



PowerShell ModuleManager - Help Manual

© 2020 by SAPIEN Technologies Inc., all rights reserved

1. Welcome to PowerShell ModuleManager	3
2. Introduction	4
About PowerShell ModuleManager	4
How to Buy PowerShell ModuleManager	4
3. Getting Started	6
Installing PowerShell ModuleManager	6
The User Interface	12
Local Machine and Main Window	15
Toolbars	20
Panels - Overview	22
Skins, Options and Shortcuts	24
Staying Up-to-date	28
Getting Help	28
4. Managing Modules	31
Searching for Modules	31
Installing Modules	35
Managing Installed Modules	45
Finding Installed Modules	45
Viewing Module Details	51
Updating Modules	53
Moving Modules	59
Publishing Modules	60
Disabling and Uninstalling Modules	65
Managing Repositories	84
Managing Remote Connections	92
5. Reference	98
Program Options	98
Program Options - General	99
Program Options - Panels	102
Workspace, Ribbons and Dialogs	103
The Ribbon	106
The File Tab	107
The Home Tab	108
The Tools Tab	113
The Help Tab	115

- Other Toolbars 116
- Machine Tab and Main Window 120
- Panels 127
 - Disabled Modules Panel 129
 - Information Panel 133
 - Log Panel 135
 - Remote Connections Panel 137
- Dialogs 142
- Keyboard Shortcuts 152
- Customizing Your Workspace 154**
 - Changing the color theme 154
 - Customizing the Quick Access Toolbar 155
 - Panels and Layout 159
- SAPIEN Updates 169**
- Appendices 173**
 - Appendix A: Manual and Product Version 174
 - Appendix B: Icon License Attribution 174
- 6. Frequently Asked Questions 175**

1 Welcome to PowerShell ModuleManager

Managing PowerShell modules has never been easier!



PowerShell modules are an essential component of script development. **PowerShell ModuleManager** makes it easy to search and manage modules installed locally or on a remote machine, ensuring that your modules are always up-to-date.

We are always adding and improving PowerShell ModuleManager. For a complete list of current features, visit the [PowerShell ModuleManager product page](#).

About this documentation

This help is designed to show you how to use PowerShell ModuleManager and also as an ongoing reference while you are working with the tool. You can do a quick overview to get started, work through the chapters in detail, and refer back to this guide for additional information when needed.

 This help manual covers the features and functions included in PowerShell ModuleManager version 1.1.3 and earlier.

Getting started - new users

- [Download and install PowerShell ModuleManager](#)^[6].
- Check out the [User Interface](#)^[12] section to familiarize yourself with the basics of the program.
- Review the [Managing Modules](#)^[31] section to see what features are available and how to use them.
- Visit the [support forum](#) to get help from SAPIEN staff and other experienced PowerShell ModuleManager users.

2 Introduction

This section provides an overview of the PowerShell ModuleManager features, lets you know how to get answers to your questions, and shows you how to purchase directly online or through a reseller.

2.1 About PowerShell ModuleManager

PowerShell ModuleManager is a professional tool for managing your PowerShell modules.

Key Features

- **Search** the PowerShell Gallery or any other repository for new modules.
- **Install, remove, or temporarily disable** modules.
- Easily keep modules **up-to-date**.
- **Remove duplicate and outdated module versions** to prevent version confusion.
- **Filter and find** specific modules on your machine.
- Manage modules on **remote machines**.
- **Publish** modules to registered repositories.

What's New

We are always updating and improving PowerShell ModuleManager. You can learn about the latest product updates on our blog and in the release build log.

- Check out the PowerShell ModuleManager articles on the [SAPIEN blog](#).
- View the PowerShell ModuleManager [version history](#).
- Submit [feedback and suggestions](#).

2.2 How to Buy PowerShell ModuleManager

You can buy PowerShell ModuleManager online with all major credit cards. As soon as your transaction completes, you will be able to [download and install](#) ⁶ the program.

For answers to your pre-order questions, check out the [SAPIEN Frequently Asked Questions](#) or post in the [Trial Software / Pre-sales Technical Questions](#) forum.

Order link and PowerShell ModuleManager product page

Online orders:

<https://www.sapien.com/store/powershell-modulemanager>

Worldwide authorized resellers:

<https://www.sapien.com/company/resellers>

PowerShell ModuleManager product page:

https://www.sapien.com/software/powershell_modulemanager

3 Getting Started

This section provides information to help you install PowerShell ModuleManager and keep it up-to-date, and also to understand the general layout of the user interface and know how to get help.


3.1 Installing PowerShell ModuleManager

To get started using PowerShell ModuleManager, follow the instructions below to download and install the program.

Downloading PowerShell ModuleManager

All SAPIEN Technologies software products are downloadable only. Download registered products from your [SAPIEN Account Registered Products page](#).

Select the 64-bit version of PowerShell ModuleManager to download. The installer software will save to your default download folder (e.g., *PMG19Setup_1.1.2_060519_x64*).

 Starting with the PowerShell ModuleManager 2020 product release, 32-bit versions are no longer available. Current owners of a license that includes a 32-bit product will have access to that from their [SAPIEN Account Registered Products page](#).

Want to try before you buy? You can [download a trial version here](#).

Prerequisites

PowerShell Requirements

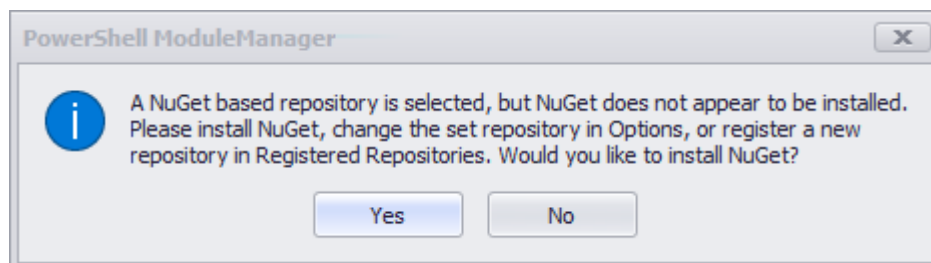
PowerShell v5.0 is required and is, by default, installed with Windows 10. Depending on demand, in the future, PowerShell ModuleManager may support older PowerShell versions.

Required PowerShell Modules

The PowerShellGet and PackageManagement modules are required. These modules are, by default, installed with Windows 10; we recommend installing the latest versions.

PowerShell Gallery Requirements

To install modules from the PowerShell Gallery, [NuGet](#) must be installed via PowerShell. If NuGet is not already installed, PowerShell ModuleManager will prompt to install NuGet for you:



Internet Requirements

An internet connection is required if the PowerShell repository designated in Options is an online

source (Options > General > PowerShell > Repository).

Security and Permissions

PowerShell ModuleManager needs to run with elevated permissions to update modules stored in the Program Files directory.

The following prompt appears when the program is installed or launched: *"Do you want to allow this app to make changes to your device?"*

Select **Yes**.

Installing PowerShell ModuleManager

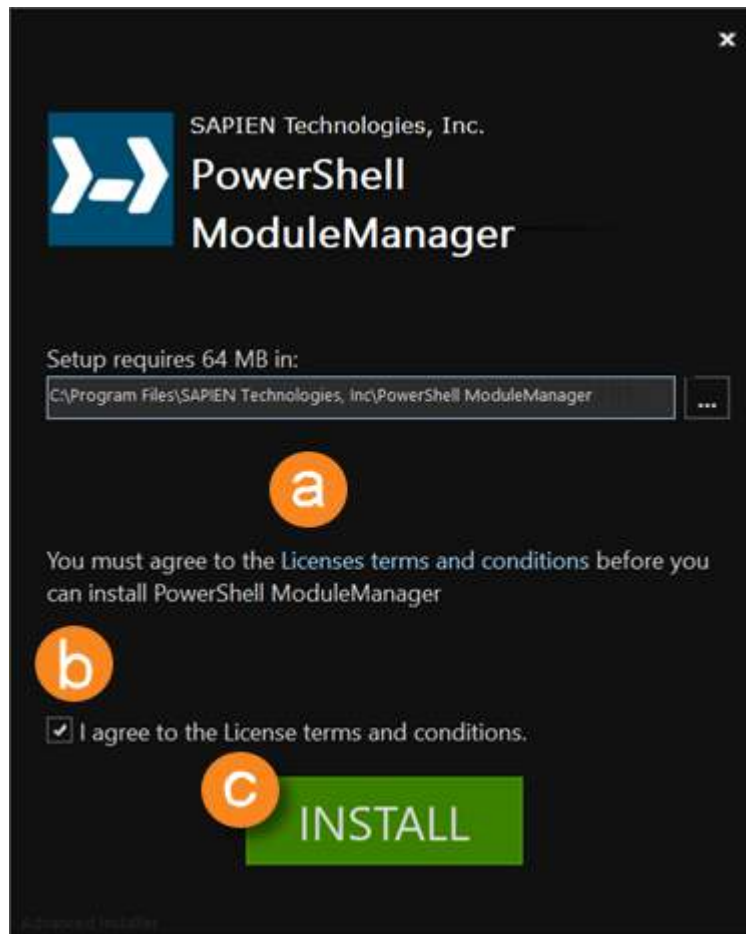
Follow these instructions to install PowerShell ModuleManager.

How to install PowerShell ModuleManager

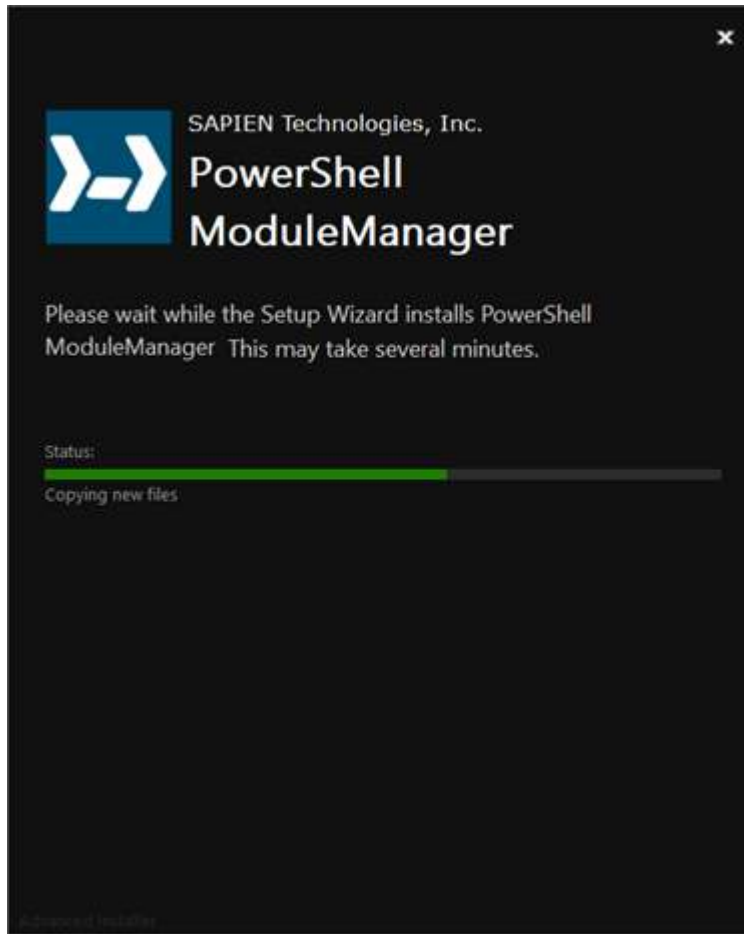
1. In your default download folder, double-click on the downloaded program (e.g., *PMG19Setup_1.1.2_060519_x64*).
2. Reply **Yes** to the *"Do you want to allow this app to make changes to your device?"* prompt.

The installation wizard will first check several items, such as available disk space and the presence of previous builds. If the environment is adequate, the installer will display the legal agreement which you must accept to proceed:

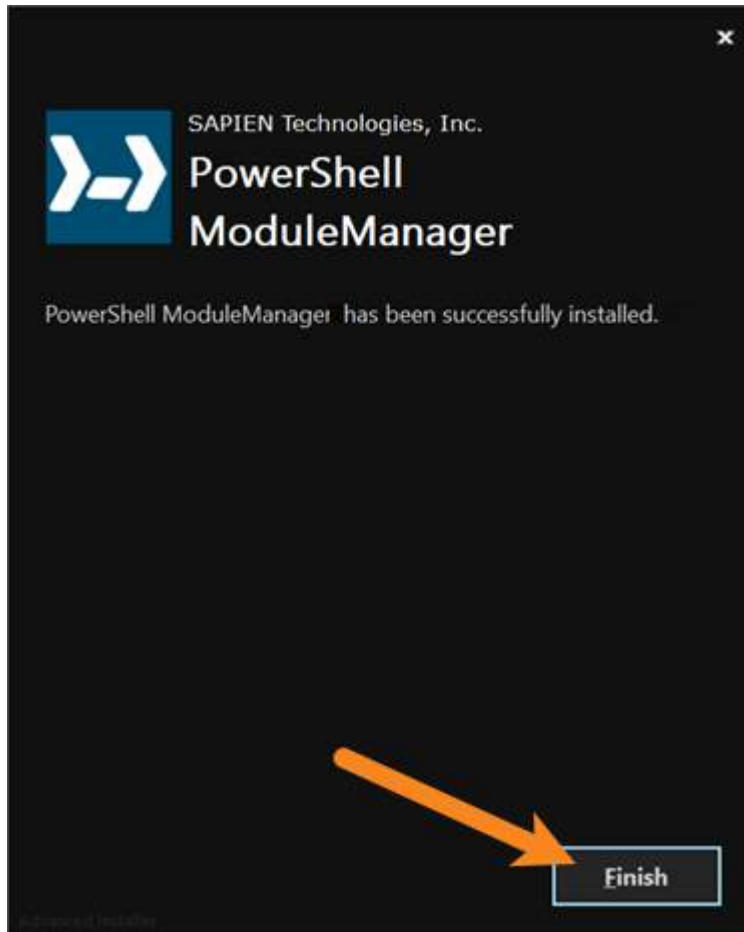
- a. **Read** the terms of the license agreement.
- b. **Accept** the terms of the license agreement. You should never accept license terms unless you have read them, and you understand them.
- c. Once you have accepted the terms, click **Install**.



3. The installation may take several minutes.



4. When PowerShell ModuleManager successfully completes the installation, click **Finish**.



Troubleshooting Installation

If you encounter problems installing PowerShell ModuleManager, please report them in the [Installation Issues support forum](#).

To output to a log file use these Installer Log parameters: `Installer.exe /xenoui /qn /L*v .\PMG_Install.log`

Activating and Deactivating PowerShell ModuleManager

Product Activation

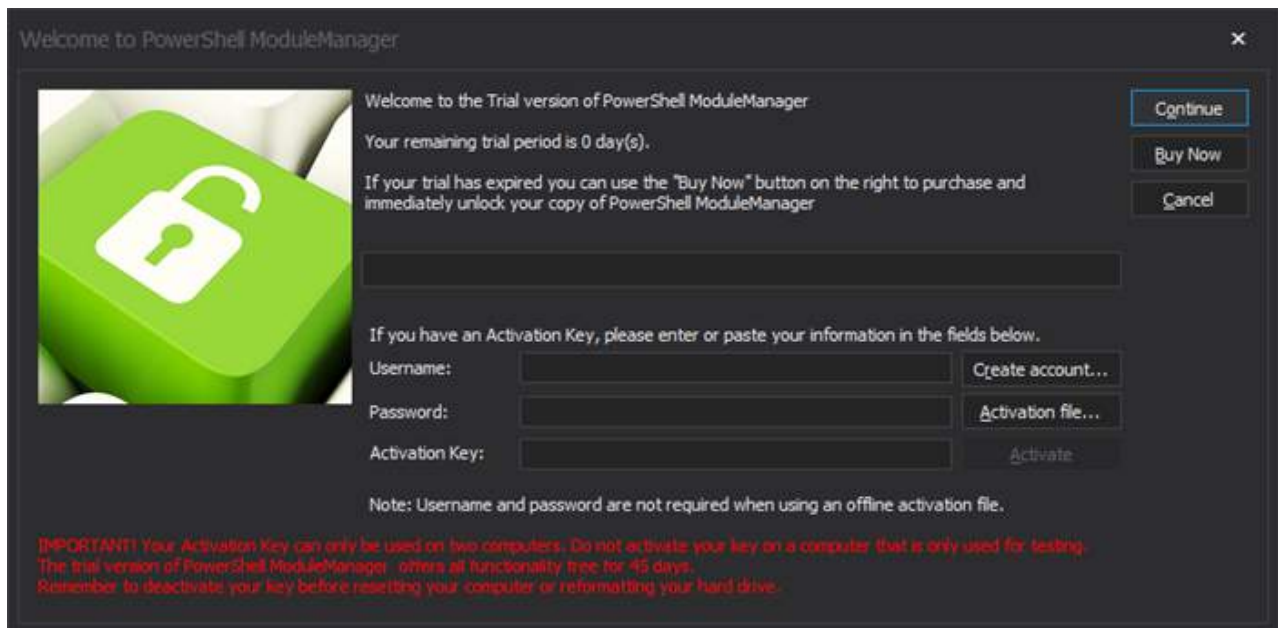
Registration is required to activate and operate the product, and also to obtain any customer service or technical support benefits. Registration only takes a few moments to complete and provides you with access to special offers including preferred pricing on renewals. *You will need an active internet*

connection to complete product registration.

An active internet connection may not be required if you have a legitimate reason for needing [offline access](#). To request offline activation [please fill out this request](#). All requests are considered on a case-by-case basis.

To activate PowerShell ModuleManager

The first time you launch a SAPIEN product, the Welcome screen is displayed.



The steps to activate the product vary depending on whether or not you already have a SAPIEN account.

👍 Follow the steps in the [Quick Guide to SAPIEN Software Activation](#) to activate the software.

If you are unable to activate the product, contact sales@sapien.com.

Product Deactivation

As outlined in our [End-User License Agreement](#), each single-user activation key is entitled to a maximum of two devices to be activated and operating at any given time. You may deactivate your devices to free up your activations at your own leisure, but there are also certain circumstances where proper deactivation is crucial in order to prevent the loss of your allotted two activations.

❗ Uninstalling the software from your device does **not** deactivate the activation key.

To deactivate your activation key

In the top-right of PowerShell ModuleManager above the ribbon, click the gold certificate button.



The Activation Information window will open.

