



## **Cisco TEO PowerShell Snap-in CLI Guide**

Release 2.3  
April 2012

**Americas Headquarters**  
Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, CA 95134-1706  
USA  
<http://www.cisco.com>  
Tel: 408 526-4000  
800 553-NETS (6387)  
Fax: 408 527-0883

Text Part Number: OL-24571-02

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

*Cisco TEO PowerShell Snap-in CLI Guide*

© 2011–2012 Cisco Systems, Inc. All rights reserved.



# CONTENTS

## **Preface**   vii

Organization	vii
Conventions	viii
Product Documentation	ix
Documentation Formats	ix
Guides and Release Notes	ix
Online Help	ix
Open Source License Acknowledgements	ix
Obtaining Documentation and Submitting a Service Request	ix

---

## **CHAPTER 1**

### **Installing TEO PowerShell CLI Snap-In**   1-1

PowerShell CLI Snap-In System Requirements	1-2
PowerShell 32-bit Support	1-2
Supported Platforms	1-2
Adding TEO PowerShell CLI Snap-In	1-3
PowerShell CLI Snap-in DLL Registration	1-3
Set Execution Policy	1-3
TEO PowerShell CLI Snap-In Shortcut	1-3

---

## **CHAPTER 2**

### **Understanding TEO Commandlets**   2-1

Command Line Syntax	2-2
Command Parameters	2-2
Command Line Errors	2-2
Process Commandlets	2-3
Disable-TEOProcess Commandlet	2-3
Syntax	2-3
Disable-TEOProcess Parameters	2-3
Input Type	2-5
Return Type	2-5
Examples	2-5
Disable-TEOProcess Help	2-6
Enable-TEOProcess Commandlet	2-6
Syntax	2-6
Enable-TEOProcess Parameters	2-6

- Input Type 2-8
- Return Type 2-8
- Examples 2-9
- Enable-TEOProcess Help 2-9
- Get-TEOProcess Commandlet 2-9
  - Syntax 2-9
  - Get-TEOProcess Parameters 2-9
  - Input Type 2-10
  - Return Type 2-11
  - Examples 2-11
  - Get-TEOProcess Help 2-11
- Get-TEOProcess Instance Commandlet 2-12
  - Syntax 2-12
  - Get-TEOProcess Parameters 2-12
  - Input Type 2-13
  - Return Type 2-13
  - Examples 2-14
  - Input Type 2-14
  - Return Type 2-14
  - Get-TEOProcessInstance Help 2-14
- Start-TEOProcess Commandlet 2-15
  - Syntax 2-15
  - Start-TEOProcess Parameters 2-15
  - Input Type 2-19
  - Return Type 2-19
  - Examples 2-19
  - Start-TEOProcess Help 2-20
- Stop-TEOProcessInstance Commandlet 2-20
  - Syntax 2-20
  - Stop-TEOProcessInstance Parameters 2-20
  - Input Type 2-21
  - Return Type 2-21
  - Examples 2-21
  - Stop-TEOProcessInstance Help 2-22
- Target Commandlets 2-23
  - Disable-TEOTarget Commandlet 2-23
    - Syntax 2-23
    - Disable-TEOTarget Parameters 2-23
    - Input Type 2-25
    - Return Type 2-25

Examples	2-25
Disable-TEOTarget Help	2-26
Enable-TEOTarget Commandlet	2-26
Syntax	2-26
Enable-TEOTarget Parameters	2-26
Input Type	2-28
Return Type	2-28
Examples	2-29
Enable-TEOTarget Help	2-29
Get-TEOTarget Commandlet	2-29
Syntax	2-29
Get-TEOTarget Parameters	2-29
Input Type	2-30
Return Type	2-31
Examples	2-31
Get-TEOTarget Help	2-31

---

**INDEX**





# Preface

---

**Revised: April 2012, OL-24571-02**

Tidal Enterprise Orchestrator's powerful process automation engine provides the logical constructs necessary to support the most complex requirements to automate the administrative and operational tasks necessary to manage these systems.

Microsoft Windows PowerShell integrates with the Microsoft .NET Framework and provides an environment to perform administrative tasks by execution of commandlets, executables, or by instantiating regular .NET classes.



**Note**

---

For additional information about Microsoft Windows PowerShell, refer to the [Windows PowerShell Owner's Manual](#).

---

The TEO Command Line Interface (CLI) Snap-In leverages the Windows PowerShell console to execute scripts using Tidal Enterprise Orchestrator commandlets. This allows IT professionals to more easily control system administration and accelerate automation. The CLI is included in the standard TEO installation and will only execute against the local TEO server under the credentials of the current user running the CLI command.

This guide is intended to provide information on the use of the PowerShell command line interface Snap-In for Tidal Enterprise Orchestrator. The information is organized into the following chapters.

## Organization

This guide includes the following sections:

Chapter 1	<a href="#">Installing TEO PowerShell CLI Snap-In</a>	Provides installation instructions on Tidal Enterprise Orchestrator PowerShell CLI Snap-in
Chapter 2	<a href="#">Understanding TEO Commandlets</a>	Provides overview of the available TEO PowerShell CLI Snap-In commandlets

# Conventions

This guide uses the following conventions:

Convention	Indication
<b>bold font</b>	Commands and keywords and user—entered text appear in <b>bold font</b> .
<i>italic font</i>	Document titles, new or emphasized terms, and arguments for which you supply values are in <i>italic font</i> .
[ ]	Elements in square brackets are optional.
{ x   y   z }	Required alternative keywords are grouped in braces and separated by vertical bars.
[ x   y   z ]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.
<code>courier font</code>	Terminal sessions and information the system displays appear in <code>courier font</code> .
< >	Nonprinting characters such as passwords are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.



## Note

Means *reader take note*.



## Tip

Means *the following information will help you solve a problem*.



## Caution

Means *reader be careful*. In this situation, you might perform an action that could result in equipment damage or loss of data.



## Timesaver

Means *the described action saves time*. You can save time by performing the action described in the paragraph.



## Warning

Means *reader be warned*. In this situation, you might perform an action that could result in **bodily injury**.

# Product Documentation

## Documentation Formats

Documentation is provided in the following electronic formats:

- Adobe® Acrobat® PDF files
- Online help

You must have Adobe® Reader® installed to read the PDF files. Adobe Reader installation programs for common operating systems are available for free download from the Adobe Web site at [www.adobe.com](http://www.adobe.com).

## Guides and Release Notes

You can download the TEO product documentation from Cisco.com. Release Notes can be found on Cisco.com and the product CD.

## Online Help

Online help is available and can be accessed using the following methods:

- Click **Help** on any dialog box in the application to open the help topic in a pane to the right of the dialog box.
- In the Tidal Enterprise Orchestrator console:
  - Click the **Help Pane**  icon on the toolbar to open the help topic in a pane to the right of the Results pane.
  - Click **Help** on the menu bar.

## Open Source License Acknowledgements

Licenses and notices for open source software used in Tidal Enterprise Orchestrator can be found in the [Open Source License Acknowledgements](#) on Cisco.com. If you have any questions about the open source contained in this product, please email [external—opensource—requests@cisco.com](mailto:external—opensource—requests@cisco.com).

## Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

<http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html>

Subscribe to the *What's New in Cisco Product Documentation* as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS Version 2.0.





# CHAPTER 1

## Installing TEO PowerShell CLI Snap-In

---

Before attempting to install the TEO PowerShell CLI Snap-In, it is recommended that the user reviews the system requirements and prerequisites before beginning the installation process.

