

Sitecore PowerShell Extension 6.3

Contents

1. Install Sitecore Powershell Extension	--	3
2. Sitecore PowerShell Commands	--	12
3. Sitecore PowerShell Commands – Save Script & Use	--	30
4. Sitecore PowerShell Commands – Backup and Restore package Script	--	34
5. Sitecore PowerShell Commands – Bulk Update	--	40
6. Sitecore PowerShell Commands – UI of PowerShell	--	44
7. Sitecore PowerShell Commands – Remoting	--	46
8. Sitecore PowerShell Commands – Remoting Security	--	52
9. Unicorn	--	54
10. Sitecore PowerShell Commands with Unicorn, SPE, Remoting	--	69
11. Sitecore PowerShell Commands with Unicorn – Backup & Restore	--	80
12. Sitecore Migration with Unicorn 4.1.5, SPE 6.3 & Remoting	--	87
13. Appendix	--	89

1. Install Sitecore Powershell Extension

The screenshot shows the Sitecore Marketplace homepage. At the top, there's a navigation bar with links for 'SERVICES', 'SIGN IN', and 'DISCOVER' (which is highlighted in red). Below the header, a large banner says 'Discover Sitecore modules' with a magnifying glass icon. The main content area has three sections: 'Trend watch', 'Recommended', and 'Downloaded'. The 'Trend watch' section lists items like 'Sitecore Community' and 'Sitecore Retina Images'. The 'Recommended' section lists five modules: 'SITECORE INSTANCE MANAGER', 'SITECORE POWERSHELL CONSOLE', 'SITECORE LOG ANALYZER', 'SITECORE ROCKS', and 'ANALYTICS DATABASE MANAGER'. The 'Downloaded' section lists five modules: 'SITECORE POWERSHELL CONSOLE' (highlighted with a red border), 'SITECORE INSTANCE MANAGER', 'SITECORE LOG ANALYZER', 'ANALYTICS DATABASE MANAGER', and '301 REDIRECT MODULE'.

Rank	Module	Downloads
1.	SITECORE POWERSHELL CONSOLE	8933x
2.	SITECORE INSTANCE MANAGER	1931x
3.	SITECORE LOG ANALYZER	1638x
4.	ANALYTICS DATABASE MANAGER	583x
5.	301 REDIRECT MODULE	535x

1. Install Sitecore Powershell Extension

Download Location : <https://github.com/SitecorePowerShell/Console/releases/>

The screenshot shows a GitHub release page for the Sitecore PowerShell Extensions. At the top, there are links for 'All improvements' and 'All Fixes'. Below that is a section titled 'Known Issues' with a bulleted list:

- Same as issues documented in 6.0 release.
- You may encounter an error related to moving a dictionary item. If upgrading with SXA previously installed you should check out [#1185](#) for more details.
- [#1212](#) For existing installations you may encounter an issue where IIS has locked the folder containing the icons.

Below this is a 'File Hashes' section containing two lines of PowerShell code:

```
Get-FileHash -Path ".\Sitecore.PowerShell.Extensions-6.3.zip" -Algorithm SHA256  
Get-FileHash -Path ".\Sitecore.PowerShell.Extensions-6.3.scwdp.zip" -Algorithm SHA256
```

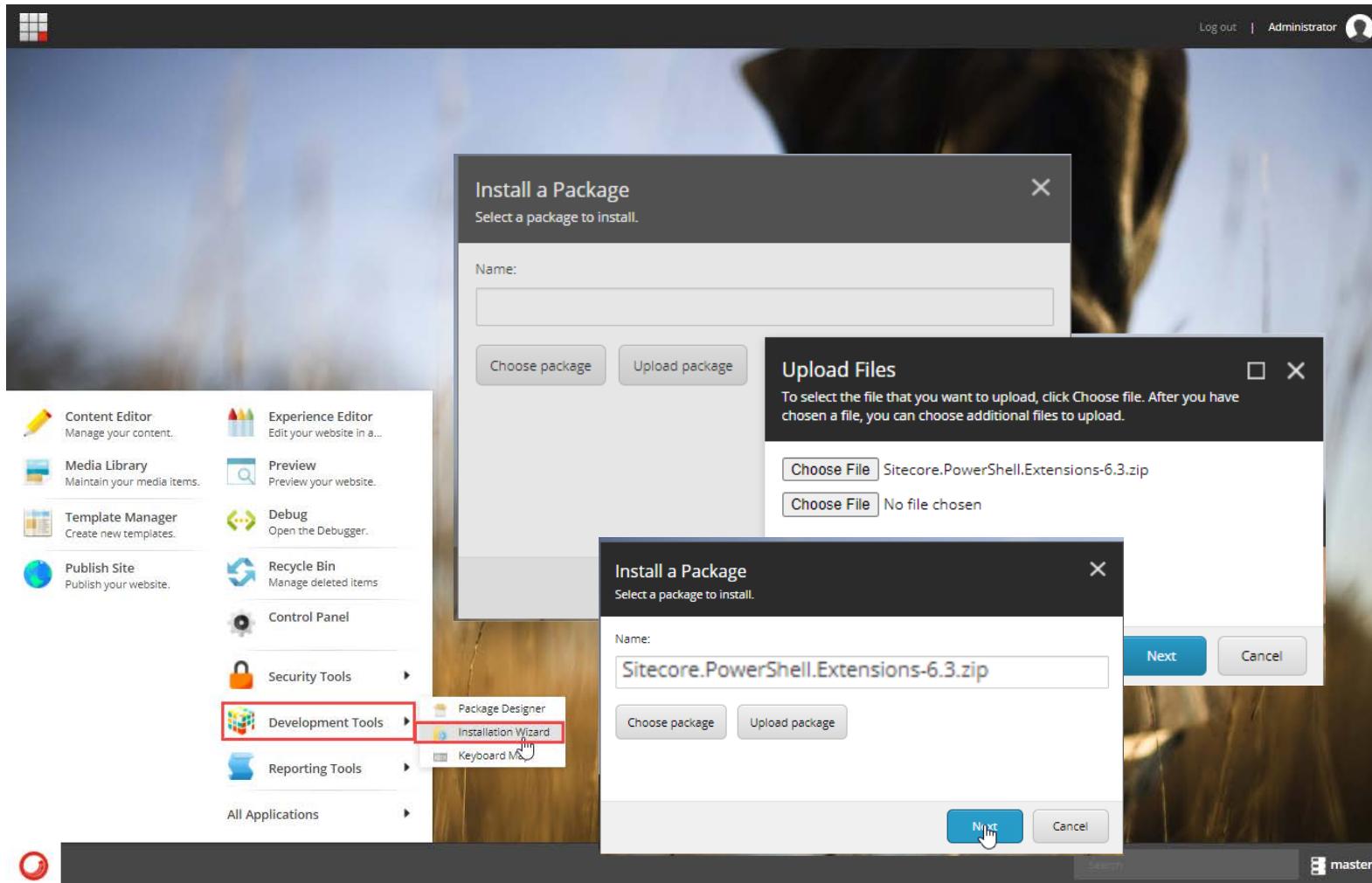
Under the 'Assets' heading, there are six items listed:

Asset	Size
Sitecore.PowerShell.Extensions-6.3.scwdp.zip	1.71 MB
Sitecore.PowerShell.Extensions-6.3.zip	5.02 MB
SPE.Minimal-6.3.zip	317 KB
SPE.Remoting-6.3.zip	32.8 KB
Source code (zip)	
Source code (tar.gz)	

The 'Sitecore.PowerShell.Extensions-6.3.zip' file is highlighted with a red box. At the bottom left, there is a '6.2' link and a note from user 'michaellwest' released on Oct 12, 2020. The bottom navigation bar shows 'Version 6.2'.

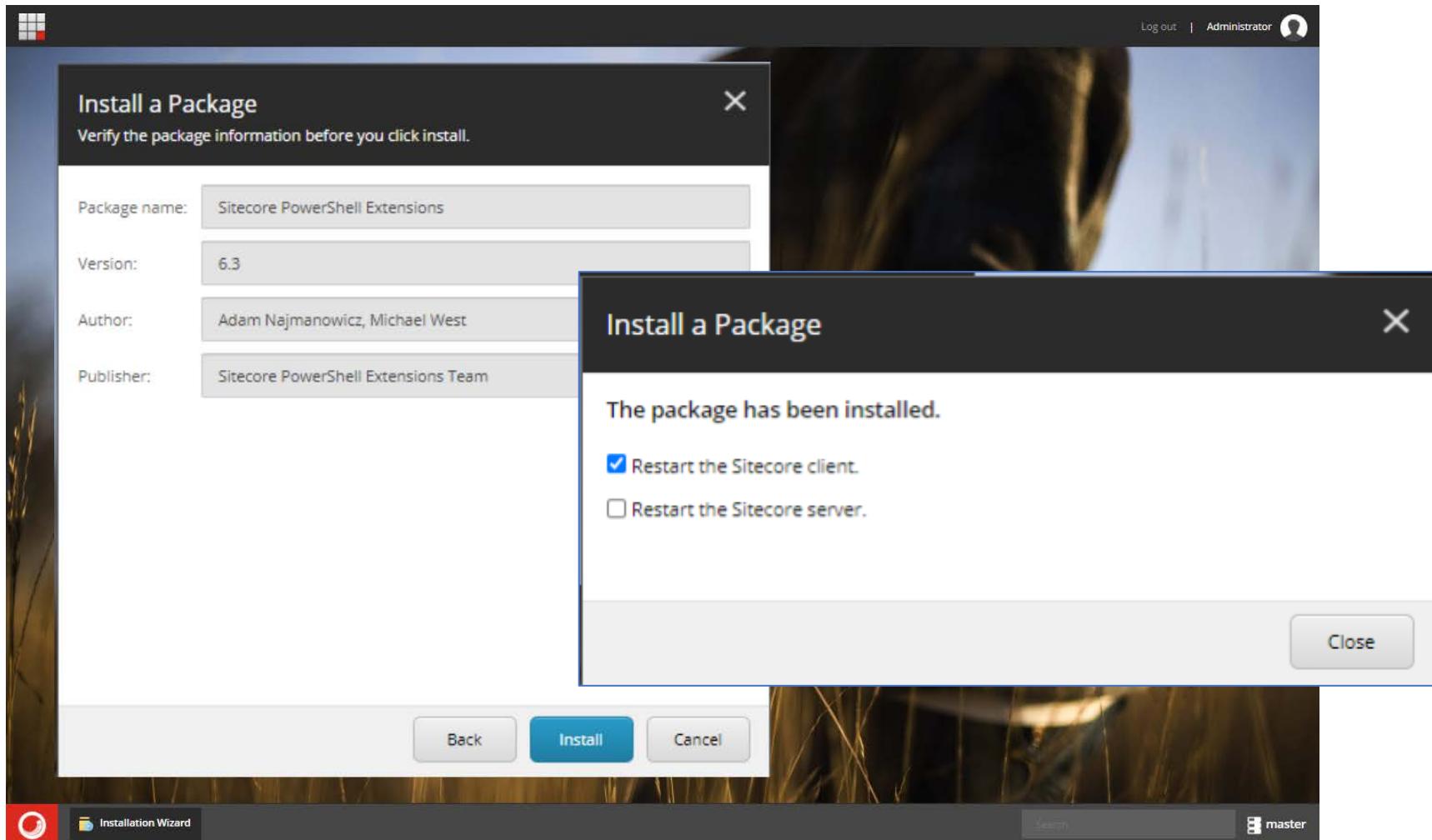
1. Install Sitecore Powershell Extension

Install “Sitecore.PowerShell.Extensions-6.3.zip” to Sitecore 10.1



1. Install Sitecore Powershell Extension

Install “Sitecore.PowerShell.Extensions-6.3.zip” to Sitecore 10.1



1. Install Sitecore Powershell Extension

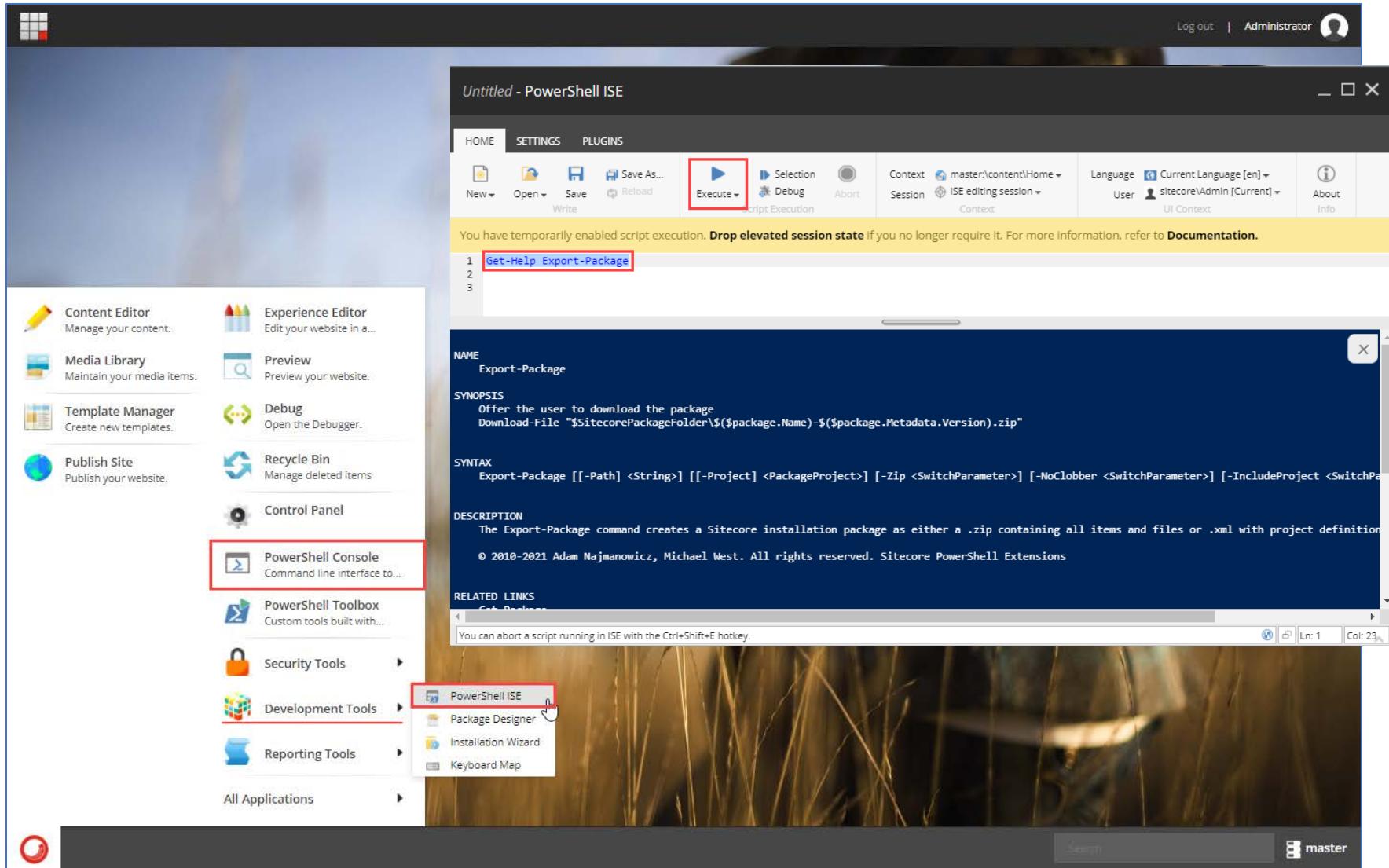
Enable “Spe.IdentityServer.config”

The screenshot shows a PowerShell console window titled "PowerShell Console". At the top, there is a message: "You have temporarily enabled script execution. Drop elevated session state if you no longer require it. For more information, refer to Documentation." Below this, the title "Sitecore PowerShell Extensions" and copyright information "Copyright © 2010-2021 Adam Najmanowicz, Michael West. All rights Reserved." are displayed. The command "PS master:\>" is shown at the bottom of the console.

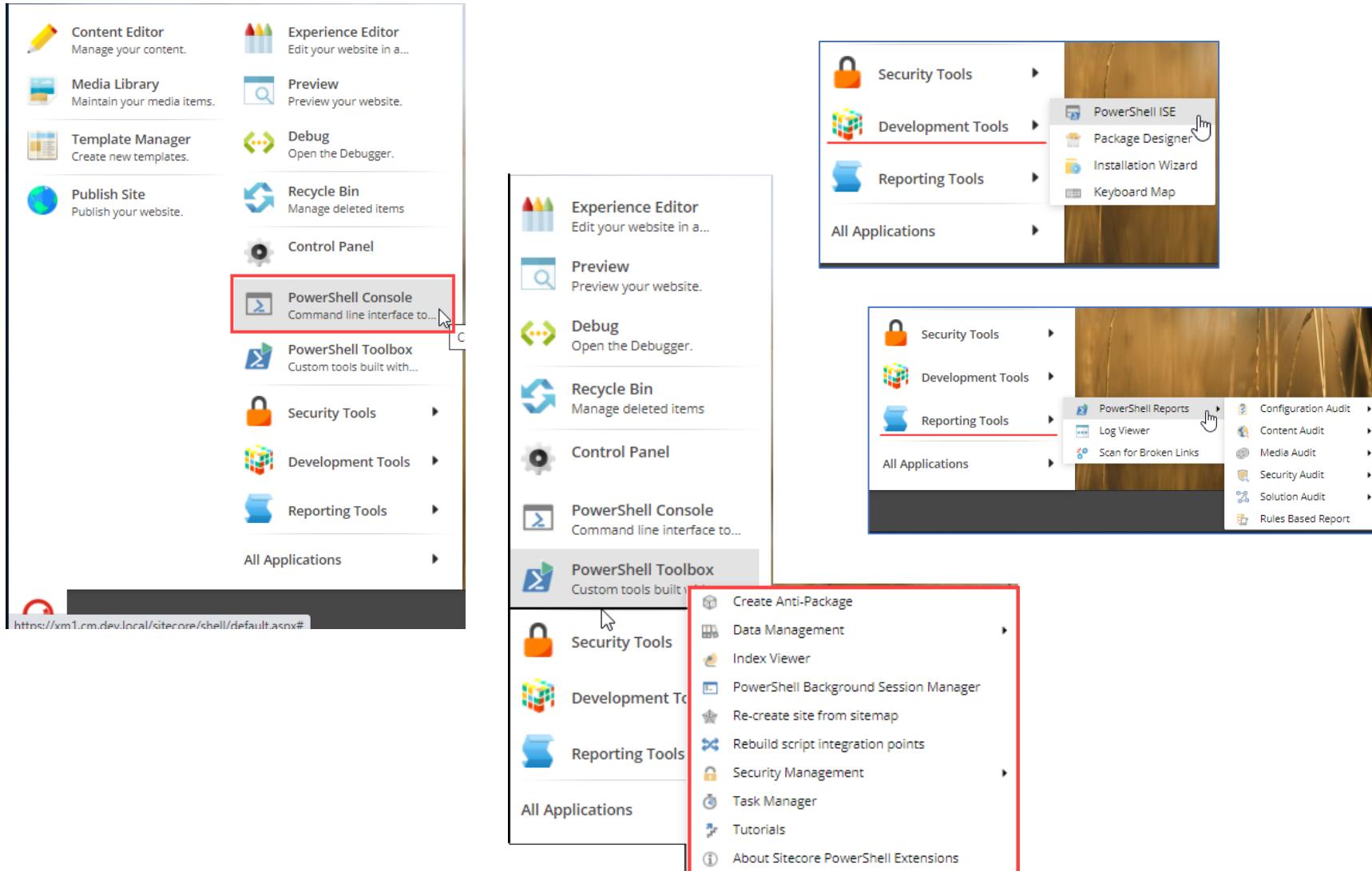
The main part of the screenshot is a file listing interface. The path "inetpub > wwwroot > XM1.cm.dev.local > App_Config > Include > Spe" is highlighted with a red border. Below this, a table lists several configuration files:

Name	Date modified	Type	Size
Spe.config	8/17/2021 2:40 PM	XML Configuration File	23 KB
Spe.Events.config.disabled	8/17/2021 2:40 PM	DISABLED File	5 KB
Spe.IdentityServer.config	8/17/2021 2:40 PM	XML Configuration File	2 KB
Spe.IdentityServer.config.disabled.Save	8/17/2021 2:40 PM	SAVE File	2 KB
Spe.Minimal.config.disabled	8/17/2021 2:40 PM	DISABLED File	1 KB

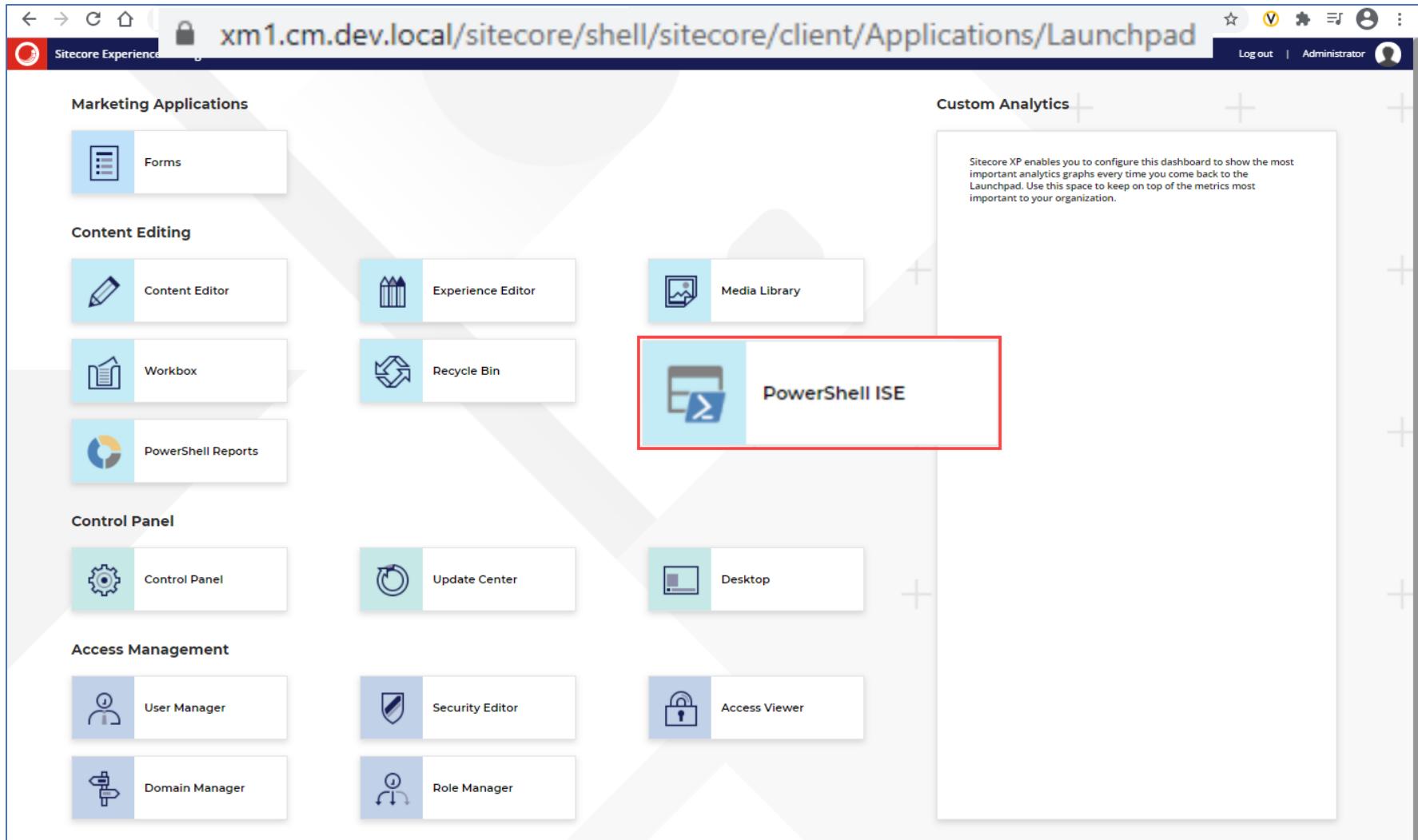
1. Install Sitecore Powershell Extension



1. Install Sitecore Powershell Extension



1. Install Sitecore Powershell Extension



1. Install Sitecore Powershell Extension

The screenshot shows the Sitecore PowerShell Extension interface in the PowerShell Integrated Scripting Environment (ISE). The top menu bar includes 'HOME', 'SETTINGS', and 'PLUGINS'. The 'PLUGINS' tab is active, indicated by a red box around it. The toolbar below has buttons for 'New', 'Open', 'Save', 'Save As...', 'Reload', 'Execute' (with dropdowns for 'Selection' and 'Debug'), 'Abort', 'Context' (showing 'master:\content\Home'), 'Session' (showing 'ISE editing session'), 'Language' (set to 'Current Language [en]'), and 'User' (set to 'sitecore\Admin [Current]'). A yellow status bar at the bottom says, 'You have temporarily enabled script execution. Drop elevated session state if you no longer require it. For more information, refer to Documentation.' The main pane displays PowerShell commands and their output:

```
1 Get-Help Format-Table
2 Get-ChildItem -path "/sitecore/content" -recurse
3

Name      : Home
DisplayName : Home
HasChildren : True
Language   : en
Version    : 1
ID        : {110D559F-DEA5-42EA-9C1C-8A5DF7E70EF9}
Database   : master
TemplateName: Sample Item
ItemPath   : /sitecore/content/Home

Name      : Getting Started with Sitecore Helix
DisplayName : Getting Started with Sitecore Helix
HasChildren : False
Language   : en
Version    : 1
ID        : {1D88A329-E1A1-44E3-9498-45EF7D00289E}
Database   : master
TemplateName: Default
ItemPath   : /sitecore/content/Home/Getting Started with Sitecore Helix

Name      : Global Datasources
DisplayName : Global Datasources
HasChildren : True
Language   : en
Version    : 1
ID        : {088308AC-184C-40EF-A675-D64EC435357A}
```

At the bottom, a note says, 'You can abort a script running in ISE with the Ctrl+Shift+E hotkey.'

2. Sitecore PowerShell Commands

Get-Help Get-Item -Full

Get-Help Foreach-Object -Verbose

Get-Help Export-Package -Examples

Get-Command - CommandType Cmdlet

Get-Command Get-*

Get-Member

Command	Description
<hr/>	
Get-Item	Returns an object at the specified path.
Get-ChildItem	Returns children at the specified path. Supports recursion.
Get-Help	Returns the help documentation for the specified command or document.
Get-Command	Returns a list of commands.
Get-Member	Returns the methods and properties for the specified object.
ForEach-Object	Enumerates over the objects passed through the pipeline.
Where-Object	Enumerates over the objects passed through the pipeline and filters objects.
Select-Object	Returns objects from the pipeline with the specified properties and filters objects.
Sort-Object	Sorts the pipeline objects with the specified criteria; usually a property name.

2. Sitecore PowerShell Commands

<https://doc.sitecorepowershell.com/working-with-items>

- PowerShell ISE : Integrated Scripting Environment

The screenshot shows the Sitecore PowerShell ISE interface. The title bar says "Untitled - PowerShell ISE". The menu bar includes HOME, SETTINGS, and PLUGINS. The toolbar has buttons for New, Open, Save, Save As..., Execute, Selection, Debug, Abort, and Reload. The status bar at the bottom says "You can abort a script running in ISE with the Ctrl+Shift+E hotkey." The main area displays the following content:

```
1 Get-Help Get-Item -Full
```

By path

```
Get-Item -Path master:\content\home
```

By Id

```
Get-Item -Path master: -ID "{110D559F-DEA5-42EA-9C1C-8A5DF7E70EF9}"
```

By Uri

```
Get-Item -Path master: -Uri "sitecore://master/{110D559F-DEA5-42EA-9C1C-8A5DF7E70EF9}?lang=en&ver=1"
```

With Sitecore query

```
Get-Item -Path master: -Query
"/sitecore/content//*[@@templatename='Sample Item']"
```

The left sidebar shows the cmdlet's properties:

- NAME**: Get-Item
- SYNTAX**:
Get-Item [-Path] <string[]> [-Filter <string>] [-Database <string>]
Get-Item -LiteralPath <string[]> [<string>] [-ID <string>] [-Database <string>]
- PARAMETERS**:
 - AmbiguousPaths
 - Required?: false
 - Position?: Named
 - Accept pipeline input?: false
 - Parameter set name: (All)
 - Aliases: None

2. Sitecore PowerShell Commands

Get-Item

The screenshot shows the PowerShell ISE interface with several examples of the `Get-Item` cmdlet.