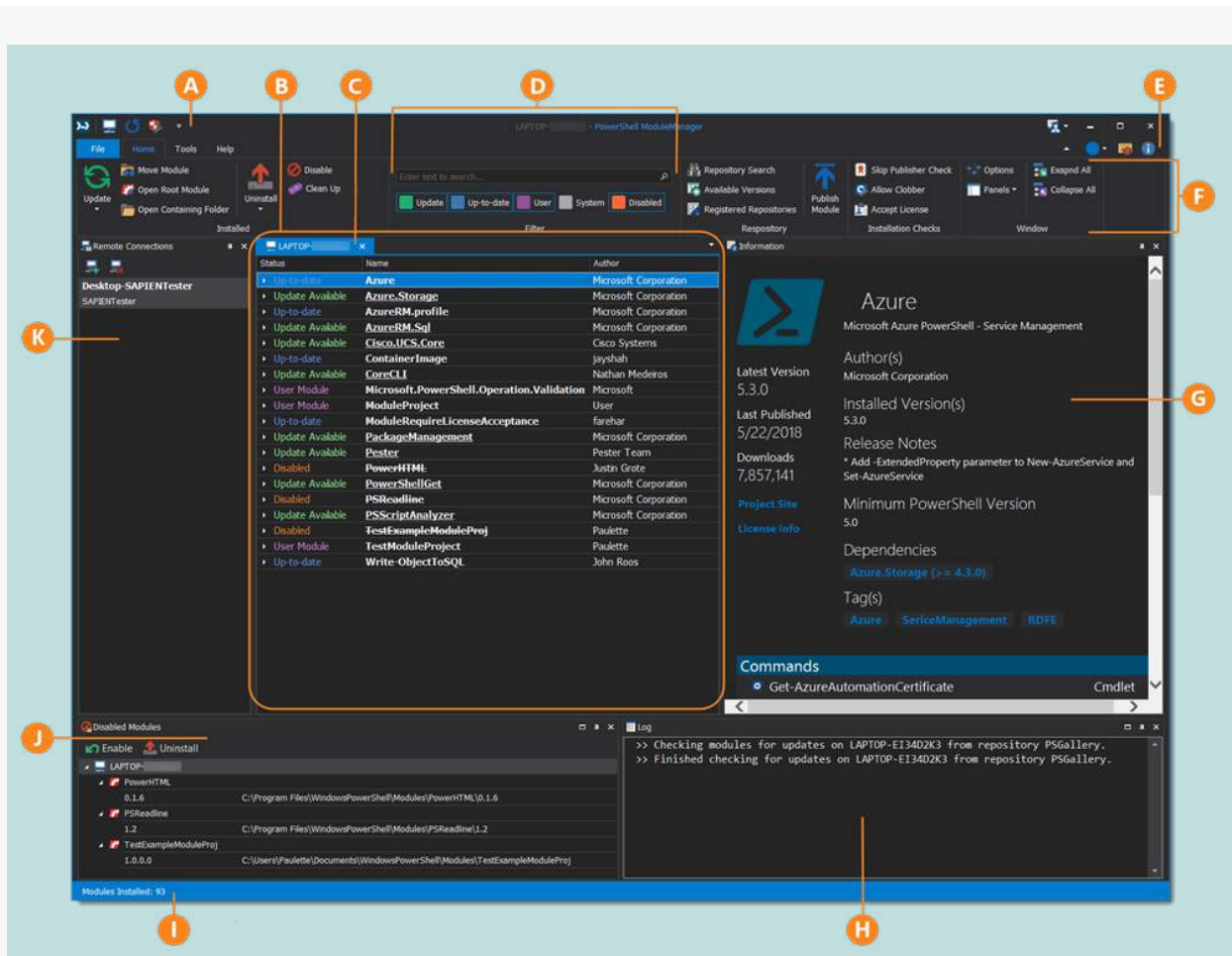
👍 Follow the steps in the [SAPIEN Software Activation / Deactivation FAQ](#) to deactivate your activation key.

3.2 The User Interface

This section provides a basic overview of the PowerShell ModuleManager user interface. Details for all PowerShell ModuleManager workspace elements and features can be found in the [Reference](#)^[98] section.

The User Interface

The PowerShell ModuleManager workspace lets you view, search, and manage modules using various panels, windows, and dialogs. The workspace places many of the most common operations in easily accessible buttons and toolbars to simplify your module management.



PowerShell ModuleManager user interface

A. Quick Access Toolbar **B.** Main Window **C.** Local Machine **D.** Search and Filter **E.** Ribbon Header Buttons **F.** Ribbon Toolbar **G.** Information Panel **H.** Log Panel **I.** Status Bar **J.** Disabled Modules Panel **K.** Remote Connections Panel

Quick Access Toolbar

The [Quick Access Toolbar](#)^[117] on the top-left of the program window provides easy access to frequently used functions, and you can easily [add your own controls](#)^[155] to this toolbar.

Main Window

The [main window](#)^[16] is located in the center of the PowerShell ModuleManager workspace and is a central area for managing modules installed on the [Local Machine](#)^[15] and any [Remote Connections](#)^[92]. When the [Program Options](#)^[98] are accessed, they will also appear in the main window.

Local Machine

The [Local Machine](#)^[15] tab is located in the main window and is used to manage locally installed modules.

Search and Filter

The Filter section on the Home ribbon allows you to [search and filter](#)^[31] installed modules.

Ribbon Header Buttons

The [Ribbon Header Buttons](#)^[119] on the top-right of the ribbon provide options to minimize the ribbon, change color-themed skins, view license information, and display product information.

Ribbon Toolbar

The Ribbon Toolbar, or [Ribbon](#)^[106], is organized in tabs, with each tab containing tools grouped logically by function. Some tools also provide access to dialogs with additional options.

Information Panel

The [Information panel](#)^[133] displays details for installed modules

Log Panel

The [Log panel](#)^[135] displays PowerShell ModuleManager output messages.

Status Bar

The [Status Bar](#)^[120] displays the current number of installed modules.

Disabled Modules Panel

The [Disabled Modules panel](#)^[129] displays all disabled modules, where they can easily be enabled or uninstalled.

Remote Connections Panel

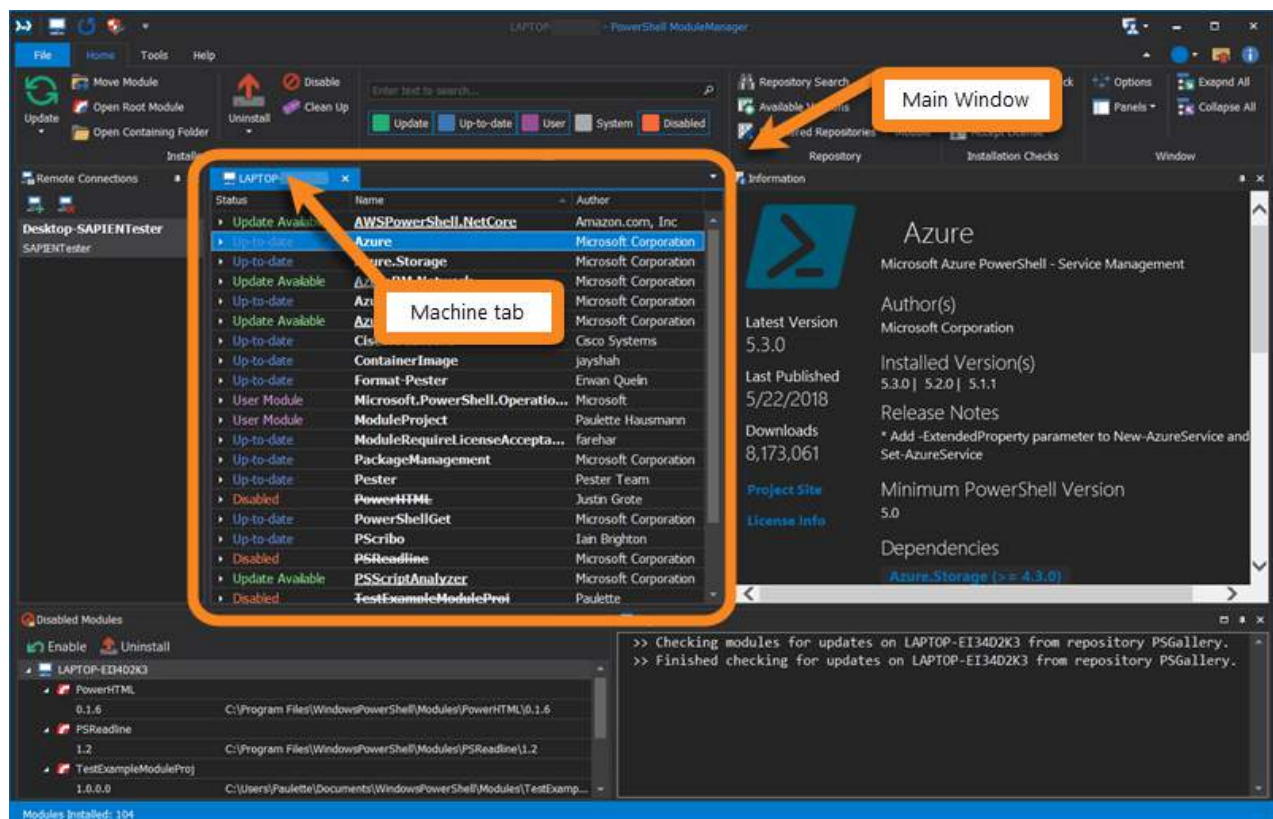
Use the [Remote Connections panel](#)^[137] to add, manage, and remove remote connections. When a remote connection is added and opened, it will appear as a tab in the [main window](#)^[16].

3.2.1 Local Machine and Main Window

Installed modules are managed in the [Machine tab](#)^[120], which is centrally located in the [main window](#)^[16] of the PowerShell ModuleManager workspace. Both local and remote machines will appear as tabs with the name of the machine.

Local Machine

The modules installed on your **local machine** can be accessed from the machine tab in the [main window](#)^[16] of the PowerShell ModuleManager workspace:



Machine tab (Local Machine) in the Main Window

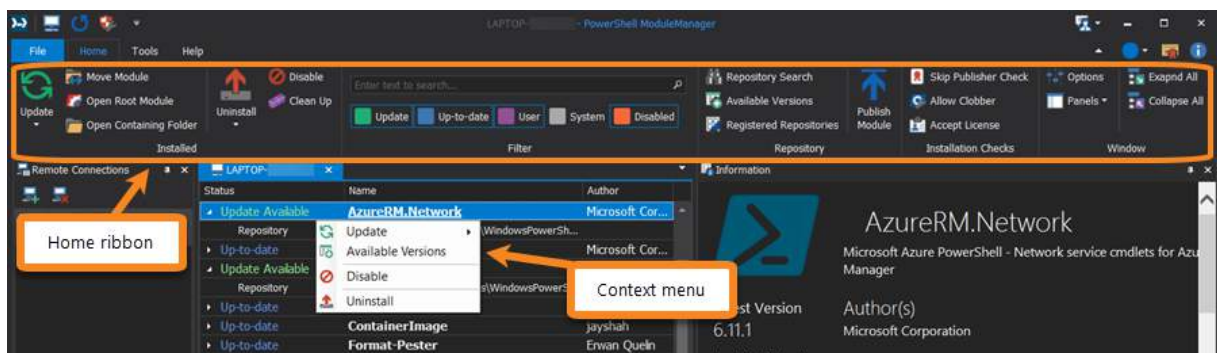
How to access module options

To access options for installed modules

1. Highlight a module in the machine tab
2. Select the desired control on the Home ribbon

-OR-

Right-click and select an option in the context menu



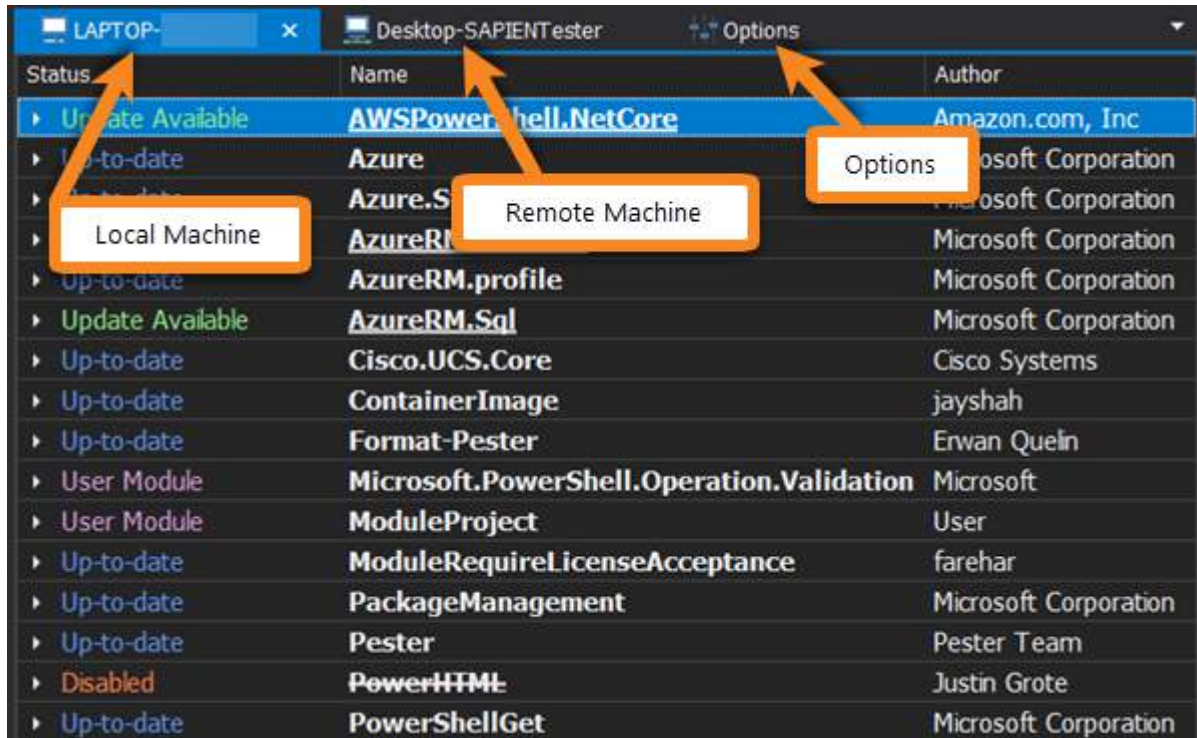
i The options available on the Home ribbon and the context menu will vary, depending on whether the tab selected is a local machine or a remote connection.

See [Finding Installed Modules](#)^[45] for details on how to search, sort, filter, and view all modules in the machine tab.

Main Window

The **main window** provides a centralized location for managing modules installed on your local and remote machines. The main window also displays the program Options.

The local machine, remote machines, and program options display as tabs in the main window:

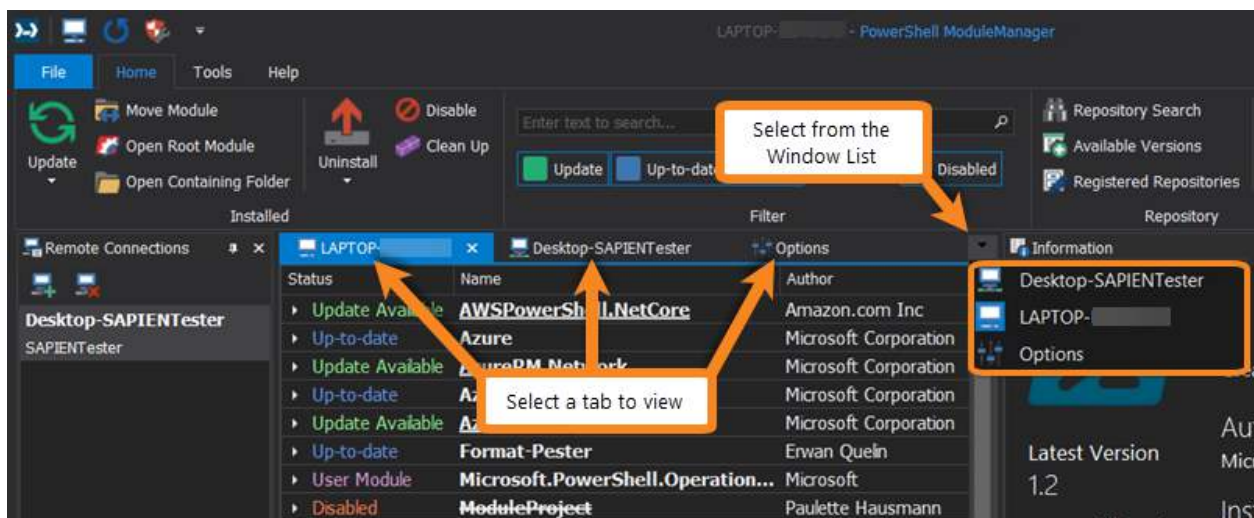


Main Window - Local Machine tab, Remote Machine tab, Options tab

How to navigate between tabs

To navigate between tabs in the main window


- Click the desired tab, or select an option in the Window List drop-down on the top-right:

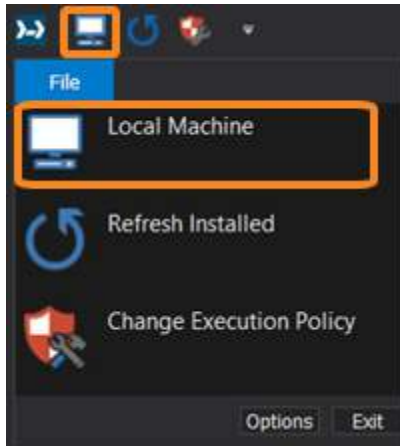


How to view the local machine

To view the local machine

- By default, the local machine is always visible in the main window.

👍 If the local machine tab is not visible in the main window you can reopen it by clicking the  button on either the Quick Access Toolbar or the File menu (**File > Local Machine**):

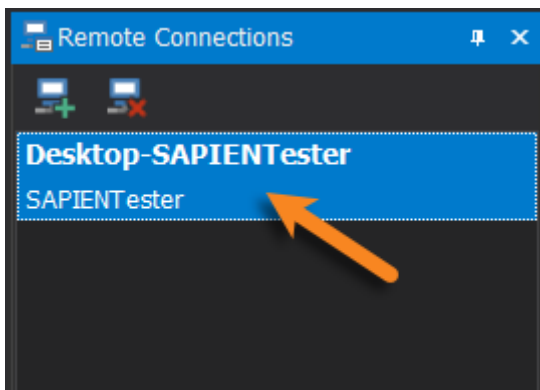


How to view a remote machine

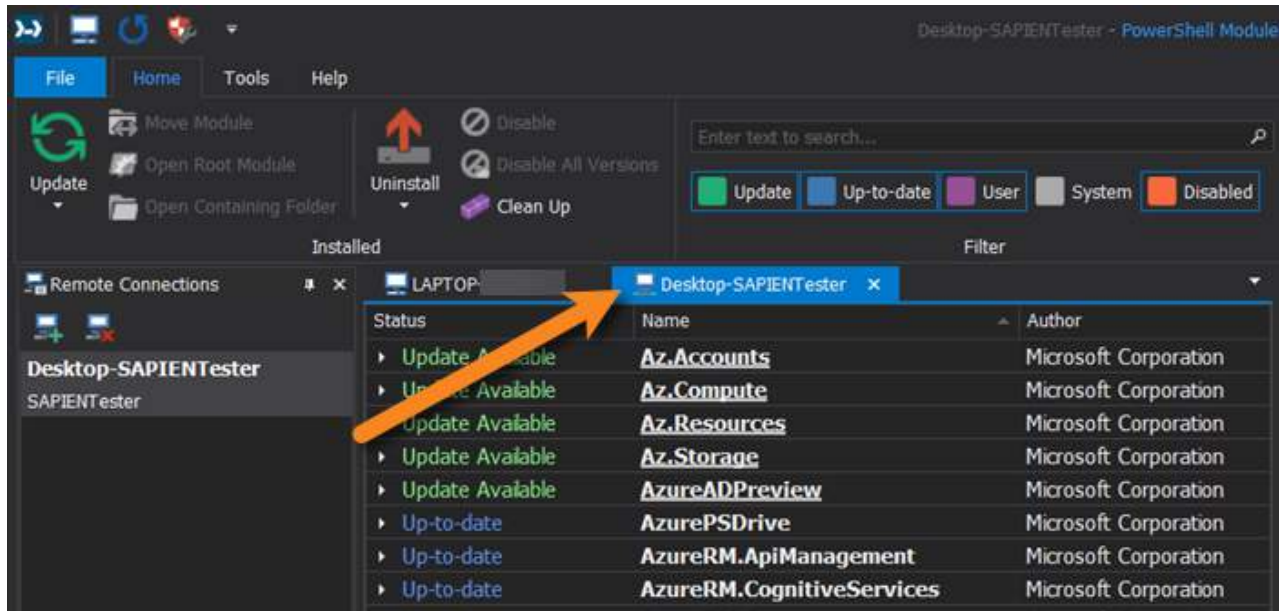
- To view a remote machine, you must first [add a remote connection](#) ⁹³.