The TEO PowerShell CLI Snap-In is shipped with the installation CD that is used to install the application. This chapter guides the user through the process of installing Tidal Enterprise Orchestrator PowerShell CLI Snap-in.

### Topics covered in this chapter

- [PowerShell CLI Snap-In System Requirements, page 1-2](#)
- [Adding TEO PowerShell CLI Snap-In, page 1-3](#)

# PowerShell CLI Snap-In System Requirements

The TEO PowerShell CLI Snap-In requires the Microsoft Windows PowerShell 1.0 as a software requirement.



**Note**

For additional information about Microsoft Windows PowerShell, refer to the [Windows PowerShell Owner's Manual](#).

## Supported Platforms

Windows PowerShell is supported on the following platforms:

- Windows Server 2003 SP2
- Windows Server 2008 SP1

## Adding TEO PowerShell CLI Snap-In

The TEO PowerShell CLI Snap-in files are included in the Tidal Enterprise Orchestrator program folder upon installation of TEO. The TEO installer automatically detects whether Windows PowerShell is installed on the machine. If Windows PowerShell is installed, then TEO will register the TEO PowerShell CLI Snap-in DLL. If PowerShell is not installed, then refer to [How to Download Windows PowerShell 1.0](#) on the Microsoft website to access the free Windows PowerShell download.

A script is provided in the TEO program folder which allows users to automatically perform the functions necessary to add the PowerShell CLI Snap-in. This script will only work on 32-bit machines.

To add the TEO PowerShell CLI Snap-In:

In the TEO program folder, double-click **Tidal.Automation.CLI.Register.vbs**. The script launches and performs the following functions which add and configure the TEO snap-in to Microsoft Windows PowerShell.

- Registers the PSSnapin.dll
- Changes the Set Execution Policy to *RemoteSigned*
- Creates a Shortcut for TEO PowerShell

## PowerShell CLI Snap-in DLL Registration

If PowerShell is already installed on the machine where TEO is installed, then the *Tidal.Automation.CLI.CorePSSnapin.dll* file is automatically registered when the *Tidal.Automation.CLI.Register.vbs* script is launched. If PowerShell is installed after the TEO installation, then the user should copy the 32-bit version of the *installutil.exe* file located in the following file path:

```
C:\Windows\microsoft.net\Framework\v2.0.50727
```

The user should then paste the *installutil.exe* file into the TEO program directory and then run the following:

```
C:\Program Files\Cisco\Tidal Enterprise Orchestrator\installutil.exe
Tidal.Automation.CLI.CorePSSnapin.dll
```

**Note**

If the user is on a 64-bit operating system, the file path should begin with: *C:\Program Files(x86)* instead of *C:\Program Files*.

## Set Execution Policy

If it is the first time the PS snap-in is run, the set the execution policy will be set to *Restricted*. The *Tidal.Automation.CLI.Register.vbs* script changes the setting to *Remote Signed*. *RemoteSigned* ensures that remote scripts are digitally signed before the PowerShell CLI snap-in can run scripts.

To verify the Set Execution policy is set to *RemoteSigned* in PowerShell, run `PowerShell Get-ExecutionPolicy`. If this command returns *Restricted*, SCOM will not be able to discover or monitor TEO managed targets.

## TEO PowerShell CLI Snap-In Shortcut

The *Tidal.Automation.CLI.Register.vbs* script creates the shortcut automatically.





## CHAPTER 2

# Understanding TEO Commandlets

---

TEO PowerShell CLI Snap-in consists of commandlets which allow users to view details of the existing processes and targets in the TEO server. Using the PowerShell CLI Snap-in, the user will be able to generate a list of processes and targets, as well as enable and disable specific processes and targets.

To see a list of all the TEO commandlets, run the following commandlet in the PowerShell console:

```
get-command *-TEO*
```

This chapter provides an overview of the available TEO PowerShell CLI Snap-In commandlets. Users will be able to use the command line interface to view the properties for a process and target. Each listed command will contain the following sections to explain different aspects of the command.

- **Syntax**—Displays how to enter the command. The syntax section uses certain punctuation conventions to denote characteristics of the command options.
- **Examples**—Displays some ways to use the command.
- **Parameters**—Explains each parameter for the command
- **Help**—Provides the available help commandlets for the command

### Topics covered in this chapter

- [Command Line Syntax, page 2-2](#)
- [Process Commandlets, page 2-3](#)
- [Target Commandlets, page 2-23](#)

## Command Line Syntax

Command syntax is listed for each command. The following describe how certain conventions are used to characterize command options.

- Any argument not enclosed in brackets is required.
- When options are enclosed in square brackets, i.e., [ and ], they are optional and not required.
- If options are separated with a vertical bar, i.e., |, either one or the other option must be used.
- The argument for an option may be required or optional, depending on the command.

## Command Parameters

Parameters are available for each command. When specifying multiple values for a parameter, use commas to separate the values. For example, for `<parameter-name> <value1>, <value2>`, users cannot use the `-Name`, `-ID`, and `-InputObject` parameters in the same command.

The following table is a description of each parameter.

Parameter	Description
<code>-name &lt;String[]&gt;</code>	Specifies one or more processes or targets by name.
<code>-id &lt;Guid[]&gt;</code>	Specifies one or more processes by process ID.
<code>-passThru &lt;SwitchParameter&gt;</code>	Passes the object created by this commandlet through the pipeline.
<code>-confirm</code>	Prompts you for confirmation before executing the command.
<code>&lt;CommonParameters&gt;</code>	This commandlet supports the common parameters: <code>-Verbose</code> , <code>-Debug</code> , <code>-ErrorAction</code> , <code>-ErrorVariable</code> , and <code>-OutVariable</code> .  For more information, type, <code>get-help about_commonparameters</code>

## Command Line Errors

When executing a TEO PowerShell command, the following errors will display if the following occur:

- Process with the specified name does not exist
- Process with the specified name is found, but no process instances exist
- Process with the specified ID does not exist
- If an identity variable is given, but does not exist in the current domain.

# Process Commandlets

The TEO commandlet consists of generating process detail information and performing a minimal amount of action against a specific process. The process-specific commandlets will display the list of all processes defined in the TEO server that can be started by the user.

The default display of a process is a table that includes the following columns:

Column	Description
Name	The name of the process
ID	The ID of the process
Enabled	<i>True</i> if the process is enabled, and <i>False</i> otherwise

## Disable-TEOProcess Commandlet

The *Disable-TEOProcess* commandlet disables processes. Users can identify a particular process by process name or process ID, or pass a process configuration through the pipeline to *Disable-TEOProcess*. For *Disable-TEOProcess*, the default method is by process name.

### Syntax

- `Disable-TEOProcess [-name] <String[]> [<CommonParameters>]`
- `Disable-TEOProcess -id <Guid[]> [<CommonParameters>]`
- `Disable-TEOProcess -inputObject <ProcessConfiguration[]> [<CommonParameters>]`
- `Disable-TEOProcess -passThru <ProcessConfiguration[]> [<CommonParameters>]`
- `Disable-TEOProcess -confirm [<CommonParameters>]`
- `Disable-TEOProcess [-whatIf] [<CommonParameters>]`

### Disable-TEOProcess Parameters

When specifying multiple values for a parameter, use commas to separate the values.

#### Example:

```
<parameter-name> <value1>, <value2>
```

Users cannot use the *-Name*, *-ID*, and *-InputObject* parameters in the same command.

Without parameters, *Disable-TEOProcess* disables all of the processes on the server, as though the user typed `Disable-TEOProcess *`.