Untitled (*) - PowerShell ISE

Script Execution

```
1 Get-Item -Path master:\content\home
2
3
```

Output Window

Name	Children	Language	Version	Id	TemplateName
Home	True	en	1	{110D559F-DEA5-42EA-9C1C-8A5DF7E70EF9}	Sample Item

Help Table

Name	Drives	Example
Alias	Alias	Get-Item -Path alias:\dir
CmsItemProvider	core, master, web	Get-Item -Path master:\
Environment	Env	Get-Item -Path env:\HOMEPATH
FileSystem	C, D, F, H	Get-Item -Path c:\Windows
Function	Function	Get-Item -Path function:\prompt
Registry	HKLM, HKCU	Get-Item -Path hklm:\SOFTWARE
Variable	Variable	Get-Item -Path variable:\PSVersionTable

Script Editor

```
1 Get-Item -Path master: -Uri "sitecore://master/{110D559F-DEA5-42EA-9C1C-8A5DF7E70EF9}?lang=en&ver=1"
```

Output Window

Name	Children	Language	Version	Id	TemplateName
Home	True	en	1	{110D559F-DEA5-42EA-9C1C-8A5DF7E70EF9}	Sample Item

2. Sitecore PowerShell Commands

Get-ChildItem

```
$DefaultPath = "/sitecore/content"
$NamePart = "Default"
```

```
Get-ChildItem -path $DefaultPath -recurse ` 
| Where-Object { $_.TemplateName -match $NamePart } ` 
| format-Table DisplayName, Name, TemplateName
```

```
1 $DefaultPath = "/sitecore/content"
2
3 Get-ChildItem -path $DefaultPath -recurse ` 
4 | format-Table DisplayName, Name, TemplateName
5 |
```

DisplayName	Name	TemplateName
-----	-----	-----
Home	Home	Sample Item
Getting Started with Sitecore Helix	Getting Started with Sitecore Helix	Default
Global Datasources	Global Datasources	Folder
Hero Image	Hero Image	Hero Image

```
1 $DefaultPath = "/sitecore/content"
2 $NamePart = "Default"
3
4 Get-ChildItem -path $DefaultPath -recurse ` 
5 | Where-Object { $_.TemplateName -match $NamePart } ` 
6 | format-Table DisplayName, Name, TemplateName
```

DisplayName	Name	TemplateName
-----	-----	-----
Getting Started with Sitecore Helix	Getting Started with Sitecore Helix	Default

Get-ChildItem Property

```
Name      : Getting Started with Sitecore Helix
DisplayName : Getting Started with Sitecore Helix
HasChildren : False
Language   : en
Version    : 1
ID         : {1D8BA329-E1A1-44E3-949B-45EF7D00289E}
Database   : master
TemplateName: Default
ItemPath   : /sitecore/content/Home/Getting Started with Sitecore Helix
```

2. Sitecore PowerShell Commands

- Get-Help Where-Object

2. Sitecore PowerShell Commands

- `-OrderBy <string>, -Property <string array>, -First <int>`

```
$TemplatePart = "layout"
$OrderBy = "Path"
$Fields = @("Name", "Language", "ItemId", "TemplateName", "Path")
Find-Item -Index sitecore_master_index -Property $Fields -OrderBy $OrderBy -First 10 ` 
| Where-Object { $_.TemplateName -match $TemplatePart }
```

--- Results ---

```
ItemId      : {5861BEF6-7848-4F6A-BB65-8FDFD098159E}
Language    : en
Name        : Website
Path         : /sitecore/layout/layouts/project/website
TemplateName : Layout Folder

ItemId      : {DA279311-B061-47C8-B26E-75F1BFA85EDF}
Language    : en
Name        : Default
Path         : /sitecore/layout/layouts/project/website/default
TemplateName : Layout
```

2. Sitecore PowerShell Commands

Untitled (*) - PowerShell ISE

HOME SETTINGS PLUGINS

New Open Save Save As... Execute Selection Abort

Reload Script Execution

Context master:\media library Session ISE editing session Context

Language Current Language [en] User sitecore\Admin [Current] UI Context

About Info

You have temporarily enabled script execution. **Drop elevated session state** if you no longer require it. For more information, refer to [Documentation](#).

```
1 Get-Help Format-Table
2 $TemplatePart = "Template"
3 Find-Item -Index sitecore_master_index `
4     | Where-Object { $_.TemplateName -match $TemplatePart }
5
```

NAME
Format-Table

SYNTAX
Format-Table [[-Property] <Object[]> [-AutoSize] [-HideTableHeaders] [-Wrap] [-GroupBy <Object>] [-View <string>] [-ShowError] [-DisplayError] [-Force] [-Expand {
[<CommonParameters>]

ALIASES
ft

REMARKS
None

"Get-Help Format-Table"
will show all field names

Name : Body Copy Content
ItemId : {8FE10CB5-0B38-4F10-9BAC-5E9D3685D4BD}
Language : en
Version : 1
DatabaseName : master
Path : /sitecore/templates/feature/pagecontent/body copy/body copy content
TemplateName : Template section

You can abort a script running in ISE with the Ctrl+Shift+E hotkey.

Ln: 5 Col: 0

2. Sitecore PowerShell Commands

Untitled (*) - PowerShell ISE

HOME SETTINGS PLUGINS

New Open Save Save As... Execute Selection Abort Context master:\media library Session ISE editing session Language Current Language [en] User sitecore\Admin [Current] Context UI Context About Info

You have temporarily enabled script execution. **Drop elevated session state** if you no longer require it. For more information, refer to [Documentation](#).

```
1 Get-Help Get-ChildItem
2
3 $path = "master:\content\home\"
4 Get-ChildItem -Path $path -Recurse |
5   Where-Object { $_.Name -like "*Sitecore*" } |
6     #Select-Object -Property Name, ID, database, ItemPath
7     Sort-Object -Property Name
```

SYNTAX

```
Get-ChildItem [[-Path] <string[]> [[-Filter] <string>] [-Include <string[]>] [-Exclude <string[]>] [-Recurse] [-Depth <uint32>] [-Force] [-Name] [-UseTransactions] [-AmbiguousPaths] [-Item <Item>] [-ID <string>] [-WithParent] [<CommonParameters>]
Get-ChildItem [[-Filter] <string>] -LiteralPath <string[]> [-Include <string[]>] [-Exclude <string[]>] [-Recurse] [-Depth <uint32>] [-Force] [-Name] [-UseTransactions] [-AmbiguousPaths] [-Item <Item>] [-ID <string>] [-WithParent] [<CommonParameters>]
```

ALIASES

```
gci
ls
dir
```

REMARKS

```
None
```

Name : Getting Started with Sitecore Helix
DisplayName : Getting Started with Sitecore Helix
HasChildren : False
Language : en
Version : 1
ID : {1D8BA329-E1A1-44E3-949B-45EF7D00289E}
Database : master
TemplateName : Default
ItemPath : /sitecore/content/Home/Getting Started with Sitecore Helix

You can abort a script running in ISE with the Ctrl+Shift+E hotkey.

2. Sitecore PowerShell Commands

- Get-ChildItem & Format-Table

```
Format-Table [[-Property] <Object[]>] [-AutoSize] [-HideTableHeaders] [-Wrap] [-GroupBy <Object>] [-View <string>] [-ShowError] [-DisplayError] [-Force] [-Expand {CoreOnly | EnumOnly | Both}] [-InputObject <psobject>] [<CommonParameters>]
```

```
1 Get-Item -Path master: -Query "/sitecore/Media Library"
2
3 Get-ChildItem -path "/sitecore/Media Library" -recurve ` 
4 | format-Table DisplayName, Name, ID, TemplateName
5
```

Name	Children	Language	Version	ID	TemplateName
media library	True	en	1	{3D6658D8-A0BF-4E75-B3E2-D050FABC4E1}	Main section
DisplayName	Name		ID		TemplateName
Default Website	Default Website		{4D8A1C4D-1351-4DD1-A502-0660F01FEDC7}		Media folder
cover	cover		{04DAD0FD-DB66-4070-881F-17264CA257E1}		Jpeg
sc_logo	sc_logo		{094AE03-02E7-4868-80CB-19926661FB77}		Image
Files	Files		{730C1699-87A4-4CA7-A9CE-294AD7151F13}		Media folder
Images	Images		{15451229-7534-44EF-815D-D93D6170BFCB}		Media folder
Hero_Logo	Hero_Logo		{B46CD53F-28BD-4BC5-BE8A-95AE1CF62E9E}		Image
shutterstock_7507262231	shutterstock_7507262231		{4D22FD50-93D5-4484-A24E-20F0C11F61B4}		Jpeg
System	System		{7F86429E-6000-4A55-B046-ACD9D8110211}		Media folder
Forms	Forms		{9D479E0E-6000-4A55-B046-ACD9D8110211}		Media folder
Form	Form		{6590E671-6000-4A55-B046-ACD9D8110211}		Media folder
Simulator Backgrounds	Simulator Backgrounds		{76DB8BDE-6000-4A55-B046-ACD9D8110211}		Media folder
Android Phablet	Android Phablet		{439294C7-6000-4A55-B046-ACD9D8110211}		Media folder
Android Phablet Landscape	Android Phablet Landscape		{3C4269A1-6000-4A55-B046-ACD9D8110211}		Media folder
Android Phone	Android Phone		{98EF248E-6000-4A55-B046-ACD9D8110211}		Media folder
Android Phone_landscape	Android Phone_landscape		{E5391198-6000-4A55-B046-ACD9D8110211}		Media folder
Android Tablet	Android Tablet		{B8415FE4-6000-4A55-B046-ACD9D8110211}		Media folder
Android Tablet_landscape	Android Tablet_landscape		{A0C1DCFD-3D90-478F-864F-D448D9467ABB}		Jpeg
Blackberry	Blackberry		{65AE88A6-BE9B-4022-9BFF-49E66E3DC6D3}		Jpeg
Feature Phone	Feature Phone		{F6RA438D-B112-48FF-AE3C-4F0C046868D6}		Jpeg

You can abort a script running in ISE with the Ctrl+Shift+E hotkey.

```
Get-Item -Path master: -Query "/sitecore/Media Library"
Get-ChildItem -path "/sitecore/Media Library" -recurve ` 
| format-Table DisplayName, Name, ID, TemplateName
```

2. Sitecore PowerShell Commands

Find-Item : <https://doc.sitecorepowershell.com/appendix/indexing/find-item>

```
Find-Item [-Index] <String> [-Criteria <SearchCriteria[]>] [-QueryType <Type>] [-OrderBy <String>] [-First <Int32>] [-Last <Int32>] [-Skip <Int32>] [<CommonParameters>]

Find-Item [-Index] <String> [-Where <String>] [-WhereValues <Object[]>] [-Filter <String>] [-FilterValues <Object[]>] [-FacetOn <String[]>] [-FacetMinCount <Int32>] [-QueryType <Type>] [-OrderBy <String>] [-First <Int32>] [-Last <Int32>] [-Skip <Int32>] [<CommonParameters>]

Find-Item [-Index] <String> [-WherePredicate <Expression<Func<SearchResultItem, bool>>>] [-FilterPredicate <Expression<Func<SearchResultItem, bool>>>] [-QueryType <Type>] [-OrderBy <String>] [-First <Int32>] [-Last <Int32>] [-Skip <Int32>] [<CommonParameters>]

Find-Item [-Index] <String> [-ScopeQuery <String>] [-QueryType <Type>] [-OrderBy <String>] [-First <Int32>] [-Last <Int32>] [-Skip <Int32>] [<CommonParameters>]
```

```
1  @{ Filter = "Equals"; Field = "_templatename"; Value = "Sample Item"; },  
2  @{ Filter = "DescendantOf"; Value = (Get-Item "master:/content/") }
```

Filter = “one of below values”

Equals, StartsWith, Contains, ContainsAny, ContainsAll, EndsWith, LessThan, GreaterThan
, DescendantOf - performs a Contains with the field _path
, Fuzzy
, InclusiveRange - performs a Between using int, double, datetime, and stringtypes
, ExclusiveRange - same as InclusiveRange
, MatchesRegex - use something like ^.*\$
, MatchesWildcard - use something like H?li*m

2. Sitecore PowerShell Commands

Find-Item : <https://doc.sitecorepowershell.com/appendix/indexing/find-item>

```
1 $criteria = @(
2     @{Filter = "startsWith"; Field = "_fullpath"; Value = "/sitecore/content/" }
3 )
4 $props = @{
5     Index = "sitecore_master_index"
6     Criteria = $criteria
7 }
8
9 Find-Item @props -First 5 | Select-Object #-Expand "Fields"
10 |
```

Name	Lang.	Version	Database	Item Id	Template Name
Global Datasources	en	1	master	{088308AC-184C-40EF-A675-D64EC435357A}	Folder
Home	en	1	master	{110D559F-DEA5-42EA-9C1C-8A5DF7E70EF9}	Sample Item
Getting Started with Sitecore...	en	1	master	{1D8BA329-E1A1-44E3-949B-45EF7D00289E}	Default
Hero Image	en	1	master	{4B3DD090-5D93-43F0-9BFE-31E2D8EED515}	Hero Image

Use “SearchResultItem” to find Fields :

- **_smallcreateddate** - CreatedDate
- **_smallupdateddate** - Updated
- **_group** - ID
- **_template** - TemplateId
- **_templatename** - TemplateName
- **_fullpath** - Path

```
$props = @{
    Index = "sitecore_master_index"
    Where = 'Paths.Contains(@0)'
    WhereValues = [ID]::Parse("{0DE95AE4-41AB-4D01-9EB0-67441B7C2450}")
    Filter = 'Language = @0'
    FilterValues = "en-CA"
}

Find-Item @props
```

2. Sitecore PowerShell Commands

Find-Item : <https://doc.sitecorepowershell.com/appendix/indexing/find-item>

```
1 #Get-Help Format-Table
2 $NamePart = "*layout*"
3 Find-Item -Index sitecore_master_index ` 
4 | Where-Object { $_.Path -Like $NamePart } ` 
5 | Format-Table ItemId, Language, TemplateName, Path
```

ItemId	Language	TemplateName	Path
{DA279311-B061-47C8-B26E-75F1BFA85EDF}	en	Layout	/sitecore/layout/layouts/project/website/default
{5861BEF6-7848-4F6A-BB65-8FDFD098159E}	en	Layout Folder	/sitecore/layout/layouts/project/website
{11352249-34E8-43D9-AA55-8F5C3261AC2A}	en	View rendering	/sitecore/layout/renderings/feature/pagecontent/body copy
{EE8850A7-4C93-4F6B-9258-678BE7863BE2}	en	Rendering Folder	/sitecore/layout/renderings/feature/pagecontent
{9C44C385-8785-43A1-B02A-514C596A7284}	en	View rendering	/sitecore/layout/renderings/feature/pagecontent/hero image
{7CC5F576-FA33-4773-B94E-22DC9CEEB8D}	en	PowerShell Script	/sitecore/system/modules/powershell/script library/spe/reporting/content reports/reports/solution
{3B93315F-C240-4682-A45F-CDB09D0F2F65}	da	PowerShell Rule	/sitecore/system/modules/powershell/script library/spe/extensions/authorable reports/reports/rule
{3B93315F-C240-4682-A45F-CDB09D0F2F65}	en	PowerShell Rule	/sitecore/system/modules/powershell/script library/spe/extensions/authorable reports/reports/rule
{8E5FDEAE-4C9D-4A8B-A58B-A502C0CF9128}	da	PowerShell Script Library	/sitecore/system/modules/powershell/script library/spe/tools/copy renderings/content editor/conten
{8E5FDEAE-4C9D-4A8B-A58B-A502C0CF9128}	en	PowerShell Script Library	/sitecore/system/modules/powershell/script library/spe/tools/copy renderings/content editor/conten
{CC45B728-3141-4B2D-BC6E-CA74EE04AE02}	da	PowerShell Script	/sitecore/system/modules/powershell/script library/spe/tools/copy renderings/content editor/conten
{CC45B728-3141-4B2D-BC6E-CA74EE04AE02}	en	PowerShell Script	/sitecore/system/modules/powershell/script library/spe/tools/copy renderings/content editor/conten
{AFA7A862-9433-4EFE-ACA4-487A35CA0B6C}	da	PowerShell Script	/sitecore/system/modules/powershell/script library/spe/tools/copy renderings/content editor/conten
{AFA7A862-9433-4EFE-ACA4-487A35CA0B6C}	en	PowerShell Script	/sitecore/system/modules/powershell/script library/spe/tools/copy renderings/content editor/conten

```
ItemId      : {8FE10CB5-0838-4F10-9BAC-5E9D3685D4BD}
Language    : en
Version     : 1
DatabaseName : master
Path        : /sitecore/templates/feature/pagecontent/body copy/body copy content
TemplateName : Template section
```

2. Sitecore PowerShell Commands

Find-Item : <https://doc.sitecorepowershell.com/appendix/indexing/find-item>

```

1 $DefaultPath = "/sitecore/content/"
2 $NamePart = "hero"
3 Find-Item -Index sitecore_master_index `
4   -Where 'Name.Contains(@0) and Path.StartsWith(@1)' `
5   -WhereValues $NamePart, $DefaultPath `
6   | format-Table Name, ItemId, Language, Templatename, Path |
7

```

```

$DefaultPath = "/sitecore/content/"
$NamePart = "hero"
Find-Item -Index sitecore_master_index `
-Where 'Name.Contains(@0) and Path.StartsWith(@1)' `
-WhereValues $NamePart, $DefaultPath `
| format-Table Name, ItemId, Language, Templatename, Path

```

Name	ItemId	Language	TemplateName	Path
Hero Image {4B3DD090-5D93-43F0-9BFE-31E2D8EED515}	en	Hero Image	/sitecore/content/global datasources/hero image	


```

1 $NamePart = "sitecore"
2 $DefaultPath = "/sitecore/"
3 Find-Item -Index sitecore_master_index `
4   -Where 'Path.Contains(@0) and Path.StartsWith(@1)' `
5   -WhereValues $NamePart, $DefaultPath `
6   -Orderby Path `
7   | format-Table Name, ItemId, Language, Templatename, Path |

```

Name	ItemId	Language	TemplateName	Path
Global Datasources	{088308AC-184C-40EF-A675-D64EC435357A}	en	Folder	/sitecore/content/global datasources
Hero Image	{4B3DD090-5D93-43F0-9BFE-31E2D8EED515}	en	Hero Image	/sitecore/content/global datasources
Home	{110D559F-DEA5-42EA-9C1C-8A5DF7E70EF9}	en	Sample Item	/sitecore/content/home
Getting Started with Sitecore Helix	{1D88A329-E1A1-44E3-949B-45EF7D00289E}	en	Default	/sitecore/content/home/getting-started-with-sitecore-helix
Website	{5861BEF6-7848-4F6A-BB65-BFD0D98159E}	en	Layout Folder	/sitecore/layout/layouts/project/web
Default	{DA279311-B061-47C8-B26E-75F18FA85EDF}	en	Layout	/sitecore/layout/layouts/project/web
PageContent	{EE8850A7-4C93-4F6B-9258-67B8E78638E2}	en	Randomizing Folder	/sitecore/layout/layouts/project/web
Body Copy	{11352249-34E8-43D9-AAA5-8F5C3261AC2A}	en		
Hero Image	{9C44C3B5-87B5-43A1-B02A-514C596A72B4}	en		
Powershell	{DADC7C07-2606-4F1C-98C6-FB8162E55257}	da		
Powershell	{DADC7C07-2606-4F1C-98C6-FB8162E55257}	en		
Console Colors	{42F0A0E6-F121-432A-B21D-D40C53560563}	da		
Console Colors	{42F0A0E6-F121-432A-B21D-D40C53560563}	ja-JP		
Black	{81184848-ECF9-4448-8515-DFD8C83AC41B}	da		
Black	{81184848-ECF9-4448-8515-DFD8C83AC41B}	en		
Black	{81184848-ECF9-4448-8515-DFD8C83AC41B}	ja-JP		
Blue	{B0085463-EE85-4888-8473-078943D7EF08}	en		
Blue	{B0085463-EE85-4888-8473-078943D7EF08}	ja-JP		
Blue	{B0085463-EE85-4888-8473-078943D7EF08}	da		
Cyan	{E2E5425E-3524-472F-9F50-1EE7B383F315}	da		
Cyan	{E2E5425E-3524-472F-9F50-1EE7B383F315}	en		
Cyan	{E2E5425E-3524-472F-9F50-1EE7B383F315}	ja-JP		

2. Sitecore PowerShell Commands

- Get-Help Get-Command

NAME

Get-Command

SYNTAX

```
Get-Command [[-ArgumentList] <Object[]>] [-Verb <string[]>] [-Noun <string[]>] [-Module <string[]>] [-FullyQualifiedModule <ModuleSpecification[]>] [-TotalCount <int>] [-Syntax] [-ShowCommandInfo] [-All] [-ListImported] [-ParameterName <string[]>] [-ParameterType <PSTypeName[]>] [<CommonParameters>]
```

```
Get-Command [[-Name] <string[]>] [[-ArgumentList] <Object[]>] [-Module <string[]>] [-FullyQualifiedModule <ModuleSpecification[]>] [- CommandType {Alias | Function | Filter | Cmdlet | ExternalScript | Application | Script | Workflow | Configuration | All}] [-TotalCount <int>] [-Syntax] [-ShowCommandInfo] [-All] [-ListImported] [-ParameterName <string[]>] [-ParameterType <PSTypeName[]>] [<CommonParameters>]
```

- Get-Command Get-Item

CommandType	Name	Version	Source
Cmdlet	Get-Item	3.0.0.0	Microsoft.PowerShell.Management

2. Sitecore PowerShell Commands

- Get-Command - CommandType Cmdlet

CommandType	Name	Version	Source
Cmdlet	Add-AppvClientConnectionGroup	1.0.0.0	AppvClient
Cmdlet	Add-AppvClientPackage	1.0.0.0	AppvClient
Cmdlet	Add-AppvPublishingServer	1.0.0.0	AppvClient
Cmdlet	Add-AppxPackage	2.0.1.0	Appx
Cmdlet	Add-AppxProvisionedPackage	3.0	Dism
Cmdlet	Add-AppxVolume	2.0.1.0	Appx
Cmdlet	Add-BaseTemplate		
Cmdlet	Add-BitsFile	2.0.0.0	BitsTransfer
Cmdlet	Add-CertificateEnrollmentPolicyServer	1.0.0.0	PKI
Cmdlet	Add-Computer	3.0.0.0	Microsoft.PowerShell.Management
Cmdlet	Add-Content	3.0.0.0	Microsoft.PowerShell.Management
Cmdlet	Add-History	3.0.0.0	Microsoft.PowerShell.Core
Cmdlet	Add-ItemAcl		
Cmdlet	Add-ItemVersion		
Cmdlet	Add-JobTrigger	1.1.0.0	PSScheduledJob
Cmdlet	Add-KdsRootKey	1.0.0.0	Kds
Cmdlet	Add-LocalGroupMember	1.0.0.0	Microsoft.PowerShell.LocalAccounts
Cmdlet	Add-Member	3.0.0.0	Microsoft.PowerShell.Utility
Cmdlet	Add-PlaceholderSetting		
Cmdlet	Add-PSSnapin	3.0.0.0	Microsoft.PowerShell.Core
Cmdlet	Add-Rendering		
Cmdlet	Add-RoleMember		
Cmdlet	Add-RoleMember	21.1.18245	SqlServer
Cmdlet	Add-SignerRule	1.0	ConfigCI
Cmdlet	Add-SqlAvailabilityDatabase	21.1.18245	SqlServer

2. Sitecore PowerShell Commands

- **Get-Command - CommandType Cmdlet - Module Microsoft.PowerShell.Management**

Cmdlet	Add-Computer	Get-ItemPropertyValue	Remove-EventLog	Start-Process
Cmdlet	Add-Content	Get-Location	Remove-Item	Start-Service
Cmdlet	Checkpoint-Computer	Get-Process	Remove-ItemProperty	Start-Transaction
Cmdlet	Clear-Content	Get-PSDrive	Remove-PSDrive	Stop-Computer
Cmdlet	Clear-EventLog	Get-PSProvider	Remove-WmiObject	Stop-Process
Cmdlet	Clear-Item	Get-Service	Rename-Computer	Stop-Service
Cmdlet	Clear-ItemProperty	Get-TimeZone	Rename-Item	Suspend-Service
Cmdlet	Clear-RecycleBin	Get-Transaction	Rename-ItemProperty	Test-ComputerSecureChannel
Cmdlet	Complete-Transaction	Get-WmiObject	Reset-ComputerMachinePassword	Test-Connection
Cmdlet	Convert-Path	Invoke-Item	Resolve-Path	Test-Path
Cmdlet	Copy-Item	Invoke-WmiMethod	Restart-Computer	Undo-Transaction
Cmdlet	Copy-ItemProperty	Join-Path	Restart-Service	Use-Transaction
Cmdlet	Debug-Process	Limit-EventLog	Restore-Computer	Wait-Process
Cmdlet	Disable-ComputerRestore	Move-Item	Resume-Service	Write-EventLog
Cmdlet	Enable-ComputerRestore	Move-ItemProperty	Set-Clipboard	
Cmdlet	Get-ChildItem	New-EventLog	Set-Content	
Cmdlet	Get-Clipboard	New-Item	Set-Item	
Cmdlet	Get-ComputerInfo	New-ItemProperty	Set-ItemProperty	
Cmdlet	Get-ComputerRestorePoint	New-PSDrive	Set-Location	
Cmdlet	Get-Content	New-Service	Set-Service	
Cmdlet	Get-ControlPanelItem	New-WebServiceProxy	Set-TimeZone	
Cmdlet	Get-EventLog	Pop-Location	Set-WmiInstance	
Cmdlet	Get-HotFix	Push-Location	Show-ControlPanelItem	
Cmdlet	Get-Item	Register-WmiEvent	Show-EventLog	
Cmdlet	Get-ItemProperty	Remove-Computer	Split-Path	

2. Sitecore PowerShell Commands

- `Get-Command - CommandType Cmdlet - Module Microsoft.PowerShell.Utility`

Cmdlet	Add-Member	Format-Table	Import-PSSession	Send-MailMessage
Cmdlet	Add-Type	Format-Wide	Invoke-Expression	Set-Alias
Cmdlet	Clear-Variable	Get-Alias	Invoke-RestMethod	Set-Date
Cmdlet	Compare-Object	Get-Culture	Invoke-WebRequest	Set-PSBreakpoint
Cmdlet	ConvertFrom-Csv	Get-Date	Measure-Command	Set-TraceSource
Cmdlet	ConvertFrom-Json	Get-Event	Measure-Object	Set-Variable
Cmdlet	ConvertFrom-String	Get-EventSubscriber	New-Alias	Show-Command
Cmdlet	ConvertFrom-StringData	Get-FormatData	New-Event	Sort-Object
Cmdlet	Convert-String	Get-Host	New-Object	Start-Sleep
Cmdlet	ConvertTo-Csv	Get-Member	New-TimeSpan	Tee-Object
Cmdlet	ConvertTo-Html	Get-PSBreakpoint	New-Variable	Trace-Command
Cmdlet	ConvertTo-Json	Get-PSCallStack	Out-File	Unblock-File
Cmdlet	ConvertTo-Xml	Get-Random	Out-GridView	Unregister-Event
Cmdlet	Debug-Runspace	Get-Runspace	Out-Printer	Update-FormatData
Cmdlet	Disable-PSBreakpoint	Get-RunspaceDebug	Out-String	Update-List
Cmdlet	Disable-RunspaceDebug	Get-TraceSource	Read-Host	Update-TypeData
Cmdlet	Enable-PSBreakpoint	Get-TypeData	Register-EngineEvent	Wait-Debugger
Cmdlet	Enable-RunspaceDebug	Get-UICulture	Register-ObjectEvent	Wait-Event
Cmdlet	Export-Alias	Get-Unique	Remove-Event	Write-Debug
Cmdlet	Export-Clixml	Get-Variable	Remove-PSBreakpoint	Write-Error
Cmdlet	Export-Csv	Group-Object	Remove-TypeData	Write-Host
Cmdlet	Export-FormatData	Import-Alias	Remove-Variable	Write-Information
Cmdlet	Export-PSSession	Import-Clixml	Select-Object	Write-Output
Cmdlet	Format-Custom	Import-Csv	Select-String	Write-Progress
Cmdlet	Format-List	Import-LocalizedData	Select-Xml	Write-Verbose
...	Write-Warning

2. Sitecore PowerShell Commands

- `Get-Command - CommandType Cmdlet - Module PackageManagement`

CommandType	Name	Version	Source
-----	-----	-----	-----
Cmdlet	Find-Package	1.4.7	PackageManagement
Cmdlet	Find-PackageProvider	1.4.7	PackageManagement
Cmdlet	Get-Package	1.4.7	PackageManagement
Cmdlet	Get-PackageProvider	1.4.7	PackageManagement
Cmdlet	Get-PackageSource	1.4.7	PackageManagement
Cmdlet	Import-PackageProvider	1.4.7	PackageManagement
Cmdlet	Install-Package	1.4.7	PackageManagement
Cmdlet	Install-PackageProvider	1.4.7	PackageManagement
Cmdlet	Register-PackageSource	1.4.7	PackageManagement
Cmdlet	Save-Package	1.4.7	PackageManagement
Cmdlet	Set-PackageSource	1.4.7	PackageManagement
Cmdlet	Uninstall-Package	1.4.7	PackageManagement
Cmdlet	Unregister-PackageSource	1.4.7	PackageManagement

- `Get-Command - Verb Get`

3. Sitecore PowerShell Commands – Save Script & Use

Untitled (*) - PowerShell ISE

HOME SETTINGS PLUGINS

New Open Save Save As... 1. Reload Selection Execute Debug Abort

Context master:\media library Session ISE editing session Language Current Language [en] User sitecore\Admin [Current] Context UI Context About Info

```
1 $path = "master:\media library\"  
2 get-childitem -Path $path -recurse  
3 | format-table -property DisplayName, {$_.Paths.Path}  
4  
5  
6  
7
```

DisplayName \$_.Paths.Path

Default Website /sitecore/media library/Default Website
cover /sitecore/media library/Default Website/cover
sc_logo /sitecore/media library/Default Website/sc_logo
Files /sitecore/media library/Files
Images /sitecore/media library/Images
Hero_Logo /sitecore/media library/Images/Hero_Logo
shutterstock_7507262231 /sitecore/media library/Images/shutterstock_7507262231
System /sitecore/media library/System
Forms /sitecore/media library/System/Forms
Form /sitecore/media library/System/Form
Simulator Backgrounds /sitecore/media library/System/Simulator Backgrounds
Android Phablet /sitecore/media library/System/Simulator Backgrounds
Android Phablet Landscape /sitecore/media library/System/Simulator Backgrounds
Android Phone /sitecore/media library/System/Simulator Backgrounds
Android Phone_landscape /sitecore/media library/System/Simulator Backgrounds
Android Tablet /sitecore/media library/System/Simulator Backgrounds
Android Tablet_landscape /sitecore/media library/System/Simulator Backgrounds
Blackberry /sitecore/media library/System/Simulator Backgrounds
Feature Phone /sitecore/media library/System/Simulator Backgrounds
HD TV /sitecore/media library/System/Simulator Backgrounds
iPad /sitecore/media library/System/Simulator Backgrounds
iPad_landscape /sitecore/media library/System/Simulator Backgrounds
iPhone /sitecore/media library/System/Simulator Backgrounds
iPhone_landscape /sitecore/media library/System/Simulator Backgrounds
Windows Phone /sitecore/media library/System/Simulator Backgrounds
Windows Phone Landscape /sitecore/media library/System/Simulator Backgrounds
Template_Thumbnails /sitecore/media library/System/Template Thumbnails

You can abort a script running in ISE with the Ctrl+Shift+E hotkey.

