Table Of Contents

Chapter 1

Activity Parameters

The following table lists the recommended names and functionality for activity parameters.

Append
Data type: SwitchParameter

Implement this parameter so that the user can add content to the end of a resource when the parameter is specified.

CaseSensitive
Data type: SwitchParameter

Implement this parameter so the user can require case sensitivity when the parameter is specified.

Command
Data type: String

Implement this parameter so the user can specify a command string to run.

CompatibleVersion
Data type: System.Version object

Implement this parameter so the user can specify the semantics that the cmdlet must be compatible with for compatibility with previous versions.

Compress
Data type: SwitchParameter

Implement this parameter so that data compression is used when the parameter is specified.

Compress
Data type: Keyword

Implement this parameter so that the user can specify the algorithm to use for data compression.

Continuous
Data type: SwitchParameter

Implement this parameter so that data is processed until the user terminates the cmdlet when the parameter is specified. If the parameter is not specified, the cmdlet processes a predefined amount of data and then terminates the operation.

Create
Data type: SwitchParameter

Implement this parameter to indicate that a resource is created if one does not already exist when the parameter is specified.

Delete
Data type: SwitchParameter

Implement this parameter so that resources are deleted when the cmdlet has completed its operation when the parameter is specified.

Drain
Data type: SwitchParameter

Implement this parameter to indicate that outstanding work items are processed before the cmdlet processes new data when the parameter is specified. If the parameter is not specified, the work items are processed immediately.

Erase
Data type: Int32

Implement this parameter so that the user can specify the number of times a resource is erased before it is deleted.

ErrorLevel
Data type: Int32

Implement this parameter so that the user can specify the level of errors to report.

Exclude
Data type: String[]

Implement this parameter so that the user can exclude something from an activity. For more information about how to use input filters, see Input Filter Parameters.

Filter
Data type: Keyword

Implement this parameter so that the user can specify a filter that selects the resources upon which to perform the cmdlet action. For more information about how to use input filters, see Input Filter Parameters.

Follow
Data type: SwitchParameter

Implement this parameter so that progress is tracked when the parameter is specified.

Force
Data type: SwitchParameter

Implement this parameter to indicate that the user can perform an action even if restrictions are encountered when the parameter is specified. The parameter does not allow security to be compromised. For example, this parameter lets a user overwrite a read-only file.

Include
Data type: String[]

Implement this parameter so that the user can include something in an activity. For more information about how to use input filters, see Input Filter Parameters.

Incremental
Data type: SwitchParameter

Implement this parameter to indicate that processing is performed incrementally when the parameter is specified. For example, this parameter lets a user perform incremental backups that back up files only since the last backup.

InputObject
Data type: Object

Implement this parameter when the cmdlet takes input from other cmdlets. When you define an *InputObject* parameter, always specify the **ValueFromPipeline** keyword when you declare the **Parameter** attribute. For more information about using input filters, see Input Filter Parameters.

Insert
Data type: SwitchParameter

Implement this parameter so that the cmdlet inserts an item when the parameter is specified.

Interactive
Data type: SwitchParameter

Implement this parameter so that the cmdlet works interactively with the user when the parameter is specified.

Interval
Data type: HashTable

Implement this parameter so that the user can specify a hash table of keywords that contains the values. The following example shows sample values for the *Interval* parameter: `-interval @{ResumeScan=15; Retry=3}`.

Log
Data type: SwitchParameter

Implement this parameter audit the actions of the cmdlet when the parameter is specified.

NoClobber
Data type: SwitchParameter

Implement this parameter so that the resource will not be overwritten when the parameter is specified. This parameter generally applies to cmdlets that create new objects so that they can be prevented from overwriting existing objects with the same name.

Notify
Data type: SwitchParameter

Implement this parameter so that the user will be notified that the activity is complete when the parameter is specified.

NotifyAddress
Data type: E-mail address

Implement this parameter so that the user can specify the e-mail address to use to send a notification when the *Notify* parameter is specified.

Overwrite
Data type: SwitchParameter

Implement this parameter so that the cmdlet overwrites any existing data when the parameter is specified.

Prompt
Data type: String

Implement this parameter so that the user can specify a prompt for the cmdlet.

Quiet
Data type: SwitchParameter

Implement this parameter so that the cmdlet suppresses user feedback during its actions when the parameter is specified.

