



**Full credit is given to all the above companies including the Operating System that this PDF file was generated!**

### ***Windows PowerShell Get-Help on Cmdlet 'ConvertTo-Xml'***

**PS:\>Get-HELP ConvertTo-Xml -Full**

#### **NAME**

ConvertTo-Xml

#### **SYNOPSIS**

Creates an XML-based representation of an object.

#### **SYNTAX**

```
ConvertTo-Xml [-InputObject] <System.Management.Automation.PSObject> [-As {Stream | String | Document}] [-Depth <System.Int32>] [-NoTypeInformation]  
[<CommonParameters>]
```

#### **DESCRIPTION**

The `ConvertTo-Xml` cmdlet creates an XML-based (/dotnet/api/system.xml.xmldocument) representation of one or more .NET objects. To use this cmdlet, pipe one or more objects to the cmdlet, or use the InputObject parameter to specify the object.

When you pipe multiple objects to `ConvertTo-Xml` or use the InputObject parameter to submit multiple objects, `ConvertTo-Xml` returns a single, in-memory XML

document that includes representations of all the objects.

This cmdlet is similar to Export-Clixml (./Export-Clixml.md) except that `Export-Clixml` stores the resulting XML in a Common Language Infrastructure (CLI)

(<https://www.ecma-international.org/publications-and-standards/standards/ecma-335/>) file that can be reimported as objects with Import-Clixml (./Import-Clixml.md).

`ConvertTo-Xml` returns an in-memory representation of an XML document, so you can continue to process it in PowerShell. `ConvertTo-Xml` doesn't have an option to convert objects to CLI XML.

## PARAMETERS

-As <System.String>

Determines the output format. The acceptable values for this parameter are:

- `String` - Returns a single string.
- `Stream` - Returns an array of strings.
- `Document` - Returns an XmlDocument object.

The default value is `Document`.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Depth <System.Int32>

Specifies how many levels of contained objects are included in the XML representation. The default value is 1.

For example, if the object's properties also contain objects, to save an XML representation of the properties

Page 2/5

contained objects, you must specify a depth  
of 2.

The default value can be overridden for the object type in the Types.ps1xml files. For more information, see  
[about\\_Types.ps1xml](#)

([..../Microsoft.PowerShell.Core/About/about\\_Types.ps1xml.md](#)).

Required? false  
Position? named  
Default value None  
Accept pipeline input? False  
Accept wildcard characters? false

-InputObject <System.Management.Automation.PSObject>

Specifies the object to be converted. Enter a variable that contains the objects, or type a command or expression that gets the objects. You can also pipe objects  
to `ConvertTo-XML`.

Required? true  
Position? 0  
Default value None  
Accept pipeline input? True (ByValue)  
Accept wildcard characters? false

-NoTypeInformation <System.Management.Automation.SwitchParameter>

Omits the Type attribute from the object nodes.

Required? false  
Position? named  
Default value False  
Accept pipeline input? False  
Accept wildcard characters? false

## <CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about\_CommonParameters (<https://go.microsoft.com/fwlink/?LinkId=113216>).

## INPUTS

System.Management.Automation.PSObject

You can pipe any object to this cmdlet

## OUTPUTS

System.String

When you use the As parameter and set the value to `string`, this cmdlet returns the XML as a string. When the value is `stream`, this cmdlet returns an array of strings.

System.Xml.XmlDocument

By default, this cmdlet returns an XML document.

## NOTES

----- Example 1: Convert a date to XML -----

Get-Date | ConvertTo-Xml

This command converts the current date (a DateTime object) to XML.

----- Example 2: Convert processes to XML -----

```
ConvertTo-Xml -As "Document" -InputObject (Get-Process) -Depth 3
```

This command converts the process objects that represent all the processes on the computer into an XML document. The objects are expanded to a depth of three levels.

## RELATED LINKS

Online

Version:

[https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/convertto-xml?view=powershell-5.1&WT.mc\\_id=ps-gethelp](https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/convertto-xml?view=powershell-5.1&WT.mc_id=ps-gethelp)

[ConvertTo-Csv](#)

[ConvertTo-Html](#)

[Export-Clixml](#)

[Get-Date](#)

[Import-Clixml](#)