To view a remote machine

- In the Remote Connections panel, double-click on an existing connection:



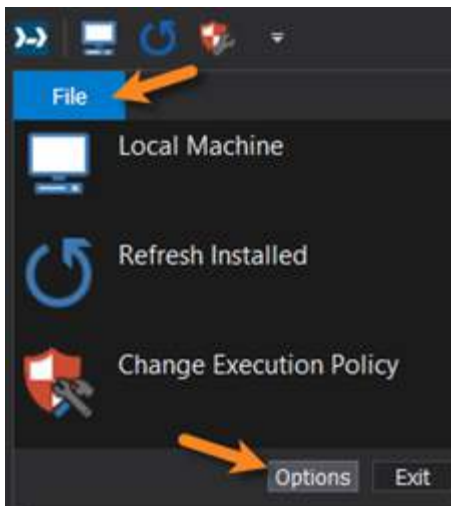
A new tab with the computer's name will appear in the main window when the remote machine is connected:



How to view the program options

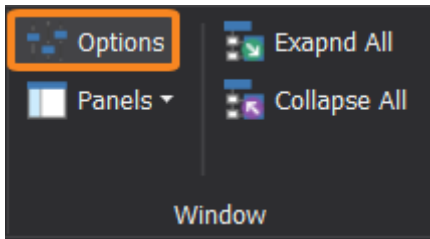
To view the program options

- On the ribbon, select **File > Options**:



-OR-

- On the ribbon, select **Home > Options**:

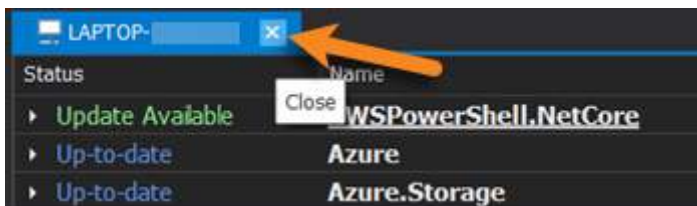



The program Options will appear as a tab in the main window.

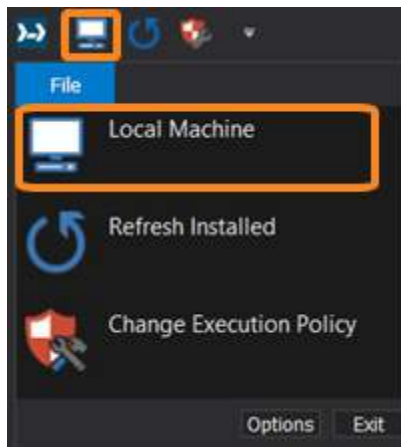
How to close a tab in the main window

To close a tab in the main window

- Click the X on the right of the tab:

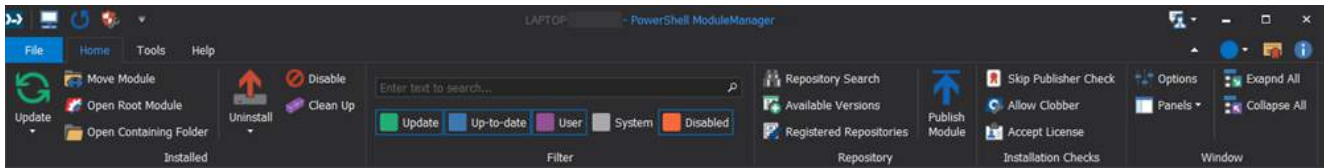


👍 If the local machine tab is not visible in the main window you can reopen it by clicking the  button on either the Quick Access Toolbar or the File menu (**File > Local Machine**):



3.2.2 Toolbars

Virtually all of PowerShell ModuleManager's functions are available on the Ribbon Toolbar, or "**Ribbon**." The ribbon is context-sensitive, automatically displaying the functions relevant to what you are doing at the moment. Functions that are not available are grayed out.



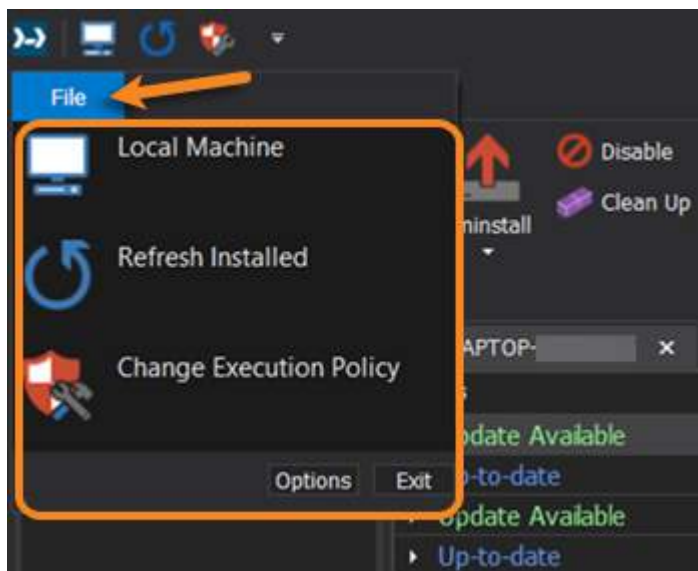
The Ribbon

The File Menu and Quick Access Toolbar



File Menu

The File tab on the ribbon has a menu of frequently used functions:



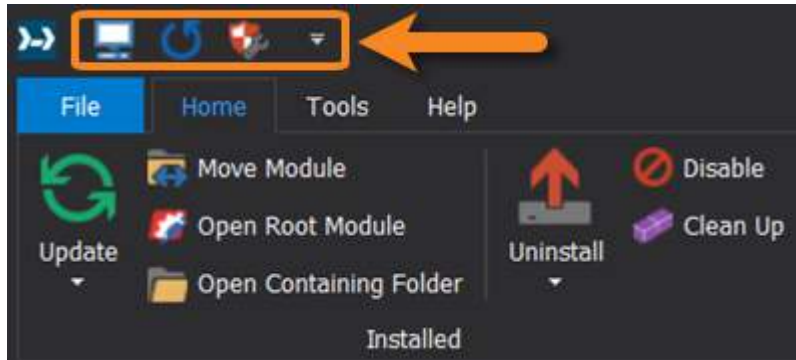
The File Menu

The File Menu options:

Local Machine	View modules installed on the local machine.
Refresh Installed (F5)	Refresh information for all installed modules.
Change Execution Policy	Opens the Script Security Center GUI interface, where you can change the PowerShell execution policy for both 32-bit and 64-bit shells.
Options	Access the PowerShell ModuleManager program options.
Exit	Close the PowerShell ModuleManager application.

Quick Access Toolbar

The configurable mini toolbar above the Ribbon provides direct access to frequently used functions:

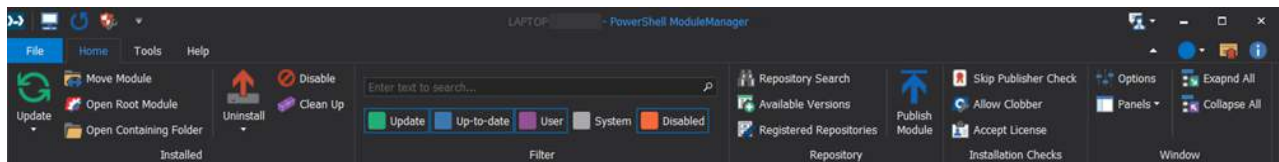


Quick Access Toolbar

👍 You can easily [add your own frequently used functions](#)^[155] to the Quick Access Toolbar.

The Ribbon Toolbar

The [Ribbon](#)^[106] is organized in tabs, each of which contains groups of tools:







Each tab on the ribbon contains tools grouped into sections based on related functions. For example, the **Home** ribbon > **Installed** section has controls used for managing installed modules—Updating, Moving, Disabling, and so on. Some controls also provide access to dialogs with additional options.

3.2.3 Panels - Overview


The PowerShell ModuleManager workspace has dockable window panels that are used by dedicated features or to display output.

PowerShell ModuleManager Panels

Disabled Modules 	Displays disabled modules. Use this panel to enable and uninstall modules.
Information 	Displays detailed information for modules highlighted in the machine tab.
Log 	Displays PowerShell ModuleManager output messages.
Remote Connections 	Displays remote connections. Use this panel to add, remove, and connect to remote machines.

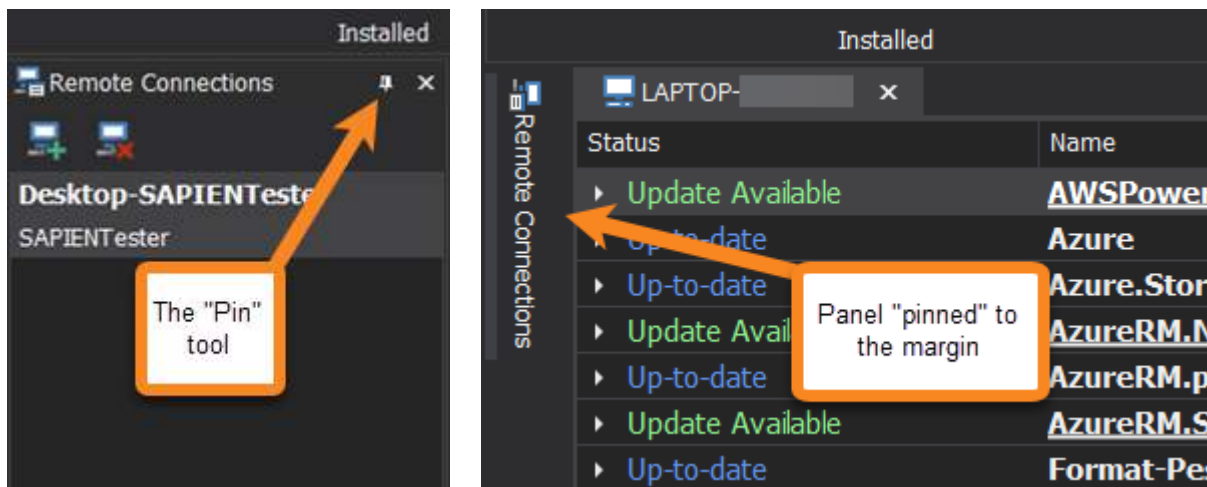
Undocking and docking panels

Undocking, docking, and pinning PowerShell ModuleManager panels

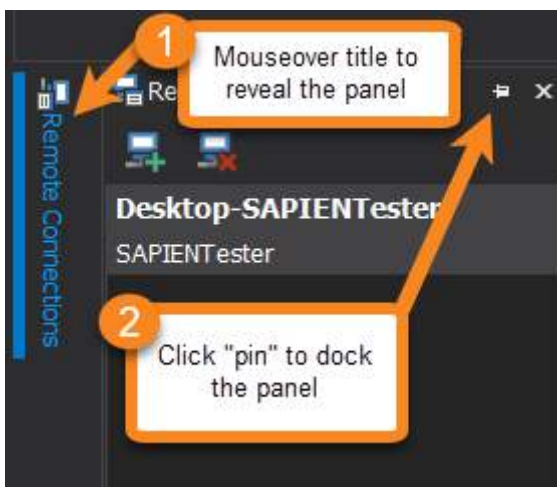
- Double-click on a panel title bar or click-and-drag it out of the program window to display or "float" it in a separate window.
- To dock a floating panel, drag it into the program window and drop it onto one of the [blue docking indicators](#) .
- You can also "pin" (auto-hide) individual panels to the margin by clicking on the pin tool in the top-right of the panel's title bar (see below).

Pinning a panel to the margin

- Click the "pin" tool on the top-right of the panel title bar to auto-hide the panel to the margin of the window. The panel will then be hidden and will only display when you mouseover the panel title in the margin.



- To "unpin" the panel from the margin and return it to its most recently docked position, simply mouseover the title to display the panel and then click the "pin" tool in the title bar. The panel will return to its most recent docked position.



i For more information about working with panels, see Customizing Your Workspace > [Panels and Layout](#) ¹⁵⁹.

3.2.4 Skins, Options and Shortcuts

PowerShell ModuleManager provides options for customization and configuration of the program workspace. You can change the color theme, designate the default repository, control if you are prompted when uninstalling modules, and more.

Many program functions can be accessed using predefined keyboard shortcuts, and you can also add frequently used functions to the [Quick Access Toolbar](#) ¹⁵⁵. You can even reconfigure the layout of the PowerShell ModuleManager [panels](#) ¹⁵⁹.

Changing the color-themed skin

You can change PowerShell ModuleManager's user interface color theme from the **Skins** menu above the Ribbon:



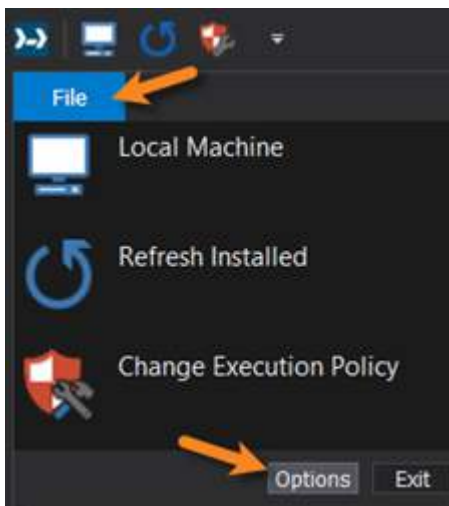
Skins button

Program configuration options

The program Options allow you to control various module settings such as cache, uninstall, and update behavior. You can also set the Repository and Installation Scope for a selected machine, reset the Quick Access Toolbar, reset or reload the Panel layout, and more.

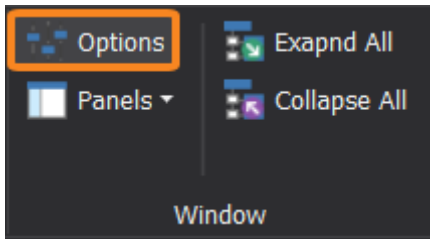
To access the program Options


- On the ribbon, select **File > Options**:



-OR-

- On the ribbon, select **Home > Options**:

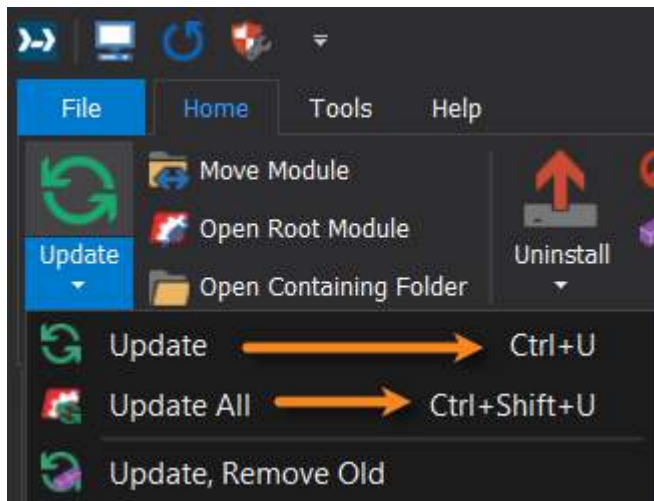


 View Reference > [Program Options](#)^[98] for complete details about the available program options.

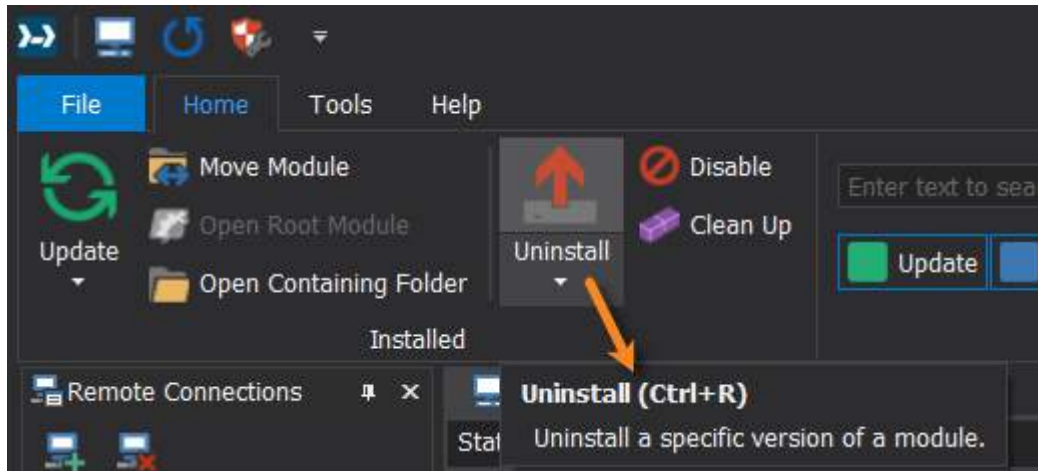
Keyboard shortcuts and accelerators

Keyboard shortcuts

Some PowerShell ModuleManager functions are configured with [keyboard shortcuts](#)^[153], which display in the menu options and ribbon button tooltips:



Keyboard shortcuts in menu options



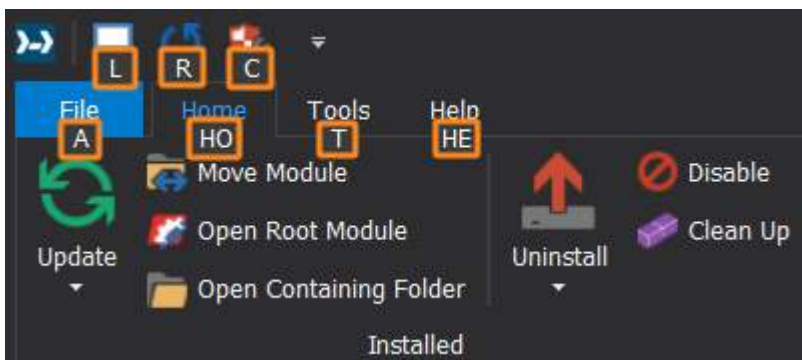
Keyboard shortcuts in ribbon button tooltips

i View Reference > Workspace, Ribbons and Dialogs > [Keyboard Shortcuts](#)¹⁵² for the complete list of available keyboard shortcuts.

Accelerator keys

The ribbon interface also has accelerator keys.

1. Press the **ALT** key once to display small icons showing accelerator letters and codes on the ribbon controls:




Accelerator keys

2. Type the accelerator letters on the keyboard to perform the corresponding actions. If multiple letters are displayed, type them one after another—these are not key combinations!

Additional accelerator keys will be displayed to access any menus in the individual ribbon groups.

3.3 Staying Up-to-date


We are continually updating PowerShell ModuleManager, both to remove bugs and to add and improve product features. We recommend always staying current with the most recent version to ensure that you are taking advantage of the latest features, functionality, and product stability.

