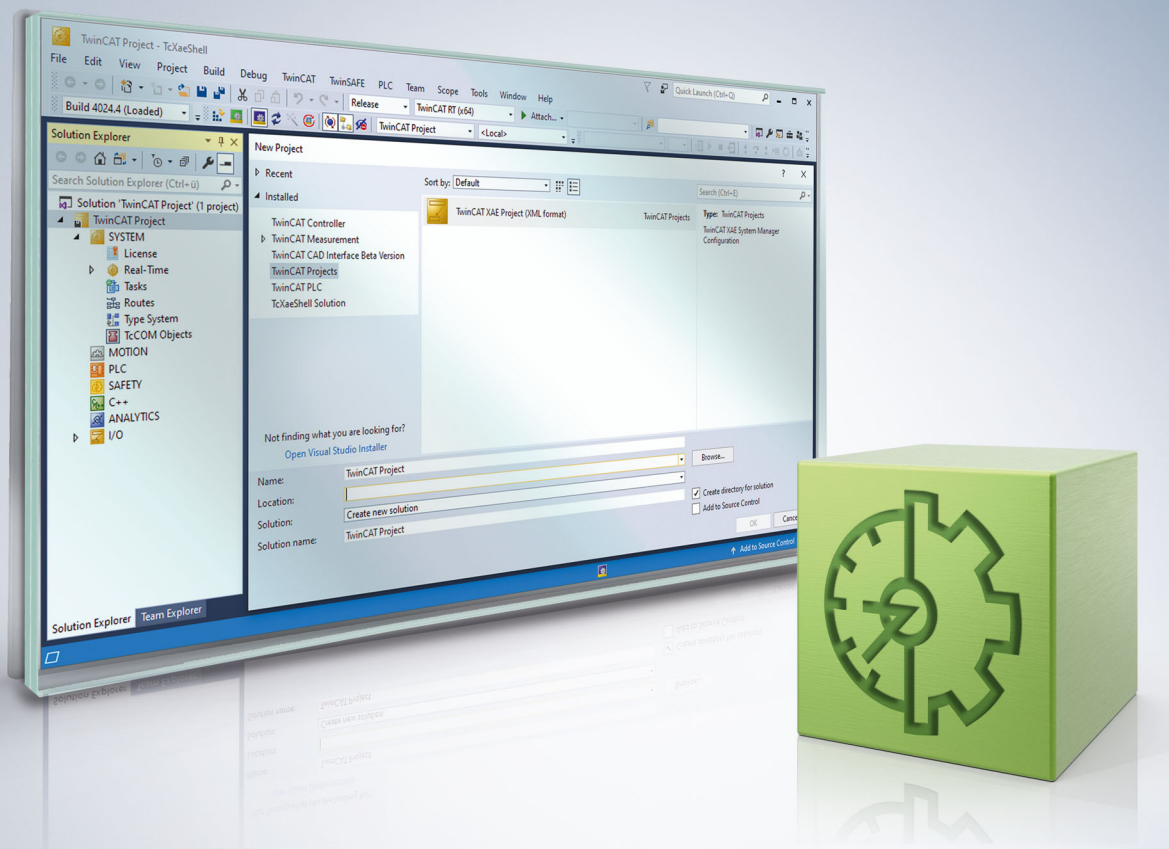
BECKHOFF New Automation Technology

Manual | EN

TE1000

TwinCAT 3 | ADS Powershell Module



2024-02-19 | Version: 1.3.0

Table of contents

1 Foreword

1.1 Notes on the documentation

This description is intended exclusively for trained specialists in control and automation technology who are familiar with the applicable national standards.
For installation and commissioning of the components, it is absolutely necessary to observe the documentation and the following notes and explanations.
The qualified personnel is obliged to always use the currently valid documentation.

The responsible staff must ensure that the application or use of the products described satisfies all requirements for safety, including all the relevant laws, regulations, guidelines, and standards.

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.
We reserve the right to revise and change the documentation at any time and without notice.
No claims to modify products that have already been supplied may be made on the basis of the data, diagrams, and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered and licensed trademarks of Beckhoff Automation GmbH.
If third parties make use of designations or trademarks used in this publication for their own purposes, this could infringe upon the rights of the owners of the said designations.

Patents

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
and similar applications and registrations in several other countries.

EtherCAT®

EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany

Copyright

1.2 For your safety

Safety regulations

Read the following explanations for your safety.
Always observe and follow product-specific safety instructions, which you may find at the appropriate places in this document.

Exclusion of liability

All the components are supplied in particular hardware and software configurations which are appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

Personnel qualification

This description is only intended for trained specialists in control, automation, and drive technology who are familiar with the applicable national standards.

Signal words

The signal words used in the documentation are classified below. In order to prevent injury and damage to persons and property, read and follow the safety and warning notices.

Personal injury warnings

⚠ DANGER

Hazard with high risk of death or serious injury.

⚠ WARNING

Hazard with medium risk of death or serious injury.

⚠ CAUTION

There is a low-risk hazard that could result in medium or minor injury.

Warning of damage to property or environment

NOTICE

The environment, equipment, or data may be damaged.

Information on handling the product

This information includes, for example:
recommendations for action, assistance or further information on the product.

1.3 Notes on information security

The products of Beckhoff Automation GmbH & Co. KG (Beckhoff), insofar as they can be accessed online, are equipped with security functions that support the secure operation of plants, systems, machines and networks. Despite the security functions, the creation, implementation and constant updating of a holistic security concept for the operation are necessary to protect the respective plant, system, machine and networks against cyber threats. The products sold by Beckhoff are only part of the overall security concept. The customer is responsible for preventing unauthorized access by third parties to its equipment, systems, machines and networks. The latter should be connected to the corporate network or the Internet only if appropriate protective measures have been set up.

In addition, the recommendations from Beckhoff regarding appropriate protective measures should be observed. Further information regarding information security and industrial security can be found in our https://www.beckhoff.com/secguide.

Beckhoff products and solutions undergo continuous further development. This also applies to security functions. In light of this continuous further development, Beckhoff expressly recommends that the products are kept up to date at all times and that updates are installed for the products once they have been made available. Using outdated or unsupported product versions can increase the risk of cyber threats.

To stay informed about information security for Beckhoff products, subscribe to the RSS feed at https://www.beckhoff.com/secinfo.

2 Overview

What is Powershell

Excerpt from Wikipedia: “PowerShell is a task automation and configuration management framework from Microsoft, consisting of a command-line shell and associated scripting language built on the .NET Framework and .NET Core.”

“In PowerShell, administrative tasks are generally performed by cmdlets (pronounced command-lets), which are specialized .NET classes implementing a particular operation. Sets of cmdlets may be combined into scripts, executables (which are standalone applications), or by instantiating regular .NET classes (or WMI/COM Objects). These work by accessing data in different data stores, like the file system or registry, which are made available to the PowerShell runtime via PowerShell providers.” (link)

These Cmdlets are packaged and deployed in so called Powershell Modules.

What is the Powershell Ads Module (Extension) named ‘TcXaeMgmt’

The so called **TcXaeMgmt** module contains a number of useful Cmdlets and Providers for TwinCAT Management/Administration and accessing Ads Devices natively via Powershell.

- Route Management (Add-AdsRoute, Remove-AdsRoute) and Broadcast search (Get-AdsRoute)
- Testing Route Connections (Test-AdsRoute, Get-AdsState)
- Establish Ads Communication channels via Sessions and Connection (New-TcSession, Close-TcSession)
- Type-Safe Read/Write Value Access via ADS protocol (Read-TcValue, Write-TcValue)
- Symbol and DataType Browsing (Get-TcSymbol, Get-TcDataType)
- Further Administrative Tasks (Copy-AdsFile, Get-TcVersion, etc.)

The package is published via the Powershell Gallery at https://www.powershellgallery.com/packages/TcXaeMgmt/.

Versions

There exist two different series of the ‘TcXaeMgmt’ module:

Versions	Powershell Version	TwinCAT Version	Description
6.X [▶ 31]	Windows Powershell 5.1 Microsoft Powershell >= 6.0	>= 3.1.4024.10	Platform independent version
3.X	Windows Powershell >= 3.0	All	Running on Windows Versions >= Windows 7

Please be aware of the Differences Microsoft Powershell vs. Windows Powershell.

3 Requirements

Version 5.X	Version 3.2.X
• Microsoft Powershell >= 6.0 or Windows Powershell 5.1 • Installed TwinCAT 3.1.4024.10 or newer (minimum RT / ADS level)	• Windows 7 SP1 and newer • Windows Powershell 4.0 and newer • .NET Framework 4.5 and newer • Installed TwinCAT 3 or TwinCAT 2 (minimum RT / ADS level)

4 Installation and Activation

4.1 Installation by PackageManager (PowershellGallery, Internet access necessary)

Installation by Package Manager (Powershell Gallery)

For newer versions of Windows and Powershell the most easiest way to install the **TcXaeMgmt** Module is to use the Powershell Gallery. Powershell Gallery access is available without installation if one of the following setups is already on the system:

Windows 10 or newer

Windows Server 2016 or newer

Windows Management Framework (WMF) 5.0 or newer

PowerShell 6 or newer.

In this case just type

PS> Install-Module -Name TcXaeMgmt

from the Powershell console.

Further information about the package and its installation is available on the Gallery Website: https://www.powershellgallery.com/packages/TcXaeMgmt/

Installation of PowershellGet Module

In all other cases the **PowershellGet** module must be installed on the machine most likely. On Powershell the availability can be checked with the following command:

```
PS> get-module PowershellGet -listavailable

Directory: C:\Program Files\WindowsPowerShell\Modules

ModuleType Version Name ExportedCommands
---------- ------- ---- ----------------
Script 2.2.5 PowerShellGet {Find-Command, Find-DSCResource, Find-Module, Find-RoleCapability...}
```

Please assure, that at minimum Version 2.2.5 is available.

The Powershell version can be determined as follows:

```
PS> $PSVersionTable

Name Value
---- -----
PSVersion 5.0.10514.6
WSManStackVersion 3.0
SerializationVersion 1.1.0.1
CLRVersion 4.0.30319.42000
BuildVersion 10.0.10514.6
PSCompatibleVersions {1.0, 2.0, 3.0, 4.0...}
PSRemotingProtocolVersion 2.3
```

If the **PowershellGet** Module is not existing please follow the instructions of the following websites

- https://www.powershellgallery.com/packages/PackageManagement/
- https://learn.microsoft.com/en-us/powershell/gallery/powershellget/install-powershellget?view=powershellget-2.x

Now, when the **PowershellGet** Module is available, the command

```
PS> Install-Module -Name TcXaeMgmt
```

from the Powershell console should work.

4.2 Manual Installation without Internet access

Because the TwinCAT XAE Management Powershell Module (**TcXaeMgmt**) is now available on the Powershell Gallery (https://www.powershellgallery.com/packages/TcXaeMgmt/) it is not necessary to activate the Powershell Module manually if Internet access available. For completeness and if no Internet is present the following steps show the manual installation process.

Check Installed Powershell Module

The TwinCAT Installation includes the setup for the Powershell **TcXaeMgmt** Module. It should already be existing under the folder

```
[TWINCATINSTALL]/AdsApi/Powershell/TcXaeMgmt
```

where [TWINCATINSTALL] indicates the TwinCAT root folder (c:\TwinCAT by default).

Check the Powershell Cmdlet Execution policy

What is left actually to the user is to activate that module in the Powershell environment.

```
PS> get-executionPolicy
Restricted
```

If the policy is not set to 'Unrestricted' or 'RemoteSigned', Powershell does not allow to process scripts or Cmdlets. For more information, please see

```
PS> get-help about_Execution_Policies
```

If the execution policy is restricted, it has to be set to 'RemoteSigned' by a Powershell console with administrative rights:

```
PS C:\tfs> Set-ExecutionPolicy RemoteSigned

Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution
policy might expose you to the security risks described in the about_Execution_Policies help topic
at http://go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the execution policy?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): y
```

Extend Powershell Module Search Path (PSModulePath)

As next step, the Powershell Search path for Powershell Modules must be extended, so that Powershell can find the TcXaeMgmt Cmdlets in the

```
[TWINCATINSTALL]/AdsApi/Powershell/
```

folder.

Please check the Enviroment variable 'PSModulePath':

```
PS> $env:PSModulePath
D:\Users\User\Documents\WindowsPowerShell\Modules\;C:\Program Files\WindowsPowerShell\Modules;C:
\Windows\system32\WindowsPowerShell\v1.0\Modules\;
```

If the TwinCAT AdsApi path is not contained in the PSModulePath Variable, it should be set via the Windows Control Panel → System Properties → Advanced → Environment Variables Dialog.

Please add

```
[TWINCATINSTALLDIR]\AdsApi\Powershell\
```

e.g with

```
C:\TwinCAT\AdsApi\Powershell
```

to the System wide 'PSModulePath' Variable. After a Powershell Console restart, the new setting should be available:

```
PS C:\tfs> $env:PSModulePath
D:\Users\User\Documents\WindowsPowerShell\Modules\;C:\Program Files\WindowsPowerShell\Modules;C:
\Windows\system32\WindowsPowerShell\v1.0\Modules;c:\TwinCAT\AdsApi\Powershell\
```

4.3 Check succeeded installation

To check that the TcXaeMgmt Module can be loaded successfully and is operational type:

```
PS > get-module TcXaeMgmt -listavailable

Directory: C:\Program Files\WindowsPowerShell\Modules

ModuleType Version Name ExportedCommands
---------- ------- ---- ----------------
Binary 3.1.1122 TcXaeMgmt {Add-AdsRoute, Close-TcSession, Copy-AdsFile, Get-AdsRoute...}
```

Now the contained cmdlets can be accessed:

```
PS> get-command -module TcXaeMgmt

CommandType Name
----------- ----
Cmdlet Add-AdsRoute
Cmdlet Close-TcSession
Cmdlet Copy-AdsFile
Cmdlet Get-AdsRoute
Cmdlet Get-AdsState
Cmdlet Get-TcDataType
Cmdlet Get-TcSession
Cmdlet Get-TcSymbol
Cmdlet Get-TcTargetInfo
Cmdlet Get-TcVersion
Cmdlet New-TcSession
Cmdlet Read-TcValue
Cmdlet Remove-AdsRoute
Cmdlet Set-AdsState
Cmdlet Test-AdsRoute
Cmdlet Write-TcValue


PS> get-adsstate

Name State OK Time (ms) Address
--------- ----- -- --------- -------
CX_000001 Run True 2 14.5.137.176.1.1
```

Help is included for an overview of the features and concepts:

```
PS > get-help about_TcXaeMgmt
```

Or for specific Cmdlet information:

```
PS > get-help Read-TcValue -full
```

5 How To

5.1 Getting Help

Getting list of Cmdlets (Commands) in the **TcXaeMgmt** module.

```
PS> get-command -module TcXaeMgmt
```

Get Syntax information about a Cmdlet.

```
PS> get-help get-AdsRoute
```

Get Examples for a Cmdlet.

```
PS> get-help get-AdsRoute -examples
```

Get Full Help for a Cmdlet.

```
PS> get-help get-AdsRoute -full
```

Get Full Help for a Cmdlet online from TwinCAT InfoSys.

```
PS> get-help get-AdsRoute -online
```

Get summary information about the **TcXaeMgmt** Powershell module.

```
PS> get-help about_TcXaeMgmt
```

5.2 Argument Completers

Beneath predictive intellisense, Argument Completers provide dynamic tab completion for parameter values while typing Powershell Cmdlets into the shell.

PowerShell provides completions on input to provide hints, enable discovery, and speed up input entry. Command names, parameter names, argument values and file paths can all be completed by pressing the <Tab> key.

Powershell also provides a MenuComplete function that's bound to <Ctrl+Space>. The MenuComplete function displays a list of matching completions below the command line.

The **TcXaeMgmt** Powershell module supports Argument Completers in many variations.

Device Address Completers

Typing Ctrl+Space while inserting the Parameters that specify the Address (here NetId, Route or Address), opens a list of registered routes of the Target System + the local system. These are the reachable routes.

```
PS> $s = new-TcSession -NetId 172.17.60.197.1.1
MYSYSTEM (192.168.0.2.1.1) REMOTESYSTEM (192.168.0.3.1.1)
```

```
PS> $s = new-TcSession -Route MYSYSTEM
MYSYSTEM (192.168.0.2.1.1) REMOTESYSTEM (192.168.0.3.1.1)
```

```
PS> $s = new-TcSession -Address 192.168.0.2
MYSYSTEM (192.168.0.2.1.1) REMOTESYSTEM (192.168.0.3.1.1)
```

Symbol Path Completers

The -Path Parameter describes the path to the symbol. If the target device system (with symbolic information) is already known by the Cmdlet (here by -Session parametrization), the argument completer can load the symbolic information from the target system and present the available Symbol Path options in the list.

If the target address is unavailable or not specified, no selection is proposed.

```
PS> $s = new-TcSession -NetId Local -port 851
```

```
PS> read-TcValue -session $s -Path '.tc2vBool'
'.tc2vBool' '.tc2vStruct' 'FB_Test' 'MAIN'
'.tc2vInt' '.tc2vStructArray' 'Global_Version' 'Slow'
'.tc2vIntArray' 'Constants'
'GVL' 'TwinCAT_SystemInfoVarList'
```

Depending on the size of the target symbolic information, downloading the symbolic information can take some time!

Using the -session parameter here instead of -netId and -port ensures that the Symbolic Information is loaded only one time and cached in the Session object!

DataType Name Completers

Here, the same session object is used for accessing the list of Data types.

```
PS> $s = new-TcSession -NetId Local -port 851
```

```
PS> get-TcDataType -session $s -name 'BIT'
'BIT' 'BOOL' 'BYTE'
```

Depending on the size of the target symbolic information, downloading the symbolic information can take some time!

Using the -session parameter here instead of -netId and -port ensures that the Symbolic Information is loaded only one time and cached in the Session object!

And many more …

Argument completers are available for many Cmdlet Parameters. Just try out <TAB> and <CTRL + TAB>.

5.3 Getting registered Routes and Broadcast Search

Getting locally registered routes:

```
PS> Get-AdsRoute

Name NetId Protocol TLS Address FingerPrint
---- ----- -------- --- ------- -----------
CX_111111 192.168.0.2.1.1 TcpIP 192.168.0.2 a783645203c138fb49abc0d10eac4f…
CX_222222 192.168.0.3.1.1 TcpIP 192.168.0.3
```

Broadcast Search:

```
PS> Get-AdsRoute -all

Name NetId Protocol TLS Address FingerPrint
---- ----- -------- --- ------- -----------
MYDESKTOP 192.168.0.5.1.1 TcpIP X 192.168.0.5 2f72d63cba3069b5e15a3983fbfa1da7914f7c1…
CX_111111 192.168.0.2.1.1 TcpIP X 192.168.0.2 1ab3994a2ee6fbcdc999d5d1735cb9d5a12366d…
CX_222222 192.168.0.3.1.1 TcpIP X 192.168.0.3 4864cc5d2fbdcfe2128ddaab222fe1a6faa0a4d…
CX_333333 192.168.0.4.1.1 TcpIP X 192.168.0.4 efcd5570e981784342bcf2b2fd03b54b22746c4…
```

5.4 Adding / Removing Routes

Search Routes and add the result at the Local TwinCAT System

```
PS> Get-AdsRoute -All -name "Tc3*"

Name            NetId             Address       Sub TcVersion RTSystem
----            -----             -------       --- --------- --------
TC3TestA1-CP67x 192.168.0.105.1.1 192.168.0.105     3.1.4021  Win7
TC3Test13-C6650 172.17.60.239.1.1 192.168.0.156     2.11.2246 Win7


PS> $cred = Get-Credential -Message "Get Credentials" -UserName "UserName"

PS> Add-AdsRoute -Credential $cred -Address "TC3TestA1-CP67x" -temporary -passthru

Name            NetId             Address       Sub TcVersion RTSystem
----            -----             -------       --- --------- --------
TC3TestA1-CP67x 192.168.0.105.1.1 192.168.0.105     3.1.4021  Win7

PS> Get-AdsRoute -name "TC3TestA1-CP67x" | Test-AdsRoute
```

Search for Systems that start with the name "TC3*", then asks the user for Credentials and adds the Route as 'temporary' (with TC2 compatible security, clear text password).

Afterwards, the connection is checked via 'Test-AdsRoute'.

The route is specified by its name (ComputerName).

To find out the address of the route an under the hood broadcast search is necessary what means that the target system must be online available in the network.

Add a route to the local system (Single Sided)

```
PS> Add-AdsRoute -name Test -NetId 1.2.3.4.1.1 -IPOrHostName 1.2.3.4
```

Adds a Route named 'Test' to the local routes with the specified NetId and IPAddress.

Because NetId and IPOrHostName are defined AND no credentials are set, this route is added locally only.

Be aware that to get the route functional, the target system must define the backroute.

Add a single sided temporary route

```
Add-AdsRoute -name "TestRoute" -NetId 1.2.3.4.1.1 -IPOrHostName 1.2.3.4 -Temporary -
RemotePersistance None
```

Adding a route 'TestRoute' single sided and temporary only to the local system.

The remote device doesn't need to be online.

Add a self-signed route

```
PS> Get-AdsRoute -All -name "Tc3*"

Name            NetId             Address       Sub TcVersion RTSystem
----            -----             -------       --- --------- --------
TC3TestA1-CP67x 192.168.0.105.1.1 192.168.0.105     3.1.4021  Win7
TC3Test13-C6650 172.17.60.239.1.1 192.168.0.156     2.11.2246 Win7


PS> $cred = Get-Credential -Message "Get Credentials" -UserName "UserName"

PS> Add-AdsRoute -Credential $cred -name "TC3TestA1-CP67x" -selfSigned -passthru

Name            NetId             Address       Sub TcVersion RTSystem
----            -----             -------       --- --------- --------
TC3TestA1-CP67x 192.168.0.105.1.1 192.168.0.105     3.1.4021  Win7

PS> Get-AdsRoute -name "TC3TestA1-CP67x" | Test-AdsRoute
```

Search for Systems that start with the name "TC3*", then asks the user for Credentials and adds the Route with 'SelfSigned' AdsSecure settings.

Afterwards, the connection is checked via 'Test-AdsRoute'.

The route is specified by its name (ComputerName).

To find out the address of the route an under the hood broadcast search is necessary what means that the target system must be online available in the network.

Add a route with (S)hared (C)ertification (A)Authority (SCA)

```
PS> Add-AdsRoute -Address 192.168.0.105 -sca -paththru

Name            NetId             Address       Sub TcVersion RTSystem
----            -----             -------       --- --------- --------
TC3TestA1-CP67x 192.168.0.105.1.1 192.168.0.105     3.1.4021  Win7
```

Searches for the system with the specified IPAddress, and add the Route with Shared Certification Authority settings without password.

The precondition is, that valid certificates are already established on both (engineering and remote) systems, within their StaticRoutes.xml files.

The route is specified by its Address only.

Because the NetId is missing a broadcast search is necessary what means that the target system must be online available in the network.

Add a Route with (N)etwork (A)Address (T)ranslation (NAT)

```
PS> $cred = Get-Credential -Message "Get Credentials" -UserName "UserName"

PS> Add-AdsRoute -Credential $cred -NetId 192.168.0.105 -Nat 1.2.3.4.1.1

Name            NetId       Address       Sub TcVersion RTSystem
--------------- ----------- ------------- --- --------- ------------
TC3TestA1-CP67x 1.2.3.4.1.1 192.168.0.105     3.1.4024  Win10 (2004)
```

Add a route with a local network address translation (NAT AmsNetId) to project a remote AmsNetId (RemoteNetId) locally to a different address.

Adding a (S)hared (C)ertificate (A)uthority route

```
PS> $route = get-adsroute CX_01234 -all
PS> $route

Name             NetId                TLS   Address          FingerPrint
----             -----                ---   -------          -----------
CX_01234         172.17.60.197.1.1    X     172.17.60.197    7835dae7a079c4f296c84109b2e6d7156b66e6b
cc39e386c3576d7535…

PS> $route | add-adsroute -SharedCertAuth -IgnoreCN -passthru

Name             NetId                TLS   Address          FingerPrint
----             -----                ---   -------          -----------
CX_01234         172.17.60.197.1.1    X     172.17.60.197    7835dae7a079c4f296c84109b2e6d7156b66e6b
cc39e386c3576d7535…
```

Broadcast search for a Device with Hostname CX_01234 and adding of a ADSSecure route via 'Shared Certificate Authority' (SCA) to the local system.

Both systems must contain certificates derived from the same root CA certificate.

Adding a route with UserName/Password (P)re (S)hared (Key) (PSK)

```
PS> $cred = get-credential
UserName: MyUser
Password: ********

PS> $route = get-adsroute CX_01234 -all
PS> $route

Name             NetId                TLS   Address          FingerPrint
----             -----                ---   -------          -----------
CX_01234         172.17.60.197.1.1    X     172.17.60.197    7835dae7a079c4f296c84109b2e6d7156b66e6b
cc39e386c3576d7535…

PS> $route | add-adsroute -PreSharedKey -Credential $cred

Name             NetId                TLS   Address          FingerPrint
----             -----                ---   -------          -----------
CX_01234         172.17.60.197.1.1    X     172.17.60.197    7835dae7a079c4f296c84109b2e6d7156b66e6b
cc39e386c3576d7535…
```

Broadcast search for a Device with Hostname CX_01234 and adding of a ADSSecure route via 'Preshared key' (UserName, Password) to the local system.

The target system must already contain the preshared key configuration (as PSK Identity/Password) in its StaticRoutes.xml configuration file.

Adding a route with BinaryKey (P)re (S)hared (Key) (PSK)

```
PS> $route = get-adsroute CX_01234 -all
PS> $route

Name             NetId                TLS   Address          FingerPrint
----             -----                ---   -------          -----------
CX_01234         172.17.60.197.1.1    X     172.17.60.197    7835dae7a079c4f296c84109b2e6d7156b66e6b
cc39e386c3576d7535…

PS> $route | add-adsroute -PreSharedKey -Identity MyUser -
```

```
BinaryKey 1,2,3,4,5,6,7,8,9,0xa,0xb,0xc,0xd,0xe,0xf

Name             NetId               TLS   Address          FingerPrint
----             -----               ---   -------          -----------
CX_01234         172.17.60.197.1.1   X     172.17.60.197    7835dae7a079c4f296c84109b2e6d7156b66e6b
cc39e386c3576d7535…
```

Broadcast search for a Device with Hostname CX_01234 and adding of a ADSSecure route via 'Preshared key' (Identity, BinaryKey) to the local system.

The target system must already contain the preshared key configuration (as Psk Identity/BinaryKey) in its StaticRoutes.xml configuration file.

Removing Routes by Address

```
PS> Get-AdsRoute

Name            NetId             Address       Sub TcVersion RTSystem
----            -----             -------       --- --------- --------
CP-15ECA0       192.168.0.128.1.1 192.168.0.178     0.0       Unknown
TC3TESTA1-CP67X 192.168.0.105.1.1 192.168.0.105     0.0       Unknown

PS> Remove-AdsRoute -Name "CP-15ECA0","TC3TESTA1*"
```

Removes the Routes "CP-15ECA0" and "TC3TESTA1-CP67X" from the local system.

Removing Routes from the local registered configuration

```
PS> Get-AdsRoute | Remove-AdsRoute -silent
```

Removes all registered routes from the local system.

5.5 Setting Device AmsNetId

⚠ CAUTION

CRASH and Reboot

With actual TwinCAT Versions the command crashes and reboots the device!

```
PS> Set-AmsNetId -NewId 1.1.1.1.1.1

Changeing AmsNetId of target system.
Change the NetId of system '192.168.0.2.1.1' to '1.1.1.1.1.1'
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "Y"): y
Changing the NetId of system '192.168.0.2.1.1' to '1.1.1.1.1.1' is succeeded. All preexisting connec
tions to this system a invalid now. A reboot of this system is necessary!
```

Sets the AmsNetId of the Local system to '1.1.1.1.1.1'.

5.6 Configure Device Realtime Settings

```
PS> Set-RTimeCpuSettings -SharedCores 6

Setting CPU cores
Setting WindowsCores: 6, IsolatedCores: 6 to device '192.168.0.146.1.1'?
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "Y"): y
Number of processors successfully set to '6'. A reboot is necessary to activate settings!
```

Sets the CPU Core Settings to 6 Shared and 6 Isolated on a 12 Core System.

5.7 (Re)Starting and Stopping TwinCAT

Restart TwinCAT (Succeeding)

```
PS> Restart-TwinCat -command Reset -force

Ok Target     NetId             Port  ErrorCode Requested Original Reached Latency (ms)
```

```
-- ------     -----             ----  --------- --------- -------- ------- ------------
X  CX_1111    192.168.0.2.1.1   10000 Succeeded Reset     Run      Run     2853
```

Restarts the local TwinCAT System. The ‘X’ in the ‘Ok’ Column indicates the success.

Restart TwinCAT (Failing)

```
PS> Restart-TwinCat -command Reset -force

WARNING: 192.168.0.2.1.1:10 ERR |
 18:26:28:108 |'TCOM Server' (10): Device 1 (EtherCAT) (Adapter): Failed to connect to network adapt
er!
WARNING: 192.168.0.2.1.1:10 WRN |
 18:26:28:108 |'TCOM Server' (10): PREOP to SAFEOP of 'Device 1 (EtherCAT) (Adapter)' (0x03010011) f
ailed - 'request is aborted' 0x9811071F
WARNING: 192.168.0.2.1.1:10000 ERR |
 18:26:28:117 |'TwinCAT System' (10000): Sending ams command >> Init12\IO: Set State TComObj SAFEOP:
 Set Objects (4) to SAFEOP >> AdsError: 1823 (0x71f, ADS ERROR: device aborted the action) << failed
!

Ok Target     NetId             Port  ErrorCode Requested Original Reached Latency (ms)
-- ------     -----             ----  --------- --------- -------- ------- ------------
   CX_1111    192.168.0.2.1.1   10000 Succeeded Reset     Config   Config  3427
```

Calls a Reset to the local SystemService that fails!

Error log messages will be logged out.

Restart TwinCAT and Analyze Errors on failed start.

```
PS> Restart-TwinCAT -command Reset -force | select-object -ExpandProperty LogMessages

WARNING: 192.168.0.2.1.1:10 ERR |
 18:20:45:969 |'TCOM Server' (10): Device 1 (EtherCAT) (Adapter): Failed to connect to network adapt
er!
WARNING: 192.168.0.2.1.1:10 WRN |
 18:20:45:969 |'TCOM Server' (10): PREOP to SAFEOP of 'Device 1 (EtherCAT) (Adapter)' (0x03010011) f
ailed - 'request is aborted' 0x9811071F
WARNING: 192.168.0.2.1.1:10000 ERR |
 18:20:45:979 |'TwinCAT System' (10000): Sending ams command >%gt; Init12\IO: Set State TComObj SAFE
OP: Set Objects (4) to SAFEOP >%gt; AdsError: 1823 (0x71f, ADS ERROR: device aborted the action) <<
failed!

Type    TimeStamp    DeviceName     Port  Message
----    ---------    ----------     ----  -------
Message 18:20:44.874 TwinCAT System 10000 TwinCAT System Restart initiated from AmsNetId: 192.168.0.
2.1.1 port 34564.
Message 18:20:44.879 TwinCAT System 10000 Saving configuration of COM server TcVnService !
Message 18:20:44.880 TwinCAT System 10000 Saving configuration of COM server TcEventLogger !
Message 18:20:44.970 TwinCAT System 10000 Shutting down COM Server TcVnService !
Message 18:20:44.971 TwinCAT System 10000 Shutting down COM Server TcEventLogger !
Message 18:20:45.745 TwinCAT System 10000 Loading configuration of COM server TcVnService !
Message 18:20:45.747 TwinCAT System 10000 Loading configuration of COM server TcEventLogger !
Message 18:20:45.748 TwinCAT System 10000 Initializing COM Server TcVnService !
Message 18:20:45.773 TwinCAT System 10000 Initializing COM Server TcEventLogger !
Message 18:20:45.783 TwinCAT System 10000 TcIoEth Server started: TcIoEth.
Message 18:20:45.791 TwinCAT System 10000 TcRtsObjects Server started: TcRtsObjects.
Message 18:20:45.798 TwinCAT System 10000 TcIoECat Server started: TcIoECat.
Message 18:20:45.805 TwinCAT System 10000 TcIo Server started: TcIo.
Message 18:20:45.814 TwinCAT System 10000 TcPlc30 Server started: TcPlc30.
Message 18:20:45.821 TwinCAT System 10000 TcRTime Server started: TcRTime.
Message 18:20:45.927 License Server 30    license validation status is Valid(3)
Error   18:20:45.969 TCOM Server    10    Device 1 (EtherCAT) (Adapter): Failed to connect to networ
k adapter!
Warning 18:20:45.969 TCOM Server    10    PREOP to SAFEOP of 'Device 1 (EtherCAT) (Adapter)' (0x0301
0011) failed - 'request is aborted' 0x9811071F
Error   18:20:45.979 TwinCAT System 10000 Sending ams command >%gt; Init12\IO: Set State TComObj SAF
EOP: Set Objects (4) to SAFEOP >%gt; AdsError: 1823 (0x71f, ADS ERROR: device aborted the action) %l
t;%lt; failed!
Message 18:20:47.879 TwinCAT System 10000 Loading configuration of COM server TcVnService
Message 18:20:47.881 TwinCAT System 10000 Loading configuration of COM server TcEventLogger
Message 18:20:47.882 TwinCAT System 10000 Initializing COM Server TcVnService
Message 18:20:47.910 TwinCAT System 10000 Initializing COM Server TcEventLogger
Message 18:20:47.922 TwinCAT System 10000 TCIO Server started: TCIO.
Message 18:20:47.931 TwinCAT System 10000 TCRTIME Server started: TCRTIME.
Message 18:20:47.939 TwinCAT System 10000 TCRTSOBJECTS Server started: TCRTSOBJECTS.
Message 18:20:47.948 TwinCAT System 10000 TCIOETH Server started: TCIOETH.
```

```
Message 18:20:47.956 TwinCAT System 10000 TCIOECAT Server started: TCIOECAT.
Message 18:20:47.963 TwinCAT System 10000 TCIODRIVERS Server started: TCIODRIVERS.
Message 18:20:48.078 TwinCAT System 10000 Starting COM Server TcVnService
Message 18:20:48.078 TwinCAT System 10000 Starting COM Server TcEventLogger
```

Tries to restart the local TwinCAT system and write the log messages to the output.

This Command fails.

5.8 Starting and Stopping the PLC and other ADS Servers

Start the PLC on the local system

```
PS> Set-AdsState -port 851 -command Run -force

Ok Target    NetId             Port  ErrorCode            Requested   Original Reached Latency (ms)
-- ------    -----             ----  ---------            ---------   -------- ------- ------------
X  CX-11111  1.1.1.1.1.1        851  NoError              Run         Stop     Run     293
```

The ‘X’ indicates the success.

Start the PLC on all registered routes

```
PS> $r = Get-AdsRoute
PS> Set-AdsState -port 851 -command Run -InputObject $r -force

Ok Target    NetId             Port  ErrorCode            Requested   Original Reached Latency (ms)
-- ------    -----             ----  ---------            ---------   -------- ------- ------------
X  CX-11111  1.1.1.1.1.1        851  NoError              Run         Stop     Run     293
X  CX-22222  1.1.1.2.1.1        851  NoError              Run         Stop     Run     357
X  CX-22222  1.1.1.3.1.1        851  NoError              Run         Stop     Run     218
X  CX-22222  1.1.1.4.1.1        851  NoError              Run         Stop     Run     324
```

Start the PLC on all registered target systems. The ‘X’ indicates the success.

Restart the TwinCAT System Service

```
PS> Set-AdsState -port 10000 -command Reset -force

Ok Target     NetId             Port  ErrorCode Requested Original Reached Latency(ms)
-- ------     -----             ----  --------- --------- -------- ------- ------------
X  MySystem   172.168.0.1.1.1  10000  NoError   Reset     Run      Run     5007
```

Restart the local System Service. This is an alternative to the ‘**Restart-TwinCAT**’ Cmdlet.

5.9 Reading / Writing Data

5.9.1 Connect to a target device / ADS Server

In the **TwinCAT** world, every software/hardware instance participating at the **ADS** communication network is an ADS Server. To access these servers various implementations of ADS Clients for different programming languages are available.

To establish an ADS connection in Powershell, the **‘New-TcSession’** Cmdlet is used. Internally it uses the .NET **‘TwinCAT.Ads.AdsClient’** object for communicating.

If using the **-NetId**, **-Port**, **-Route** or **-Address** Arguments of the **‘TcXaeMgmt’** Module Cmdlets, the connection (and therefore the AdsClient) is established on the fly. The connection is shut down after the single Cmdlet is executed.

The advantage of the ‘Session’ is, that leaves the connection established until **‘Close-TcSession’** is called, or the Powershell host process (the Powershell-Session) ends.

Existing ADS Sessions can be stored in variables or can be accessed by **‘Get-TcSession’.**

These Cmdlets use ArgumentCompleters proposing registered Routes.

Open an ADS Session

```
PS> New-TcSession -NetId '172.17.62.105.1.1' -port 851

ID Address                Connected State     Cycles Losses LastError Established         LastSuccee
d
-- -------                --------- -----     ------ ------ --------- -----------
          ------------
5  172.17.62.105.1.1:851 True       Succeeded 0      0               2024-01-08T12:50:58
```

Establishes a new Ads Session/Connection to the specified NetId/Port address.

Opens an ADS Session, use it and closes it afterwards

```
PS> $session = New-TcSession -NetId '1.2.3.4.1.1' -port 10000
PS> $session | Get-AdsState

Target               NetId             Port   State      Latency
                                                           (ms)
------               -----             ----   -----      -------
CX_1234              1.2.3.4.1.1       10000  Config     3

PS> $session | Close-TcSession
```

Opens a session to the registered route with AmsNetId: 1.2.3.4.1.1 and closes the ADS Session again.

Open an ADS Session with wildcard name pattern.

```
PS> $route = Get-AdsRoute -Name "Tc3*"
PS> $session = New-TcSession -Route $route -Port 851
PS> $session

ID Address                Connected State     Cycles Losses LastError Established         LastSuccee
d
-- -------                --------- -----     ------ ------ --------- -----------
          ------------
5  172.17.62.105.1.1:851  True      Succeeded 0      0               2024-01-08T12:50:58
```

Establishes a new ADS Session/Connection to the specified route destination that has the name pattern "tc3*" via port 851 (PLC1).