Parameter	Description	Option	Default Value
-name <String[]>	Specifies one or more processes by name.  To specify multiple processes, use commas to separate the process names or use wildcard characters.  The parameter name <i>-Name</i> is optional.	Required?	false
		Position?	1
		Default value	Null
		Accept pipeline input?	true (ByPropertyName)
		Accept wildcard characters?	true
-id <Guid[]>	Specifies one or more processes by process ID.  To specify multiple IDs, use commas to separate the IDs.  To find the ID of a process, type <code>Get-TEOProcess</code> .	Required?	true
		Position?	named
		Default value	Null
		Accept pipeline input?	
		Accept wildcard characters?	
-inputObject <ProcessConfiguration[]>	Accepts process configurations as input to the commandlet.  Enter a variable that contains the objects or type a command or expression that gets the objects.	Required?	true
		Position?	named
		Default value	Null
		Accept pipeline input?	
		Accept wildcard characters?	
-passThru <SwitchParameter>	Passes the object created by this commandlet through the pipeline.  By default, this commandlet does not pass any objects through the pipeline.	Required?	false
		Position?	named
		Default value	False
		Accept pipeline input?	false
		Accept wildcard characters?	false

Parameter	Description	Option	Default Value
-confirm	Prompts the user for confirmation before executing the command	Required?	false
		Position?	named
		Default value	
		Accept pipeline input?	false
		Accept wildcard characters?	false
<CommonParameters>	<p>This commandlet supports the common parameters:</p> <ul style="list-style-type: none"> <li>-Verbose</li> <li>-Debug</li> <li>-ErrorAction</li> <li>-ErrorVariable</li> <li>-OutVariable</li> </ul> <p>For more information, type:</p> <ul style="list-style-type: none"> <li>get-help</li> <li>about_commonparameters</li> </ul>		

## Input Type

Processes with the *Name* or *Id* property specified, or all processes if neither is specified.

## Return Type

None by default. If *-passthru* is specified, the content object is passed through the pipeline.

## Examples

The following list provides examples for the `Disable-TEOProcess` command.

- This command disables all processes on the server.
 

```
C:\PS>Disable-TEOProcess
```
- This command disables the processes *Restart Windows Server* and *Run Antivirus Checks*. The command uses the *Name* parameter to specify the processes, but omits the optional parameter name.
 

```
C:\PS>Disable-TEOProcess "Restart Windows Server", "Run Antivirus Checks"
```
- This command disables processes by their process IDs.
 

```
C:\PS>Disable-TEOProcess -id "235DB653-BBB7-4CAA-97D3-1F4DA4DF0474",
"07964CBD-F168-43C7-90A5-57684B60B169"
```
- These commands disable all processes on the server that have names start with "Run" and display the processes in a tabular format.
 

```
C:\PS>$a = Get-TEOProcess "Run*"
Disable-TEOProcess -inputobject $a -passThru | format-table
```

## Disable-TEOProcess Help

The following are the help commandlets for TEO PowerShell CLI Snap-in.

Type	Commandlet
General information	Get-Help Disable-TEOProcess -detailed
Technical information	Get-Help Disable-TEOProcess -full
Parameters information	Get-Help About_commonparameters

## Enable-TEOProcess Commandlet

The *Enable-TEOProcess* commandlet enables processes that are on the server.

Users can identify a particular process by process name or process ID, or pass a process configuration through the pipeline to *Enable-TEOProcess*. For *Enable-TEOProcess*, the default method is by process name.

### Syntax

- Enable-TEOProcess [[-name] <String[]>] [<CommonParameters>]
- Enable-TEOProcess -id <Guid[]> [<CommonParameters>]
- Enable-TEOProcess -inputObject <ProcessConfiguration[]> [<CommonParameters>]
- Enable-TEOProcess -passThru <ProcessConfiguration[]> [<CommonParameters>]
- Enable-TEOProcess -confirm [<CommonParameters>]
- Enable-TEOProcess [-whatIf] [<CommonParameters>]

### Enable-TEOProcess Parameters

When specifying multiple values for a parameter, use commas to separate the values.

#### Example:

```
<parameter-name> <value1>, <value2>
```

Users cannot use the *-Name*, *-ID*, and *-InputObject* parameters in the same command.

Without parameters, *Enable-TEOProcess* enables all of the processes on the server, as though the user typed: `Enable-TEOProcess *`.

Parameter	Description	Option	Default Value
-name <String[]>	Specifies one or more processes by name.  To specify multiple processes, use commas to separate the process names or use wildcard characters.  The parameter name <i>-Name</i> is optional.	Required?	false
		Position?	1
		Default value	Null
		Accept pipeline input?	true (ByPropertyName)
		Accept wildcard characters?	true
-id <Guid[]>	Specifies one or more processes by process ID.  To specify multiple IDs, use commas to separate the IDs.  To find the ID of a process, type <code>Get-TEOProcess</code> .	Required?	true
		Position?	named
		Default value	Null
		Accept pipeline input?	
		Accept wildcard characters?	
-inputObject <ProcessConfiguration[]>	Accepts process configurations as input to the commandlet.  Enter a variable that contains the objects or type a command or expression that gets the objects.	Required?	true
		Position?	named
		Default value	Null
		Accept pipeline input?	
		Accept wildcard characters?	
-passThru <SwitchParameter>	Passes the object created by this commandlet through the pipeline.  By default, this commandlet does not pass any objects through the pipeline.	Required?	false
		Position?	named
		Default value	False
		Accept pipeline input?	false
		Accept wildcard characters?	false

-inputObject	Accepts process configurations as input to the commandlet.	Required?	true
<ProcessConfiguration[ ]>	Enter a variable that contains the objects or type a command or expression that gets the objects.	Position?	named
		Default value	Null
		Accept pipeline input?	
		Accept wildcard characters?	
-passThru	Passes the object created by this commandlet through the pipeline.	Required?	false
<SwitchParameter>	By default, this commandlet does not pass any objects through the pipeline.	Position?	named
		Default value	False
		Accept pipeline input?	false
		Accept wildcard characters?	false

## Input Type

Processes with the *Name* or *Id* property specified, or all processes if neither is specified.

## Return Type

None by default. If *-passthru* is specified, the content object is passed through the pipeline.

## Examples

- This command enables all processes on the server.
 

```
C:\PS>Enable-TEOProcess
```
- This command enables the processes *Restart Windows Server* and *Run Antivirus Checks*. It uses the *Name* parameter to specify the processes, but omits the optional parameter name.
 

```
C:\PS>Enable-TEOProcess "Restart Windows Server", "Run Antivirus Checks"
```
- This command identifies the processes by their process IDs.
 

```
Enable-TEOProcess -id "235DB653-BBB7-4CAA-97D3-1F4DA4DF0474",
"07964CBD-F168-43C7-90A5-57684B60B169"
```
- These commands enable all processes on the server that have names start with "Run" and display the process in a tabular format.
 

```
C:\PS>$a = Get-TEOProcess "Run*"
Enable-TEOProcess -inputobject $a -passThru | format-table
```

## Enable-TEOProcess Help

The following are the help commandlets for TEO PowerShell CLI Snap-in.

Type	Commandlet
General information	Get-Help Enable-TEOProcess -detailed
Technical information	Get-Help Enable-TEOProcess -full
Parameters information	Get-Help About_commonparameters

## Get-TEOProcess Commandlet

The *Get-TEOProcess* commandlet retrieves a process configuration for each process. Users can identify a particular process by process name or process ID, or pass a process configuration through the pipeline to *Get-TEOProcess*. For *Get-TEOProcess*, the default method is by process name.