Recurse
Data type: SwitchParameter

Implement this parameter so that the cmdlet recursively performs its actions on resources when the parameter is specified.

Repair
Data type: SwitchParameter

Implement this parameter so that the cmdlet will attempt to correct something from a broken state when the parameter is specified.

RepairString
Data type: String

Implement this parameter so that the user can specify a string to use when the *Repair* parameter is specified.

Retry
Data type: Int32

Implement this parameter so the user can specify the number of times the cmdlet will attempt an action.

Select
Data type: Keyword array

Implement this parameter so that the user can specify an array of the types of items.

Stream
Data type: SwitchParameter

Implement this parameter so the user can stream multiple output objects through the pipeline when the parameter is specified.

Strict
Data type: SwitchParameter

Implement this parameter so that all errors are handled as terminating errors when the parameter is specified.

TempLocation
Data type: String

Implement this parameter so the user can specify the location of temporary data that is used during the operation of the cmdlet.

Timeout
Data type: Int32

Implement this parameter so that the user can specify the timeout interval (in milliseconds).

Truncate
Data type: SwitchParameter

Implement this parameter so that the cmdlet will truncate its actions when the parameter is specified. If the parameter is not specified, the cmdlet performs another action.

Verify
Data type: SwitchParameter

Implement this parameter so that the cmdlet will test to determine whether an action has occurred when the parameter is specified.

Wait
Data type: SwitchParameter

Implement this parameter so that the cmdlet will wait for user input before continuing when the parameter is specified.

WaitTime
Data type: Int32

Implement this parameter so that the user can specify the duration (in seconds) that the cmdlet will wait for user input when the *Wait* parameter is specified.

See Also

Concepts
Windows PowerShell Reference
Other Resources
Cmdlet Parameters
Writing a Windows PowerShell Cmdlet

Date and Time Parameters

The following table lists recommended names and functionality for parameters that handle date and time information. Date and time parameters are typically used to record when something is created or accessed.

Accessed
Data type: SwitchParameter

Implement this parameter so that when it is specified the cmdlet will operate on the resources that have been accessed based on the date and time specified by the *Before* and *After* parameters.

If this parameter is specified, the *Created* and *Modified* parameters must be not be specified.

After
Data type: DateTime

Implement this parameter to specify the date and time after which the cmdlet was used. For the *After* parameter to work, the cmdlet must also have an *Accessed*, *Created*, or *Modified* parameter. And, that parameter must be set to **true** when the cmdlet is called.

Before
Data type: DateTime

Implement this parameter to specify the date and time before which the cmdlet was used. For the *Before* parameter to work, the cmdlet must also have an *Accessed*, *Created*, or *Modified* parameter. And, that parameter must be set to **true** when the cmdlet is called.

Created
Data type: SwitchParameter

Implement this parameter so that when it is specified the cmdlet will operate on the resources that have been created based on the date and time specified by the *Before* and *After* parameters.

If this parameter is specified, the *Accessed* and *Modified* parameters must not be specified.

Exact
Data type: SwitchParameter

Implement this parameter so that when it is specified the resource term must match the resource name exactly. When the parameter is not specified the resource term and name do not need to match exactly.

Modified
Data type: DateTime

Implement this parameter so that when it is specified the cmdlet will operate on resources that have been changed based on the date and time specified by the *Before* and *After* parameters.

If this parameter is specified, the *Accessed* and *Created* parameters must not be specified.

See Also

Concepts
Windows PowerShell Reference
Other Resources
Cmdlet Parameters
Writing a Windows PowerShell Cmdlet

Format Parameters

The following table lists recommended names and functionality for parameters that are used to format or to generate data.

As
Data type: Keyword

Implement this parameter to specify the cmdlet output format. For example, possible values could be Text or Script.

Binary
Data type: SwitchParameter

Implement this parameter to indicate that the cmdlet handles binary values.

Encoding
Data type: Keyword

Implement this parameter to specify the type of encoding that is supported. For example, possible values could be ASCII, UTF8, Unicode, UTF7, BigEndianUnicode, Byte, and String.

NewLine
Data type: SwitchParameter

Implement this parameter so that the newline characters are supported when the parameter is specified.

ShortName
Data type: SwitchParameter

Implement this parameter so that short names are supported when the parameter is specified.

Width
Data type: Int32

Implement this parameter so that the user can specify the width of the output device.

Wrap
Data type: SwitchParameter

Implement this parameter so that text wrapping is supported when the parameter is specified.