List established ADS Sessions

```
PS> Get-TcSession

ID Address                Connected State     Cycles Losses LastError Established         LastSuccee
d
-- -------                --------- -----     ------ ------ --------- -----------
          ------------
5  172.17.62.105.1.1:851  True      Succeeded 0      0               2024-01-08T12:50:58
```

Lists all actual initiated sessions.

Close ADS Sessions

```
PS> Close-TcSession -Id 5
```

Closes the session with id 5.

5.9.2 Symbolic / Handle / IndexGroup IndexOffset Access

5.9.2.1 Symbolic Access

Because the symbolic information can be downloaded from the target (done by **Get-TcSymbol**), the **Read-TcValue** and **Write-TcValue** Cmdlets are able to use .NET objects type-safe and seamlessly. All data – if possible – is marshaled transparently via the network.

Reading a value using a symbol object

```
PS> $session = New-TcSession -NetId '1.2.3.4.5.6' -Port 851
PS> $symbol = $session | get-TcSymbol -Path 'TwinCAT_SystemInfoVarList._AppInfo.ProjectName'
PS> $symbol | Read-TcValue

ADS_DynSymbols
```

Create an ADS Session/Connection, determine the 'ProjectName' Symbol from the running PLC Project, read the current value of the symbol and print it to the console.

5.9.2.2 Handle Access

Register/Unregister a handle to a Symbol Path

```
PS> $session = New-TcSession -NetId '1.2.3.4.5.6' -Port 851
PS> $handleInfo = $session | Register-AdsHandle -
Path 'TwinCAT_SystemInfoVarList._AppInfo.ProjectName'
PS> $handleInfo = Register-AdsHandle -Path 'TwinCAT_SystemInfoVarList._AppInfo.ProjectName' -
Session $s
PS> $handleInfo

InstancePath                                    Result  Handle
------------                                    ------  ------
TwinCAT_SystemInfoVarList._AppInfo.ProjectName NoError 0x428000FC (1115685116)

PS> Read-TcValue -Session $session -IndexGroup SymbolValueByHandle -IndexOffset $handleInfo.Handle -
Type String

MyProject

PS> $handle | Unregister-AdsHandle -Session $session
PS> $session | Close-tcsession
```

Opens a new device session, registers a Symbol Handle to the ProjectName of the running PLC Project, Reads the value by handle unregisters the handle and closes the session again.

5.9.2.3 IndexGroup/IndexOffset Access

Read from IndexGroup/IndexOffset and cast to a .NET type

```
PS> Read-TcValue -session $session -IndexGroup 0x4040 -IndexOffset 0x1247A8 -ValueType String
ADS_DynSymbols
```

Reads a string typed value from IndexGroup / IndexOffset.

In this example the ProjectName of the running PLC Project resides at that ProcessImage Address.

Read from IndexGroup/IndexGroup generically (number of bytes)

```
PS> Read-TcValue -IndexGroup 0x4040 -IndexOffset 0x1247a8 -NetId 192.168.0.105.1.1 -port 851 -
size 0xff | format-hex


           00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000   41 44 53 5F 44 79 6E 53 79 6D 62 6F 6C 73 00 00  ADS_DynSymbols..
00000010   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000020   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000030   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000040   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000050   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000060   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000070   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000080   11 00 01 01 A0 86 01 00 14 00 5E 01 21 C2 15 00  .... ?....^.!A..
00000090   00 7F F1 57 3B 83 6C 07 1E 00 00 00 00 00 00 00  .ôñW;?l.........
000000A0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000B0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000C0   41 44 53 5F 44 79 6E 53 79 6D 62 6F 6C 73 5F 50  ADS_DynSymbols_P
000000D0   6C 63 54 61 73 6B 00 00 00 00 00 00 00 00 00 00  lcTask..........
000000E0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000F0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00      ................
```

Reads 256 Bytes via IndexGroup/IndexOffset from the specified target system and prints the out formatted as hexdump.

5.9.3 Read Write Values

5.9.3.1 Read

The **Read-TcValue** Cmdlet can be called via IndexGroup/IndexOffset (Raw Data), via Handle or with Symbolic Information (Symbol path or Symbol Object). If symbolic information is used, then the calls will be type safe and the returned data is automatically mapped to appropriate (dynamically) created .NET types.

Read type safe data with symbolic information

```
PS> $session = New-TcSession -NetId '1.2.3.4.5.6' -Port 851
PS> $symbol = $session | get-TcSymbol -Path 'TwinCAT_SystemInfoVarList._AppInfo.ProjectName'
PS> $symbol | Read-TcValue

ADS_DynSymbols
```

Create an ADS Session/Connection, determine the 'ProjectName' Symbol from the running PLC Project, read the current value of the symbol and print it to the console.

Read amount of bytes from IndexGroup/IndexOffset

```
PS> Read-TcValue -IndexGroup 0x4040 -IndexOffset 0x1247a8 -NetId 172.17.62.105.1.1 -port 851 -
size 0xff | format-hex


           00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000   41 44 53 5F 44 79 6E 53 79 6D 62 6F 6C 73 00 00  ADS_DynSymbols..
00000010   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000020   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000030   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000040   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000050   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000060   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000070   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000080   11 00 01 01 A0 86 01 00 14 00 5E 01 21 C2 15 00  .... ?....^.!A..
00000090   00 7F F1 57 3B 83 6C 07 1E 00 00 00 00 00 00 00  .⌂ñW;?l.........
000000A0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000B0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000C0   41 44 53 5F 44 79 6E 53 79 6D 62 6F 6C 73 5F 50  ADS_DynSymbols_P
000000D0   6C 63 54 61 73 6B 00 00 00 00 00 00 00 00 00 00  lcTask..........
000000E0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000F0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00      ................
```

Reads 256 Bytes via IndexGroup/IndexOffset from the specified target system and prints the out formatted as hexdump.

Read data with IndexGroup/IndexOffset and specify the returned data type.

```
PS> Read-TcValue -session $session -IndexGroup 0x4040 -IndexOffset 0x1247A8 -ValueType String
ADS_DynSymbols
```

Reads a string typed value from IndexGroup / IndexOffset.

In this example the ProjectName of the running PLC Project resides at that ProcessImage Address.

Read data by Symbol handle

```
PS> $route = Get-AdsRoute -Name 'CX-123456'
PS> $session = $route | New-TcSession -Port 851
PS> $handle = $session | Send-TcReadWrite -IndexGroup SymbolHandleByName -WriteValue "GVL.vgInt" -
ReadType Int32 -force
PS> $session | Read-TcValue -IndexGroup SymbolValueByHandle -IndexOffset $handle -ValueType Int16
42
```

Create a session to the PLC (Port 851) of a target system, determine the SymbolHandle by InstancePath and use this handle to read its 'Int16' Value (INT on PLC System).

Accessing Complex Objects (e.g Structs or FBs)

Also complex types will be resolved transparently. E.g. a DT type with 3 fields (vBool, vInt, vString) on the PLC side can be accessed as a whole.

Reading the Symbol Information

```
PS> $symbol2 = $session | Get-TcSymbol -path 'MAIN.vmSimpleStruct'
```

Getting the value of the struct. The subvalues will be created type-safe on-the-fly as .NET Types

```
PS> $val2 = $symbol2 | Read-TcValue
PS> $val2

vBool   vInt vString                          PSValue
-----   ---- -------                          -------
 True -12121 QWERTZUIOPÜASDFGHJKLÖÄYXCVBNM;:_ ...

PS> $val2 | Get-Member

   TypeName: System.Management.Automation.PSCustomObject

Name        MemberType   Definition
----        ----------   ----------
Equals      Method       bool Equals(System.Object obj)
GetHashCode Method       int GetHashCode()
GetType     Method       type GetType()
ToString    Method       string ToString()
PSValue     NoteProperty DynamicValue PSValue=...
vBool       NoteProperty bool vBool=True
vInt        NoteProperty short vInt=-12121
vString     NoteProperty string vString=QWERTZUIOPÜASDFGHJKLÖÄYXCVBNM;:_
```

Also SubValues can be accessed. This doesn't trigger a new ADS Read (and uses internally cached data). The data is consistent to the timepoint of the 'Read-TcValue'

```
PS> $val2.vString
QWERTZUIOPÜASDFGHJKLÖÄYXCVBNM;:_
```

To access the more detailed internal Data of the Value, the 'PSValue' can be accessed.

```
PS> $val2.PSValue

Symbol              DataType        ByteSize TimeStamp           CachedRaw
------              --------        -------- ---------           ---------
MAIN.vmSimpleStruct ST_SimpleStruct 165      2024-01-11T15:24:33 01 A7 D0 51 00 57 00 45 00 52 00 54
 00 5A 00 55

PS> $val2.PSValue | format-list *

ValueFactory : TwinCAT.ValueAccess.DynamicValueFactory
TimeStamp    : 1/11/2024 3:24:33 PM +01:00
Symbol       : MAIN.vmSimpleStruct (IG: 0x4040, IO: 0x10e7b8, Size: 165 bytes)'
UpdateMode   : None
ParentValue  :
RootValue    : ...
Age          : 00:07:18.2760087
DataType     : ST_SimpleStruct
CachedRaw    : System.ReadOnlyMemory<Byte>[165]
IsPrimitive  : False
vBool        : True
vInt         : -12121
vString      : QWERTZUIOPÜASDFGHJKLÖÄYXCVBNM;:_
```

5.9.3.2 Write

The **Write-TcValue** Cmdlet can be called via IndexGroup/IndexOffset (Raw Data), via Handle or with Symbolic Information (Symbol path or Symbol Object). If symbolic information is used, then the calls will be type safe and the write data is automatically mapped to appropriate (Plc) Data on the target side, dynamically.

Write Symbolic Data

```
PS> $session = New-TcSession -NetId 1.2.3.4.5.6 -Port 851
PS> $projectNameSymbol = $session | Get-TcSymbol -path "*ProjectName"
PS> $projectNameSymbol

InstanceName DataType   Size InstancePath
------------ --------   ---- ------------
ProjectName  STRING(63) 64   TwinCAT_SystemInfoVarList._AppInfo.ProjectName

PS> $projectNameSymbol | Read-TcValue

OldProjectName

PS> $projectNameSymbol | Write-TcValue -Value "NewProjectName" -force
PS> $projectNameSymbol | ReadTcValue

NewProjectName
```

This example shows how to create a session, determining the Symbol 'ProjectName within the _AppInfo Struct on a running PLC project and reading its value.

After that, the Value will be overwritten with 'NewProjectName'.

Write IndexGroup/IndexOffset Data

```
Write-TcValue -session $session -IndexGroup 0x4040 -IndexOffset 0x1247A8 -Value "NewProjectName"
```

Writes a string typed Value to the specified IndexGroup/IndexOffset Address.

5.9.3.3 ReadWrite

ADS ReadWrite requests send data to the ADS Server and get a Data response in one roundtrip. The Powershell command to request an ADS ReadWrite is the **Send-TcReadWrite** Cmdlet.

Send an ADS ReadWrite Request with IndexGroup/IndexOffset (Return Data as string)

```
PS> Send-TcReadWrite -NetId 1.2.3.4.5.6 -Port 851 -IndexGroup SymbolValueByName -IndexOffset 0 -
WriteValue "TwinCAT_SystemInfoVarList._AppInfo.ProjectName" -ReadType string -ReadLength 1024

ReadWrite access of process image on target '1.2.3.4.5.6:851':
Start ReadWrite operation WriteData: 'TwinCAT_SystemInfoVarList._AppInfo.ProjectName'(IG:0xf004,IO:0
x0000,Len:47),
Read: Type 'System.String' (Len:'1024) on target '1.2.3.4.5.6:851'?
[Y] Yes[A] Yes to All[N] No[L] No to All[S] Suspend[?] Help(default is "Y"): y
ADS_DynSymbols
```

Sends a Read/Write request with index group 0xf004 (SymbolValueByName) and offset 0.

The write data will be initialized with the project symbol path and an returned (read) string (Default encoded) returned.

Send an ADS ReadWrite Request with IndexGroup/IndexOffset (return data as byte{})

```
PS> Send-TcReadWrite -NetId 1.2.3.4.5.6 -Port 851 -IndexGroup SymbolValueByName -
WriteValue "TwinCAT_SystemInfoVarList._AppInfo.ProjectName" -ReadLength 64 | format-hex

ReadWrite access of process image on target '1.2.3.4.5.6:851':
Start ReadWrite operation WriteData: 'TwinCAT_SystemInfoVarList._AppInfo.ProjectName'(IG:0xf004,IO:0
x0000,Len:47),
Read: Type 'System.Byte[]' (Len:'64) on target '1.2.3.4.5.6:851'?
[Y] Yes[A] Yes to All[N] No[L] No to All[S] Suspend[?] Help(default is "Y"): y
```

```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000   41 44 53 5F 44 79 6E 53 79 6D 62 6F 6C 73 00 00  ADS_DynSymbols..
00000010   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000020   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000030   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

Sends a Read/Write request with index group 0xf004 (SymbolValueByName) and offset 0.

The write data will be initialized with the project symbol path and the returned (read) data is by default a byte array of 64 bytes.

The result value will be formatted as hex code.

Send an ADS ReadWrite Request with IndexGroup/IndexOffset (using handle)

```
PS> $route = Get-AdsRoute -Name 'CX-123456'
PS> $session = $route | New-TcSession -Port 851
PS> $handle = $session | Send-TcReadWrite -IndexGroup SymbolHandleByName -WriteValue "GVL.vgInt" -
ReadType Int32 -force
PS> $session | Read-TcValue -IndexGroup SymbolValueByHandle -IndexOffset $handle -ValueType Int16
42
```

Create a session to the PLC (Port 851) of a target system, determine the SymbolHandle by InstancePath and use this handle to read its 'Int16' Value (INT on PLC System).

5.9.4 Calling RpcMethods and accessing PLC Properties

RpcMethods can be directly called type safe on the value dynamically read by a symbolic **Read-TcValue.** The RpcMethod In-/ and Out-/ Parameters are automatically marshalled to/from they .NET counterparts – so that the access is easy and transparently.

Accessing the RPC Methods needs to open an ADS Session and loading the symbolic information. Here we open a new session to the local PLC.

```
PS> $session = new-tcSession -port 851
PS> $rpcSymbol | Get-TcSymbol -path 'MAIN.fbRpc'
```

The ‘rpcSymbol’ object now contains the Symbol of the FB instance ‘fbRcp’.

This FB has a RpcMethod ‘AddValues’ with 2 INT parameters and a Property ‘PropInt’ defined (on the PLC side).

Information about these defined symbol instance can be browsed.

```
PS> $rpcSymbol

InstancePath Category DataType  Size Static Persistant IG   IO
------------ -------- --------  ---- ------ ---------- --   --
MAIN.fbRpc   Struct   FB_RpcPou 1152 False  False      4040 10DEE8
```

Beneath Properties and Methods of the Instance base class, the ‘rpcSymbol’ contains dynamically generated access properties and methods.

```
PS> $rpcSymbol | Get-Member

   TypeName: TwinCAT.TypeSystem.DynamicStructInstance

…
Name                       MemberType Definition
----                       ---------- ----------
AddValues                  Dynamic    dynamic AddValues
AddValuesAsync             Dynamic    dynamic AddValuesAsync
PropInt                    Dynamic    dynamic PropInt
….
```

Accessing the Rpc Method metadata of the symbol:

```
PS> $rpcSymbol.RpcMethods

Name                 ReturnType            Declaration
```

```
----                 ----------          -----------
AddValues            INT                 INT AddValues([in] INT i1,[in] INT i2)
__getPropInt         INT                 INT __getPropInt()
__setPropInt                             __setPropInt([in] INT PropInt)
```

The method can be called transparently in Powershell

```
PS> $rpcSymbol.AddValues(39,3)
42
```

The same is true for the dynamic property

```
PS> $rpcSymbol.PropInt

InstancePath       Category  DataType Size Static Persistant IG   IO
------------       --------  -------- ---- ------ ---------- --   --
MAIN.fbRpc.PropInt Primitive INT      2    False  False      4040 10DEE8

PS> $rpcSymbol.PropInt | Read-TcValue

29224
```

5.10 TwinCAT / Device Diagnosis

5.10.1 Testing Device Availability and Latencies

The **Test-AdsRoute** Cmdlet provides information about

- System availability
- AdsServer existence
- Available System Ports
- Access latencies/roundtrip times

Check local PLC availability

```
PS> Test-AdsRoute -Port 851

Name                 NetId             Port   Latency Result
                                              (ms)
----                 -----             ----   ------- ------
CX-11111             192.168.0.2.1.1   851    3       Ok
```

Test the Port 851 of the local system (PLC 1) for availablity.

Check the registered routes for availabilty

```
PS> Get-AdsRoute | Test-AdsRoute

Name                 NetId             Port   Latency Result
                                              (ms)
----                 -----             ----   ------- ------
CX-11111             192.168.0.2.1.1   10000  4       Ok
CX-22222             192.168.0.3.1.1   10000          Failed
CX-33333             192.168.0.4.1.1   10000  4       Ok
```

Get the locally registered routes and test if they are reachable (on AmsPort 10000)

Port scan of the local TwinCAT System

```
PS> Test-AdsRoute -OnlinePorts

Name                 NetId             Port   Latency Result
   (ms)
----                 -----             ----   ------- ------
CX-11111             192.168.0.2.1.1   10     0.6     Ok
CX-11111             192.168.0.2.1.1   11     1.3     Ok
CX-11111             192.168.0.2.1.1   12     1.2     Ok
CX-11111             192.168.0.2.1.1   30     3       Ok
```

```
CX-11111             192.168.0.2.1.1   131    75      Ok
CX-11111             192.168.0.2.1.1   32829  125     Ok
CX-11111             192.168.0.2.1.1   340    122     Ok
CX-11111             192.168.0.2.1.1   850    171     Ok
CX-11111             192.168.0.2.1.1   32830  174     Ok
CX-11111             192.168.0.2.1.1   351    171     Ok
CX-11111             192.168.0.2.1.1   350    172     Ok
CX-11111             192.168.0.2.1.1   270    219     Ok
CX-11111             192.168.0.2.1.1   851    220     Ok
```

Scans the propagated AmsPorts for the local system.

Measure latencies of a specific ADS Port

```
Test-AdsRoute -port 10000 -count 10

Name     NetId            Port  Latency Result
(ms)
----     -----            ----  ------- ------
MYSYSTEM 192.168.56.1.1.1 10000 1.5     Ok
MYSYSTEM 192.168.56.1.1.1 10000 1.6     Ok
MYSYSTEM 192.168.56.1.1.1 10000 3       Ok
MYSYSTEM 192.168.56.1.1.1 10000 1.9     Ok
MYSYSTEM 192.168.56.1.1.1 10000 1.9     Ok
MYSYSTEM 192.168.56.1.1.1 10000 1.8     Ok
MYSYSTEM 192.168.56.1.1.1 10000 1.9     Ok
MYSYSTEM 192.168.56.1.1.1 10000 1.9     Ok
MYSYSTEM 192.168.56.1.1.1 10000 3       Ok
MYSYSTEM 192.168.56.1.1.1 10000 1.7     Ok
```

5.10.2 Get the local AmsNetId or Route

Get the Local AmsNetId

```
PS> Get-AmsNetId

192.168.0.2.1.1
```

Gets the AmsNetId of the Local system.

Get the Local System as Route object

```
PS> Get-AdsRoute -local

Name                            NetId               Protocol   TLS   Address          FingerPrint
----                            -----               --------   ---   -------          -----------
MYSYSTEM                        192.168.56.1.1.1    TcpIP      X     192.168.60.213   2f72d63cba30
69b5e15a3983fbfa1da7914f7c1…
```

5.10.3 Get information about a TwinCAT device

Getting information from Broadcast Search

```
PS> Get-AdsRoute -Name CX-1CEEDA -All
Name            NetId               Address       Sub Version   RTSystem
----            -----               -------       --- -------   --------
CX-1CEEDA       5.16.136.222.1.1    192.168.0.139     3.1.4020  Win7
```

Getting extended information about the Hardware, Image and OS

```
PS> Get-TcTargetInfo

Target  NetId            Version    OS    Image Device CPUArch SystemId
Fingerprint
------  -----            -------    --    ----- ------ ------- --------
                             -----------
CX_1111 192.168.0.1.1.1  3.1.4026.2 Win10              AMD64   5b42297a-a9dd-6623-1780-
e52074e54f71 2f72d63cba3069b5e1…
```

Getting TwinCAT License information

```
PS> $session = New-TcSession -Route TC3TESTA1-CP67X -Port 30
PS> Get-TcLicense -Status All -name *scope* -session $session

Name                          Valid ValidityCode ExpireTime   Available Used VolumeNo
----                          ----- ------------ ----------   --------- ---- --------
TC3 Scope Server                  X        Valid            CPU License      0        0
TC3 Scope View Professional       X        Valid            CPU License      0        0
```

Create a session to the License Server on target 'TC3TESTA1-CP67X' and return all valid and invalid licenses that contain 'scope' in their name.

Getting actual TwinCAT Router information of a target system

```
PS> Get-TcRouterInfo

Target  Result TotalMem(kb)  AvailMem(kb)  Ports Drivers Transports Mailbox Size(kb)  Mailbox Queue
------  ------ ------------- ------------- ----- ------- ---------- ----------------- -------------
CX_1234 Ok     32768         32759         31    4       11         0                 0
```

Get router information from the local system.

5.10.4 Accessing Symbolic Information

Getting Root Symbols

```
PS> Get-TcSymbol -port 851
InstanceName              DataType Size InstancePath
------------              -------- ---- ------------
tc2vBool                  BOOL     1    .tc2vBool
tc2vInt                   INT      2    .tc2vInt
Constants                          0    Constants
GVL                                0    GVL
MAIN                               0    MAIN
Slow                               0    Slow
TwinCAT_SystemInfoVarList          0    TwinCAT_SystemInfoVarList
```

Get the root symbolic information from the local system (Port 851):

Browsing through Symbols recursively and filter with wildcards

```
PS> $session = New-TcSession -Name 'CX_123456' -port 851
PS> $session | Get-TcSymbol -recurse | where InstanceName -like 'Project*'

 InstanceName DataType   Size InstancePath
 ------------ --------   ---- ------------
 ProjectName  STRING(63) 64   TwinCAT_SystemInfoVarList._AppInfo.ProjectName
```

Gets an ADS-Session/Connection to the target system CX_123456 on port 851, downloads the symbol information recursively and returns all Instances where the instance name is like the pattern 'Project*'.

Browsing DataTypes

```
PS> Get-TcDataType -port 851
Name                      Size     Category   BaseType
----                      ----     --------   --------
BYTE                      1        Primitive
WORD                      2        Primitive
DINT                      4        Primitive
UDINT                     4        Primitive
DWORD                     4        Primitive
E_ByteEnum                1        Enum       BYTE
FB_Test                   12424    Struct
PLC.PlcAppSystemInfo      256      Struct
PLC.PlcTaskSystemInfo     128      Struct
POINTER TO BYTE           4        Pointer    BYTE
R_Range                   2        Alias      INT (-6..12)
REFERENCE TO BOOL         4        Reference  BOOL
ST_SimpleStruct           166      Struct
```

```
STRING(80)                 81       String
...
```

5.10.5 Getting Realtime Performance Information

Get the Realtime CPU Settings

```
PS> Get-RTimeCpuSettings

NetId                Windows    NonWin     RealTime  Cpu Type  Cpu      CpuFrequency
                     Cores      Cores      Cores               Family   (GHz)
-----                ---------- ---------- ---------- -------- -------- ------------
172.17.60.167.1.1    22         2          1           0        4        3793
```

Getting the CPU Settings of the local system.

Get the Realtime Latency of the CPU

```
PS> Get-RTimeLatency

NetId                CoreId Latency    MaxLatency Limit
                            (us)         (us)
---------------      ------ ---------- ---------- -----
5.91.172.198.1.1     1      0          20         0
```

Getting the Realtime latency of all Realtime cores on the local system.

Test the Realtime Latency of specific CPU cores.

```
PS> Get-RTimeLatency -NetId 5.91.172.198.1.1  -core 1 -count 5 -Delay 0

NetId                CoreId Latency    MaxLatency Limit
                            (us)         (us)
---------------      ------ ---------- ---------- -----
5.91.172.198.1.1     1      0          20         0
5.91.172.198.1.1     1      0          20         0
5.91.172.198.1.1     1      0          20         0
5.91.172.198.1.1     1      0          20         0
5.91.172.198.1.1     1      0          20         0
```

Get the Realtime Latency of the System with NetId 5.91.172.198.1.1 and CoreId 1 5 times without delay between values.

Get Realtime Performance Date of all CPU cores.

```
PS> Get-RTimePerformance

NetId                CoreId LastDelay    MaxDelay    DelayLimit Load (%) MaxLoad
                            (us)         (us)        (us)               (%)
-----                ------ ---------- ---------- ---------- -------- --------
192.168.0.2.1.1      1      0          109        0          0        80
192.168.0.2.1.1      2      0          109        0          0        80
```

Getting the Performance Data from all Realtime CPUs on the local target system.

Test Realtime Performance Data of specific CPU cores.

```
PS> Get-RTimePerformance -core 1 -count 5 -Delay 0 -noReset

NetId                CoreId LastDelay  MaxDelay   DelayLimit Load (%) MaxLoad
                            (us)       (us)        (us)               (%)
-----                ------ ---------- ---------- ---------- -------- --------
192.168.0.2.1.1      1      0          1659       0          0        80
192.168.0.2.1.1      1      0          1659       0          0        80
192.168.0.2.1.1      1      0          1659       0          0        80
192.168.0.2.1.1      1      0          1659       0          0        80
192.168.0.2.1.1      1      0          1659       0          0        80
```

Getting the Performance Data from the local System (Core 1) 5 times as fast as possible.

The MaxDelay will not be reset on each call.

6 TcXaeMgmt Version 6.X

This is the Platform independent version of the ‘TcXaeMgmt’ Module. This can run on all Platforms that are supported by ‘Microsoft Powershell’ and ‘Windows Powerhell’ >= Version 5.

Differences between Microsoft Powershell and Windows Powershell are documented here.

Supported TwinCAT Versions are TwinCAT 3.1.4024 and newer. If an older Version of TwinCAT is installed locally, please use the Version 3.X series of the ‘TcXaeMgmt’ module.

6.1 AdsFileProvider

```
PS> New-PSDrive -name CX_01234 -PSProvider AdsFileProvider -Address CX_01234 -Root ''
PS> dir CX_01234:

Mode              LastWriteTime                    Length     Name
----              -------------                    ------     ----
d----             30.11.2021 16:11:31                         BootDir
d----             03.12.2021 01:17:20                         BootProject
d----             17.03.2021 14:33:53                         ConfigDir
d----             03.12.2021 01:17:20                         Generic
d----             18.06.2021 08:00:22                         InstallDir
d----             03.12.2021 01:17:20                         RepositoryDir
d----             03.12.2021 15:32:03                         TargetDir

> cd CX_01234:/BootDir

PS CX_01234:\BootDir> dir

Mode              LastWriteTime                    Length     Name
----              -------------                    ------     ----
d----             05.10.2021 10:36:34                         CurrentConfig
-a---             05.10.2021 10:36:34              4563       CurrentConfig.tszip
-a---             05.10.2021 10:36:34              17113      CurrentConfig.xml
-a---             30.11.2021 16:11:31              126976     LoggedEvents.db
d----             27.10.2021 11:32:43                         Plc
```

More Information about Providers

```
PS> get-help about_providers
```

Example: Create a new AdsFileProvider Drive to the TwinCAT Device CX_01234

```
> New-PSDrive -name CX_01234 -PSProvider AdsFileProvider -Address CX_01234 -Root ''

Name           Used (GB)     Free (GB) Provider        Root                 CurrentLocation
----           ---------     --------- --------        ----                 ---------------
CX_01234                               AdsFileProvider \TargetDir
```

Example: Browse the files on the TwinCAT Device CX_01234

```
> dir

Mode              LastWriteTime                    Length     Name
----              -------------                    ------     ----
d----             26.11.2021 17:44:27                         CACerts
-a---             14.03.2012 14:50:50              619        DefaultConfig.xml
d----             11.05.2021 14:42:45                         License
d----             18.06.2021 08:01:03                         Resource
d----             17.03.2021 15:15:51                         Routes
d----             18.06.2021 08:00:33                         StartMenuAdmin
d----             17.03.2021 14:33:35                         StartUp
-a---             30.11.2021 18:46:08              2253       StaticRoutes.xml
-a---             01.02.2012 16:42:58              494        TargetFeatures.xml
-a---             17.03.2021 14:42:50              3113       TcSelfSigned.xml
```

Example: Read the content of the StaticRouts.xml on target CX_01234

```
r> get-content .\StaticRoutes.xml
<?xml version="1.0"?>
<TcConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```
        <RemoteConnections>
                <Route>
                        <Name>TargetIPC</Name>
                        <Address>172.17.60.147</Address>
                        <NetId>172.17.60.147.1.1</NetId>
                        <Type>TCP_IP</Type>
                        <Tls IgnoreCn="true">
                                <Ca>...</Ca>
                        </Tls>
                </Route>
                <Server>
                        <Tls IgnoreCn="true">
                                <Ca>c:\twincat\3.1\target\CACerts\RootCA.pem</Ca>
                                <Cert>c:\twincat\3.1\target\CACerts\TargetIPC.crt</Cert>
                                <Key>c:\twincat\3.1\target\CACerts\TargetIPC.key</Key>
                        </Tls>
                </Server>
        </RemoteConnections>
</TcConfig>
```

6.2 AdsSymbolProvider

Binds the target device symbolic information to a PSDrive. To register a symbol server as

PSDrive type (here the Target Route 'CX_01234' with AmsPort: 851)

```
PS> New-PSDrive -Name CX_01234_Symbols -PSProvider AdsSymbolProvider -Address CX_01234 -Port 851 -
Root
PS> cd CX_01234_Symbols:
PS> CX_01234_Symbols:> dir
```

6.3 TcXaeMgmt

about_TcXaeMgmt

PowerShell TwinCAT XAE Management Console (**TcXaeMgmt**)

SHORT DESCRIPTION

Cmdlets for managing and accessing ADS Routes, Reading/Writing Values and managing Remote targets.

LONG DESCRIPTION

The Powershell TwinCAT Management Console is a PowerShell module that provides

a number of useful cmdlets for TwinCAT System Management and for communicating

with ADS devices over the ADS protocol.

This includes the following tasks/features:

- Establishing/Removing Route Connections (**Add-AdsRoute**, **Remove-AdsRoute**)
- Browsing Routes locally and within the network (**Broadcast Search**, **Get-AdsRoute**)
- Getting remote device states and information (**Get-AdsState**, **Get-TcTargetInfo**, **Get-TcVersionInfo**)
- Establishing and Closing Remote communication sessions (**New-TcSession**,**Get-TcSession**,**Close-TcSession**)
- Browsing Symbol Information (**Get-TcSymbol**, **Get-TcDataType**)
- Reading/Writing raw and symbolic values (**Read-TcValue**, **Write-TcValue**, **Send-TcReadWrite**)
- Uploading/Downloading files to/from remote devices (**Copy-AdsFile**)
- Browsing License information (**Get-TcLicense**)

This Module is usable under all Powershell Version \>= 5.1 including

'Windows Powershell' and 'Powershell Core' Versions.

As Prerequisite the **TcXaeMgmt** Module needs a local TwinCAT installation

larger equals than TwinCAT 4024.10. There are no limitations to access

other/older TwinCAT Versions remotely.

PREREQUISITES

\>= TwinCAT 3.1.4024.10 (XAR Runtime or Full) (local installation)

POWERSHELL COMPATIBILITY

\>= Windows Powershell 5.1

\>= Powershell (Core) 6.0

CMDLETS

To see what cmdlets are provided by the TcXaeMgmt Module, execute the command:

```
PS> Get-Command -Module TcXaeMgmt -CommandType Cmdlet
```

The actual TcXaeMgmt cmdlets are listed below:

Add-AdsRoute [▶ 68]

Cmdlet for adding TwinCAT Routes.

Add-MqttRoute [▶ 80]

Adds an MQTT route to the destination system.

Close-TcSession [▶ 84]

Closes the specified session object.

Copy-AdsFile [▶ 86]

Uploads / Downloads files from/to TwinCAT target.

Get-AdsRoute [▶ 89]

List routes on a TwinCAT System / Broadcast search.

Get-AdsState [▶ 94]

Gets the Ads State of a TwinCAT Target.

Get-AmsRouterEndpoint [▶ 98]

Get the actual AmsConfiguration / RouterEndpoint of the process.

Get-EcBoxes [▶ 99]

Gets the EtherCAT Boxes actually loaded ton the target system.

Get-EcFrameStatistics [▶ 101]

Gets the EtherCAT Frame statistics from an ETherCAT master.

Get-EcMaster [▶ 103]

Gets the Ads State of a TwinCAT Target.

Read-TcValue [▶ 152]

Reads values from TwinCAT devices.

Register-AdsHandle [▶ 159]

Registers and returns a symbol handle.

Register-AdsNatRoute [▶ 162]

Changes an standard Route to an AmsNAT route on the target system (obsolete).

Remove-AdsRoute [▶ 165]

Remove an ADS Route.

Remove-MqttRoute [▶ 169]

Remove a MQQT Route.

Reset-IoFreeRun [▶ 171]

Resets the IO FreeRun state on the specified target.

Restart-AdsComputer [▶ 174]

Restarts ("reboots") the operating system on local and remote TwinCAT computers.

Send-TcReadWrite [▶ 179]

Sends a Read/Write access to ADS Server / TwinCAT Devices.

Set-AdsState [▶ 186]

Set the ADS State of a TwinCAT Target.

Set-AmsRouterEndpoint [▶ 192]

Sets the AmsConfiguration (Loopback address and port, RouterEndpoint).

Set-IoFreeRun [▶ 194]

Sets the IO FreeRun state of the target.

Stop-AdsComputer [▶ 197]

Stops (shuts down) local and remote TwinCAT computers.

Test-AdsRoute [▶ 201]

Test the specified route connection.

Unregister-AdsHandle [▶ 205]

Unregisters a symbol handle.

Write-TcValue [▶ 209]

Write values to TwinCAT devices.

EXAMPLES

Getting Route

```
PS> $route = get-adsroute TC3TEST*
PS> $route

Name            NetId             Address       Sub Version RTSystem
----            -----             -------       --- ------- --------
TC3TESTA1-CP67X 172.17.62.105.1.1 172.17.62.105     0.0     Unknown
```

Create Session

```
PS> $session = New-TcSession -Route $route -Port 851
PS> $session

ID Address               IsConnected EstablishedAt
-- -------               ----------- -------------
1  172.17.62.105.1.1:851 True        12/12/2016 12:22:02 PM
```

Read Ads Value (Struct)

```
PS> $v1 = Read-TcValue -SessionId 1 -Path "GVL.vgStruct"
PS> $v1

vBool   : True
vByte   : 123
vWord   : 12345
vDWord  : 12345678
vSInt   : -121
vUSInt  : 212
vInt    : -12121
vUInt   : 21212
vDInt   : -1212121
vUDInt  : 2121212
vReal   : 123,456
vLReal  : 1234567890,12346
vString : QWERTZUIOPÜASDFGHJKLÖÄYXCVBNM;:_
vTime   : 01:02:03.0040000
vTod    : 23:45:06.7890000
vDate   : 17.11.2005 00:00:00
vDT     : 17.11.2005 12:34:56
vAlias  : 8
vEnum   : 8
vRange  : 7
PSValue : ...
```

Read Ads Value (Boolean)

```
PS> $v2 = Read-TcValue -SessionId 1 -Path "Main.bChange"
PS> $v2
False
```

Read Ads Value (Array of Strings)

```
PS> $v3 = Read-TcValue -SessionId 1 -path "GVL.vgaString"

Dimensions                    Elements
PSValue
----------                    --------
                                                                 -------
{TwinCAT.TypeSystem.Dimension} {QWERTZUIOPÜASDFGHJKLÖÄYXCVBNM;:_, _:;MNBVCXYÄÖLKJHGFDSAÜPOIUZTREWQ}
...
```

Read Array Of Structs

```
PS> $v4 = Read-TcValue -SessionId 1 -path "GVL.vgastruct"


Dimensions                    Elements
----------                    --------
```

```
{TwinCAT.TypeSystem.Dimension} {@{vBool=True; vByte=123; vWord=12345; vDWord=12345678; vSInt=-121; v
USInt=212; vInt=-12121; vUInt=21212; vDInt=-1212121; vUD...
```

Dump Array Elements

```
PS> $v4.Dimensions.ElementCount
2

PS> $v4.Elements

vBool   : True
vByte   : 123
vWord   : 12345
vDWord  : 12345678
vSInt   : -121
vUSInt  : 212
vInt    : -12121
vUInt   : 21212
vDInt   : -1212121
vUDInt  : 2121212
vReal   : 123,456
vLReal  : 1234567890,12346
vString : QWERTZUIOPÜASDFGHJKLÖÄYXCVBNM;:_
vTime   : 01:02:03.0040000
vTod    : 23:45:06.7890000
vDate   : 17.11.2005 00:00:00
vDT     : 17.11.2005 12:34:56
vAlias  : 8
vEnum   : 8
vRange  : 7
PSValue : ...

vBool   : False
vByte   : 234
vWord   : 23456
vDWord  : 23456789
vSInt   : 121
vUSInt  : 131
vInt    : 12121
vUInt   : 13131
vDInt   : 1212121
vUDInt  : 1313131
vReal   : 456,321
vLReal  : 987654321,123457
vString : _:;MNBVCXYÄÖLKJHGFDSAÜPOIUZTREWQ
vTime   : 11:22:33.0440000
vTod    : 11:22:33.4440000
vDate   : 22.01.1999 00:00:00
vDT     : 22.01.1999 11:22:33
vAlias  : 9
vEnum   : 9
vRange  : -5
PSValue : ...