Select Script Library

Select the Library that you want to save your script to.

master

SPE

Core

Extensions

Maintenance

Reporting

Samples

Automatically show quick info section

Enforce user password expiration

License Expiration

Random desktop background

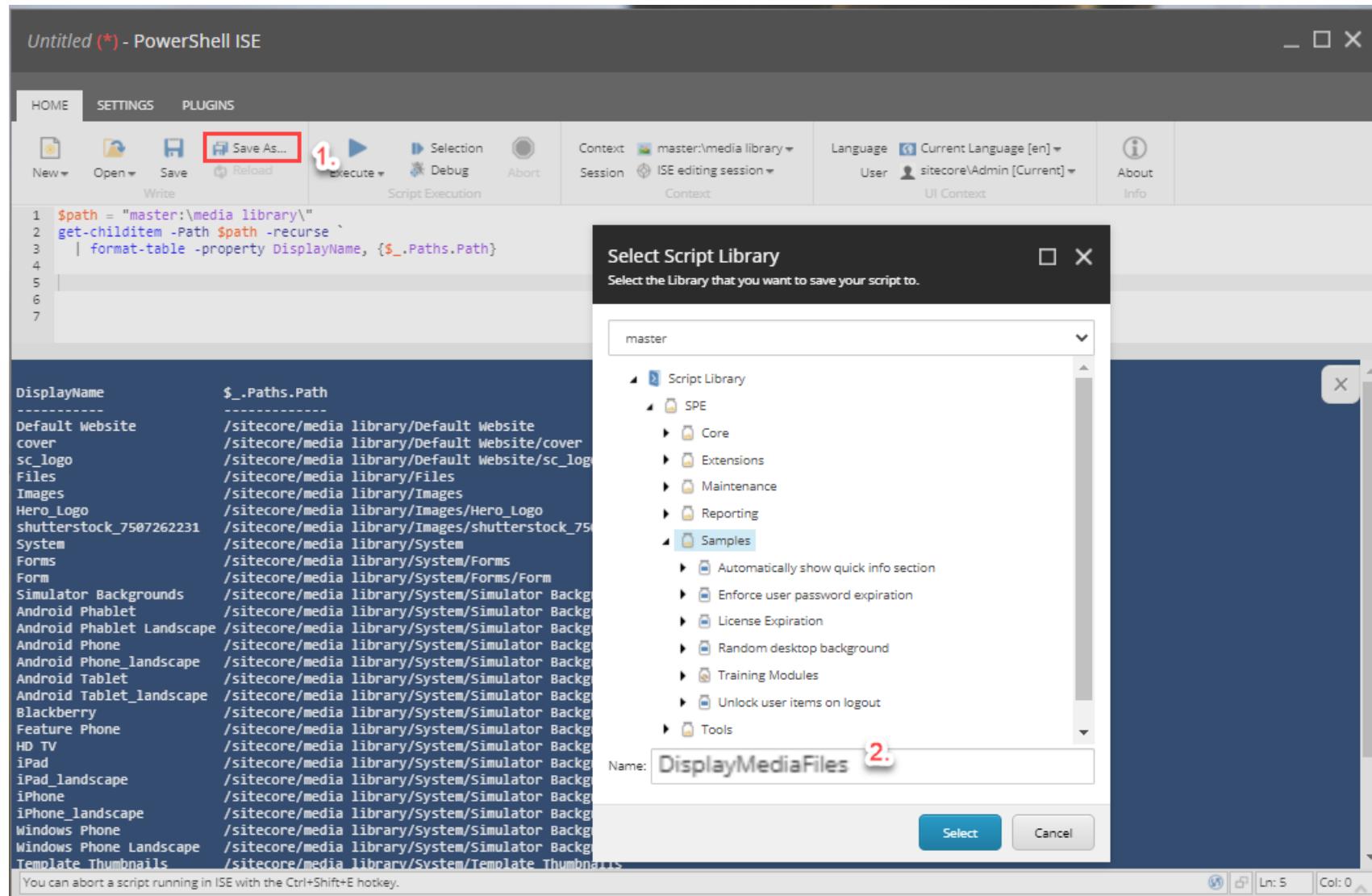
Training Modules

Unlock user items on logout

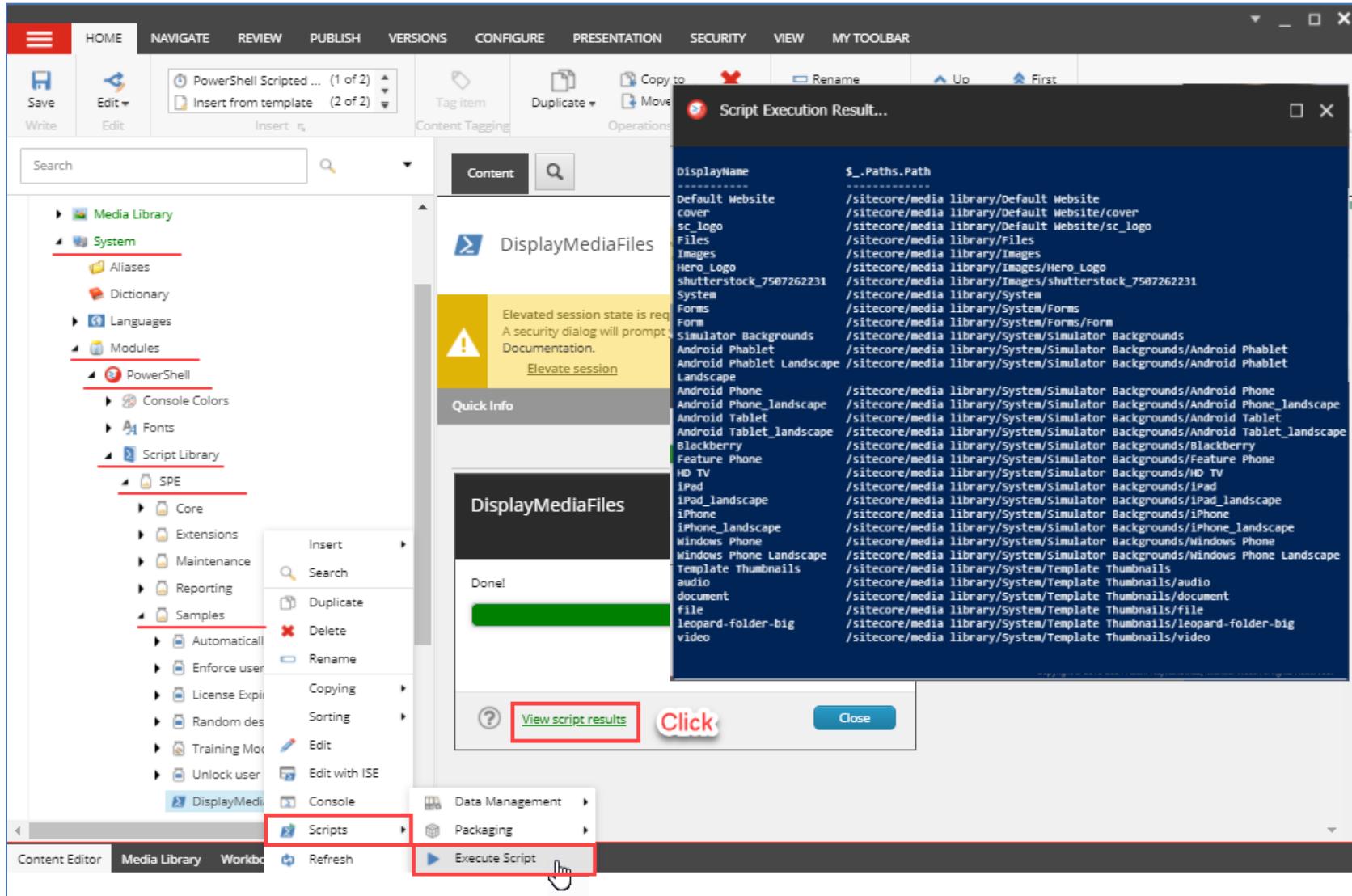
Tools

Name: 2. DisplayMediaFiles

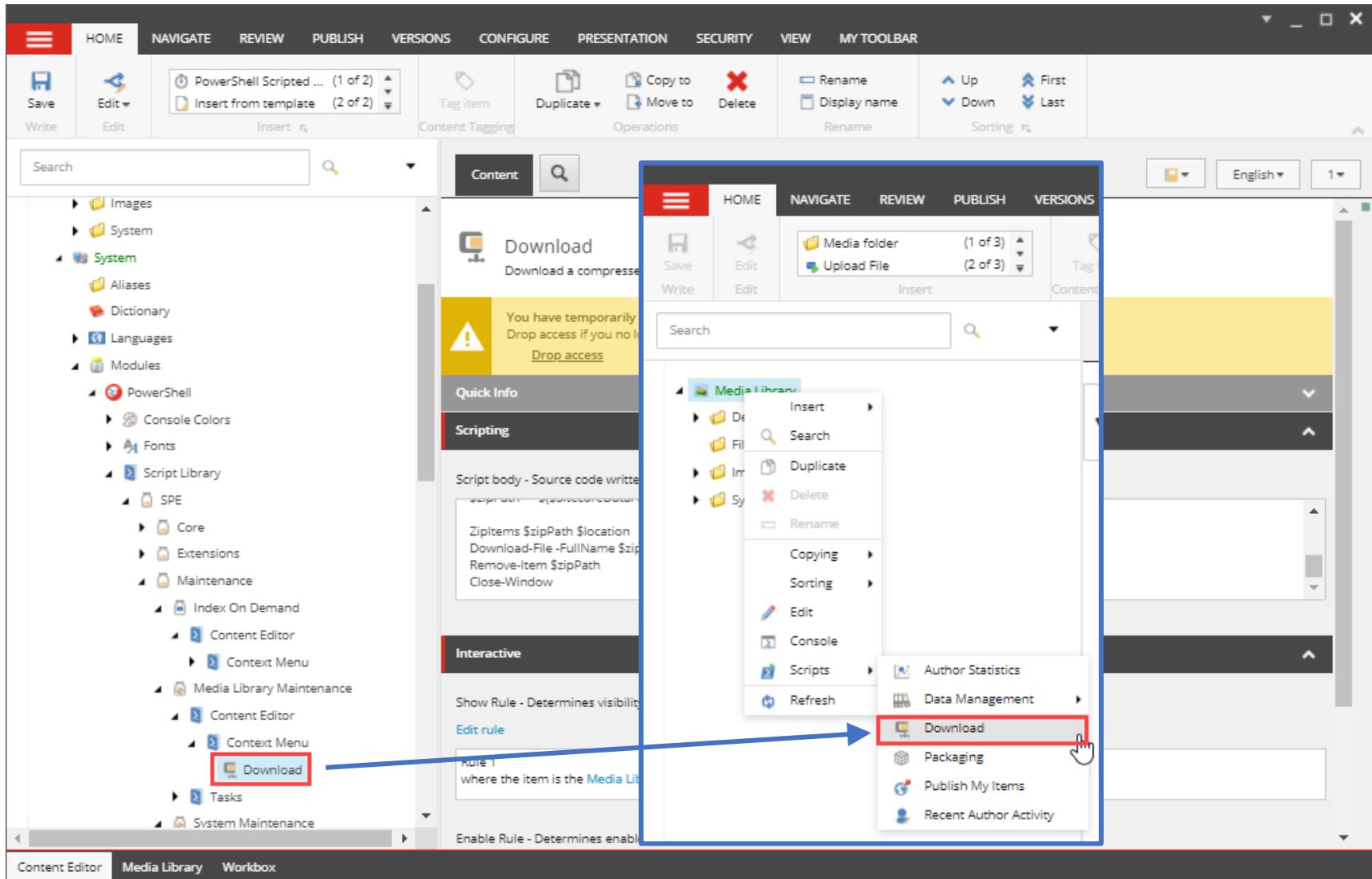
Select Cancel



3. Sitecore PowerShell Commands – Save Script & Use



3. Sitecore PowerShell Commands – Built-In Script “Media Library Download” & Use



3. Sitecore PowerShell Commands – Built-In Script “Media Library Download” & Use

```
function ZipItems( $zipArchive, $sourcedir )
{
    Set-Location $sourcedir
    [System.Reflection.Assembly]::Load("WindowsBase,Version=3.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35") > $null
    $ZipPackage=[System.IO.Packaging.ZipPackage]::Open($zipArchive, [System.IO.FileMode]::OpenOrCreate,[System.IO.FileAccess]::ReadWrite)
    $items = @(Get-Item $sourceDir) + (Get-ChildItem -recurse $sourceDir)
    [byte[]]$buff = new-object byte[] 40960
    $i = 0;
    ForEach ($item In $items) {
        $i++
        if(([Sitecore.Resources.Media.MediaManager]::HasMediaContent($item))){
            $mediaItem = New-Object "Sitecore.Data.Items.MediaItem" $item;
            $mediaStream = $mediaItem.GetMediaStream();
            $fileName = Resolve-Path -Path $item.ProviderPath -Relative
            $fileName = "$fileName.$($item.Extension)".Replace("\","/").Replace("./","/");
            "Added: $fileName"
            Write-Progress -Activity "Zipping Files " -CurrentOperation "Adding $fileName" -Status "$i out of $($items.Length)" -PercentComplete ($i *100 / $items.Length)
            $partUri = New-Object System.Uri($fileName, [System.UriKind]::Relative)
            $partUri = [System.IO.Packaging.PackUriHelper]::CreatePartUri($partUri);
            $part=$ZipPackage.CreatePart($partUri, "application/zip", [System.IO.Packaging.CompressionOption]::Maximum)
            $stream=$part.GetStream();
            do {
                $count = $mediaStream.Read($buff, 0, $buff.Length)
                $stream.Write($buff, 0, $count)
            } while ($count -gt 0)
            $stream.Close()
            $mediaStream.Close()
        }
    }
    $ZipPackage.Close()
}
$location = get-location
$time = Get-Date -format "yyyy-MM-d_hhmss"
$zipName = Split-Path -leaf $location | % { $_ -replace " ", "" }
$zipPath = "$($SitecoreDataFolder)\$zipName-$time.zip"

ZipItems $zipPath $location
Download-File -FullName $zipPath > $null
Remove-Item $zipPath
Close-Window
```

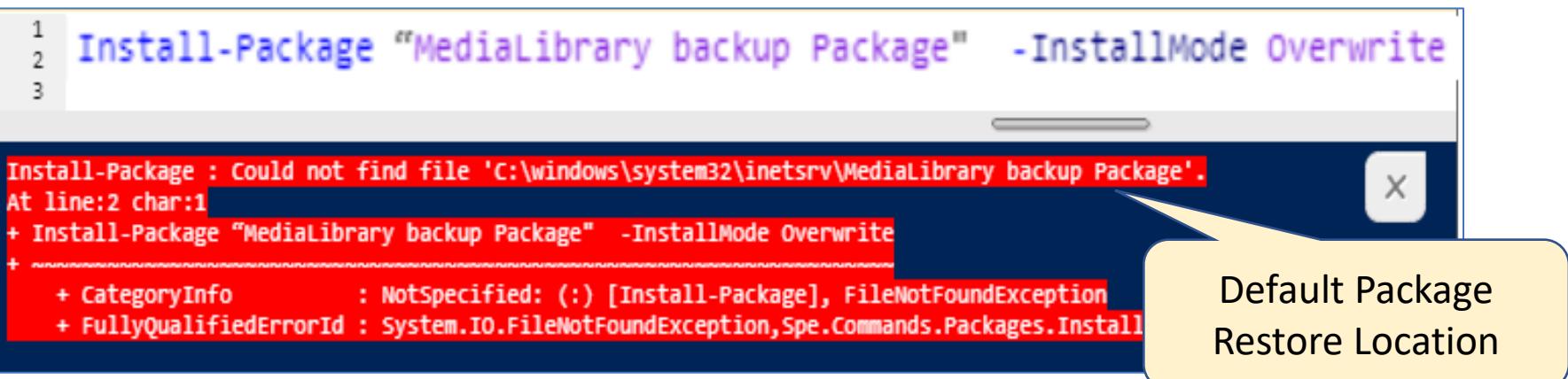
4. Sitecore PowerShell Commands – Backup and Restore package Script

- Backup as Package

```
$SaveFolder = "C:\Temp\"  
$package = new-package "MediaLibrary backup Package";  
$package.Sources.Clear();  
$package.Metadata.Version = "2.0";  
$source = Get-Item 'master:/Media Library/' | New-ItemSource -Name 'MediaLibrary' -InstallMode Overwrite  
$package.Sources.Add($source);  
Export-Package -Project $package -Path "$($SaveFolder)$($package.Name)- $($package.Metadata.Version).zip" -Zip
```

- Restore Package

```
Install-Package -Path "C:\Temp\MediaLibrary backup Package-2.0.zip" -InstallMode Overwrite
```



The screenshot shows a PowerShell window with the following content:

```
1 Install-Package "MediaLibrary backup Package" -InstallMode Overwrite  
2  
3  
Install-Package : Could not find file 'C:\windows\system32\inetsrv\MediaLibrary backup Package'.  
At line:2 char:1  
+ Install-Package "MediaLibrary backup Package" -InstallMode Overwrite  
+ CategoryInfo          : NotSpecified: () [Install-Package], FileNotFoundException  
+ FullyQualifiedErrorId : System.IO.FileNotFoundException, Sitecore.Commands.Packages.Install
```

A callout bubble points to the error message with the text "Default Package Restore Location".

```
Install-Package [[-Path] <String>] [-InstallMode <Undefined | Overwrite | Merge | Skip | SideBySide>] [-MergeMode <Undefined | Clear | Append | Merge>] [-DisableIndexing <SwitchParameter>] [<CommonParameters>]
```

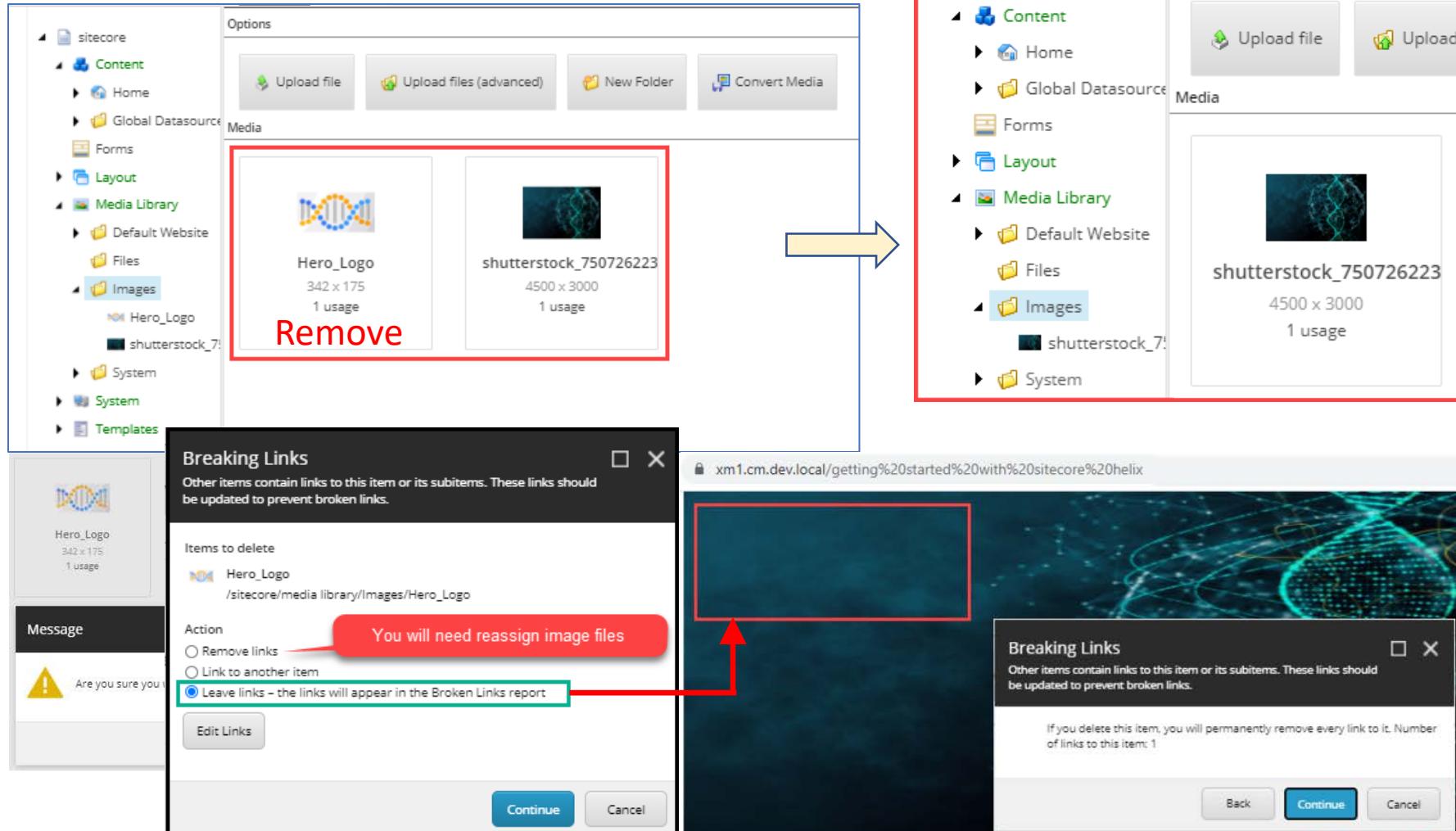
4. Sitecore PowerShell Commands – Backup and Restore package Script

- Backup Package Structure

TEMP > MediaLibrary backup Package-2.0.zip						
Name	Type	Compressed size	Password ...	Size	Ratio	Date modified
package.zip	Compressed (zipped) Fol...	7,890 KB	No	7,904 KB	1%	8/13/2021 4:52 PM
Name Type Local > Temp > Temp1_MediaLibrary backup Package-2.0.zip > package.zip > blob > master						
Name	Type	Compressed size	Password ...	Size	Ratio	Date modified
blob	File folder	10 KB	No	10 KB	2%	8/13/2021 4:52 PM
installer	File folder	1 KB	No	1 KB	1%	8/13/2021 4:52 PM
items	File folder	29 KB	No	29 KB	7%	8/13/2021 4:52 PM
metadata	File folder	26 KB	No	26 KB	2%	8/13/2021 4:52 PM
properties	File folder	78 KB	No	78 KB	3%	8/13/2021 4:52 PM
project	File	25 KB	No	25 KB	2%	8/13/2021 4:52 PM
version	File	58 KB	No	58 KB	47%	8/13/2021 4:52 PM
467fc79c-a076-4ac3-87bb-b66f1972ff5e	File	27 KB	No	27 KB	11%	8/13/2021 4:52 PM
> Temp1_MediaLibrary backup Package-2.0.zip > package.zip > items > master > sitecore > media library >						
Name	Type	Compressed size	Password ...	Size	Ratio	Date modified
{3D6658D8-A0BF-4E75-B3E2-D050FAB...	File folder	KB	No	KB	0%	8/13/2021 4:52 PM
Default Website	File folder	KB	No	KB	0%	8/13/2021 4:52 PM
Files	File folder	KB	No	KB	0%	8/13/2021 4:52 PM
Images	File folder	KB	No	KB	0%	8/13/2021 4:52 PM
System	File folder	KB	No	KB	0%	8/13/2021 4:52 PM
MediaLibrary backup Package-2.0.zip > package.zip > items > master > sitecore > media library > Images >						
Name	Type	Compressed size	Password ...	Size	Ratio	Date modified
{15451229-7534-44EF-815D-D93D6170BFCB}	File folder	File	No	File	No	8/13/2021 4:52 PM
Hero_Logo	File folder	File	No	File	No	8/13/2021 4:52 PM
shutterstock_7507262231	File folder	File	No	File	No	8/13/2021 4:52 PM
dfa43cff-2277-40c2-b162-124689c10f21	File	30 KB	No	55 KB	45%	8/13/2021 4:52 PM

4. Sitecore PowerShell Commands

- Remove item & Publish



4. Sitecore PowerShell Commands

- Remove item & Publish

The screenshot illustrates the use of Sitecore PowerShell Extensions for managing Sitecore items.

Publish Site Dialog: Shows the configuration for publishing the website. The "Smart publish - publish differences between source and target database" option is selected. The publishing language is set to "Select all".

Publish Site Confirmation: Displays the results of the publication: "The website has been published. Items processed: 3186." A detailed log shows: Job started: Publish to 'web', Items created: 0, Items deleted: 0, Items updated: 0.

Browser Error: A red box highlights the URL bar showing the site URL: `xm1.cm.dev.local/getting%20started%20with%20sitecore%20helix`. The page displays a "Server Error in '/' Application." message with the following details:

- Value cannot be null.**
- Parameter name: item**
- Description:** An unhandled exception occurred.
- Exception Details:** System.ArgumentNullException: Value cannot be null.
Parameter name: item

Source Error: A code snippet from a PowerShell script is shown:

```
Line 1: $path = "master:\media library\"  
Line 2: get-childitem -Path $path -recurse  
Line 3: | where-object { $_.ItemPath -Like "*Images"  
Line 4: | Sort-Object -Property ItemPath  
Line 5: | format-Table Name, Id, Language, Template
```

Stack Trace: Shows the source file path: `c:\inetpub\wwwroot\x`.

Breaking Links Dialog: This dialog appears when attempting to delete an item that has links. It lists the item to be deleted: `Hero_Logo` (`/sitecore/media library/Images/Hero_Logo`). It includes an "Action" section with three options:

- Remove links** (selected)
- Link to another item
- Leave links - the links will appear in the Broken Links report

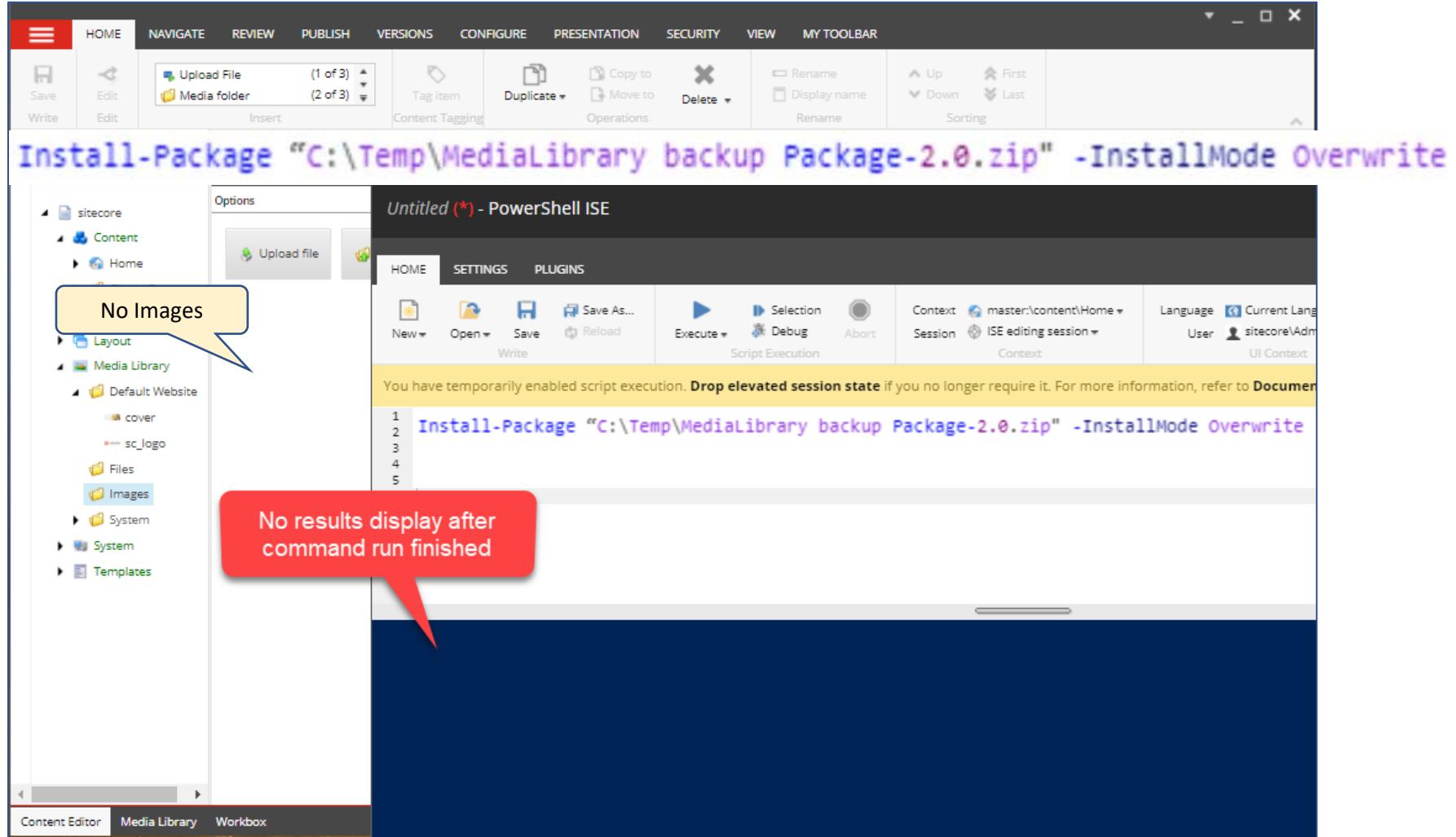
A "Edit Links" button is also present. The "Continue" and "Cancel" buttons are at the bottom right.