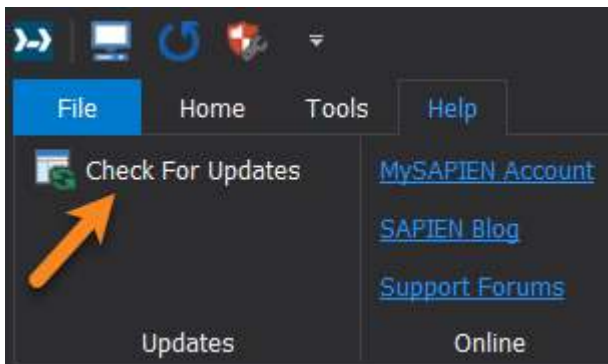
 The details for every PowerShell ModuleManager release are available in the [version history](#).

Check for Updates


By default, PowerShell ModuleManager will automatically check for software updates. You can also manually check for updates.


To manually check for updates

- On the **Help** ribbon (Updates section) > click **Check For Updates** to open the [SAPIEN Updates](#)  tool and see if there is a new PowerShell ModuleManager build available:



3.4 Getting Help

This help manual has been designed to provide all the information you will need for using PowerShell ModuleManager. In addition to the information in this guide, you can also ask questions in the [online support forums](#) .

 View the latest PowerShell ModuleManager tips, product feature demonstrations, and release details on [our blog](#).


Displaying the help manual

- Select the **Help** ribbon > **Product Manual**.

User forums and support

Use the SAPIEN forums to get help with PowerShell ModuleManager, get answers to your scripting questions, submit feature requests, and more.


- [PowerShell ModuleManager forum](#)
- [Scripting support forums](#)
- [All SAPIEN support options](#)

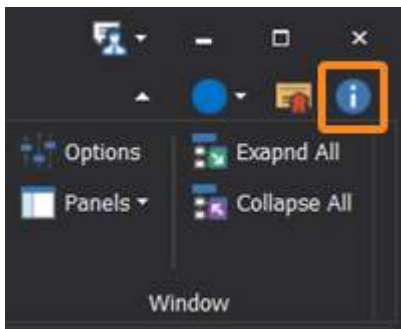
 To report a problem in the PowerShell ModuleManager forum, you will need to provide your SAPIEN product and OS [version information](#) ²⁹.

How to copy version information

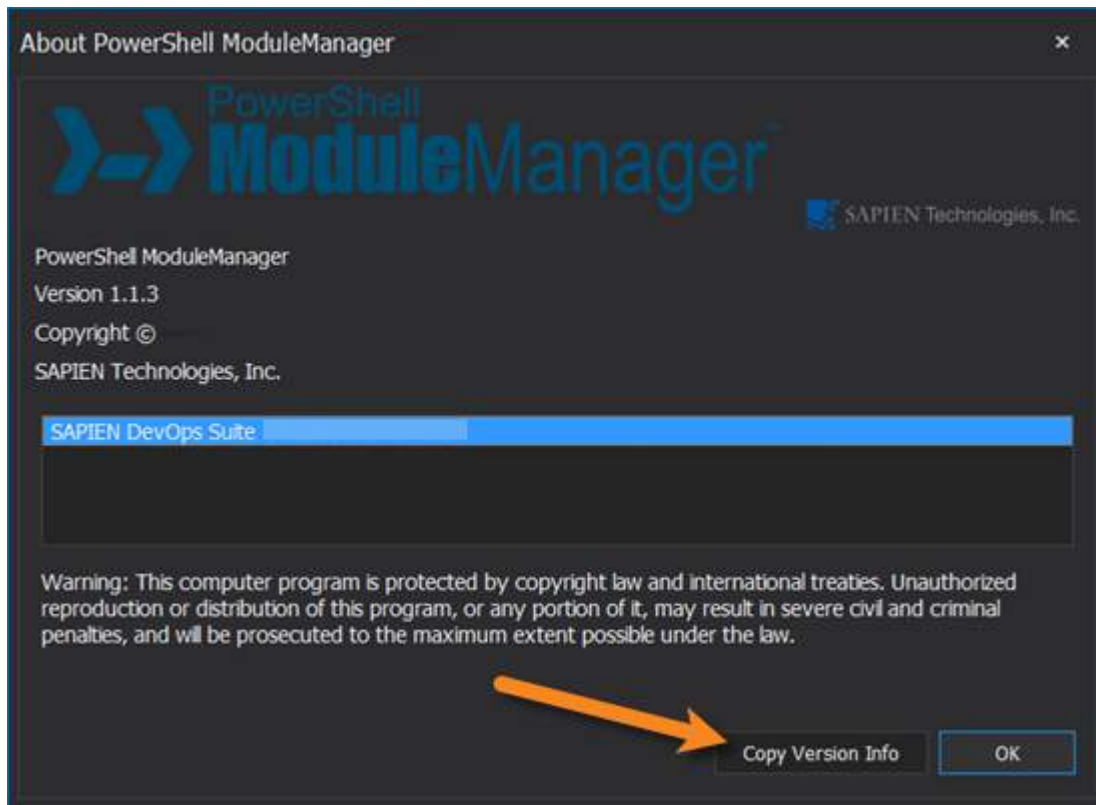
To report a problem in the [PowerShell ModuleManager forum](#), you will need to include the product version and build, and also your OS version and build—and indicate 32 or 64-bit for each.

To copy the required version information

1. Click the  **About** button in the top-right of the PowerShell ModuleManager workspace:

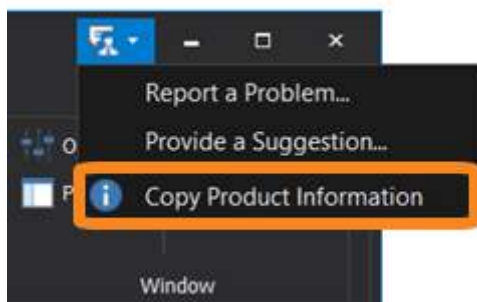


2. In the About PowerShell ModuleManager window, select **Copy Version Info**:



-OR-

On the Send Feedback menu in the top-right of the PowerShell ModuleManager workspace, select **Copy Product Information**:



3. Paste the version information into your [PowerShell ModuleManager forum](#) post.

4 Managing Modules

This section is designed as a "How-To" guide covering the basic tasks you will use when working with PowerShell ModuleManager. The topics are presented roughly in the order that you would perform the tasks, but you can easily jump to any topic and work your way through.

Once you have learned the basics in this section, we encourage you to explore the PowerShell ModuleManager [Program Options](#)^[98] and also see how you can [Customize Your Workspace](#)^[154].

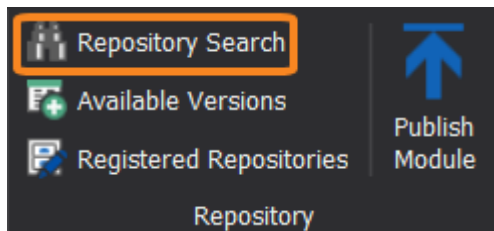
Detailed information on all of the PowerShell ModuleManager functions are located in the [Reference](#)^[98] section.

4.1 Searching for Modules

You can search for modules from the PowerShell Gallery or any other registered repository, and install modules directly from the **Repository Search** dialog. To search for modules already added, see [Finding Installed Modules](#)^[45].

Repository Search

Repository Search (*Ctrl+F*) displays module information from registered repositories and allows you to add modules to your system.

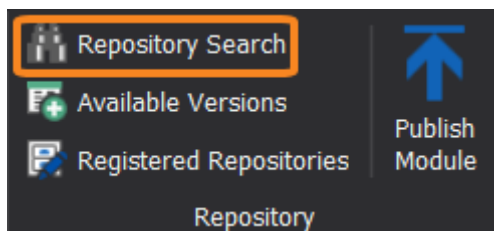


Home ribbon > Repository Search

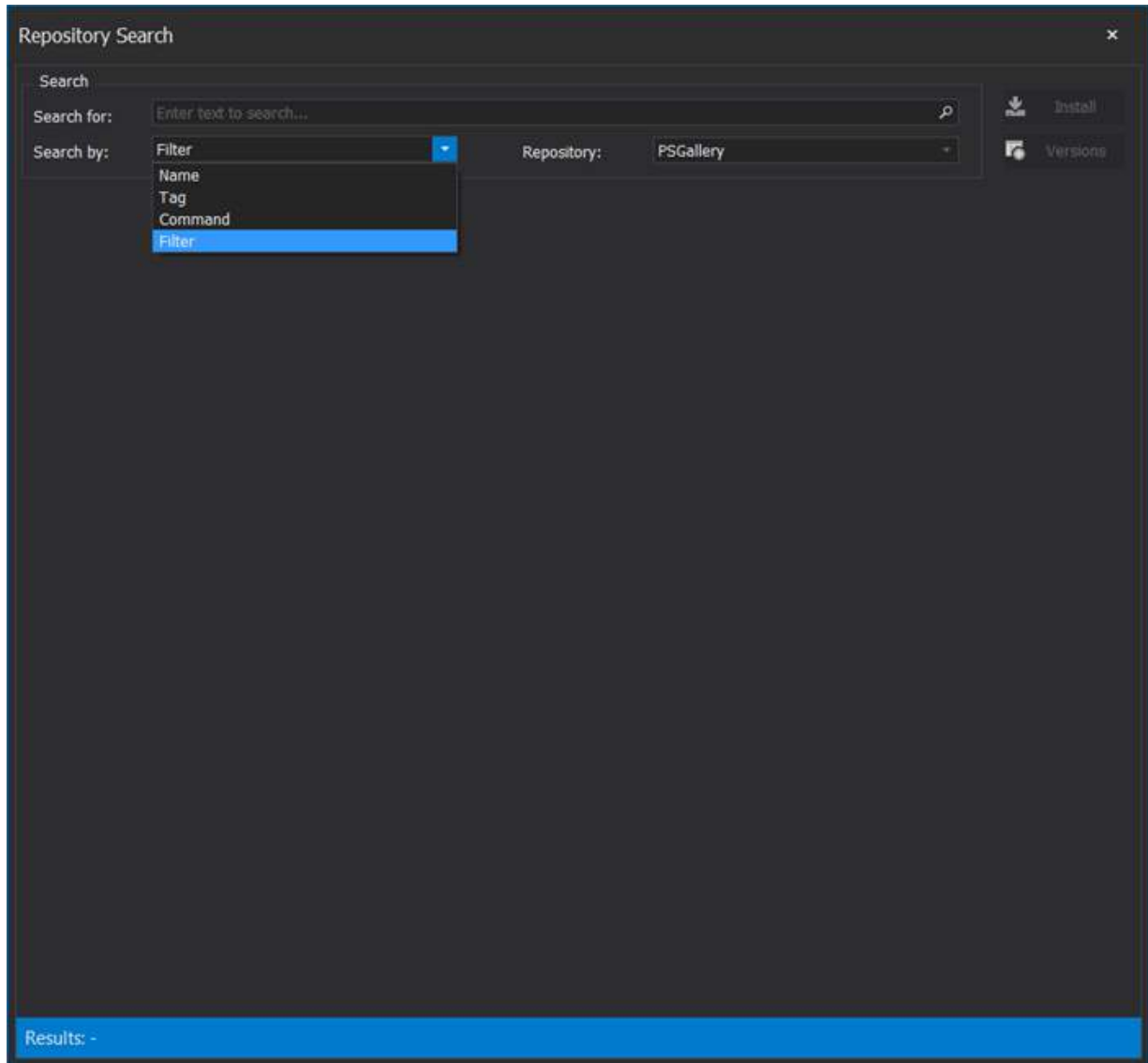
How to search for modules using Repository Search

To search for modules

1. On the ribbon, select **Home > Repository Search** (*Ctrl+F*):

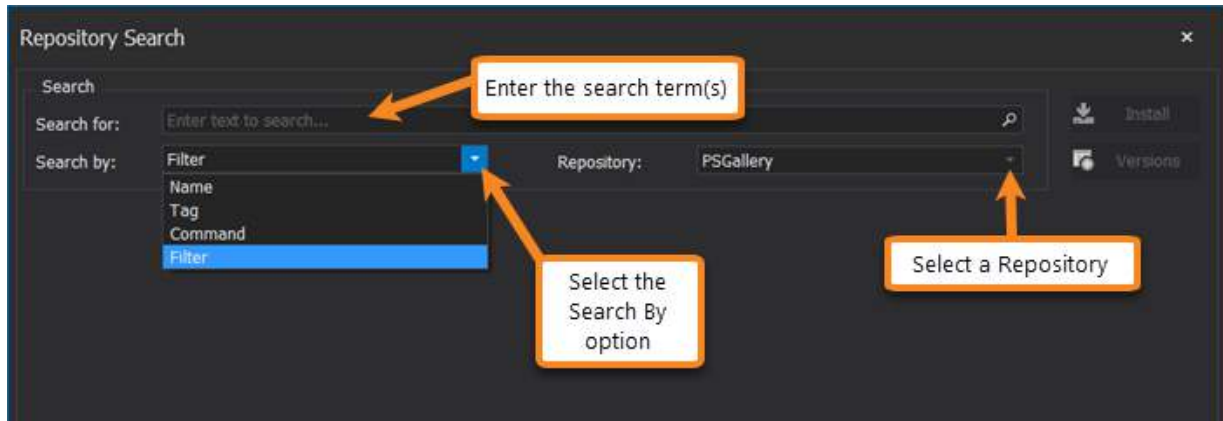


The Repository Search dialog will open:



Repository Search Dialog

2. Enter the search criteria in the Repository Search dialog, and then press the <Enter> key:



Repository Search - Field Definitions:

Search for Enter the search term(s).

Search by Select an option from the drop-down list.

Name

Specifies the names of one or more modules to search for.

👍 This is the only parameter that supports wildcard characters.

Tag

Keywords used to help find modules.

Command

Cmdlets, functions, or aliases found in the module.

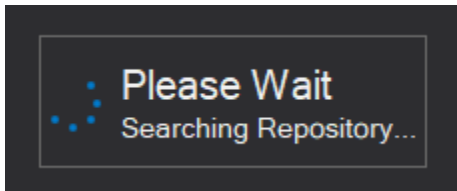
Filter

Filters through the name, description, and tags for the search argument based on the PackageManagement module provider-specific search syntax. For NuGet modules, this is the equivalent of searching by using the Search bar on the PowerShell Gallery website.

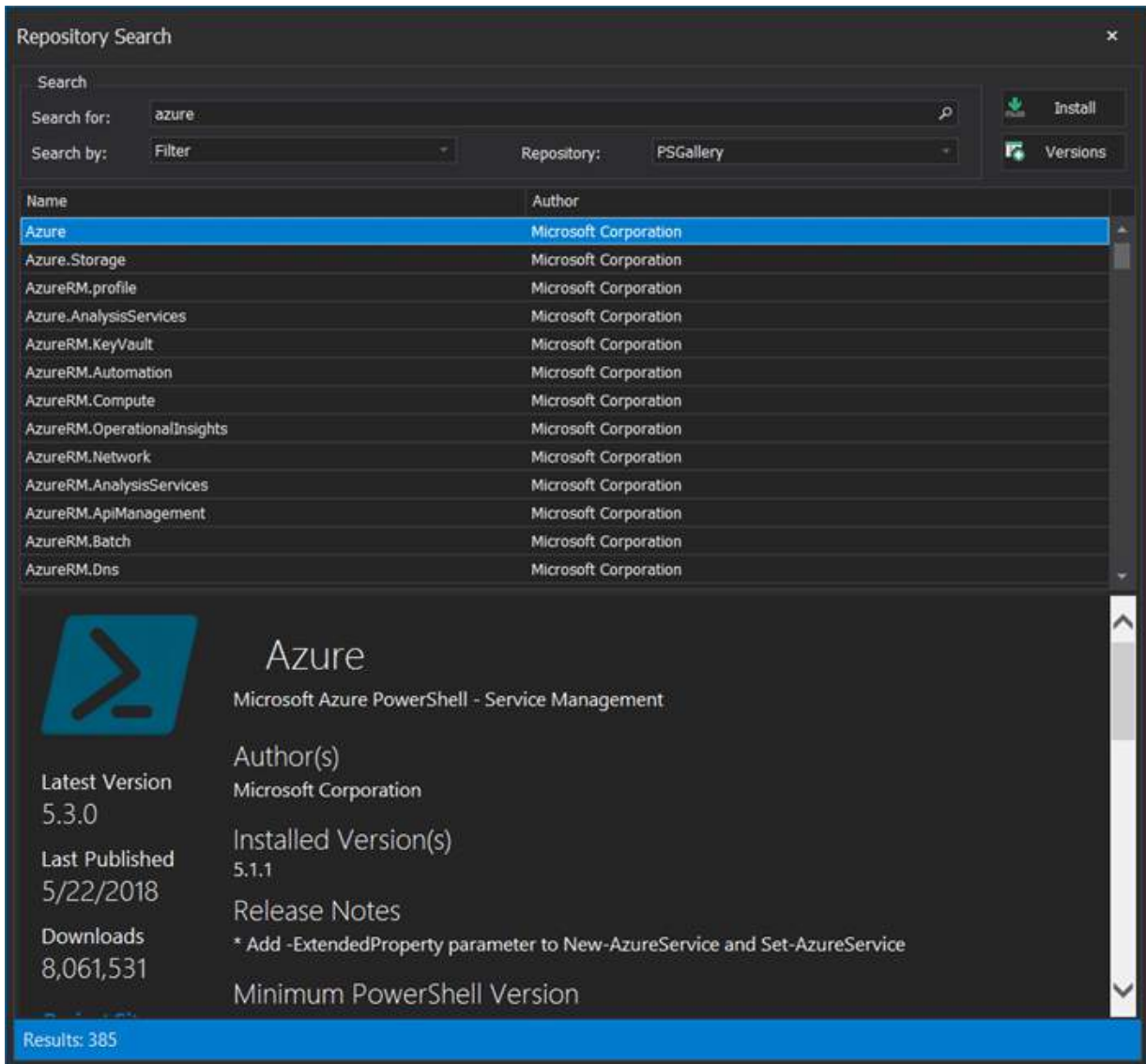
Repository Select a repository from the drop-down list.

ℹ The default registered repository is the PowerShell Gallery (PSGallery). This setting can be changed in **File > Options > General > PowerShell**.

PowerShell ModuleManager will perform the search:



The Repository Search dialog displays the search results:

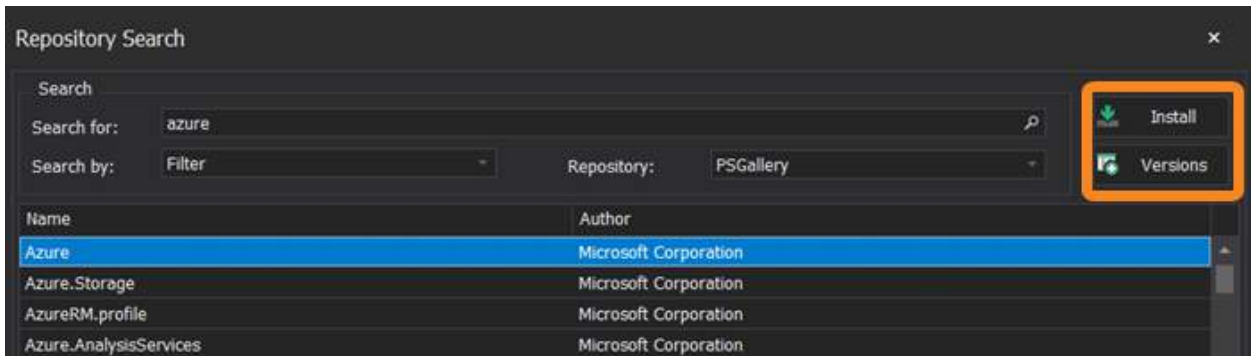


👍 Click on a module to view the module details.