## Browse Data Types (Query by Category)
PS> $session | Get-TcDataType | where Category -eq "Array" }

Name                      Size     Category   Comment          ElementType      Dimensions       Mem
bers
----                      ----     --------   -------          -----------      ----------
       -------
ARRAY [-1..1] OF INT      6        Array                       INT              {TwinCAT.Type...
ARRAY [-10..-8] OF BOOL   3        Array                       BOOL             {TwinCAT.Type...
ARRAY [0..1] OF A_Alias   4        Array                       A_Alias          {TwinCAT.Type...
```

Browse DataTypes by name

```
PS> $session | Get-TcDataType -name "Array*"
```

Browse all Symbols recursively

```
PS> $session | Get-TcSymbol -recurse
... returns all symbols
```

Browse Symbols recursivly by Symbol Path (Here specific array index 'TaskInfo[1]'(

```
PS> $session | Get-TcSymbol -recurse -path "*TaskInfo``[1``]*","*.ProjectName"

InstanceName            DataType              Size InstancePath
             Comment
------------            --------              ---- ------------
                                                   -------
ProjectName             STRING(63)            64   TwinCAT_SystemInfoVarList._AppInfo.ProjectName
_TaskInfo[1]            PLC.PlcTaskSystemInfo 128  TwinCAT_SystemInfoVarList._TaskInfo[1]
ObjId                   OTCID                 4    TwinCAT_SystemInfoVarList._TaskInfo[1].ObjId
CycleTime               UDINT                 4    TwinCAT_SystemInfoVarList._TaskInfo[1].CycleTime
Priority                UINT                  2    TwinCAT_SystemInfoVarList._TaskInfo[1].Priority
AdsPort                 UINT                  2    TwinCAT_SystemInfoVarList._TaskInfo[1].AdsPort
CycleCount              UDINT                 4    TwinCAT_SystemInfoVarList._TaskInfo[1].CycleCount
DcTaskTime              LINT                  8    TwinCAT_SystemInfoVarList._TaskInfo[1].DcTaskTime
LastExecTime            UDINT                 4    TwinCAT_SystemInfoVarList._TaskInfo[1].LastExecTi
me
FirstCycle              BOOL                  1    TwinCAT_SystemInfoVarList._TaskInfo[1].FirstCycle
CycleTimeExceeded       BOOL                  1    TwinCAT_SystemInfoVarList._TaskInfo[1].CycleTimeE
xceeded
InCallAfterOutputUpdate BOOL                  1    TwinCAT_SystemInfoVarList._TaskInfo[1].InCallAfte
rOutputUpdate
RTViolation             BOOL                  1    TwinCAT_SystemInfoVarList._TaskInfo[1].RTViolatio
n
TaskName                STRING(63)            64   TwinCAT_SystemInfoVarList._TaskInfo[1].TaskName
```

Browse only Symbols ending with path *.ProjectName

```
PS>$project = Get-TcSymbol -Session $session -recurse -path "*.ProjectName"

InstanceName DataType   Size InstancePath                                  Comment
------------ --------   ---- ------------                                  -------
ProjectName  STRING(63) 64   TwinCAT_SystemInfoVarList._AppInfo.ProjectName
```

Ads Read ProjectName

```
PS>$project | Read-TcValue -Session $session
ADS_DynSymbols
```

Ads Write ProjectName

```
PS>$project | Write-TcValue -Session $session -Value "NewProjectName"
PS>$project | Read-TcValue -Session $session
NewProjectName
```

ReadWrite by Symbol Path

```
PS>Read-TcValue -SessionId 1 -Path "Main.bChange"
false
PS>Write-TcValue -SessionId 1 -Symbol "Main.bChange" -Value True
PS>Read-TcValue -SessionId 1 -Path "GVL.vgBool"
PS>Write-TcValue -SessionId 1 -Path "GVL.vgBool" -value $true
```

ReadWrite by Piping

```
PS> $projectNameSymbol = $session | Get-TcSymbol -Recurse -path "*ProjectName"
PS> $projectNameSymbol | Read-TcValue -SessionId 1
PS> $projectNameSymbol | Write-TcValue -SessionId 1 -Value "NewProjectName"
PS> $projectNameSymbol | Read-TcValue -SessionId 1
```

Get Target Information

```
PS> get-adsroute | Get-TcTargetInfo

Target          Version      Level OS    Image Device CPUArch
------          -------      ----- --    ----- ------ -------
TC3TESTA1-CP67X 3.1.4021.131 CP    Win7              IntelX86
```

```
PS> get-adsroute | Get-TcVersion

Major  Minor  Build  Revision
```

```
-----  -----  -----  --------
3      1      4021   131
```

PROVIDERS

The TcXaeMgmt module includes the AdsSymbolProvider and the AdsFileProvider

AdsSymbolProvider

Binds the target device symbolic information to a PSDrive. To register a symbol server as

PSDrive type (here the Target Route 'CX_01234' with AmsPort: 851)

```
PS> New-PSDrive -Name CX_01234_Symbols -PSProvider AdsSymbolProvider -Address CX_01234 -Port 851 -
Root
PS> cd CX_01234_Symbols:
PS> CX_01234_Symbols:> dir
```

AdsFileProvider

```
PS> New-PSDrive -name CX_01234 -PSProvider AdsFileProvider -Address CX_01234 -Root ''
PS> dir CX_01234:

Mode              LastWriteTime                    Length     Name
----              -------------                    ------     ----
d----             30.11.2021 16:11:31                         BootDir
d----             03.12.2021 01:17:20                         BootProject
d----             17.03.2021 14:33:53                         ConfigDir
d----             03.12.2021 01:17:20                         Generic
d----             18.06.2021 08:00:22                         InstallDir
d----             03.12.2021 01:17:20                         RepositoryDir
d----             03.12.2021 15:32:03                         TargetDir

> cd CX_01234:/BootDir

PS CX_01234:\BootDir> dir

Mode              LastWriteTime                    Length     Name
----              -------------                    ------     ----
d----             05.10.2021 10:36:34                         CurrentConfig
-a---             05.10.2021 10:36:34              4563       CurrentConfig.tszip
-a---             05.10.2021 10:36:34              17113      CurrentConfig.xml
-a---             30.11.2021 16:11:31              126976     LoggedEvents.db
d----             27.10.2021 11:32:43                         Plc
```

More Information about Providers

```
PS> get-help about_providers
```

Example: Create a new AdsFileProvider Drive to the TwinCAT Device CX_01234

```
> New-PSDrive -name CX_01234 -PSProvider AdsFileProvider -Address CX_01234 -Root ''

Name           Used (GB)     Free (GB) Provider        Root                 CurrentLocation
----           ---------     --------- --------        ----                 ---------------
CX_01234                               AdsFileProvider \TargetDir
```

Example: Browse the files on the TwinCAT Device CX_01234

```
> dir

Mode              LastWriteTime                    Length     Name
----              -------------                    ------     ----
d----             26.11.2021 17:44:27                         CACerts
-a---             14.03.2012 14:50:50              619        DefaultConfig.xml
d----             11.05.2021 14:42:45                         License
d----             18.06.2021 08:01:03                         Resource
d----             17.03.2021 15:15:51                         Routes
d----             18.06.2021 08:00:33                         StartMenuAdmin
d----             17.03.2021 14:33:35                         StartUp
-a---             30.11.2021 18:46:08              2253       StaticRoutes.xml
```

```
-a---            01.02.2012 16:42:58            494        TargetFeatures.xml
-a---            17.03.2021 14:42:50            3113       TcSelfSigned.xml
```

Example: Read the content of the StaticRouts.xml on target CX_01234

```
r> get-content .\StaticRoutes.xml
<?xml version="1.0"?>
<TcConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <RemoteConnections>
                <Route>
                        <Name>TargetIPC</Name>
                        <Address>172.17.60.147</Address>
                        <NetId>172.17.60.147.1.1</NetId>
                        <Type>TCP_IP</Type>
                        <Tls IgnoreCn="true">
                                <Ca>...</Ca>
                        </Tls>
                </Route>
                <Server>
                        <Tls IgnoreCn="true">
                                <Ca>c:\twincat\3.1\target\CACerts\RootCA.pem</Ca>
                                <Cert>c:\twincat\3.1\target\CACerts\TargetIPC.crt</Cert>
                                <Key>c:\twincat\3.1\target\CACerts\TargetIPC.key</Key>
                        </Tls>
                </Server>
        </RemoteConnections>
</TcConfig>
```

FEEDBACK

Please submit any feedback, including defects and enhancement requests,

to

support@beckhoff.com

We are also interested in suggestions you may have for cmdlets. Over

time, we hope to be able to add some more features.

NOTE

To see what functions are provided by TcXaeMgmt, execute the command:

```
PS> Get-Command -Module TcXaeMgmt -CommandType Function
```

For more information, most of the cmdlets have help associated with

them e.g.:

```
PS> Get-Help Add-AdsRoute -full
```

The definitive information on a cmdlet's parameters can be obtained

by executing:

```
PS> Get-Command Add-AdsRoute -syntax
```

or more tersely:

```
PS> gcm Add-AdsRoute -syn
```

SEE ALSO

Documentation TcXaeMgmt Module

About the TcXaeMgmt Module

Beckhoff Homepage

PS> get-help about_providers

KEYWORDS

- ADS
- TwinCAT
- ManagementConsole
- Routes

6.4 Get-AmsNetId

SYNOPSIS

Get the local NetId of the TwinCAT System.

SYNTAX

```
Get-AmsNetId [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

DESCRIPTION

This Cmdlet returns the AmsNetId of the local TwinCAT System.

For getting more information about the local system the Cmdlets 'Get-AdsRoute -local' or 'Get-TcTargetInfo' can be used.

EXAMPLES

EXAMPLE 1

```
PS> Get-AmsNetId

            192.168.0.2.1.1
```

Gets the AmsNetId of the Local system.

PARAMETERS

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

OUTPUTS

NOTES

6.5 Restart-TwinCAT

SYNOPSIS

Restarts or Resets a specified TwinCAT System.

SYNTAX

NetId (Default)

```
Restart-TwinCAT [[-NetId] <AmsNetId[]>] [-Quiet] [-Force] [-StateOnly] [-Timeout <Int32>] [-
NoReinit] [-NoWait]
 [-WaitTimeout <Int32>] [-PollingRate <Int32>] [-Command <AdsStateCommand>] [-ThrowError]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

Route

```
Restart-TwinCAT [-InputObject] <IRoute[]> [-Quiet] [-Force] [-StateOnly] [-Timeout <Int32>] [-
NoReinit]
 [-NoWait] [-WaitTimeout <Int32>] [-PollingRate <Int32>] [-Command <AdsStateCommand>] [-ThrowError]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

AddressStr

```
Restart-TwinCAT [-Address] <String[]> [-Quiet] [-Force] [-StateOnly] [-Timeout <Int32>] [-
NoReinit] [-NoWait]
 [-WaitTimeout <Int32>] [-PollingRate <Int32>] [-Command <AdsStateCommand>] [-ThrowError]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

Session

```
Restart-TwinCAT -Session <ISession[]> [-Quiet] [-Force] [-StateOnly] [-Timeout <Int32>] [-
NoReinit] [-NoWait]
 [-WaitTimeout <Int32>] [-PollingRate <Int32>] [-Command <AdsStateCommand>] [-ThrowError]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

SessionId

```
Restart-TwinCAT -SessionId <Int32[]> [-Quiet] [-Force] [-StateOnly] [-Timeout <Int32>] [-
NoReinit] [-NoWait]
 [-WaitTimeout <Int32>] [-PollingRate <Int32>] [-Command <AdsStateCommand>] [-ThrowError]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

DESCRIPTION

This Cmdlet Restarts or Resets the specified TwinCAT System dependant of its command parameter.

The TwinCAT system will end up in ADS State 'Run' or 'Config'.

EXAMPLES

EXAMPLE 1

```
PS> Restart-TwinCat -command Reset -force

WARNING: 192.168.0.2.1.1:10 ERR |
 18:26:28:108 |'TCOM Server' (10): Device 1 (EtherCAT) (Adapter): Failed to connect to network adapt
er!
WARNING: 192.168.0.2.1.1:10 WRN |
```

```
 18:26:28:108 |'TCOM Server' (10): PREOP to SAFEOP of 'Device 1 (EtherCAT) (Adapter)' (0x03010011) f
ailed - 'request is aborted' 0x9811071F
WARNING: 192.168.0.2.1.1:10000 ERR |
 18:26:28:117 |'TwinCAT System' (10000): Sending ams command >> Init12\IO: Set State TComObj SAFEOP:
 Set Objects (4) to SAFEOP >> AdsError: 1823 (0x71f, ADS ERROR: device aborted the action) << failed
!

Ok Target     NetId             Port  ErrorCode Requested Original Reached Latency (ms)
-- ------     -----             ----  --------- --------- -------- ------- ------------
   CX_1111    192.168.0.2.1.1  10000  Succeeded Reset     Config   Config  3427
```

Calls a Reset to the local SystemService that fails!.

Error log messages will be logged out.

EXAMPLE 2

```
PS> Restart-TwinCAT -command Reset -force | select-object -ExpandProperty LogMessages

WARNING: 192.168.0.2.1.1:10 ERR |
 18:20:45:969 |'TCOM Server' (10): Device 1 (EtherCAT) (Adapter): Failed to connect to network adapt
er!
WARNING: 192.168.0.2.1.1:10 WRN |
 18:20:45:969 |'TCOM Server' (10): PREOP to SAFEOP of 'Device 1 (EtherCAT) (Adapter)' (0x03010011) f
ailed - 'request is aborted' 0x9811071F
WARNING: 192.168.0.2.1.1:10000 ERR |
 18:20:45:979 |'TwinCAT System' (10000): Sending ams command >%gt; Init12\IO: Set State TComObj SAFE
OP: Set Objects (4) to SAFEOP >%gt; AdsError: 1823 (0x71f, ADS ERROR: device aborted the action) <<
failed!

Type    TimeStamp    DeviceName     Port  Message
----    ---------    ----------     ----  -------
Message 18:20:44.874 TwinCAT System 10000 TwinCAT System Restart initiated from AmsNetId: 192.168.0.
2.1.1 port 34564.
Message 18:20:44.879 TwinCAT System 10000 Saving configuration of COM server TcVnService !
Message 18:20:44.880 TwinCAT System 10000 Saving configuration of COM server TcEventLogger !
Message 18:20:44.970 TwinCAT System 10000 Shutting down COM Server TcVnService !
Message 18:20:44.971 TwinCAT System 10000 Shutting down COM Server TcEventLogger !
Message 18:20:45.745 TwinCAT System 10000 Loading configuration of COM server TcVnService !
Message 18:20:45.747 TwinCAT System 10000 Loading configuration of COM server TcEventLogger !
Message 18:20:45.748 TwinCAT System 10000 Initializing COM Server TcVnService !
Message 18:20:45.773 TwinCAT System 10000 Initializing COM Server TcEventLogger !
Message 18:20:45.783 TwinCAT System 10000 TcIoEth Server started: TcIoEth.
Message 18:20:45.791 TwinCAT System 10000 TcRtsObjects Server started: TcRtsObjects.
Message 18:20:45.798 TwinCAT System 10000 TcIoECat Server started: TcIoECat.
Message 18:20:45.805 TwinCAT System 10000 TcIo Server started: TcIo.
Message 18:20:45.814 TwinCAT System 10000 TcPlc30 Server started: TcPlc30.
Message 18:20:45.821 TwinCAT System 10000 TcRTime Server started: TcRTime.
Message 18:20:45.927 License Server 30    license validation status is Valid(3)
Error   18:20:45.969 TCOM Server    10    Device 1 (EtherCAT) (Adapter): Failed to connect to networ
k adapter!
Warning 18:20:45.969 TCOM Server    10    PREOP to SAFEOP of 'Device 1 (EtherCAT) (Adapter)' (0x0301
0011) failed - 'request is aborted' 0x9811071F
Error   18:20:45.979 TwinCAT System 10000 Sending ams command >%gt; Init12\IO: Set State TComObj SAF
EOP: Set Objects (4) to SAFEOP >%gt; AdsError: 1823 (0x71f, ADS ERROR: device aborted the action) %l
t;%lt; failed!
Message 18:20:47.879 TwinCAT System 10000 Loading configuration of COM server TcVnService
Message 18:20:47.881 TwinCAT System 10000 Loading configuration of COM server TcEventLogger
Message 18:20:47.882 TwinCAT System 10000 Initializing COM Server TcVnService
Message 18:20:47.910 TwinCAT System 10000 Initializing COM Server TcEventLogger
Message 18:20:47.922 TwinCAT System 10000 TCIO Server started: TCIO.
Message 18:20:47.931 TwinCAT System 10000 TCRTIME Server started: TCRTIME.
Message 18:20:47.939 TwinCAT System 10000 TCRTSOBJECTS Server started: TCRTSOBJECTS.
Message 18:20:47.948 TwinCAT System 10000 TCIOETH Server started: TCIOETH.
Message 18:20:47.956 TwinCAT System 10000 TCIOECAT Server started: TCIOECAT.
Message 18:20:47.963 TwinCAT System 10000 TCIODRIVERS Server started: TCIODRIVERS.
Message 18:20:48.078 TwinCAT System 10000 Starting COM Server TcVnService
Message 18:20:48.078 TwinCAT System 10000 Starting COM Server TcEventLogger
```

Tries to restart the local TwinCAT system and write the log messages to the output.

This Command fails.

PARAMETERS

-NetId

The NetId address where to set the state (Local system by default).

Multiple values are allowed.

```
Type: AmsNetId[]
Parameter Sets: NetId
Aliases:

Required: False
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-InputObject

Target route(s), where to set the state.

Multiple values are allowed.

```
Type: IRoute[]
Parameter Sets: Route
Aliases: Destination, Route

Required: True
Position: 1
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-Address

The address of the system where to set the state.

This can be the RouteName, NetId, the HostName or the IPAddress.

Wildcards and multiple values are permitted.

```
Type: String[]
Parameter Sets: AddressStr
Aliases: Name

Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-Session

The ADS Session to use for the Cmdlet.

Multiple sessions are allowed.

```
Type: ISession[]
Parameter Sets: Session
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-SessionId

Specifies the Session (with unique ID) to use for the Cmdlet (multiple values are allowed)

```
Type: Int32[]
Parameter Sets: SessionId
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Quiet

Sets the Quiet mode of the command.

The Cmdlet then returns a $true or $false but not the actual states of the targets.

The return value will be $true if all operations succeed and it will be $false if at least one have failed.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Force

Forces the command (no confirmation, Resets the FailFastHandler)

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-StateOnly

This Cmdlet return only the AdsState instead of full information.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Timeout

The communication ADS timeout in milliseconds.

A value \<= 0 sets the Default (5000 ms).

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-NoReinit

Activates a state check before sending WriteControl if the target system is already in the expected target state

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-NoWait

The -NoWait parameter skips the waiting for the target end state.

If set, the Cmdlet returns immediatly after sending the WriteControl request, without waiting for the state change.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-WaitTimeout

The wait timout for the state change in ms.

This Cmdlet waits for the target state changes which is limited by this WaitTimeout.

A value \<= 0 sets the Default (45000 ms).

This parameter is only used if -NoWait is not set.

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-PollingRate

The Wait polling rate in Milliseconds.

A value \<= 0 sets the Default polling rate (200 ms for local systems, 1000ms for remote systems).

This parameter is only used, if -NoWait is not set.

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-Command

The timeout to wait for restart.

A value of 0 disables the timeout.

A value \<= 0 sets the Default (5000 ms).

Possible values: None, Reset, Start, Stop, Reconfig, Run

```
Type: AdsStateCommand
Parameter Sets: (All)
Aliases:
Accepted values: None, Reset, Start, Stop, Reconfig, Run

Required: False
Position: Named
Default value: Reset
Accept pipeline input: False
Accept wildcard characters: False
```

-ThrowError

Throws an error, if the target system(s) not reaching the expected state.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Confirm

Prompts you for confirmation before running the cmdlet.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: cf

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-WhatIf

Shows what would happen if the cmdlet runs.

The cmdlet is not run.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: wi
```

```
Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.IRoute[]

Target route(s), where to set the state.

Multiple values are allowed.

TwinCAT.ISession[]

The ADS Session to use for the Cmdlet.

Multiple sessions are allowed.

OUTPUTS

NOTES

6.6 Set-AmsNetId

SYNOPSIS

Sets the AmsNetId of a device.

SYNTAX

NetId (Default)

```
Set-AmsNetId [[-Target] <AmsNetId>] [-NewId] <AmsNetId> [-Timeout <Int32>] [-Force]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

Route

```
Set-AmsNetId [-NewId] <AmsNetId> [-InputObject] <IRoute> [-Timeout <Int32>] [-Force]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

AddressStr

```
Set-AmsNetId [-NewId] <AmsNetId> [-Address] <String> [-Timeout <Int32>] [-Force]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

Session

```
Set-AmsNetId [-NewId] <AmsNetId> -Session <ISession> [-Timeout <Int32>] [-Force]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

SessionId

```
Set-AmsNetId [-NewId] <AmsNetId> -SessionId <Int32> [-Timeout <Int32>] [-Force]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

DESCRIPTION

This Cmdlet configurates the AmsNetId (TargetNetId) of a device.

A reboot is necessary after configuration (e.g.

with 'Restart-AdsComputer') To contact the target system, it must be available as actual route.

All actual connections to that systems via ADS are not valid anymore after calling this Cmdlet.

EXAMPLES

EXAMPLE 1

```
PS> Set-AmsNetId -NewId 1.1.1.1.1.1

Changeing AmsNetId of target system.
Change the NetId of system '192.168.0.2.1.1' to '1.1.1.1.1.1'
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "Y"): y
Changing the NetId of system '192.168.0.2.1.1' to '1.1.1.1.1.1' is succeeded. All preexisting connec
tions to this system a invalid now. A reboot of this system is necessary!
```

Sets the AmsNetId of the Local system to '1.1.1.1.1.1'.

PARAMETERS

-Target

NetId of the target system.

```
Type: AmsNetId
Parameter Sets: NetId
Aliases:

Required: False
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-NewId

The New NetId.

```
Type: AmsNetId
Parameter Sets: (All)
Aliases:

Required: True
Position: 2
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-InputObject

The route object where to set the CPUs.

This parameter support pipelining.

```
Type: IRoute
Parameter Sets: Route
Aliases: Destination, Route

Required: True
```

```
Position: 1
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-Address

Target names/addresses where to configure the CPU.

Wildcards are permitted.

```
Type: String
Parameter Sets: AddressStr
Aliases: Name

Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-Session

The Session where to configure the CPU (supports pipeline)

```
Type: ISession
Parameter Sets: Session
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-SessionId

Specifies the Session (with unique ID) where to configure the CPU Core settings.

```
Type: Int32
Parameter Sets: SessionId
Aliases:

Required: True
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-Timeout

The communication ADS timeout in milliseconds.

A value of 0 disables the timeout.

A value \<= 0 sets the Default (5000 ms).

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-Force

Forces this command.

It suppresses the ShouldContinue settings and bypasses the FailFastInterceptor to retry communication in every case.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Confirm

Prompts you for confirmation before running the cmdlet.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: cf

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-WhatIf

Shows what would happen if the cmdlet runs.

The cmdlet is not run.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: wi

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.IRoute

The route object where to set the CPUs.

This parameter support pipelining.

TwinCAT.ISession

The Session where to configure the CPU (supports pipeline)

OUTPUTS

NOTES

6.7 Set-RTimeCpuSettings

SYNOPSIS

Sets the Windows (Shared) CPU cores and Isolated cores for TwinCAT.

SYNTAX

NetIdPortShared (Default)

```
Set-RTimeCpuSettings [[-NetId] <AmsNetId>] -SharedCores <Int32> [-Timeout <Int32>] [-Force]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

NetIdPortIsolated

```
Set-RTimeCpuSettings [[-NetId] <AmsNetId>] -IsolatedCores <Int32> [-Timeout <Int32>] [-Force]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

NetIdPortReset

```
Set-RTimeCpuSettings [[-NetId] <AmsNetId>] [-Reset] [-Timeout <Int32>] [-Force]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

RouteShared

```
Set-RTimeCpuSettings [-InputObject] <IRoute> -SharedCores <Int32> [-Timeout <Int32>] [-Force]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

RouteIsolated

```
Set-RTimeCpuSettings [-InputObject] <IRoute> -IsolatedCores <Int32> [-Timeout <Int32>] [-Force]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

RouteReset

```
Set-RTimeCpuSettings [-InputObject] <IRoute> [-Reset] [-Timeout <Int32>] [-Force]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

AddressStrShared

```
Set-RTimeCpuSettings [-Address] <String> -SharedCores <Int32> [-Timeout <Int32>] [-Force]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

AddressStrIsolated

```
Set-RTimeCpuSettings [-Address] <String> -IsolatedCores <Int32> [-Timeout <Int32>] [-Force]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

AddressStrReset

```
Set-RTimeCpuSettings [-Address] <String> [-Reset] [-Timeout <Int32>] [-Force]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

SessionShared

```
Set-RTimeCpuSettings -Session <ISession> -SharedCores <Int32> [-Timeout <Int32>] [-Force]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

SessionIsolated

```
Set-RTimeCpuSettings -Session <ISession> -IsolatedCores <Int32> [-Timeout <Int32>] [-Force]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

SessionReset

```
Set-RTimeCpuSettings -Session <ISession> [-Reset] [-Timeout <Int32>] [-Force]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

SessionIdShared

```
Set-RTimeCpuSettings -SessionId <Int32> -SharedCores <Int32> [-Timeout <Int32>] [-Force]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

SessionIdIsolated

```
Set-RTimeCpuSettings -SessionId <Int32> -IsolatedCores <Int32> [-Timeout <Int32>] [-Force]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

SessionIdReset

```
Set-RTimeCpuSettings -SessionId <Int32> [-Reset] [-Timeout <Int32>] [-Force]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

DESCRIPTION

This Cmdlet configurates the CPU settings.

A reboot is necessary after configuration.

EXAMPLES

EXAMPLE 1

```
PS> Set-RTimeCpuSettings -SharedCores 6

Setting CPU cores
Setting WindowsCores: 6, IsolatedCores: 6 to device '172.17.62.146.1.1'?
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "Y"): y
Number of processors successfully set to '6'. A reboot is necessary to activate settings!
```

Sets the CPU Core Settings to 6 Shared and 6 Isolated on a 12 Core System

PARAMETERS

-NetId

NetId of the target system.

```
Type: AmsNetId
Parameter Sets: NetIdPortShared, NetIdPortIsolated, NetIdPortReset
Aliases:

Required: False
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-InputObject

The route object where to set the CPUs.

This parameter support pipelining.

```
Type: IRoute
Parameter Sets: RouteShared, RouteIsolated, RouteReset
Aliases: Destination, Route

Required: True
Position: 1
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-Address

Target names/addresses where to configure the CPU.

Wildcards are permitted.

```
Type: String
Parameter Sets: AddressStrShared, AddressStrIsolated, AddressStrReset
Aliases: Name

Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-Session

The Session where to configure the CPU (supports pipeline)

```
Type: ISession
Parameter Sets: SessionShared, SessionIsolated, SessionReset
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-SessionId

Specifies the Session (with unique ID) where to configure the CPU Core settings.

```
Type: Int32
Parameter Sets: SessionIdShared, SessionIdIsolated, SessionIdReset
Aliases:

Required: True
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-SharedCores

The number of shared windows cores (Isolated cores will be: isolatedCores = allCores - sharedCores.

```
Type: Int32
Parameter Sets: NetIdPortShared, RouteShared, AddressStrShared, SessionShared, SessionIdShared
Aliases:

Required: True
Position: Named
Default value: 0
```

```
Accept pipeline input: False
Accept wildcard characters: False
```

-IsolatedCores

Number of Isolated cores.

Shared cores will be: sharedCores = allCores - isolatedCores.

```
Type: Int32
Parameter Sets: NetIdPortIsolated, RouteIsolated, AddressStrIsolated, SessionIsolated, SessionIdIsol
ated
Aliases:

Required: True
Position: Named
Default value: 0
Accept pipeline input: False
Accept wildcard characters: False
```

-Reset

Resets the number of shared cores.

After reboot all Cores are dedicated to windows (shared).

```
Type: SwitchParameter
Parameter Sets: NetIdPortReset, RouteReset, AddressStrReset, SessionReset, SessionIdReset
Aliases:

Required: True
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Timeout

The communication ADS timeout in milliseconds.

A value of 0 disables the timeout.

A value \<= 0 sets the Default (5000 ms).

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-Force

Forces this command.

It suppresses the ShouldContinue settings and bypasses the FailFastInterceptor to retry communication in every case.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Confirm

Prompts you for confirmation before running the cmdlet.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: cf

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-WhatIf

Shows what would happen if the cmdlet runs.

The cmdlet is not run.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: wi

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.IRoute

The route object where to set the CPUs.

This parameter support pipelining.

TwinCAT.ISession

The Session where to configure the CPU (supports pipeline)

OUTPUTS

NOTES

6.8 Start-AdsProcess

SYNOPSIS

Start a process via ADS on the target system.

SYNTAX

NetIdPort (Default)

```
Start-AdsProcess [[-NetId] <AmsNetId[]>] -FilePath <String> [-ArgumentList <String[]>] [-
WorkingDir <String>]
 [-Timeout <Int32>] [-Force] [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

Route

```
Start-AdsProcess [-InputObject] <IRoute[]> -FilePath <String> [-ArgumentList <String[]>] [-
WorkingDir <String>]
 [-Timeout <Int32>] [-Force] [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

AddressStr

```
Start-AdsProcess [-Address] <String[]> -FilePath <String> [-ArgumentList <String[]>] [-
WorkingDir <String>]
 [-Timeout <Int32>] [-Force] [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

Session

```
Start-AdsProcess -Session <ISession[]> -FilePath <String> [-ArgumentList <String[]>] [-
WorkingDir <String>]
 [-Timeout <Int32>] [-Force] [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

SessionId

```
Start-AdsProcess -SessionId <Int32[]> -FilePath <String> [-ArgumentList <String[]>] [-
WorkingDir <String>]
 [-Timeout <Int32>] [-Force] [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

DESCRIPTION

This Cmdlet starts a processs on the Target system.

EXAMPLES

EXAMPLE 1

```
PS> Start-AdsProcess -Address CX_1234 -path "notepad.exe"
```

Starts the notepad.exe on the target system.

PARAMETERS

-NetId

NetId(s) of the target system.

```
Type: AmsNetId[]
Parameter Sets: NetIdPort
Aliases:
```

```
Required: False
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-InputObject

The route object where to get the Target information from..

```
Type: IRoute[]
Parameter Sets: Route
Aliases: Destination, Route

Required: True
Position: 1
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-Address

Target names/addresses.

These can consist of RouteName, NetID, HostName or IPAddress.

Wildcards are permitted.

```
Type: String[]
Parameter Sets: AddressStr
Aliases: Name

Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-Session

The Session to use for the value read.

```
Type: ISession[]
Parameter Sets: Session
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-SessionId

Specifies the Session (with unique ID) to use for the value read.

```
Type: Int32[]
Parameter Sets: SessionId
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-FilePath

Path to the executable of the target system.

```
Type: String
Parameter Sets: (All)
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-ArgumentList

Specifies parameters or parameter values to use when this cmdlet starts the process.

Arguments can be accepted as a single string with the arguments separated by spaces, or as an array of strings separated by commas.

The cmdlet joins the array into a single string with each element of the array separated by a single space.

```
Type: String[]
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-WorkingDir

The working directory on the target system

```
Type: String
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Timeout

The communication ADS timeout in milliseconds.

A value of 0 disables the timeout.

A value \<= 0 sets the Default (5000 ms).

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-Force

Force reading value.

This flag bypasses the FailFastInterceptor to retry communication in every case.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
```

```
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.IRoute[]

The route object where to get the Target information from..

TwinCAT.ISession[]

The Session to use for the value read.

OUTPUTS

NOTES

6.9 About TcXaeMgmt

PowerShell TwinCAT XAE Management Console (**TcXaeMgmt**)

SHORT DESCRIPTION

Cmdlets for managing and accessing ADS Routes, Reading/Writing Values and managing Remote targets.

LONG DESCRIPTION

The Powershell TwinCAT Management Console is a PowerShell module that provides

a number of useful cmdlets for TwinCAT System Management and for communicating

with ADS devices over the ADS protocol.

This includes the following tasks/features:

- Establishing/Removing Route Connections (**Add-AdsRoute**, **Remove-AdsRoute**)
- Browsing Routes locally and within the network (**Broadcast Search**, **Get-AdsRoute**)
- Getting remote device states and information (**Get-AdsState**, **Get-TcTargetInfo**, **Get-TcVersionInfo**)
- Establishing and Closing Remote communication sessions (**New-TcSession**, **Get-TcSession**, **Close-TcSession**)
- Browsing Symbol Information (**Get-TcSymbol**, **Get-TcDataType**)

- Reading/Writing raw and symbolic values (**Read-TcValue**, **Write-TcValue**, **Send-TcReadWrite**)
- Uploading/Downloading files to/from remote devices (**Copy-AdsFile**)
- Browsing License information (**Get-TcLicense**)

This Module is usable under all Powershell Version >= 5.1 including

'Windows Powershell' and 'Powershell Core' Versions.

As Prerequisite the **TcXaeMgmt** Module needs a local TwinCAT installation

larger equals than TwinCAT 4024.10. There are no limitations to access

other/older TwinCAT Versions remotely.

PREREQUISITES

>= TwinCAT 3.1.4024.10 (XAR Runtime or Full) (local installation)

POWERSHELL COMPATIBILITY

>= Windows Powershell 5.1

>= Powershell (Core) 6.0

CMDLETS

To see what cmdlets are provided by the TcXaeMgmt Module, execute the command:

```
PS> Get-Command -Module TcXaeMgmt -CommandType Cmdlet
```

The actual TcXaeMgmt cmdlets are listed below:

Add-AdsRoute [▶ 68]

Cmdlet for adding TwinCAT Routes.

Add-MqttRoute [▶ 80]

Adds an MQTT route to the destination system.

Close-TcSession [▶ 84]

Closes the specified session object.

Copy-AdsFile [▶ 86]

Uploads / Downloads files from/to TwinCAT target.

Get-AdsRoute [▶ 89]

List routes on a TwinCAT System / Broadcast search.

Get-AdsState [▶ 94]

Gets the Ads State of a TwinCAT Target.

Get-AmsRouterEndpoint [▶ 98]

Get the actual AmsConfiguration / RouterEndpoint of the process.

Get-EcBoxes [▶ 99]

Gets the EtherCAT Boxes actually loaded ton the target system.

Get-TcVersion [▶ 146]

Get the TwinCAT Version of a target system.

New-TcSession [▶ 149]

Create a new session to a TwinCAT Target.

Read-TcValue [▶ 152]

Reads values from TwinCAT devices.

Register-AdsHandle [▶ 159]

Registers and returns a symbol handle.

Register-AdsNatRoute [▶ 162]

Changes an standard Route to an AmsNAT route on the target system (obsolete).

Remove-AdsRoute [▶ 165]

Remove an ADS Route.

Remove-MqttRoute [▶ 169]

Remove a MQQT Route.

Reset-IoFreeRun [▶ 171]

Resets the IO FreeRun state on the specified target.

Restart-AdsComputer [▶ 174]

Restarts ("reboots") the operating system on local and remote TwinCAT computers.

Send-TcReadWrite [▶ 179]

Sends a Read/Write access to ADS Server / TwinCAT Devices.

Set-AdsState [▶ 186]

Set the ADS State of a TwinCAT Target.

Set-AmsRouterEndpoint [▶ 192]

Sets the AmsConfiguration (Loopback address and port, RouterEndpoint).

Set-IoFreeRun [▶ 194]

Sets the IO FreeRun state of the target.

Stop-AdsComputer [▶ 197]

Stops (shuts down) local and remote TwinCAT computers.

Test-AdsRoute [▶ 201]

Test the specified route connection.

Unregister-AdsHandle [▶ 205]

Unregisters a symbol handle.

Write-TcValue [▶ 209]

Write values to TwinCAT devices.

EXAMPLES

Getting Route

```
PS> $route = get-adsroute TC3TEST*
PS> $route

Name            NetId             Address       Sub Version RTSystem
----            -----             -------       --- ------- --------
TC3TESTA1-CP67X 172.17.62.105.1.1 172.17.62.105     0.0     Unknown
```

Create Session

```
PS> $session = New-TcSession -Route $route -Port 851
PS> $session

ID Address               IsConnected EstablishedAt
-- -------               ----------- -------------
1  172.17.62.105.1.1:851 True        12/12/2016 12:22:02 PM
```

Read Ads Value (Struct)

```
PS> $v1 = Read-TcValue -SessionId 1 -Path "GVL.vgStruct"
PS> $v1

vBool   : True
vByte   : 123
vWord   : 12345
vDWord  : 12345678
vSInt   : -121
vUSInt  : 212
vInt    : -12121
vUInt   : 21212
vDInt   : -1212121
vUDInt  : 2121212
vReal   : 123,456
vLReal  : 1234567890,12346
vString : QWERTZUIOPÜASDFGHJKLÖÄYXCVBNM;:_
vTime   : 01:02:03.0040000
vTod    : 23:45:06.7890000
vDate   : 17.11.2005 00:00:00
vDT     : 17.11.2005 12:34:56
vAlias  : 8
vEnum   : 8
vRange  : 7
PSValue : ...
```

Read Ads Value (Boolean)

```
PS> $v2 = Read-TcValue -SessionId 1 -Path "Main.bChange"
PS> $v2
False
```

Read Ads Value (Array of Strings)

```
PS> $v3 = Read-TcValue -SessionId 1 -path "GVL.vgaString"

Dimensions                    Elements
PSValue
----------                    --------
                                                                 -------
{TwinCAT.TypeSystem.Dimension} {QWERTZUIOPÜASDFGHJKLÖÄYXCVBNM;:_, _:;MNBVCXYÄÖLKJHGFDSAÜPOIUZTREWQ}
...
```

Read Array Of Structs

```
PS> $v4 = Read-TcValue -SessionId 1 -path "GVL.vgastruct"


Dimensions                    Elements
----------                    --------
{TwinCAT.TypeSystem.Dimension} {@{vBool=True; vByte=123; vWord=12345; vDWord=12345678; vSInt=-121; v
USInt=212; vInt=-12121; vUInt=21212; vDInt=-1212121; vUD...
```

Dump Array Elements

```
PS> $v4.Dimensions.ElementCount
2

PS> $v4.Elements

vBool   : True
vByte   : 123
vWord   : 12345
vDWord  : 12345678
vSInt   : -121
vUSInt  : 212
vInt    : -12121
vUInt   : 21212
vDInt   : -1212121
vUDInt  : 2121212
vReal   : 123,456
vLReal  : 1234567890,12346
vString : QWERTZUIOPÜASDFGHJKLÖÄYXCVBNM;:_
vTime   : 01:02:03.0040000
vTod    : 23:45:06.7890000
vDate   : 17.11.2005 00:00:00
vDT     : 17.11.2005 12:34:56
vAlias  : 8
vEnum   : 8
vRange  : 7
PSValue : ...

vBool   : False
vByte   : 234
vWord   : 23456
vDWord  : 23456789
vSInt   : 121
vUSInt  : 131
vInt    : 12121
vUInt   : 13131
vDInt   : 1212121
vUDInt  : 1313131
vReal   : 456,321
vLReal  : 987654321,123457
vString : _:;MNBVCXYÄÖLKJHGFDSAÜPOIUZTREWQ
vTime   : 11:22:33.0440000
vTod    : 11:22:33.4440000
vDate   : 22.01.1999 00:00:00
vDT     : 22.01.1999 11:22:33
vAlias  : 9
vEnum   : 9
vRange  : -5
PSValue : ...

