installation:admin`. Scopes do not subsume each other.

Activation gate

The SolutionCenter API ships **default-OFF in 10.1.5 GA**. API cmdlets surface 503 `service-unavailable` until the operator sets `SolutionCenterApi.Enabled = true` in `TWebServices` machine settings AND the running build is `>= MinSafeBuildForSolutionCenterApi`. Service-lifecycle cmdlets (`Install-FrameworkX*`) are unaffected by the gate. `Get-FrameworkXOpenApi` is anonymous and bypasses the gate.

TLS notes

- **PowerShell 7+:** `Connect-FrameworkX -SkipCertificateCheck` is honored per-call (`Invoke-RestMethod -SkipCertificateCheck`). Production deployments should use a properly-signed certificate.
- **PowerShell 5.1:** `-SkipCertificateCheck` installs a process-wide `ServicePointManager.ServerCertificateValidationCallback`, restored when `Disconnect-FrameworkX` runs. The module also force-enables TLS 1.2 (the PS 5.1 default rejects modern endpoints).

Error handling

API calls failing with non-2xx responses raise terminating errors of id `FrameworkX.Api.HttpError`. The error message includes the RFC 7807 Problem Details payload from the server. Catch with `try { ... } catch { ... }`:

```
try {
    Get-FrameworkXLicenseInfo
}
catch {
    Write-Warning "License query failed: $($_.Exception.Message)"
}
```

CLI cmdlets (`Install-* / Uninstall-*`) raise terminating errors when `TManageServices.exe` reports a non-zero exit code (= number of failed operations) or when the Inno Setup installer reports a failure exit code.

Out of scope (v1.0 of this module)

- File operations (`/api/v1/files/...`) — chunked upload / download, exists, delete-solution. Wrapped in v1.1.
 - Visual dashboard UI — tracked in a separate Tatsoft v2 work item.
 - MSI distribution package.
 - Bulk fan-out (managing many machines from a single shell).
-

Related

- [IT Deployment RunBook](#) — the broader IT Ops guide.
 - [SolutionCenter API Reference](#) — full API spec, OpenAPI doc, scope model.
-