Table: A table showing the results of the PowerShell command execution. It lists items under the "Images" folder.

Name	ID	Language	TemplateName	ItemPath
Images	{15451229-7534-44EF-815D-D93D6170BFCB}	en	Media folder	/sitecore/media library/Images

4. Sitecore PowerShell Commands –Restore package Script

- Restore Package : Refresh folder after restore-package



4. Sitecore PowerShell Commands –Refresh folder after restore-package

Images is restored

Restore 'Backup Media Package' will only restore 'Media Library' and will not recover Removed links because it is out of backup scope

```
1 $path = "master:\media library\"  
2 get-childitem -Path $path -recurse `| where-object { $_.ItemPath -Like "*Images*" } `| Sort-Object -Property ItemPath `| format-Table Name, Id, Language, Templatename, ItemPath  
3 #Import-Package "C:\Temp\MediaLibrary backup Package.zip" -InstallMode Overwrite
```

Name	ID	Language	TemplateName	ItemPath
Images	{15451229-7534-44EF-815D-D93D6170BFCB}	en	Media folder	/sitecore/media library/Images
Hero Logo	{3F91B07C-490F-4799-A933-761C1490C573}	en	Image	/sitecore/media library/Images/Hero Logo
shutterstock_750726223	{19300968-D8B0-4C20-B141-A1AF9DF7C909}	en	Jpeg	/sitecore/media library/Images/shutterstock_750726223

5. Sitecore PowerShell Commands – Bulk Update

- List up item's property For “ZipCode” Field

```
$NamePart = "Media folder"
$DefaultPath = "master:/Media Library/"
$DefaultPath = "master:/Media Library/Images"
Get-ChildItem -Path $DefaultPath -Recurse |
    where-object { $_.TemplateName -notmatch "Media folder" } |
    Foreach-Object {
        "Name: " + $_.Name + ", Zip Code: " + $_.ZipCode + ", Path: " + $_.ItemPath;
        $_.Fields["ZipCode"];
    }
```

Name: shutterstock_7507262231, Zip Code: 777888999, Path: /sitecore/media library/Images/shutterstock_7507262231	
ContainsStandardValue : False	ResetBlank : False
ContainsFallbackValue : False	Section : Location
IsValidForLanguageFallback :	SectionSortorder : 400
InheritsValueFromOtherItem : False	Shared : True
FallbackValueSource :	ShouldBeTranslated : False
InnerItem : Sitecore.Data.Items.Item	Sortorder : 787
Database : master	Source :
Definition : Sitecore.Data.Templates.TemplateField	Style :
Description :	Title : Zip Code
DisplayName : Zip Code	ToolTip :
SectionDisplayName : Location	Translatable : False
HasBlobStream : False	Type : Single-Line Text
HasValue : True	TypeKey : single-line text
HelpLink :	Unversioned : False
ID : {FB5408F1-B2EB-47C9-A55F-F01B2FC5A551}	Validation :
InheritedValue : 777888999	ValidationText :
IsBlobField : False	Value : 777888999
IsModified : False	
Item : Sitecore.Data.Items.Item	
Key : zipcode	
Language : en	
SharedLanguageFallbackEnabled : False	
Name : ZipCode	

5. Sitecore PowerShell Commands – Bulk Update

- Update all selected item's "ZipCode" Field

```
$NamePart = "Media folder"
$DefaultPath = "master:/Media Library/Images"
Get-ChildItem -Path $DefaultPath -Recurse ` 
| where-object { $_.TemplateName -notmatch "Media folder" } ` 
| Foreach-Object {
    "Name: " + $_.Name + ", Zip Code: " + $_.ZipCode + ", Path: " + $_.ItemPath;
    $_.Editing.BeginEdit();
    $_.Fields["ZipCode"].Value = "92606";
    $_.Editing.EndEdit();
}
```

```
1 $NamePart = "Media folder"
2 $DefaultPath = "master:/Media Library/Images"
3 Get-ChildItem -Path $DefaultPath -Recurse ` 
4 | where-object { $_.TemplateName -notmatch "Media folder" } ` 
5 | Foreach-Object {
6     "Name: " + $_.Name + ", Zip Code: " + $_.ZipCode + ", Path: " + $_.ItemPath;
7     $_.Editing.BeginEdit();
8     $_.Fields["ZipCode"].Value = "92606";
9     $_.Editing.EndEdit();
10 }
```

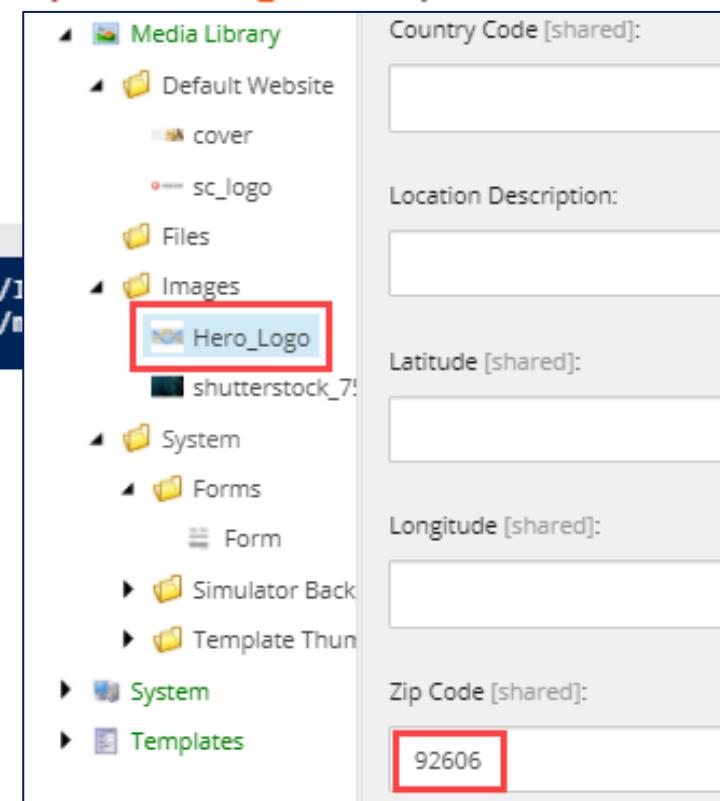
```
Name: Hero_Logo, Zip Code: 123456, Path: /sitecore/media library/Images/Hero_Logo
True
Name: shutterstock_7507262231, Zip Code: 777888999, Path: /sitecore/media library/Images/shutterstock_7507262231
True
```

5. Sitecore PowerShell Commands – Bulk Update

- Check Results : “ZipCode” Field

```
1 $NamePart = "Media folder"
2 $DefaultPath = "master:/Media Library/Images"
3 Get-ChildItem -Path $DefaultPath -Recurse
4 | where-object { $_.TemplateName -notmatch "Media folder" }
5 |
6 | Foreach-Object {
7     "Name: " + $_.Name + ", Zip Code: " + $_.ZipCode + ", Path: " + $_.ItemPath;
8     <#
9     $_.Editing.BeginEdit();
10    $_.Fields["ZipCode"].Value = "92606";
11    $_.Editing.EndEdit();
12    #>
13 }
```

```
Name: Hero_Logo, Zip Code: 92606, Path: /sitecore/media library/
Name: shutterstock_7507262231, Zip Code: 92606, Path: /sitecore/m
```



5. Sitecore PowerShell Commands – Bulk Update

- Check Results : “ZipCode” Field

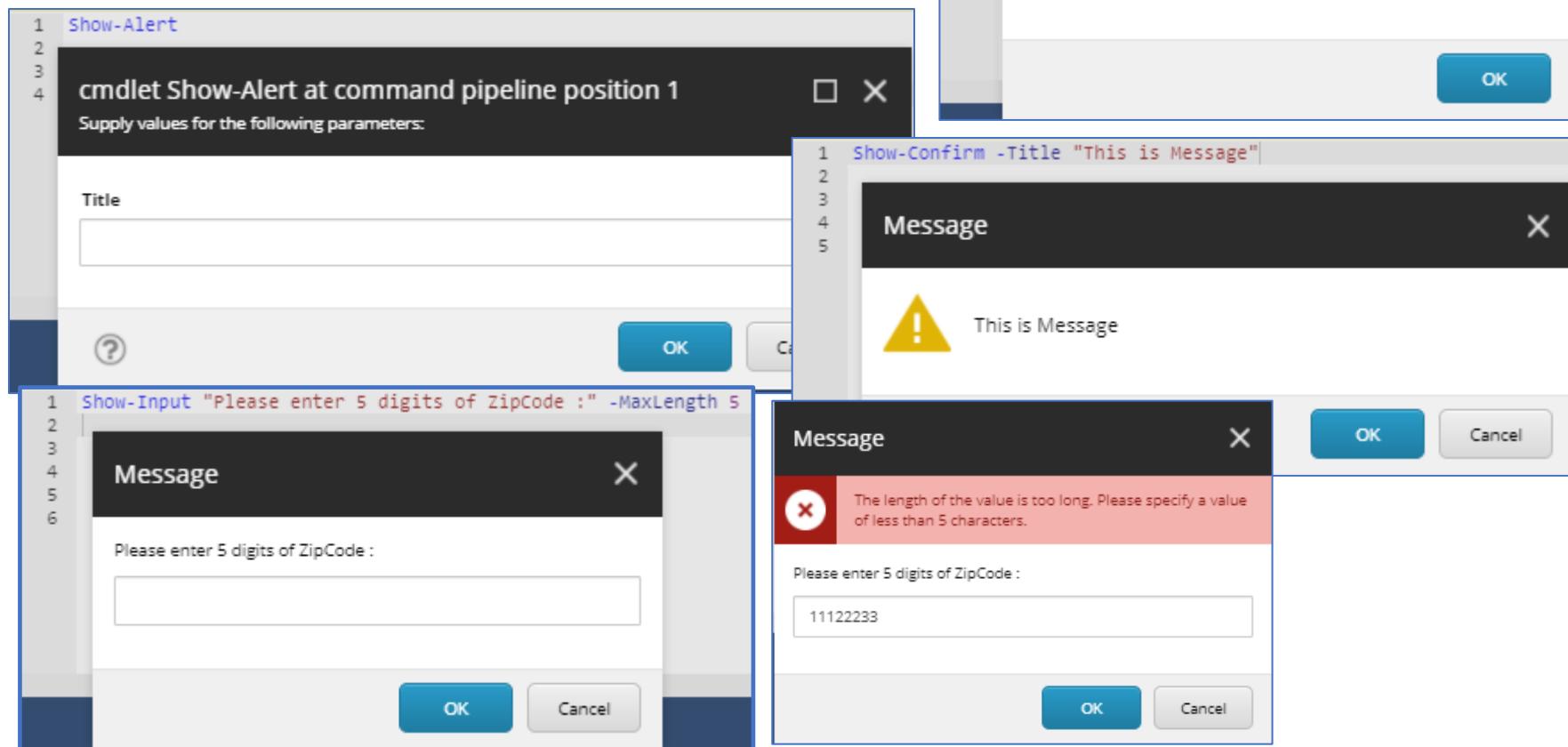
```
$NamePart = "Media folder"
$DefaultPath = "master:/Media Library/Images"
Get-ChildItem -Path $DefaultPath -Recurse ` 
| where-object { $_.TemplateName -notmatch "Media folder" } ` 
| Foreach-Object {
    "Name: " + $_.Name + ", Zip Code: " + $_.ZipCode + ", Path: " + $_.ItemPath;
    $_.ZipCode = "92868";
    "After Updated Zip Code: " + $_.ZipCode;
}
```

```
1 $NamePart = "Media folder"
2 $DefaultPath = "master:/Media Library/Images"
3 Get-ChildItem -Path $DefaultPath -Recurse ` 
4 | where-object { $_.TemplateName -notmatch "Media folder" } ` 
5 | Foreach-Object {
6     "Name: " + $_.Name + ", Zip Code: " + $_.ZipCode + ", Path: " + $_.ItemPath;
7     $_.ZipCode = "92868";
8     "After Updated Zip Code: " + $_.ZipCode;
9 }
10
11
```

```
Name: Hero_Logo, Zip Code: 92606, Path: /sitecore/media library/Images/Hero_Logo
After Updated Zip Code: 92868
Name: shutterstock_7507262231, Zip Code: 92606, Path: /sitecore/media library/Images/shutterstock_7507262231
After Updated Zip Code: 92868
```

6. Sitecore PowerShell Commands – UI of PowerShell

- Alert
- Confirmation
- User Input



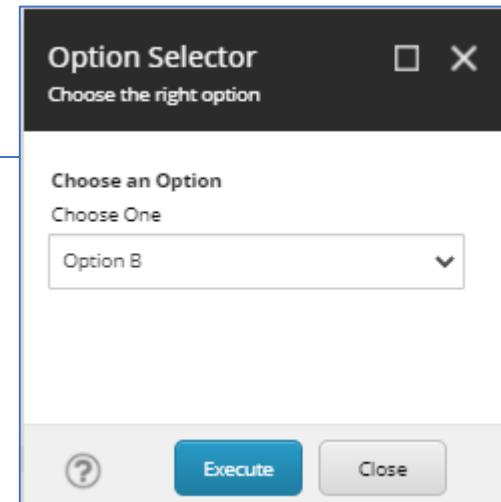
6. Sitecore PowerShell Commands – UI of PowerShell

```
$Options = @{
    "Option A" = "a"
    "Option B" = "b"
}

$props = @{
    Title = "Option Selector"
    Description = "Choose the right option"
    Width = 300
    Height = 300
    Showhint = $true
    OkButtonName = "Execute"
    CancelButtonName = "Close"
    parameters = @(
        @{
            Name = "InputOption";
            Title="Choose an Option";
            Options=$Options;
            Tooltip="Choose One"
        }
    )
}

$dialoResult = Read-Variable @props
if ($dialoResult -ne "ok") {
    Exit
}
Write-Host "Value entered by user: " $InputOption
```

<https://www.oshyn.com/blog/read-variable-sitecore-powershell>



The screenshot shows a modal dialog titled "Option Selector" with the sub-instruction "Choose the right option". The dialog has a dropdown menu labeled "Choose an Option" which contains the value "Option B". At the bottom of the dialog are two buttons: "Execute" (highlighted in blue) and "Close". Below the dialog, a status bar displays the message "Value entered by user: b".

```
$props = @{
    Title = "Option Selector"
    Description = "Choose the right option"
    Width = 300
    Height = 300
    Showhint = $true
    OkButtonName = "Execute"
    CancelButtonName = "Close"
    parameters = @(
        @{
            Name = "InputOption";
            Title="Choose an Option";
            Options=$Options;
            Tooltip="Choose One"
        }
    )
}

$dialoResult = Read-Variable @props
if ($dialoResult -ne "ok") {
    Exit
}
Write-Host "Value entered by user: " $InputOption
```

7. Sitecore PowerShell Commands – Remoting

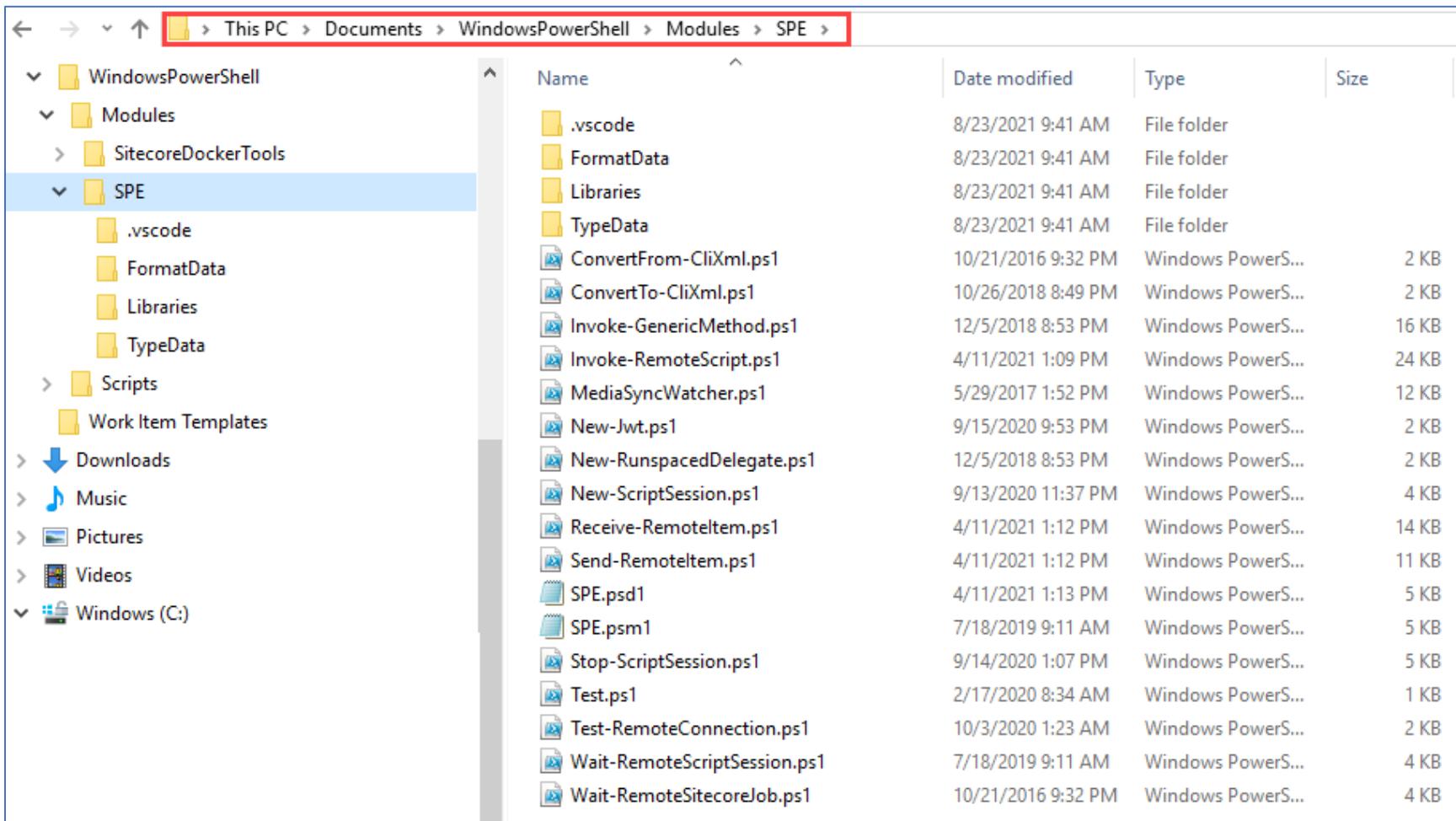
Download : <https://github.com/SitecorePowerShell/Console/releases/tag/6.3>

The screenshot shows the GitHub release page for Sitecore PowerShell Extensions version 6.3. The page includes the following sections:

- Latest release:** 6.3 (released by michaellwest on Jun 11, 5 commits to master)
- Version 6.3:** Survived the pandemic.
- Known Issues:** (Listed below the version note)
- File Hashes:** (Listed below the known issues)
- Assets:** 6 items listed:
 - Sitecore.PowerShell.Extensions-6.3.scwdp.zip (1.71 MB)
 - Sitecore.PowerShell.Extensions-6.3.zip (5.02 MB)
 - SPE.Minimal-6.3.zip (317 KB)
 - SPE.Remoting-6.3.zip** (32.8 KB) - This item is highlighted with a red box.
 - Source code (zip)

7. Sitecore PowerShell Commands – Remoting

Extract Zip File to Document Folder :

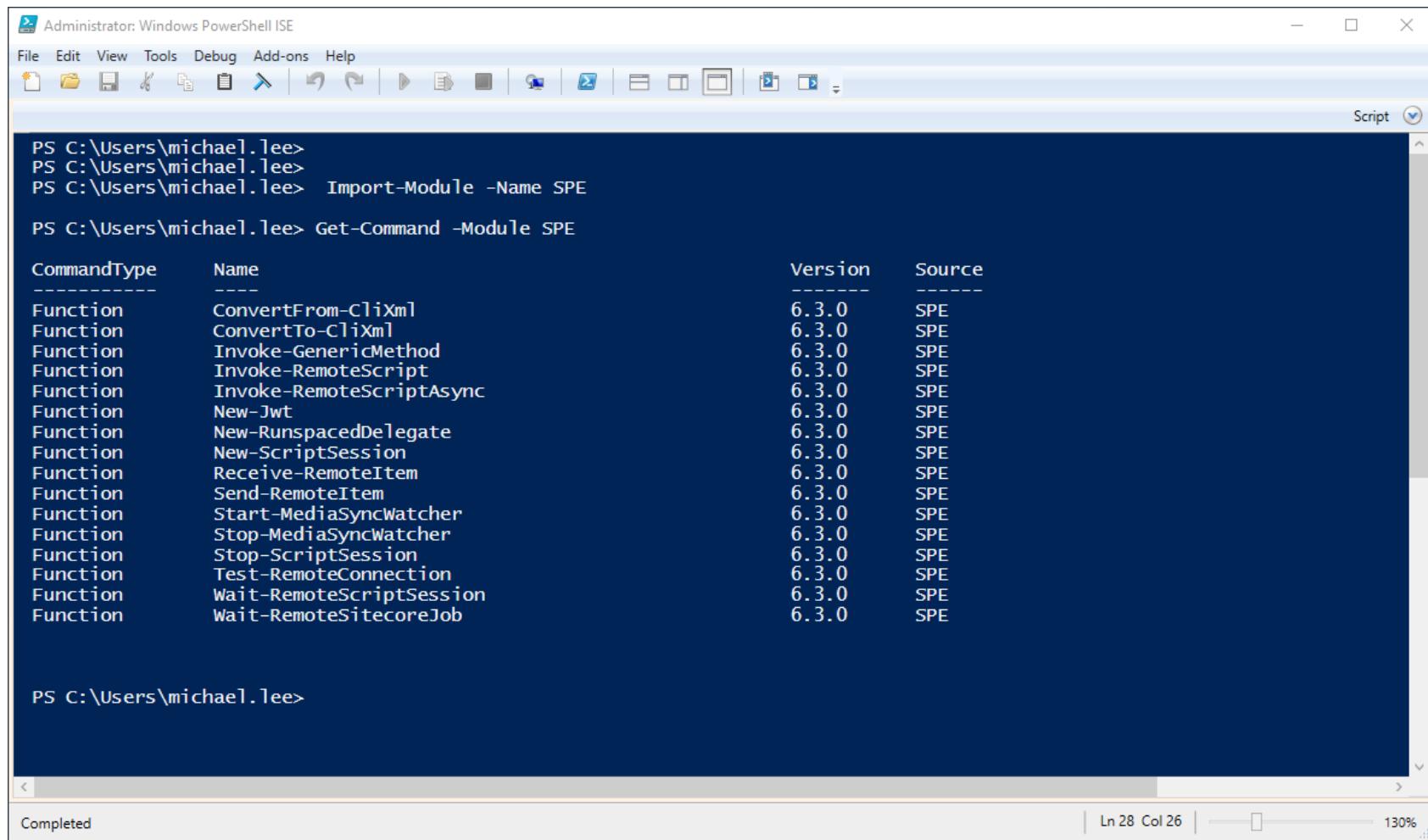


The screenshot shows a Windows File Explorer window with the following path highlighted: This PC > Documents > WindowsPowerShell > Modules > SPE. The 'SPE' folder is selected and highlighted with a blue selection bar.

Name	Date modified	Type	Size
.vscode	8/23/2021 9:41 AM	File folder	
FormatData	8/23/2021 9:41 AM	File folder	
Libraries	8/23/2021 9:41 AM	File folder	
TypeData	8/23/2021 9:41 AM	File folder	
ConvertFrom-CliXml.ps1	10/21/2016 9:32 PM	Windows PowerShell script file	2 KB
ConvertTo-CliXml.ps1	10/26/2018 8:49 PM	Windows PowerShell script file	2 KB
Invoke-GenericMethod.ps1	12/5/2018 8:53 PM	Windows PowerShell script file	16 KB
Invoke-RemoteScript.ps1	4/11/2021 1:09 PM	Windows PowerShell script file	24 KB
MediaSyncWatcher.ps1	5/29/2017 1:52 PM	Windows PowerShell script file	12 KB
New-Jwt.ps1	9/15/2020 9:53 PM	Windows PowerShell script file	2 KB
New-RunspacedDelegate.ps1	12/5/2018 8:53 PM	Windows PowerShell script file	2 KB
New-ScriptSession.ps1	9/13/2020 11:37 PM	Windows PowerShell script file	4 KB
Receive-Remoteltem.ps1	4/11/2021 1:12 PM	Windows PowerShell script file	14 KB
Send-Remoteltem.ps1	4/11/2021 1:12 PM	Windows PowerShell script file	11 KB
SPE.psd1	4/11/2021 1:13 PM	Windows PowerShell script file manifest	5 KB
SPE.psm1	7/18/2019 9:11 AM	Windows PowerShell script file	5 KB
Stop-ScriptSession.ps1	9/14/2020 1:07 PM	Windows PowerShell script file	5 KB
Test.ps1	2/17/2020 8:34 AM	Windows PowerShell script file	1 KB
Test-RemoteConnection.ps1	10/3/2020 1:23 AM	Windows PowerShell script file	2 KB
Wait-RemoteScriptSession.ps1	7/18/2019 9:11 AM	Windows PowerShell script file	4 KB
Wait-RemoteSitecoreJob.ps1	10/21/2016 9:32 PM	Windows PowerShell script file	4 KB

7. Sitecore PowerShell Commands – Remoting

PS > Import-Module –Name SPE **(Windows PowerShell)**



The screenshot shows a Windows PowerShell ISE window with the title "Administrator: Windows PowerShell ISE". The menu bar includes File, Edit, View, Tools, Debug, Add-ons, and Help. The toolbar contains various icons for file operations. The main pane displays the following PowerShell session:

```
PS C:\Users\michael.tee>
PS C:\Users\michael.tee>
PS C:\Users\michael.tee> Import-Module -Name SPE

PS C:\Users\michael.tee> Get-Command -Module SPE

 CommandType      Name          Version   Source
 ----           ----          -----   -----
 Function        ConvertFrom-Clixml    6.3.0    SPE
 Function        ConvertTo-Clixml    6.3.0    SPE
 Function        Invoke-GenericMethod 6.3.0    SPE
 Function        Invoke-RemoteScript  6.3.0    SPE
 Function        Invoke-RemoteScriptAsync 6.3.0    SPE
 Function        New-Jwt            6.3.0    SPE
 Function        New-RunspacedDelegate 6.3.0    SPE
 Function        New-ScriptSession    6.3.0    SPE
 Function        Receive-RemoteItem   6.3.0    SPE
 Function        Send-RemoteItem    6.3.0    SPE
 Function        Start-MediaSyncWatcher 6.3.0    SPE
 Function        Stop-MediaSyncWatcher 6.3.0    SPE
 Function        Stop-ScriptSession   6.3.0    SPE
 Function        Test-RemoteConnection 6.3.0    SPE
 Function        Wait-RemoteScriptSession 6.3.0    SPE
 Function        Wait-RemoteSitecoreJob 6.3.0    SPE

PS C:\Users\michael.tee>
```

The bottom status bar indicates "Completed" and "Ln 28 Col 26". The zoom level is set to 130%.