## Browse Data Types (Query by Category)
PS> $session | Get-TcDataType | where Category -eq "Array" }

Name                      Size     Category   Comment          ElementType      Dimensions       Mem
bers
----                      ----     --------   -------          -----------      ----------
       -------
ARRAY [-1..1] OF INT      6        Array                       INT              {TwinCAT.Type...
ARRAY [-10..-8] OF BOOL   3        Array                       BOOL             {TwinCAT.Type...
ARRAY [0..1] OF A_Alias   4        Array                       A_Alias          {TwinCAT.Type...
....
```

Browse DataTypes by name

```
PS> $session | Get-TcDataType -name "Array*"
```

Browse all Symbols recursively

```
PS> $session | Get-TcSymbol -recurse
... returns all symbols
```

Browse Symbols recursivly by Symbol Path (Here specific array index 'TaskInfo[1]'(

```
PS> $session | Get-TcSymbol -recurse -path "*TaskInfo``[1``]*","*.ProjectName"

InstanceName            DataType              Size InstancePath
             Comment
------------            --------              ---- ------------
                                                   -------
ProjectName             STRING(63)            64   TwinCAT_SystemInfoVarList._AppInfo.ProjectName
_TaskInfo[1]            PLC.PlcTaskSystemInfo 128  TwinCAT_SystemInfoVarList._TaskInfo[1]
ObjId                   OTCID                 4    TwinCAT_SystemInfoVarList._TaskInfo[1].ObjId
CycleTime               UDINT                 4    TwinCAT_SystemInfoVarList._TaskInfo[1].CycleTime
Priority                UINT                  2    TwinCAT_SystemInfoVarList._TaskInfo[1].Priority
AdsPort                 UINT                  2    TwinCAT_SystemInfoVarList._TaskInfo[1].AdsPort
CycleCount              UDINT                 4    TwinCAT_SystemInfoVarList._TaskInfo[1].CycleCount
DcTaskTime              LINT                  8    TwinCAT_SystemInfoVarList._TaskInfo[1].DcTaskTime
LastExecTime            UDINT                 4    TwinCAT_SystemInfoVarList._TaskInfo[1].LastExecTi
me
FirstCycle              BOOL                  1    TwinCAT_SystemInfoVarList._TaskInfo[1].FirstCycle
CycleTimeExceeded       BOOL                  1    TwinCAT_SystemInfoVarList._TaskInfo[1].CycleTimeE
xceeded
InCallAfterOutputUpdate BOOL                  1    TwinCAT_SystemInfoVarList._TaskInfo[1].InCallAfte
rOutputUpdate
RTViolation             BOOL                  1    TwinCAT_SystemInfoVarList._TaskInfo[1].RTViolatio
n
TaskName                STRING(63)            64   TwinCAT_SystemInfoVarList._TaskInfo[1].TaskName
```

Browse only Symbols ending with path *.ProjectName

```
PS>$project = Get-TcSymbol -Session $session -recurse -path "*.ProjectName"

InstanceName DataType   Size InstancePath                                    Comment
------------ --------   ---- ------------                                    -------
ProjectName  STRING(63) 64   TwinCAT_SystemInfoVarList._AppInfo.ProjectName
```

Ads Read ProjectName

```
PS>$project | Read-TcValue -Session $session
ADS_DynSymbols
```

Ads Write ProjectName

```
PS>$project | Write-TcValue -Session $session -Value "NewProjectName"
PS>$project | Read-TcValue -Session $session
NewProjectName
```

ReadWrite by Symbol Path

```
PS>Read-TcValue -SessionId 1 -Path "Main.bChange"
false
PS>Write-TcValue -SessionId 1 -Symbol "Main.bChange" -Value True
PS>Read-TcValue -SessionId 1 -Path "GVL.vgBool"
PS>Write-TcValue -SessionId 1 -Path "GVL.vgBool" -value $true
```

ReadWrite by Piping

```
PS> $projectNameSymbol = $session | Get-TcSymbol -Recurse -path "*ProjectName"
PS> $projectNameSymbol | Read-TcValue -SessionId 1
PS> $projectNameSymbol | Write-TcValue -SessionId 1 -Value "NewProjectName"
PS> $projectNameSymbol | Read-TcValue -SessionId 1
```

Get Target Information

```
PS> get-adsroute | Get-TcTargetInfo

Target          Version      Level OS   Image Device CPUArch
```

```
------          --------      ----- --   ----- ------ -------
TC3TESTA1-CP67X 3.1.4021.131 CP    Win7              IntelX86
```

```
PS> get-adsroute | Get-TcVersion

Major  Minor  Build  Revision
-----  -----  -----  --------
3      1      4021   131
```

PROVIDERS

The TcXaeMgmt module includes the AdsSymbolProvider and the AdsFileProvider

AdsSymbolProvider

Binds the target device symbolic information to a PSDrive. To register a symbol server as

PSDrive type (here the Target Route 'CX_01234' with AmsPort: 851)

```
PS> New-PSDrive -Name CX_01234_Symbols -PSProvider AdsSymbolProvider -Address CX_01234 -Port 851 -
Root ''
PS> cd CX_01234_Symbols:
PS CX_01234_Symbols:> dir
```

AdsFileProvider

```
PS> New-PSDrive -name CX_01234 -PSProvider AdsFileProvider -Address CX_01234 -Root ''
PS> cd CX_01234:
PS> dir

PS CX_01234:\> (dir).FullName
    CX_01234:\BootDir
    CX_01234:\BootProject
    CX_01234:\ConfigDir
    CX_01234:\Generic
    CX_01234:\InstallDir
    CX_01234:\Tc3Repository
    CX_01234:\TargetDir

PS CX_01234:\> cd BootDir
PS CX_01234:\BootDir>

PS CX_01234:\BootDir> (dir).FullName
    CX_01234:\BootDir\Current.cap
    CX_01234:\BootDir\CurrentConfig.tszip
    CX_01234:\BootDir\CurrentConfig.xml
    CX_01234:\BootDir\LoggedEvents.db
    CX_01234:\BootDir\Plc
    CX_01234:\BootDir\TCNC.bootdata
```

```
PS> get-help about_providers
```

FEEDBACK

Please submit any feedback, including defects and enhancement requests,

to

support@beckhoff.com

We are also interested in suggestions you may have for cmdlets. Over

time, we hope to be able to add some more features.

NOTE

To see what functions are provided by TcXaeMgmt, execute the command:

```
PS> Get-Command -Module TcXaeMgmt -CommandType Function
```

For more information, most of the cmdlets have help associated with

them e.g.:

```
PS> Get-Help Add-AdsRoute -full
```

The definitive information on a cmdlet's parameters can be obtained

by executing:

```
PS> Get-Command Add-AdsRoute -syntax
```

or more tersely:

```
PS> gcm Add-AdsRoute -syn
```

SEE ALSO

Documentation TcXaeMgmt Module

About the TcXaeMgmt Module

Beckhoff Homepage

```
PS> get-help about_providers
```

KEYWORDS

- ADS
- TwinCAT
- ManagementConsole
- Routes

6.10 Add-AdsRoute

SYNOPSIS

Cmdlet for adding TwinCAT Routes.

SYNTAX

Routes (Default)

```
Add-AdsRoute [-RemotePersistance <RoutePersistanceType>] -InputObject <IRoute[]> [-
Destination <String>]
 [-DestinationCredential <PSCredential>] -Credential <PSCredential> [-HostName] [-Temporary] [-
Unidirectional]
 [-Quiet] [-Force] [-Nat <AmsNetId>] [-PassThru] [-ProgressAction <ActionPreference>] [-WhatIf] [-
Confirm]
 [<CommonParameters>]
```

Address

```
Add-AdsRoute [-Name <String>] [-Address] <String[]> [-RemotePersistance <RoutePersistanceType>]
 [-BroadcastTimeout <Int32>] [-Destination <String>] [-DestinationCredential <PSCredential>]
 [-Credential <PSCredential>] [-HostName] [-Temporary] [-Unidirectional] [-Quiet] [-Force] [-
Nat <AmsNetId>]
 [-PassThru] [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

AddressPSK

```
Add-AdsRoute [-Name <String>] [-Address] <String[]> [-BroadcastTimeout <Int32>] [-
Destination <String>]
 [-DestinationCredential <PSCredential>] -Credential <PSCredential> [-HostName] [-Temporary] [-
PreSharedKey]
 [-Unidirectional] [-Quiet] [-Force] [-Nat <AmsNetId>] [-PassThru] [-
```

```
ProgressAction <ActionPreference>]
 [-WhatIf] [-Confirm] [<CommonParameters>]
```

AddressPSKKey

```
Add-AdsRoute [-Name <String>] [-Address] <String[]> [-BroadcastTimeout <Int32>] [-
Destination <String>]
 [-DestinationCredential <PSCredential>] -Identity <String> -BinaryKey <Byte[]> [-HostName] [-
Temporary]
 [-PreSharedKey] [-Unidirectional] [-Quiet] [-Force] [-Nat <AmsNetId>] [-PassThru]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

AddressSCA

```
Add-AdsRoute [-Name <String>] [-Address] <String[]> [-BroadcastTimeout <Int32>] [-
Destination <String>]
 [-DestinationCredential <PSCredential>] [-HostName] [-Temporary] [-SharedCertAuth] [-IgnoreCN]
 [-Unidirectional] [-Quiet] [-Force] [-Nat <AmsNetId>] [-PassThru] [-
ProgressAction <ActionPreference>]
 [-WhatIf] [-Confirm] [<CommonParameters>]
```

AddressSSC

```
Add-AdsRoute [-Name <String>] [-Address] <String[]> [-BroadcastTimeout <Int32>] [-
Destination <String>]
 [-DestinationCredential <PSCredential>] -Credential <PSCredential> [-HostName] [-Temporary] [-
SelfSigned]
 [-FingerPrint <String>] [-Unidirectional] [-Quiet] [-Force] [-Nat <AmsNetId>] [-PassThru]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

NetId

```
Add-AdsRoute [-Name <String>] [-NetId] <AmsNetId> [-IPOrHostName <String>]
 [-RemotePersistance <RoutePersistanceType>] [-Destination <String>] [-
DestinationCredential <PSCredential>]
 [-Credential <PSCredential>] [-HostName] [-Temporary] [-Unidirectional] [-Quiet] [-Force] [-
Nat <AmsNetId>]
 [-PassThru] [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

NetIdPSK

```
Add-AdsRoute [-Name <String>] [-NetId] <AmsNetId> [-IPOrHostName <String>] [-Destination <String>]
 [-DestinationCredential <PSCredential>] -Credential <PSCredential> [-HostName] [-Temporary] [-
PreSharedKey]
 [-Unidirectional] [-Quiet] [-Force] [-Nat <AmsNetId>] [-PassThru] [-
ProgressAction <ActionPreference>]
 [-WhatIf] [-Confirm] [<CommonParameters>]
```

NetIdPSKKey

```
Add-AdsRoute [-Name <String>] [-NetId] <AmsNetId> [-IPOrHostName <String>] [-Destination <String>]
 [-DestinationCredential <PSCredential>] -Identity <String> -BinaryKey <Byte[]> [-HostName] [-
Temporary]
 [-PreSharedKey] [-Unidirectional] [-Quiet] [-Force] [-Nat <AmsNetId>] [-PassThru]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

NetIdSCA

```
Add-AdsRoute [-Name <String>] [-NetId] <AmsNetId> [-IPOrHostName <String>] [-Destination <String>]
 [-DestinationCredential <PSCredential>] [-HostName] [-Temporary] [-SharedCertAuth] [-IgnoreCN]
 [-Unidirectional] [-Quiet] [-Force] [-Nat <AmsNetId>] [-PassThru] [-
ProgressAction <ActionPreference>]
 [-WhatIf] [-Confirm] [<CommonParameters>]
```

NetIdSSC

```
Add-AdsRoute [-Name <String>] [-NetId] <AmsNetId> [-IPOrHostName <String>] [-Destination <String>]
 [-DestinationCredential <PSCredential>] -Credential <PSCredential> [-HostName] [-Temporary] [-
SelfSigned]
```

```
 [-FingerPrint <String>] [-Unidirectional] [-Quiet] [-Force] [-Nat <AmsNetId>] [-PassThru]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

RoutesPSK

```
Add-AdsRoute -InputObject <IRoute[]> [-Destination <String>] [-DestinationCredential <PSCredential>]
 -Credential <PSCredential> [-HostName] [-Temporary] [-PreSharedKey] [-Unidirectional] [-Quiet] [-
Force]
 [-Nat <AmsNetId>] [-PassThru] [-ProgressAction <ActionPreference>] [-WhatIf] [-
Confirm] [<CommonParameters>]
```

RoutesPSKKey

```
Add-AdsRoute -InputObject <IRoute[]> [-Destination <String>] [-DestinationCredential <PSCredential>]
 -Identity <String> -BinaryKey <Byte[]> [-HostName] [-Temporary] [-PreSharedKey] [-
Unidirectional] [-Quiet]
 [-Force] [-Nat <AmsNetId>] [-PassThru] [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm]
 [<CommonParameters>]
```

RoutesSCA

```
Add-AdsRoute -InputObject <IRoute[]> [-Destination <String>] [-DestinationCredential <PSCredential>]
 [-HostName] [-Temporary] [-SharedCertAuth] [-IgnoreCN] [-Unidirectional] [-Quiet] [-Force] [-
Nat <AmsNetId>]
 [-PassThru] [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

RoutesSSC

```
Add-AdsRoute -InputObject <IRoute[]> [-Destination <String>] [-DestinationCredential <PSCredential>]
 -Credential <PSCredential> [-HostName] [-Temporary] [-SelfSigned] [-FingerPrint <String>] [-
Unidirectional]
 [-Quiet] [-Force] [-Nat <AmsNetId>] [-PassThru] [-ProgressAction <ActionPreference>] [-WhatIf] [-
Confirm]
 [<CommonParameters>]
```

DESCRIPTION

Adds a Route to the destination target System (Temporary or statically).

Dependant on the used parameters, this Cmdlet uses an internal broadcast search to determine the target system Addresses (NetId, HostName or IPAddress) to establish a full defined route.

To find the specified target it is necessary that the target system is running/online and reachable from the local system.

Another use case is to establish a route on the local system for preparation before the targeting system is available.

In that case the full target address represented by the -NetID and -IPOrHostName parameter must be given, while leaving out the -Credential parameter.

EXAMPLES

EXAMPLE 1

```
> Get-AdsRoute -All -name "Tc3*"

Name            NetId             Address       Sub TcVersion RTSystem
----            -----             -------       --- --------- --------
TC3TestA1-CP67x 172.17.62.105.1.1 172.17.62.105     3.1.4021  Win7
TC3Test13-C6650 172.17.60.239.1.1 172.17.62.156     2.11.2246 Win7


PS> $cred = Get-Credential -Message "Get Credentials" -UserName "UserName"

PS> Add-AdsRoute -Credential $cred -name "TC3TestA1-CP67x" -temporary -passthru

Name            NetId             Address       Sub TcVersion RTSystem
----            -----             -------       --- --------- --------
TC3TestA1-CP67x 172.17.62.105.1.1 172.17.62.105     3.1.4021  Win7
```

```
PS> Get-AdsRoute -name "TC3TestA1-CP67x" | Test-AdsRoute
```

Search for Systems that start with the name "TC3*", then asks the user for Credentials and adds the Route as 'temporary' (with TC2 compatible security, clear text password).

Afterwards, the connection is checked via 'Test-AdsRoute'.

The route is specified by its name (ComputerName).

To find out the address of the route an under the hood broadcast search is necessary what means that the target system must be online available in the network.

EXAMPLE 2

```
> Add-AdsRoute -name Test -NetId 1.2.3.4.1.1 -IPOrHostName 1.2.3.4
```

Adds a Route named 'Test' to the local routes with the specified NetId and IPAddress.

Because NetId and IPOrHostName are defined AND no credentials are set, this route is added locally only.

Be aware that to get the route functional, the target system must define the backroute.

EXAMPLE 3

```
> Get-AdsRoute -All -name "Tc3*"

Name            NetId             Address       Sub TcVersion RTSystem
----            -----             -------       --- --------- --------
TC3TestA1-CP67x 172.17.62.105.1.1 172.17.62.105     3.1.4021  Win7
TC3Test13-C6650 172.17.60.239.1.1 172.17.62.156     2.11.2246 Win7


PS> $cred = Get-Credential -Message "Get Credentials" -UserName "UserName"

PS> Add-AdsRoute -Credential $cred -name "TC3TestA1-CP67x" -selfSigned -passthru

Name            NetId             Address       Sub TcVersion RTSystem
----            -----             -------       --- --------- --------
TC3TestA1-CP67x 172.17.62.105.1.1 172.17.62.105     3.1.4021  Win7

PS> Get-AdsRoute -name "TC3TestA1-CP67x" | Test-AdsRoute
```

Search for Systems that start with the name "TC3*", then asks the user for Credentials and adds the Route with 'SelfSigned' AdsSecure settings.

Afterwards, the connection is checked via 'Test-AdsRoute'.

The route is specified by its name (ComputerName).

To find out the address of the route an under the hood broadcast search is necessary what means that the target system must be online available in the network.

EXAMPLE 4

```
PS> Add-AdsRoute -Address 172.17.62.105 -sca -paththru

Name            NetId             Address       Sub TcVersion RTSystem
----            -----             -------       --- --------- --------
TC3TestA1-CP67x 172.17.62.105.1.1 172.17.62.105     3.1.4021  Win7
```

Searches for the system with the specified IPAddress, and add the Route with Shared Certification Authority settings without password.

The precondition is, that valid certificates are already established on both (engineering and remote) systems, within their StaticRoutes.xml files.

The route is specified by its Address only.

Because the NetId is missing a broadcast search is necessary what means that the target system must be online available in the network.

EXAMPLE 5

```
PS> $cred = Get-Credential -Message "Get Credentials" -UserName "UserName"

PS> Add-AdsRoute -Credential $cred -NetId 172.17.62.105 -Nat 1.2.3.4.1.1

Name            NetId       Address       Sub TcVersion RTSystem
--------------- ----------- ------------- --- --------- ------------
TC3TestA1-CP67x 1.2.3.4.1.1 172.17.62.105     3.1.4024  Win10 (2004)
```

Add a route with a local network address translation (NAT AmsNetId) to project a remote AmsNetId (RemoteNetId) locally to a different address.

EXAMPLE 6

```
Add-AdsRoute -name "TestRoute" -NetId 1.2.3.4.1.1 -IPOrHostName 1.2.3.4 -Temporary -
RemotePersistance None
```

Adding a route 'TestRoute' single sided and temporary only to the local system.

The remote device doesn't need to be online.

EXAMPLE 7

```
PS> $route = get-adsroute CX_01234 -all
PS> $route

Name             NetId                TLS   Address          FingerPrint
----             -----                ---   -------          -----------
CX_01234         172.17.60.197.1.1    X     172.17.60.197    7835dae7a079c4f296c84109b2e6d7156b66e6b
cc39e386c3576d7535…

PS> $route | add-adsroute -SharedCertAuth -IgnoreCN -passthru

Name             NetId                TLS   Address          FingerPrint
----             -----                ---   -------          -----------
CX_01234         172.17.60.197.1.1    X     172.17.60.197    7835dae7a079c4f296c84109b2e6d7156b66e6b
cc39e386c3576d7535…
```

Broadcast search for a Device with Hostname CX_01234 and adding of a ADSSecure route via 'Shared Certificate Authority' (SCA) to the local system.

Both systems must contain certificates derived from the same root CA certificate.

EXAMPLE 8

```
PS> $cred = get-credential
UserName: MyUser
Password: ********

PS> $route = get-adsroute CX_01234 -all
PS> $route

Name             NetId                TLS   Address          FingerPrint
----             -----                ---   -------          -----------
CX_01234         172.17.60.197.1.1    X     172.17.60.197    7835dae7a079c4f296c84109b2e6d7156b66e6b
cc39e386c3576d7535…

PS> $route | add-adsroute -PreSharedKey -Credential $cred

Name             NetId                TLS   Address          FingerPrint
----             -----                ---   -------          -----------
CX_01234         172.17.60.197.1.1    X     172.17.60.197    7835dae7a079c4f296c84109b2e6d7156b66e6b
cc39e386c3576d7535…
```

Broadcast search for a Device with Hostname CX_01234 and adding of a ADSSecure route via 'Preshared key' (UserName, Password) to the local system.

The target system must already contain the preshared key configuration (as Psk Identity/Password) in its StaticRoutes.xml configuration file.

EXAMPLE 9

```
PS> $route = get-adsroute CX_01234 -all
PS> $route

Name             NetId                TLS   Address          FingerPrint
----             -----                ---   -------          -----------
CX_01234         172.17.60.197.1.1    X     172.17.60.197    7835dae7a079c4f296c84109b2e6d7156b66e6b
cc39e386c3576d7535…

PS> $route | add-adsroute -PreSharedKey -Identity MyUser -
BinaryKey 1,2,3,4,5,6,7,8,9,0xa,0xb,0xc,0xd,0xe,0xf

Name             NetId                TLS   Address          FingerPrint
----             -----                ---   -------          -----------
CX_01234         172.17.60.197.1.1    X     172.17.60.197    7835dae7a079c4f296c84109b2e6d7156b66e6b
cc39e386c3576d7535…
```

Broadcast search for a Device with Hostname CX_01234 and adding of a ADSSecure route via 'Preshared key' (Identity, BinaryKey) to the local system.

The target system must already contain the preshared key configuration (as Psk Identity/BinaryKey) in its StaticRoutes.xml configuration file.

PARAMETERS

-Name

The name of the route(s) to add.

If the Routes address is ambiguous and more than one route will be found online for adding then the route names will be numbered to be distinct.

Without setting this parameter, the default route name will be its Computername / Hostname.

```
Type: String
Parameter Sets: Address, AddressPSK, AddressPSKKey, AddressSCA, AddressSSC, NetId, NetIdPSK, NetIdPS
KKey, NetIdSCA, NetIdSSC
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Address

The address for the ADS route.

This can be the RouteName, NetId, the HostName or the IPAddress.

```
Type: String[]
Parameter Sets: Address, AddressPSK, AddressPSKKey, AddressSCA, AddressSSC
Aliases: TargetAddress

Required: True
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-NetId

The AmsNetID for the ADS route to add.

If no further IPAddress or HostName is specified with the -IPOrHostName parameter, a broadcast search is triggered to find an online device.

If a single sided route should be added, specify the IPAddress or HostName Parameter in combination with -RemotePersistance:None and without -Credential.

```
Type: AmsNetId
Parameter Sets: NetId, NetIdPSK, NetIdPSKKey, NetIdSCA, NetIdSSC
Aliases: TargetNetId

Required: True
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-IPOrHostName

The HostName Address of the target route or the IPAddress.

Because the HostName or IPAddress is necessary in addition to the NetId for a functional route, the Add-AdsRoute Cmdlet tries to detect the HostName/IPAddress via a Broadcast search in the Network when it is not specified.

That means the target must be available and reachable within the network in that case.

If not the Add-AdsRoute Cmdlet will fail.

If the IP or HostName in combination with the -NetId is specified, the target availability is not necessary and Add-Route will register the Route whatever is specified as address.

```
Type: String
Parameter Sets: NetId, NetIdPSK, NetIdPSKKey, NetIdSCA, NetIdSSC
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-RemotePersistance

The persistance type of the remmote route.

None/Server means no remote route will be created.

Other valid values are 'Static' or 'Temporary'

Possible values: None, Server, Temporary, Static

```
Type: RoutePersistanceType
Parameter Sets: Routes, Address, NetId
Aliases:
Accepted values: None, Server, Temporary, Static

Required: False
Position: Named
Default value: Static
Accept pipeline input: False
Accept wildcard characters: False
```

-BroadcastTimeout

(Broadcast) Search Timeout for searching the unregistered target in seconds (Default 0, Dynamic detection).

```
Type: Int32
Parameter Sets: Address, AddressPSK, AddressPSKKey, AddressSCA, AddressSSC
Aliases:

Required: False
Position: Named
Default value: 0
```

```
Accept pipeline input: False
Accept wildcard characters: False
```

-InputObject

The input Ads Routes.

```
Type: IRoute[]
Parameter Sets: Routes, RoutesPSK, RoutesPSKKey, RoutesSCA, RoutesSSC
Aliases: Route, TargetRoute

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-Destination

The Destination Address, where the route is added.

```
Type: String
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-DestinationCredential

The credentials of the destination system, where to add the route.

Local system by default.

```
Type: PSCredential
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Credential

Credentials of the route to be added to the destination system.

This parameter is only necessary, when a bidirectional route will be added.

When specifying

IMPORTANT: Please be aware, that in the current version, tha password is transferred as clear text through the network.

Use this only in safe subnetworks.

```
Type: PSCredential
Parameter Sets: Routes, AddressPSK, AddressSSC, NetIdPSK, NetIdSSC, RoutesPSK, RoutesSSC
Aliases: TargetCredential

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

```
Type: PSCredential
Parameter Sets: Address, NetId
Aliases: TargetCredential

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Identity

The BinaryKey can be used instead of the credential Password on SecureSettings.PreSharedKeys (PSK).

There is no function for this parameter on other security settings.

```
Type: String
Parameter Sets: AddressPSKKey, NetIdPSKKey, RoutesPSKKey
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-BinaryKey

The BinaryKey can be used instead of the credential Password on SecureSettings.PreSharedKeys (PSK).

There is no function for this parameter on other security settings.

```
Type: Byte[]
Parameter Sets: AddressPSKKey, NetIdPSKKey, RoutesPSKKey
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-HostName

If set, the route will be registered as HostName

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Temporary

If set, the Route will be registered as temporary route.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-SelfSigned

Gets or sets the SelfSigned (SSC) mode for adding the route.

```
Type: SwitchParameter
Parameter Sets: AddressSSC, NetIdSSC, RoutesSSC
Aliases: SSC

Required: True
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-FingerPrint

The Fingerprint used for adding the route.

This parameter can be used when the parameter '-SelfSigned' is set.

If specified, the found OnlineTarget will be checked against this fingerprint.

If not specified, the 'Add-AdsRoute' Cmdlet doesn't check the fingerprint, always adding the route.

Using the fingerprint means that only single routes can be added, multi-adding routes with fingerprint is not supported.

```
Type: String
Parameter Sets: AddressSSC, NetIdSSC, RoutesSSC
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-PreSharedKey

Gets or sets the PreSharedKey (PSK) mode for adding the route.

Because a valid certificate is expected at the target, it is not necessary to enter credentials.

```
Type: SwitchParameter
Parameter Sets: AddressPSK, AddressPSKKey, NetIdPSK, NetIdPSKKey, RoutesPSK, RoutesPSKKey
Aliases: PSK

Required: True
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-SharedCertAuth

Gets or sets the SharedCertificateAuthority (SCA) mode for adding the route.

Because a valid certificate is expected at the target, it is not necessary to enter credentials.

```
Type: SwitchParameter
Parameter Sets: AddressSCA, NetIdSCA, RoutesSCA
Aliases: SCA

Required: True
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-IgnoreCN

Gets or sets the 'Ignore Common Name' mode for SharedCertificateAuthority (SCA) while adding the route.

The "CommonName" of the certificate must correspond to the name used when establishing the connection in the certificate.

This behavior can be deactivated by this option.

```
Type: SwitchParameter
Parameter Sets: AddressSCA, NetIdSCA, RoutesSCA
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Unidirectional

Gets or sets the unidirectional setting.

The Unidirectional setting registers the ADS Route as 'one-way' channel.

That means that the engineering/source system (thats where the route request is initiated) can send requests to the remote target, but not in the opposite direction.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Quiet

The Quiet parameter suppresses the 'ShouldProcess' messsage and the Cmdlet will be processed without further user confirmation.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: Silent

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Force

Forces the command (no confirmation, Resets the FailFastHandler)

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Nat

The 'Nat' parameter sets the local representation of the routes AmsNetId.

All (local) addressing to this netId will be translated to the remote/network AmsAddress of the route.

If using the '-Nat' parameter, the 'Add-AdsRoute' Cmdlet is limited to single route additions.

Multi-adding is not supported.

This Parameter can be used with TwinCAT Versions \>= 3.1.4024.11.

```
Type: AmsNetId
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-PassThru

If the passthrough parameter is set, the successfully created route will be returned as object.

By default, this Cmdlet will not create any output.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Confirm

Prompts you for confirmation before running the cmdlet.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: cf

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-WhatIf

Shows what would happen if the cmdlet runs.

The cmdlet is not run.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: wi
```

```
Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.IRoute[]

The input Ads Routes.

OUTPUTS

NOTES

6.11 Add-MqttRoute

SYNOPSIS

Adds an MQTT route to the destination system.

SYNTAX

Default (Default)

```
Add-MqttRoute [-Address] <String> [-Port] <Int32> [[-Topic] <String>] [-Destination <String>] [-
Quiet] [-Force]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

Identity

```
Add-MqttRoute [-Address] <String> [-Port] <Int32> [[-Topic] <String>] [-Destination <String>]
 -Credential <PSCredential> [-IdentityCaseSensitive] [-Quiet] [-Force] [-
ProgressAction <ActionPreference>]
 [-WhatIf] [-Confirm] [<CommonParameters>]
```

Psk

```
Add-MqttRoute [-Address] <String> [-Port] <Int32> [[-Topic] <String>] [-Destination <String>]
 -Identity <String> -PreSharedKey <String> [-Quiet] [-Force] [-ProgressAction <ActionPreference>] [-
WhatIf]
 [-Confirm] [<CommonParameters>]
```

SCA

```
Add-MqttRoute [-Address] <String> [-Port] <Int32> [[-Topic] <String>] [-Destination <String>] -
CA <String>
 -Cert <String> -Key <String> [-Quiet] [-Force] [-ProgressAction <ActionPreference>] [-WhatIf] [-
Confirm]
 [<CommonParameters>]
```

DESCRIPTION

This Cmdlet adds an MQTT route to the destination system.

To add the route, the Address of a MQTT route must be specified.

EXAMPLES

EXAMPLE 1

```
PS> Add-MqttRoute -Address 1.2.3.4 -port 42
```

Adds the MQTT route to an MQTT Broker system with the IPAddress '1.2.3.4' and Port '42' on the local system.

EXAMPLE 2

```
PS> Add-MqttRoute -Address MqttSystem -port 42 -Destination CX_1234
```

Adds the MQTT route on the destination System 'CX_1234' to the MQTT Broker with Address '1.2.3.4' and Port '42'.

PARAMETERS

-Address

The Address of the MQTT Broker to add.

This can be the HostName or the IPAddress.

```
Type: String
Parameter Sets: (All)
Aliases:

Required: True
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Port

The TCP/IP Port of the MQTT Broker to add.

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: True
Position: 1
Default value: 0
Accept pipeline input: False
Accept wildcard characters: False
```

-Topic

The MQTT Topic string under which this MQTT Consumer sends/receives data.

```
Type: String
Parameter Sets: (All)
Aliases:

Required: False
Position: 2
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Destination

The Destination Address, where the MQTT route is added remotely.

```
Type: String
Parameter Sets: (All)
Aliases:
```

```
Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Credential

Credentials of the Preshared Key Identity.

IMPORTANT: Please be aware, that in the current version, tha password is transferred as clear text through the network.

Use this only in safe subnetworks.

```
Type: PSCredential
Parameter Sets: Identity
Aliases: TargetCredential

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-IdentityCaseSensitive

Key will be generated Sha256(Identity+Pwd), Identity in upper case if 'IdentityCaseSensitive' = false - UTF8

```
Type: SwitchParameter
Parameter Sets: Identity
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Identity

The identity name used to talk to the MQTT message broker (Preshared Key method).

```
Type: String
Parameter Sets: Psk
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-PreSharedKey

The Preshared key used togehter with the identity for MQTT message broker communication.

```
Type: String
Parameter Sets: Psk
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-CA

Path to the Certificate Authority file.

Certificates of MQTT broker, signed by this CA will be accepted for connection.

The file must be already located on the target system.

```
Type: String
Parameter Sets: SCA
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Cert

Path to the public key Certificate (X.509).

The file must be already located on the target system.

```
Type: String
Parameter Sets: SCA
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Key

Path of the private Key file of the X.509 Certifcate.

The file must be already located on the target system.

```
Type: String
Parameter Sets: SCA
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Quiet

The Quiet parameter suppresses the 'ShouldProcess' messsage and the Cmdlet will be processed without further user confirmation.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: Silent

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Force

Forces the command (no confirmation, Resets the FailFastHandler)

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
```

```
Accept pipeline input: False
Accept wildcard characters: False
```

-Confirm

Prompts you for confirmation before running the cmdlet.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: cf

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-WhatIf

Shows what would happen if the cmdlet runs.

The cmdlet is not run.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: wi

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

OUTPUTS

NOTES

6.12 Close-TcSession

SYNOPSIS

Closes the specified session.

SYNTAX

Default (Default)

```
Close-TcSession -Id <Int32> [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

Session

```
Close-TcSession -InputObject <ISession> [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

DESCRIPTION

This Cmdlet closes the specified Point-To-Point Connection to the TwinCAT Target that is represented by the returned session object.

All registered SessionProvider types of Sessions can be used here.

If using ADS as protocol, this Cmdlet is equivalent to Close/Dispose/Disconnect an ADS Client.

EXAMPLES

EXAMPLE 1

```
PS> $session = New-TcSession -NetId '1.2.3.4.1.1' -port 851
PS> $session | Get-AdsState

Name    State  OK   Time (ms) Address
----    -----  --   --------- -------
CX_1234 Config True 3         1.2.3.4.1.1:10000

PS> $session | Close-TcSession
```

Opens a session to the registered route with AmsNetId: 1.2.3.4.1.1 and closes the ADS Session again.:

PARAMETERS

-Id

The session object to close is specified by this session ID.

```
Type: Int32
Parameter Sets: Default
Aliases:

Required: True
Position: Named
Default value: 0
Accept pipeline input: False
Accept wildcard characters: False
```

-InputObject

The Session object to close.

```
Type: ISession
Parameter Sets: Session
Aliases: Session

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.ISession

The Session object to close.

OUTPUTS

NOTES

6.13 Copy-AdsFile

SYNOPSIS

Uploads / Downloads files from/to TwinCAT target.

SYNTAX

NetId (Default)

```
Copy-AdsFile [-Path] <String> [[-Destination] <String>] [-Directory <PathSpecifier>] [-Upload] [-
Force]
 [-NetId <AmsNetId>] [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

Route

```
Copy-AdsFile [-Path] <String> [[-Destination] <String>] [-Directory <PathSpecifier>] [-Upload] [-
Force]
 -InputObject <IRoute> [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

AddressStr

```
Copy-AdsFile [-Path] <String> [[-Destination] <String>] [-Directory <PathSpecifier>] [-Upload] [-
Force]
 -Address <String> [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

SessionId

```
Copy-AdsFile [-Path] <String> [[-Destination] <String>] [-Directory <PathSpecifier>] [-Upload] [-
Force]
 -SessionId <Int32> [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

DESCRIPTION

This Cmdlet implements ADS file transfer operations with TwinCAT Systems.

EXAMPLES

EXAMPLE 1

```
PS > Copy-AdsFile -address  CX_00001 -path CurrentConfig.xml -Destination c:\tmp\Config1.xml -
Directory BootDir
```

Downloads the the CurrentConfig.xml from the BootDir of the target system to 'c:\tmp\Config1.xml'

EXAMPLE 2

```
PS > Copy-AdsFile -address CX_00001 -upload -path c:\tmp\Config1.xml -
destination CurrentConfig.xml -Directory BootDir
```

Uploads the file "c:\tmp\Config1.xml" on local system to the Target BootFolder of system CX_00001

EXAMPLE 3

```
PS > Copy-AdsFile -address CX_0001 -path c:\ReadMe.txt -destination d:\tmp\
```

Downloads the File "C:\ReadMe.txt" form System CX_0001 to the local system and store it under d:\tmp\ReadMe.txt

PARAMETERS

-Path

The source path specifier, where the file is taken from.

If this Cmdlet is in Download mode, this is the specifier or FullPath of the (remote) file, dependant of the StandardFolder Parameter.

In case of 'Uploading' this is the FullPath of the file to be transferred.

```
Type: String
Parameter Sets: (All)
Aliases:

Required: True
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Destination

The Destination path/specifier, where the file is stored.

If the Cmdlet is in Download mode, this has to be the FullPath of the target location.

In case of 'Uploading' this can be the FileName or a FullPath dependent of the StandardDirectory Parameter.

```
Type: String
Parameter Sets: (All)
Aliases:

Required: False
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Directory

The Directory specifier on the remote system.

The Default is "Generic".

Possible values: Generic, BootDir, TargetDir, ConfigDir, InstallDir, RepositoryDir, UserPath1, UserPath2, UserPath3, UserPath4, UserPath5, UserPath6, UserPath7, UserPath8, UserPath9

```
Type: PathSpecifier
Parameter Sets: (All)
Aliases:
Accepted values: Generic, BootDir, TargetDir, ConfigDir, InstallDir, RepositoryDir, UserPath1, UserP
ath2, UserPath3, UserPath4, UserPath5, UserPath6, UserPath7, UserPath8, UserPath9

Required: False
Position: Named
Default value: Generic
Accept pipeline input: False
Accept wildcard characters: False
```

-Upload

Switches the Cmdlet to Upload mode.

If not set, the Cmdlet is in 'Download' mode.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Force

Forces to create the Directory on the target side (and overwrites any preexisting file).

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-NetId

The address (AmsNetId) of the system where the file is Downloaded from / Uploaded to (Default: Local)

```
Type: AmsNetId
Parameter Sets: NetId
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-InputObject

The address (Route) of the system where the file is Downloaded from / Uploaded to (Default: Local)

```
Type: IRoute
Parameter Sets: Route
Aliases: Route

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-Address

The address of the system where the file is Downloaded from / Uploaded to (Default: Local) This can be the RouteName, NetId, the HostName or the IPAddress.

