



python



PowerShell

FPDF Library
PDF generator

Full credit is given to the above companies including the OS that this PDF file was generated!

PowerShell Get-Help on command 'ConvertTo-Json'

PS C:\Users\wahid> Get-Help ConvertTo-Json

NAME

ConvertTo-Json

SYNOPSIS

Converts an object to a JSON-formatted string.

SYNTAX

```
ConvertTo-Json [-InputObject] <System.Object> [-Compress] [-Depth  
<System.Int32>] [<CommonParameters>]
```

DESCRIPTION

The `ConvertTo-Json` cmdlet converts any .NET object to a string in JavaScript Object Notation (JSON) format. The properties are converted to field names, the field values are converted to property values, and the methods are removed.

You can then use the `ConvertFrom-Json` cmdlet to convert a JSON-formatted string to a JSON object, which is easily managed in PowerShell.

between servers and web-based apps.

This cmdlet was introduced in Windows PowerShell 3.0.

PARAMETERS

`-Compress <System.Management.Automation.SwitchParameter>`

Omits white space and indented formatting in the output string.

`-Depth <System.Int32>`

Specifies how many levels of contained objects are included in the JSON representation. The value can be any number from ``1`` to ``100``. The default value is ``2``. ``ConvertTo-Json`` emits a warning if the number of levels in an input object exceeds this number.

`-InputObject <System.Object>`

Specifies the objects to convert to JSON format. Enter a variable that contains the objects, or type a command or expression that gets the objects. You can also pipe an object to ``ConvertTo-Json``.

The `InputObject` parameter is required, but its value can be null (``$null``) or an empty string. When the input object is ``$null``, ``ConvertTo-Json`` does not generate any output. When the input object is an empty string, ``ConvertTo-Json`` returns an empty string.

`<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>).

----- Example 1 -----

(Get-UICulture).Calendar | ConvertTo-Json

```
{
  "MinSupportedDateTime": "\Date(-62135596800000)V",
  "MaxSupportedDateTime": "\Date(253402300799999)V",
  "AlgorithmType": 1,
  "CalendarType": 1,
  "Eras": [
    1
  ],
  "TwoDigitYearMax": 2029,
  "IsReadOnly": false
}
```

This command uses the `ConvertTo-Json` cmdlet to convert a `GregorianCalendar` object to a JSON-formatted string.

----- Example 2 -----

```
@{Account="User01";Domain="Domain01";Admin="True"} | ConvertTo-Json -Compress
```

```
{"Domain":"Domain01","Account":"User01","Admin":"True"}
```

This command shows the effect of using the `Compress` parameter of `ConvertTo-Json`. The compression affects only the appearance of the string, not its validity.

----- Example 3 -----

```
Get-Date | Select-Object -Property * | ConvertTo-Json
```

```
{
  "DisplayHint": 2,
  "DateTime": "Friday, January 13, 2012 8:06:16 PM",
  "Date": "\Date(1326441600000)V",
```

```

"Day": 13,
"DayOfWeek": 5,
"DayOfYear": 13,
"Hour": 20,
"Kind": 2,
"Millisecond": 221,
"Minute": 6,
"Month": 1,
"Second": 16,
"Ticks": 634620819762218083,
"TimeOfDay": {
    "Ticks": 723762218083,
    "Days": 0,
    "Hours": 20,
    "Milliseconds": 221,
    "Minutes": 6,
    "Seconds": 16,
    "TotalDays": 0.83768775241087956,
    "TotalHours": 20.104506057861109,
    "TotalMilliseconds": 72376221.8083,
    "TotalMinutes": 1206.2703634716668,
    "TotalSeconds": 72376.22180829999
},
"Year": 2012
}

```

This example uses the ``ConvertTo-Json`` cmdlet to convert a `System.DateTime` object from the ``Get-Date`` cmdlet to a JSON-formatted string. The command uses the ``Select-Object`` cmdlet to get all (`` ``) of the properties of the `DateTime` object. The output shows the JSON string that ``ConvertTo-Json`` returned.

----- Example 4 -----

DisplayHint : 2
DateTime : October 12, 2018 10:55:52 PM
Date : 2018-10-12 12:00:00 AM
Day : 12
DayOfWeek : 5
DayOfYear : 285
Hour : 22
Kind : 2
Millisecond : 768
Minute : 55
Month : 10
Second : 52
Ticks : 636749817527683372
TimeOfDay : @{Ticks=825527683372; Days=0; Hours=22; Milliseconds=768;
Minutes=55; Seconds=52;
TotalDays=0.95547185575463; TotalHours=22.9313245381111;
TotalMilliseconds=82552768.3372;
TotalMinutes=1375.87947228667; TotalSeconds=82552.7683372}
Year : 2018

This example shows how to use the `ConvertTo-Json`` and `ConvertFrom-Json`` cmdlets to convert an object to a JSON string and a JSON object.

REMARKS

To see the examples, type: "get-help ConvertTo-Json -examples".

For more information, type: "get-help ConvertTo-Json -detailed".

For technical information, type: "get-help ConvertTo-Json -full".

For online help, type: "get-help ConvertTo-Json -online"