## Syntax

- Get-TEOProcess [[-name] <String[]>] [<CommonParameters>]
- Get-TEOProcess -id <Guid[]> [<CommonParameters>]
- Get-TEOProcess -inputObject <ProcessConfiguration[]> [<CommonParameters>]

## Get-TEOProcess Parameters

When specifying multiple values for a parameter, use commas to separate the values.

### Example:

```
<parameter-name> <value1>, <value2>
```

Users cannot use the *-Name*, *-ID*, and *-InputObject* parameters in the same command.

Without parameters, *Get-TEOProcess* gets all of the processes on the server, as though the user typed:

```
Get-TEOProcess *
```

Parameter	Description	Option	Default Value
-name <String[]>	Specifies one or more processes by name.  To specify multiple processes, use commas to separate the process names or use wildcard characters.  The parameter name <i>-Name</i> is optional.	Required?	false
		Position?	1
		Default value	Null
		Accept pipeline input?	true (ByPropertyName)
		Accept wildcard characters?	true
-id <Guid[]>	Specifies one or more processes by process ID.  To specify multiple IDs, use commas to separate the IDs.  To find the ID of a process, type <i>Get-TEOProcess</i> .	Required?	true
		Position?	named
		Default value	Null
		Accept pipeline input?	true (ByPropertyName)
		Accept wildcard characters?	false
-inputObject <ProcessConfiguration[]>	Accepts process configurations as input to the commandlet.  Enter a variable that contains the objects or type a command or expression that gets the objects.	Required?	true
		Position?	named
		Default value	Null
		Accept pipeline input?	true (ByValue)
		Accept wildcard characters?	false
<CommonParameters>	This commandlet supports the common parameters:  -Verbose -Debug -ErrorAction -ErrorVariable -OutVariable  For more information, type: <i>get-help</i> <i>about_commonparameters</i>		

## Input Type

Processes with the *Name* or *Id* property specified, or all processes if neither is specified.

## Return Type

Process Configurations

## Examples

- This command gets the configurations of all of the processes on the server.

```
C:\PS>Get-TEOProcess
```

- This command gets the configurations for the processes *Restart Windows Server* and *Run Antivirus Checks*. It uses the *Name* parameter to specify the processes, but it omits the optional parameter name.

The pipeline operator (*|*) passes the data to the *Format-List* commandlet, which displays the properties of the *Restart Windows Server* and *Run Antivirus Checks* process configurations.

```
C:\PS>Get-TEOProcess "Restart Windows Server", "Run Antivirus Checks" | format-list
```

- This command identifies the processes by their process IDs.

```
Get-TEOProcess -id "235DB653-BBB7-4CAA-97D3-1F4DA4DF0474",
"07964CBD-F168-43C7-90A5-57684B60B169"
```

- This command gets the configurations for all the disabled processes. It uses the *Get-TEOProcess* commandlet to get the configurations of all the processes. The pipeline operator (*|*) passes the process configurations to the *Where-Object* commandlet, which selects only the object with the *Enabled* property is set to false (*0*).

```
C:\PS>Get-TEOProcess | where-object {$_.Enabled -eq 0}
```

- These commands list the configurations of the processes on the server grouped by the *Enabled* status. The first command gets the configuration of all of the processes on the server and stores them in the *\$a* variable.

The second command uses the *InputObject* parameter to pass the process configurations stored in *\$a* to *Get-TEOProcess*. The pipeline operator passes the objects to the *Format-Table* commandlet, which formats the processes by using the *Enabled* view.

```
C:\PS>$a = Get-TEOProcess
```

```
Get-TEOProcess -inputobject $a | format-table -view Enabled
```

## Get-TEOProcess Help

The following are the help commandlets for TEO PowerShell CLI Snap-in.

Type	Commandlet
General information	<code>Get-Help Get-TEOProcess -detailed</code>
Technical information	<code>Get-Help Get-TEOProcess -full</code>
Parameters information	<code>Get-Help About_commonparameters</code>

## Get-TEOProcess Instance Commandlet

The *Get-TEOProcessInstance* commandlet retrieves a process instances for each process. Users can identify a particular process by process name or process ID, or pass a process object through the pipeline to *Get-TEOProcessInstance*. For *Get-TEOProcessInstance*, the default method is by process name.

### Syntax

- `Get-TEOProcessInstance [-name] <Guid[]> [<CommonParameters>]`
- `Get-TEOProcessInstance -id <Guid[]> [<CommonParameters>]`
- `Get-TEOProcessInstance -inputObject <ProcessConfiguration[]> [<CommonParameters>]`
- `Get-TEOProcessInstance [-beginTime <DateTime>] [<CommonParameters>]`
- `Get-TEOProcessInstance [-endTime <DateTime>] [<CommonParameters>]`

### Get-TEOProcess Parameters

When specifying multiple values for a parameter, use commas to separate the values.

#### Example:

```
<parameter-name> <value1>, <value2>
```

Users cannot use the *-Name*, *-ID*, and *-InputObject* parameters in the same command.

Without parameters, *Get-TEOProcessInstance* gets all of the process instances on the server since midnight, as though the user typed: `Get-TEOProcessInstance *`.

Parameter	Description	Option	Default Value
-name <String[]>	Specifies one or more processes by name.  To specify multiple processes, use commas to separate the process names or use wildcard characters.  The parameter name <i>-Name</i> is optional.	Required?	false
		Position?	1
		Default value	Null
		Accept pipeline input?	true (ByPropertyName)
		Accept wildcard characters?	true
-id <Guid[]>	Specifies one or more processes by process ID.  To specify multiple IDs, use commas to separate the IDs.  To find the ID of a process, type <code>Get-TEOProcess</code> .	Required?	true
		Position?	named
		Default value	Null
		Accept pipeline input?	true (ByPropertyName)
		Accept wildcard characters?	false

Parameter	Description	Option	Default Value
-inputObject <ProcessConfiguration[ ]>	Accepts process configurations as input to the commandlet.  Enter a variable that contains the objects or type a command or expression that gets the objects.	Required?	true
		Position?	named
		Default value	Null
		Accept pipeline input?	true (Byvalue)
		Accept wildcard characters?	false
-beginTime <ProcessConfiguration[ ]>	The beginning of the time range that the process instances are created within.  It defaults to midnight of the current date.	Required?	false
		Position?	named
		Default value	Midnight
		Accept pipeline input?	true (Byvalue)
		Accept wildcard characters?	false
-endTime <ProcessConfiguration[ ]>	The end of the time range that the process instances are created within.  It defaults to the current time.	Required?	false
		Position?	named
		Default value	Current time
		Accept pipeline input?	true (Byvalue)
		Accept wildcard characters?	false
<CommonParameters>	This commandlet supports the common parameters:  -Verbose -Debug -ErrorAction -ErrorVariable -OutVariable  For more information, type: get-help about_commonparameters		

## Input Type

Processes with the *Name* or *Id* property specified, or all processes if neither is specified.

## Return Type

Process Instance Configuration Object(s)

## Examples

- This command gets a list of all of the process instances on the server since midnight.

```
C:\PS>Get-TEOProcessInstance
```

For a definition of each column, see additional notes in *Get-Help Get-TEOProcessInstance -Full*.

- This command gets the process instance configurations for the processes *Restart Windows Server* and *Run Antivirus Checks* that are started since midnight. It uses the *Name* parameter to specify the processes, but omits the optional parameter name.