```
Type: String
Parameter Sets: AddressStr
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-SessionId

The target system address is derived from the Session Information where the file is Downloaded from / Uploaded to.

```
Type: Int32
Parameter Sets: SessionId
Aliases: Id

Required: True
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.IRoute

The address (Route) of the system where the file is Downloaded from / Uploaded to (Default: Local)

OUTPUTS

NOTES

6.14 Get-AdsRoute

SYNOPSIS

List routes on a TwinCAT System / Broadcast search.

SYNTAX

GetRoutes (Default)

```
Get-AdsRoute [-BroadcastTimeout <Int32>] [[-Address] <String[]>] [-InputObject <IRoute>]
 [-Access <RouteAccessType>] [-Force] [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

Broadcast

```
Get-AdsRoute [-All] [-BroadcastTimeout <Int32>] [-NetAdapter <String[]>] [[-Address] <String[]>]
 [-InputObject <IRoute>] [-Force] [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

LocalSystem

```
Get-AdsRoute [-InputObject <IRoute>] [-Local] [-
ProgressAction <ActionPreference>] [<CommonParameters>]
```

DESCRIPTION

This Cmdlet can list the routes configured on a TwinCAT local/remote system, or start determining all TwinCAT Systems within the current subnet.

EXAMPLES

EXAMPLE 1

```
PS> Get-AdsRoute

Name            NetId             Address       Sub TcVersion   RTSystem
----            -----             -------       --- ---------   ---------
CP-15ECA0       172.17.62.128.1.1 172.17.62.178     [UNKNOWN]   [UNKNOWN]
CP-15ECA1       172.17.62.105.1.1 172.17.62.105     [UNKNOWN]   [UNKNOWN]
```

Lists all registered local routes.

Because only the local port 10000 is addressed, the TcVersion and RTSystem is unknown (the Cmdlet doesn't contact the targets and doesn't produce additional roundtrips.

EXAMPLE 2

```
PS> get-AdsRoute -All
Name            NetId               Address       Sub Version   RTSystem
----            -----               -------       --- -------   --------
CX-1CEEDA       5.16.136.222.1.1    172.17.62.139     3.1.4020  Win7
CX-20BC62       5.32.188.98.1.1     172.17.62.90      3.1.4020  CE6.0
CX-10A87B       5.16.168.123.1.1    172.17.62.140     2.11.2254 CE7.0
CP-15ECA0       172.17.62.128.1.1   172.17.62.178     3.1.4021  Win7
CX-0A7F60       5.10.127.96.1.1     172.17.62.148     3.1.4020  XP
CX2030-B4018    172.17.60.157.1.1   172.17.60.159     2.11.2256 Win7
CP_11BB16       5.17.187.22.1.1     172.17.60.180     2.11.2038 CE6.0
CX-128CE5       172.17.60.165.1.1   172.17.62.191     2.11.2237 CE7.0
CX-124218       5.18.66.24.1.1      172.17.60.192     3.1.4021  Win7
CX-1D82AA       172.17.62.180.1.1   172.17.62.180     3.1.4021  Win8
CX_0AB4F0       5.10.180.240.1.1    172.17.60.195     2.11.2243 XP
CP-1DFA0A       172.17.62.118.1.1   172.17.62.118     3.1.4021  Win7
CX-AF0001       172.17.62.75.1.1    172.17.62.70      3.1.4020  Win10
```

Start a Broadcast search from the local system and lists the devices within the connected network.

EXAMPLE 3

```
PS> Get-AdsRoute -Name "Tc3*"

 Name            NetId             Address       Sub Version RTSystem
----            -----             -------       --- ------- --------
TC3TESTA1-CP67X 172.17.62.105.1.1 172.17.62.105     0.0     Unknown
```

Get the (actual) route assigned to the local system that has the name pattern "Tc3*"

EXAMPLE 4

```
PS> Get-AdsRoute -All | where TcVersion -lt "3.1.0.0"
Name            NetId              Address       Sub Version   RTSystem
----            -----              -------       --- -------   --------
TC3Test17-C6930 172.17.62.98.1.1   172.17.62.98      2.11.2234 Win7
CX2030-B4018    172.17.60.157.1.1  172.17.60.159     2.11.2256 Win7
CX-10A87B       5.16.168.123.1.1   172.17.62.140     2.11.2254 CE7.0
TC3Test13-C6650 172.17.60.239.1.1  172.17.62.156     2.11.2246 Win7
ECATTest01      172.17.61.6.1.1    172.17.61.31      2.11.2239 Win7
CX-128CE5       172.17.60.165.1.1  172.17.62.191     2.11.2237 CE7.0
CX_0AB4F0       5.10.180.240.1.1   172.17.60.195     2.11.2243 XP
CP_11BB16       5.17.187.22.1.1    172.17.60.180     2.11.2038 CE6.0
```

Find out all TwinCAT Systems within the network with Version numbers lower than '3.1.0.0'

EXAMPLE 5

```
PS> $runningAdaptors = @(get-netadapter | Where-Object -Property Status -eq Up)

PS> get-AdsRoute -all -NetAdapterName $runningAdaptors[0].Name -verbose

VERBOSE: Broadcast search from system 'CX_11111' ...
VERBOSE: Broadcasting over the Network Adapter(s) 'Ethernet 1'

Name         NetId               Protocol   TLS   Address          FingerPrint
----         -----               --------   ---   -------          -----------
CX_22222     192.168.0.2.1.1     TcpIP      X     192.168.0.2      xxxxxxxxxxx

VERBOSE: Broadcast search finished. Found '1' route(s)
```

Determines the first active Network adapter for broadcasting and returns the found targets.

PARAMETERS

-All

Broadcast switch.

If activated a broadcast search is triggered within the local network.

The search can be constrained additionally by the -Address/-Name parameter.

Searching by Address (direct access of targets if no wildcards, otherwise using Broadcast search): - HostName: Searching the target by dns resolution and then via IP (fallback broadcast search filtering DeviceName/Hostname, not working over subnets!) - IPAddress: Directly accessing via IP (works also over subnets) - AmsNetId: Working via Broadcast search (not working over subnet segments!) Searching by Name: Works always via Broadcast search, wildcards permitted

```
Type: SwitchParameter
Parameter Sets: Broadcast
Aliases: Broadcast

Required: True
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-BroadcastTimeout

(Broadcast) Search Timeout in Seconds (Default 0 Seconds) 0 Seconds means that the Length of the search operation will be determined dynamically.

If for a longer period no targets are arriving.

```
Type: Int32
Parameter Sets: GetRoutes, Broadcast
Aliases:

Required: False
Position: Named
```

```
Default value: 0
Accept pipeline input: False
Accept wildcard characters: False
```

-NetAdapter

The Network adapter name of the NetworkAdapter to use.

By default e.g.

Broadcast searches go out via all active Network adapters.

This parameter can be use to restrict this.

```
Type: String[]
Parameter Sets: Broadcast
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Address

The Name / Address of the route to get.

The address of the route can be coded as NetId, the HostName or the IPAddress in string representation.

Wildcards are permitted.

```
Type: String[]
Parameter Sets: GetRoutes, Broadcast
Aliases: Name

Required: False
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-InputObject

The Destination address specifies the target, where the the routes are determined.

Use this to get the registered routes of a remote system.

The Destination system can be specified by RouteName (route name on local system), AmsNetId, IPAddress or HostName

```
Type: IRoute
Parameter Sets: (All)
Aliases: Destination

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Access

Defines the strategy how to get the remote routes.

Valid methods are 'Merged', 'Actual', 'Configuration', 'Registry'.

The default value is 'Default' / 'Merged'

Possible values: None, Actual, Registry, Configured, Merged, Default

```
Type: RouteAccessType
Parameter Sets: GetRoutes
Aliases:
Accepted values: None, Actual, Registry, Configured, Merged, Default

Required: False
Position: Named
Default value: Merged
Accept pipeline input: False
Accept wildcard characters: False
```

-Local

If set, the local system route will be returned.

By default a list of the actual registered routes will be returned.

```
Type: SwitchParameter
Parameter Sets: LocalSystem
Aliases: Self

Required: True
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Force

If set, the broadcast search won't use cached routes.

The Route will be determined by broadcast always.

Only available with the -All parameter.

```
Type: SwitchParameter
Parameter Sets: GetRoutes, Broadcast
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

OUTPUTS

NOTES

6.15 Get-AdsState

SYNOPSIS

Gets the Ads State of a TwinCAT Target.

SYNTAX

NetIdPort (Default)

```
Get-AdsState [[-NetId] <AmsNetId[]>] [[-Port] <Int32>] [-Quiet] [-StateOnly] [-Force] [-
Timeout <Int32>]
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

AddressStr

```
Get-AdsState [[-Port] <Int32>] [[-Address] <String[]>] [-Quiet] [-StateOnly] [-Force] [-
Timeout <Int32>]
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

Session

```
Get-AdsState [[-Port] <Int32>] -Session <ISession[]> [-Quiet] [-StateOnly] [-Force] [-
Timeout <Int32>]
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

SessionId

```
Get-AdsState [[-Port] <Int32>] -SessionId <Int32[]> [-Quiet] [-StateOnly] [-Force] [-
Timeout <Int32>]
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

Route

```
Get-AdsState [[-Port] <Int32>] [-InputObject] <IRoute[]> [-Quiet] [-StateOnly] [-Force] [-
Timeout <Int32>]
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

DESCRIPTION

This command let gets the ADS state of a TwinCAT target.

EXAMPLES

EXAMPLE 1

```
PS > Get-AdsState 1.2.3.4.5.6

            Name   State  OK   Time (ms) Address
            ----   -----  --   --------- -------
            WORK01 Config True 0         1.2.3.4.5.6
```

Gets the actual AdsState from the remote target with NetId 1.2.3.4.5.6.

EXAMPLE 2

```
PS> Get-AdsState

            Name   State  OK   Time (ms) Address
            ----   -----  --   --------- -------
            WORK01 Config True 0         1.2.3.4.5.6
```

Gets the actual AdsState from the Local system.

EXAMPLE 3

```
PS> Get-AdsState 1.2.3.4,CX_0130C7
```

Gets the AdsState of target system with IPAddress 1.2.3.4 and Route name 'CX_0130C7'.

EXAMPLE 4

```
PS> get-route | get-adsState

            Name      State  OK   Time (ms) Address
            ----      -----  --   --------- -------
            WORK01    Config True 0         1.2.3.4.5.6
            CX_0130C7 Config True 0         5.1.48.199.1.1
```

Get the current target state from all registered routes.

EXAMPLE 5

```
PS> get-adsroute | get-adsstate -port 10000 -stateOnly
            Invalid
            Config
```

Gets the AdsState information from all actual routes.

EXAMPLE 6

```
PS> get-adsroute | get-adsstate -port 10000 -quiet
            false
            true
```

Gets availability information from all actual routes.

PARAMETERS

-NetId

The Addresses of the target systems, where to get the AdsState.

```
Type: AmsNetId[]
Parameter Sets: NetIdPort
Aliases:

Required: False
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Port

The AmsPort of the target system.

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: 2
Default value: 10000
```

```
Accept pipeline input: False
Accept wildcard characters: False
```

-Address

The address(es) where to get the State.

This can be the RouteName, NetId, the HostName or the IPAddress.

Wildcards are permitted.

```
Type: String[]
Parameter Sets: AddressStr
Aliases:

Required: False
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-Session

The Session to use for the Cmdlet.

```
Type: ISession[]
Parameter Sets: Session
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-SessionId

Specifies the Session (with unique ID) to use for the Cmdlet.

```
Type: Int32[]
Parameter Sets: SessionId
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-InputObject

The target systems, where to get the AdsState from.

```
Type: IRoute[]
Parameter Sets: Route
Aliases: Destination,Route

Required: True
Position: 1
Default value: None
Accept pipeline input: True (ByPropertyName, ByValue)
Accept wildcard characters: False
```

-Quiet

The quiet mode

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:
```

```
Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-StateOnly

The StateOnly mode

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Force

Forced Mode

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Timeout

The communication ADS timeout in milliseconds.

A value of 0 disables the timeout.

A value \<= 0 sets the Default (5000 ms).

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.ISession[]

The Session to use for the Cmdlet.

TwinCAT.IRoute[]

The target systems, where to get the AdsState from.

OUTPUTS

NOTES

6.16 Get-AmsRouterEndpoint

SYNOPSIS

Get the actual AmsConfiguration / RouterEndpoint of the process.

SYNTAX

```
Get-AmsRouterEndpoint [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

DESCRIPTION

This Cmdlet returns actual process-wide settings for the AmsConfiguration that define the TCP/IP Loopback Address that is used from the communication between AdsClient/Server and Router locally.

For more information please have a look at the SetAmsRouterEndpointCmdlet Cmdlet help.

EXAMPLES

EXAMPLE 1

```
PS> Get-AmsRouterEndpoint

            AddressFamily Address    Port
            ------------- -------    ----
            InterNetwork  127.0.0.1  48898
```

Returns the actual Loopback endpoint settings.

PARAMETERS

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
```

```
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

OUTPUTS

NOTES

6.17 Get-EcBoxes

SYNOPSIS

Gets the EtherCAT Boxes actually loaded on the specified target system.

SYNTAX

```
Get-EcBoxes [-InputObject] <EcMaster> [-Configured] [-Timeout <Int32>] [-
ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

DESCRIPTION

This command list the EtherCAT Boxes actually loaded on the target system.

EXAMPLES

EXAMPLE 1

```
PS> $m = Get-EcMaster -NetId 5.62.192.46.1.1
PS> $m | Get-EcBoxes

Pos Name             Type                Port  State CrcError        FW   HW    Production
--- ----             ----                ----  ----- ---------       --   --    ----------
2   Term 2 (EL1808)  EL1808-0000-0018    1001  PreOp A:0,B:0          7    0    2021-5-29
3   Term 3 (EL2088)  EL2088-0000-0018    1002  PreOp A:0,B:0          9    0    2021-6-04
4   Term 4 (EL2624)  EL2624-0000-0018    1003  PreOp A:0,B:0          12   1    2021-5-25
5   Term 5 (EL3064)  EL3064-0000-0020    1004  PreOp A:0,B:0          15   9    2021-6-01
6   Term 6 (EL4004)  EL4004-0000-0020    1005  PreOp A:0,B:0          19   4    2021-5-31
7   Term 7 (EL6021)  EL6021-0000-0021    1006  PreOp A:0,B:0          13   9    2021-6-01
8   Term 8 (EL9110)  EL9110-0000-0018    1007  PreOp A:0,B:0          14   0    2021-5-17
9   Term 9 (EL1004)  EL1004-0000-0016    1008  PreOp A:0,B:0          0    0    2000-1-02
10  Term 10 (EL2008) EL2008-0000-0016    1009  PreOp A:0,B:0          0    0    2000-1-02
```

Get the EtherCAT Master from NetID 5.62.192.46.1.1 and scan the connected (online) Boxes below this master

EXAMPLE 2

```
PS> $m = Get-EcMaster -NetId 5.62.192.46.1.1
PS> $m | Get-EcBoxes -configured

Pos Name             Type                Port  State CrcError         FW   HW    Production
--- ----             ----                ----  ----- ------------     --   --    ----------
2   Term 2 (EL1808)  EL1808-0000-0018    1001  PreOp A:0,B:0          7    0    2021-5-29
3   Term 3 (EL2088)  EL2088-0000-0018    1002  PreOp A:0,B:0          9    0    2021-6-04
4   Term 4 (EL2624)  EL2624-0000-0018    1003  PreOp A:0,B:0          12   1    2021-5-25
5   Term 5 (EL3064)  EL3064-0000-0020    1004  PreOp A:0,B:0          15   9    2021-6-01
6   Term 6 (EL4004)  EL4004-0000-0020    1005  PreOp A:0,B:0          19   4    2021-5-31
7   Term 7 (EL6021)  EL6021-0000-0021    1006  PreOp A:0,B:0          13   9    2021-6-01
```

```
8   Term 8 (EL9110)  EL9110-0000-0018     1007  PreOp A:0,B:0          14    0    2021-5-17
9   Term 9 (EL1004)  EL1004-0000-0016     1008  PreOp A:0,B:0          0     0    2000-1-02
10  Term 10 (EL2008) EL2008-0000-0016     1009  PreOp A:0,B:0          0     0    2000-1-02
```

Get the EtherCAT Master from NetID 5.62.192.46.1.1 and scan the configured Boxes below this master

PARAMETERS

-InputObject

The EtherCAT master where to scan the slaves.

```
Type: EcMaster
Parameter Sets: (All)
Aliases:

Required: True
Position: 1
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-Configured

The List of Configured Boxes will be returned if specified.

If not specified, this Cmdlet will return the Online Boxes.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Timeout

The communication ADS timeout in milliseconds.

A value of 0 disables the timeout.

A value \<= 0 sets the Default (5000 ms).

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: 5000
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

EtherCAT.EcMaster

The EtherCAT master where to scan the slaves.

OUTPUTS

NOTES

6.18 Get-EcFrameStatistics

SYNOPSIS

Gets the EtherCAT Frame statistics from an EtherCAT master.

SYNTAX

NetIdPortList (Default)

```
Get-EcFrameStatistics [-Timeout <Int32>] [-Count <Int32>] [-Delay <Int32>] [-
ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

Default

```
Get-EcFrameStatistics [-InputObject] <EcMaster> [-Timeout <Int32>] [-Count <Int32>] [-Delay <Int32>]
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

DESCRIPTION

Gets the EtherCAT Frame statistics from an ETherCAT master.

EXAMPLES

EXAMPLE 1

```
PS> $m = Get-EcMaster -NetId 5.62.192.46.1.1
PS> $m | Get-EcFrameStatistics -count 5 -delay 0

Frames(1/s)  Queued(1/s)  Lost(1/
s)    QueuedLost   TotalFrames  TotalQueued  TotalLost    TotalQueued
                                      (1/s)
------------ ------------ ----------   ------------ -----------  -----------  ---------
    -----------
100          30           0            0            1524222      572157       0            0
100          30           0            0            1524232      572160       0            0
101          40           0            0            1524242      572164       0            0
101          40           0            0            1524252      572168       0            0
99           39           0            0            1524262      572172       0            0
```

Getting the EtherCAT frame statistics of an EtherCAT Master.

This example calculates the statistics 5 times with a minimal delay (0).

PARAMETERS

-InputObject

The EtherCAT master.

```
Type: EcMaster
Parameter Sets: Default
Aliases:

Required: True
Position: 1
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-Timeout

ADS Communication timeout

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: 5000
Accept pipeline input: False
Accept wildcard characters: False
```

-Count

Specifies the number of statistic requests (Default is 1)

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: 1
Accept pipeline input: False
Accept wildcard characters: False
```

-Delay

Delay in Seconds between requests in Seconds (Default is 1s)

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: 1
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

EtherCAT.EcMaster

The EtherCAT master.

OUTPUTS

NOTES

6.19 Get-EcMaster

SYNOPSIS

Gets actually loaded EtherCAT Master devices on the target system.

SYNTAX

NetIdPortList (Default)

```
Get-EcMaster [-Id <Int32>] [-Timeout <Int32>] [-
ProgressAction <ActionPreference>] [<CommonParameters>]
```

NetIdPort

```
Get-EcMaster [-NetId <AmsNetId>] [-Id <Int32>] [-Timeout <Int32>] [-
ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

AddressStr

```
Get-EcMaster [-Address] <String> [-Id <Int32>] [-Timeout <Int32>] [-
ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

Session

```
Get-EcMaster -Session <ISession> [-Id <Int32>] [-Timeout <Int32>] [-
ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

DESCRIPTION

This command lists the loaded EtherCAT master devices on the specified target system.

The devices must be already loaded and being active on the EtherCAT Network.

EXAMPLES

EXAMPLE 1

```
PS> Get-EcMaster -NetId 5.91.172.198.1.1

 ID   Name                 Type                 DeviceNetId      Port   Slaves     Slaves     State
PortCrcError
                                                                      (Online)   (Config)
 --   ----                 ----                 -----------      ----   ---------- ---------- -----
 ------------
```

```
 1    Device 1 (EtherCAT)  P06b80001 R00000000  5.91.172.198.2.1 65535  9          9          Op
A:0,B:0,C:0,D:0

 PS> Get-EcBoxes -InputObject $m

Pos Name                 Type                 Port  State CrcError         FW   HW   Production
--- ----                 ----                 ----  ----- --------         --   --   ----------
2   Term 2 (EL1808)      EL1808-0000-0018     1001  Op    A:0,B:0          7    0    2021-5-29
3   Term 3 (EL2088)      EL2088-0000-0018     1002  Op    A:0,B:0          9    0    2021-6-04
4   Term 4 (EL2624)      EL2624-0000-0018     1003  Op    A:0,B:0          12   1    2021-5-25
5   Term 5 (EL3064)      EL3064-0000-0020     1004  Op    A:0,B:0          15   9    2021-6-01
6   Term 6 (EL4004)      EL4004-0000-0020     1005  Op    A:0,B:0          19   4    2021-5-31
7   Term 7 (EL6021)      EL6021-0000-0021     1006  Op    A:0,B:0          13   9    2021-6-01
8   Term 8 (EL9110)      EL9110-0000-0018     1007  Op    A:0,B:0          14   0    2021-5-17
9   Term 9 (EL1004)      EL1004-0000-0016     1008  Op    A:0,B:0          0    0    2000-1-02
10  Term 10 (EL2008)     EL2008-0000-0016     1009  Op    A:0,B:0          0    0    2000-1-02
```

Getting EtherCAT master and connected boxes from target system.

PARAMETERS

-NetId

Gets or sets the NetId of the target system.

```
Type: AmsNetId
Parameter Sets: NetIdPort
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Address

The address(es) of the target system(s) where to get the EtherCAT Master devices.

This can be the RouteName, NetId, the HostName or the IPAddress.

Wildcards are permitted.

```
Type: String
Parameter Sets: AddressStr
Aliases:

Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-Session

The Session to use for the Cmdlet, must be connected to port 300, R0_IO

```
Type: ISession
Parameter Sets: Session
Aliases: InputObject

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-Id

Specifies the DeviceId to scan by the Cmdlet.

If not specified, all Devices will be scanned.

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-Timeout

The communication ADS timeout in milliseconds.

A value of 0 disables the timeout.

A value \<= 0 sets the Default (5000 ms).

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.ISession

The Session to use for the Cmdlet, must be connected to port 300, R0_IO

OUTPUTS

NOTES

6.20 Get-IODevice

SYNOPSIS

Gets actually loaded IO Devices of the target system.

SYNTAX

NetIdPortList (Default)

```
Get-IODevice [-Id <Int32>] [-Timeout <Int32>] [-
ProgressAction <ActionPreference>] [<CommonParameters>]
```

NetIdPort

```
Get-IODevice [-NetId <AmsNetId>] [-Id <Int32>] [-Timeout <Int32>] [-
ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

AddressStr

```
Get-IODevice [-Address] <String> [-Id <Int32>] [-Timeout <Int32>] [-
ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

Session

```
Get-IODevice -Session <ISession> [-Id <Int32>] [-Timeout <Int32>] [-
ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

DESCRIPTION

This command lists the actually loaded IO Devices of the target system.

The list can be filted by specific Device IDs.

EXAMPLES

EXAMPLE 1

```
PS>  Get-IODevice -NetId 5.62.192.46.1.1

ID DeviceName          DeviceType              DeviceNetId    BoxesCount
-- ----------          ----------              -----------    ----------
1  Device 1 (EtherCAT) EtherCAT_DirectModeV210 172.16.1.3.2.1 4
```

Getting the IO Devices from NetID 5.62.192.46.1.1

EXAMPLE 2

```
PS> (Get-IODevice -Address CX_01234 -Id 1).Boxes

ID Name              BoxType         Port Comment
-- ----              -------         ---- -------
1  Box 1 (IFC2422)   EtherCAT_EXXXXX 1001
2  Box 2 (IFC2421m)  EtherCAT_EXXXXX 1002
3  Box 3 (IFC2421m1) EtherCAT_EXXXXX 1003
4  Box 4 (IFC2421m2) EtherCAT_EXXXXX 1004
```

Getting the Boxes of Device with Id 1 from the target system with Name/Address CX_01234

PARAMETERS

-NetId

The address where to get the device.

```
Type: AmsNetId
Parameter Sets: NetIdPort
Aliases:

Required: False
Position: Named
```

```
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Address

The address(es) where to get Devices.

This can be the RouteName, NetId, the HostName or the IPAddress.

Wildcards are permitted.

```
Type: String
Parameter Sets: AddressStr
Aliases:

Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-Session

The Session to use for the Cmdlet, must be connected to port 300, R0_IO

```
Type: ISession
Parameter Sets: Session
Aliases: InputObject

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-Id

Specifies the DeviceId to scan by the Cmdlet.

If not specified, all Devices will be scanned.

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: 0
Accept pipeline input: False
Accept wildcard characters: False
```

-Timeout

The communication ADS timeout in milliseconds.

A value of 0 disables the timeout.

A value \<= 0 sets the Default (5000 ms).

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.ISession

The Session to use for the Cmdlet, must be connected to port 300, R0_IO

OUTPUTS

NOTES

6.21 Get-IoFreeRun

SYNOPSIS

Gets the IO FreeRun State of the specified target.

SYNTAX

NetIdPortList (Default)

```
Get-IoFreeRun [-Timeout <Int32>] [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

NetIdPort

```
Get-IoFreeRun [-NetId <AmsNetId>] [-Timeout <Int32>] [-
ProgressAction <ActionPreference>] [<CommonParameters>]
```

AddressStr

```
Get-IoFreeRun [-Address] <String> [-Timeout <Int32>] [-
ProgressAction <ActionPreference>] [<CommonParameters>]
```

Session

```
Get-IoFreeRun -Session <ISession> [-Timeout <Int32>] [-
ProgressAction <ActionPreference>] [<CommonParameters>]
```

DESCRIPTION

This command gets the IO FreeRun state of specified target when the target is in config mode.

If the target system is not in config mode, a warning is produced.

EXAMPLES

EXAMPLE 1

```
PS> Get-IoFreeRun -NetId 5.62.192.46.1.1

$true
```

Getting the IO FreeRun State from NetID 5.62.192.46.1.1

PARAMETERS

-NetId

The address where to get the free run state.

```
Type: AmsNetId
Parameter Sets: NetIdPort
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Address

The address(es) where to get the free run state.

This can be the RouteName, NetId, the HostName or the IPAddress.

Wildcards are permitted.

```
Type: String
Parameter Sets: AddressStr
Aliases:

Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-Session

The Session to use for the Cmdlet, must be connected to port 300, R0_IO

```
Type: ISession
Parameter Sets: Session
Aliases: InputObject

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-Timeout

The communication ADS timeout in milliseconds.

A value of 0 disables the timeout.

A value \<= 0 sets the Default (5000 ms).

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
```

```
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.ISession

The Session to use for the Cmdlet, must be connected to port 300, R0_IO

OUTPUTS

NOTES

6.22 Get-MqttRoute

SYNOPSIS

Remove a MQTT Route.

SYNTAX

```
Get-MqttRoute [-Destination <String>] [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

DESCRIPTION

Removes a MQTT Route of the specified system.

EXAMPLES

EXAMPLE 1

```
PS> Get-MqttRoute -destination CX_1234
Address       TcpPort Topic  Qos Security
-------       ------- -----  --- --------
192.168.2.1   44124   Topic1     TLS
192.200.2.2   44124   Topic2     PSK
192.200.3.3   44124   Topic3     None
```

Gets the MQTT Routes registered on the destination System 'CX_1234'.

PARAMETERS

-Destination

The destination address, where to Remove the specified Mqtt route.

This can be the NetId, the HostName or the IPAddress

```
Type: String
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

OUTPUTS

NOTES

6.23 Get-RTimeCpuSettings

SYNOPSIS

Getting the Cpu Settings of the TwinCAT System

SYNTAX

NetIdPort (Default)

```
Get-RTimeCpuSettings [-NetId <AmsNetId>] [-Timeout <Int32>] [-ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

AddressStr

```
Get-RTimeCpuSettings [-Address] <String> [-Timeout <Int32>] [-ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

Session

```
Get-RTimeCpuSettings -Session <ISession> [-Timeout <Int32>] [-ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

DESCRIPTION

This command lists the actually configured Relatime, Windows and Realtime cores of the TwinCAT System.

EXAMPLES

EXAMPLE 1

```
PS> Get-RTimeCpuSettings

NetId               Windows    NonWin     RealTime  Cpu Type  Cpu      CpuFrequency
                    Cores      Cores      Cores                Family   (GHz)
-----               ---------- ---------- ---------- -------- -------- ------------
172.17.60.167.1.1   22         2          1           0        4        3793
```

Getting the CPU Settings of the local system.

PARAMETERS

-NetId

The AmsNetId of the Target system.

```
Type: AmsNetId
Parameter Sets: NetIdPort
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Address

The target address(es) where to get the CPU Settings.

This can be the RouteName, NetId, the HostName or the IPAddress.

Wildcards are permitted.

```
Type: String
Parameter Sets: AddressStr
Aliases:

Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-Session

The Session to use for the Cmdlet, must be connected to port 300, R0_IO

```
Type: ISession
Parameter Sets: Session
Aliases: InputObject

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-Timeout

The communication ADS timeout in milliseconds.

A value of 0 disables the timeout.

A value \<= 0 sets the Default (5000 ms).

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.ISession

The Session to use for the Cmdlet, must be connected to port 300, R0_IO

OUTPUTS

NOTES

6.24 Get-RTimeLatency

SYNOPSIS

Get the latency of TwinCAT Realtime Cores of the specified TwinCAT target system.

SYNTAX

NetIdPort (Default)

```
Get-RTimeLatency [-NetId <AmsNetId>] [-Core <Int32>] [-ScanTimeout <Int32>] [-Timeout <Int32>] [-
Count <Int32>]
 [-Delay <Int32>] [-NoReset] [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

AddressStr

```
Get-RTimeLatency [-Address] <String> [-Core <Int32>] [-ScanTimeout <Int32>] [-Timeout <Int32>] [-
Count <Int32>]
 [-Delay <Int32>] [-NoReset] [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

Session

```
Get-RTimeLatency -Session <ISession> [-Core <Int32>] [-ScanTimeout <Int32>] [-Timeout <Int32>] [-
Count <Int32>]
 [-Delay <Int32>] [-NoReset] [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

DESCRIPTION

This commands lists the Realtime Cores of specified TwinCAT target systems.

The values can be repeated by count parameter and a repeat delay can be set.

EXAMPLES

EXAMPLE 1

```
PS> Get-RTimeLatency

NetId                CoreId Latency    MaxLatency Limit
                            (us)         (us)
----------------     ------ ---------- ---------- -----
5.91.172.198.1.1     1      0          20         0
```

Getting the Realtime latency of all Realtime cores on the local system.

EXAMPLE 2

```
PS> Get-RTimeLatency -NetId 5.91.172.198.1.1  -core 1 -count 5 -Delay 0

NetId                CoreId Latency    MaxLatency Limit
                            (us)         (us)
----------------     ------ ---------- ---------- -----
5.91.172.198.1.1     1      0          20         0
5.91.172.198.1.1     1      0          20         0
5.91.172.198.1.1     1      0          20         0
5.91.172.198.1.1     1      0          20         0
5.91.172.198.1.1     1      0          20         0
```

Get the Realtime Latency of the System witt NetId 5.91.172.198.1.1 and CoreId 1 5 times without delay between values.

PARAMETERS

-NetId

The AmsNetId of the target system.

The Default is the local system (if left out).

```
Type: AmsNetId
Parameter Sets: NetIdPort
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Address

The address(es) where to get the Realtime latency.

This can be the RouteName, NetId, the HostName or the IPAddress.

Multiple Addresses and Wildcards are permitted.

```
Type: String
Parameter Sets: AddressStr
Aliases:

Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-Session

The Session to use for the Cmdlet, must be connected to port 300, R0_IO

```
Type: ISession
Parameter Sets: Session
Aliases: InputObject

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-Core

Specifies the CoreID of the Realtime Core.

If not specified, this Cmdlet returns all Realtime Cores.

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-ScanTimeout

Scannning timeout in milliseconds (Default 5000 ms) This is the timeout for each single ADS roundtrip used by this Cmdlet.

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: 0
Accept pipeline input: False
Accept wildcard characters: False
```

-Timeout

The communication ADS timeout in milliseconds.

A value of 0 disables the timeout.

A value \<= 0 sets the Default (5000 ms).

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
```

```
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-Count

Specifies the number request repeatations (Default is 1).

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: 1
Accept pipeline input: False
Accept wildcard characters: False
```

-Delay

Delay in Milliseconds between requests (Default is 1000ms)

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: 1000
Accept pipeline input: False
Accept wildcard characters: False
```

-NoReset

By default, the Maximum Latency of the data is reset in each polling cycle.

Switching to -NoReset remains the maximum Latency between calls.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.ISession

The Session to use for the Cmdlet, must be connected to port 300, R0_IO

OUTPUTS

NOTES

6.25 Get-RTimePerformance

SYNOPSIS

Gets the Realtime Performance of the specified system.

SYNTAX

NetIdPort (Default)

```
Get-RTimePerformance [-NetId <AmsNetId>] [-Core <Int32>] [-Count <Int32>] [-Delay <Int32>] [-
Timeout <Int32>]
 [-NoReset] [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

AddressStr

```
Get-RTimePerformance [-Address] <String> [-Core <Int32>] [-Count <Int32>] [-Delay <Int32>] [-
Timeout <Int32>]
 [-NoReset] [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

Session

```
Get-RTimePerformance -Session <ISession> [-Core <Int32>] [-Count <Int32>] [-Delay <Int32>] [-
Timeout <Int32>]
 [-NoReset] [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

DESCRIPTION

Gets the Realtime Performance of the specified TwinCAT Target systems.

The output can be filtered for specified Realtime CPUs and contains the actual CPU Latency and CPU Load.

This Cmdlet is preliminary and subject of change.

Performance Data is not supported before TwinCAT Build 4026.

EXAMPLES

EXAMPLE 1

```
PS> Get-RTimePerformance

NeId                CoreId LastDelay    MaxDelay    DelayLimit Load (%) MaxLoad
   (us)          (us)       (us)              (%)
-----               ------ ----------  ---------- ---------- -------- --------
192.168.0.2.1.1     1      0           109        0          0        80
192.168.0.2.1.1     2      0           109        0          0        80
```

Getting the Performance Data from all Realtime CPUs on the local target system.

EXAMPLE 2

```
PS> Get-RTimePerformance -core 1 -count 5 -Delay 0 -noReset

NetId               CoreId LastDelay  MaxDelay   DelayLimit Load (%) MaxLoad
```

```
                          (us)       (us)        (us)               (%)
-----                ------ ---------- ---------- ---------- -------- --------
192.168.0.2.1.1      1      0          1659       0          0        80
192.168.0.2.1.1      1      0          1659       0          0        80
192.168.0.2.1.1      1      0          1659       0          0        80
192.168.0.2.1.1      1      0          1659       0          0        80
192.168.0.2.1.1      1      0          1659       0          0        80
```

Getting the Performance Data from the local System (Core 1) 5 times as fast as possible.

The MaxDelay will not be reset on each call.

PARAMETERS

-NetId

The AmsNetId of the target system.

Uses the Local system if empty.

```
Type: AmsNetId
Parameter Sets: NetIdPort
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Address

The address(es) where to get the performance data.

This can be the RouteName, NetId, the HostName or the IPAddress.

Multiple addresses and wildcards are permitted.

```
Type: String
Parameter Sets: AddressStr
Aliases:

Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-Session

The Session to use for the Cmdlet, must be connected to port 300, R0_IO

```
Type: ISession
Parameter Sets: Session
Aliases: InputObject

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-Core

Specifies the ID of the Core where to determine the performance data.

If not specified, all Realtime CPUs will be returned.

```
Type: Int32
Parameter Sets: (All)
Aliases:
```

```
Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-Count

Specifies the number of performance requests (Default is 1) per target and CPU.

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: 1
Accept pipeline input: False
Accept wildcard characters: False
```

-Delay

Delay in Milliseconds between performance requests (Default is 1000ms)

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: 1000
Accept pipeline input: False
Accept wildcard characters: False
```

-Timeout

The communication ADS timeout in milliseconds.

A value of 0 disables the timeout.

A value \<= 0 sets the Default (5000 ms).

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-NoReset

By default, the Maximum Delay of the PerformanceData before getting new data is reset.

Switching to -NoReset remains the maximum Delay between calls.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.ISession

The Session to use for the Cmdlet, must be connected to port 300, R0_IO

OUTPUTS

NOTES

6.26 Get-TcDataType

SYNOPSIS

Get the DataTypes from a TwinCAT target system / Device.

SYNTAX

NetIdPort (Default)

```
Get-TcDataType [[-Name] <String[]>] [-NetId <AmsNetId>] -Port <Int32> [-Force]
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

Route

```
Get-TcDataType [[-Name] <String[]>] -Route <IRoute> -Port <Int32> [-Force] [-
ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

AddressStr

```
Get-TcDataType [[-Name] <String[]>] -Address <String> -Port <Int32> [-Force]
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

Session

```
Get-TcDataType [[-Name] <String[]>] -Session <ISession> [-Force] [-
ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

SessionId

```
Get-TcDataType [[-Name] <String[]>] -SessionId <Int32> [-Force] [-ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

DESCRIPTION

This Cmdlet get the DataTypes from a target system if symbolic information is provided by the device (Symbol Server running).

The DataTypes can be determined via different Providers (e.g.

ADS, MQTT, OPC, see the '-Provider' parameter.)

EXAMPLES

EXAMPLE 1

```
PS> Get-TcDataType -port 851
Name                        Size     Category   BaseType
----                        ----     --------   --------
BYTE                        1        Primitive
WORD                        2        Primitive
DINT                        4        Primitive
UDINT                       4        Primitive
DWORD                       4        Primitive
E_ByteEnum                  1        Enum       BYTE
FB_Test                     12424    Struct
PLC.PlcAppSystemInfo        256      Struct
PLC.PlcTaskSystemInfo       128      Struct
POINTER TO BYTE             4        Pointer    BYTE
R_Range                     2        Alias      INT (-6..12)
REFERENCE TO BOOL           4        Reference  BOOL
ST_SimpleStruct             166      Struct
STRING(80)                  81       String
...
```

Get the data types from the local system (Port 851):

EXAMPLE 2

```
PS> $types = Get-TcDataType -Name 'ST_*' -NetId 1.2.3.4.5.6 -Port 851
```

Gets the DataTypes with name pattern 'ST_*' from the NetId / Port address symbol server.

EXAMPLE 3

```
PS> $session = New-TcSession -Name 'CX_123456' -port 851
PS> Get-TcDataType -Session $session | where ByteSize -gt 1KB
```

Gets an ADS-Session/Connection to the target system CX_123456 on port 851, downloads the datatype information and returns all the DataTypes that are larger than 1KB of Size.

PARAMETERS

-Name

The data type name(s) to get.

Wildcards are permitted.