7. Sitecore PowerShell Commands – Remoting

Reference: <https://doc.sitecorepowershell.com/security>

- **Remoting** - Used when passing scripts to SPE for execution. Enable when using the **SPE Remoting** module. Service associated with `RemoteAutomation.ashx`.
- **RESTful v2** - Used when the url contains all the information needed to execute a script saved in the SPE library. Service associated with `RemoteScriptCall.ashx`.
 - Required for the following features: **PowerShell Web API**.
- **File Download** - Used when the url contains all the information needed to download a file from the server. Enable when using the **SPE Remoting** module. Service associated with `RemoteScriptCall.ashx`.
- **File Upload** - Used when the url contains all the information needed to upload a file to the server. Enable when using the **SPE Remoting** module. Service associated with `RemoteScriptCall.ashx`.
- **Media Download** - Used when the url contains all the information needed to download a media item from the server. Enable when using the **SPE Remoting** module. Service associated with `RemoteScriptCall.ashx`.
- **Media Upload** - Used when the url contains all the information needed to upload a media item to the server. Enable when using the **SPE Remoting** module. Service associated with `RemoteScriptCall.ashx`.
- **Handle Download** - Used when a file is downloaded through the Sitecore interface. Enable when using the **SPE Remoting** module. Service associated with `RemoteScriptCall.ashx`.
 - Required for the following features: **Out-Download** command.
- **Client** - Used for the SPE Console. Service associated with `PowerShellWebService.ashx`.
 - Required for the following features: **PowerShell Console**, **PowerShell ISE**,
- **Execution** - Used when SPE checks if the user has access to run the application.
 - Required for the following features: **Download File dialog**, **PowerShell Script Runner**, **Content Editor** (Context Menu, Insert Options, Ribbon).
- **RESTful v1** - Used in early version of SPE. Avoid using this if possible. Service associated with `RemoteScriptCall.ashx`.

```
<sitecore>
  <powershell>
    <services>
      <restfulv1 enabled="false" />
      <restfulv2 enabled="false" />
      <remoting enabled="false" /> True
      <fileDownload enabled="false" />
      <fileUpload enabled="false" />
      <mediaDownload enabled="false" />
      <mediaUpload enabled="false" />
      <handleDownload enabled="true" />
      <client enabled="true" />
      <execution enabled="true" />
    </services>
  </powershell>
</sitecore>
```

Restart IIS Website

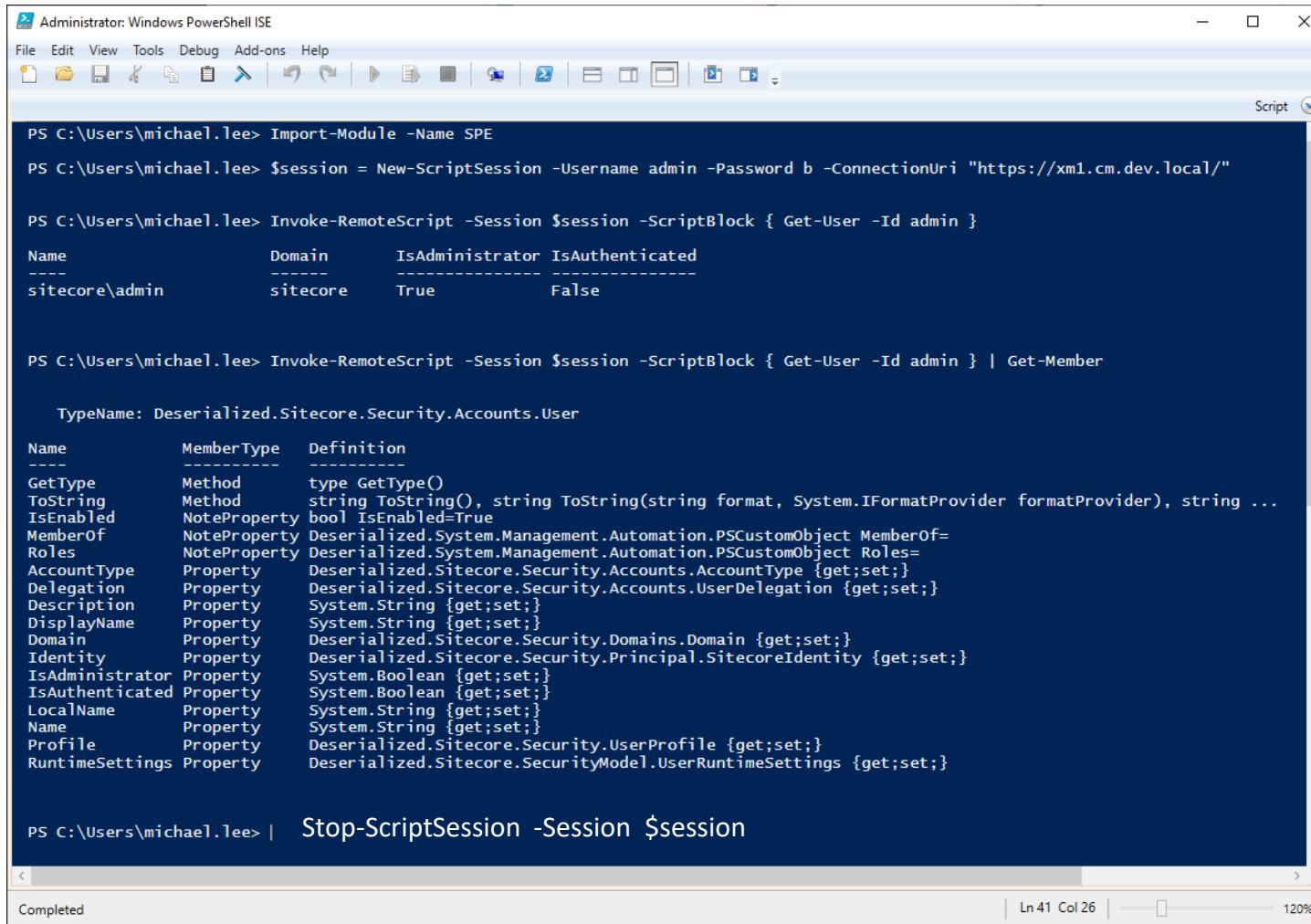
7. Sitecore PowerShell Commands – Remoting

```
70 <controlSources>
71   <source mode="on" namespace="Spe.Client.Controls" assembly="Spe" />
72   <source mode="on" namespace="Spe.Client.Applications" folder="/sitecore modules/Shell/PowerShell/" deep="true" />
73 </controlSources>
74 <powershell>
75   <workingDatabase>
76   <integrationPoints>
77   <services>
78     <restfulv1 enabled="true" requireSecureConnection="false">
79     <restfulv2 enabled="true" requireSecureConnection="false">
80       <remoting enabled="true" requireSecureConnection="true">
81         <authorization>
82           <patch:delete />
83         </authorization>
84         <authorization>
85           <add Permission="Allow" IdentityType="Role" Identity="sitecore\PowerShell Extensions Remoting" />
86           <add Permission="Allow" IdentityType="User" Identity="sitecore\admin" />
87         </authorization>
88       </remoting>
89       <fileDownload enabled="true" requireSecureConnection="false">
90       <fileUpload enabled="true" requireSecureConnection="false">
91       <mediaDownload enabled="true" requireSecureConnection="false">
92       <mediaUpload enabled="true" requireSecureConnection="false">
93       <handleDownload enabled="true" requireSecureConnection="false">
94       <client enabled="true" requireSecureConnection="false">
95       <execution enabled="true" requireSecureConnection="false">
96     </services>
97     <authenticationProvider type="Spe.Core.Settings.Authorization.SharedSecretAuthenticationProvider, Spe">
98       <!--<sharedSecret></sharedSecret> Uncomment this line and provide a strong randomized shared secret here. At least
99       <sharedSecret>FC6699B1752D75E60C96D8139B2518412376804FC7CDD58595BC47DFBF47199A</sharedSecret>
```

C:\inetpub\wwwroot\XM1.cm.dev.local\App_Config\Include\Spe\Spe.config

7. Sitecore PowerShell Commands – Remoting

Reference: <https://youtu.be/fGvT8eDdWrg>



The screenshot shows a Windows PowerShell ISE window with the following command history:

```
PS C:\Users\michael.lee> Import-Module -Name SPE
PS C:\Users\michael.lee> $session = New-ScriptSession -Username admin -Password b -ConnectionUri "https://xm1.cm.dev.local/"
PS C:\Users\michael.lee> Invoke-RemoteScript -Session $session -ScriptBlock { Get-User -Id admin }
Name          Domain      IsAdministrator IsAuthenticated
---          -----      -----           -----
sitecore\admin    sitecore        True            False

PS C:\Users\michael.lee> Invoke-RemoteScript -Session $session -ScriptBlock { Get-User -Id admin } | Get-Member

TypeName: Deserialized.Sitecore.Security.Accounts.User

Name      MemberType  Definition
---      -----
GetType  Method      type GetType()
ToString Method      string ToString(string format, System.IFormatProvider formatProvider), string ...
IsEnabled NoteProperty bool IsEnabled=True
MemberOf NoteProperty Deserialized.System.Management.Automation.PSCustomObject MemberOf=
Roles    NoteProperty Deserialized.System.Management.Automation.PSCustomObject Roles=
AccountType Property   Deserialized.Sitecore.Security.Accounts.AccountType {get;set;}
Delegation Property  Deserialized.Sitecore.Security.Accounts.UserDelegation {get;set;}
Description Property  System.String {get;set;}
DisplayName Property  System.String {get;set;}
Domain    Property   Deserialized.Sitecore.Security.Domains.Domain {get;set;}
Identity  Property   Deserialized.Sitecore.Security.Principal.SitecoreIdentity {get;set;}
IsAdministrator Property System.Boolean {get;set;}
IsAuthenticated Property System.Boolean {get;set;}
LocalName  Property  System.String {get;set;}
Name      Property  System.String {get;set;}
Profile   Property   Deserialized.Sitecore.Security.UserProfile {get;set;}
RuntimeSettings Property Deserialized.Sitecore.SecurityModel.UserRuntimeSettings {get;set;}

PS C:\Users\michael.lee> | Stop-ScriptSession -Session $session
```

The window includes standard Windows icons in the toolbar and a status bar at the bottom.

8. Sitecore PowerShell Commands – Remoting Security

Reference: <https://youtu.be/ZkOjG91q1a0>

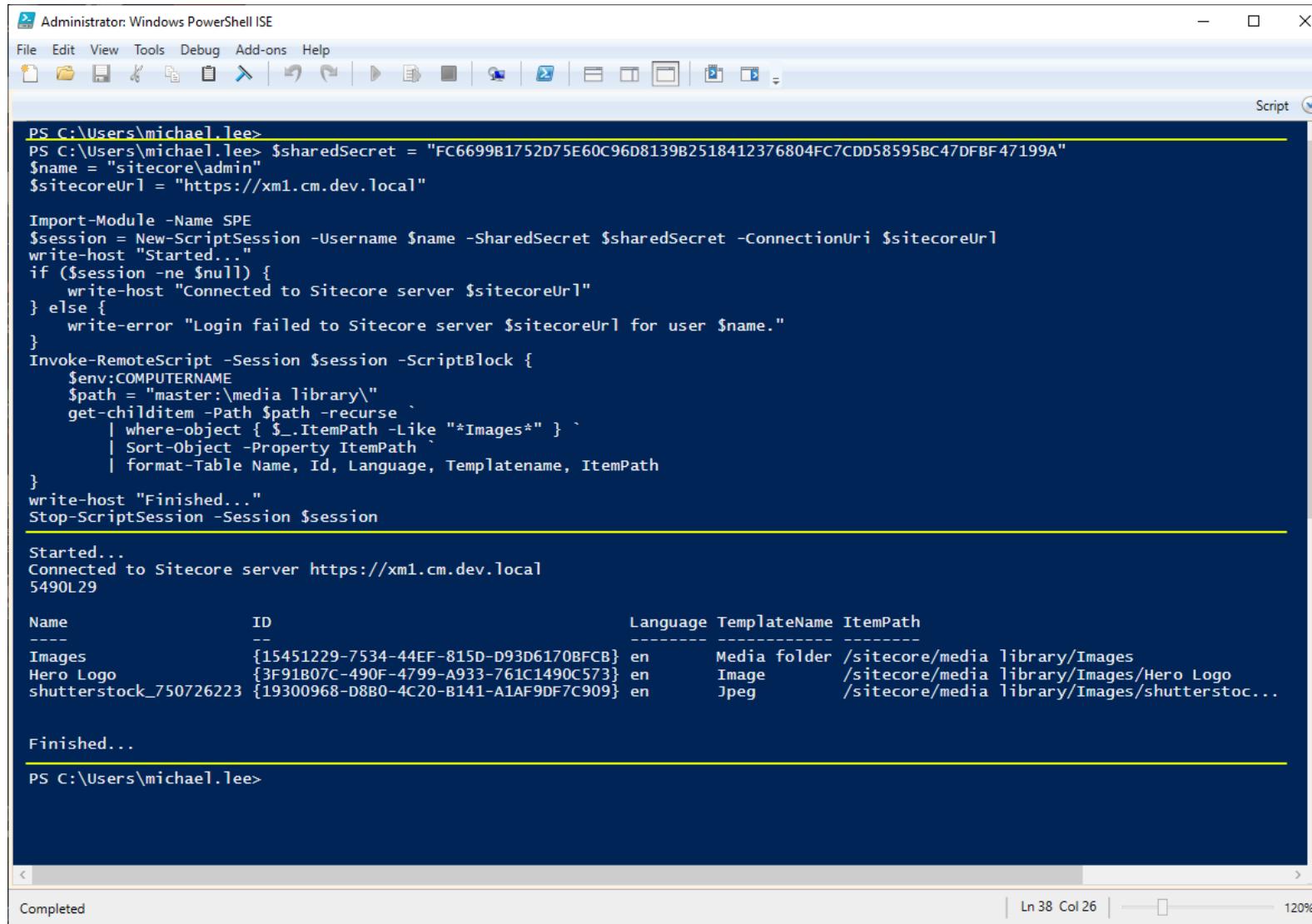
Gibson Research Corporation passwords : <https://www.grc.com/passwords.htm>

FC6699B1752D75E60C96D8139B2518412376804FC7CDD58595BC47DFBF47199A

SPE.Config

```
<configuration>
  <sitecore role:require="Standalone or ContentManagement" security:require="Sitecore">
    <powershell>
      <authenticationProvider type="Spe.Core.Settings.Authorization.SharedSecretAuthenticationProvider, Spe">
        <sharedSecret>FC6699B1752D75E60C96D8139B2518412376804FC7CDD58595BC47DFBF47199A</sharedSecret>
        <allowedAudiences hint="list">
          <audience>https://xm1.cm.dev.local/</audience>
        </allowedAudiences>
      </authenticationProvider>
    </powershell>
  </sitecore>
</configuration>
```

8. Sitecore PowerShell Commands – Remoting Security



The screenshot shows a Windows PowerShell ISE window with the following content:

```
Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
PS C:\Users\michael.lee>
PS C:\Users\michael.lee> $sharedSecret = "FC6699B1752D75E60C96D8139B2518412376804FC7CDD58595BC47DFBF47199A"
$name = "sitecore\admin"
$sitecoreUrl = "https://xm1.cm.dev.local"

Import-Module -Name SPE
$session = New-ScriptSession -Username $name -SharedSecret $sharedSecret -ConnectionUri $sitecoreUrl
write-host "Started..."
if ($session -ne $null) {
    write-host "Connected to Sitecore server $sitecoreUrl"
} else {
    write-error "Login failed to Sitecore server $sitecoreUrl for user $name."
}
Invoke-RemoteScript -Session $session -ScriptBlock {
    $env:COMPUTERNAME
    $path = "master:\media library\
    get-childitem -Path $path -recurse
        | where-object { $_.ItemPath -Like "*Images*" } ` 
        | Sort-Object -Property ItemPath
        | format-Table Name, Id, Language, TemplateName, ItemPath
}
write-host "Finished..."
Stop-ScriptSession -Session $session

Started...
Connected to Sitecore server https://xm1.cm.dev.local
5490L29

Name ID Language TemplateName ItemPath
--- -- -- -
Images {15451229-7534-44EF-815D-D93D6170BFCB} en Media folder /sitecore/media library/Images
Hero Logo {3F91B07C-490F-4799-A933-761C1490C573} en Image /sitecore/media library/Images/Hero Logo
shutterstock_750726223 {19300968-D8B0-4C20-B141-A1AF9DF7C909} en Jpeg /sitecore/media library/Images/shutterstoc...

Finished...
PS C:\Users\michael.lee>
```

The output shows the execution of a PowerShell script to connect to a Sitecore server and list items from the media library. The results are displayed as a table:

Name	ID	Language	TemplateName	ItemPath
Images	{15451229-7534-44EF-815D-D93D6170BFCB}	en	Media folder	/sitecore/media library/Images
Hero Logo	{3F91B07C-490F-4799-A933-761C1490C573}	en	Image	/sitecore/media library/Images/Hero Logo
shutterstock_750726223	{19300968-D8B0-4C20-B141-A1AF9DF7C909}	en	Jpeg	/sitecore/media library/Images/shutterstoc...

9. Unicorn

Why do we need serialization?

> Moving Sitecore development items between environments

- > Templates
- > Presentational items: layouts, renderings and sublayouts
- > Setting items

> Populating test/local environments with content from live instances

> **Sitecore packages**

- > Simple
- > Error-prone
- > Suitable for ad-hoc actions and full module packages

Unicorn :

Unicorn is a utility for Sitecore that solved the issue of moving templates, rendering and other database items between Sitecore instances and automatically serialize items

9. Unicorn

Install 'Unicorn and Rainbow' from NuGet

The screenshot shows the NuGet Package Manager interface with the search term 'unicorn' entered. The results list several packages, with the top result, 'Unicorn', highlighted by a red box. The right side of the screen displays the detailed view for the 'Unicorn' package, including its description, version (Latest stable 4.1.5), download count (1.66M), and an 'Install' button also highlighted with a red box. Below the description, there's a 'Description' section with details about the package's purpose and dependencies. To the right of the main window, the Visual Studio Solution Explorer shows a solution named 'GettingStarted' containing four projects: 1. Project, 2. Feature, 3. Foundation, and 4. Unicorn, which contains a sub-project named 'Serialization'. The 'GettingStarted.Unicorn.Serialization' project is expanded, showing its contents: Connected Services, Properties, References, App_Config (with an 'Include' folder), Controllers, Models, Views, and Web.config.

Browse Installed Updates NuGet Package Manager: GettingStarted.Unicorn.Serialization

unicorn Include prerelease Package source: nuget.org

Unicorn by Kam Figy, Mark Cassidy, 1.66M downloads 4.1.5
Unicorn is a utility for Sitecore that solves the issue of moving templates, renderings, and other database items between Sitecore instances. This package contains both the...

Unicorn.Core by Kam Figy, Mark Cassidy, 1.68M downloads 4.1.5
Unicorn is a utility for Sitecore that solves the issue of moving templates, renderings, and other database items between Sitecore instances.

Unicorn.Roles by Kam Figy, Mark Cassidy, 1.13M downloads 4.1.5
Adds security role syncing to Unicorn. This package contains both the core library and the configuration for it, which is appropriate for web projects. Install Unicorn.Roles.C...

Unicorn.Roles.Core by Kam Figy, Mark Cassidy, 1.19M downloads 4.1.5
Adds security role syncing to Unicorn. Install the non-core package to get config files.

Unicorn.Users by Kam Figy, Mark Cassidy, 758K downloads 4.1.5
Adds user syncing to Unicorn. This package contains both the core library and the configuration for it, which is appropriate for web projects. Install Unicorn.Users.Core if...

Unicorn.Users.Core by Kam Figy, Mark Cassidy, 800K downloads 4.1.5
Adds user syncing to Unicorn. Install the non-core package to get config files.

Unicorn.MSBuild by Bartłomiej Mucha, 6.8K downloads 1.0.0
Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.
 Do not show this again

Solution Explorer Search Solution Explorer (Ctrl+;) Solution 'GettingStarted' (4 of 4 projects)

1. Project
2. Feature
3. Foundation
4. Unicorn
 Serialization

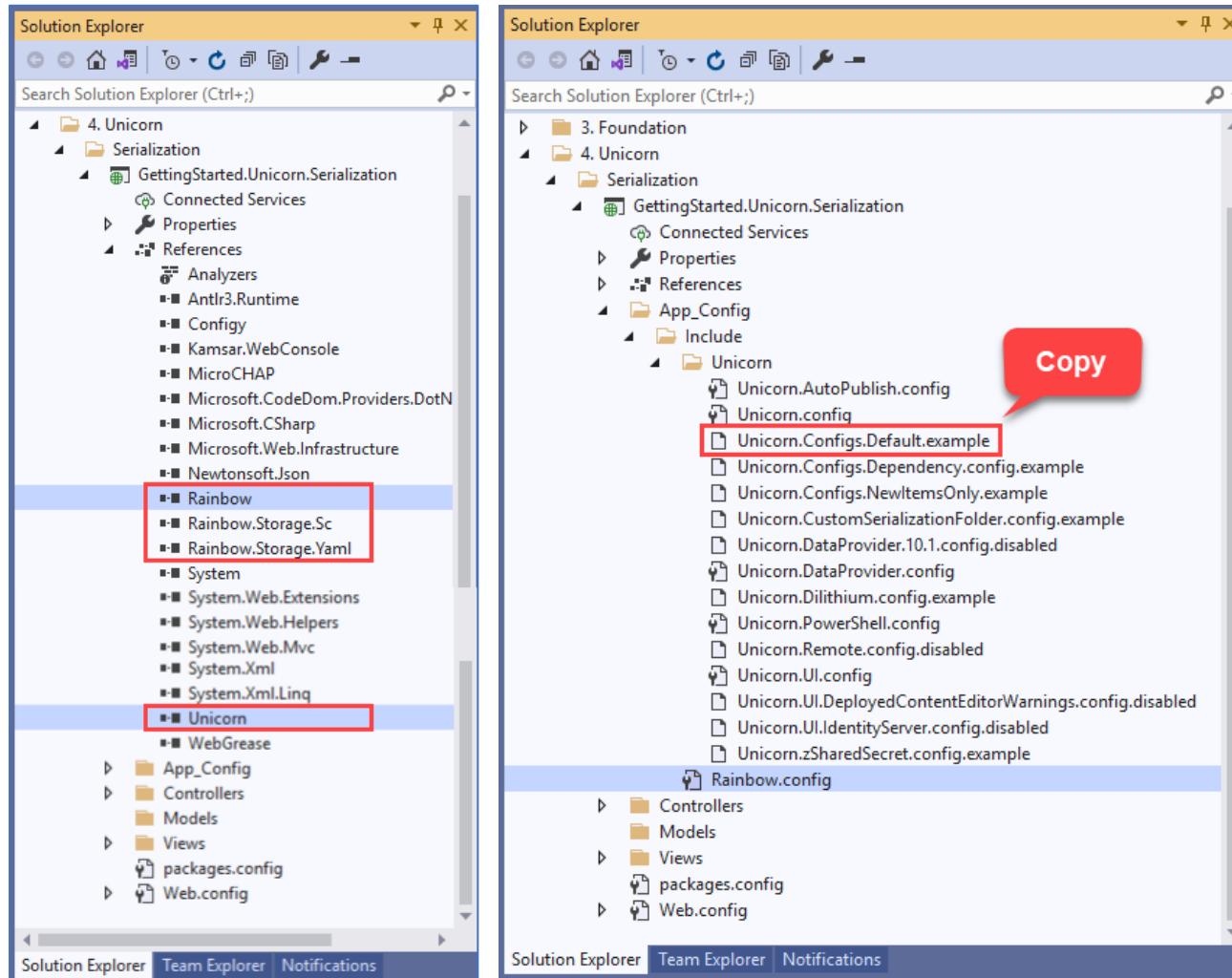
GettingStarted.Unicorn.Serialization

- Connected Services
- Properties
- References
- App_Config
 - Include
- Controllers
- Models
- Views
- Web.config

Solution Explorer Team Explorer Notifications Properties

9. Unicorn

Install 'Unicorn and Rainbow' from NuGet



1. Add new folder under Include folder
2. Copy example config and then put it new folder
3. Edit Config

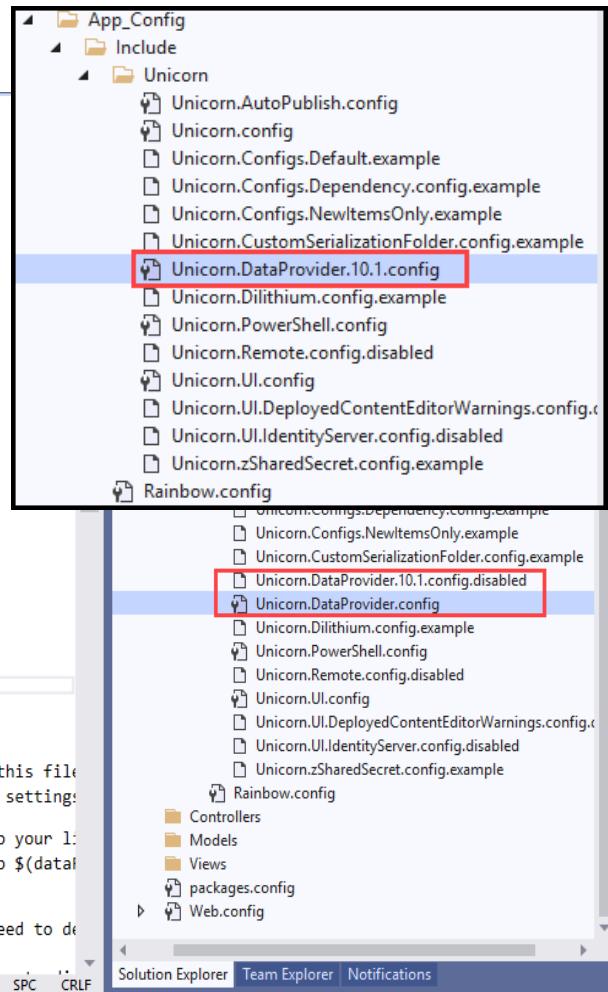
9. Unicorn

Create Unicorn Config, Change default Config

```
1 UNICORN README
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17 **** IMPORTANT NOTE FOR SITECORE VERSION 10.1+ ****
18
19 - Sitecore 10.1 introduces a change in the <database> and <dataProviders> sections of the Sitecore configuration.
20 - There is no easy way to make a patch configuration file that targets both
21 - Therefore; if you are on Sitecore 10.1 or above, replace
22   - Unicorn.DataProvider.config          (delete it)
23   - Unicorn.DataProvider.10.1.config.disabled (rename it to Unicorn.DataProvider.10.1.config) WITH
24
25 ABSOLUTELY DO READ THIS :P
26 ****
27 ****
28
29
30
31 Thanks for installing Unicorn! Here are some tips to get you started:
32
33 First, you should set up a configuration. This tells Unicorn what you want to keep serialized, among other things.
34 Make a copy of App_Config\Include\Unicorn\Unicorn.Configs.Default.example and rename it to .config. You can place this file
35 Review the comments in the example configuration file and edit the values as you see fit (especially the predicate setting).
36
37 It's probably also worth it to review the other App_Config\Include\Unicorn\*.config files too: make sure they're to your liking.
38 In particular, the TargetDataStore in Unicorn.config controls where serialized items are written to (it defaults to $(datastore)).
39 If you alter any of the defaults, do it in config patches so that you leave the default configuration intact.
40
41 Run a build, so that the Unicorn assemblies are copied to your bin folder. If you develop out of webroot you may need to do
42 Open the Unicorn control panel (defaults to /unicorn.aspx)
```

108% No issues found

Ln: 30 Ch: 1 SPC: CRLF



9. Unicorn

- Set targetDataStorage
- Set predicate
- Set enableTransparentSync

The screenshot shows a Visual Studio interface with the following details:

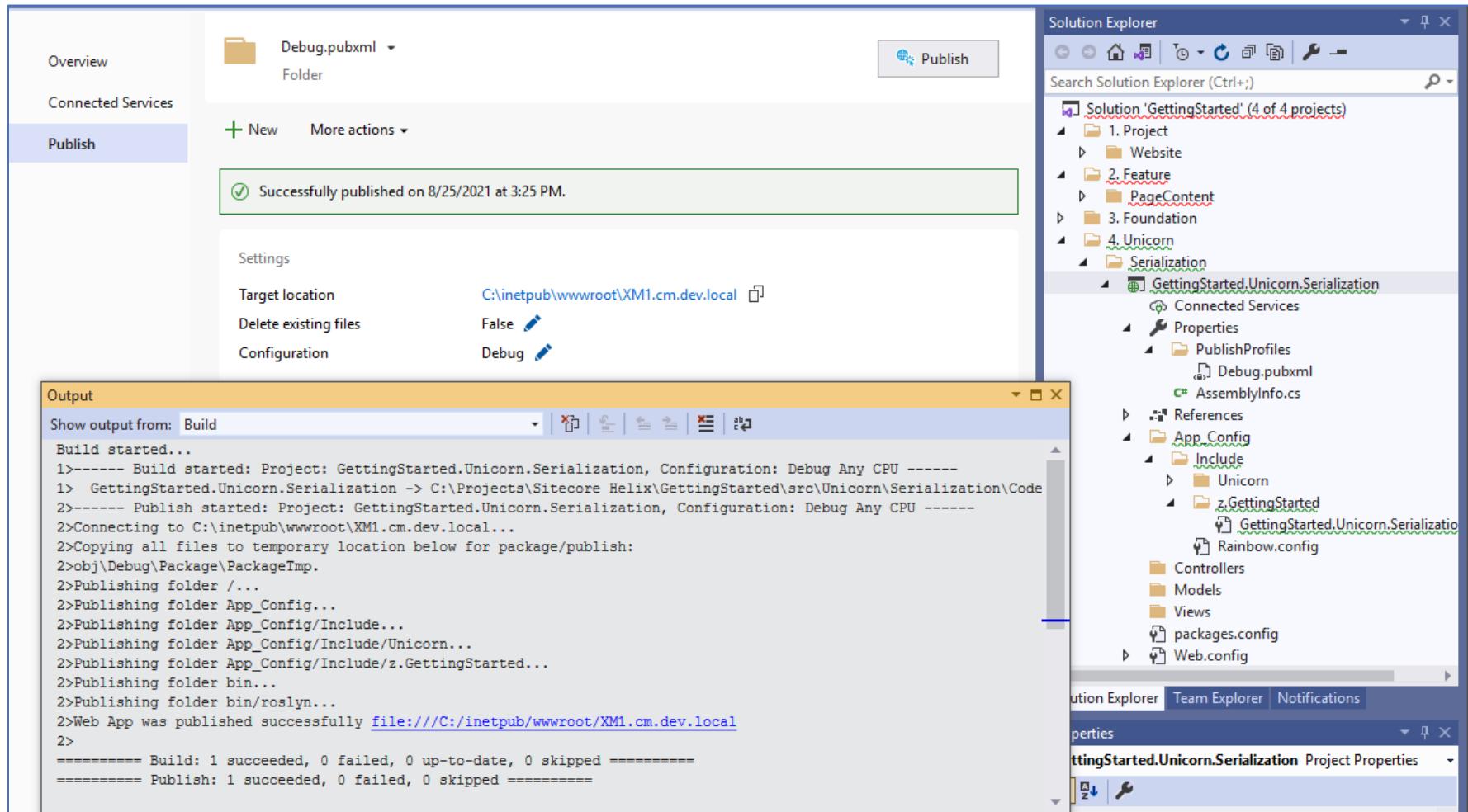
- Solution Explorer:** Shows a project named "4.Unicorn" containing a "GettingStarted.Unicorn.Serialization" folder. This folder includes files like "Connected Services", "Properties", "References", "App.Config", "Include", and "Unicorn".
- Code Editor:** Displays XML configuration code for Sitecore's Unicorn module. A specific line of code is highlighted with a blue box:

```
Automatic Sync : <dataProviderConfiguration enableTransparentSync="false" />
```
- Text Block:** Below the code editor, there is a text block with the following content:

```
:targetDataStore  
predicate  
<include name="content" database="master" path="/sitecore/content" />  
<include name="media library" database="master" path="/sitecore/media library" />
```
- Status Bar:** At the bottom, the status bar shows "Ln: 20 Ch: 17 TABS CRLF".