The pipeline operator (|) passes the data to the *Format-List* commandlet, which displays the properties of the *Restart Windows Server* and *Run Antivirus Checks* process instance configurations.

```
C:\PS>Get-TEOProcessInstance "Restart Windows Server", "Run Antivirus Checks" |
format-list
```

- This command identifies the processes by their process IDs.

```
Get-TEOProcessInstance -id "235DB653-BBB7-4CAA-97D3-1F4DA4DF0474",
"07964CBD-F168-43C7-90A5-57684B60B169"
```

- This command gets all process instances since midnight that are failed. It uses the *Get-TEOProcessInstance* commandlet to get all processes since midnight.

The pipeline operator (|) passes the process instance configurations to the *Where-Object* commandlet, which selects only the object with the *State* property matches 'Failed\*'.  
 Note: The *Where-Object* commandlet is not available in PowerShell Core 6.0.0 and earlier.

```
C:\PS>Get-TEOProcessInstance | where-object {$_.State -like 'Failed*'}
```

- These commands list the process instances on the server since midnight grouped them by *State*. The first command gets all of the processes on the computer and stores them in the *\$a* variable. The second command uses the *InputObject* parameter to pass the process objects stored in *\$a* to *Get-ProcessInstance*. The pipeline operator passes the objects to the *Format-Table* commandlet, which formats the process instances by using the *State* view.

```
C:\PS>$a = Get-TEOProcess
```

```
Get-TEOProcessInstance -inputobject $a | format-table -view State
```

## Input Type

Processes with the *Name* or *Id* property specified, or all processes if neither is specified.

## Return Type

Process Instance Configuration Object(s)

## Get-TEOProcessInstance Help

The following are the help commandlets for TEO PowerShell CLI Snap-in.

Type	Commandlet
General information	Get-Help Get-TEOProcessInstance -detailed
Technical information	Get-Help Get-TEOProcessInstance -full
Parameters information	Get-Help About_commonparameters

## Start-TEOProcess Commandlet

The *Start-TEOProcess* commandlet starts a process. Users can identify a particular process by process name or process ID, or pass a process configuration through the pipeline to *Start-TEOProcess*. For *Start-TEOProcess*, the default method is by process name.

### Syntax

- Start-TEOProcess [-name] <String> [<CommonParameters>]
- Start-TEOProcess -id <Guid> [<CommonParameters>]
- Start-TEOProcess -inputObject <ProcessConfiguration> [<CommonParameters>]
- Start-TEOProcess -passThru <ProcessInstanceConfiguration[]> [<CommonParameters>]
- Start-TEOProcess -confirm [<CommonParameters>]
- Start-TEOProcess -waitForCompletion <> [<CommonParameters>]
- Start-TEOProcess -timeout <> [<CommonParameters>]
- Start-TEOProcess -inputVariables <> [<CommonParameters>]
- Start-TEOProcess [-whatIf] [<CommonParameters>]

### Start-TEOProcess Parameters

When specifying multiple values for a parameter, use commas to separate the values.

#### Example:

```
<parameter-name> <value1>, <value2>
```

Users cannot use the *-Name*, *-ID*, and *-InputObject* parameters in the same command.

Parameter	Description	Option	Default Value
-name <String>	Specifies a processes by name.  The parameter name <i>-Name</i> is optional.	Required?	true
		Position?	1
		Default value	Null
		Accept pipeline input?	true (ByPropertyName)
		Accept wildcard characters?	true
-id <Guid>	Specifies a processes by process ID.  To find the ID of a process, type <i>Get-TEOProcess</i> .	Required?	true
		Position?	named
		Default value	Null
		Accept pipeline input?	
		Accept wildcard characters?	

Parameter	Description	Option	Default Value
-inputObject <ProcessConfiguration >	Accepts process configurations as input to the commandlet.  Enter a variable that contains the objects or type a command or expression that gets the objects.	Required?	true
		Position?	named
		Default value	Null
		Accept pipeline input?	
		Accept wildcard characters?	
-passThru <SwitchParameter>	Passes the process instances created by this commandlet through the pipeline.  By default, this commandlet does not pass any objects through the pipeline.	Required?	false
		Position?	named
		Default value	
		Accept pipeline input?	false
		Accept wildcard characters?	false
-confirm	Prompts the user for confirmation before executing the command	Required?	false
		Position?	named
		Default value	
		Accept pipeline input?	false
		Accept wildcard characters?	false
-waitForCompletion	Wait for the process to finish execution	Required?	false
		Position?	named
		Default value	
		Accept pipeline input?	false
		Accept wildcard characters?	false
-timeout	The timeout (in seconds) for the commandlet to wait for the process to finish executing	Required?	false
		Position?	named
		Default value	
		Accept pipeline input?	false
		Accept wildcard characters?	false

Parameter	Description	Option	Default Value
-inputObject <ProcessConfiguration >	Accepts process configurations as input to the commandlet.  Enter a variable that contains the objects or type a command or expression that gets the objects.	Required?	true
		Position?	named
		Default value	Null
		Accept pipeline input?	
		Accept wildcard characters?	
-passThru <SwitchParameter>	Passes the process instances created by this commandlet through the pipeline.  By default, this commandlet does not pass any objects through the pipeline.	Required?	false
		Position?	named
		Default value	
		Accept pipeline input?	false
		Accept wildcard characters?	false
-confirm	Prompts the user for confirmation before executing the command	Required?	false
		Position?	named
		Default value	
		Accept pipeline input?	false
		Accept wildcard characters?	false
-waitForCompletion	Wait for the process to finish execution	Required?	false
		Position?	named
		Default value	
		Accept pipeline input?	false
		Accept wildcard characters?	false
-timeout	The timeout (in seconds) for the commandlet to wait for the process to finish executing	Required?	false
		Position?	named
		Default value	
		Accept pipeline input?	false
		Accept wildcard characters?	false

Parameter	Description	Option	Default Value
-inputObject <ProcessConfiguration >	Accepts process configurations as input to the commandlet.  Enter a variable that contains the objects or type a command or expression that gets the objects.	Required?	true
		Position?	named
		Default value	Null
		Accept pipeline input?	
		Accept wildcard characters?	
-passThru <SwitchParameter>	Passes the process instances created by this commandlet through the pipeline.  By default, this commandlet does not pass any objects through the pipeline.	Required?	false
		Position?	named
		Default value	
		Accept pipeline input?	false
		Accept wildcard characters?	false
-confirm	Prompts the user for confirmation before executing the command	Required?	false
		Position?	named
		Default value	
		Accept pipeline input?	false
		Accept wildcard characters?	false
-waitForCompletion	Wait for the process to finish execution	Required?	false
		Position?	named
		Default value	
		Accept pipeline input?	false
		Accept wildcard characters?	false
-timeout	The timeout (in seconds) for the commandlet to wait for the process to finish executing	Required?	false
		Position?	named
		Default value	
		Accept pipeline input?	false
		Accept wildcard characters?	false

Parameter	Description	Option	Default Value
-inputVariables	List of input parameters (Name=Value) for the process  <b>Note</b> "variable=value" is NOT the same as "variable = value"  The added space after "variable" is treated like a character in the variable name	Required?	false
		Position?	named
		Default value	
		Accept pipeline input?	false
		Accept wildcard characters?	false

## Input Type

Processes with the *Name* or *Id* property specified

## Return Type

None by default. If *-passthru* is specified, the content object is passed through the pipeline.