```
Type: String[]
Parameter Sets: (All)
Aliases:

Required: False
Position: 0
Default value: None
```

```
Accept pipeline input: False
Accept wildcard characters: True
```

-NetId

The NetID address of the target system where to load the datatypes (Local by default).

```
Type: AmsNetId
Parameter Sets: NetIdPort
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Route

The Route object where to load the datatypes from (RouteTarget.Local by default).

```
Type: IRoute
Parameter Sets: Route
Aliases: Destination

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Address

The address where to load the datatype descriptions.

This can be the RouteName, NetId, the HostName or the IPAddress.

Wildcards are permitted.

```
Type: String
Parameter Sets: AddressStr
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-Port

The Port where to load the datatype descriptions.

```
Type: Int32
Parameter Sets: NetIdPort, Route, AddressStr
Aliases:

Required: True
Position: Named
Default value: 10000
Accept pipeline input: False
Accept wildcard characters: False
```

-Session

The session object to use for datatype upload.

```
Type: ISession
Parameter Sets: Session
Aliases: InputObject
```

```
Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-SessionId

The unique session Identifier that represents the session to ose for the datatype upload.

```
Type: Int32
Parameter Sets: SessionId
Aliases: Id

Required: True
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-Force

Forces to reload the data types (forces to reload the internal cache).

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.ISession

The session object to use for datatype upload.

OUTPUTS

NOTES

6.27 Get-TcEvent

SYNOPSIS

Gets TwinCAT events from event logs on local and remote computers.

SYNTAX

```
Get-TcEvent [-MaxEvents <Int32>] [-ComputerName <String>] [-Credential <PSCredential>] [-
Level <String[]>]
 [-Source <String[]>] [-StartTime <DateTimeOffset>] [-EndTime <DateTimeOffset>] [-ID <Int32[]>]
 [-Timeout <Int32>] [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

DESCRIPTION

TwinCAT creates LogEntries in the Application log.

These most important entries for system diagnostics source from the TcSysUI Application and the TwinCAT System Service (TcSysSrv) containing the logentries from the TwinCAT Drivers.

This Get-TcEvent Cmdlet gets these events from the local system and as well from other reachable systems within the corporate network.

If you're not running PowerShell as an Administrator, you might see error messages that you cannot retrieve information about a log.

EXAMPLES

EXAMPLE 1

```
PS> get-tcevent -MaxEvents 30

   ProviderName: TcSysUi

TimeCreated                     Id LevelDisplayName Message
-----------                     -- ---------------- -------
17.08.2021 15:56:44              3 Information Process startup apps was already triggerd.
17.08.2021 15:56:44              1 Information Process startup apps after reaching RUN state.

  ProviderName: TcSysSrv

TimeCreated                     Id LevelDisplayName Message
-----------                     -- ---------------- -------
17.08.2021 15:56:44             66 Information Starting COM Server TcEventLogger !
17.08.2021 15:56:44          20000 Information TwinCAT System Message: Source: License Server; Times
tamp: 8/17/2021 3:56:44 PM 345 ms Message: license validation status is Valid(3)
17.08.2021 15:56:44             15 Information TcRTime Server started: TcRTime.
17.08.2021 15:56:44             15 Information TcRtsObjects Server started: TcRtsObjects.
17.08.2021 15:56:44             15 Information TcPlc30 Server started: TcPlc30.
17.08.2021 15:56:44             15 Information TcIo Server started: TcIo.
17.08.2021 15:56:44             69 Information Initializing COM Server TcEventLogger !
17.08.2021 15:56:44             71 Information Loading configuration of COM server TcEventLogger !

  ProviderName: TcSysUi

TimeCreated                     Id LevelDisplayName Message
-----------                     -- ---------------- -------
17.08.2021 15:56:44              2 Information Process startup apps skipped after reaching state '4'
.
17.08.2021 15:56:44              2 Information Process startup apps skipped after reaching state '6'
.

  ProviderName: TcSysSrv

TimeCreated                     Id LevelDisplayName Message
-----------                     -- ---------------- -------
17.08.2021 15:56:43             67 Information Stopping COM Server TcEventLogger !
```

```
17.08.2021 15:56:43             28 Information TCIODRIVERS Server stopped.
17.08.2021 15:56:43             28 Information TCIOECAT Server stopped.
17.08.2021 15:56:43             28 Information TCIOETH Server stopped.
17.08.2021 15:56:43             28 Information TCRTSOBJECTS Server stopped.
17.08.2021 15:56:43             28 Information TCRTIME Server stopped.
17.08.2021 15:56:43             28 Information TCIO Server stopped.
17.08.2021 15:56:43             68 Information Shutting down COM Server TcEventLogger !
17.08.2021 15:56:42             70 Information Saving configuration of COM server TcEventLogger !


  ProviderName: TcSysUi

TimeCreated                     Id LevelDisplayName Message
-----------                     -- ---------------- -------
17.08.2021 15:56:42              2 Information Process startup apps skipped after reaching state '17
'.


  ProviderName: TcSysSrv

TimeCreated                     Id LevelDisplayName Message
-----------                     -- ---------------- -------
17.08.2021 15:56:42             33 Information TwinCAT System Restart initiated from AmsNetId: 172.1
7.60.197.1.1 port 32894.


  ProviderName: TcSysUi

TimeCreated                     Id LevelDisplayName Message
-----------                     -- ---------------- -------
17.08.2021 15:50:56              2 Information Process startup apps skipped after reaching state '15
'.


  ProviderName: TcSysSrv

TimeCreated                     Id LevelDisplayName Message
-----------                     -- ---------------- -------
17.08.2021 15:50:56             66 Information Starting COM Server TcEventLogger !
17.08.2021 15:50:55             15 Information TCIODRIVERS Server started: TCIODRIVERS.
```

Gets last 30 events (all events) on the local system.

EXAMPLE 2

```
PS> get-tcevent -computerName CX_1234 -Level Critical, Error, Warning -source TcSysSrv -
StartTime([DateTime]::Now -[TimeSpan]::FromDays(1))

  ProviderName: TcSysSrv

TimeCreated                     Id LevelDisplayName Message
-----------                     -- ---------------- -------
17.08.2021 15:50:53             89 Error Error: .. checking TwinCAT Licenses!
17.08.2021 15:50:53          20000 Error TwinCAT System Message: Source: License Server; Timestamp:
8/17/2021 3:50:53 PM 145 ms Message: License Violation: License 'TC3 PLC' not found, Requested by 'P
LC1 Instance', LicenseId = {666…
```

Getting the logged errors/warnings of system 'CX_1234' of the last 24 hours and filter the events for the Event provider 'TcSysSrv' (the TwinCAT System Service)

EXAMPLE 3

```
PS> get-tcevent --StartTime '2021-08-17 15:50:55' -EndTime '2021-08-17 15:55:56' -source TcSysSrv

   ProviderName: TcSysSrv

TimeCreated                     Id LevelDisplayName Message
-----------                     -- ---------------- -------
17.08.2021 15:50:56             66 Information Starting COM Server TcEventLogger !
17.08.2021 15:50:55             15 Information TCIODRIVERS Server started: TCIODRIVERS.
17.08.2021 15:50:55             15 Information TCIOECAT Server started: TCIOECAT.
17.08.2021 15:50:55             15 Information TCIOETH Server started: TCIOETH.
17.08.2021 15:50:55             15 Information TCRTSOBJECTS Server started: TCRTSOBJECTS.
17.08.2021 15:50:55             15 Information TCRTIME Server started: TCRTIME.
17.08.2021 15:50:55             15 Information TCIO Server started: TCIO.
17.08.2021 15:50:55             69 Information Initializing COM Server TcEventLogger !
17.08.2021 15:50:55             71 Information Loading configuration of COM server TcEventLogger !
```

```
17.08.2021 15:50:55             67 Information Stopping COM Server TcEventLogger !
17.08.2021 15:50:55             68 Information Shutting down COM Server TcEventLogger !
```

Show EventLog of TwinCAT SystemService between two points in time.

PARAMETERS

-MaxEvents

Specifies the maximum number of events that are returned.

Enter an integer such as 100.

The default is to return all TwinCAT events.

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-ComputerName

Specifies the name of the computer that this cmdlet gets events from the event logs.

Type the NetBIOS name, an IP address, or the fully qualified domain name (FQDN) of the computer.

The default value is the local computer, localhost.

```
Type: String
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Credential

Specifies a user account that has permission to perform this action.

The default value is the current user.

Type a user name, such as User01 or Domain01\User01.Or, enter a PSCredential object, such as one generated by the Get-Credential cmdlet.

```
Type: PSCredential
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Level

The log level, that is determined.

By default, all EventLog content will be returned.

Allowed values are: - All - Critical - Error - Warning - Informational - Verbose

```
Type: String[]
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: All
Accept pipeline input: False
Accept wildcard characters: False
```

-Source

The log level, that is determined.

By default, all TwinCAT EventLog content will be returned.

Actually allowed values are: - All - TcSysUI - TcSysSrv - TcEventLogger - Tc3ScopeServer - 'TF3300 TwinCAT 3 Scope Server' - 'TwinCAT3 Scope Server'

```
Type: String[]
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: All
Accept pipeline input: False
Accept wildcard characters: False
```

-StartTime

Filters the EventLog to return only entries that were produced after the specified start time.

By default the start time is not set, returning Events from the beginning of the log.

```
Type: DateTimeOffset
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-EndTime

Filters the EventLog to return only entries that were produced up to the specified end time.

By default the end time is not set, returning the newest entries.

```
Type: DateTimeOffset
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-ID

Filters the EventLog to the specified EventIds.

```
Type: Int32[]
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
```

```
Accept pipeline input: False
Accept wildcard characters: False
```

-Timeout

Timeout of the Read event log operation (default: no timeout).

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

OUTPUTS

NOTES

6.28 Get-TcLicense

SYNOPSIS

Get TwinCAT License information.

SYNTAX

NetIdPort (Default)

```
Get-TcLicense [-Name <String[]>] [-OrderId <String[]>] [-NetId <AmsNetId>] [-
Status <LicenseStatus>] [-Force]
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

Route

```
Get-TcLicense [-Name <String[]>] [-OrderId <String[]>] -Route <IRoute> [-Status <LicenseStatus>] [-
Force]
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

AddressStr

```
Get-TcLicense [-Name <String[]>] [-OrderId <String[]>] -Address <String> [-
Status <LicenseStatus>] [-Force]
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

Session

```
Get-TcLicense [-Name <String[]>] [-OrderId <String[]>] -Session <ISession> [-
Status <LicenseStatus>] [-Force]
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

SessionId

```
Get-TcLicense [-Name <String[]>] [-OrderId <String[]>] -SessionId <Int32> [-
Status <LicenseStatus>] [-Force]
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

DESCRIPTION

This Cmdlet gets information about TwinCAT licenses from the target system.

To contact the target system, it must be available as actual route or the local system.

EXAMPLES

EXAMPLE 1

```
PS> Get-TcLicense
```

Get the the valid licenses from the local system.

EXAMPLE 2

```
PS> $session = New-TcSession -Route TC3TESTA1-CP67X -Port 30
PS> $session | Get-TcLicense -Status All -name *scope*

Name                        Valid ValidityCode ExpireTime   Available Used VolumeNo
----                        ----- ------------ ----------   --------- ---- --------
TC3 Scope Server                X        Valid            CPU License      0        0
TC3 Scope View Professional     X        Valid            CPU License      0        0
```

Create a session to the License Server on target 'TC3TESTA1-CP67X' and return all valid and invalid licenses that contain 'scope' in their name.

EXAMPLE 3

```
PS> Get-TcLicense -Route TC3TESTA1-CP67X -Status Valid

Name                             Valid ValidityCode ExpireTime   Available Used VolumeNo
----                             ----- ------------ ----------   --------- ---- --------
TC3 C++ / MatSim                     X        Valid            CPU License    0        0
TC3 CNC                              X        Valid            CPU License    0        0
TC3 Target For Matlab Simulink       X        Valid            CPU License    0        0
TC3 CNC Axis                         X        Valid            CPU License    0        0
TC3 Serial-Communication             X        Valid            CPU License    0        0
TC3 NC PTP Axes Pack unlimited       X        Valid            CPU License    0        0
TC3 PLC / C++ / MatSim               X        Valid            CPU License    0        0
TC3 Kinematic Transformation L4      X        Valid            CPU License    0        0
TC3 NC Camming                       X        Valid            CPU License    0        0
TC3 PLC-HMI Web                      X        Valid            CPU License    0        0
TC3 NC Flying Saw                    X        Valid            CPU License    0        0
TC3 CNC Spline                       X        Valid            CPU License    0        0
TC3 SMS-SMTP                         X        Valid            CPU License    0        0
TC3 Hydraulic Positioning            X        Valid            CPU License    0        0
TC3 Kinematic Transformation L1      X        Valid            CPU License    0        0
...
```

Connect to the License Server on target 'TC3TESTa1-CP67X' and return all valid licenses.

EXAMPLE 4

```
> Get-TcLicense -NetId 172.17.60.153.1.1 -Status Invalid | format-list


Id                : 4c256767-e6e6-4af5-bd68-9f7abad0c200
Name              : TC3 ADS
ExpireTime        : 8/17/2017 12:00:00 AM
ValidityCode      : Expired
Valid             : False
AvailableLicenses : 0
UsedLicenses      : 0
VolumeNo          : 0

Id                : 66689887-ccbd-452c-ac9a-039d997c6e66
Name              : TC3 PLC
ExpireTime        : 8/17/2017 12:00:00 AM
ValidityCode      : Expired
Valid             : False
AvailableLicenses : 0
UsedLicenses      : 0
VolumeNo          : 0

Id                : 3ff18e97-7754-401b-93fb-70544de28a13
Name              : TC3 IO
ExpireTime        : 8/17/2017 12:00:00 AM
ValidityCode      : Expired
Valid             : False
AvailableLicenses : 0
UsedLicenses      : 0
VolumeNo          : 0
```

Connect to NetId 172.17.60.153.1.1, determine all invalid licenses and format the result into a list.

EXAMPLE 5

```
> Get-TcLicense -OrderId TE*

OrderID Name                                Valid ValidityCode ExpireTime   Available   Used VolumeNo
------- ----                                ----- ------------ ----------   ---------   ---- --------
TE1400 TC3 Target For Matlab Simulink           X        Valid            CPU License      0        0
TE1500 TC3 Valve-Diagram-Editor                 X        Valid            CPU License      0        0
TE1120 TC3 XCAD Interface                       X        Valid            CPU License      0        0
TE1510 TC3 Cam-Design-Tool                      X        Valid            CPU License      0        0
TE1110 TC3 Simulation Manager                   X        Valid            CPU License      0        0
TE1111 TC3 EtherCAT Simulation                  X        Valid            CPU License      0        0
TE1410 TC3 Interface For Matlab Simulink        X        Valid            CPU License      0        0
TE1300 TC3 Scope View Professional              X        Valid            CPU License      0        0
```

Get the valid licenses from local system and filter them for OrderIds starting with TE*.

PARAMETERS

-Name

The name of the license to get.

Wildcards are permitted.

```
Type: String[]
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-OrderId

The OrderID of the license.

Wildcards are permitted.

```
Type: String[]
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-NetId

The NetID address of the target system where to load the licenses (Local by default).

```
Type: AmsNetId
Parameter Sets: NetIdPort
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Route

The Route object where to load the licenses from (RouteTarget.Local by default).

```
Type: IRoute
Parameter Sets: Route
Aliases: Destination

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-Address

The address where to load the licenses.

This can be the RouteName, NetId, the HostName or the IPAddress.

Wildcards are permitted.

```
Type: String
Parameter Sets: AddressStr
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-Session

The session object to use for license upload.

This must target port 30 (AmsPort.R0_LicenseServer).

```
Type: ISession
Parameter Sets: Session
Aliases: InputObject

Required: True
Position: Named
Default value: None
```

```
Accept pipeline input: False
Accept wildcard characters: False
```

-SessionId

The unique session Identifier that represents the session to use for the license upload.

```
Type: Int32
Parameter Sets: SessionId
Aliases: Id

Required: True
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-Status

The Status parameter selects the Licenses to return.

Available is 'Valid' (the valid licenses), 'Invalid' (the invalid licenses) and 'All' ('Valid' + 'Invalid') licenses.

The Default Value is 'All'

Possible values: None, Valid, Invalid, All

```
Type: LicenseStatus
Parameter Sets: (All)
Aliases:
Accepted values: None, Valid, Invalid, All

Required: False
Position: Named
Default value: All
Accept pipeline input: False
Accept wildcard characters: False
```

-Force

Force reading value.

This flag bypasses internal caches and the FailFastInterceptor to retry communication in every case.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.IRoute

The Route object where to load the licenses from (RouteTarget.Local by default).

OUTPUTS

NOTES

6.29 Get-TcRouterInfo

SYNOPSIS

Gets the router status information of the specified target system.

SYNTAX

NetIdPort (Default)

```
Get-TcRouterInfo [[-NetId] <AmsNetId[]>] [-Timeout <Int32>] [-Count <Int32>] [-Delay <Int32>]
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

Route

```
Get-TcRouterInfo [-InputObject] <IRoute[]> [-Timeout <Int32>] [-Count <Int32>] [-Delay <Int32>]
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

AddressStr

```
Get-TcRouterInfo [-Address] <String[]> [-Timeout <Int32>] [-Count <Int32>] [-Delay <Int32>]
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

Session

```
Get-TcRouterInfo -Session <ISession[]> [-Timeout <Int32>] [-Count <Int32>] [-Delay <Int32>]
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

SessionId

```
Get-TcRouterInfo -SessionId <Int32[]> [-Timeout <Int32>] [-Count <Int32>] [-Delay <Int32>]
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

DESCRIPTION

This Cmdlet gets status information from the specified target system.

To contact the target system, it must be available as actual route or must be the local system.

The status information contains the amout of overall router memory and the used memory.

Furthermore the number of active connections and the size of the actual router mailbox will be shown.

EXAMPLES

EXAMPLE 1

```
PS> Get-TcRouterInfo

Target  Result TotalMem(kb)  AvailMem(kb)  Ports Drivers Transports Mailbox Size(kb)  Mailbox Queue
------  ------ ------------- ------------- ----- ------- ---------- ----------------- -------------
CX_1234 Ok     32768         32759         31    4       11         0                 0
```

Get router information from the local system.

PARAMETERS

-NetId

NetId(s) of the target system.

```
Type: AmsNetId[]
Parameter Sets: NetIdPort
Aliases:

Required: False
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-InputObject

The route object where to get the Target information from..

```
Type: IRoute[]
Parameter Sets: Route
Aliases: Destination, Route

Required: True
Position: 1
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-Address

Target names/addresses.

These can consist of RouteName, NetID, HostName or IPAddress.

Wildcards are permitted.

```
Type: String[]
Parameter Sets: AddressStr
Aliases: Name

Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-Session

The Session to use for the value read.

```
Type: ISession[]
Parameter Sets: Session
Aliases:

Required: True
Position: Named
Default value: None
```

```
Accept pipeline input: True (ByPropertyName)
Accept wildcard characters: False
```

-SessionId

Specifies the Session (with unique ID) to use for the value read.

```
Type: Int32[]
Parameter Sets: SessionId
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Timeout

The communication ADS timeout in milliseconds.

A value of 0 disables the timeout.

A value \<= 0 sets the Default (5000 ms).

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: 2500
Accept pipeline input: False
Accept wildcard characters: False
```

-Count

Specifies the number of repeats for this Cmdlet (Default is 1).

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: 1
Accept pipeline input: False
Accept wildcard characters: False
```

-Delay

The Delay in Seconds between repeated requests in Seconds (Default is 1s).

A delay of 0 Seconds means as fast as possible.

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: 1
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.IRoute[]

The route object where to get the Target information from..

TwinCAT.ISession[]

The Session to use for the value read.

OUTPUTS

NOTES

6.30 Get-TcSession

SYNOPSIS

List the currently established Sessions.

SYNTAX

Default (Default)

```
Get-TcSession [-Force] [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

Id

```
Get-TcSession -Id <Int32> [-Force] [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

DESCRIPTION

This Cmdlet lists all actually Point-To-Point connections to TwinCAT Targets in form of their session representation.

Different types of Sessions can be accessed via the registered types of SessionProviders (e.g.

ADS, MQTT, OPC).

EXAMPLES

EXAMPLE 1

```
PS> Get-TcSession
```

Lists all actual initiated sessions.

PARAMETERS

-Id

Specifies the ID of the session to get.

```
Type: Int32
Parameter Sets: Id
Aliases: SessionID

Required: True
Position: Named
Default value: 0
Accept pipeline input: False
Accept wildcard characters: False
```

-Force

Forces the Cmdlet to determine also the internal used sessions.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

OUTPUTS

NOTES

6.31 Get-TcSymbol

SYNOPSIS

Get the symbols from a TwinCAT target system / Device.

SYNTAX

NetIdPort (Default)

```
Get-TcSymbol [[-Path] <String[]>] [-NetId <AmsNetId>] -Port <Int32> [-Recurse] [-Force] [-
IncludeArrays]
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

AddressStr

```
Get-TcSymbol [[-Path] <String[]>] -Address <String> -Port <Int32> [-Recurse] [-Force] [-
IncludeArrays]
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

Route

```
Get-TcSymbol [[-Path] <String[]>] -Route <IRoute> -Port <Int32> [-Recurse] [-Force] [-IncludeArrays]
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

Session

```
Get-TcSymbol [[-Path] <String[]>] -Session <ISession> [-Recurse] [-Force] [-IncludeArrays]
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

SessionId

```
Get-TcSymbol [[-Path] <String[]>] -SessionId <Int32> [-Recurse] [-Force] [-IncludeArrays]
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

NetIdPortLiteral

```
Get-TcSymbol -LiteralPath <String[]> [-NetId <AmsNetId>] -Port <Int32> [-Recurse] [-Force] [-
IncludeArrays]
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

AddressStrLiteral

```
Get-TcSymbol -LiteralPath <String[]> -Address <String> -Port <Int32> [-Recurse] [-Force] [-
IncludeArrays]
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

RouteLiteral

```
Get-TcSymbol -LiteralPath <String[]> -Route <IRoute> -Port <Int32> [-Recurse] [-Force] [-
IncludeArrays]
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

SessionLiteral

```
Get-TcSymbol -LiteralPath <String[]> -Session <ISession> [-Recurse] [-Force] [-IncludeArrays]
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

SessionIdLiteral

```
Get-TcSymbol -LiteralPath <String[]> -SessionId <Int32> [-Recurse] [-Force] [-IncludeArrays]
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

DESCRIPTION

This Cmdlet get the symbolic information from a target system if symbols are provided.

The information can be determined via different Providers (e.g.

ADS, MQTT, OPC).

EXAMPLES

EXAMPLE 1

```
PS> Get-TcSymbol -port 851
InstanceName              DataType Size InstancePath
------------              -------- ---- ------------
tc2vBool                  BOOL     1    .tc2vBool
tc2vInt                   INT      2    .tc2vInt
Constants                          0    Constants
GVL                                0    GVL
MAIN                               0    MAIN
Slow                               0    Slow
TwinCAT_SystemInfoVarList          0    TwinCAT_SystemInfoVarList
```

Get the root symbolic information from the local system (Port 851):

EXAMPLE 2

```
PS>$session = New-TcSession -NetId 1.2.3.4.5.6 -Port 851
PS>$session | Get-TcSymbol "TwinCAT_SystemInfoVarList._AppInfo" -recurse

 InstanceName       DataType             Size InstancePath
------------       --------             ---- ------------
_AppInfo           PLC.PlcAppSystemInfo 256  TwinCAT_SystemInfoVarList._AppInfo
ObjId              OTCID                4    TwinCAT_SystemInfoVarList._AppInfo.ObjId
TaskCnt            UDINT                4    TwinCAT_SystemInfoVarList._AppInfo.TaskCnt
OnlineChangeCnt    UDINT                4    TwinCAT_SystemInfoVarList._AppInfo.OnlineChangeCnt
Flags              DWORD                4    TwinCAT_SystemInfoVarList._AppInfo.Flags
AdsPort            UINT                 2    TwinCAT_SystemInfoVarList._AppInfo.AdsPort
BootDataLoaded     BOOL                 1    TwinCAT_SystemInfoVarList._AppInfo.BootDataLoaded
OldBootData        BOOL                 1    TwinCAT_SystemInfoVarList._AppInfo.OldBootData
AppTimestamp       DT                   4    TwinCAT_SystemInfoVarList._AppInfo.AppTimestamp
KeepOutputsOnBP    BOOL                 1    TwinCAT_SystemInfoVarList._AppInfo.KeepOutputsOnBP
ShutdownInProgress BOOL                 1    TwinCAT_SystemInfoVarList._AppInfo.ShutdownInProgress
LicensesPending    BOOL                 1    TwinCAT_SystemInfoVarList._AppInfo.LicensesPending
BSODOccured        BOOL                 1    TwinCAT_SystemInfoVarList._AppInfo.BSODOccured
TComSrvPtr         ITComObjectServer    4    TwinCAT_SystemInfoVarList._AppInfo.TComSrvPtr
AppName            STRING(63)           64   TwinCAT_SystemInfoVarList._AppInfo.AppName
ProjectName        STRING(63)           64   TwinCAT_SystemInfoVarList._AppInfo.ProjectName
```

Create a session to the target system '1.2.3.4.5.6' Port: 851 and get the symbol 'TwinCAT_SystemInfoVarList._AppInfo' and its subsymbols recursively.

EXAMPLE 3

```
PS> $session = New-TcSession -Name 'CX_123456' -port 851
PS> $session | Get-TcSymbol -recurse | where InstanceName -like 'Project*'

 InstanceName DataType   Size InstancePath
 ------------ --------   ---- ------------
 ProjectName  STRING(63) 64   TwinCAT_SystemInfoVarList._AppInfo.ProjectName
```

Gets an ADS-Session/Connection to the target system CX_123456 on port 851, downloads the symbol information recursively and returns all Instances where the instance name is like the pattern 'Project*'.

EXAMPLE 4

```
PS> $s = New-TcSession -port 851
PS> $s | Get-TcSymbol -path '.tc2vStructArray`[0`]','.tc2vStructArray`[1`]'

InstancePath        Category DataType        Size Static Persistant IG   IO
------------        -------- --------        ---- ------ ---------- ---- ------
.tc2vStructArray[0] Struct   ST_SimpleStruct 165  False  False      4040 117942
.tc2vStructArray[1] Struct   ST_SimpleStruct 165  False  False      4040 1179E7
```

Gets an ADS-Session/connection to the local system PLC (Port 851) and get two Array Elements.

Because the -path parameter uses the wildcard paramater '\[' and '\]' these characters must be escaped with backtick '\`'.

EXAMPLE 5

```
PS> $s = New-TcSession -port 851
PS> $s | Get-TcSymbol -literalPath '.tc2vStructArray[0]','.tc2vStructArray[1]'

InstancePath        Category DataType        Size Static Persistant IG   IO
------------        -------- --------        ---- ------ ---------- ---- ------
.tc2vStructArray[0] Struct   ST_SimpleStruct 165  False  False      4040 117942
.tc2vStructArray[1] Struct   ST_SimpleStruct 165  False  False      4040 1179E7
```

Gets an ADS-Session/connection to the local system PLC (Port 851) and gets two Array Elements with their literal path.

PARAMETERS

-Path

The instance path of the Symbol(s).

Because wildcards are permitted with path, the wildcard parameters '\[' and '\]' must be escaped with a backtick.

```
Type: String[]
Parameter Sets: NetIdPort, AddressStr, Route, Session, SessionId
Aliases:

Required: False
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-LiteralPath

The literal instance path of the symbol (Path access without wildcard).

```
Type: String[]
Parameter Sets: NetIdPortLiteral, AddressStrLiteral, RouteLiteral, SessionLiteral, SessionIdLiteral
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-NetId

The target system NetId.

```
Type: AmsNetId
Parameter Sets: NetIdPort, NetIdPortLiteral
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Route

The target system route.

```
Type: IRoute
Parameter Sets: Route, RouteLiteral
Aliases:

Required: True
Position: Named
Default value: None
```

```
Accept pipeline input: False
Accept wildcard characters: False
```

-Address

The address for the target system where to get the symbol..

This can be the RouteName, NetId, the HostName or the IPAddress.

Wildcards are permitted.

```
Type: String
Parameter Sets: AddressStr, AddressStrLiteral
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-Port

The target system port.

```
Type: Int32
Parameter Sets: NetIdPort, AddressStr, Route, NetIdPortLiteral, AddressStrLiteral, RouteLiteral
Aliases:

Required: True
Position: Named
Default value: 10000
Accept pipeline input: False
Accept wildcard characters: False
```

-Session

The session object that is used to get the symbols.

```
Type: ISession
Parameter Sets: Session, SessionLiteral
Aliases: InputObject

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-SessionId

The unique id of the session object that is used to get the symbols.

```
Type: Int32
Parameter Sets: SessionId, SessionIdLiteral
Aliases: Id

Required: True
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-Recurse

Gets the symbol recursively.

Often used in conjunction with Wildcards in -Path

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Force

Forces to reload the symbols (forces to reload the internal cache)

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-IncludeArrays

Active only in recursive mode - ignored otherwise.

This parameter forces the Cmdlet to output all symbols - even Array Elements.

Please take care because the output can be very lengthy dependent on the Size of the Array.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.ISession

The session object that is used to get the symbols.

OUTPUTS

NOTES

6.32 Get-TcTargetInfo

SYNOPSIS

Get TwinCAT Device Target information.

SYNTAX

NetIdPort (Default)

```
Get-TcTargetInfo [[-NetId] <AmsNetId[]>] [-Timeout <Int32>] [-Force] [-
ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

Route

```
Get-TcTargetInfo [-InputObject] <IRoute[]> [-Timeout <Int32>] [-Force] [-
ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

AddressStr

```
Get-TcTargetInfo [-Address] <String[]> [-Timeout <Int32>] [-Force] [-
ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

Session

```
Get-TcTargetInfo -Session <ISession[]> [-Timeout <Int32>] [-Force] [-
ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

SessionId

```
Get-TcTargetInfo -SessionId <Int32[]> [-Timeout <Int32>] [-Force] [-
ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

DESCRIPTION

This Cmdlet gets information from the specified target system.

To contact the target system, it must be available as actual route.

The information contains the TargetName, TwinCAT Version, Running Operating system, CPU Architecture and Image Information.

EXAMPLES

EXAMPLE 1

```
PS> Get-TcTargetInfo

Target          Version     Level OS   Image Device CPUArch
------          -------     ----- --   ----- ------ -------
TC3TESTA1-CP67X 3.1.4021.54 CP    Win7             IntelX86
```

Get the target information of th localSystem system.

EXAMPLE 2

```
PS> get-adsRoute | Get-TcTargetInfo

Target          Version     Level OS     Image Device CPUArch
------          -------     ----- --     ----- ------ -------
CP-15ECA0       3.1.4021.50 CP    Win7                IntelX86
TC3TESTA1-CP67X 3.1.4021.54 CP    Win7                IntelX86
```

Get the target information of the actual connected routes.

PARAMETERS

-NetId

NetId(s) of the target system.

```
Type: AmsNetId[]
Parameter Sets: NetIdPort
Aliases:

Required: False
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-InputObject

The route object where to get the Target information from..

```
Type: IRoute[]
Parameter Sets: Route
Aliases: Destination, Route

Required: True
Position: 1
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-Address

Target names/addresses.

These can consist of RouteName, NetID, HostName or IPAddress.

Wildcards are permitted.

```
Type: String[]
Parameter Sets: AddressStr
Aliases: Name

Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-Session

The Session to use for the value read.

```
Type: ISession[]
Parameter Sets: Session
Aliases:

Required: True
Position: Named
Default value: None
```

```
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-SessionId

Specifies the Session (with unique ID) to use for the value read.

```
Type: Int32[]
Parameter Sets: SessionId
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Timeout

The communication ADS timeout in milliseconds.

A value of 0 disables the timeout.

A value \<= 0 sets the Default (5000 ms).

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-Force

Force reading value.

This flag bypasses the FailFastInterceptor to retry communication in every case.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.IRoute[]

The route object where to get the Target information from..

TwinCAT.ISession[]

The Session to use for the value read.

OUTPUTS

NOTES

6.33 Get-TcVersion

SYNOPSIS

Get the TwinCAT Version of a target system.

SYNTAX

NetIdPort (Default)

```
Get-TcVersion [[-NetId] <AmsNetId[]>] [-Timeout <Int32>] [-Force] [-
ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

Route

```
Get-TcVersion [-Timeout <Int32>] [-InputObject] <IRoute[]> [-Force] [-
ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

AddressStr

```
Get-TcVersion [-Timeout <Int32>] [-Address] <String[]> [-Force] [-ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

Session

```
Get-TcVersion [-Timeout <Int32>] -Session <ISession[]> [-Force] [-ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

SessionId

```
Get-TcVersion [-Timeout <Int32>] -SessionId <Int32[]> [-Force] [-ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

DESCRIPTION

This Cmdlet gets the TwinCAT version of the specified target version and returns the version object.

EXAMPLES

EXAMPLE 1

```
PS> Get-TcVersion

Major  Minor  Build  Revision
-----  -----  -----  --------
3      1      4021   50
```

Get the TwinCAT version of the local system.

EXAMPLE 2

```
PS> Get-AdsRoute | Get-TcVersion

Major  Minor  Build  Revision
-----  -----  -----  --------
3      1      4021   50
3      1      4021   54
```

Get the TwinCAT version actual routes.

PARAMETERS

-NetId

The target address.

```
Type: AmsNetId[]
Parameter Sets: NetIdPort
Aliases:

Required: False
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Timeout

The communication ADS timeout in milliseconds.

A value of 0 disables the timeout.

A value \<= 0 sets the Default (5000 ms).

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-InputObject

The target routes where to determine the Version information.

```
Type: IRoute[]
Parameter Sets: Route
Aliases: Destination, Route

Required: True
Position: 1
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-Address

Addresses where to determine the Version information.

The Addresses can consist of NetId, IPAddress or HostName.

Wildcards are permitted.

```
Type: String[]
Parameter Sets: AddressStr
Aliases: Name

Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-Session

The Session to use for the Cmdlet.

```
Type: ISession[]
Parameter Sets: Session
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-SessionId

Specifies the Session (with unique ID) to use for the Cmdlet.

```
Type: Int32[]
Parameter Sets: SessionId
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Force

Forces reading the version.

This flag bypasses the FailFastInterceptor to retry communication in every case.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.IRoute[]

The target routes where to determine the Version information.

TwinCAT.ISession[]

The Session to use for the Cmdlet.

OUTPUTS

NOTES

6.34 New-TcSession

SYNOPSIS

Create a new session to a TwinCAT Target.

SYNTAX

NetIdPort (Default)

```
New-TcSession [[-NetId] <AmsNetId>] [-Port] <Int32> [-Force] [-ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

Route

```
New-TcSession -InputObject <IRoute> [-Port] <Int32> [-Force] [-ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

AddressStr

```
New-TcSession [-Provider <String>] [-Address] <String> [[-Port] <Int32>] [-Force]
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

DESCRIPTION

Creates a new Point-To-Point Connection to a TwinCAT Target that is represented by the returned session object.

Different types of Sessions can be accessed by the registered types of SessionProviders (e.g.

ADS, MQTT, OPC).

If using ADS as protocol, this Cmdlet is equivalent to create and connect an ADSClient.

The Address of the remote system is the AmsNetId and AmsPort.

EXAMPLES

EXAMPLE 1

```
PS> $route = Get-AdsRoute -Name "Tc3*"
PS> $session = New-TcSession -Route $route -Port 851
PS> $session
```

```
ID Address               IsConnected EstablishedAt
-- -------               ----------- -------------
5  172.17.62.105.1.1:851 True        12/12/2016 12:22:02 PM
```

Establishes a new ADS Session/Connection to the specified route destination that has the name pattern "tc3*" via port 851 (PLC1)

EXAMPLE 2

```
PS> New-TcSession -NetId '172.17.62.105.1.1' -port 851

ID Address               IsConnected EstablishedAt
-- -------               ----------- -------------
5  172.17.62.105.1.1:851 True        12/12/2016 12:22:02 PM    ///
```

Establishes a new Ads Session/Connection to the specified NetId/Port address.

EXAMPLE 3

```
PS> New-TcSession -Name 'CX_123456' -port 851

ID Address               IsConnected EstablishedAt
-- -------               ----------- -------------
5  172.17.62.105.1.1:851 True        12/12/2016 12:22:02 PM
```

Establishes a new Ads Session/Connection to the target system with the Name/HostName 'CX_123456' (Port 851).

PARAMETERS

-NetId

The NetID Address

```
Type: AmsNetId
Parameter Sets: NetIdPort
Aliases:

Required: False
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-InputObject

The route target object.

```
Type: IRoute
Parameter Sets: Route
Aliases: Destination, Route

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-Provider

Selects the session provider registered on the System (ADS by default)

```
Type: String
Parameter Sets: AddressStr
Aliases:

Required: False
Position: Named
Default value: None
```

```
Accept pipeline input: False
Accept wildcard characters: False
```

-Address

The target address of the new session.

This can be the NetId, the HostName or the IPAddress.

Wildcards are permitted.

```
Type: String
Parameter Sets: AddressStr
Aliases: Name

Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-Port

The AmsPort Address of the new session.