See [Managing Repositories](#)^[84] for information on adding, editing, and removing repositories.

To install a module

- To add the latest version of a module, select **Install** (*Ctrl+I*).



👍 To view all available versions, select [Versions](#)^[150] (*Ctrl+V*).

📘 The Install option is disabled if the most recent module version is already installed on your machine.

See [Installing Modules](#)^[35] for information about available versions and installation options.

4.2 Installing Modules

You can install modules from registered repositories using the Repository Search dialog, or you can use install other versions of installed modules from the Machine tab. PowerShell ModuleManager also provides some installation options when installing or updating modules.

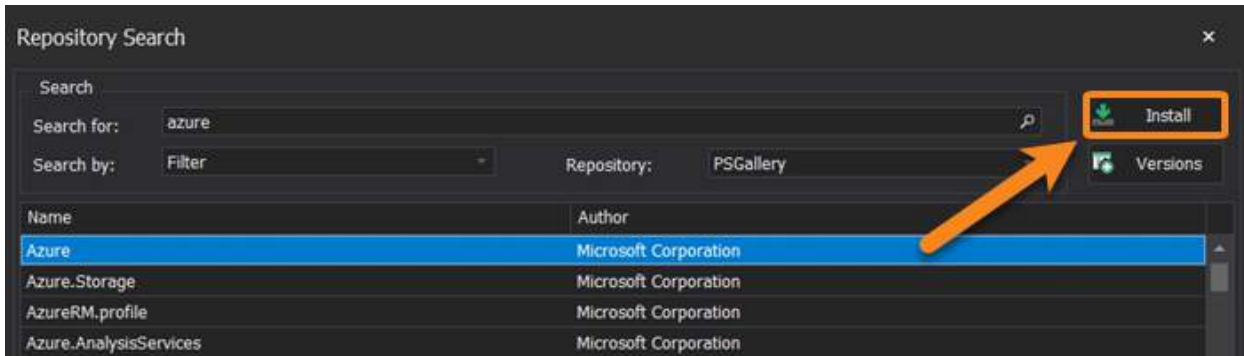
Installing from Repository Search

[Repository Search](#)^[31] allows you to search for modules in registered repositories and install the latest module version, or a specific module version.

How to install the most recent module version

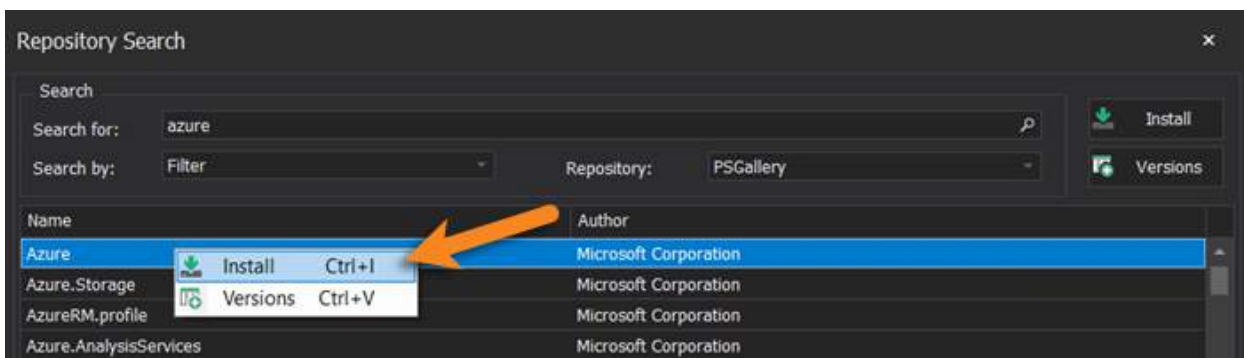
To install the most recent module version

- In the [Repository Search](#)^[31] dialog, select the module and then click the **Install** button (*Ctrl+I*):



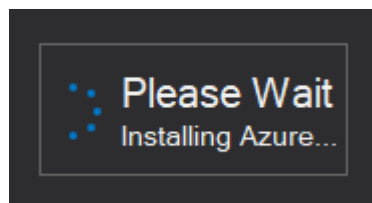
-OR-

- Right-click on the module, then select **Install** (*Ctrl+I*) from the context menu:

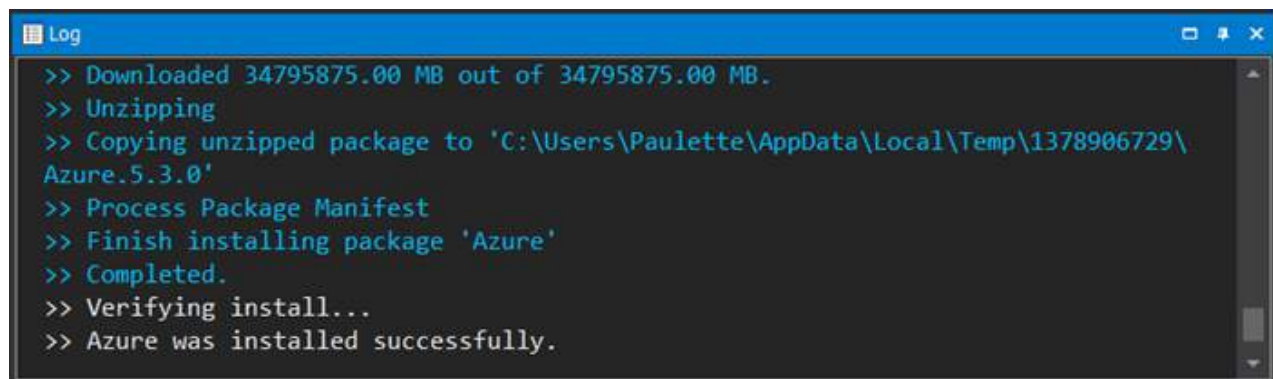


- i The **Install** options are disabled if the most current version of the module is already installed.

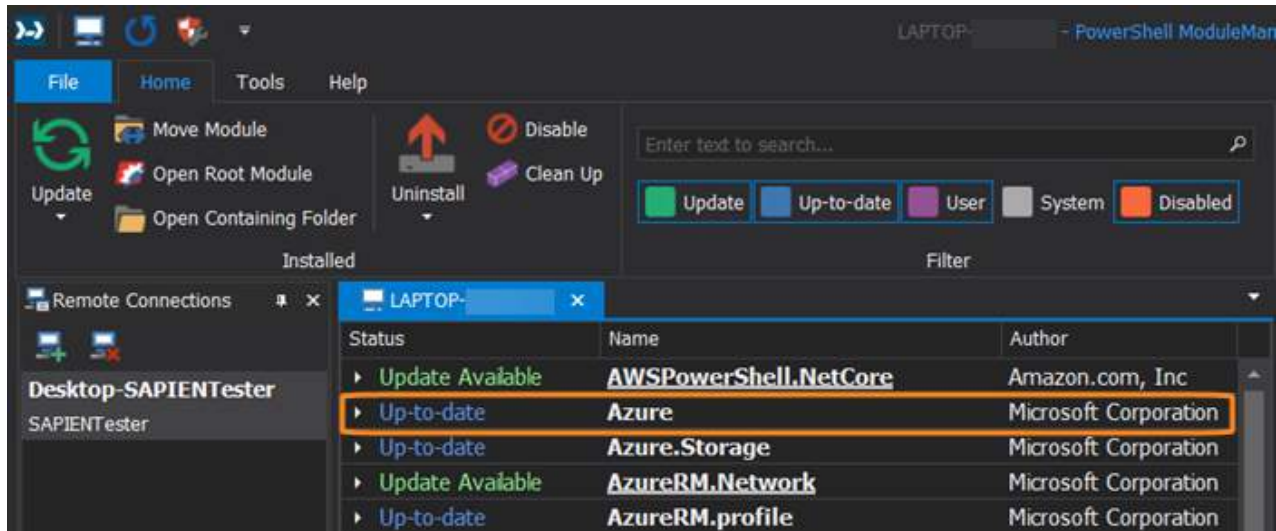
PowerShell ModuleManager will install the latest version of the selected module:



The installation progress and status are displayed in the Log panel:



When the installation is complete, the module will appear in the Machine tab:

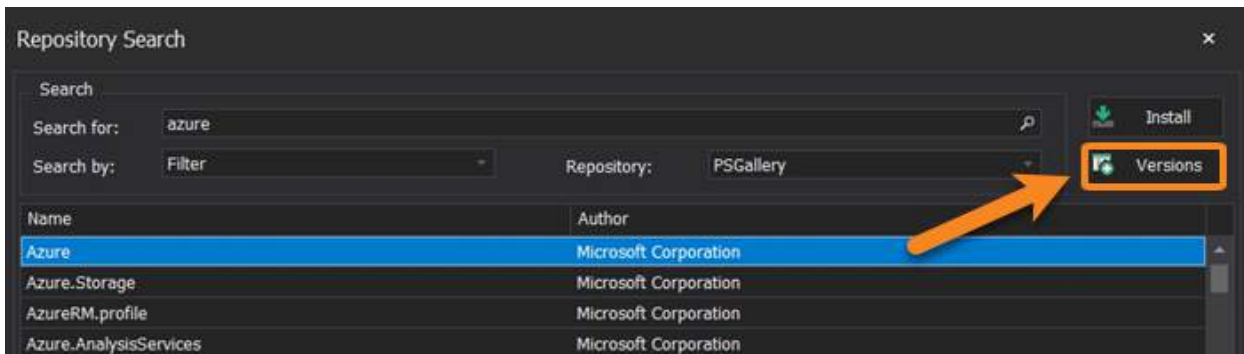


- i The 'Up-to-date' status indicates that the most current version of the module is installed.

How to install a specific module version

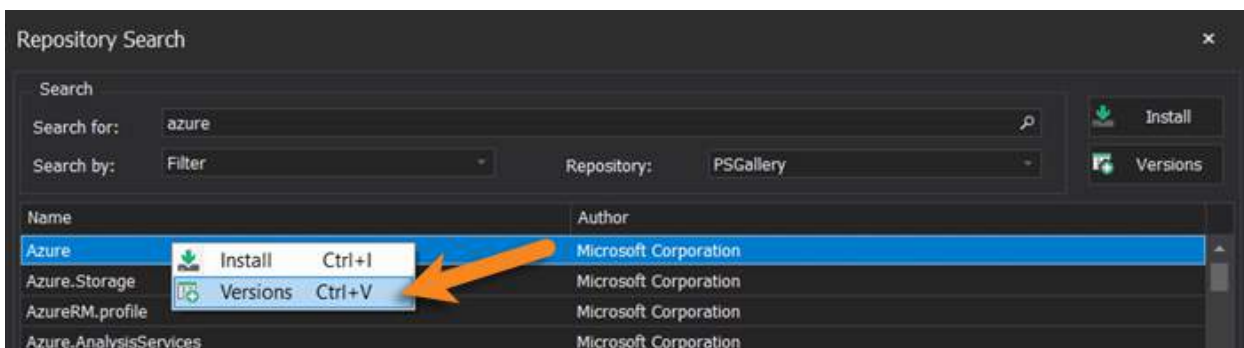
To install a specific module version

- In the [Repository Search](#) ³⁷ dialog, select the module and then click the **Versions** button (Ctrl+V):

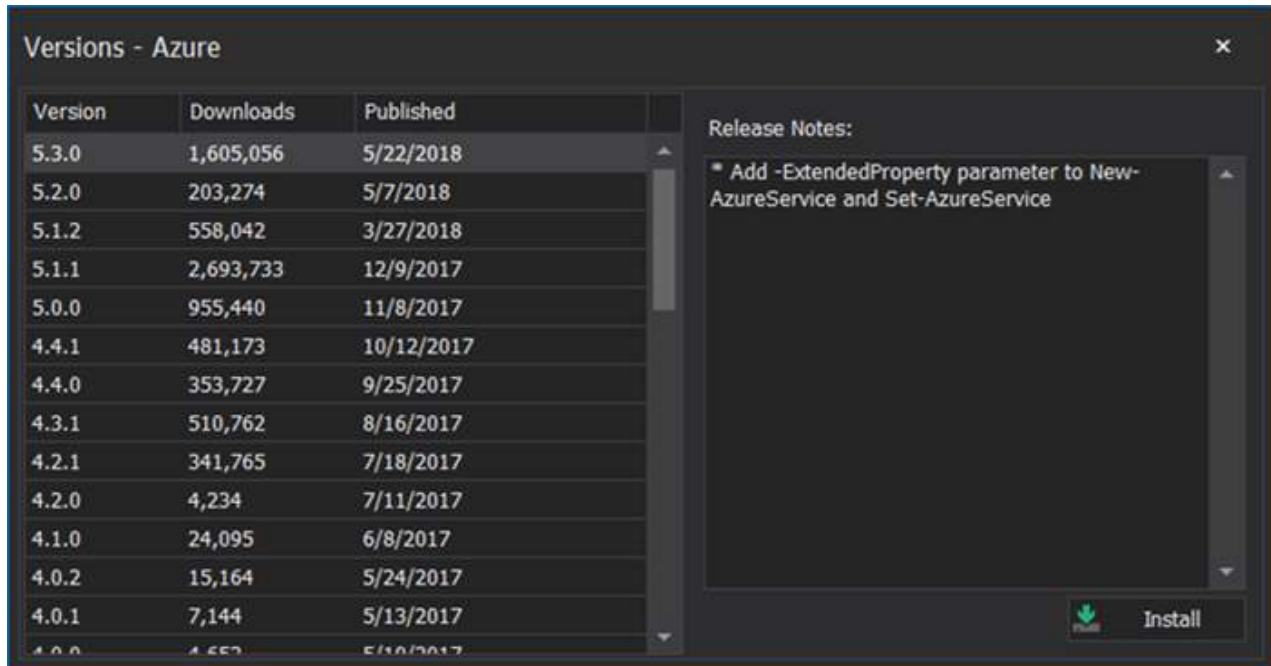


-OR-

- Right-click on the module, then select **Versions** (Ctrl+V) from the context menu:

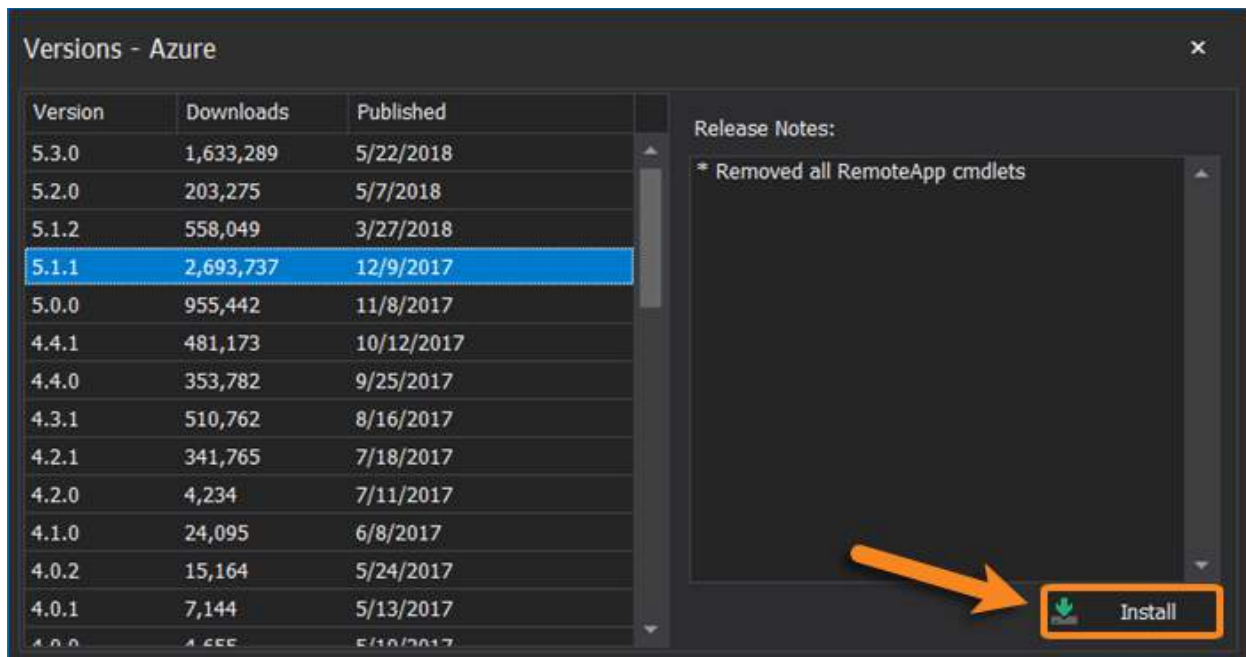


The Versions dialog will open:



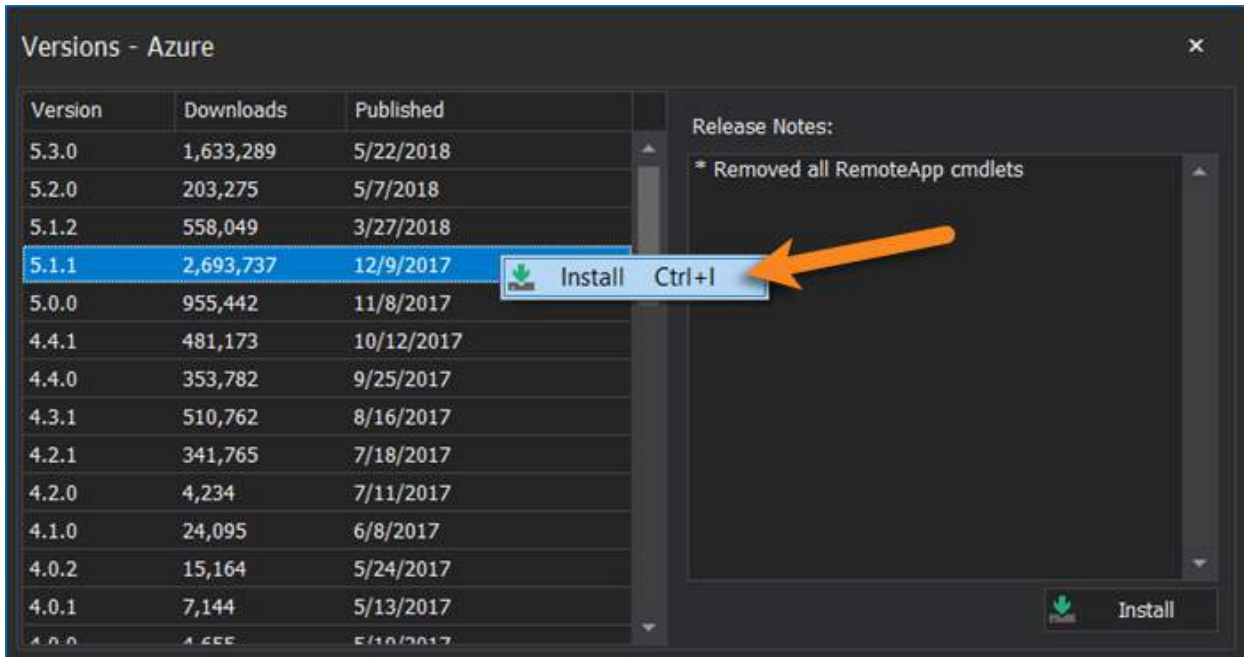
Versions Dialog

- Select a module version, then click the **Install** button (*Ctrl+I*):



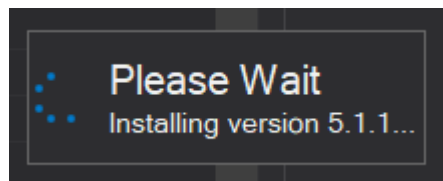
-OR-

- Right-click on a module version, then select **Install** (*Ctrl+I*) from the context menu:

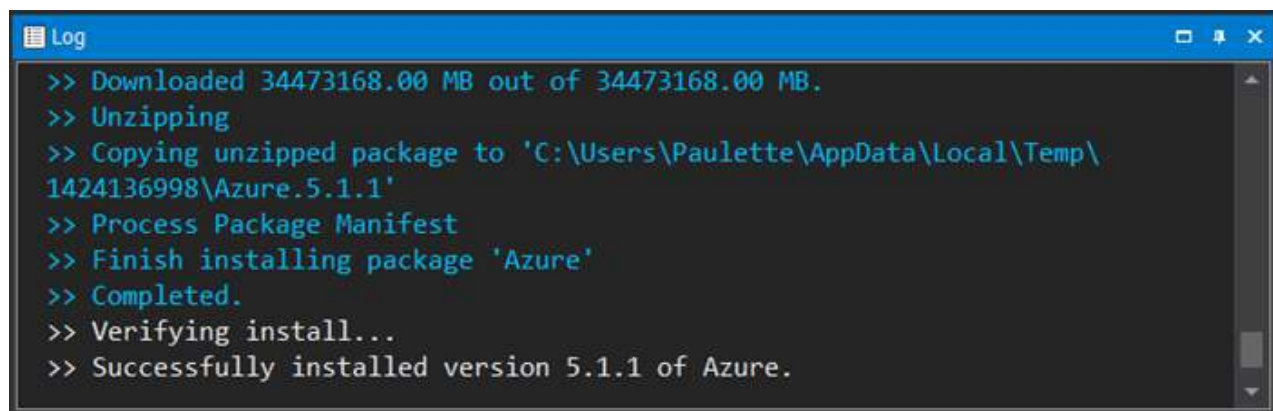


i The **Install** options are disabled if the selected module version is already installed.