## Examples

- This command starts the process *Run Antivirus Checks*. It uses the *Name* parameter to specify the process, but omits the optional parameter name.

```
C:\PS>Start-TEOProcess 'Run Antivirus Checks'
```

- This command identifies the process by its ID.

```
Start-TEOProcess -id "07964CBD-F168-43C7-90A5-57684B60B169"
```

- This command starts the process *Run Antivirus Checks* and wait up to 120 seconds for the process to finish. The created process instance will be passed through the pipeline.

If the process is not finished executing within 120 seconds. The commandlet will return. The created process instance will still be passed through the pipeline.

To wait until the process is finished without timeout, do not specify the timeout parameter.

```
C:\PS>Start-TEOProcess "Run Antivirus Checks" -waitForCompletion -timeout 120 -passThru
```

- This command will prompt the user for a confirmation on running a process. Then, it executes the process with the specified ID, passing in two input variables, a string and a number.

```
C:\PS>Start-TEOProcess -id "07964CBD-F168-43C7-90A5-57684B60B169" -confirm -inputVariables 'VariableName1=String Value', 'VariableName2=500'
```

- These commands starts the *Run Antivirus Checks* process. wait for it to complete, and pass the process instance through the pipeline to be displayed in a tabular format.

```
C:\PS>$a = Get-TEOProcess "Run Antivirus Checks"
Start-TEOProcess -inputobject $a -waitForCompletion -passThru | format-table
```

## Start-TEOProcess Help

The following are the help commandlets for TEO PowerShell CLI Snap-in.

Type	Commandlet
General information	Get-Help Start-TEOProcess -detailed
Technical information	Get-Help Start-TEOProcess -full
Parameters information	Get-Help About_commonparameters

## Stop-TEOProcessInstance Commandlet

The *Stop-TEOProcessInstance* commandlet stops running process instances. Users can identify a particular process by process ID, or pass a process instance configuration through the pipeline to *Stop-TEOProcessInstance*. For *Stop-TEOProcessInstance*, the default method is by process ID.

### Syntax

- Stop-TEOProcessInstance -id <Guid[]> [<CommonParameters>]
- Stop-TEOProcessInstance -inputObject <ProcessInstanceConfiguration[]> [<CommonParameters>]
- Stop-TEOProcessInstance -passThru <ProcessInstanceConfiguration[]> [<CommonParameters>]
- Stop-TEOProcessInstance -confirm [<CommonParameters>]
- Stop-TEOProcessInstance [-whatIf] [<CommonParameters>]

### Stop-TEOProcessInstance Parameters

When specifying multiple values for a parameter, use commas to separate the values.

#### Example:

```
<parameter-name> <value1>, <value2>
```

Users cannot use the *-Name*, *-ID*, and *-InputObject* parameters in the same command.

Parameter	Description	Option	Default Value
-id <Guid[]>	Specifies a processes by process ID. To find the ID of a process, type <code>Get-TEOProcess</code> .	Required?	true
		Position?	named
		Default value	Null
		Accept pipeline input?	
		Accept wildcard characters?	

Parameter	Description	Option	Default Value
-inputObject <ProcessConfiguration[> >	Accepts process configurations as input to the commandlet.  Enter a variable that contains the objects or type a command or expression that gets the objects.	Required?	true
		Position?	named
		Default value	Null
		Accept pipeline input?	
		Accept wildcard characters?	
-passThru <SwitchParameter>	Passes the process instances created by this commandlet through the pipeline.  By default, this commandlet does not pass any objects through the pipeline.	Required?	false
		Position?	named
		Default value	False
		Accept pipeline input?	false
		Accept wildcard characters?	false
-confirm	Prompts the user for confirmation before executing the command	Required?	false
		Position?	named
		Default value	
		Accept pipeline input?	false
		Accept wildcard characters?	false

## Input Type

Processes with the *Id* property specified

## Return Type

None by default. If *-passthru* is specified, the content object is passed through the pipeline.

## Examples

- This command stops a running process instance with the specified ID.

```
C:\PS>Stop-TEOProcessInstance "21B5346D-6E17-45d5-8CDD-6E147E491524"
```

- These commands stop all running process instances and display the process instances in a tabular format.

```
C:\PS>$a = Get-TEOProcessInstnsance | where-object {$_.State -eq 'Running'}
Stop-TEOProcessInstance -inputobject $a -passThru | format-table
```

## Stop-TEOProcessInstance Help

The following are the help commandlets for TEO PowerShell CLI Snap-in.

Type	Commandlet
General information	<code>Get-Help Stop-TEOProcessInstance -detailed</code>
Technical information	<code>Get-Help Stop-TEOProcessInstance -full</code>
Parameters information	<code>Get-Help About_commonparameters</code>

# Target Commandlets

The CLI provide a command line argument to display the list of targets defined in TEO server.

The default display of a target is a table that includes the following columns:

Column	Description
Name	The name of the target
ID	The ID of the target
Enabled	<i>True</i> if the target is enabled, and <i>False</i> otherwise

## Disable-TEOTarget Commandlet

The *Disable-TEOTarget* commandlet disables targets. Users can identify a particular target by its name or ID, or pass a target configuration through the pipeline to *Disable-TEOTarget*. For *Disable-TEOTarget*, the default method is by target name.

### Syntax

- `Disable-TEOTarget [[-name] <String[]>] [<CommonParameters>]`
- `Disable-TEOTarget -id <Guid[]> [<CommonParameters>]`
- `Disable-TEOTarget -inputObject <TargetConfiguration[]> [<CommonParameters>]`
- `Disable-TEOTarget -passThru <TargetConfiguration[]> [<CommonParameters>]`
- `Disable-TEOTarget -confirm [<CommonParameters>]`
- `Disable-TEOTarget [-whatIf] [<CommonParameters>]`

### Disable-TEOTarget Parameters

When specifying multiple values for a parameter, use commas to separate the values.

#### Example:

```
<parameter-name> <value1>, <value2>
```

Users cannot use the *-Name*, *-ID*, and *-InputObject* parameters in the same command.

Without parameters, *Disable-TEOTarget* gets all of the process instances on the server since midnight, as though the user typed: `Disable-TEOTarget *`.

Parameter	Description	Option	Default Value
-name <String[]>	Specifies one or more targets by target name.  To specify multiple targets, use commas to separate the target names or use wildcard characters.  The parameter name <i>-Name</i> is optional.	Required?	false
		Position?	1
		Default value	Null
		Accept pipeline input?	true (ByPropertyName)
		Accept wildcard characters?	true
-id <Guid[]>	Specifies one or more targets by target ID.  To specify multiple IDs, use commas to separate the IDs.  To find the ID of a process, type <code>Get-TEOTarget</code> .	Required?	true
		Position?	named
		Default value	Null
		Accept pipeline input?	
		Accept wildcard characters?	
-inputObject <TargetConfiguration[]>	Accepts target configurations as input to the commandlet.  Enter a variable that contains the objects or type a command or expression that gets the objects.	Required?	true
		Position?	named
		Default value	Null
		Accept pipeline input?	
		Accept wildcard characters?	
-passThru <SwitchParameter>	Passes the process instances created by this commandlet through the pipeline.  By default, this commandlet does not pass any objects through the pipeline.	Required?	false
		Position?	named
		Default value	False
		Accept pipeline input?	false
		Accept wildcard characters?	false

Parameter	Description	Option	Default Value
-confirm	Prompts the user for confirmation before executing the command	Required?	false
		Position?	named
		Default value	
		Accept pipeline input?	false
		Accept wildcard characters?	false
<CommonParameters>	<p>This commandlet supports the common parameters:</p> <ul style="list-style-type: none"> <li>-Verbose</li> <li>-Debug</li> <li>-ErrorAction</li> <li>-ErrorVariable</li> <li>-OutVariable</li> </ul> <p>For more information, type:</p> <ul style="list-style-type: none"> <li>get-help</li> <li>about_commonparameters</li> </ul>		

## Input Type

Targets with the *Name* or *Id* property specified, or all targets if neither is specified.