See Also

Concepts
Windows PowerShell Reference
Other Resources
Cmdlet Parameters
Writing a Windows PowerShell Cmdlet

Property Parameters

The following table lists the recommended names and functionality for property parameters.

Count
Data type: Int32

Implement this parameter so that the user can specify the number of objects to be processed.

Description
Data type: String

Implement this parameter so that the user can specify a description for a resource.

From
Data type: String

Implement this parameter so that the user can specify the reference object to get information from.

Id
Data type:

Implement this parameter so that the user can specify the identifier of a resource.

Input
Data type: String

Implement this parameter so that the user can specify the input file specification.

Location
Data type: String

Implement this parameter so that the user can specify the location of the resource.

LogName
Data type: String

Implement this parameter so that the user can specify the name of the log file to process or use.

Name
Data type: String

Implement this parameter so that the user can specify the name of the resource.

Output
Data type: String

Implement this parameter so that the user can specify the output file.

Owner

Data type: String

Implement this parameter so that the user can specify the name of the owner of the resource.

Property
Data type: String

Implement this parameter so that the user can specify the name or the names of the properties to use.

Reason
Data type: String

Implement this parameter so that the user can specify why this cmdlet is being invoked.

Regex
Data type: SwitchParameter

Implement this parameter so that regular expressions are used when the parameter is specified. When this parameter is specified, wildcard characters are not resolved.

Speed
Data type: Int32

Implement this parameter so that the user can specify the baud rate. The user sets this parameter to the speed of the resource.

State
Data type: Keyword array

Implement this parameter so that the user can specify the names of states, such as KEYDOWN.

Value
Data type: Object

Implement this parameter so that the user can specify a value to provide to the cmdlet.

Version
Data type: String

Implement this parameter so that the user can specify the version of the property.

See Also

Concepts
Windows PowerShell Reference
Other Resources
Cmdlet Parameters
Writing a Windows PowerShell Cmdlet

Quantity Parameters

The following table lists the recommended names and functionality for quantity parameters.

All
Data type: Boolean

Implement this parameter so that **true** indicates that all resources should be acted upon instead of a default subset of resources. Implement this parameter so that **false** indicates a subset of the resources.

Allocation
Data type: Int32

Implement this parameter so that the user can specify the number of items to allocate.

BlockCount
Data type: Int64

Implement this parameter so that the user can specify the block count.

Count
Data type: Int64

Implement this parameter so that the user can specify the count.

Scope
Data type: Keyword

Implement this parameter so that the user can specify the scope to operate on.

See Also

Concepts
Windows PowerShell Reference
Other Resources
Cmdlet Parameters
Writing a Windows PowerShell Cmdlet

Resource Parameters

The following table lists the recommended names and functionality for resource parameters. For these parameters, the resources could be the assembly that contains the cmdlet class or the host application that is running the cmdlet.

Application
Data type: String

Implement this parameter so that the user can specify an application.

Assembly
Data type: String

Implement this parameter so that the user can specify an assembly.

Attribute
Data type: String

Implement this parameter so that the user can specify an attribute.

Class
Data type: String

Implement this parameter so that the user can specify a Microsoft .NET Framework class.

Cluster
Data type: String

Implement this parameter so that the user can specify a cluster.

Culture
Data type: String

Implement this parameter so that the user can specify the culture in which to run the cmdlet.

Domain
Data type: String

Implement this parameter so that the user can specify the domain name.

Drive
Data type: String

Implement this parameter so that the user can specify a drive name.

Event
Data type: String

Implement this parameter so that the user can specify an event name.

Interface
Data type: String

Implement this parameter so that the user can specify a network interface name.

IpAddress
Data type: String

Implement this parameter so that the user can specify an IP address.

Job
Data type: String

Implement this parameter so that the user can specify a job.

LiteralPath
Data type: String

Implement this parameter so that the user can specify the path to a resource when wildcard characters are not supported. (Use the *Path* parameter when wildcard characters are supported.)

Mac
Data type: String

Implement this parameter so that the user can specify a media access controller (MAC) address.

ParentId
Data type: String

Implement this parameter so that the user can specify the parent identifier.

Path
Data type: String, String[]

Implement this parameter so that the user can indicate the paths to a resource when wildcard characters are supported. (Use the *LiteralPath* parameter when wildcard characters are not supported.)