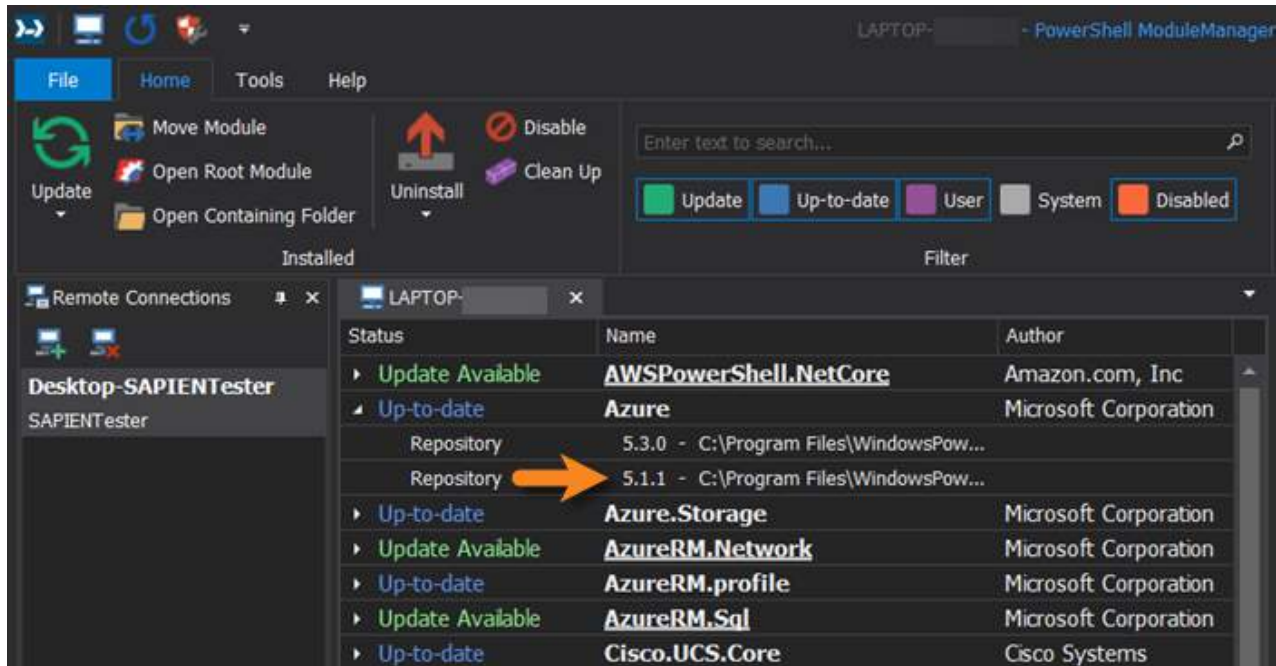
PowerShell ModuleManager will install the selected module version:



The installation progress and status are displayed in the Log panel:



When the installation is complete, the selected module version will appear in the Machine tab:



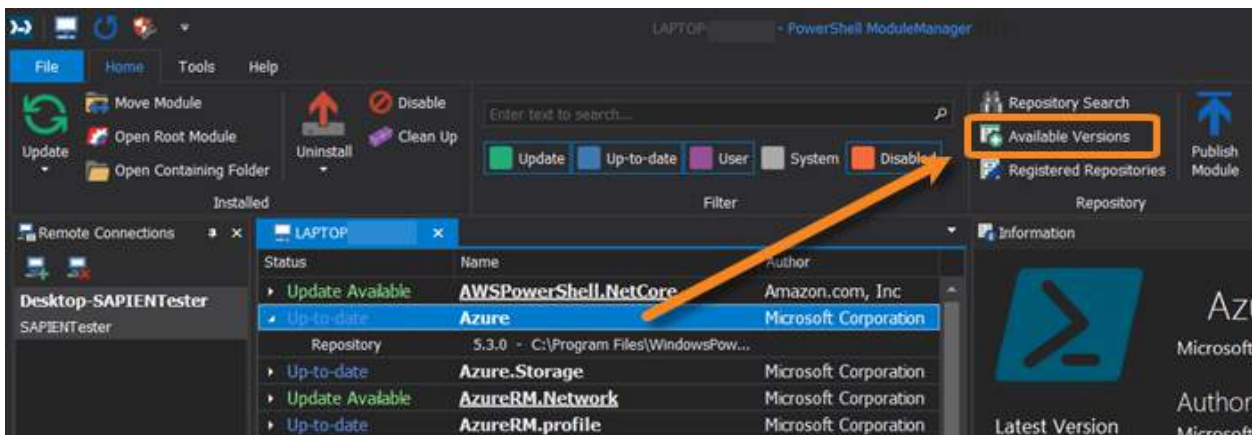
Installing from the Machine tab

The [Machine](#) ¹⁵ tab allows you to view and install additional versions of installed modules.

How to install additional module versions

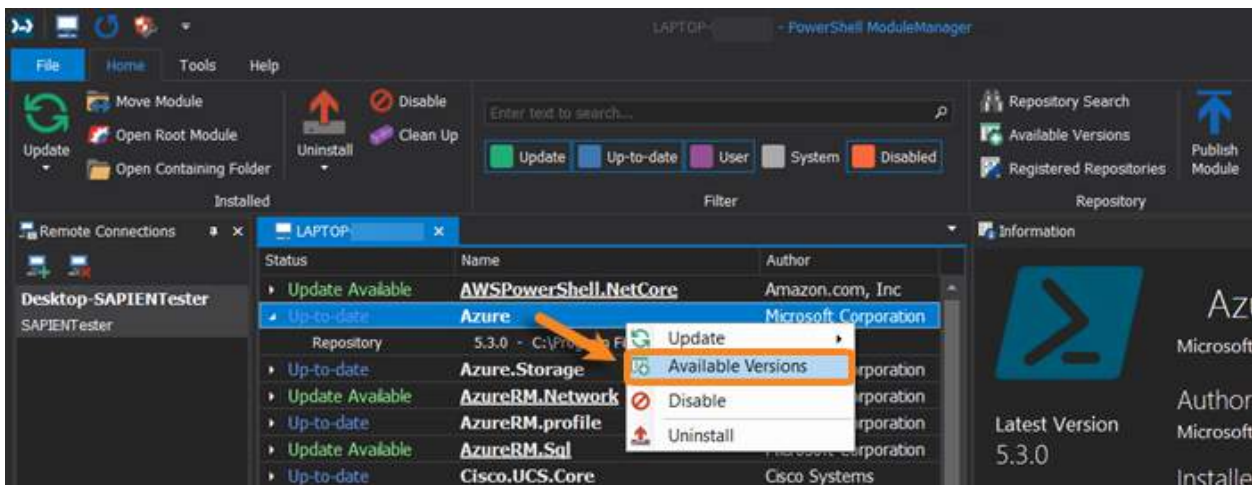
To install another module version

- In the Machine tab, select the module; then on the Home ribbon, click **Available Versions** (Ctrl + V):

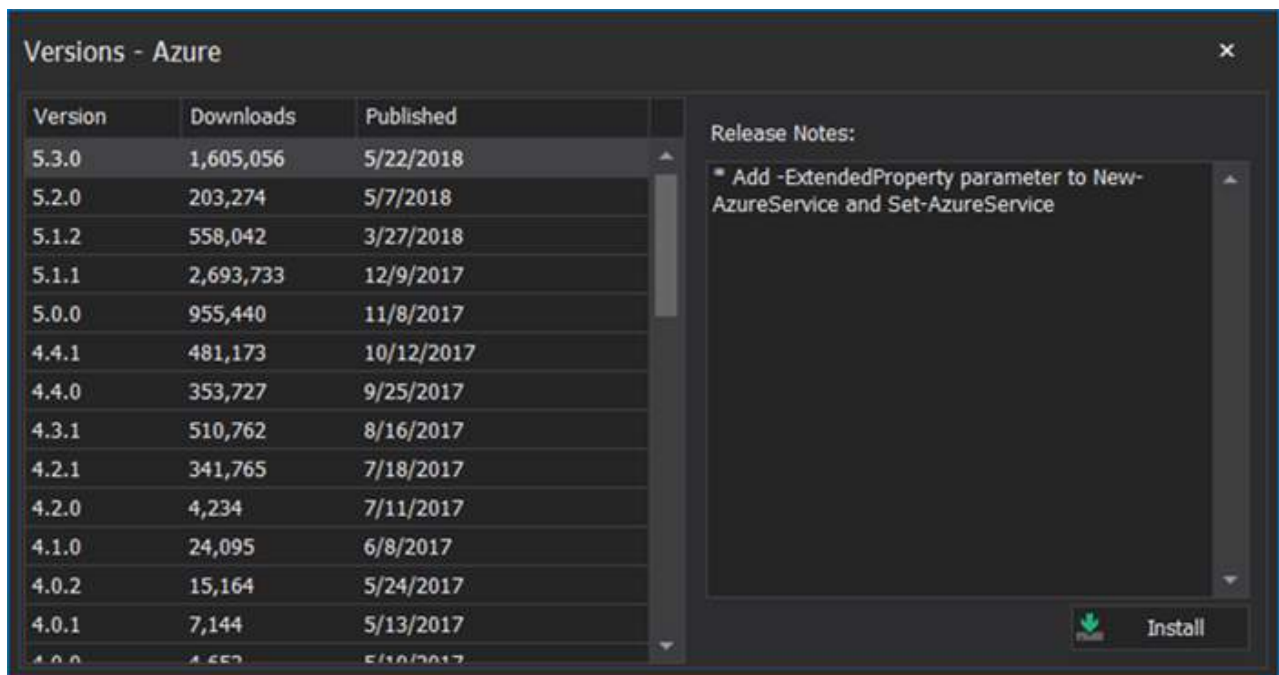


-OR-

- In the Machine tab, right-click on the module and select **Available Versions** (Ctrl + V) from the context menu:

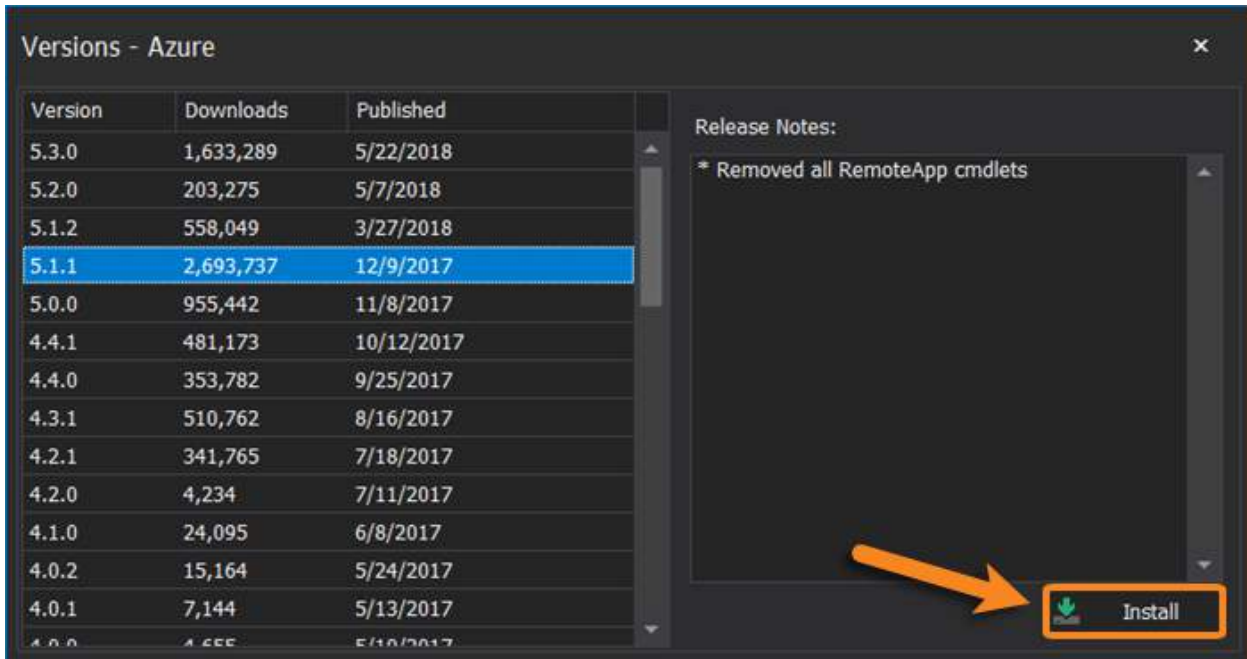


The Versions dialog will open:



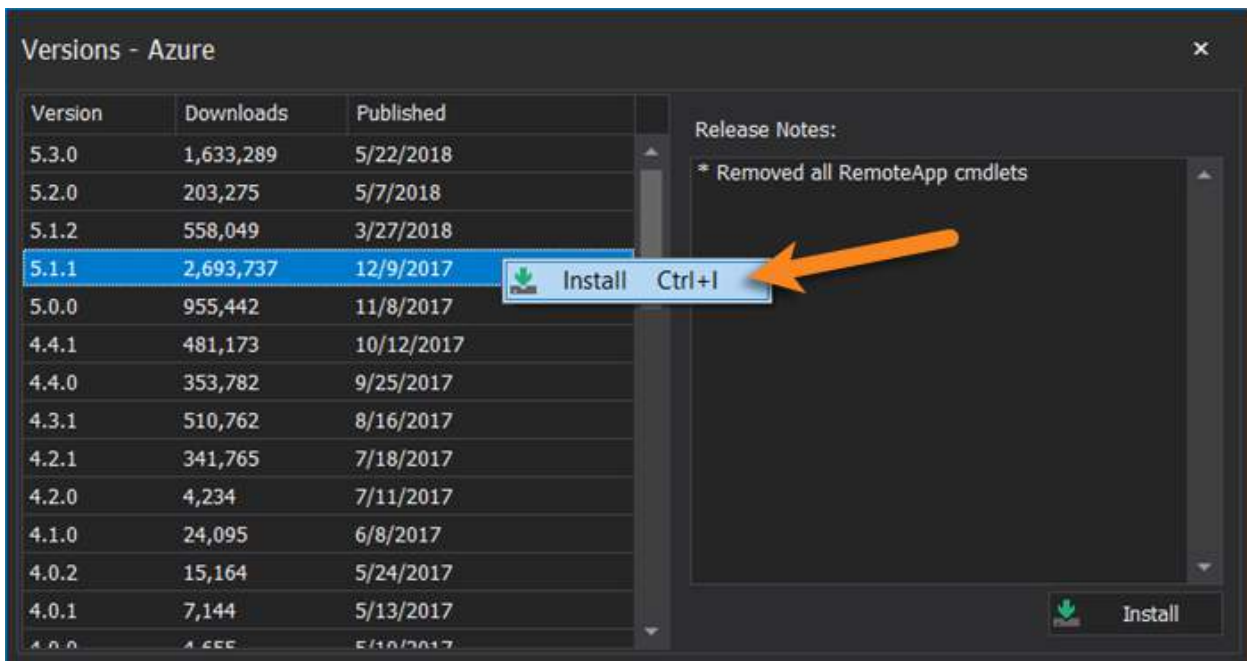
Versions Dialog

- Select a module version, then click the **Install** button (Ctrl+I):



-OR-

- Right-click on a module version, then select **Install** (*Ctrl+I*) from the context menu:



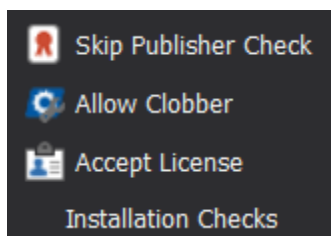
- i The **Install** options are disabled if the selected module version is already installed.

Module Installation Options

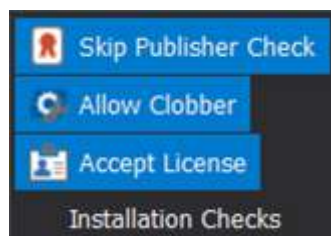
PowerShell ModuleManager allows you to control some installation options when installing or updating modules, including the default folder location.