## Return Type

None by default. If *-passthru* is specified, the content object is passed through the pipeline.

## Examples

- This command disables all targets on the server.
 

```
C:\PS>Disable-TEOTarget
```
- This command disables the targets *MyComputer1* and *MyComputer*. It uses the *Name* parameter to specify the targets, but it omits the optional parameter name.
 

```
C:\PS>Disable-TEOTarget "MyComputer1", "MyComputer2"
```
- This command identifies the targets by their target IDs.
 

```
"Disable-TEOTarget -id "235DB653-BBB7-4CAA-97D3-1F4DA4DF0474",
"07964CBD-F168-43C7-90A5-57684B60B169" "
```
- These commands disable all targets on the server that have names start with *My* and display the target configuration in a tabular format.
 

```
C:\PS>$a = Get-TEOTarget "My*"
Disable-TEOTarget -inputobject $a -passThru | format-table
```

## Disable-TEOTarget Help

The following are the help commandlets for TEO PowerShell CLI Snap-in.

Type	Commandlet
General information	Get-Help Disable-TEOTarget -detailed
Technical information	Get-Help Disable-TEOTarget -full
Parameters information	Get-Help About_commonparameters

## Enable-TEOTarget Commandlet

The *Enable-TEOTarget* commandlet enables targets. Without parameters, *Enable-TEOTargets* enables all of the targets on the server, as though the user typed: `Enable-TEOTarget *`.

You can also identify a particular target by its name or ID, or pass a target configuration through the pipeline to *Enable-TEOTarget*. For *Enable-TEOTarget*, the default method is by target name.

### Syntax

- `Enable-TEOTarget [-name] <String[]> [<CommonParameters>]`
- `Enable-TEOTarget -id <Guid[]> [<CommonParameters>]`
- `Enable-TEOTarget -inputObject <TargetConfiguration[]> [<CommonParameters>]`
- `Enable-TEOTarget -passThru <TargetConfiguration[]> [<CommonParameters>]`
- `Enable-TEOTarget -confirm [<CommonParameters>]`
- `Enable-TEOTarget [-whatIf] [<CommonParameters>]`

### Enable-TEOTarget Parameters

When specifying multiple values for a parameter, use commas to separate the values.

#### Example:

```
<parameter-name> <value1>, <value2>
```

Users cannot use the *-Name*, *-ID*, and *-InputObject* parameters in the same command.

Without parameters, *Enable-TEOTarget* gets all of the targets on the server since midnight, as though the user typed: `Enable-TEOTarget *`.

Parameter	Description	Option	Default Value
-name <String[]>	Specifies one or more targets by target name.  To specify multiple targets, use commas to separate the target names or use wildcard characters.  The parameter name <i>-Name</i> is optional.	Required?	false
		Position?	1
		Default value	Null
		Accept pipeline input?	true (ByPropertyName)
		Accept wildcard characters?	true
-id <Guid[]>	Specifies one or more targets by target ID.  To specify multiple IDs, use commas to separate the IDs.  To find the ID of a process, type <code>Get-TEOTarget</code> .	Required?	true
		Position?	named
		Default value	Null
		Accept pipeline input?	
		Accept wildcard characters?	
-inputObject <TargetConfiguration[]>	Accepts target configurations as input to the commandlet.  Enter a variable that contains the objects or type a command or expression that gets the objects.	Required?	true
		Position?	named
		Default value	Null
		Accept pipeline input?	
		Accept wildcard characters?	
-passThru <SwitchParameter>	Passes the process instances created by this commandlet through the pipeline.  By default, this commandlet does not pass any objects through the pipeline.	Required?	false
		Position?	named
		Default value	False
		Accept pipeline input?	false
		Accept wildcard characters?	false

-inputObject	Accepts target configurations as input to the commandlet.	Required?	true
<TargetConfiguration[]>		Position?	named
>	Enter a variable that contains the objects or type a command or expression that gets the objects.	Default value	Null
		Accept pipeline input?	
		Accept wildcard characters?	
-passThru	Passes the process instances created by this commandlet through the pipeline.	Required?	false
<SwitchParameter>		Position?	named
		Default value	False
	By default, this commandlet does not pass any objects through the pipeline.	Accept pipeline input?	false
		Accept wildcard characters?	false

## Input Type

Targets with the *Name* or *Id* property specified, or all targets if neither is specified.

## Return Type

None by default. If *-passthru* is specified, the content object is passed through the pipeline.

## Examples

- This command enables all targets on the server.  

```
C:\PS>Enable-TEOTarget
```
- This command enables the targets *MyComputer1* and *MyComputer2*. It uses the *Name* parameter to specify the targets, but it omits the optional parameter name.  

```
C:\PS>Enable-TEOTarget "MyComputer1", "MyComputer2"
```
- This command identifies the targets by their target IDs.  

```
"Enable-TEOTarget -id " 235DB653-BBB7-4CAA-97D3-1F4DA4DF0474",
"07964CBD-F168-43C7-90A5-57684B60B169" "
```
- These commands enable all targets on the server that have names start with "My" and display the target configuration in a tabular format.  

```
C:\PS>$a = Get-TEOTarget "My*" Enable-TEOTarget -inputobject $a -passThru |
format-table
```

## Enable-TEOTarget Help

The following are the help commandlets for TEO PowerShell CLI Snap-in.

Type	Commandlet
General information	Get-Help Enable-TEOTarget -detailed
Technical information	Get-Help Enable-TEOTarget -full
Parameters information	Get-Help About_commonparameters

## Get-TEOTarget Commandlet

The *Get-TEOTarget* commandlet retrieves a target configuration for each target. Users can also identify a particular target by its name or ID, or pass a target configuration through the pipeline to *Get-TEOTarget*. For *Get-TEOTarget*, the default method is by target name.

## Syntax

- Get-TEOTarget [[-name] <String[]>] [<CommonParameters>]
- Get-TEOTarget -id <Guid[]> [<CommonParameters>]
- Get-TEOTarget -inputObject <TargetConfiguration[]> [<CommonParameters>]

## Get-TEOTarget Parameters

When specifying multiple values for a parameter, use commas to separate the values.

### Example:

```
<parameter-name> <value1>, <value2>
```

Users cannot use the *-Name*, *-ID*, and *-InputObject* parameters in the same command.

Without parameters, *Get-TEOTarget* gets all of the targets on the server since midnight, as though the user typed: `Get-TEOTarget *`.

Parameter	Description	Option	Default Value
-name <String[]>	Specifies one or more targets by target name.  To specify multiple targets, use commas to separate the target names or use wildcard characters.  The parameter name <i>-Name</i> is optional.	Required?	false
		Position?	1
		Default value	Null
		Accept pipeline input?	true (ByPropertyName)
		Accept wildcard characters?	true
-id <Guid[]>	Specifies one or more targets by target ID.  To specify multiple IDs, use commas to separate the IDs.  To find the ID of a process, type <code>Get-TEOTarget</code> .	Required?	true
		Position?	named
		Default value	Null
		Accept pipeline input?	true (ByPropertyName)
		Accept wildcard characters?	false
-inputObject <TargetConfiguration[]>	Accepts target configurations as input to the commandlet.  Enter a variable that contains the objects or type a command or expression that gets the objects.	Required?	true
		Position?	named
		Default value	Null
		Accept pipeline input?	true (ByValue)
		Accept wildcard characters?	false
<CommonParameters>	This commandlet supports the common parameters:  -Verbose -Debug -ErrorAction -ErrorVariable -OutVariable  For more information, type: <code>get-help</code> <code>about_commonparameters</code>		

## Input Type

Targets with the *Name* or *Id* property specified, or all targets if neither is specified.