```
Type: Int32
Parameter Sets: NetIdPort, Route
Aliases:

Required: True
Position: 2
Default value: 10000
Accept pipeline input: False
Accept wildcard characters: False
```

```
Type: Int32
Parameter Sets: AddressStr
Aliases:

Required: False
Position: 2
Default value: 10000
Accept pipeline input: False
Accept wildcard characters: False
```

-Force

Forces to create the session independant of ReachableRoutes

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
```

```
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.IRoute

The route target object.

OUTPUTS

NOTES

6.35 Read-TcValue

SYNOPSIS

Reads values from TwinCAT devices.

SYNTAX

NetIdPortSymbol (Default)

```
Read-TcValue [-NetId <AmsNetId[]>] -Port <Int32> [-Path] <String> [-Extended] [-Force] [-
Timeout <Int32>]
 [-Encoding <Encoding>] [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

NetIdPortIndexed

```
Read-TcValue [-NetId <AmsNetId[]>] -Port <Int32> [-IndexGroup] <UInt32> [[-IndexOffset] <UInt32>]
 [-Size] <Int32> [-Force] [-Timeout <Int32>] [-Encoding <Encoding>] [-
ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

NetIdPortIndexedTyped

```
Read-TcValue [-NetId <AmsNetId[]>] -Port <Int32> [-IndexGroup] <UInt32> [[-IndexOffset] <UInt32>]
 [-ValueType] <Type> [-Size] <Int32> [-Force] [-Timeout <Int32>] [-Encoding <Encoding>]
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

RouteIndexed

```
Read-TcValue -Route <IRoute[]> -Port <Int32> [-IndexGroup] <UInt32> [[-IndexOffset] <UInt32>] [-
Size] <Int32>
 [-Force] [-Timeout <Int32>] [-Encoding <Encoding>] [-
ProgressAction <ActionPreference>] [<CommonParameters>]
```

RouteIndexedTyped

```
Read-TcValue -Route <IRoute[]> -Port <Int32> [-IndexGroup] <UInt32> [[-IndexOffset] <UInt32>]
 [-ValueType] <Type> [[-Size] <Int32>] [-Force] [-Timeout <Int32>] [-Encoding <Encoding>]
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

RouteSymbol

```
Read-TcValue -Route <IRoute[]> -Port <Int32> [-Path] <String> [-Extended] [-Force] [-
Timeout <Int32>]
 [-Encoding <Encoding>] [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

AddressIndexed

```
Read-TcValue -Address <String[]> -Port <Int32> [-IndexGroup] <UInt32> [[-IndexOffset] <UInt32>] [-
Size] <Int32>
 [-Force] [-Timeout <Int32>] [-Encoding <Encoding>] [-
ProgressAction <ActionPreference>] [<CommonParameters>]
```

AddressIndexedTyped

```
Read-TcValue -Address <String[]> -Port <Int32> [-IndexGroup] <UInt32> [[-IndexOffset] <UInt32>]
 [-ValueType] <Type> [[-Size] <Int32>] [-Force] [-Timeout <Int32>] [-Encoding <Encoding>]
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

AddressSymbol

```
Read-TcValue -Address <String[]> -Port <Int32> [-Path] <String> [-Extended] [-Force] [-
Timeout <Int32>]
 [-Encoding <Encoding>] [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

SessionIndexed

```
Read-TcValue -Session <ISession[]> [-IndexGroup] <UInt32> [[-IndexOffset] <UInt32>] [-
Size] <Int32> [-Force]
 [-Timeout <Int32>] [-Encoding <Encoding>] [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

SessionIndexedTyped

```
Read-TcValue -Session <ISession[]> [-IndexGroup] <UInt32> [[-IndexOffset] <UInt32>] [-
ValueType] <Type>
 [[-Size] <Int32>] [-Force] [-Timeout <Int32>] [-Encoding <Encoding>] [-
ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

SessionSymbol

```
Read-TcValue -Session <ISession[]> [-Path] <String> [-Extended] [-Force] [-Timeout <Int32>]
 [-Encoding <Encoding>] [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

SessionIdIndexed

```
Read-TcValue -SessionId <Int32[]> [-IndexGroup] <UInt32> [[-IndexOffset] <UInt32>] [-
Size] <Int32> [-Force]
 [-Timeout <Int32>] [-Encoding <Encoding>] [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

SessionIdIndexedTyped

```
Read-TcValue -SessionId <Int32[]> [-IndexGroup] <UInt32> [[-IndexOffset] <UInt32>] [-
ValueType] <Type>
 [[-Size] <Int32>] [-Force] [-Timeout <Int32>] [-Encoding <Encoding>] [-
ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

SessionIdSymbol

```
Read-TcValue -SessionId <Int32[]> [-Path] <String> [-Extended] [-Force] [-Timeout <Int32>]
 [-Encoding <Encoding>] [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

InputObject

```
Read-TcValue [-InputObject] <ISymbol> [-Force] [-Timeout <Int32>] [-Encoding <Encoding>]
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

DESCRIPTION

This Cmdlet read values from TwinCAT Devices.

The devices can be accessed via different ValueProviders.

EXAMPLES

EXAMPLE 1

```
PS> $session = New-TcSession -NetId '1.2.3.4.5.6' -Port 851
PS> $symbol = $session | get-TcSymbol -Path 'TwinCAT_SystemInfoVarList._AppInfo.ProjectName'
PS> $symbol | Read-TcValue

ADS_DynSymbols
```

Create an ADS Session/Connection, determine the 'ProjectName' Symbol from the running PLC Project, read the current value of the symbol and print it to the console.

EXAMPLE 2

```
PS> Read-TcValue -IndexGroup 0x4040 -IndexOffset 0x1247a8 -NetId 172.17.62.105.1.1 -port 851 -
size 0xff | format-hex


           00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000   41 44 53 5F 44 79 6E 53 79 6D 62 6F 6C 73 00 00  ADS_DynSymbols..
00000010   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000020   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000030   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000040   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000050   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000060   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000070   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000080   11 00 01 01 A0 86 01 00 14 00 5E 01 21 C2 15 00  .... ?....^.!A..
00000090   00 7F F1 57 3B 83 6C 07 1E 00 00 00 00 00 00 00  .ôñW;?l.........
000000A0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000B0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000C0   41 44 53 5F 44 79 6E 53 79 6D 62 6F 6C 73 5F 50  ADS_DynSymbols_P
000000D0   6C 63 54 61 73 6B 00 00 00 00 00 00 00 00 00 00  lcTask..........
000000E0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000F0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00      ................
```

Reads 256 Bytes via IndexGroup/IndexOffset from the specified target system and prints the out formatted as hexdump.

EXAMPLE 3

```
PS> Read-TcValue -session $session -IndexGroup 0x4040 -IndexOffset 0x1247A8 -ValueType String
ADS_DynSymbols
```

Reads a string typed value from IndexGroup / IndexOffset.

In this example the ProjectName of the running PLC Project resides at that ProcessImage Address.

EXAMPLE 4

```
PS> $route = Get-AdsRoute -Name 'CX-123456'
PS> $session = $route | New-TcSession -Port 851
PS> $handle = $session | Send-TcReadWrite -IndexGroup SymbolHandleByName -WriteValue "GVL.vgInt" -
ReadType Int32 -force
PS> $session | Read-TcValue -IndexGroup SymbolValueByHandle -IndexOffset $handle -ValueType Int16
42
```

Create a session to the PLC (Port 851) of a target system, determine the SymbolHandle by InstancePath and use this handle to read its 'Int16' Value (INT on PLC System).

PARAMETERS

-NetId

The NetId part of the AmsAddress for the value read.

```
Type: AmsNetId[]
Parameter Sets: NetIdPortSymbol, NetIdPortIndexed, NetIdPortIndexedTyped
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Route

Specifies the target system(s) to read value from.

```
Type: IRoute[]
Parameter Sets: RouteIndexed, RouteIndexedTyped, RouteSymbol
Aliases: Destination

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Address

The Address(es) of the system(s) where to read the value.

The Address can consist of NetId, IPAddress or HostName.

Wildcards are permitted.

```
Type: String[]
Parameter Sets: AddressIndexed, AddressIndexedTyped, AddressSymbol
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-Session

The Session to use for the value read.

```
Type: ISession[]
Parameter Sets: SessionIndexed, SessionIndexedTyped, SessionSymbol
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByPropertyName, ByValue)
Accept wildcard characters: False
```

-SessionId

Specifies the Session (with unique ID) to use for the value read.

```
Type: Int32[]
Parameter Sets: SessionIdIndexed, SessionIdIndexedTyped, SessionIdSymbol
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Port

The address Port to use for the value read.

```
Type: Int32
Parameter Sets: NetIdPortSymbol, NetIdPortIndexed, NetIdPortIndexedTyped, RouteIndexed, RouteIndexed
Typed, RouteSymbol, AddressIndexed, AddressIndexedTyped, AddressSymbol
Aliases:

Required: True
Position: Named
Default value: 10000
Accept pipeline input: False
Accept wildcard characters: False
```

-IndexGroup

The IndexGroup of the Symbol to read from target system.

Only for IndexGroup/IndexOffset access.

```
Type: UInt32
Parameter Sets: NetIdPortIndexed, NetIdPortIndexedTyped, RouteIndexed, RouteIndexedTyped, AddressInd
exed, AddressIndexedTyped, SessionIndexed, SessionIndexedTyped, SessionIdIndexed, SessionIdIndexedTy
ped
Aliases: IG

Required: True
Position: 1
Default value: 0
Accept pipeline input: False
Accept wildcard characters: False
```

-IndexOffset

The IndexOffset of the Symbol to read from the target system.

Only for IndexGroup/IndexOffset access.

```
Type: UInt32
Parameter Sets: NetIdPortIndexed, NetIdPortIndexedTyped, RouteIndexed, RouteIndexedTyped, AddressInd
exed, AddressIndexedTyped, SessionIndexed, SessionIndexedTyped, SessionIdIndexed, SessionIdIndexedTy
ped
Aliases: IO

Required: False
Position: 2
Default value: 0
Accept pipeline input: False
Accept wildcard characters: False
```

-ValueType

The dataType of the Value for a 'ReadAny' access.

Only usable with IndexGroup/IndexOffset access.

```
Type: Type
Parameter Sets: NetIdPortIndexedTyped, RouteIndexedTyped, AddressIndexedTyped, SessionIndexedTyped,
SessionIdIndexedTyped
Aliases: Type, ReadType

Required: True
```

```
Position: 3
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Size

The 'Size' of Value (in bytes) to read.

```
Type: Int32
Parameter Sets: NetIdPortIndexed, NetIdPortIndexedTyped, RouteIndexed, AddressIndexed, SessionIndexe
d, SessionIdIndexed
Aliases: ReadSize, Length

Required: True
Position: 3
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

```
Type: Int32
Parameter Sets: RouteIndexedTyped, AddressIndexedTyped, SessionIndexedTyped, SessionIdIndexedTyped
Aliases: ReadSize, Length

Required: False
Position: 3
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-Path

The instance path of the symbol to read (symbolic access).

This parameter supports wildcards.

```
Type: String
Parameter Sets: NetIdPortSymbol, RouteSymbol, AddressSymbol, SessionSymbol, SessionIdSymbol
Aliases:

Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-InputObject

The symbol object to read value from.

```
Type: ISymbol
Parameter Sets: InputObject
Aliases: Symbol

Required: True
Position: 1
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-Extended

Switch on 'ExtendedMode', what means that primitive values are not resolved to their primitive managed (powershell) counterparts, but still contain rich metadata as DynamicValues.

```
Type: SwitchParameter
Parameter Sets: NetIdPortSymbol, RouteSymbol, AddressSymbol, SessionSymbol, SessionIdSymbol
Aliases: FullMetadata

Required: False
Position: Named
Default value: False
```

```
Accept pipeline input: False
Accept wildcard characters: False
```

-Force

Force reading value.

This flag bypasses the FailFastInterceptor to retry communication in every case.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Timeout

The communication ADS timeout in milliseconds.

A value of 0 disables the timeout.

A value \<= 0 sets the Default (5000 ms).

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-Encoding

Specifies the Encoding for strings.

The DefaultEncoding is ANSI with actual code page.

```
Type: Encoding
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: System.Text.UTF8Encoding+UTF8EncodingSealed
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.ISession[]

The Session to use for the value read.

TwinCAT.TypeSystem.ISymbol

The symbol object to read value from.

OUTPUTS

NOTES

6.36 Register-AdsHandle

SYNOPSIS

Registers and returns a symbol handle.

SYNTAX

NetIdPortSymbol (Default)

```
Register-AdsHandle [-NetId <AmsNetId>] -Port <Int32> [-Path] <String[]> [-
ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

RouteSymbol

```
Register-AdsHandle -Route <IRoute> -Port <Int32> [-Path] <String[]> [-
ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

AddressSymbol

```
Register-AdsHandle -Address <String> -Port <Int32> [-Path] <String[]> [-
ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

SessionSymbol

```
Register-AdsHandle -Session <ISession> [-Path] <String[]> [-ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

SessionIdSymbol

```
Register-AdsHandle -SessionId <Int32> [-Path] <String[]> [-ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

InputObject

```
Register-AdsHandle [-InputObject] <ISymbol[]> [-
ProgressAction <ActionPreference>] [<CommonParameters>]
```

DESCRIPTION

This Cmdlet registers a symbol handle at the connected system.

The handle is returned as AdsHandleInfo.

EXAMPLES

EXAMPLE 1

```
PS> $session = New-TcSession -NetId '1.2.3.4.5.6' -Port 851
PS> $handleInfo = $session | get-AdsHandle -Path 'TwinCAT_SystemInfoVarList._AppInfo.ProjectName'
PS> $handleInfo = register-AdsHandle -Path 'TwinCAT_SystemInfoVarList._AppInfo.ProjectName' -
Session $s
PS> $handleInfo

InstancePath                                    Result Handle
------------                                    ------  ------
TwinCAT_SystemInfoVarList._AppInfo.ProjectName NoError 0x428000FC (1115685116)

PS> Read-TcValue -Session $session -IndexGroup SymbolValueByHandle -IndexOffset $handleInfo.Handle -
Type String

MyProject

PS> $handle | Unregister-AdsHandle -Session $session
PS> $session | Close-tcsession
```

Opens a new device session, registers a Symbol Handle to the ProjectName of the running PLC Project, Reads the value by handle unregisters the handle and closes the session again.

PARAMETERS

-NetId

The NetId address of the Target system

```
Type: AmsNetId
Parameter Sets: NetIdPortSymbol
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Route

Specifies the route of the target system.

```
Type: IRoute
Parameter Sets: RouteSymbol
Aliases: Destination

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Address

The Address of the target system where to register the symbol handle.

The Address can consist of RouteName, NetId, IPAddress or HostName.

Wildcards are permitted.

```
Type: String
Parameter Sets: AddressSymbol
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-Session

The Session to use (instead of addressing the target system).

```
Type: ISession
Parameter Sets: SessionSymbol
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByPropertyName, ByValue)
Accept wildcard characters: False
```

-SessionId

Specifies the Session (with unique ID) to use instead of specifying the target address.

```
Type: Int32
Parameter Sets: SessionIdSymbol
Aliases:

Required: True
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-Port

The address Port to use (always in combination with the NetId).

ArgumentCompleter is supported.

```
Type: Int32
Parameter Sets: NetIdPortSymbol, RouteSymbol, AddressSymbol
Aliases:

Required: True
Position: Named
Default value: 10000
Accept pipeline input: False
Accept wildcard characters: False
```

-Path

The instance path to the symbol.

```
Type: String[]
Parameter Sets: NetIdPortSymbol, RouteSymbol, AddressSymbol, SessionSymbol, SessionIdSymbol
Aliases:

Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-InputObject

The symbol object.

```
Type: ISymbol[]
Parameter Sets: InputObject
Aliases: Symbol

Required: True
Position: 1
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.ISession

The Session to use (instead of addressing the target system).

TwinCAT.TypeSystem.ISymbol[]

The symbol object.

OUTPUTS

NOTES

6.37 Register-AdsNatRoute

SYNOPSIS

Changes an standard Route to an AmsNAT route on the target system (obsolete).

SYNTAX

Name (Default)

```
Register-AdsNatRoute [-Name] <String> -NATNetId <AmsNetId> [-Destination <String>] [-Quiet] [-Force]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

NetId

```
Register-AdsNatRoute [-NetId] <AmsNetId> -NATNetId <AmsNetId> [-Destination <String>] [-Quiet] [-
Force]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

DESCRIPTION

This Cmdlet Changes an standard Route to an AmsNAT route on the target system.

The route must be preexisting and the cmdlet adds the RemoteNetId/AmsNAT information to the StaticRoutes.xml of the destination system.

Afterwards the destination system needs a TwinCAT Restart.

For TwinCAT Versions \>= 3.1.4024.11 (or newer), the Add-AdsRoute Cmdlet should be used with the -NAT Parameter as Replacement.

Therefore, this 'Register-AdsNatRoute' Cmdlet is classified as 'obsolete' and of limitied use and could be removed in future.

EXAMPLES

EXAMPLE 1

```
PS> Register-AdsNatRoute -Name MyRoute -NATNetId 1.2.3.4.2.2
```

Adds an AmsNAT address translation to the existing route 'MyRoute' on the local system (e.g.

from '1.2.3.4.1.1' to '1.2.3.4.2.2').

EXAMPLE 2

```
PS> Register-AdsNatRoute -NetId 1.2.3.4.1.1 -NATNetId 1.2.3.4.2.2 -Destination CX_1234
```

Adds an AmsNAT address translation to the existing route with NetId '1.2.3.4.1.1' to NATNetId '1.2.3.4.2.2' on System 'CX_1234'.

PARAMETERS

-Name

The Name of the Route where to add an AmsNAT entry.

```
Type: String
Parameter Sets: Name
Aliases:

Required: True
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-NetId

The NetID which specifies the existing route where to add an AmsNAT entry.

This NetId becomes the 'RemoteNetId' afterwards.

```
Type: AmsNetId
Parameter Sets: NetId
Aliases:

Required: True
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-NATNetId

The NATNetId (the local representation of the remote system).

```
Type: AmsNetId
Parameter Sets: (All)
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Destination

The Destination system, where the AmsNAT translation is added.

This Parameter allows RouteName, AmsNetId, IPAddress or HostName

```
Type: String
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Quiet

The Quiet parameter suppresses the 'ShouldProcess' messsage and the Cmdlet will be processed without further user confirmation.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: Silent

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Force

Forces the command (no confirmation, Resets the FailFastHandler)

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Confirm

Prompts you for confirmation before running the cmdlet.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: cf

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-WhatIf

Shows what would happen if the cmdlet runs.

The cmdlet is not run.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: wi

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

OUTPUTS

NOTES

6.38 Remove-AdsRoute

SYNOPSIS

Remove an ADS Route.

SYNTAX

Address (Default)

```
Remove-AdsRoute [-Destination <IRoute>] [-Address] <String[]> [-Quiet] [-Force] [-
Credentials <PSCredential>]
 [-Mode <RouteChangeMode>] [-ProgressAction <ActionPreference>] [-WhatIf] [-
Confirm] [<CommonParameters>]
```

NetId

```
Remove-AdsRoute [-Destination <IRoute>] [-NetId] <AmsNetId> [-Quiet] [-Force] [-
Credentials <PSCredential>]
 [-Mode <RouteChangeMode>] [-ProgressAction <ActionPreference>] [-WhatIf] [-
Confirm] [<CommonParameters>]
```

Route

```
Remove-AdsRoute [-Destination <IRoute>] [-InputObject] <RouteTargetCollection> [-Quiet] [-Force]
 [-Credentials <PSCredential>] [-Mode <RouteChangeMode>] [-ProgressAction <ActionPreference>] [-
WhatIf]
 [-Confirm] [<CommonParameters>]
```

DESCRIPTION

Removes static or temporary routes from the local system or from remote systems.

If access is available, the route is removed on both endpoints of the Route.

EXAMPLES

EXAMPLE 1

```
PS> Get-AdsRoute

Name            NetId             Address       Sub TcVersion RTSystem
----            -----             -------       --- --------- --------
CP-15ECA0       172.17.62.128.1.1 172.17.62.178     0.0       Unknown
TC3TESTA1-CP67X 172.17.62.105.1.1 172.17.62.105     0.0       Unknown

PS> Remove-AdsRoute -Name "CP-15ECA0","TC3TESTA1*"
```

Removes the Routes "CP-15ECA0" and "TC3TESTA1-CP67X" from the local system.

EXAMPLE 2

```
PS> Get-AdsRoute | Remove-AdsRoute -silent
```

Removes all registered routes from the local system.

PARAMETERS

-Destination

The destination address, where to Remove the specified route.

This can be the NetId, the HostName or the IPAddress

```
Type: IRoute
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Address

The address for the ADS route to remove.

This can be the NetId, the HostName or the IPAddress.

Wildcards are permitted.

```
Type: String[]
Parameter Sets: Address
Aliases: Name

Required: True
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-NetId

The NetID of the route to remove.

```
Type: AmsNetId
Parameter Sets: NetId
Aliases:

Required: True
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-InputObject

A collection of routes to remove (Pipeline support).

```
Type: RouteTargetCollection
Parameter Sets: Route
Aliases:

Required: True
Position: 0
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-Quiet

The Quiet parameter suppresses the 'ShouldProcess' messsage and the Cmdlet will be processed without further user confirmation.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: Silent

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Force

Forces the command (no confirmation, Resets the FailFastHandler)

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Credentials

Destination system route credentials (only if removing remotely).

```
Type: PSCredential
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: System.Management.Automation.PSCredential
Accept pipeline input: False
Accept wildcard characters: False
```

-Mode

The Mode parameter indicates if the Router should be deleted on on side or on both sides (default).

Possible values: Single, Both

```
Type: RouteChangeMode
Parameter Sets: (All)
Aliases:
Accepted values: Single, Both

Required: False
Position: Named
Default value: Both
Accept pipeline input: False
Accept wildcard characters: False
```

-Confirm

Prompts you for confirmation before running the cmdlet.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: cf

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-WhatIf

Shows what would happen if the cmdlet runs.

The cmdlet is not run.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: wi

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.RouteTargetCollection

A collection of routes to remove (Pipeline support).

OUTPUTS

NOTES

6.39 Remove-MqttRoute

SYNOPSIS

Remove a MQQT Route.

SYNTAX

Address (Default)

```
Remove-MqttRoute [-Address] <String> [[-Port] <Int32>] [-Destination <String>] [-Quiet] [-Force]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

RouteInfo

```
Remove-MqttRoute -InputObject <MqttRoute[]> [-Destination <String>] [-Quiet] [-Force]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

DESCRIPTION

Removes a MQQT Route of the specified system.

EXAMPLES

EXAMPLE 1

```
PS> Remove-MqttRoute -address 1.2.3.4 -port 42 -Destination CX_1234
```

Removes the MQTT route on the destination System 'CX_1234' to MQTT Broker with Address '1.2.3.4' and Port '42'.

PARAMETERS

-Address

The IPAddress or HostName of the Mqtt broker system to remove.

```
Type: String
Parameter Sets: Address
Aliases:

Required: True
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Port

The TCP/IP port specification.

```
Type: Int32
Parameter Sets: Address
Aliases:
```

```
Required: False
Position: 1
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-InputObject

The Mqtt routes to remove.

```
Type: MqttRoute[]
Parameter Sets: RouteInfo
Aliases: MqttRoute

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-Destination

The destination address, where to Remove the specified Mqtt route.

This can be the NetId, the HostName or the IPAddress

```
Type: String
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Quiet

The Quiet parameter suppresses the 'ShouldProcess' messsage and the Cmdlet will be processed without further user confirmation.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: Silent

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Force

Forces the command (no confirmation, Resets the FailFastHandler)

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Confirm

Prompts you for confirmation before running the cmdlet.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: cf

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-WhatIf

Shows what would happen if the cmdlet runs.

The cmdlet is not run.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: wi

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.SystemService.MqttRoute[]

The Mqtt routes to remove.

OUTPUTS

NOTES

6.40 Reset-IoFreeRun

SYNOPSIS

Resets the IO FreeRun state on the specified target.

SYNTAX

NetIdPortList (Default)

```
Reset-IoFreeRun [-Quiet] [-Timeout <Int32>] [-ProgressAction <ActionPreference>] [-WhatIf] [-
Confirm]
 [<CommonParameters>]
```

NetIdPort

```
Reset-IoFreeRun -NetId <AmsNetId> [-Quiet] [-Timeout <Int32>] [-
ProgressAction <ActionPreference>] [-WhatIf]
 [-Confirm] [<CommonParameters>]
```

AddressStr

```
Reset-IoFreeRun [[-Address] <String>] [-Quiet] [-Timeout <Int32>] [-
ProgressAction <ActionPreference>]
 [-WhatIf] [-Confirm] [<CommonParameters>]
```

Session

```
Reset-IoFreeRun -Session <ISession> [-Quiet] [-Timeout <Int32>] [-
ProgressAction <ActionPreference>] [-WhatIf]
 [-Confirm] [<CommonParameters>]
```

DESCRIPTION

Resets the IO FreeRun state on the specified target if its in Config state.

If its not in config state an error will be produced

EXAMPLES

EXAMPLE 1

```
PS> Reset-IOFreeRun -NetId 5.62.192.46.1.1
```

Reset the IO Free Run mode of the target system with NetID 5.62.192.46.1.1.

PARAMETERS

-NetId

NetId of the target system where to Reset the FreeRun.

```
Type: AmsNetId
Parameter Sets: NetIdPort
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Address

The address(es) where to set the free run.

This can be the RouteName, NetId, the HostName or the IPAddress.

Wildcards are permitted.

```
Type: String
Parameter Sets: AddressStr
Aliases:
```

```
Required: False
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-Session

The Session to use for the Cmdlet, must be connected to port 300, R0_IO

```
Type: ISession
Parameter Sets: Session
Aliases: InputObject

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-Quiet

The Quiet parameter suppresses the 'ShouldProcess' messsage and the Cmdlet will be processed without further user confirmation.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: Silent

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Timeout

The communication ADS timeout in milliseconds.

A value of 0 disables the timeout.

A value \<= 0 sets the Default (5000 ms).

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-Confirm

Prompts you for confirmation before running the cmdlet.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: cf

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-WhatIf

Shows what would happen if the cmdlet runs.

The cmdlet is not run.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: wi

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.ISession

The Session to use for the Cmdlet, must be connected to port 300, R0_IO

OUTPUTS

NOTES

6.41 Restart-AdsComputer

SYNOPSIS

Restarts ("reboots") the operating system on local and remote TwinCAT computers.

SYNTAX

NetIdPort (Default)

```
Restart-AdsComputer [-Wait] [-WaitTimeout <Int32>] [-Delay <Int32>] [[-NetId] <AmsNetId[]>] [-
Timeout <Int32>]
 [-Quiet] [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

Route

```
Restart-AdsComputer [-Wait] [-WaitTimeout <Int32>] [-Delay <Int32>] [-InputObject] <IRoute[]>
 [-Timeout <Int32>] [-Quiet] [-ProgressAction <ActionPreference>] [-WhatIf] [-
Confirm] [<CommonParameters>]
```

AddressStr

```
Restart-AdsComputer [-Wait] [-WaitTimeout <Int32>] [-Delay <Int32>] [-Address] <String[]> [-
Timeout <Int32>]
 [-Quiet] [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

Session

```
Restart-AdsComputer [-Wait] [-WaitTimeout <Int32>] [-Delay <Int32>] -Session <ISession[]> [-
Timeout <Int32>]
 [-Quiet] [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

SessionId

```
Restart-AdsComputer [-Wait] [-WaitTimeout <Int32>] [-Delay <Int32>] -SessionId <Int32[]> [-
Timeout <Int32>]
 [-Quiet] [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

DESCRIPTION

The Restart-AdsComputer cmdlet restarts the operating system on the local and remote TwinCAT computers.

You can use the parameters of Restart-AdsComputer to specify available ADS target systems to restart.

The restart can be done delayed if Users are logged into the target system (existant Session UI) or forced immediatly.

You can wait for the restart to complete before you run the next command and specify a waiting time-out.

This feature makes it practical to use Restart-AdsComputer in scripts and functions.

EXAMPLES

EXAMPLE 1

```
PS> Restart-AdsComputer CX_1111,CX_2222 -force
```

Restarts the computers CX_1111 and CX_2222 immediatly without warning logged in users on the target system and returns immediatly without waiting the finished reboot.

The Force parameter supresses the ShouldProcess query.

EXAMPLE 2

```
PS> Restart-AdsComputer -netId '1.2.3.4.1.1', '1.2.3.5.1.1' -Delay 30 -Wait -WaitTimeout 240 -force
```

Restarts the TwinCAT targets '1.2.3.4.1.1' and 1.2.3.5.1.1' without ShouldProcess query after a delay of 30 Seconds (if a user is logged in) and waits for the reboot before continuing.

The Wait timeout is set to 240 Seconds.

PARAMETERS

-Wait

Activates a wait until the rebooted system is available again.

The parameter -WaitTimeout specifies how long the script waits for the reboot.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
```

```
Accept pipeline input: False
Accept wildcard characters: False
```

-WaitTimeout

The Wait time for the reboot of the target system (default 120 Seconds).

This parameter is used in conjunction with the -Wait parameter.

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: 120
Accept pipeline input: False
Accept wildcard characters: False
```

-Delay

The delay time for the reboot/shutdown of the target system(s) in seconds.

The default is 120 Seconds.

If no user is logged in the target system the reboot/shotdown occurs always immediatly without warning.

In case of a log in, a Warning message with countdown will be presented to the user.

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: 0
Accept pipeline input: False
Accept wildcard characters: False
```

-NetId

NetId(s) of the target system.

```
Type: AmsNetId[]
Parameter Sets: NetIdPort
Aliases:

Required: False
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-InputObject

The ADS routes to shutdown/reboot.

```
Type: IRoute[]
Parameter Sets: Route
Aliases: Destination, Route

Required: True
Position: 1
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-Address

Target names/addresses.

These can consist of RouteName, NetID, HostName or IPAddress.

This parameter supports wildcards.

```
Type: String[]
Parameter Sets: AddressStr
Aliases: Name

Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-Session

The Session(s) to use for addressing the target systems.

```
Type: ISession[]
Parameter Sets: Session
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-SessionId

Specifies the Sessions (with unique ID) to use for addressing the target systems.

```
Type: Int32[]
Parameter Sets: SessionId
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Timeout

The communication ADS timeout in milliseconds.

A value of 0 disables the timeout.

A value \<= 0 sets the Default (5000 ms).

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-Quiet

The Quiet parameter suppresses the 'ShouldProcess' messsage and the Cmdlet will be processed without further user confirmation.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: Silent

Required: False
Position: Named
Default value: False
```

```
Accept pipeline input: False
Accept wildcard characters: False
```

-Confirm

Prompts you for confirmation before running the cmdlet.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: cf

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-WhatIf

Shows what would happen if the cmdlet runs.

The cmdlet is not run.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: wi

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.IRoute[]

The ADS routes to shutdown/reboot.

TwinCAT.ISession[]

The Session(s) to use for addressing the target systems.

OUTPUTS

NOTES

6.42 Send-TcReadWrite

SYNOPSIS

Sends a Read/Write access to ADS Server / TwinCAT Devices.

SYNTAX

NetIdPortIndexed (Default)

```
Send-TcReadWrite -IndexGroup <UInt32> [-IndexOffset <UInt32>] [-WriteValue <Object>] [-
WriteLength <Int32>]
 [-ReadLength <Int32>] [-ReadType <Type>] [-NetId <AmsNetId[]>] -Port <Int32> [-
Encoding <Encoding>] [-Async]
 [-Timeout <Int32>] [-Force] [-Quiet] [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm]
 [<CommonParameters>]
```

AddressIndexed

```
Send-TcReadWrite -IndexGroup <UInt32> [-IndexOffset <UInt32>] [-WriteValue <Object>] [-
WriteLength <Int32>]
 [-ReadLength <Int32>] [-ReadType <Type>] -Address <String[]> -Port <Int32> [-
Encoding <Encoding>] [-Async]
 [-Timeout <Int32>] [-Force] [-Quiet] [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm]
 [<CommonParameters>]
```

RouteIndexed

```
Send-TcReadWrite -IndexGroup <UInt32> [-IndexOffset <UInt32>] [-WriteValue <Object>] [-
WriteLength <Int32>]
 [-ReadLength <Int32>] [-ReadType <Type>] -Route <IRoute[]> -Port <Int32> [-Encoding <Encoding>] [-
Async]
 [-Timeout <Int32>] [-Force] [-Quiet] [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm]
 [<CommonParameters>]
```

SessionIndexed

```
Send-TcReadWrite -IndexGroup <UInt32> [-IndexOffset <UInt32>] [-WriteValue <Object>] [-
WriteLength <Int32>]
 [-ReadLength <Int32>] [-ReadType <Type>] -Session <ISession[]> [-Encoding <Encoding>] [-Async]
 [-Timeout <Int32>] [-Force] [-Quiet] [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm]
 [<CommonParameters>]
```

SessionIdIndexed

```
Send-TcReadWrite -IndexGroup <UInt32> [-IndexOffset <UInt32>] [-WriteValue <Object>] [-
WriteLength <Int32>]
 [-ReadLength <Int32>] [-ReadType <Type>] -SessionId <Int32[]> [-Encoding <Encoding>] [-Async]
 [-Timeout <Int32>] [-Force] [-Quiet] [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm]
 [<CommonParameters>]
```

DESCRIPTION

This Cmdlet Read/Writes values from/to TwinCAT Devices and works with different ValueProviders.

Because this is a low level data access, only IndexGroup/IndexOffset addressing is available.

IMPORTANT: Sending Read/Write commands should be done with highest care because it could destabilize the TwinCAT System when the write operation is not addressed properly.

To enhance secure operation, the user is enforced to use Length parameters in conjunction with the in/out values which will be checked by the Cmdlet.

The highest attention should also be taken with the IndexGroup/IndexOffset because that represents the Address in the Process Image and cannot be checked by principle.

To prevent that process image overwrites important data by accident please use the -WhatIf and -Confirm parameters whenever it is appropriate and inform about the $ConfirmPreference settings (PS\> get-help about_Preference_Variables) before usage of the Send-TcReadWrite Cmdlet.

EXAMPLES

EXAMPLE 1

```
PS> Send-TcReadWrite -NetId 1.2.3.4.5.6 -Port 851 -IndexGroup SymbolValueByName -IndexOffset 0 -
WriteValue "TwinCAT_SystemInfoVarList._AppInfo.ProjectName" -ReadType string -ReadLength 1024

ReadWrite access of process image on target '1.2.3.4.5.6:851':
Start ReadWrite operation WriteData: 'TwinCAT_SystemInfoVarList._AppInfo.ProjectName'(IG:0xf004,IO:0
x0000,Len:47),
Read: Type 'System.String' (Len:'1024) on target '1.2.3.4.5.6:851'?
[Y] Yes[A] Yes to All[N] No[L] No to All[S] Suspend[?] Help(default is "Y"): y
ADS_DynSymbols
```

Sends a Read/Write request with index group 0xf004 (SymbolValueByName) and offset 0.

The write data will be initialized with the project symbol path and an returned (read) string (Default encoded) returned.

EXAMPLE 2

```
PS> Send-TcReadWrite -NetId 1.2.3.4.5.6 -Port 851 -IndexGroup SymbolValueByName -
WriteValue "TwinCAT_SystemInfoVarList._AppInfo.ProjectName" -ReadLength 64 | format-hex

ReadWrite access of process image on target '1.2.3.4.5.6:851':
Start ReadWrite operation WriteData: 'TwinCAT_SystemInfoVarList._AppInfo.ProjectName'(IG:0xf004,IO:0
x0000,Len:47),
Read: Type 'System.Byte[]' (Len:'64) on target '1.2.3.4.5.6:851'?
[Y] Yes[A] Yes to All[N] No[L] No to All[S] Suspend[?] Help(default is "Y"): y


00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000   41 44 53 5F 44 79 6E 53 79 6D 62 6F 6C 73 00 00  ADS_DynSymbols..
00000010   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000020   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000030   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

Sends a Read/Write request with index group 0xf004 (SymbolValueByName) and offset 0.

The write data will be initialized with the project symbol path and the returned (read) data is by default a byte array of 64 bytes.

The result value will be formatted as hex code.

EXAMPLE 3

```
PS> $route = Get-AdsRoute -Name 'CX-123456'
PS> $session = $route | New-TcSession -Port 851
PS> $handle = $session | Send-TcReadWrite -IndexGroup SymbolHandleByName -WriteValue "GVL.vgInt" -
ReadType Int32 -force
PS> $session | Read-TcValue -IndexGroup SymbolValueByHandle -IndexOffset $handle -ValueType Int16
42
```

Create a session to the PLC (Port 851) of a target system, determine the SymbolHandle by InstancePath and use this handle to read its 'Int16' Value (INT on PLC System).

PARAMETERS

-IndexGroup

IndexGroup of the Value to ReadWrite, only for IndexGroup/IndexOffset access.

IMPORTANT: Please be aware, that writing data via IndexGroup/IndexOffset can overwrite data in the ProcessImage and posibly destabilizes the system.

No validity check is done for the symbol alignment and therefore this should be done with highest care!

```
Type: UInt32
Parameter Sets: (All)
Aliases: IG

Required: True
Position: Named
Default value: 0
Accept pipeline input: False
Accept wildcard characters: False
```

-IndexOffset

IndexOffset of the Value to write, only for IndexGroup/IndexOffset access.

IMPORTANT: Please be aware, that writing data via IndexGroup/IndexOffset simply overwrites data in the ProcessImage and can destabilize the system.

No validity check is done for the symbol alignment and therefore this should be done with highest care!

If applicable writing data via symbolic information should be preferred!

```
Type: UInt32
Parameter Sets: (All)
Aliases: IO

Required: False
Position: Named
Default value: 0
Accept pipeline input: False
Accept wildcard characters: False
```

-WriteValue

The value to write.

If no additional Length parameter is set, the Write-TcValue Cmdlet marshals this value to its appropriate size.

To not overwrite data of other symbols within the process image, special attention must be taken (see the Confirm and Whatif parameters).

```
Type: Object
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-WriteLength

The Length of the data that will be overwritten within the process image.

By default the marshal size of the object used in the -WriteValue parameter is taken.

This parameter is used to override the marshal size and can be helpful to secure the write operation - to not overwrite more data then expected.

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
```

```
Accept pipeline input: False
Accept wildcard characters: False
```

-ReadLength

The Length of the data that will be read from the process image.

By default, when not specifiying this parameter the marshalling size of the -ReadType parameter will be taken.

This -ReadLength parameter is only helpful when the marshalling size cannot be determined from the read type (e.g.

byte\[\])

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-ReadType

Use the ReadType parameter to specify the Read/Return type of the data.

If not used, this cmdlet returns the raw byte\[\] as result.

```
Type: Type
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: System.Byte[]
Accept pipeline input: False
Accept wildcard characters: False
```

-NetId

The ADS target NetID(s) of the system(s) where to read/write the Value.

More than one target will be supported.

When not specified, this argument defaults to AmsNetId.Local.

```
Type: AmsNetId[]
Parameter Sets: NetIdPortIndexed
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Route

The target system (as Route) where to read/write the value.

```
Type: IRoute[]
Parameter Sets: RouteIndexed
Aliases: Destination

Required: True
Position: Named
Default value: None
```

```
Accept pipeline input: False
Accept wildcard characters: False
```

-Address

The target address where to read/write the Value.

The Address can consist of RouteName, NetId, HostName or IPAddress.

Wildcards are permitted.

```
Type: String[]
Parameter Sets: AddressIndexed
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-Session

The session object represents the target session where to read/write the value.

```
Type: ISession[]
Parameter Sets: SessionIndexed
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByPropertyName, ByValue)
Accept wildcard characters: False
```

-SessionId

The session ID represents the target session where to read/write the value.

```
Type: Int32[]
Parameter Sets: SessionIdIndexed
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Port

The Port, where to read/write the value.