Installation Checks

The installation options appear in the Installation Checks section on the Home ribbon, and the buttons act like toggles—turning the features on or off:





*Installation Checks -
toggled OFF*

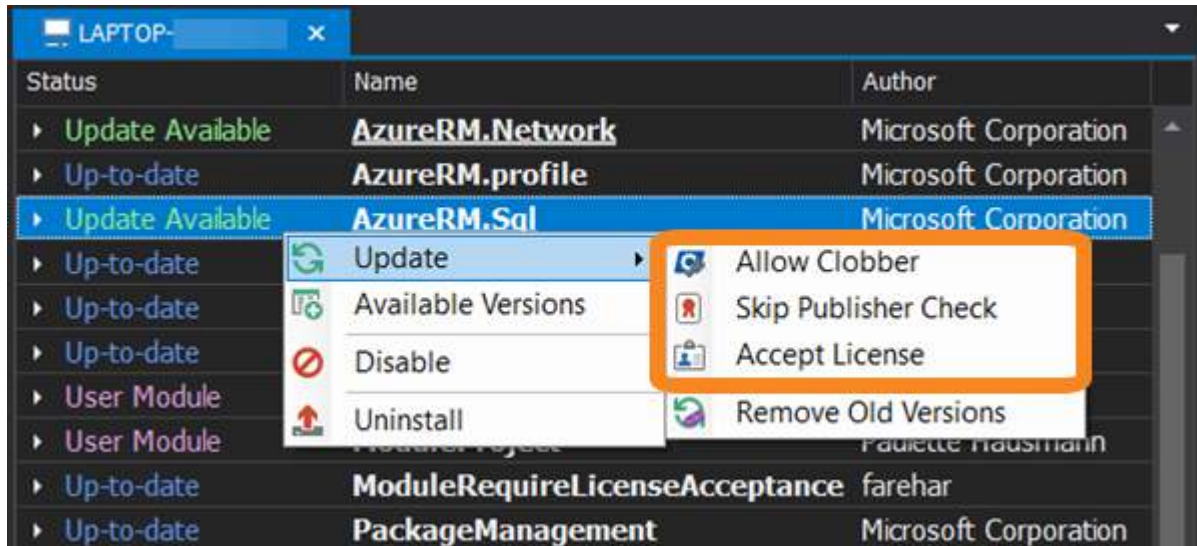


*Installation Checks -
toggled ON*

Installation Options:

Skip Publisher Check	Allows the installation of a newer module version that is digitally signed by a trusted publisher, replacing an existing, unsigned version.
Allow Clobber	Forces the installation of a module, even when another module contains an identical command.
Accept License	Accepts the license agreement during installation.  This option is not available in older versions of PowerShellGet.

 The installation options are also available in the Machine tab context menu when you right-click a module:



Installation Scope

The Installation Scope specifies where modules will be installed when updating or installing:

- **Current User**

Modules will be installed in a location that is only available to the current user:

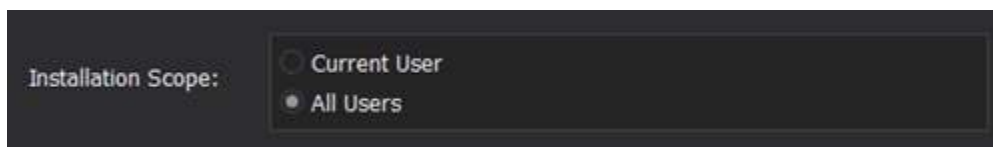
`%systemdrive%\Documents\WindowsPowerShell\Modules`

- **All Users**

Modules will be installed in a location accessible to all users of the computer:

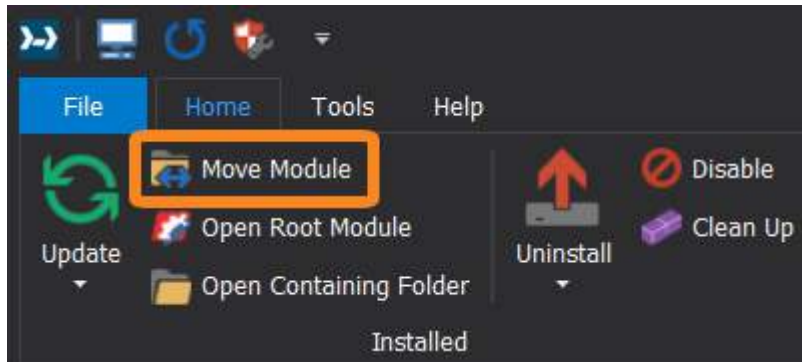
`%systemdrive%\Program Files\WindowsPowerShell\Modules`

i The default Installation Scope can be set in **File > Options > General**:



Options > General > Installation Scope

👍 The [Move Module](#)⁵⁹ control on the Home ribbon allows you to move a module's location:



Home ribbon > Move Module

4.3 Managing Installed Modules

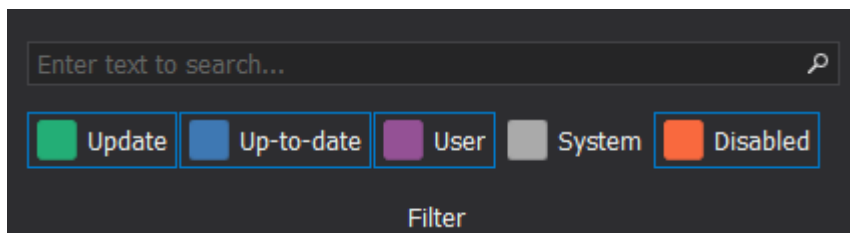
Installed modules are managed from the [Machine tab](#)^[120] which is located in the middle of the PowerShell ModuleManager [workspace](#)^[121].

4.3.1 Finding Installed Modules

Installed modules are displayed in the machine tab. This topic shows you how to view and locate specific modules and their versions.

Filter, Sort and Search

The Filter section on the Home ribbon allows you to search and filter modules installed on your machine:

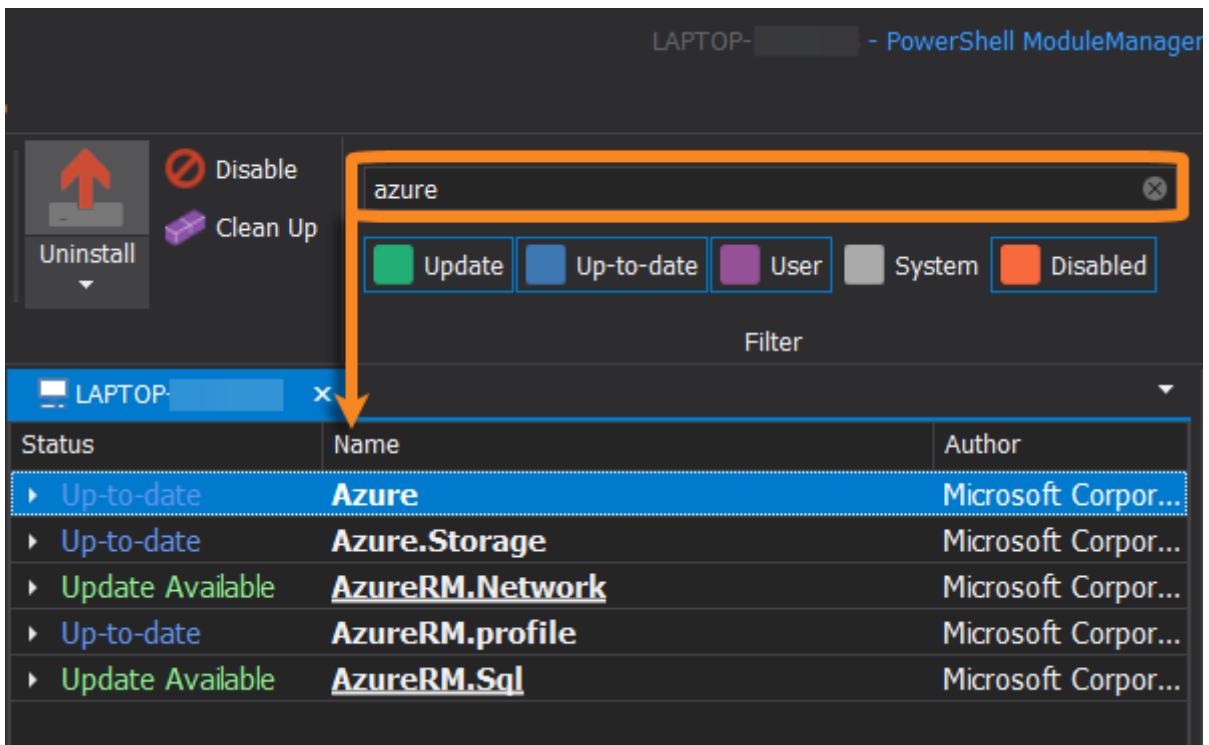


Home ribbon > Filter section

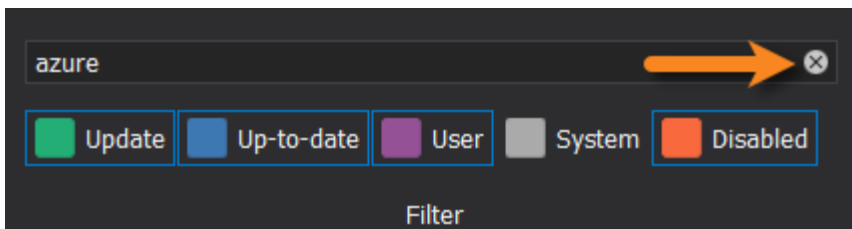
How to search for installed modules

To search for installed modules

- Enter a search term in the text box of the Filter section on the ribbon and press <Enter>. Modules containing the search term are highlighted in the machine tab:



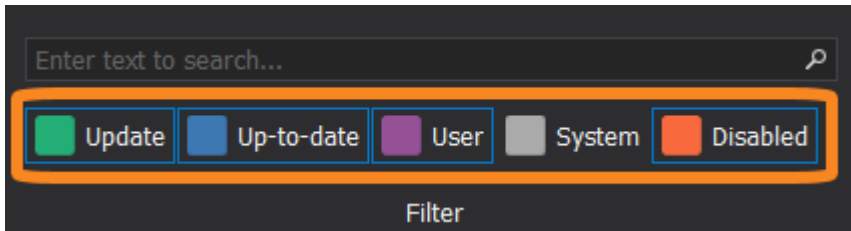
- 👍 Click the X on the right of the search text box to clear the search:



How to filter and sort installed modules

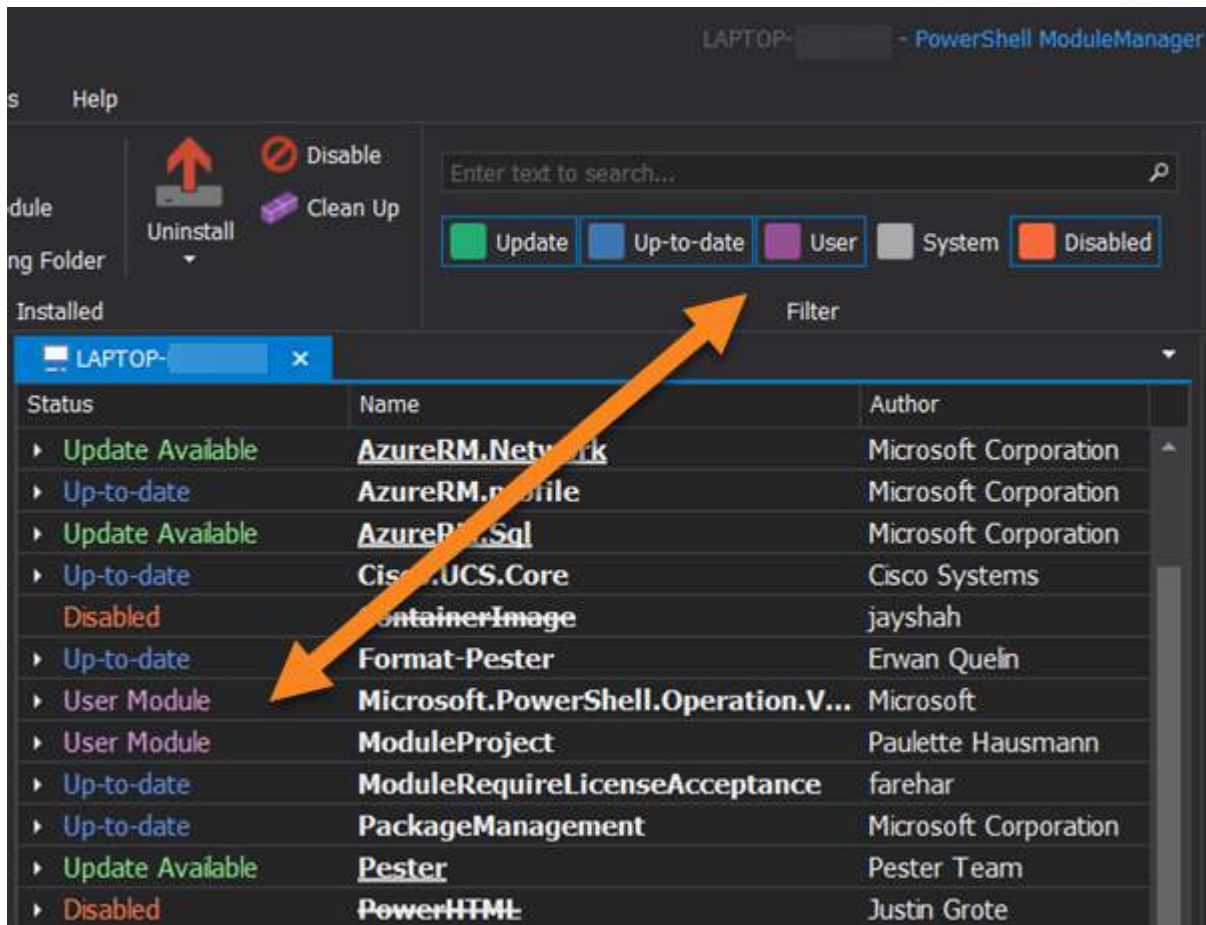
To filter installed modules

- Use the buttons in the Filter section on the ribbon to filter the modules displayed in the machine tab—click a button to check or uncheck each status type:

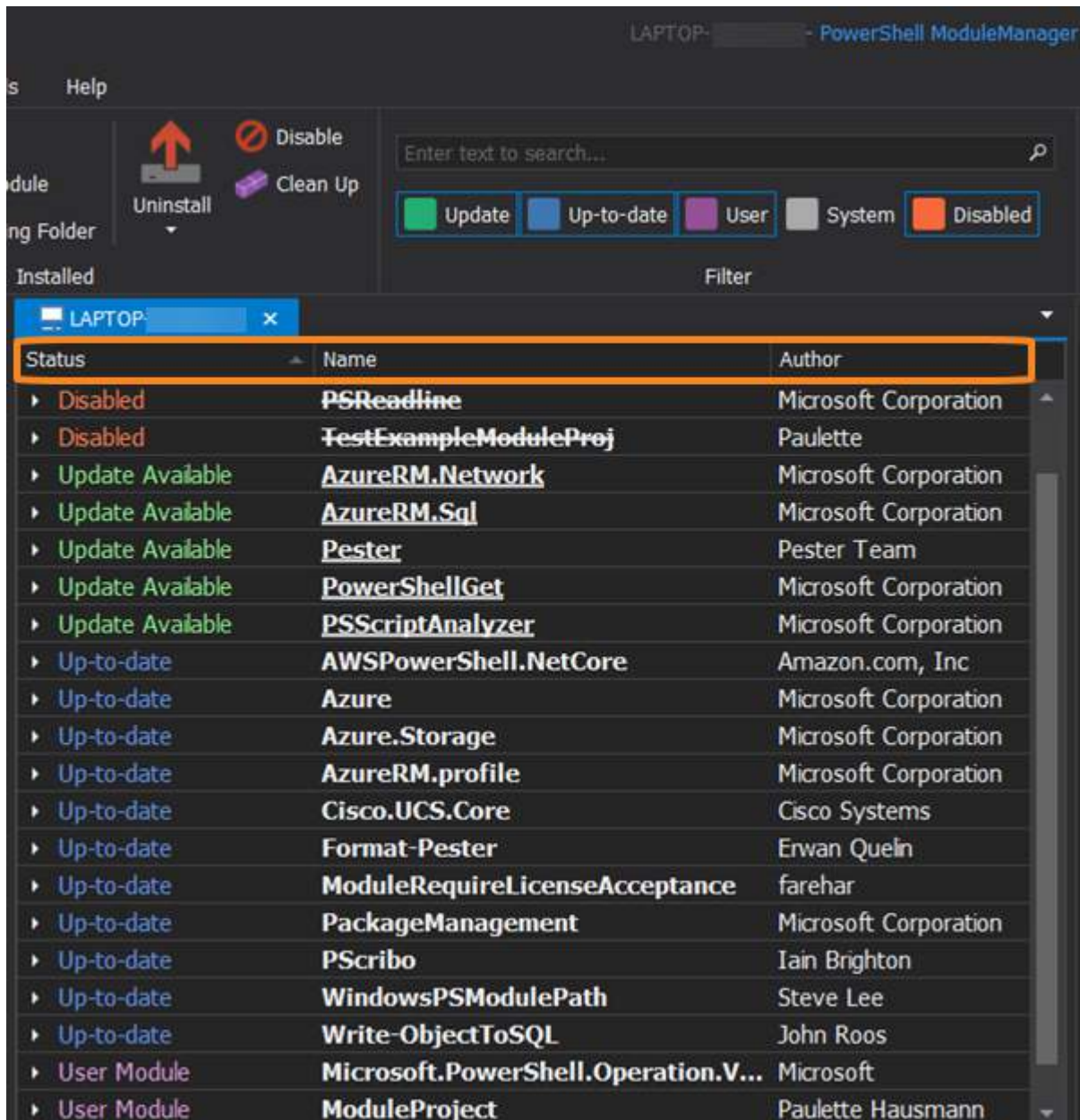


Update	A version is available that is more current than the installed version.
Up-to-date	The installed module is the most current.
User	Show modules local to your machine, or modules that PowerShell ModuleManager cannot locate on the repository designated in Options ⁹⁹ .
System	The module is in the local system folder.
Disabled	The module is disabled.

The modules in the machine tab are filtered and displayed using the same colors as the Filter buttons on the ribbon:



i You can also sort the displayed modules by clicking Status, Name, or Author at the top of each column:



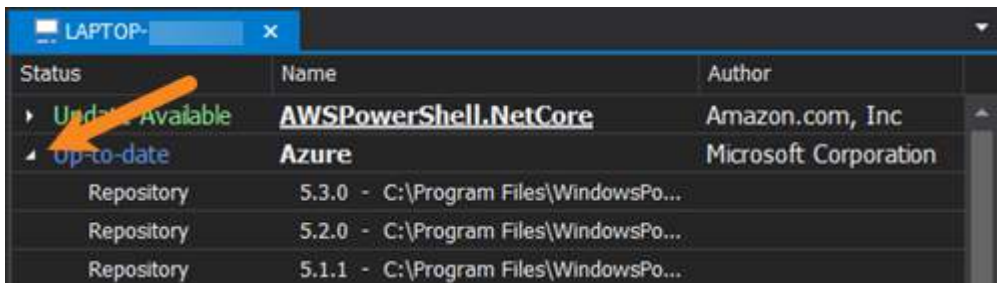
View Installed Module Versions

The machine tab provides options for viewing all installed versions of an individual module, or all installed versions of all modules.

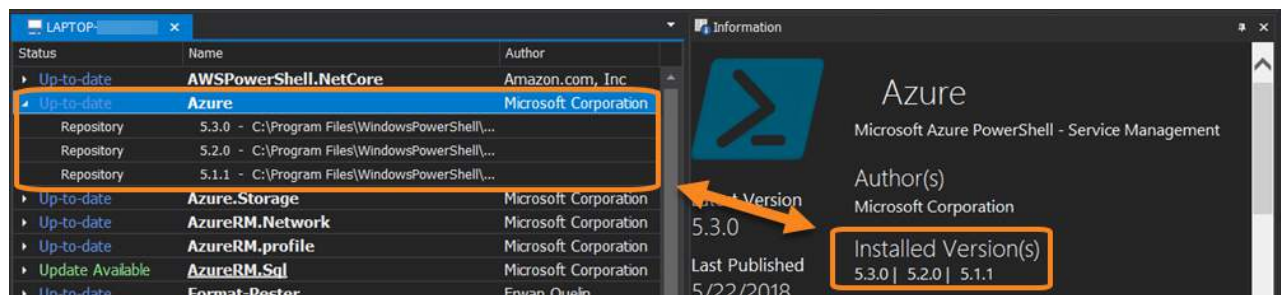
How to view all installed versions of an individual module

To view all installed versions of a module

- Click the toggle arrow to the left of the module name in the machine tab to display all installed versions of the module:



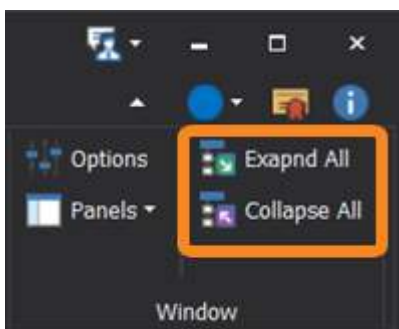
- You can also see all installed versions of the selected module in the [Information panel](#).