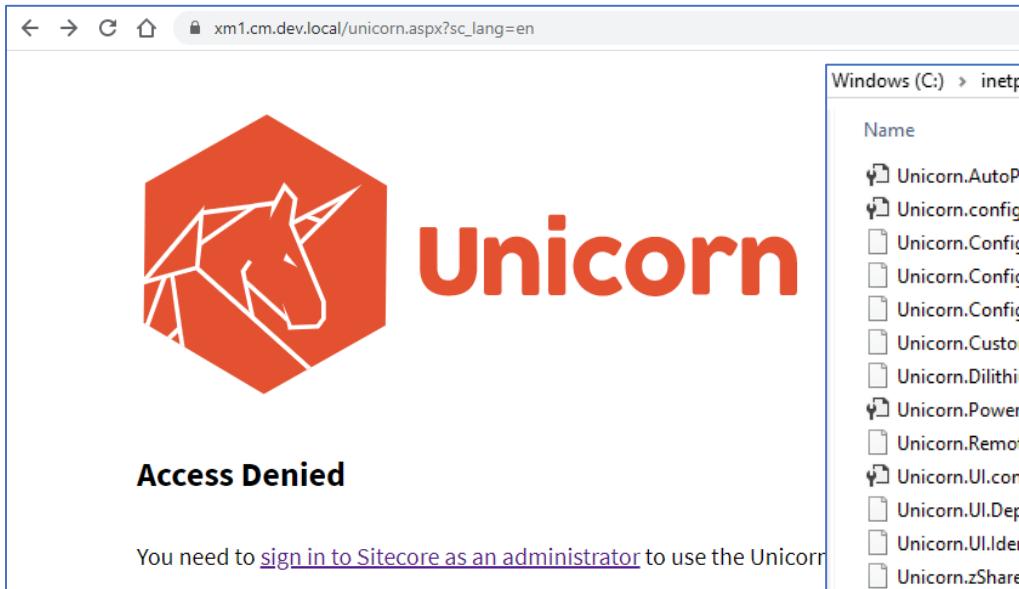
9. Unicorn

Publish – Unicorn, Rainbow, Config

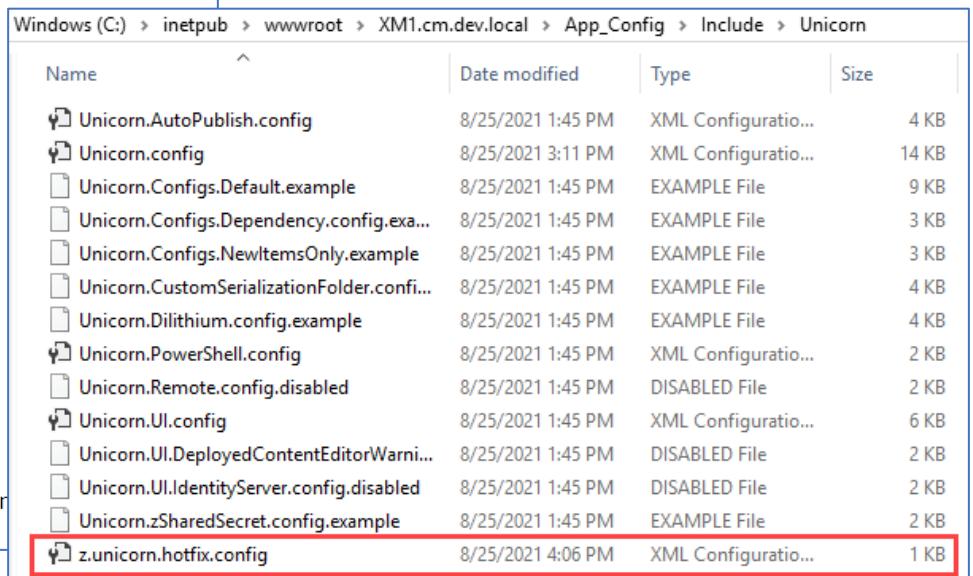


9. Unicorn

Login to Unicorn

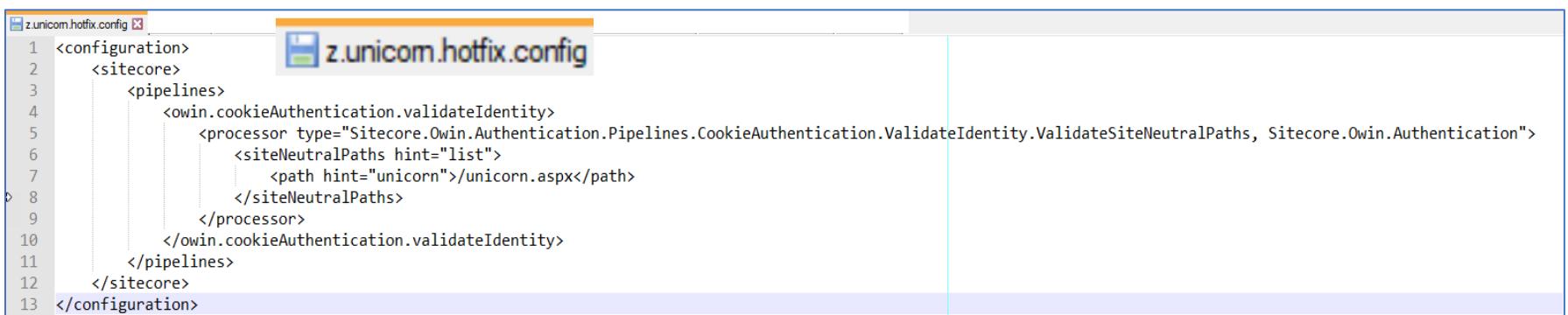


The screenshot shows a browser window with the URL `xm1.cm.dev.local/unicorn.aspx?sc_lang=en`. The page displays the Sitecore Unicorn logo (a red hexagon with a white unicorn head) and the word "Unicorn" in large red letters. Below the logo, the text "Access Denied" is displayed, followed by a message: "You need to [sign in to Sitecore as an administrator](#) to use the Unicorn".



A Windows File Explorer window is shown with the path `Windows (C:) > inetpub > wwwroot > XM1.cm.dev.local > App_Config > Include > Unicorn`. The folder contains several configuration files:

Name	Date modified	Type	Size
Unicorn.AutoPublish.config	8/25/2021 1:45 PM	XML Configuration	4 KB
Unicorn.config	8/25/2021 3:11 PM	XML Configuration	14 KB
Unicorn.Configs.Default.example	8/25/2021 1:45 PM	EXAMPLE File	9 KB
Unicorn.Configs.Dependency.config.example	8/25/2021 1:45 PM	EXAMPLE File	3 KB
Unicorn.Configs.NewItemsOnly.example	8/25/2021 1:45 PM	EXAMPLE File	3 KB
Unicorn.CustomSerializationFolder.config	8/25/2021 1:45 PM	EXAMPLE File	4 KB
Unicorn.Dilithium.config.example	8/25/2021 1:45 PM	EXAMPLE File	4 KB
Unicorn.PowerShell.config	8/25/2021 1:45 PM	XML Configuration	2 KB
Unicorn.Remote.config.disabled	8/25/2021 1:45 PM	DISABLED File	2 KB
Unicorn.Ul.config	8/25/2021 1:45 PM	XML Configuration	6 KB
Unicorn.Ul.DeployedContentEditorWarning.config	8/25/2021 1:45 PM	DISABLED File	2 KB
Unicorn.Ul.IdentityServer.config.disabled	8/25/2021 1:45 PM	DISABLED File	2 KB
Unicorn.zSharedSecret.config.example	8/25/2021 1:45 PM	EXAMPLE File	2 KB
z.unicorn.hotfix.config	8/25/2021 4:06 PM	XML Configuration	1 KB



The code editor displays the XML configuration file `z.unicorn.hotfix.config`. The file contains the following code:

```
1 <configuration>
2   <sitecore>
3     <pipelines>
4       <owin.cookieAuthentication.validateIdentity>
5         <processor type="Sitecore.Owin.Authentication.Pipelines.CookieAuthentication.ValidateIdentity.ValidateSiteNeutralPaths, Sitecore.Owin.Authentication">
6           <siteNeutralPaths hint="list">
7             <path hint="unicorn"/>/unicorn.aspx</path>
8           </siteNeutralPaths>
9         </processor>
10        </owin.cookieAuthentication.validateIdentity>
11      </pipelines>
12    </sitecore>
13  </configuration>
```

9. Unicorn

Login to Unicorn

The screenshot shows the Unicorn configuration interface. At the top left is the Sitecore logo and the word "Unicorn". Below it is a red hexagonal icon of a unicorn head. A blue callout bubble on the right side points to the text "Unicorn Configuration Name". The main content area has a red banner at the top stating: "Warning: at least one configuration has not serialized any items. review the configuration below and then perform initial serialization". Below this is a section titled "Configurations" with a sub-section "GettingStarted.Unicorn.Serialization". This section contains a red banner: "This configuration does not currently have any valid serialized items. which will write the current state of Sitecore to serialized items." Further down are sections for "Predicate", "Serialization Preset Predicate", "Target Data Store", and "Serialization File System Data Store". Each of these sections has a red box around specific configuration parameters: "content: master:/sitecore/content", "media library: master:/sitecore/media library", "Physical root path: C:\Projects\Sitecore Helix\GettingStarted\src\Unicorn\Serialization\Serialized Folder", and "Total internal SFS trees: 2".

Version 4.1.5 | [Options](#)

Warning: at least one configuration has not serialized any items. review the configuration below and then perform initial serialization.

Configurations

Some configurations prevent the 'sync all' checkbox because they include predicate base configurations.

GettingStarted.Unicorn.Serialization

GettingStarted.Unicorn.Serialization

This configuration does not currently have any valid serialized items. which will write the current state of Sitecore to serialized items.

Predicate

Predicates define which items are included or excluded in Unicorn.

Serialization Preset Predicate

Defines what to include in Unicorn based on XML configuration entries.

content: master:/sitecore/content
media library: master:/sitecore/media library

Target Data Store

Defines how items are serialized, for example to disk using YAML format.

Serialization File System Data Store

Stores serialized items on disk using the SFS tree format, where each root is a separate tree.

Serialization formatter: YAML Serialization Formatter

Physical root path: C:\Projects\Sitecore Helix\GettingStarted\src\Unicorn\Serialization\Serialized Folder
Total internal SFS trees: 2

9. Unicorn

Initial Setup

Would you like to perform an initial serialization of all configured items using the configuration outlined above now? This is required to start using this configuration.

[Perform Initial Serialization of GettingStarted.Unicorn.Serialization](#)

Click this to initial setup

xm1.cm.dev.local/unicorn.aspx?verb=Reserialize&configuration=GettingStarted.Unicorn.Serialization

 **Unicorn**

100%

Operation completed.



```
GettingStarted.Unicorn.Serialization is being reserialized.  
[D] Clearing existing items from Serialization File System Data Store (if any)  
[A] Serializing included items under root master:/sitecore/content (0de95ae4-41ab-4d01-9eb0-67441b7c2450)  
[A] Serializing included items under root master:/sitecore/media library (3d6658d8-a0bf-4e75-b3e2-d050fabcf4e1)  
GettingStarted.Unicorn.Serialization reserialization complete in 10092ms.
```

Operation completed in 10140ms. Want to [return to the control panel](#)?

9. Unicorn

Backup & Restore by Unicorn Serialization – Batch or Real time

The screenshot shows the Unicorn configuration interface at xm1.cm.dev.local/unicorn.aspx?verb=GetConfig. The page title is "Unicorn" with a red hexagonal logo. It displays the version "Version 4.1.5 | [Options](#)". Below the title, there's a section titled "Configurations" with a note about sync all checkboxes. A large green speech bubble contains the text "Backup : Sitecore database To Serialization". A red speech bubble contains the text "Restore : From Serialization To Sitecore database". At the bottom right are buttons for "Reserialize" and "Sync".

Version 4.1.5 | [Options](#)

Configurations

Some configurations prevent the 'sync all' checkbox because they include predicate base configurations.

GettingStarted.Unicorn.Serialization

GettingStarted.Unicorn.Serialization

Show Config

Backup :
Sitecore database
To Serialization

Restore :
From Serialization
To Sitecore database

Reserialize Sync

9. Unicorn

`<> Projects > Sitecore Helix > GettingStarted > src > Unicorn > Serialization > Serialized Folder > media library > media library > Images`

The left sidebar shows the project structure under `src`, with a red box highlighting the `Serialized Folder` and its subfolders: `content`, `Global Datasources`, `Home`, `media library`, `Default Website`, `Files`, `Images` (which is selected), `System`, `Forms`, `Simulator Backgrounds`, and `Template Thumbnails`.

The main pane displays the contents of the `Images` folder, showing two files: `Hero Logo.yml` and `shutterstock_750726223.yml`. The `Hero Logo.yml` file contains the following YAML configuration:

```

1  ---
2  ID: "3f91b07c-490f-4799-a933-761c1490c573"
3  Parent: "15451229-7534-44ef-815d-d93d6170bfcf"
4  Template: "f1828a2c-7e5d-4bbd-98ca-320474871548"
5  Path: /sitecore/media library/Images/Hero Logo
6  DB: master
7  SharedFields:
8  - ID: "06d5295c-ed2f-4a54-9bf2-26228d113318"
9    Hint: __Icon
10   Value: "/media/3F91B07C490F4799A933761C1490C573.ashx?h=16&thn=1&w=16"
11  - ID: "22eac599-f13b-4607-a89d-c091763a467d"
12    Hint: Width
13    Value: 342
14  - ID: "40e50ed9-ba07-4702-992e-a912738d32dc"
15    Hint: Blob
16    BlobID: "6d28546f-468f-4fe0-9ef2-20faf1e4ac2"
17    Value: iVBORw0KGgoAAAANSUhEUgAAVYAAACvCAYAAACmRfbmAAAACXBIVXMAAA7EAAA0xAGVkw4bAAAAAB3RJTUUH4wlWEiELIIkSywAAAAd
18  - ID: "6954b/c/-248/-423/-8608-436cb3b6dc0e"
19    Hint: Size
20    Value: 9344
21  - ID: "6f47a0a5-9c94-4b48-abeb-42d38def6054"
22    Hint: Mime Type
23    Value: image/png
24  - ID: "c06867fe-9a43-4c7d-b739-48780492d06f"
25    Hint: Extension
26    Value: png
27  - ID: "cb09946f-3218-4823-87d2-d5007c199a96"
28    Hint: Dimensions
29    Value: 342 x 175
30  - ID: "de2ca9e4-c117-4c8a-a139-1ff4b199d15a"
31    Hint: Height
32    Value: 175
33  Languages:
34  - Language: en
35  Versions:
36  - Version: 1
37    Fields:
38    - ID: "25bed78c-4957-4165-998a-ca1b52f67497"
39      Hint: __Created
40      Value: 20210820T173034Z
41    - ID: "5dd74568-4d4b-44c1-b513-0af5f4cd344f"
42      Hint: __Created by
43      Value: sitecore\Admin
44    - ID: "65885c44-8fcf-4a7f-94f1-ee63703fe193"
45      Hint: Alt
46      Value: Hero Logo
47

```

The right side shows a screenshot of the `Code Beautify` tool interface, specifically the `Base64 to Image` converter. It displays the Base64 string from the YAML file and the resulting generated image, which is a colorful graphic of the Sitecore logo.

Unicorn Config : targetDataStore

9. Unicorn

TEST: Remove 2 image file that monitored by Unicorn

The screenshot shows the Sitecore Media Library interface. A red callout points to the 'Remove' button on the left toolbar. Another red box highlights the 'Hero Logo' item in the 'Images' folder. A green callout labeled 'Unicorn' points to the yellow warning box in the 'Quick Info' section.

Media Library:

- Content
- Home
- Global Datasources
- Media Library
 - Default Website
 - Files
 - Images
 - Hero Logo (highlighted)
 - shutterstock_750726223
 - System
 - System
 - Templates

Remove

Hero Logo

This item is controlled by Unicorn
Changes to this item will be written to disk so they can be shared

Configuration: GettingStarted.Unicorn.Serialization
Predicate Component: media library
Physical path: C:\Projects\Sitecore Helix\GettingStarted\src\Unicorn\Serialization\Serialized Folder\media library\media library\Images\Hero Logo.yml

Quick Info

Item ID: {3F91B07C-490F-4799-A933-761C1490C573}
Item name: Hero Logo
Item path: /sitecore/media library/Images/Hero Logo

Breaking Links

Other items contain links to this item or its subitems. These links should be updated to prevent broken links.

Items to delete

Hero Logo
/sitecore/media library/Images/Hero Logo

Action

Remove links
 Link to another item
 Leave links - the links will appear in the Broken Links report

Edit Links

Continue Cancel

9. Unicorn

TEST: Remove 2 image file that monitored by Unicorn

The screenshot shows the Sitecore PowerShell Extensions interface. On the left, there's a browser window showing a 'Server Error in '/' Application' page with a 'Value cannot be null. Parameter name: item' error. The main content area displays the 'Configurations' section for the 'GettingStarted.Unicorn.Serialization' configuration. It includes a note about configurations preventing the 'sync all' checkbox, a 'Show Config' button, and two buttons: 'Reserialize' and 'Sync'. A green callout bubble points to the 'Sync' button. Below this, a yellow box contains a snippet of PowerShell code related to image fields.

Configurations

Some configurations prevent the 'sync all' checkbox because they include predicates that rely on (currently) invalid root base configurations.

GettingStarted.Unicorn.Serialization

GettingStarted.Unicorn.Serialization

Show Config

Reserialize Sync

Site display failed due to broken image link after Publish

```
Line 4: @{
Line 5:     Sitecore.Data.Fields.ImageField heroImageField = Model.Item.Fields["Hero Image"];
Line 6:     string heroImageUrl = Sitecore.Resources.Media.MediaManager.GetMediaUrl(heroImageField.MediaItem);
Line 7: }
Line 8:
```

9. Unicorn



TEST: Restore from Serialization

Operation completed.

100%

GettingStarted.Unicorn.Serialization is being synced.

```
* [U] en#1: Hero Logo
* [U] en#1: Hero Image
[U] master:/sitecore/content/Global Datasources/Hero Image (4b3dd090-5d93-43f0-9bfe-31e2d8eed515)
> Field Hero Image - en#1: Serialized '<image mediaid="{19300968-D8B0-4C20-B141-A1AF9DF7C909}" />', Source ''
[A] master:/sitecore/media library/Images/Hero Logo (3f91b07c-490f-4799-a933-761c1490c573)
[A] master:/sitecore/media library/Images/shutterstock_750726223 (19300968-d8b0-4c20-b141-a1af9df7c909)
GettingStarted.Unicorn.Serialization sync complete: 42 items evaluated, 3 items modified (2 added, 1 updated, 0 recycled) in 32183ms (~766.28ms/item).

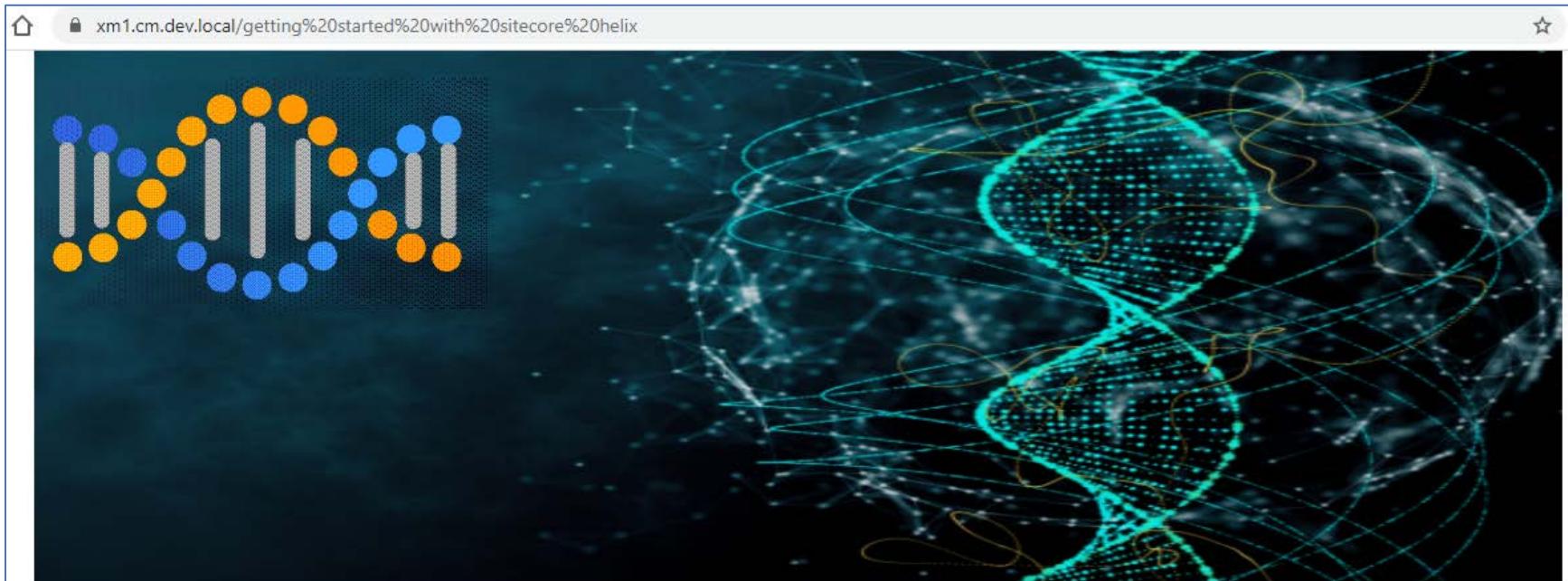
[P] Auto-publishing of synced items is beginning.
> Publishing 3 synced items in queue to web
> Published synced items to web. Statistics is not retrievable when Publish Manager is used (see setting Unicorn.UsePublishManager comments).

Operation completed in 38949ms. Want to return to the control panel?
```

Broken image link
is recovered

9. Unicorn

TEST: Publish again after Unicorn Sync – All images and database content is restored



Getting Started with Sitecore Helix - Editable

Configuring a Development Environment for Sitecore

- Windows Operating System
- Visual Studio 2019
- SQL Server 2017
- Solr 7.5

10. Sitecore PowerShell Commands with Unicorn 4.1.5, SPE 6.3 & Remoting

Reference: <https://github.com/michaellwest/spe-content-migrator>
: <https://youtu.be/ZkOjG91q1a0> - What's New in 6.2 ?

Prerequisites:

- **Unicorn** and **Rainbow** are deployed to your Sitecore instances.
- **Spe 6.0+** is installed and **SPE Remoting** is enabled on your Sitecore instances.

Setup 2 Sitecore Sites

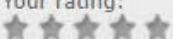
1. Install Unicorn/Rainbow into Sitecore 7.2
2. Install SPE 6.0+ and Remoting on Sitecore 7.2 – **7.2 is Not Compatible**
3. Install Unicorn/Rainbow into Sitecore 10.1 - Page 54
4. Install SPE 6.3 and Remoting on Sitecore 10.1 - Page 46

10. Sitecore PowerShell Commands with Unicorn 4.1.5, SPE 6.3 & Remoting

- **SPE 6.0 and above – Not compatible for Sitecore 7.2**

SITECORE POWERSHELL EXTENSIONS



Family: Shared Source Average rating:  61 ratings
Category: Webmaster Tools Requirements: CMS 6.4, CMS 6.4.1, CMS 6.5, CMS 6.6, CMS 7.0, CMS 7.1, CMS 7.2, CMS 7.5, SXP 8.0, SXP 8.1, SXP 8.2, SXP 9.0
Your rating: 
Tags: CMS;Administrators;ASPNET;Developers;DMS
Release date: 9/11/2012 4:10:18 AM Revision date: 9/23/2019 4:25:50 PM
Source: GitHub Provider: Sitecore PowerShell Extensions Team

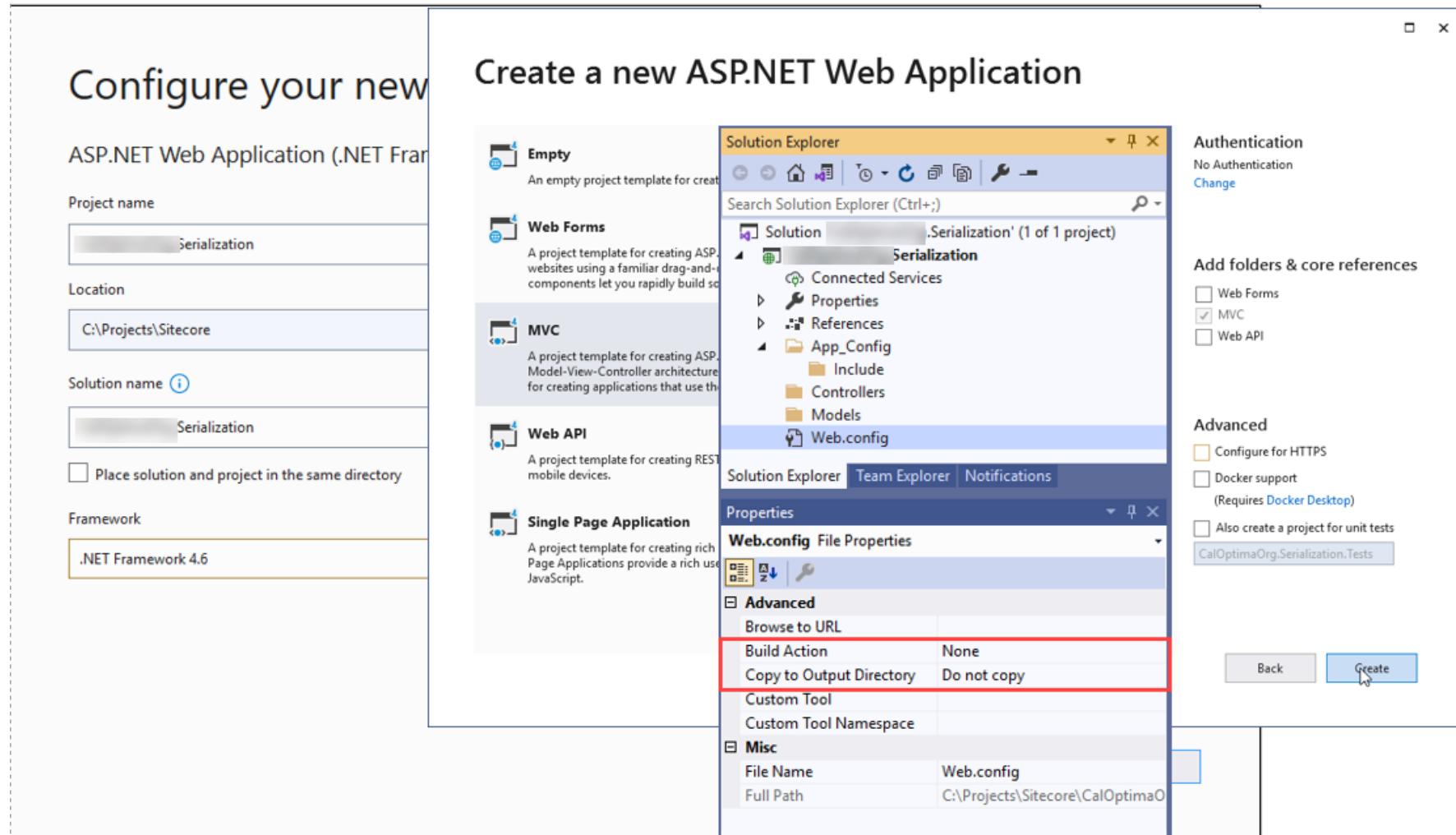
[DOWNLOAD](#) [FOLLOW](#) [REVIEW](#) 

SPE 6.0+ is not good for Sitecore 7.2 version

SPE 5.1 - Compatible for Sitecore 8.0-9.2.
SPE 5.0 - Compatible for Sitecore 8.0-9.1.
SPE 4.7+ Compatible for Sitecore 7.0-9.1.

10. Sitecore PowerShell Commands with Unicorn

- Install Unicorn 4.1.0/Rainbow into Sitecore 7.2 website



10. Sitecore PowerShell Commands with Unicorn

- Install Unicorn/Rainbow Version 4.1.0 for Sitecore 7.2

The screenshot shows the NuGet Package Manager interface. On the left, the search results for 'unicorn' are displayed, listing several packages by Kam Figy and Mark Cassidy, all versioned at 4.1.5. The 'Unicorn' package is highlighted. On the right, the detailed view for the 'Unicorn' package is shown, specifically for version 4.1.0. The 'Install' button is visible. The central pane displays the package's description, version history, authors, license, and dependencies. The 'Dependencies' section lists 'Any,Version:=v0.0' as a dependency for 'Unicorn.Core (>= 4.1.0)' and 'Rainbow (>= 2.1.0)'. The Solution Explorer on the right shows a single project named 'Serialization' containing files like App_Config, Include, Controllers, Models, and Web.config.

Browse Installed Updates NuGet Package Manager: Serialization

unicorn X Include prerelease Package source: nuget.org

Unicorn by Kam Figy, Mark Cassidy, 1.66M downloads 4.1.5
Unicorn is a utility for Sitecore that solves the issue of moving templates, renderings, and other database items between Sitecore instances.