This Parameter is used in combination with the NetId, Route or Address input parameter.

```
Type: Int32
Parameter Sets: NetIdPortIndexed, AddressIndexed, RouteIndexed
Aliases:

Required: True
Position: Named
Default value: 10000
Accept pipeline input: False
Accept wildcard characters: False
```

-Encoding

Specifies the Encoding for strings.

The Default is ANSI with actual code page.

```
Type: Encoding
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: System.Text.UTF8Encoding+UTF8EncodingSealed
Accept pipeline input: False
Accept wildcard characters: False
```

-Async

Starts the write on different threads.

Only for internal use and test purposes.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: True
Accept pipeline input: False
Accept wildcard characters: False
```

-Timeout

The communication ADS timeout in milliseconds.

A value of 0 disables the timeout.

A value \<= 0 sets the Default (5000 ms).

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-Force

Forces the Read/Write operation, even if the FailFastHandler is active.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Quiet

The Quiet parameter suppresses the 'ShouldProcess' messsage and the Cmdlet will be processed without further user confirmation.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: Silent

Required: False
Position: Named
Default value: False
```

```
Accept pipeline input: False
Accept wildcard characters: False
```

-Confirm

Prompts you for confirmation before running the cmdlet.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: cf

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-WhatIf

Shows what would happen if the cmdlet runs.

The cmdlet is not run.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: wi

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.ISession[]

The session object represents the target session where to read/write the value.

OUTPUTS

NOTES

6.43 Set-AdsState

SYNOPSIS

Writes a AdsState control request to the specified target.

SYNTAX

NetIdPort (Default)

```
Set-AdsState [-Command] <AdsStateCommand> [[-NetId] <AmsNetId[]>] [[-Port] <Int32>] [-Quiet] [-
Force]
 [-StateOnly] [-Timeout <Int32>] [-WaitTimeout <Int32>] [-PollingRate <Int32>] [-ThrowError] [-
NoReinit]
 [-Reinitialize] [-NoWait] [-ProgressAction <ActionPreference>] [-WhatIf] [-
Confirm] [<CommonParameters>]
```

AddressStr

```
Set-AdsState [-Command] <AdsStateCommand> [[-Port] <Int32>] [-Address] <String[]> [-Quiet] [-Force]
 [-StateOnly] [-Timeout <Int32>] [-WaitTimeout <Int32>] [-PollingRate <Int32>] [-ThrowError] [-
NoReinit]
 [-Reinitialize] [-NoWait] [-ProgressAction <ActionPreference>] [-WhatIf] [-
Confirm] [<CommonParameters>]
```

Route

```
Set-AdsState [-Command] <AdsStateCommand> [[-Port] <Int32>] [-InputObject] <IRoute[]> [-Quiet] [-
Force]
 [-StateOnly] [-Timeout <Int32>] [-WaitTimeout <Int32>] [-PollingRate <Int32>] [-ThrowError] [-
NoReinit]
 [-Reinitialize] [-NoWait] [-ProgressAction <ActionPreference>] [-WhatIf] [-
Confirm] [<CommonParameters>]
```

Session

```
Set-AdsState [-Command] <AdsStateCommand> -Session <ISession[]> [-Quiet] [-Force] [-StateOnly]
 [-Timeout <Int32>] [-WaitTimeout <Int32>] [-PollingRate <Int32>] [-ThrowError] [-NoReinit] [-
Reinitialize]
 [-NoWait] [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

SessionId

```
Set-AdsState [-Command] <AdsStateCommand> -SessionId <Int32[]> [-Quiet] [-Force] [-StateOnly]
 [-Timeout <Int32>] [-WaitTimeout <Int32>] [-PollingRate <Int32>] [-ThrowError] [-NoReinit] [-
Reinitialize]
 [-NoWait] [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

DESCRIPTION

This Cmdlet sets the ADS State of the specified TwinCAT Targets/AdsServers (E.g.

Start / Stop / Config / Reconfig/ Reset) For setting the SystemService (Port 10000) please see also the 'Restart-TwinCAT' Cmdlet, which is optimized for that case.

EXAMPLES

EXAMPLE 1

```
PS> Set-AdsState -port 10000 -command Reset -force

Ok Target     NetId             Port  ErrorCode Requested Original Reached Latency(ms)
```

```
-- ------     -----              ----  --------- --------- -------- ------- ------------
X  MySystem   172.168.0.1.1.1  10000  NoError   Reset     Run      Run     5007
```

Restart the local System Service

EXAMPLE 2

```
PS> $r = get-AdsRoute
PS> Set-AdsState -port 851 -command Run -InputObject $r -force

Ok Target    NetId             Port  ErrorCode            Requested   Original Reached Latency (ms)
-- ------    -----             ----  ---------            ---------   -------- ------- ------------
X  CX-11111  1.1.1.1.1.1        851  NoError              Run         Stop     Run     293
X  CX-22222  1.1.1.2.1.1        851  NoError              Run         Stop     Run     357
X  CX-22222  1.1.1.3.1.1        851  NoError              Run         Stop     Run     218
X  CX-22222  1.1.1.4.1.1        851  NoError              Run         Stop     Run     324
```

Start the PLC on all registered target systems.

PARAMETERS

-Command

The state command.

Possible values: None, Reset, Start, Stop, Reconfig, Run

```
Type: AdsStateCommand
Parameter Sets: (All)
Aliases:
Accepted values: None, Reset, Start, Stop, Reconfig, Run

Required: True
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-NetId

The NetId address where to set the state (Local system by default).

Multiple values are allowed.

```
Type: AmsNetId[]
Parameter Sets: NetIdPort
Aliases:

Required: False
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Port

Port Address of the AdsServer where to set the state (Port 10000, SystemService by default)

```
Type: Int32
Parameter Sets: NetIdPort, AddressStr, Route
Aliases:

Required: False
Position: 2
Default value: 10000
Accept pipeline input: False
Accept wildcard characters: False
```

-InputObject

Target route(s), where to set the state.

Multiple values are allowed.

```
Type: IRoute[]
Parameter Sets: Route
Aliases: Destination, Route

Required: True
Position: 1
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-Address

The address of the system where to set the state.

This can be the RouteName, NetId, the HostName or the IPAddress.

Wildcards and multiple values are permitted.

```
Type: String[]
Parameter Sets: AddressStr
Aliases: Name

Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-Session

The ADS Session to use for the Cmdlet.

Multiple sessions are allowed.

```
Type: ISession[]
Parameter Sets: Session
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-SessionId

Specifies the Session (with unique ID) to use for the Cmdlet (multiple values are allowed)

```
Type: Int32[]
Parameter Sets: SessionId
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Quiet

Sets the Quiet mode of the command.

The Cmdlet then returns a $true or $false but not the actual states of the targets.

The return value will be $true if all operations succeed and it will be $false if at least one have failed.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Force

Forces the command (no confirmation)

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-StateOnly

The StateOnly mode

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Timeout

The communication ADS timeout in milliseconds.

A value \<= 0 sets the Default Timeout (5000 ms).

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-WaitTimeout

The wait timout for the state change in ms.

This Cmdlet waits for the target state changes which is limited by this WaitTimeout.

A value \<= 0 sets the Default (45000 ms).

This parameter is only used if -NoWait is not set.

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
```

```
Accept pipeline input: False
Accept wildcard characters: False
```

-PollingRate

The Wait polling rate in Milliseconds.

A value \<= 0 sets the Default polling rate (200 ms for local systems, 1000ms for remote systems).

This parameter is only used, if -NoWait is not set.

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-ThrowError

Throws an error, if the target system(s) not reaching the expected state.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-NoReinit

Activates a state check before sending WriteControl if the target system is already in the expected target state

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Reinitialize

This parameter is obsolete and will be removed in future versions.

Please use -NoReinit instead.

This parameter forces the reinitialization of the target.

It is the opposite of the -NoReinit parameter.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: True
Accept pipeline input: False
Accept wildcard characters: False
```

-NoWait

The -NoWait parameter skips the waiting for the target end state.

If set, the Cmdlet returns immediatly after sending the WriteControl request, without waiting for the state change.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Confirm

Prompts you for confirmation before running the cmdlet.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: cf

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-WhatIf

Shows what would happen if the cmdlet runs.

The cmdlet is not run.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: wi

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.IRoute[]

Target route(s), where to set the state.

Multiple values are allowed.

TwinCAT.ISession[]

The ADS Session to use for the Cmdlet.

Multiple sessions are allowed.

OUTPUTS

NOTES

6.44 Set-AmsRouterEndpoint

SYNOPSIS

Sets the AmsConfiguration (Loopback address and port, RouterEndpoint).

SYNTAX

```
Set-AmsRouterEndpoint [[-IP] <IPAddress>] [[-Port] <Int32>] [-Quiet] [-Force]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

DESCRIPTION

This Cmdlet sets the AmsConfiguration setting of the current running process.

By default the AdsClients and AdsServers are connected to the TwinCAT Router./\>

Actually the this Cmdlet doesn't work properly.

Please use the 'AMSROUTER_LOOPBACK_IP', and 'AMSROUTER_LOOPBACK_PORT' environment variables to set the Router Loopback endpoint.

The environment variables must be set at the hosting process before the 'TcXaeMgmt' module is loaded!

Ideally, this Cmdlet is called first after loading the TcXaeMgmt module to configure the system.

The involved communication uses the local Loopback address with port 0xBF02 by default.

To enable virtualization scenarios, where AdsClient / AdsServer / Router applications run in their own (virtual) environment, this address has to be changed for the actual running process.

This can be done in two different ways: 1.Setting of the following Environment Variables before this Powershell Module (TcXaeMgmt) is loaded: $env:AmsConfiguration:LoopbackAddress = "168.0.1.1" $env:AmsConfiguration:LoopbackPort = "1234" Both Variables are optional 2.Set the AmsConfiguration with the 'Set-AmsRouterEndpoint' Cmdlet.

When processing this Cmdlet, all prexisting open AdsSessions will be invalid.\> The Default RouterEndpoint is IPAddess:127.0.0.1 and Port 0xBF02.

EXAMPLES

EXAMPLE 1

```
PS > Set-AmsRouterEndpoint -IP '168.0.1.1' -Port 1234
            Set the process wide RouterEndpoint to IPAddress '168.0.1.1' and port 1234.
```

PARAMETERS

-IP

The state/value to set.

```
Type: IPAddress
Parameter Sets: (All)
Aliases:

Required: False
Position: 0
Default value: 127.0.0.1
Accept pipeline input: False
Accept wildcard characters: False
```

-Port

The state/value to set.

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: 1
Default value: 48898
Accept pipeline input: False
Accept wildcard characters: False
```

-Quiet

The Quiet parameter suppresses the 'ShouldProcess' messsage and the Cmdlet will be processed without further user confirmation.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: Silent

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Force

Forces the command (no confirmation, Resets the FailFastHandler)

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Confirm

Prompts you for confirmation before running the cmdlet.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: cf

Required: False
Position: Named
Default value: None
```

```
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-WhatIf

Shows what would happen if the cmdlet runs.

The cmdlet is not run.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: wi

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

OUTPUTS

NOTES

6.45 Set-IoFreeRun

SYNOPSIS

Sets the IO FreeRun state of the target.

SYNTAX

NetIdPortList (Default)

```
Set-IoFreeRun [-Timeout <Int32>] [-Quiet] [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm]
 [<CommonParameters>]
```

NetIdPort

```
Set-IoFreeRun [-NetId <AmsNetId>] [-Timeout <Int32>] [-Quiet] [-
ProgressAction <ActionPreference>] [-WhatIf]
 [-Confirm] [<CommonParameters>]
```

AddressStr

```
Set-IoFreeRun [-Address] <String> [-Timeout <Int32>] [-Quiet] [-
ProgressAction <ActionPreference>] [-WhatIf]
 [-Confirm] [<CommonParameters>]
```

Session

```
Set-IoFreeRun -Session <ISession> [-Timeout <Int32>] [-Quiet] [-
ProgressAction <ActionPreference>] [-WhatIf]
 [-Confirm] [<CommonParameters>]
```

DESCRIPTION

Sets the IO FreeRun state of a TwinCAT target.

The target must be in Config mode, otherwise an error will be produced.

EXAMPLES

EXAMPLE 1

```
PS> Set-IOFreeRun -NetId 5.62.192.46.1.1
```

Setting the target system with NetID 5.62.192.46.1.1 to IO FreeRun mode.

PARAMETERS

-NetId

The address where to set the TwinCAT FreeRun state.

```
Type: AmsNetId
Parameter Sets: NetIdPort
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Address

The address(es) where to set the TwinCAT FreeRun state.

This can be the RouteName, NetId, the HostName or the IPAddress.

Wildcards are permitted.

```
Type: String
Parameter Sets: AddressStr
Aliases:

Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-Session

The Session to use for the Cmdlet

```
Type: ISession
Parameter Sets: Session
Aliases: InputObject

Required: True
```

```
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-Timeout

The communication ADS timeout in milliseconds.

A value of 0 disables the timeout.

A value \<= 0 sets the Default (5000 ms).

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-Quiet

The Quiet parameter suppresses the 'ShouldProcess' messsage and the Cmdlet will be processed without further user confirmation.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: Silent

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Confirm

Prompts you for confirmation before running the cmdlet.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: cf

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-WhatIf

Shows what would happen if the cmdlet runs.

The cmdlet is not run.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: wi

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.ISession

The Session to use for the Cmdlet

OUTPUTS

NOTES

6.46 Stop-AdsComputer

SYNOPSIS

Stops (shuts down) local and remote TwinCAT computers.

SYNTAX

NetIdPort (Default)

```
Stop-AdsComputer [-Delay <Int32>] [[-NetId] <AmsNetId[]>] [-Timeout <Int32>] [-Quiet]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

Route

```
Stop-AdsComputer [-Delay <Int32>] [-InputObject] <IRoute[]> [-Timeout <Int32>] [-Quiet]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

AddressStr

```
Stop-AdsComputer [-Delay <Int32>] [-Address] <String[]> [-Timeout <Int32>] [-Quiet]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

Session

```
Stop-AdsComputer [-Delay <Int32>] -Session <ISession[]> [-Timeout <Int32>] [-Quiet]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

SessionId

```
Stop-AdsComputer [-Delay <Int32>] -SessionId <Int32[]> [-Timeout <Int32>] [-Quiet]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

DESCRIPTION

The Stop-AdsComputer cmdlet shuts the operating system on the local and remote TwinCAT computers down.

You can use the parameters of Stop-AdsComputer to specify available ADS target systems to shutdown.

The shutdown can be done delayed if Users are logged into the target system (existant Session UI) or forced immediatly.

EXAMPLES

EXAMPLE 1

```
PS> Stop-AdsComputer CX_1111,CX_2222 -force
```

Immediate shutdown of the computers CX_1111 and CX_2222.

The Force parameter supresses the ShouldProcess query.

EXAMPLE 2

```
PS> Stop-AdsComputer -netId '1.2.3.4.1.1', '1.2.3.5.1.1' -Delay 30 -force
```

Stops the TwinCAT targets '1.2.3.4.1.1' and 1.2.3.5.1.1' without ShouldProcess query after a delay of 30 Seconds (if a user is logged in).

PARAMETERS

-Delay

The delay time for the reboot/shutdown of the target system(s) in seconds.

The default is 120 Seconds.

If no user is logged in the target system the reboot/shotdown occurs always immediatly without warning.

In case of a log in, a Warning message with countdown will be presented to the user.

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: 0
Accept pipeline input: False
Accept wildcard characters: False
```

-NetId

NetId(s) of the target system.

```
Type: AmsNetId[]
Parameter Sets: NetIdPort
Aliases:

Required: False
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-InputObject

The ADS routes to shutdown/reboot.

```
Type: IRoute[]
Parameter Sets: Route
Aliases: Destination, Route

Required: True
Position: 1
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-Address

Target names/addresses.

These can consist of RouteName, NetID, HostName or IPAddress.

This parameter supports wildcards.

```
Type: String[]
Parameter Sets: AddressStr
Aliases: Name

Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-Session

The Session(s) to use for addressing the target systems.

```
Type: ISession[]
Parameter Sets: Session
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-SessionId

Specifies the Sessions (with unique ID) to use for addressing the target systems.

```
Type: Int32[]
Parameter Sets: SessionId
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Timeout

The communication ADS timeout in milliseconds.

A value of 0 disables the timeout.

A value \<= 0 sets the Default (5000 ms).

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
```

```
Accept pipeline input: False
Accept wildcard characters: False
```

-Quiet

The Quiet parameter suppresses the 'ShouldProcess' messsage and the Cmdlet will be processed without further user confirmation.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: Silent

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Confirm

Prompts you for confirmation before running the cmdlet.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: cf

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-WhatIf

Shows what would happen if the cmdlet runs.

The cmdlet is not run.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: wi

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.IRoute[]

The ADS routes to shutdown/reboot.

TwinCAT.ISession[]

The Session(s) to use for addressing the target systems.

OUTPUTS

NOTES

6.47 Test-AdsRoute

SYNOPSIS

Test the specified route connection.

SYNTAX

AddressStr (Default)

```
Test-AdsRoute [[-Name] <String[]>] [[-Port] <Int32[]>] [-SourceRoute <RouteTarget>] [-
Mode <PingStrategy>]
 [-ScanSeconds <Int32>] [-Count <Int32>] [-Delay <Int32>] [-DefaultPorts] [-OnlinePorts] [-Quiet]
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

NetId

```
Test-AdsRoute -NetId <AmsNetId[]> [[-Port] <Int32[]>] [-SourceRoute <RouteTarget>] [-
Mode <PingStrategy>]
 [-ScanSeconds <Int32>] [-Count <Int32>] [-Delay <Int32>] [-DefaultPorts] [-OnlinePorts] [-Quiet]
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

Route

```
Test-AdsRoute [[-Port] <Int32[]>] [-SourceRoute <RouteTarget>] -InputObject <RouteTargetCollection>
 [-Mode <PingStrategy>] [-ScanSeconds <Int32>] [-Count <Int32>] [-Delay <Int32>] [-DefaultPorts] [-
OnlinePorts]
 [-Quiet] [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

DESCRIPTION

This Cmdlet establishes a connection to the specified target system and tests if the connection is working.

A Port scan can be executed.

EXAMPLES

EXAMPLE 1

```
PS> Test-AdsRoute -Port 851

Name                 NetId             Port   Latency Result
                                                (ms)
----                 -----             ----   ------- ------
CX-11111             192.168.0.2.1.1   851    3       Ok
```

Test the Port 851 of the local system (PLC 1) for availablity.

EXAMPLE 2

```
PS> Get-AdsRoute | Test-AdsRoute

Name                 NetId             Port   Latency Result
                                               (ms)
----                 -----             ----   ------- ------
CX-11111             192.168.0.2.1.1   10000  4       Ok
CX-22222             192.168.0.3.1.1   10000          Failed
CX-33333             192.168.0.4.1.1   10000  4       Ok
```

Get the locally registered routes and test if they are reachable (on AmsPort 10000)

EXAMPLE 3

```
PS> Test-AdsRoute -OnlinePorts

Name                 NetId             Port   Latency Result
   (ms)
----                 -----             ----   ------- ------
CX-11111             192.168.0.2.1.1   10     0.6     Ok
CX-11111             192.168.0.2.1.1   11     1.3     Ok
CX-11111             192.168.0.2.1.1   12     1.2     Ok
CX-11111             192.168.0.2.1.1   30     3       Ok
CX-11111             192.168.0.2.1.1   131    75      Ok
CX-11111             192.168.0.2.1.1   32829  125     Ok
CX-11111             192.168.0.2.1.1   340    122     Ok
CX-11111             192.168.0.2.1.1   850    171     Ok
CX-11111             192.168.0.2.1.1   32830  174     Ok
CX-11111             192.168.0.2.1.1   351    171     Ok
CX-11111             192.168.0.2.1.1   350    172     Ok
CX-11111             192.168.0.2.1.1   270    219     Ok
CX-11111             192.168.0.2.1.1   851    220     Ok
```

Scans the propagated AmsPorts for the local system.

PARAMETERS

-Name

The name(s) or address(es) of the systems to test.

These can consist of RouteName, NetID, HostName or IPAddress.

Wildcards are permitted.

```
Type: String[]
Parameter Sets: AddressStr
Aliases: Address

Required: False
Position: 0
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-NetId

The NetId(s) of the target system to test (AmsNetId.Local by default)

```
Type: AmsNetId[]
Parameter Sets: NetId
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByPropertyName, ByValue)
Accept wildcard characters: False
```

-Port

The Port(s) of the target system to test.

```
Type: Int32[]
Parameter Sets: (All)
Aliases:

Required: False
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-SourceRoute

The source system where to test the Route.

```
Type: RouteTarget
Parameter Sets: (All)
Aliases: Source

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-InputObject

The routes targets to test with this Cmdlet.

```
Type: RouteTargetCollection
Parameter Sets: Route
Aliases: Destination

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-Mode

The Ping Strategy (PingStrategy.Ads by default)

Possible values: None, IP, HostName, IPOrHostName, Ads, AdsGetState, AdsTestConnection, Default

```
Type: PingStrategy
Parameter Sets: (All)
Aliases:
Accepted values: None, IP, HostName, IPOrHostName, Ads, AdsGetState, AdsTestConnection, Default

Required: False
Position: Named
Default value: AdsGetState
Accept pipeline input: False
Accept wildcard characters: False
```

-ScanSeconds

Sets timeout for each single test request.

The test fails if a (single) response isn't received before the timeout expires.

The default timeout is 2 seconds.

```
Type: Int32
Parameter Sets: (All)
Aliases: TTL, TimeToLive, TimeoutSeconds

Required: False
Position: Named
```

```
Default value: 2
Accept pipeline input: False
Accept wildcard characters: False
```

-Count

Specifies the number of echo/ping requests to send.

The default value is 1.

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: 1
Accept pipeline input: False
Accept wildcard characters: False
```

-Delay

Specifies the interval between pings/tests, in seconds.

The default value is 1 Second.

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: 1
Accept pipeline input: False
Accept wildcard characters: False
```

-DefaultPorts

Tests all default ports.

The following ports will be tested: 10000, 300, 301, 302, 303, 501, 801, 811, 821, 831, 850, 851, 852, 853, 854, 855, 19200 The 'DefaultPorts' switch overrides the 'Port' parameter.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-OnlinePorts

Determines all active/online ports from the target and tests them.

The 'OnlinePorts' switch overrides the 'DefaultPorts' and 'Port' parameters.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Quiet

The Quiet mode.

Returns a boolean only ($true, if one ping succeeded and $false if all failed)

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.Ads.AmsNetId[]

The NetId(s) of the target system to test (AmsNetId.Local by default)

TwinCAT.RouteTargetCollection

The routes targets to test with this Cmdlet.

OUTPUTS

NOTES

6.48 Unregister-AdsHandle

SYNOPSIS

Unregisters a symbol handle.

SYNTAX

NetIdPortHandle (Default)

```
Unregister-AdsHandle [-NetId <AmsNetId>] -Port <Int32> [-Handle] <UInt32[]>
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

NetIdPortInfo

```
Unregister-AdsHandle [-NetId <AmsNetId>] -Port <Int32> [-InputObject] <AdsHandleInfo[]>
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

RouteHandle

```
Unregister-AdsHandle -Route <IRoute> -Port <Int32> [-Handle] <UInt32[]> [-
ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

RouteInfo

```
Unregister-AdsHandle -Route <IRoute> -Port <Int32> [-InputObject] <AdsHandleInfo[]>
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

AddressHandle

```
Unregister-AdsHandle -Address <String> -Port <Int32> [-Handle] <UInt32[]> [-
ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

AddressInfo

```
Unregister-AdsHandle -Address <String> -Port <Int32> [-InputObject] <AdsHandleInfo[]>
 [-ProgressAction <ActionPreference>] [<CommonParameters>]
```

SessionHandle

```
Unregister-AdsHandle -Session <ISession> [-Handle] <UInt32[]> [-ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

SessionInfo

```
Unregister-AdsHandle -Session <ISession> [-InputObject] <AdsHandleInfo[]> [-
ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

SessionIdHandle

```
Unregister-AdsHandle -SessionId <Int32> [-Handle] <UInt32[]> [-ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

SessionIdInfo

```
Unregister-AdsHandle -SessionId <Int32> [-InputObject] <AdsHandleInfo[]> [-
ProgressAction <ActionPreference>]
 [<CommonParameters>]
```

DESCRIPTION

This Cmdlet unregisters an already registered symbol handle from the target system.

The Cmdlet supports raw \[uint\] handles or AdsHandleInfo objects.

EXAMPLES

EXAMPLE 1

```
PS> $session = New-TcSession -NetId '1.2.3.4.5.6' -Port 851
PS> $handleInfo = $session | get-AdsHandle -Path 'TwinCAT_SystemInfoVarList._AppInfo.ProjectName'
PS> $handleInfo = register-AdsHandle -Path 'TwinCAT_SystemInfoVarList._AppInfo.ProjectName' -
Session $s
PS> $handleInfo

InstancePath                                    Result Handle
```

```
------------                                        ------  ------
TwinCAT_SystemInfoVarList._AppInfo.ProjectName NoError 0x428000FC (1115685116)

PS> Read-TcValue -Session $session -IndexGroup SymbolValueByHandle -IndexOffset $handleInfo.Handle -
Type String

MyProject

PS> $handle | Unregister-AdsHandle -Session $session
PS> $session | Close-tcsession
```

Opens a new device session, registers a Symbol Handle to the ProjectName of the running PLC Project, Reads the value by handle unregisters the handle and closes the session again.

PARAMETERS

-NetId

The NetId part of the device target address.

```
Type: AmsNetId
Parameter Sets: NetIdPortHandle, NetIdPortInfo
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Route

Specifies the target system.

```
Type: IRoute
Parameter Sets: RouteHandle, RouteInfo
Aliases: Destination

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Address

The target address of the system.

The Address can consist of RouteName, NetId, IPAddress or HostName.

Wildcards are permitted and ArgumentCompleter is supported.

```
Type: String
Parameter Sets: AddressHandle, AddressInfo
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-Session

The Session object (instead of specifieng the target system address).

```
Type: ISession
Parameter Sets: SessionHandle, SessionInfo
Aliases:

Required: True
Position: Named
```

```
Default value: None
Accept pipeline input: True (ByPropertyName, ByValue)
Accept wildcard characters: False
```

-SessionId

Specifies the Session (with unique ID) to use (instead of specifying the address).

```
Type: Int32
Parameter Sets: SessionIdHandle, SessionIdInfo
Aliases:

Required: True
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-Port

The address Port to use.

ClearText names for the Port and ArgumentCompleter are supported.

```
Type: Int32
Parameter Sets: NetIdPortHandle, NetIdPortInfo, RouteHandle, RouteInfo, AddressHandle, AddressInfo
Aliases:

Required: True
Position: Named
Default value: 10000
Accept pipeline input: False
Accept wildcard characters: False
```

-Handle

The instance path of the symbol to read (symbolic access).

This parameter supports wildcards.

```
Type: UInt32[]
Parameter Sets: NetIdPortHandle, RouteHandle, AddressHandle, SessionHandle, SessionIdHandle
Aliases:

Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-InputObject

The AdsHandleInfo object (produced by Register-AdsHandle Cmdlet)

```
Type: AdsHandleInfo[]
Parameter Sets: NetIdPortInfo, RouteInfo, AddressInfo, SessionInfo, SessionIdInfo
Aliases: HandleInfo

Required: True
Position: 1
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga
```

```
Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.ISession

The Session object (instead of specifieng the target system address).

TwinCAT.Management.Automation.AdsHandleInfo[]

The AdsHandleInfo object (produced by Register-AdsHandle Cmdlet)

OUTPUTS

NOTES

6.49 Write-TcValue

SYNOPSIS

Write values to TwinCAT devices.

SYNTAX

NetIdPortSymbol (Default)

```
Write-TcValue [-NetId <AmsNetId[]>] -Port <Int32> [-Value <Object>] [-Path] <String> [-
Encoding <Encoding>]
 [-Timeout <Int32>] [-Force] [-Quiet] [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm]
 [<CommonParameters>]
```

NetIdPortIndexed

```
Write-TcValue [-NetId <AmsNetId[]>] -Port <Int32> [-Value <Object>] [-IndexGroup] <UInt32>
 [[-IndexOffset] <UInt32>] [-Size <Int32>] [-Encoding <Encoding>] [-Timeout <Int32>] [-Force] [-
Quiet]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

RouteIndexed

```
Write-TcValue -Route <IRoute[]> -Port <Int32> [-Value <Object>] [-IndexGroup] <UInt32>
 [[-IndexOffset] <UInt32>] [-Size <Int32>] [-Encoding <Encoding>] [-Timeout <Int32>] [-Force] [-
Quiet]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

RouteSymbol

```
Write-TcValue -Route <IRoute[]> -Port <Int32> [-Value <Object>] [-Path] <String> [-
Encoding <Encoding>]
 [-Timeout <Int32>] [-Force] [-Quiet] [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm]
 [<CommonParameters>]
```

AddressIndexed

```
Write-TcValue -Address <String[]> -Port <Int32> [-Value <Object>] [-IndexGroup] <UInt32>
 [[-IndexOffset] <UInt32>] [-Size <Int32>] [-Encoding <Encoding>] [-Timeout <Int32>] [-Force] [-
Quiet]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

AddressSymbol

```
Write-TcValue -Address <String[]> -Port <Int32> [-Value <Object>] [-Path] <String> [-
Encoding <Encoding>]
 [-Timeout <Int32>] [-Force] [-Quiet] [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm]
 [<CommonParameters>]
```

SessionIndexed

```
Write-TcValue -Session <ISession[]> [-Value <Object>] [-IndexGroup] <UInt32> [[-
IndexOffset] <UInt32>]
 [-Size <Int32>] [-Encoding <Encoding>] [-Timeout <Int32>] [-Force] [-Quiet]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

SessionSymbol

```
Write-TcValue -Session <ISession[]> [-Value <Object>] [-Path] <String> [-Encoding <Encoding>]
 [-Timeout <Int32>] [-Force] [-Quiet] [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm]
 [<CommonParameters>]
```

SessionIdIndexed

```
Write-TcValue -SessionId <Int32[]> [-Value <Object>] [-IndexGroup] <UInt32> [[-
IndexOffset] <UInt32>]
 [-Size <Int32>] [-Encoding <Encoding>] [-Timeout <Int32>] [-Force] [-Quiet]
 [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

SessionIdSymbol

```
Write-TcValue -SessionId <Int32[]> [-Value <Object>] [-Path] <String> [-Encoding <Encoding>] [-
Timeout <Int32>]
 [-Force] [-Quiet] [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

InputObject

```
Write-TcValue [-Value <Object>] [-InputObject] <ISymbol> [-Encoding <Encoding>] [-
Timeout <Int32>] [-Force]
 [-Quiet] [-ProgressAction <ActionPreference>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

DESCRIPTION

This Cmdlet writes values to TwinCAT Devices.

The devices can be accessed via different ValueProviders.

All sorts of ADS-addressing will be supported by this Cmdlet: Addressing by IndexGroup / IndexOffset (see IndexGroup, IndexOffset paramters) Addressing by Instance Path (see path parameter) Addressing by Symbol (see InputObject / Symbol parameter)

IMPORTANT: Writing values should be done with highest care because it could destabilize the TwinCAT System when the write operation is not addressed properly.

While writing with available symbol information is not critical and should be preferred the size and position of symbol data is known within the process image), the access via Instance path is less secure.

The size of the overwritten data is not known and therefore not checked by the Cmdlet.

The highest attention should be taken with write IndexGroup / IndexOffset write operations because beneath the unknown data size even the position of the data is not checked.

The data is written directly into the process image.

EXAMPLES

EXAMPLE 1

```
PS> $session = New-TcSession -NetId 1.2.3.4.5.6 -Port 851
PS> $projectNameSymbol = $session | Get-TcSymbol -path "*ProjectName"
PS> $projectNameSymbol

InstanceName DataType   Size InstancePath
------------ --------   ---- ------------
ProjectName  STRING(63) 64   TwinCAT_SystemInfoVarList._AppInfo.ProjectName

PS> $projectNameSymbol | Read-TcValue

OldProjectName

PS> $projectNameSymbol | Write-TcValue -Value "NewProjectName" -force
PS> $projectNameSymbol | ReadTcValue

NewProjectName
```

This example shows how to create a session, determining the Symbol 'ProjectName within the _AppInfo Struct on a running PLC project and reading its value.

After that, the Value will be overwritten with 'NewProjectName'.

EXAMPLE 2

```
Write-TcValue -session $session -IndexGroup 0x4040 -IndexOffset 0x1247A8 -Value "NewProjectName"
```

Writes a string typed Value to the specified IndexGroup/IndexOffset Address.

PARAMETERS

-NetId

The ADS target NetID(s) of the system(s) where to write the Value.

More than one target will be supported.

```
Type: AmsNetId[]
Parameter Sets: NetIdPortSymbol, NetIdPortIndexed
Aliases:

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Route

The target system (as Route) where to write the value.

```
Type: IRoute[]
Parameter Sets: RouteIndexed, RouteSymbol
Aliases: Destination

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Address

The target address where to write the Value.

The Address can consist of RouteName, NetId, HostName or IPAddress.

Wildcards are permitted.

```
Type: String[]
Parameter Sets: AddressIndexed, AddressSymbol
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-Session

The session object represents the target session where to write the value.

```
Type: ISession[]
Parameter Sets: SessionIndexed, SessionSymbol
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: True (ByPropertyName, ByValue)
Accept wildcard characters: False
```

-SessionId

The session ID represents the target session where to write the value.

```
Type: Int32[]
Parameter Sets: SessionIdIndexed, SessionIdSymbol
Aliases:

Required: True
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-Port

The Port, where to write the value.

This Parameter is used in combination with the NetId, Route or Address input parameter.

```
Type: Int32
Parameter Sets: NetIdPortSymbol, NetIdPortIndexed, RouteIndexed, RouteSymbol, AddressIndexed, Addres
sSymbol
Aliases:

Required: True
Position: Named
Default value: 10000
Accept pipeline input: False
Accept wildcard characters: False
```

-Value

The value to write.

If no additional Length parameter is set, the Write-TcValue Cmdlet marshalls this value to its appropiate size.

To not overwrite data of other symbols within the process image, special attention must be taken (see the Confirm and Whatif parameters).

```
Type: Object
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
```

```
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-IndexGroup

IndexGroup of the Value to write, only for IndexGroup/IndexOffset access.

IMPORTANT: Please be aware, that writing data via IndexGroup/IndexOffset simply overwrites data in the ProcessImage and can destabilize the system.

No validity check is done for the symbol alignment and therefore this should be done with highest care!

If applicable writing data via symbolic information should be preferred!

```
Type: UInt32
Parameter Sets: NetIdPortIndexed, RouteIndexed, AddressIndexed, SessionIndexed, SessionIdIndexed
Aliases: IG

Required: True
Position: 1
Default value: 0
Accept pipeline input: False
Accept wildcard characters: False
```

-IndexOffset

IndexOffset of the Value to write, only for IndexGroup/IndexOffset access.

IMPORTANT: Please be aware, that writing data via IndexGroup/IndexOffset simply overwrites data in the ProcessImage and can destabilize the system.

No validity check is done for the symbol alignment and therefore this should be done with highest care!

If applicable writing data via symbolic information should be preferred!

```
Type: UInt32
Parameter Sets: NetIdPortIndexed, RouteIndexed, AddressIndexed, SessionIndexed, SessionIdIndexed
Aliases: IO

Required: False
Position: 2
Default value: 0
Accept pipeline input: False
Accept wildcard characters: False
```

-Size

The Length of the data that will be overwritten within the process image.

IMPORTANT: Please be aware, that writing data via IndexGroup/IndexOffset simply overwrites data in the ProcessImage and could destabilize the system.

No further validity check is done for the symbol alignment and therefore this should be done with highest care (best with use of the -Confirm and -Whatif Cmdlet arguments).

If applicable writing data via symbolic information should be preferred!

```
Type: Int32
Parameter Sets: NetIdPortIndexed, RouteIndexed, AddressIndexed, SessionIndexed, SessionIdIndexed
Aliases: Length, WriteSize

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-Path

The instance path to the symbol to write (Symbolic access).

Wildcards are permitted.

```
Type: String
Parameter Sets: NetIdPortSymbol, RouteSymbol, AddressSymbol, SessionSymbol, SessionIdSymbol
Aliases:

Required: True
Position: 1
Default value: None
Accept pipeline input: False
Accept wildcard characters: True
```

-InputObject

The symbol object on which to write the value.

```
Type: ISymbol
Parameter Sets: InputObject
Aliases: Symbol

Required: True
Position: 1
Default value: None
Accept pipeline input: True (ByValue)
Accept wildcard characters: False
```

-Encoding

Specifies the Encoding for strings.

The Default is ANSI with actual code page.

```
Type: Encoding
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: System.Text.UTF8Encoding+UTF8EncodingSealed
Accept pipeline input: False
Accept wildcard characters: False
```

-Timeout

The communication ADS timeout in milliseconds.

A value of 0 disables the timeout.

A value \<= 0 sets the Default (5000 ms).

```
Type: Int32
Parameter Sets: (All)
Aliases:

Required: False
Position: Named
Default value: -1
Accept pipeline input: False
Accept wildcard characters: False
```

-Force

Forces the write operation, even if the FailFastHandler is active.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases:
```

```
Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Quiet

The Quiet parameter suppresses the 'ShouldProcess' messsage and the Cmdlet will be processed without further user confirmation.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: Silent

Required: False
Position: Named
Default value: False
Accept pipeline input: False
Accept wildcard characters: False
```

-Confirm

Prompts you for confirmation before running the cmdlet.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: cf

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-ProgressAction

{{ Fill ProgressAction Description }}

```
Type: ActionPreference
Parameter Sets: (All)
Aliases: proga

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

-WhatIf

Shows what would happen if the cmdlet runs.

The cmdlet is not run.

```
Type: SwitchParameter
Parameter Sets: (All)
Aliases: wi

Required: False
Position: Named
Default value: None
Accept pipeline input: False
Accept wildcard characters: False
```

CommonParameters

This cmdlet supports the common parameters: -Debug, -ErrorAction, -ErrorVariable, -InformationAction, -InformationVariable, -OutVariable, -OutBuffer, -PipelineVariable, -Verbose, -WarningAction, and -WarningVariable. For more information, see about_CommonParameters.

INPUTS

TwinCAT.ISession[]

The session object represents the target session where to write the value.

TwinCAT.TypeSystem.ISymbol

The symbol object on which to write the value.

OUTPUTS

NOTES

More Information:
www.beckhoff.com/te1000

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com