We recommend that you develop this parameter so that it supports the full "provider:path" syntax used by providers. We also recommend that you develop it so that it works with as many providers as possible.

Port
Data type: Integer, String

Implement this parameter so that the user can specify an integer value for networking or a string value such as "biztalk" for other types of port.

Printer
Data type: Integer, String

Implement this parameter so that the user can specify the printer for the cmdlet to use.

Size
Data type: Int32

Implement this parameter so that the user can specify a size.

TID
Data type: String

Implement this parameter so that the user can specify a transaction identifier (TID) for the cmdlet.

Type

Data type: String

Implement this parameter so that the user can specify the type of resource on which to operate.

URL

Data type: String

Implement this parameter so that the user can specify a Uniform Resource Locator (URL).

User

Data type: String

Implement this parameter so that the user can specify their name or the name of another user.

See Also

Concepts
Windows PowerShell Reference
Other Resources
Cmdlet Parameters
Writing a Windows PowerShell Cmdlet

Security Parameters

The following table lists the recommended names and functionality for parameters used to provide security information for an operation, such as parameters that specify certificate key and privilege information.

ACL

Data type: String

Implement this parameter to specify the access control level of protection for a catalog or for a Uniform Resource Identifier (URI).

CertFile

Data type: String

Implement this parameter so that the user can specify the name of a file that contains one of the following:

- A Base64 or Distinguished Encoding Rules (DER) encoded x.509 certificate
- A Public Key Cryptography Standards (PKCS) #12 file that contains at least one certificate and key

CertIssuerName

Data type: String

Implement this parameter so that the user can specify the name of the issuer of a certificate or so that the user can specify a substring.

CertRequestFile

Data type: String

Implement this parameter to specify the name of a file that contains a Base64 or DER-encoded PKCS #10 certificate request.

CertSerialNumber

Data type: String

Implement this parameter to specify the serial number that was issued by the certification authority.

CertStoreLocation

Data type: String

Implement this parameter so that the user can specify the location of the certificate store. The location is typically a file path.

CertSubjectName

Data type: String

Implement this parameter so that the user can specify the issuer of a certificate or so that the user can specify a substring.

CertUsage

Data type: String

Implement this parameter to specify the key usage or the enhanced key usage. The key can be represented as a bit mask, a bit, an object identifier (OID), or a string.

Credential

Data type: PSCredential

Implement this parameter so that the cmdlet will automatically prompt the user for a user name or password. A prompt for both is displayed if a full credential is not supplied directly.

CSPName

Data type: String

Implement this parameter so that the user can specify the name of the certificate service provider (CSP).

CSPType

Data type: Integer

Implement this parameter so that the user can specify the type of CSP.

Group

Data type: String

Implement this parameter so that the user can specify a collection of principals for access. For more information, see the description of the *Principal* parameter.

KeyAlgorithm

Data type: String

Implement this parameter so that the user can specify the key generation algorithm to use for security.

KeyContainerName
Data type: String

Implement this parameter so that the user can specify the name of the key container.

KeyLength
Data type: Integer

Implement this parameter so that the user can specify the length of the key in bits.

Operation
Data type: String

Implement this parameter so that the user can specify an action that can be performed on a protected object.

Principal
Data type: String

Implement this parameter so that the user can specify a unique identifiable entity for access.

Privilege
Data type: String

Implement this parameter so that the user can specify the right a cmdlet needs to perform an operation for a particular entity.

Privileges
Data type: Array of privileges

Implement this parameter so that the user can specify the rights that a cmdlet needs to perform its operation for a particular entry.

Role
Data type: String

Implement this parameter so that the user can specify a set of operations that can be performed by an entity.

SaveCred
Data type: SwitchParameter

Implement this parameter so that credentials that were previously saved by the user will be used when the parameter is specified.

Scope
Data type: String

Implement this parameter so that the user can specify the group of protected objects for the cmdlet.

SID
Data type: String

Implement this parameter so that the user can specify a unique identifier that represents a principal.

Trusted
Data type: SwitchParameter

Implement this parameter so that trust levels are supported when the parameter is specified.

TrustLevel
Data type: Keyword

Implement this parameter so that the user can specify the trust level that is supported. For example, possible values include internet, intranet, and fulltrust.

See Also

Concepts
Windows PowerShell Reference
Other Resources
Cmdlet Parameters
Writing a Windows PowerShell Cmdlet