Unicorn.Core by Kam Figy, Mark Cassidy, 1.69M downloads 4.1.5
Unicorn is a utility for Sitecore that solves the issue of moving templates, renderings, and other database items between Sitecore instances.

Unicorn.Roles by Kam Figy, Mark Cassidy, 1.13M downloads 4.1.5
Adds security role syncing to Unicorn. This package contains both the core library and the configuration for it, which is appropriate for web...

Unicorn.Roles.Core by Kam Figy, Mark Cassidy, 1.19M download 4.1.5
Adds security role syncing to Unicorn. Install the non-core package to get config files.

Unicorn.Users by Kam Figy, Mark Cassidy, 759K downloads 4.1.5
Adds user syncing to Unicorn. This package contains both the core library and the configuration for it, which is appropriate for web proj...

Unicorn.Users.Core by Kam Figy, Mark Cassidy, 801K download 4.1.5
Adds user syncing to Unicorn. Install the non-core package to get config files.

Unicorn.MSBuild by Bartłomiej Mucha, 6.82K downloads 1.0.0

Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.

Do not show this again

Unicorn nuget.org

Version: 4.1.0 Install

Options

Description

Unicorn is a utility for Sitecore that solves the issue of moving templates, renderings, and other database items between Sitecore instances. This package contains both the Unicorn core library and the configuration for it, which is appropriate for web projects. Install Unicorn.Core if you only want the library.

Version: 4.1.0

Author(s): Kam Figy, Mark Cassidy

License: [View License](#)

Date published: Sunday, June 2, 2019 (6/2/2019)

Project URL: <https://github.com/SitecoreUnicorn/Unicorn>

Report Abuse: <https://www.nuget.org/packages/Unicorn/4.1.0/ReportAbuse>

Tags: sitecore, serialization

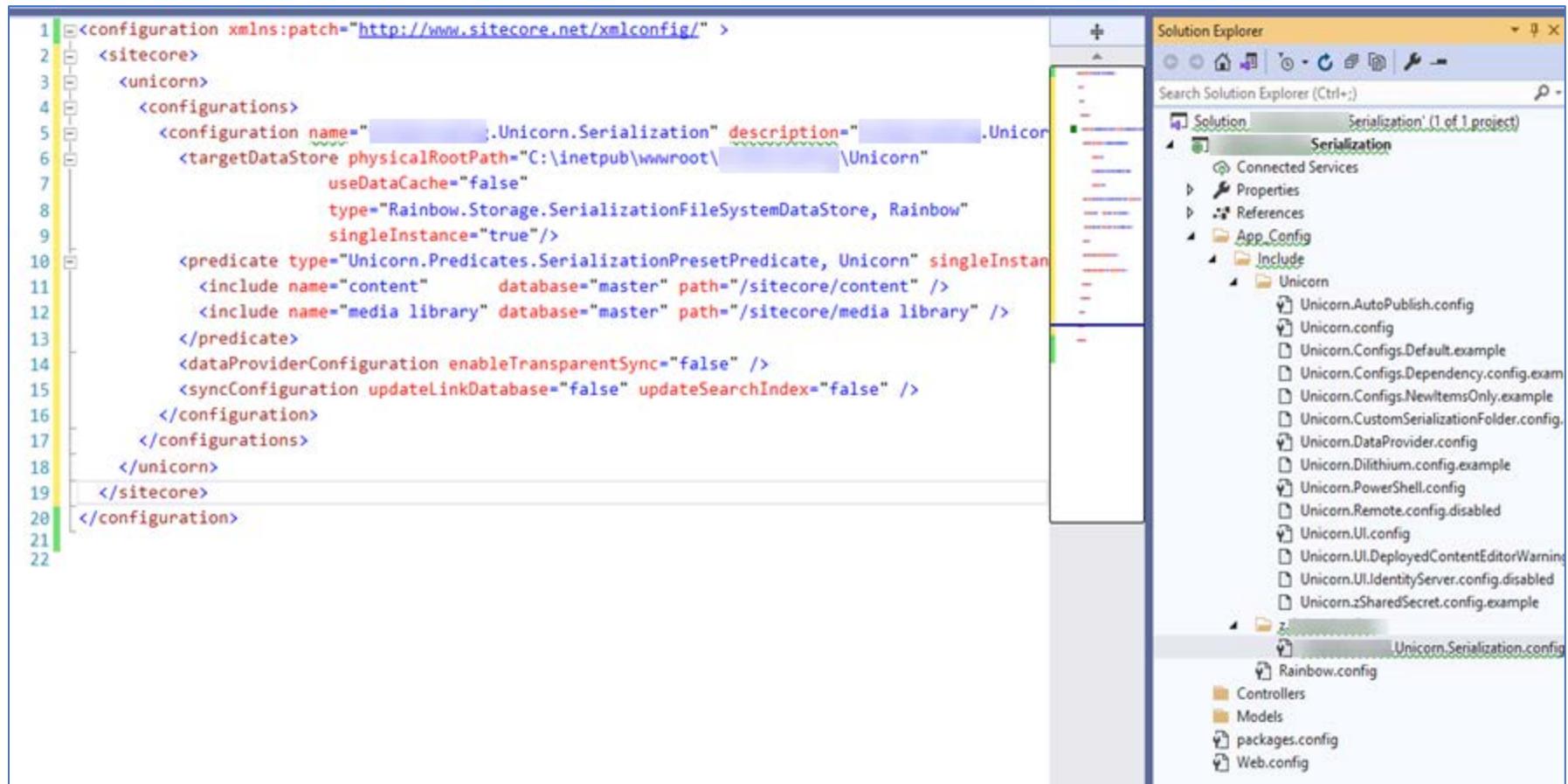
Dependencies

- Any,Version:=v0.0
 - Unicorn.Core (>= 4.1.0)
 - Rainbow (>= 2.1.0)

Solution Explorer Team Explorer Notifications

10. Sitecore PowerShell Commands with Unicorn

- Set Unicorn Config

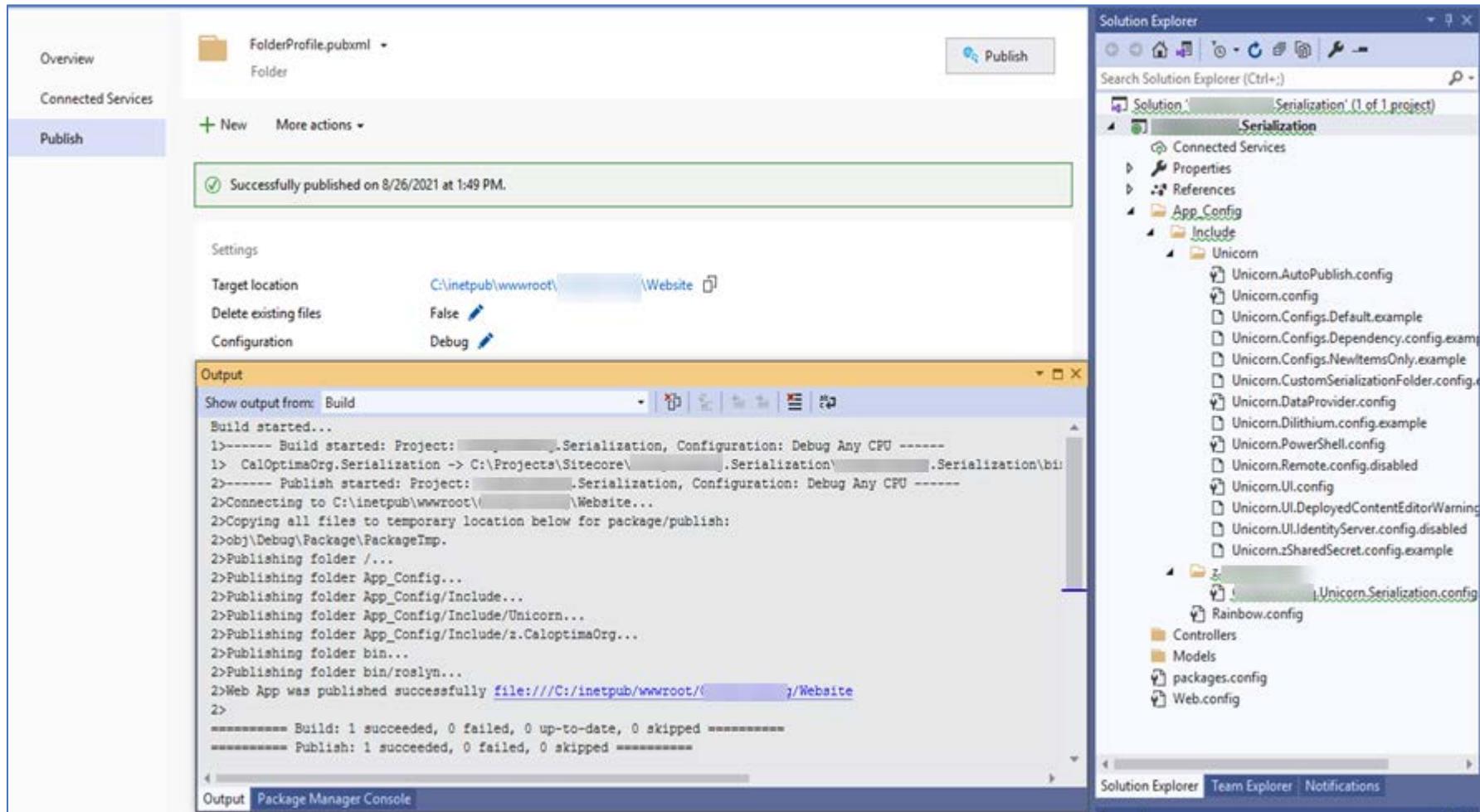


The screenshot shows the Visual Studio interface with the Solution Explorer and a code editor. The code editor displays Sitecore configuration XML for Unicorn serialization. The Solution Explorer shows a project named 'Serialization' containing various configuration files like 'Unicorn.config', 'Rainbow.config', and 'Web.config'. The 'App_Config' folder contains 'Include' and 'Unicorn' subfolders with files such as 'Unicorn.AutoPublish.config', 'Unicorn.Serialization.config', and 'Unicorn.zSharedSecret.config.example'.

```
<configuration xmlns:patch="http://www.sitecore.net/xmlconfig/" >
  <sitecore>
    <unicorn>
      <configurations>
        <configuration name=".Unicorn.Serialization" description=".Unicorn"
          <targetDataStore physicalRootPath="C:\inetpub\wwwroot\Unicorn"
            useDataCache="false"
            type="Rainbow.Storage.SerializationFileSystemDataStore, Rainbow"
            singleInstance="true"/>
        <predicate type="Unicorn.Predicates.SerializationPresetPredicate, Unicorn" singleInstance="true">
          <include name="content" database="master" path="/sitecore/content" />
          <include name="media library" database="master" path="/sitecore/media library" />
        </predicate>
        <dataProviderConfiguration enableTransparentSync="false" />
        <syncConfiguration updateLinkDatabase="false" updateSearchIndex="false" />
      </configuration>
    </configurations>
  </unicorn>
</sitecore>
</configuration>
```

10. Sitecore PowerShell Commands with Unicorn

- Publish to Sitecore 7.2 Website



10. Sitecore PowerShell Commands with Unicorn

- Unicorn installed on Sitecore 7.2 Website

Version 4.1.0 | Options

Not secure | /unicorn.aspx?verb=

Warning: at least one configuration has not serialized any items yet. Unicorn cannot operate properly until this is complete. Please review the configuration below and then perform initial serialization if it is accurate.

.Unicorn.Serialization

Unicorn.Serialization

This configuration does not currently have any valid serialized items. You cannot sync it until you perform an initial serialization, which will write the current state of Sitecore to serialized items.

Predicate

Predicates define which items are included or excluded in Unicorn.

Serialization Preset Predicate

Defines what to include in Unicorn based on XML configuration entries.

content: master:/sitecore/content
media library: master:/sitecore/media library

Target Data Store

Defines how items are serialized, for example to disk using YAML format.

Serialization File System Data Store

Stores serialized items on disk using the SFS tree format, where each root is a separate tree.

Serialization formatter: YAML Serialization Formatter
Physical root path: C:\inetpub\wwwroot\Unicorn
Total internal SFS trees: 2



Unicorn

10. Sitecore PowerShell Commands with Unicorn 4.1.0

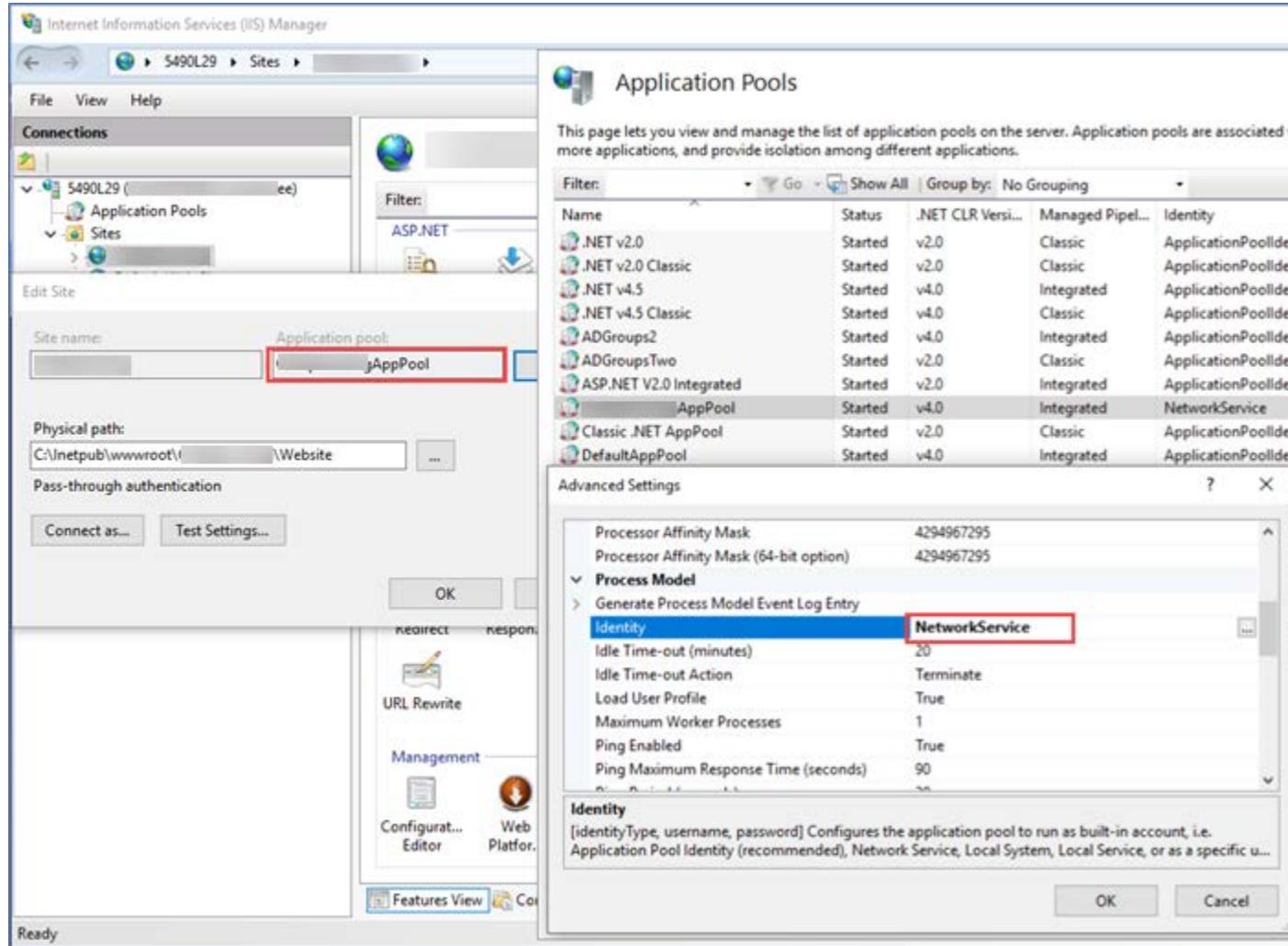
The screenshot shows a browser window with two tabs. The top tab is titled ".Unicorn.Serialization" and displays the "Initial Setup" page for Unicorn. It asks if you want to perform an initial serialization of all configured items. A red button labeled "Perform Initial Serialization of .Unicorn.Serialization" is visible. A speech bubble points to a red error message: "Serialization Folder access denied (Under website)". The bottom tab is also titled ".Unicorn.Serialization" and shows a PowerShell session output. The output includes a log message about reserialization, clearing existing items from the serialization file system data store, and an error message: "ERROR: Access to the path 'C:\inetpub\wwwroot\...\Unicorn\content' is denied. (System.IO.IOException)". The error details the stack trace for the directory deletion process. At the bottom of the PowerShell window, there is a message: "Operation completed in 1032ms. Want to [return to the control panel?](#)".

```
g.Unicorn.Serialization is being reserialized.
[D] Clearing existing items from Serialization File System Data Store (if any)
ERROR: Access to the path 'C:\inetpub\wwwroot\...\Unicorn\content' is denied. (System.IO.IOException)
at System.IO.Directory.DeleteHelper(String fullPath, String userPath, Boolean recursive, Boolean throwOnTopLevelDirectoryNotFound, WIN32_FIND_DATA& data)
at System.IO.Directory.Delete(String fullPath, String userPath, Boolean recursive, Boolean checkHost)
at Rainbow.Storage.SerializationFileSystemDataStore.ClearAllFiles()
at Rainbow.ActionRetryer.Perform(Action action, Int32 delayInMs)
at Rainbow.Storage.SerializationFileSystemDataStore.Clear()
at Unicorn.SerializationHelper.ReserializeConfigurations(IConfiguration[] configurations, IProgressStatus progress, ILogger additionalLogger)

Operation completed in 1032ms. Want to return to the control panel?
```

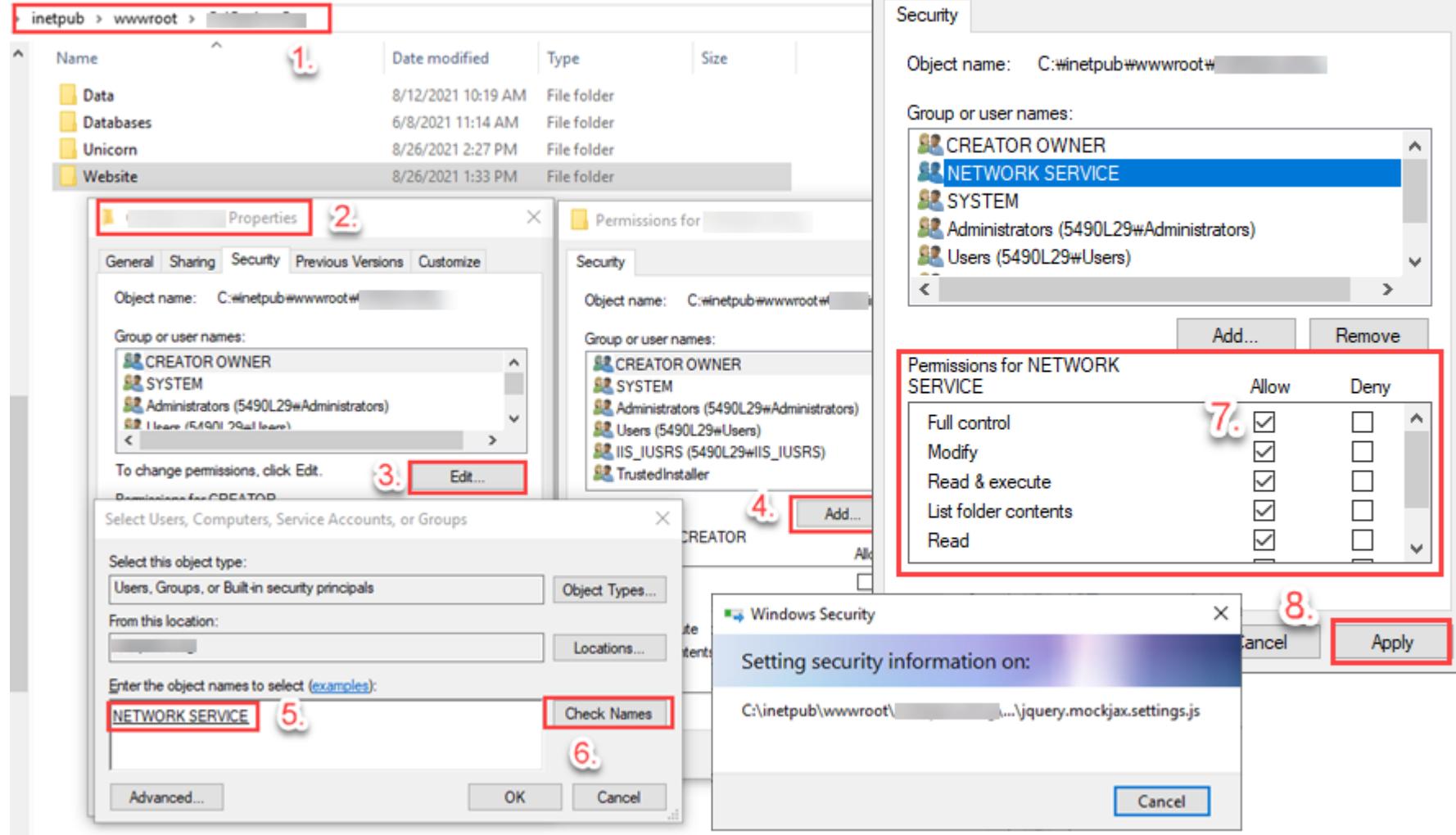
10. Sitecore PowerShell Commands with Unicorn 4.1.0

- Find WebSite's ApplicationPool Identity



10. Sitecore PowerShell Commands with Unicorn 4.1.0

- Add Identity User to Website folder property



10. Sitecore PowerShell Commands with Unicorn 4.1.0

- Rerun Initial Setup

return to the control panel?'."/>

unicorn.aspx?verb=Reserialize&configuration=.Unicorn.Serialization

Not secure | .Unicorn.Serialization

100%

This PC > Windows (C:) > inetpub > wwwroot > [redacted] > Unicorn > content >

Name	Date modified	Type	Size
content	8/27/2021 3:56 PM	File folder	
content.yaml	8/27/2021 3:56 PM	Yaml Source File	3 KB

wwwroot
applications
aspnet_client
[redacted]
Data
Databases
Unicorn
content
media library
Website

.Unicorn.Serialization is being reserialized.
[D] Clearing existing items from Serialization File System Data Store (if any)
[A] Serializing included items under root master:/sitecore/content (0de95ae4-41ab-4d01-9eb0-67441b7c2450)
[A] Serializing included items under root master:/sitecore/media library (3d6658d8-a0bf-4e75-b3e2-d050fabcf4e1)
.Unicorn.Serialization reserialization complete in 820ms.

Operation completed in 822ms. Want to [return to the control panel?](#)

11. Sitecore PowerShell Commands with Unicorn – Backup & Restore

1. Backup Sitecore serialize to Backup Folder

GettingStarted.Unicorn.Serialization Details

Predicate
Predicates define which items are included or excluded in Unicorn.

Serialization Preset Predicate
Defines what to include in Unicorn based on XML configuration entries.

content: master:/sitecore/content
media library: master:/sitecore/media library
templates: master:/sitecore/templates
layout: master:/sitecore/layout

Target Data Store
Defines how items are serialized, for example to disk using YAML format.

Serialization File System Data Store
Stores serialized items on disk using the SFS tree format, where each root is a separate tree.
Serialization formatter: YAML Serialization Formatter
Physical root path: C:\Projects\Sitecore Helix\GettingStarted\src\Unicorn\Serialization\Serialized Folder
Total internal SFS trees: 4

GettingStarted.Unicorn.Serialization

GettingStarted.Unicorn.Serialization

Show Config

Reserialize Sync

11. Sitecore PowerShell Commands with Unicorn – Backup & Restore

1. Backup Source Sitecore serialize folder to Backup folder location

The screenshot shows the Sitecore Experience Platform interface. On the left, there is a navigation tree with items like 'content', 'media library', 'Default Webs', 'Files', 'Images', and 'System'. A 'Serialized Folder' item under 'Unicorn' is selected. On the right, there is a file browser window showing two YAML files: 'Hero Logo.yml' and 'shutterstock_750726223.yml'. Below the file browser, a web browser window displays the URL 'xm1.cm.dev.local/Getting%20Started%20With%20Sitecore%20Helix'. The page content includes the Sitecore logo, a woman smiling, and the text 'xm1.cm.dev.local/'. At the bottom of the browser window, there is a dark banner with the URL again.

Getting Started with Sitecore Helix - Editable

Configuring a Development Environment for Sitecore

11. Sitecore PowerShell Commands with Unicorn – Backup & Restore

2. Copy Source Serialized folder to Target site Serialized folder

Name	Date modified	Type	Size
Hero Logo.yml	8/30/2021 2:09 PM	YML File	14 KB
shutterstock_750726223.yml	8/30/2021 2:09 PM	YML File	6,976 KB

xm1cm.Unicorn.Serialization Details

Predicate

Predicates define which items are included or excluded in Unicorn.

Serialization Preset Predicate

Defines what to include in Unicorn based on XML configuration entries.

content: master:/sitecore/content
media library: master:/sitecore/media library
templates: master:/sitecore/templates
layout: master:/sitecore/layout

Target Data Store

Defines how items are serialized, for example to disk using YAML format.

Serialization File System Data Store

Stores serialized items on disk using the SFS tree format, where each root is a separate tree.

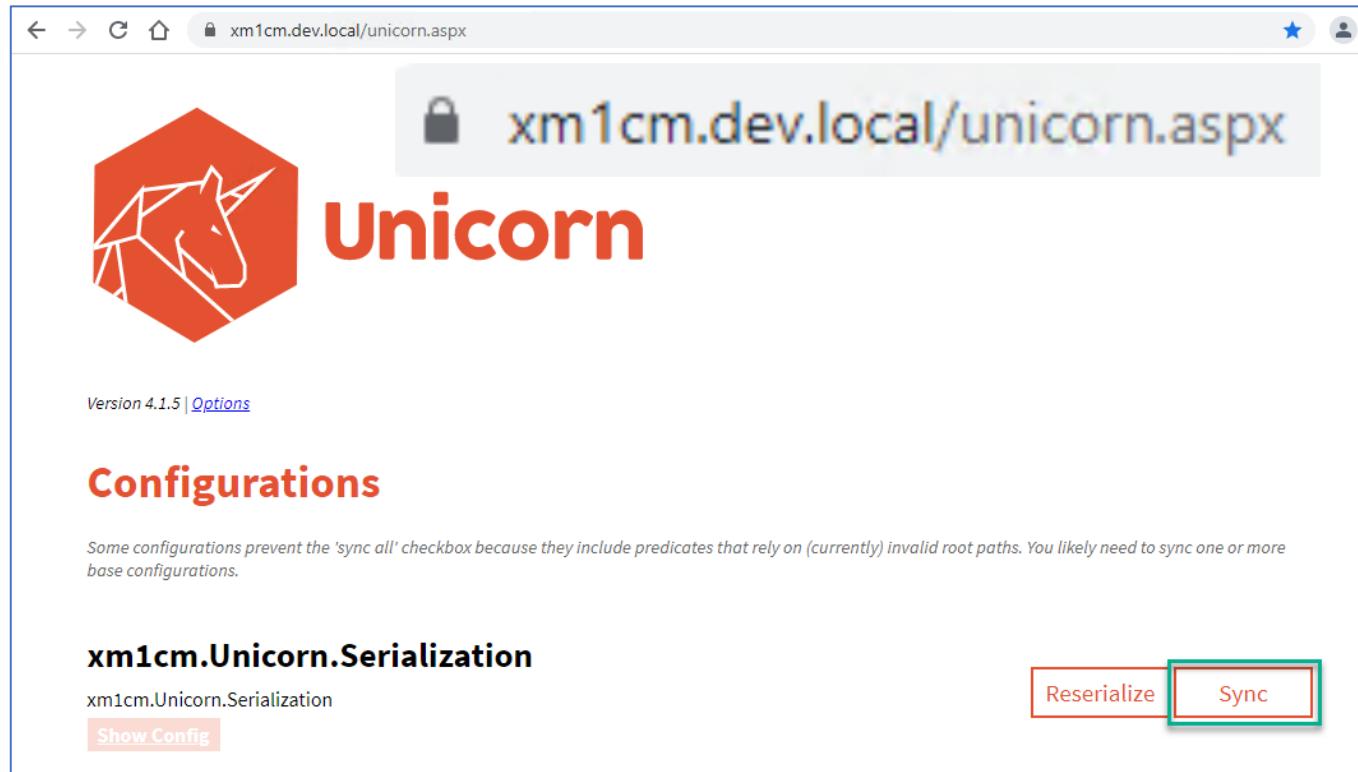
Serialization formatter: YAML Serialization Formatter

Physical root path: D:\Data\Unicorn\xm1cm.TargetDataStore
Total internal SFS trees: 4

11. Sitecore PowerShell Commands with Unicorn – Backup & Restore

3. Restore from Serialized folder to Target Site

Target site Unicorn



The screenshot shows the Unicorn interface running on the URL `xm1cm.dev.local/unicorn.aspx`. The page features a red hexagonal logo with a white unicorn head and the word "Unicorn" in large red letters. Below the logo, it says "Version 4.1.5 | [Options](#)". A section titled "Configurations" contains a note about sync checkboxes. At the bottom, there's a section for "xm1cm.Unicorn.Serialization" with a "Show Config" button, and two buttons: "Reserialize" (in red) and "Sync" (highlighted with a green box).

11. Sitecore PowerShell Commands with Unicorn – Backup & Restore

4. Restore

The screenshot shows the Sitecore navigation bar on the right and a 'Publish Site' dialog box in the center.