## Return Type

Target Configurations

## Examples

- This command gets the configurations of all of the targets on the server. For a definition of each column, see *Get-Help Get-TEOTarget -Full*.

```
C:\PS>Get-TEOTarget
```

- This command gets the configurations for the targets *MyComputer1* and *MyComputer2*. It uses the *Name* parameter to specify the targets, but it omits the optional parameter name.

The pipeline operator (|) passes the data to the *Format-List* commandlet, which displays the properties of the *MyComputer1* and *MyComputer2* target configurations.

```
C:\PS>Get-TEOTarget "MyComputer1", "MyComputer2" | format-list
```

- Users can identify the targets by their target IDs.

```
"Get-TEOTarget -id "235DB653-BBB7-4CAA-97D3-987648ABEDF4",
"07964CBD-F168-43C7-90A5-57684598D439"
```

- This command gets the configurations for all the disabled targets. It uses the *Get-TEOTarget* commandlet to get the configurations of all the targets.

The pipeline operator (|) passes the target configurations to the *Where-Object* commandlet, which selects only the object with the *Enabled* property is set to false (0).

```
C:\PS>Get-TEOTarget | where-object {$_.Enabled -eq 0}
```

- These commands list the configurations of the targets on the server grouped by the *Enabled* status. The first command gets the configuration of all of the targets on the server and stores them in the *\$a* variable.

The second command uses the *InputObject* parameter to pass the target configurations stored in *\$a* to *Get-TEOTarget*. The pipeline operator passes the objects to the *Format-Table* commandlet, which formats the targets by using the *Enabled* view.

```
C:\PS>$a = Get-TEOTarget
Get-TEOTarget -inputobject $a | format-table -view Enabled
```

## Get-TEOTarget Help

The following are the help commandlets for TEO PowerShell CLI Snap-in.

Type	Commandlet
General information	<i>Get-Help Get-TEOTarget -detailed</i>
Technical information	<i>Get-Help Get-TEOTarget -full</i>
Parameters information	<i>Get-Help About_commonparameters</i>





## INDEX

---

### A

Adding TEO PowerShell CLI Snap-In [1-3](#)

---

### C

Command Line Errors [2-2](#)

Command Line Syntax [2-2](#)

Command Parameters [2-2](#)

---

### D

Disable-TEOProcess [2-3](#)

Disable TEOProcess Examples [2-5](#)

Disable-TEOProcess Help [2-6](#)

Disable-TEOProcess Input Type [2-5](#)

Disable-TEOProcess Parameters [2-3](#)

Disable-TEOProcess Return Type [2-5](#)

Disable-TEOProcess Syntax [2-3](#)

Disable-TEOTarget Commandlet [2-23](#)

Disable-TEOTarget Examples [2-25](#)

Disable-TEOTarget Help [2-26](#)

Disable-TEOTarget Input Type [2-25](#)

Disable-TEOTarget Parameters [2-23](#)

Disable-TEOTarget Return Type [2-25](#)

Disable-TEOTarget Syntax [2-23](#)

---

### E

Enable-TEOProcess [2-6](#)

Enable-TEOProcess Examples [2-9](#)

Enable-TEOProcess Help [2-9](#)

Enable-TEOProcess Input Type [2-8](#)

Enable-TEOProcess Parameters [2-6](#)

Enable-TEOProcess Return Type [2-8](#)

Enable-TEOProcess Syntax [2-6](#)

Enable-TEOTarget Commandlet [2-26](#)

Enable-TEOTarget Examples [2-29](#)

Enable-TEOTarget Help [2-29](#)

Enable-TEOTarget Input Type [2-28](#)

Enable-TEOTarget Parameters [2-26](#)

Enable-TEOTarget Return Type [2-28](#)

Enable-TEOTarget Syntax [2-26](#)

Examples [2-5, 2-9, 2-11, 2-14, 2-19, 2-21, 2-25, 2-29](#)

---

### G

Get-TEOProcess Commandlet [2-9](#)

Get-TEOProcess Examples [2-11](#)

Get-TEOProcess Help [2-11](#)

Get-TEOProcess Input Type [2-10](#)

Get-TEOProcess Instance Commandlet [2-12](#)

Get-TEOProcess Instance Examples [2-14](#)

Get-TEOProcess Instance Help [2-14](#)

Get-TEOProcess Instance Input Type [2-13, 2-14](#)

Get-TEOProcess Instance Parameters [2-12](#)

Get-TEOProcess Instance Return Type [2-13, 2-14](#)

Get-TEOProcess Instance Syntax [2-12](#)

Get-TEOProcess Parameters [2-9](#)

Get-TEOProcess Return Type [2-11](#)

Get-TEOProcess Syntax [2-9](#)

Get-TEOTarget Commandlet [2-29](#)

Get-TEOTarget Examples [2-31](#)

Get-TEOTarget Help [2-31](#)

Get-TEOTarget Input Type [2-30](#)

Get-TEOTarget Parameters [2-29](#)

Get-TEOTarget Return Type [2-31](#)

Get-TEOTarget Syntax [2-29](#)

## H

Help [2-6, 2-9, 2-11, 2-14, 2-20, 2-22, 2-26, 2-29, 2-31](#)

## I

Input Type [2-5, 2-8, 2-10, 2-13, 2-19, 2-21, 2-25, 2-28, 2-30](#)

Installing TEO PowerShell CLI Snap-In [1-1](#)

## P

Parameters [2-9, 2-12, 2-15, 2-20, 2-23, 2-26, 2-29](#)

PowerShell 32-bit Support [1-2](#)

PowerShell CLI Snap-In System Requirements [1-2](#)

PowerShell Snap-In DLL Registration [1-3](#)

Process Commandlets [2-3](#)

## R

Register PowerShell Snap-In DLL [1-3](#)

Remote Sign [1-3](#)

Return Type [2-5, 2-8, 2-11, 2-13, 2-14, 2-19, 2-21, 2-25, 2-28](#)

## S

Set Execution Policy [1-3](#)

Start-TEOProcess Commandlet [2-15](#)

Start-TEOProcess Examples [2-19](#)

Start-TEOProcess Help [2-20](#)

Start-TEOProcess Input Type [2-19](#)

Start-TEOProcess Parameters [2-15](#)

Start-TEOProcess Return Type [2-19](#)

Start-TEOProcess Syntax [2-15](#)

Stop-TEOProcessInstance Commandlet [2-20](#)

Stop-TEOProcessInstance Examples [2-21](#)

Stop-TEOProcessInstance Help [2-22](#)

Stop-TEOProcessInstance Input Type [2-21](#)

Stop-TEOProcessInstance Parameters [2-20](#)

Stop-TEOProcessInstance Return Type [2-21](#)

Stop-TEOProcessInstance Syntax [2-20](#)

Supported Platforms [1-2](#)

Syntax [2-6, 2-9, 2-12, 2-15, 2-20, 2-23, 2-26, 2-29](#)

## T

Target Commandlets [2-23](#)

TEO PowerShell CLI Snap-In Shortcut [1-3](#)

## U

Understanding TEO Commandlets [2-1](#)