Publish Site Dialog:

- Operation completed.**
- Title:** Publish Site
- Subtitle:** Select the relevant publishing settings for your website
- Publishing Options:**
 - Incremental publish - publish only changed items.
 - Smart publish - publish differences between source and target database.
 - Republish - publish everything.
- Publishing language:**
 - Select all
 - English
- Publishing targets:**
 - Internet (web)
- Buttons:** Publish (blue) and Cancel

Console Output:

```
xm1cm.Unicorn.Serialization
[A] master:/sitecore/con
[A] master:/sitecore/con
[A] master:/sitecore/tem
[A] master:/sitecore/lay
[A] master:/sitecore/con
[A] master:/sitecore/con
[A] master:/sitecore/tem
xm1cm.Unicorn.Serialization sync complete: 1/20 items evaluated, 22 items modified (22 added)
[P] Auto-publishing of synced items is beginning.
> Publishing 22 synced items in queue to web
> Published synced items to web. Statistics is not retrievable when Publish Manager is used

Operation completed in 8658ms. Want to return to the control panel?
```

Navigation Bar:

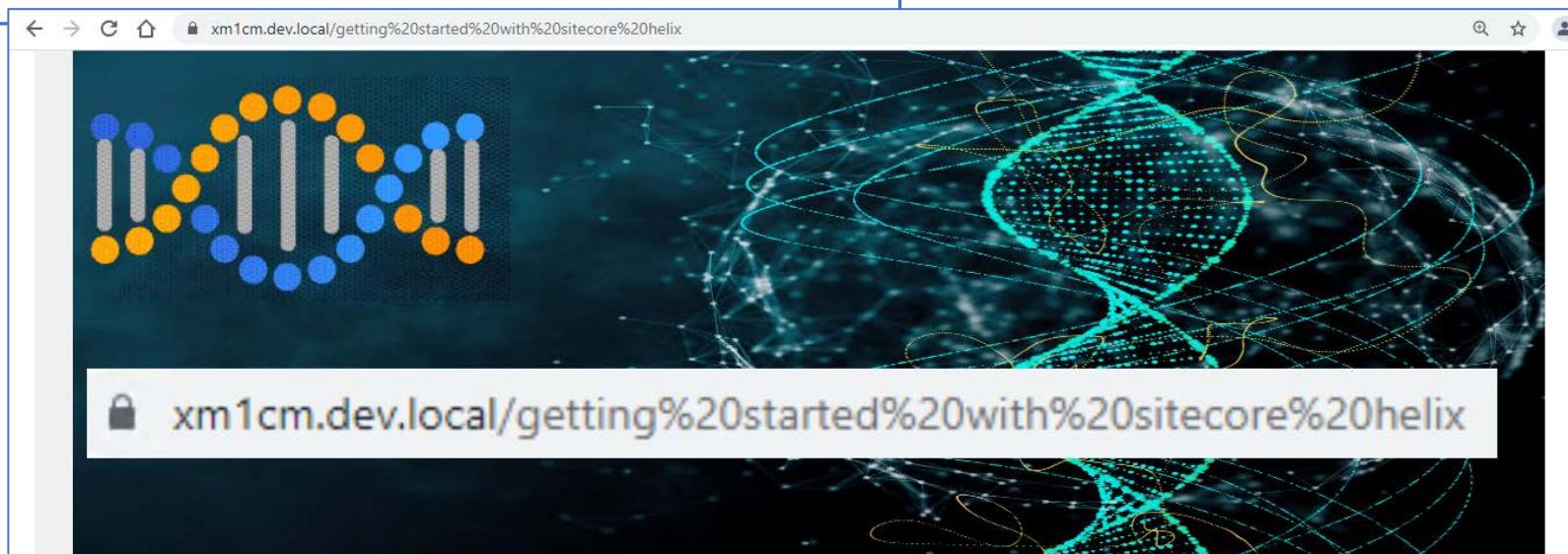
- Content Editor (Manage your content)
- Experience Editor (Edit your website in a...)
- Media Library (Maintain your media items)
- Preview (Preview your website)
- Template Manager (Create new templates)
- Publish Site** (Publish your website) - This item is highlighted with a red border.
- Debug (Open the Debugger)
- Recycle Bin (Manage deleted items)
- Control Panel
- PowerShell Console (Command line interface to...)
- PowerShell Toolbox (Custom tools built with...)
- Security Tools
- Development Tools
- Reporting Tools
- All Applications

11. Sitecore PowerShell Commands with Unicorn – Backup & Restore

4. Restore : wwwroot subfolder – Views folder from Source Site to Target site

inetpub > wwwroot > xm1cm.dev.local > Views >			
Name	Date modified	Type	S
FormBuilder	8/29/2021 11:16 AM	File folder	
Layouts	8/30/2021 2:24 PM	File folder	
PageContent	8/30/2021 2:24 PM	File folder	
Web.config	6/1/2021 12:08 PM	XML Configuration...	

Unicorn only works between sitecore database and backup folder (wwwroot folder also need to be updated)



Getting Started with Sitecore Helix - Editable

11. Sitecore PowerShell Commands with Unicorn – Backup & Restore

The screenshot shows the Sitecore Content Editor interface. On the left, the Content tree displays the structure of the site. The 'Home' item under 'Content' is selected. A yellow warning box is overlaid on the right side of the screen, containing the following information:

- This item is controlled by Unicorn
- Changes to this item will be written to disk so they can be shared with others.
- Configuration: xm1cm.Unicorn.Serialization
- Predicate Component: content
- Physical path: D:\Data\Unicorn\xm1cm.TargetDataStore\content\content\Home.yml

The main content area shows the 'Home' page's 'Data' section, which includes fields for 'Title' and 'Text'. The 'Text' field contains descriptive text about Sitecore's customer engagement capabilities. At the bottom of the page, there are links for 'Show editor', 'Suggest fix', and 'Edit HTML'.

12. Sitecore Migration with Unicorn 4.1.5, SPE 6.3 & Remoting

Location : <https://github.com/michaellwest/spe-content-migrator>

The screenshot shows a Windows PowerShell ISE window with a dark theme. The script being run is located at `PS C:\inetpub\wwwroot\XM1.cm.dev.local\SPE>`. The script itself is as follows:

```
Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
PS C:\inetpub\wwwroot\XM1.cm.dev.local\SPE>
PS C:\inetpub\wwwroot\XM1.cm.dev.local\SPE>
Clear-Host

$scriptDirectory = & {
    if ($psISE) {
        Split-Path -Path $psISE.CurrentFile.FullPath
    } else {
        $PSScriptRoot
    }
}
. "$($scriptDirectory)\Copy-RainbowContent.ps1"

$sourceSession = New-ScriptSession -user "Admin" -pass 'b' -conn "https://sourcesite.local"
$destinationSession = New-ScriptSession -user "Admin" -pass 'b' -conn "https://destinationsite.local"

$copyProps = @{
    "WhatIf"=$true
    "CopyBehavior"="CompareRevision"
    "Recurse"=$true
    "RemoveNotInSource"=$false
    "ClearAllCaches"=$true
    "LogLevel"="Detailed"
    "CheckDependencies"=$false
    "BoringMode"=$false
    "FailOnError"=$false
}

$copyProps["SourceSession"] = $sourceSession
$copyProps["DestinationSession"] = $destinationSession
# Default Home
Copy-RainbowContent @copyProps -RootId '{110D559F-DEA5-42EA-9C1C-8A5DF7E70E'
```

Annotations on the right side of the code highlight specific parts:

- A green callout points to the line `$PSScriptRoot` with the text "Can be replaced by Security String at <sharedSecret>".
- A red callout points to the URL `"https://sourcesite.local"` with the text "Source Sitecore Site URL".
- A red callout points to the URL `"https://destinationsite.local"` with the text "Target Sitecore Site URL".
- A green callout points to the line `"SourceSession" = $sourceSession` with the text "Source Site Root Address".

Below the PowerShell window, a GitHub repository interface for `michaellwest / Spe-Content-Migrator` is shown. It lists two files: `Copy-RainbowContent.ps1` and `MigrationSample.ps1`.

12. Sitecore Migration with Unicorn 4.1.5, SPE 6.3 & Remoting

Setup 2 Sitecore 10.1 Sites

xm1cm.Unicorn.Serialization Details

Predicate

Predicates define which items are included or excluded in Unicorn.

Serialization Preset Predicate

Defines what to include in Unicorn based on XML configuration entries.

content: master:/sitecore/content

media library: master:/sitecore/media library

templates: master:/sitecore/templates

layout: master:/sitecore/layout

Target Data Store

Defines how items are serialized, for example to disk using YAML format.

Serialization File System Data Store

Stores serialized items on disk using the SFS tree format, where each root is a separate tree.

Serialization formatter: YAML Serialization Formatter

Physical root path: D:\Data\Unicorn\xm1cm.TargetDataStore

Total internal SFS trees: 4

xm2.cm.Unicorn.Serialization Details

Predicate

Predicates define which items are included or excluded in Unicorn.

Serialization Preset Predicate

Defines what to include in Unicorn based on XML configuration entries.

content: master:/sitecore/content

media library: master:/sitecore/media library

templates: master:/sitecore/templates

layout: master:/sitecore/layout

Target Data Store

Defines how items are serialized, for example to disk using YAML format.

Serialization File System Data Store

Stores serialized items on disk using the SFS tree format, where each root is a separate tree.

Serialization formatter: YAML Serialization Formatter

Physical root path: D:\Data\Unicorn\xm2.cm.TargetDataStore

Total internal SFS trees: 4

12. Sitecore Migration with Unicorn 4.1.5, SPE 6.3 & Remoting

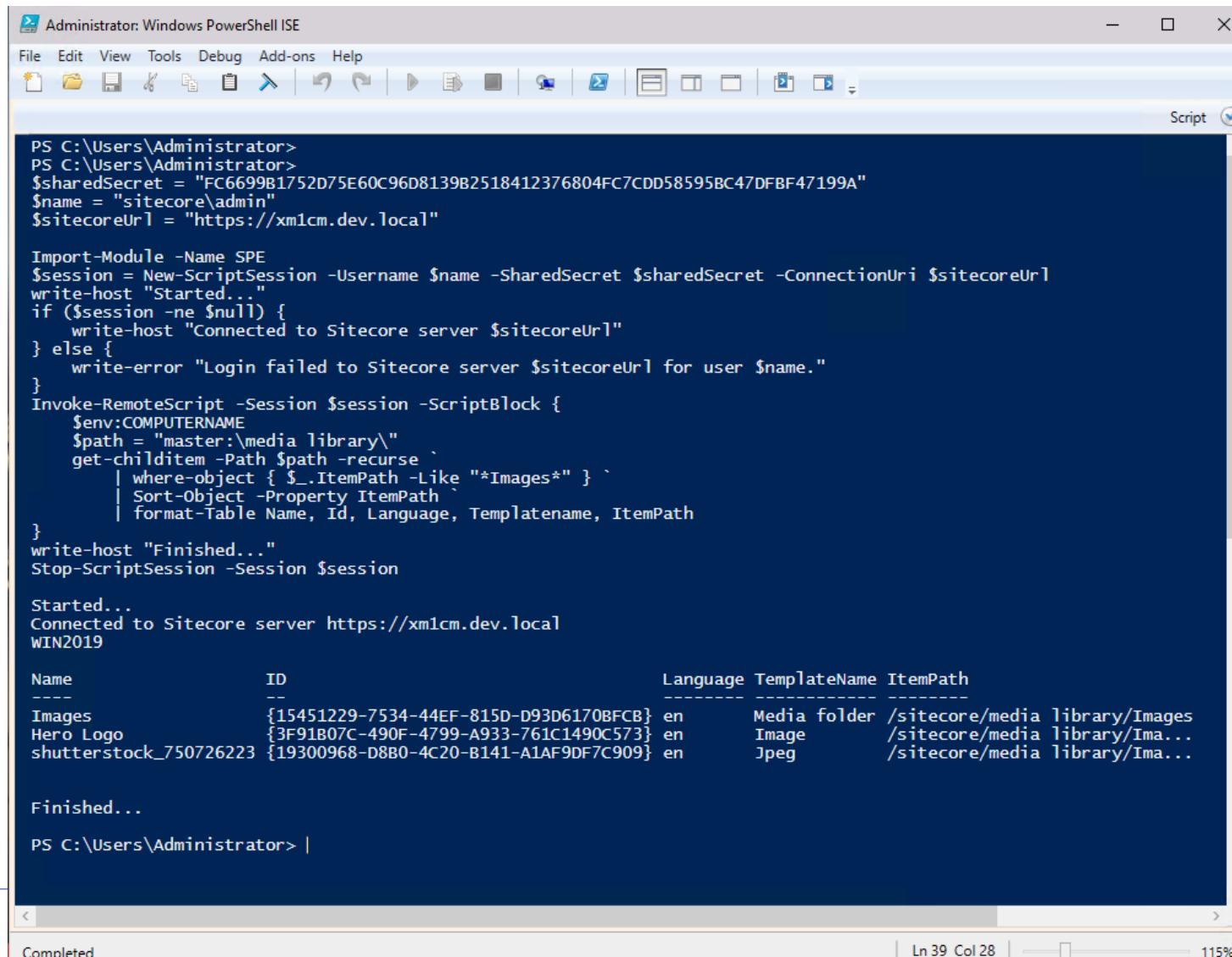
Setup 2 Sitecore 10.1 Sites – Target Sitecore has default setting

The screenshot displays two separate Sitecore 10.1 environments side-by-side, each with its own URL in the browser address bar:

- Left Shell (xm1cm.dev.local):** Shows the Sitecore navigation tree on the left and a content editor on the right. A specific item under 'Content' is highlighted with a yellow warning box containing the text: "This item is controlled by a predicate component". Below this, it says "Configuration: xm1cm.dev.local", "Predicate Component: /sitecore/system/predicates/allow-new-items", and "Physical path: D:\Data\Website\Content\Home".
- Right Shell (xm2.cm.dev.local):** Shows a similar Sitecore interface. Its navigation tree includes 'sitecore', 'Content', 'Forms', 'Layout', 'Media Library', 'Images', 'System', and 'Templates'. The 'Content' section on the right includes buttons for 'Upload File' and 'Media folder'.

12. Sitecore Migration with Unicorn 4.1.5, SPE 6.3 & Remoting

Setup 2 Sitecore 10.1 Sites – SPE Remoting – Source Site



The screenshot shows a Windows PowerShell ISE window with the following content:

```
Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
PS C:\Users\Administrator>
PS C:\Users\Administrator>
$sharedSecret = "FC6699B1752D75E60C96D8139B2518412376804FC7CDD58595BC47DFBF47199A"
$name = "sitecore\admin"
$sitecoreUrl = "https://xm1cm.dev.local"

Import-Module -Name SPE
$session = New-ScriptSession -Username $name -SharedSecret $sharedSecret -ConnectionUri $sitecoreUrl
write-host "Started..."
if ($session -ne $null) {
    write-host "Connected to Sitecore server $sitecoreUrl"
} else {
    write-error "Login failed to Sitecore server $sitecoreUrl for user $name."
}
Invoke-RemoteScript -Session $session -ScriptBlock {
    $env:COMPUTERNAME
    $path = "master:\media library\"`n
    get-childitem -Path $path -recurse`n
        | where-object { $_.ItemPath -Like "*Images*" }`n
        | Sort-Object -Property ItemPath`n
        | format-Table Name, Id, Language, Templatename, ItemPath
}
write-host "Finished..."
Stop-ScriptSession -Session $session

Started...
Connected to Sitecore server https://xm1cm.dev.local
WIN2019

Name           ID          Language  TemplateName  ItemPath
----           --          -----      -----       -----
Images         {15451229-7534-44EF-815D-D93D6170BFCB}  en      Media   folder   /sitecore/media library/Images
Hero Logo      {3F91B07C-490F-4799-A933-761C1490C573}  en      Image    /sitecore/media library/IMA...
shutterstock_750726223 {19300968-D8B0-4C20-B141-A1AF9DF7C909}  en      Jpeg     /sitecore/media library/IMA...
```

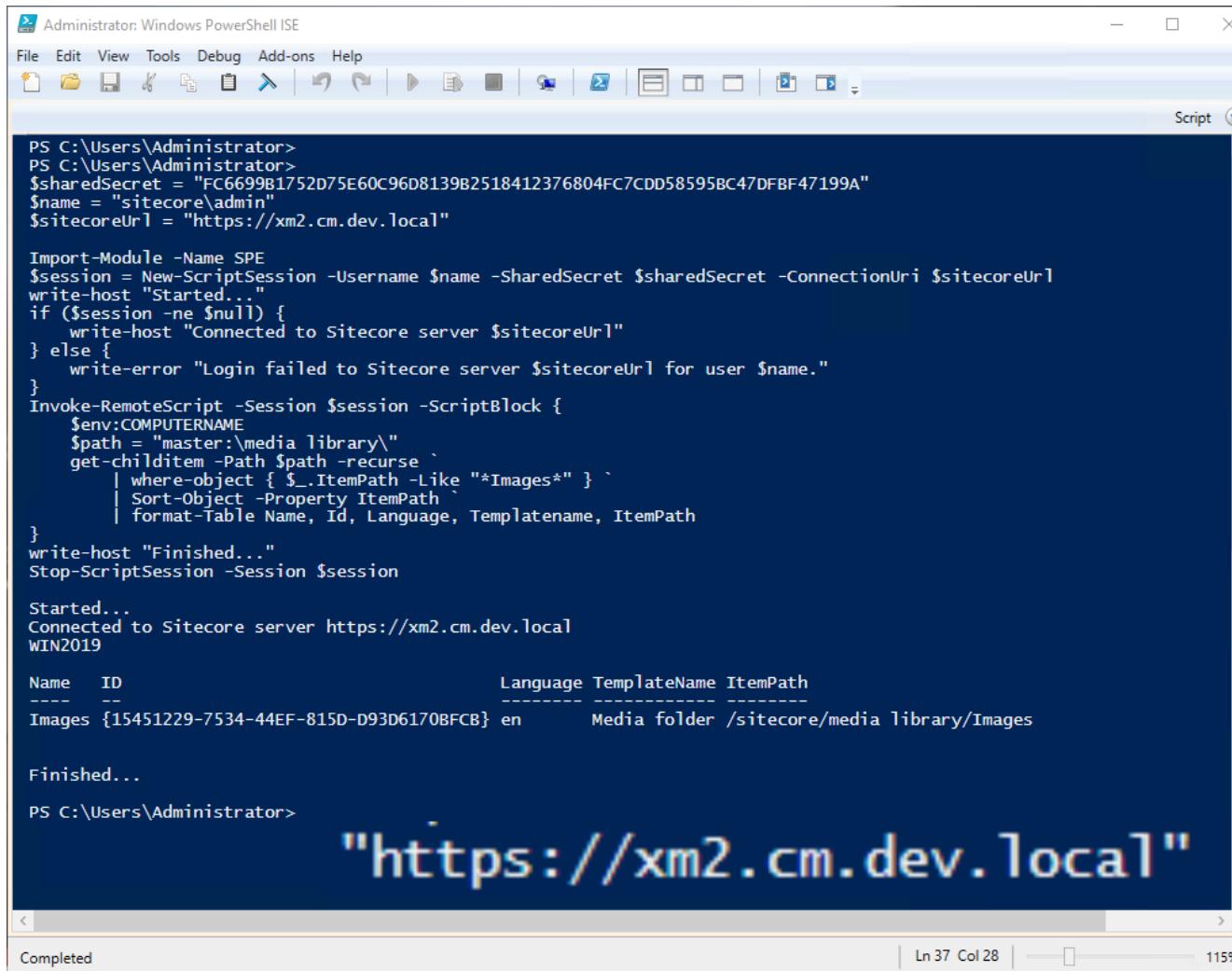
The script connects to a Sitecore instance at `https://xm1cm.dev.local` using the `sitecore\admin` account. It then runs a remote PowerShell session to execute a command that retrieves all items under the `master:\media library\` path where the item path contains `*Images*`. The results are displayed in a table:

Name	ID	Language	TemplateName	ItemPath
Images	{15451229-7534-44EF-815D-D93D6170BFCB}	en	Media	/sitecore/media library/Images
Hero Logo	{3F91B07C-490F-4799-A933-761C1490C573}	en	Image	/sitecore/media library/IMA...
shutterstock_750726223	{19300968-D8B0-4C20-B141-A1AF9DF7C909}	en	Jpeg	/sitecore/media library/IMA...

The process is completed successfully.

12. Sitecore Migration with Unicorn 4.1.5, SPE 6.3 & Remoting

Setup 2 Sitecore 10.1 Sites – SPE Remoting – Target Site



The screenshot shows a Windows PowerShell ISE window with the following content:

```
PS C:\Users\Administrator>
PS C:\Users\Administrator>
$sharedSecret = "FC6699B1752D75E60C96D8139B2518412376804FC7CDD58595BC47DFBF47199A"
$name = "sitecore\admin"
$sitecoreUrl = "https://xm2.cm.dev.local"

Import-Module -Name SPE
$session = New-ScriptSession -Username $name -SharedSecret $sharedSecret -ConnectionUri $sitecoreUrl
write-host "Started..."
if ($session -ne $null) {
    write-host "Connected to Sitecore server $sitecoreUrl"
} else {
    write-error "Login failed to Sitecore server $sitecoreUrl for user $name."
}
Invoke-RemoteScript -Session $session -ScriptBlock {
    $env:COMPUTERNAME
    $path = "master:\media library\"`n
    get-childitem -Path $path -recurse`n
    | where-object { $_.ItemPath -Like "*Images*" }`n
    | Sort-Object -Property ItemPath`n
    | format-Table Name, Id, Language, Templatename, ItemPath
}
write-host "Finished..."
Stop-ScriptSession -Session $session

Started...
Connected to Sitecore server https://xm2.cm.dev.local
WIN2019

Name      ID          Language  TemplateName  ItemPath
----      --          -----      -----        -----
Images  {15451229-7534-44EF-815D-D93D6170BFCB}  en       Media folder  /sitecore/media library/Images

Finished...

PS C:\Users\Administrator>
```

A large, stylized watermark or background text "https://xm2.cm.dev.local" is overlaid on the bottom half of the terminal window.

12. Sitecore Migration with Unicorn 4.1.5, SPE 6.3 & Remoting

Setup SPE Migration Script

```
1  Clear-Host  
2  
3  Import-Module -Name SPE -Force  
4  
5  $scriptDirectory = & {  
6      if ($psISE) {  
7          Split-Path -Path $psISE.CurrentFile  
8      } else {  
9          $PSScriptRoot  
10     }  
11 }  
12 . "$($scriptDirectory)\Copy-RainbowContent.ps1"  
13  
14 $sourceSession = New-ScriptSession -user "iisexpress"  
15 $destinationSession = New-ScriptSession -u  
16  
17 $copyProps = @{  
18     "WhatIf"=$true  
19     "CopyBehavior"="CompareRevision"  
20     "Recurse"=$true  
21     "RemoveNotInSource"=$false  
22     "ClearAllCaches"=$true  
23     "LogLevel"="Normal"  
24     "CheckDependencies"=$false  
25     "BoringMode"=$false  
26     "FailOnError"=$false  
27 }  
28  
29 $copyProps["SourceSession"] = $sourceSession  
30 $copyProps["DestinationSession"] = $destinationSession  
31 # Default Home  
32 Copy-RainbowContent @copyProps -RootId "{0DE95AE4-41AB-4D01-9EB0-67441B7C2450}" * >&1 | Tee-Object "$($scriptDirectory)\Migration-DefaultHome.log"
```

Content - [content]
Web site content.

You cannot edit this item because it is protected.
To unprotect the item, click Unprotect Item on the Configure tab or click Unprotect Item.

[Unprotect Item](#)

Quick Info

Item ID:	{0DE95AE4-41AB-4D01-9EB0-67441B7C2450}
Item name:	content - Display Name: Content
Item path:	/sitecore/content
Template:	/sitecore/templates/System/Main section - {E3E2D58C-DF95-4230-ADC9-279924CECE84}
Created from:	[unknown]
Item owner:	[unknown]

12. Sitecore Migration with Unicorn 4.1.5, SPE 6.3 & Remoting

The screenshot shows the Sitecore Content Editor with a yellow warning banner stating: "This item is controlled by Unicorn. Changes to this item will be written to disk so they can be shared with other instances." Below the banner, it shows configuration details: "Configuration: xm1cm.Unicorn.Serialization", "Predicate Component: content", and "Physical path: D:\Data\Unicorn\xm1cm.TargetDataStore\".

The left pane displays the Sitecore navigation tree, starting with "sitecore" > "Content" > "Home". The "Home" item has a yellow star icon next to it. The "Images" folder under "Media Library" contains a file named "Hero Logo" with the physical path "D:\Data\Unicorn\xm1cm.TargetDataStore\Images\Hero Logo\shutterstock_750726223".

The right pane shows a PowerShell window with the following log output:

```
[17:21:35:379] [Options] RootId = {110D559F-DEA5-42EA-9C1C-8A5DF7E70EF9}; Recurse = True;
[17:21:35:381] [Options] CopyBehavior = Overwrite; RemoveNotInSource = False;
[17:21:35:383] - Querying item list from source
[17:21:35:750] - Found 2 item(s) in 0.359 seconds
[17:21:36:363] Spinning up jobs to transfer content
[17:21:36:428] [Status] 100% complete (2)
[17:21:36:626] - Clearing all caches in the destination
[17:21:36:633] [Done] Completed in a record 1.327 seconds! 🎉
[17:21:36:643] - Transferred 0 MB of item data
[17:21:36:646] Processed 2
[17:21:36:647] ✓ Updated 2
[17:21:36:649] ✓ Skipped 0
[17:21:36:650] ✓ Errored 0
[17:21:36:652] ✓ Pulled 0
[17:21:36:655] ✓ Pushed 0
```

At the bottom of the PowerShell window, the command PS D:\Sitecore 10.1 Install\SPE\PS1> is visible.

12. Sitecore Migration with Unicorn 4.1.5, SPE 6.3 & Remoting

Future Use

Appendix

Reference:

<https://github.com/michaellwest/spe-content-migrator>

<https://youtu.be/QanxZI14xE>

<https://youtu.be/A3vcVivdSf8>

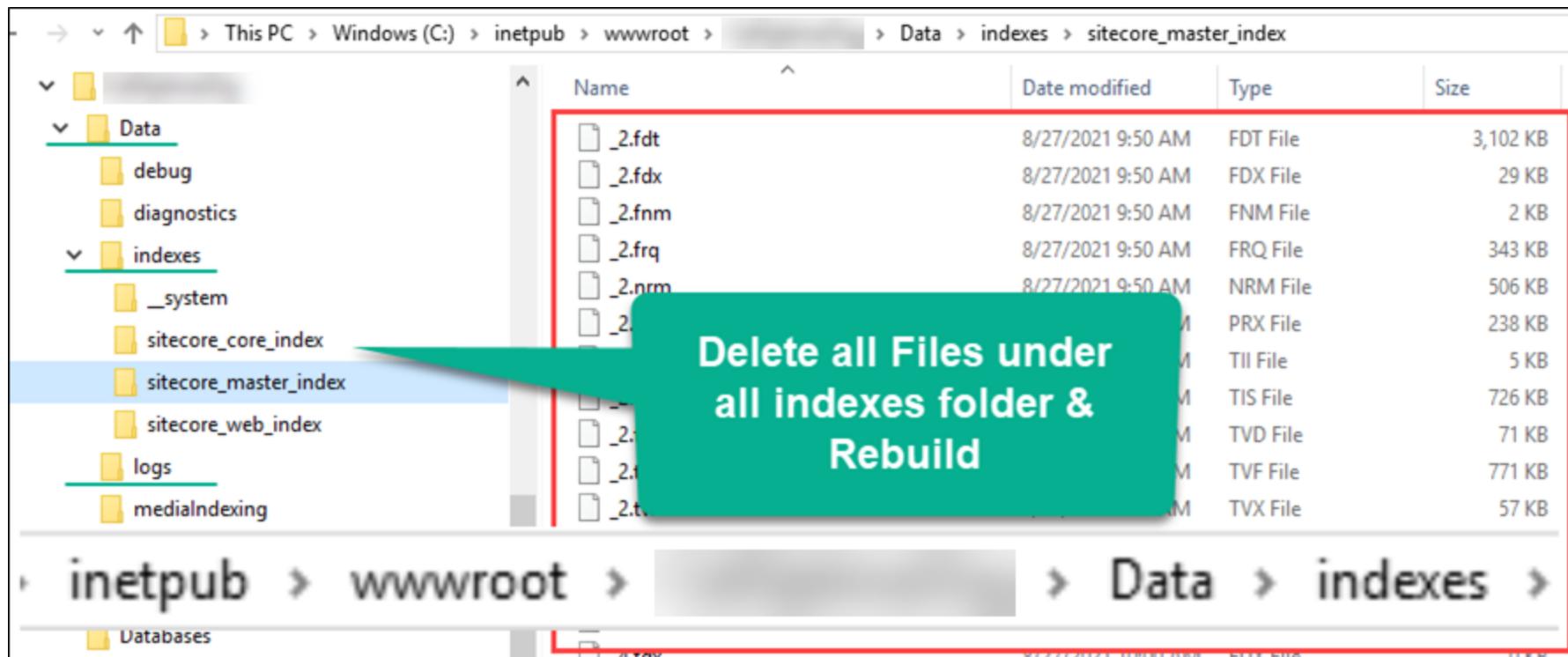
<https://youtu.be/BYvCcmmmWvo>

<https://youtu.be/5Ne5ydd-9rM>

<https://youtu.be/z3anJtoxR8E>

Appendix

- **Rebuild Indexes on Sitecore 7.2**



Appendix

- **Rebuild Indexes on Sitecore 7.2**