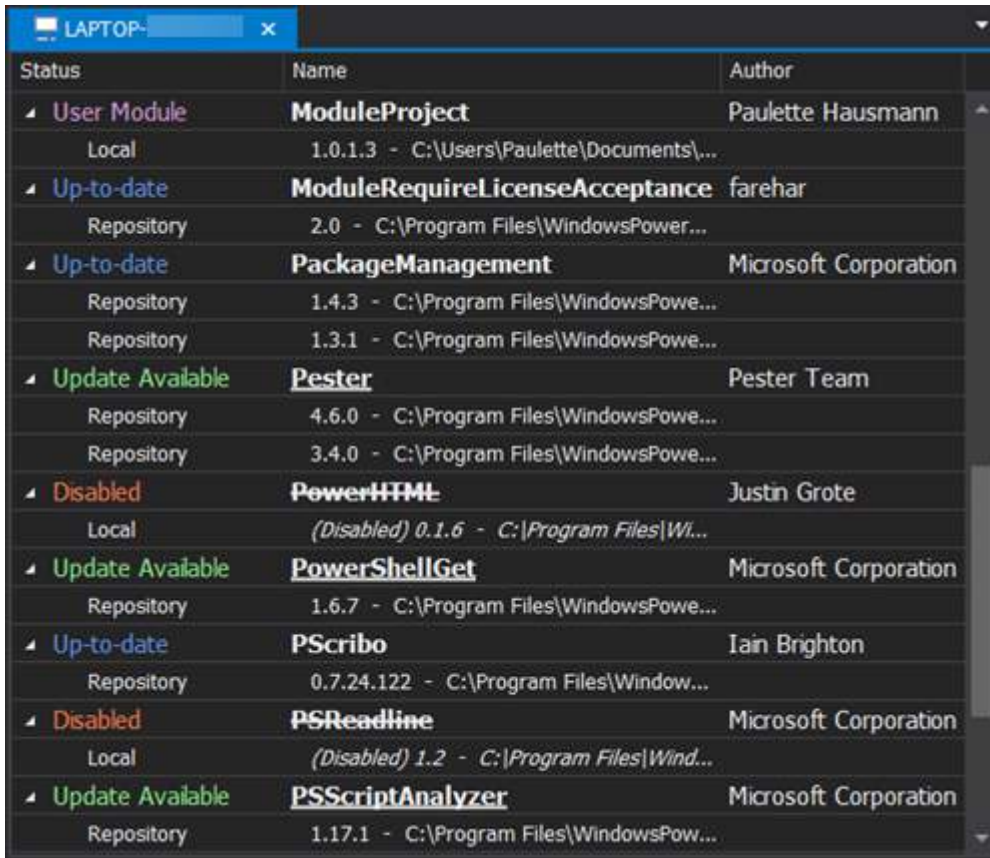


How to view all installed versions of all modules

To display all versions of all installed modules

- Click the **Expand All** / **Collapse All** buttons on the ribbon (Home > Window) to view/hide information for all installed modules:





Status	Name	Author
User Module	ModuleProject	Paulette Hausmann
Local	1.0.1.3 - C:\Users\Paulette\Documents\...	
Up-to-date	ModuleRequireLicenseAcceptance	farehar
Repository	2.0 - C:\Program Files\WindowsPower...	
Up-to-date	PackageManagement	Microsoft Corporation
Repository	1.4.3 - C:\Program Files\WindowsPowe...	
Repository	1.3.1 - C:\Program Files\WindowsPowe...	
Update Available	Pester	Pester Team
Repository	4.6.0 - C:\Program Files\WindowsPowe...	
Repository	3.4.0 - C:\Program Files\WindowsPowe...	
Disabled	PowerHTML	Justin Grote
Local	(Disabled) 0.1.6 - C:\Program Files\Wi...	
Update Available	PowerShellGet	Microsoft Corporation
Repository	1.6.7 - C:\Program Files\WindowsPowe...	
Up-to-date	PScribo	Iain Brighton
Repository	0.7.24.122 - C:\Program Files\Window...	
Disabled	PSReadline	Microsoft Corporation
Local	(Disabled) 1.2 - C:\Program Files\Wind...	
Update Available	PSScriptAnalyzer	Microsoft Corporation
Repository	1.17.1 - C:\Program Files\WindowsPow...	

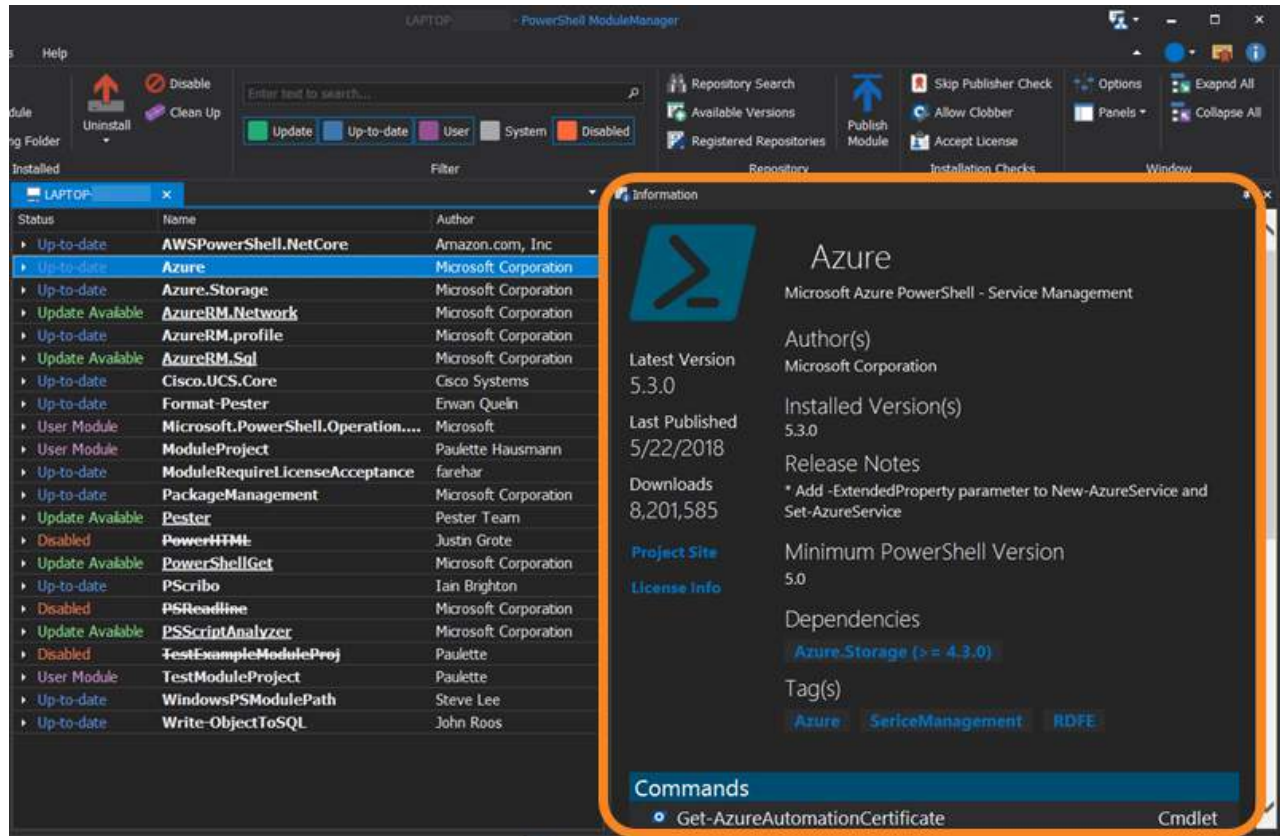
Machine tab - Expand All

4.3.2 Viewing Module Details

PowerShell ModuleManager displays details for installed modules in the **Information** panel.


Information Panel

The Information panel is in the middle of the PowerShell ModuleManager workspace, on the right side. Selecting a module in the machine tab will display the module details in the Information panel:



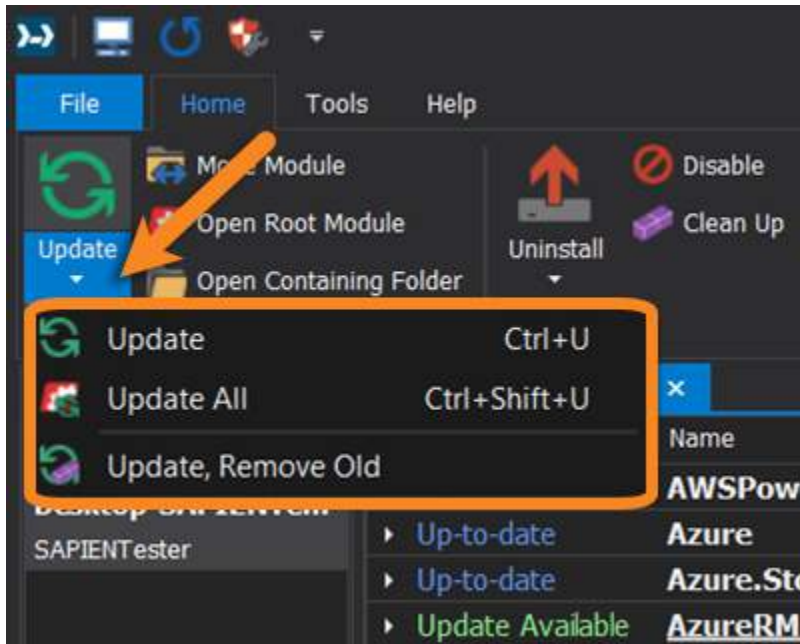
Information Panel displaying highlighted module details

Information panel contents

Description	A short description of the module.
Latest Version	The latest version of the module available in the repository.
Last Published	The last date the module was updated in the repository.
Installed Version(s)	The version(s) of the module installed on your machine.
Minimum PowerShell Version	The minimum PowerShell version required to run the module.
Dependencies	If dependent module(s) are listed, the dependent module version(s) will also be installed.
Tags	Keywords used to search for modules in PowerShell repositories.  Click on a Tag to open a search for modules with the same tag.
Commands	Cmdlets, functions, or aliases contained in the module.

4.3.3 Updating Modules

PowerShell ModuleManager provides a number of options for updating installed modules:



Home ribbon > Update menu

▲ About the PowerShell ModuleManager 'update' function

When performing a module *update*, PowerShell ModuleManager will check the set repository to see if a version is available that is newer than the installed version:

- If a newer version is available, it will be installed.
- If a newer version is not available, the details in the [Information panel](#)⁵¹ will be updated with any new information.

Set Repository

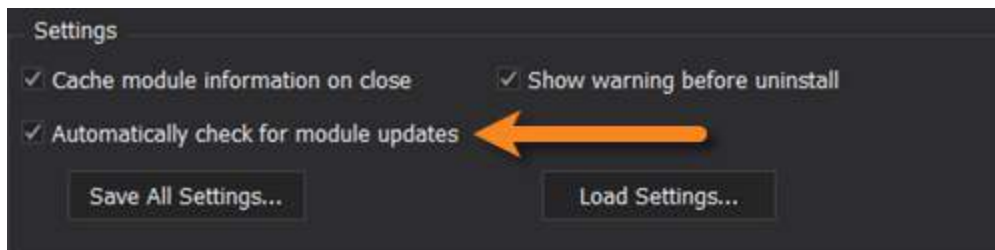
The default registered repository is the PowerShell Gallery (PSGallery). This can be changed in **File > Options > General > PowerShell**:



Options > General > Powershell

Automatic Updates Check

During program startup, by default PowerShell ModuleManager automatically checks for updates by comparing the installed modules to the modules in the set repository. This can be changed in **File > Options > General > Settings**:

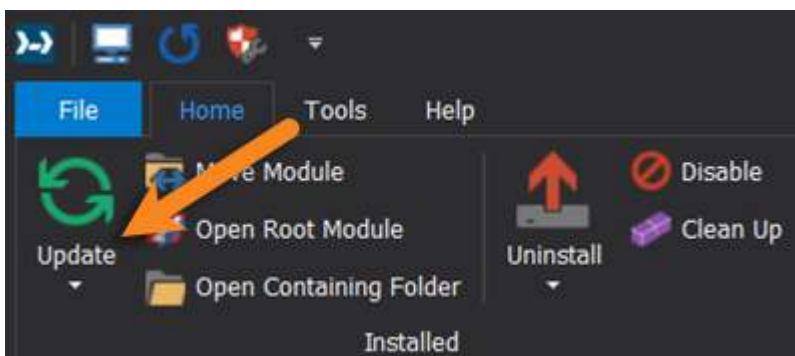


i To learn about other options available when updating modules, view [Module Installation Options](#)⁴³.

How to update a module

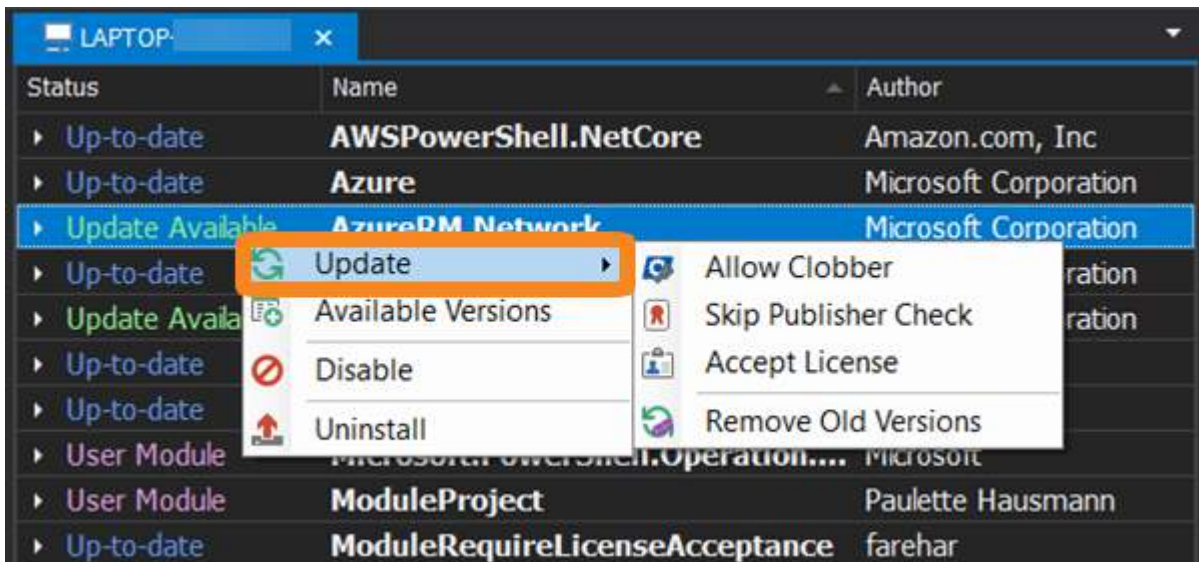
To update a module

- Select a module in the Machine tab, and then on the Home ribbon click **Update** (*Ctrl+U*):



-OR-

- Right-click a module in the Machine tab, and select **Update** from the context menu:

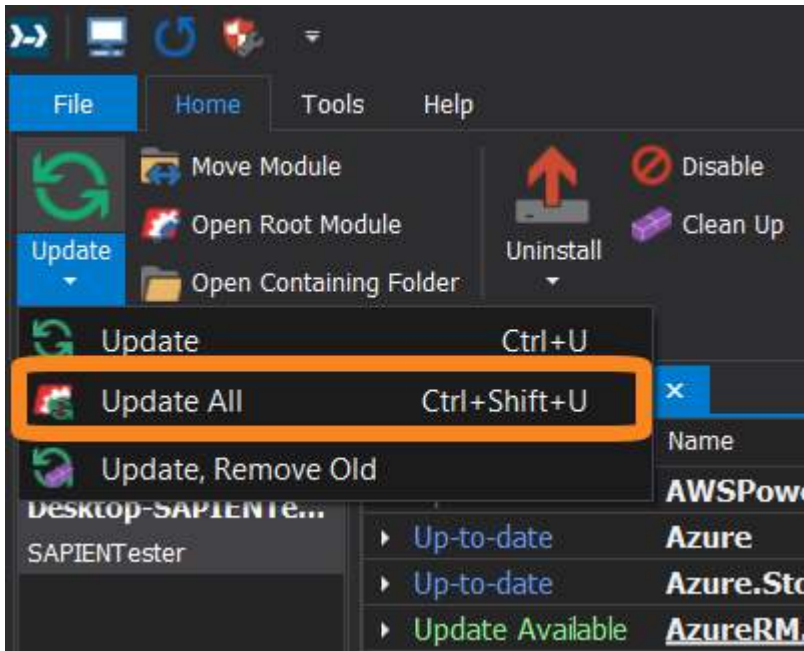


- i** You can also select an [Installation Option](#)⁴³ when updating a module from the Machine tab context menu.

How to update ALL modules

To update ALL modules

- In the Update drop-down list on the Home ribbon, select **Update All** (*Ctrl+Shift+U*):



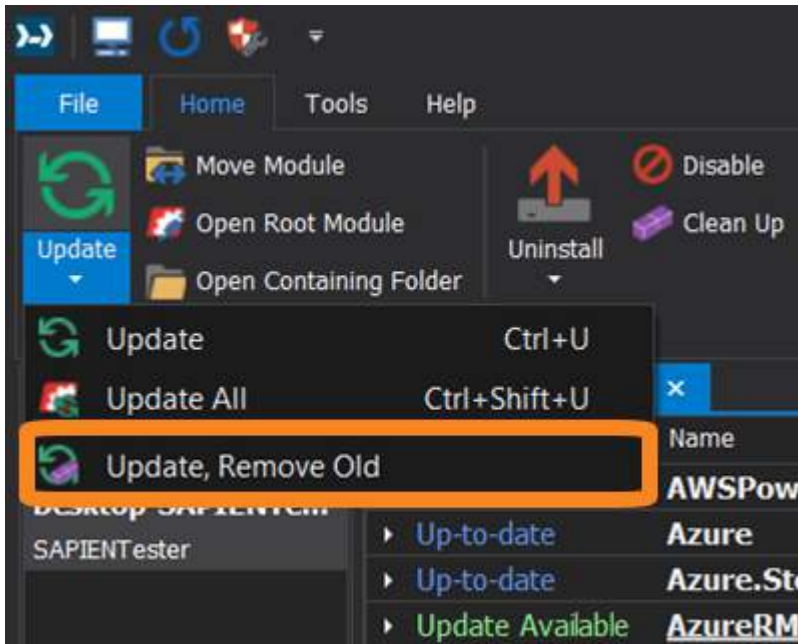
The most recent version of all modules will be installed.

- i The update time will vary depending on how many modules you have installed.

How to update a module and remove old versions

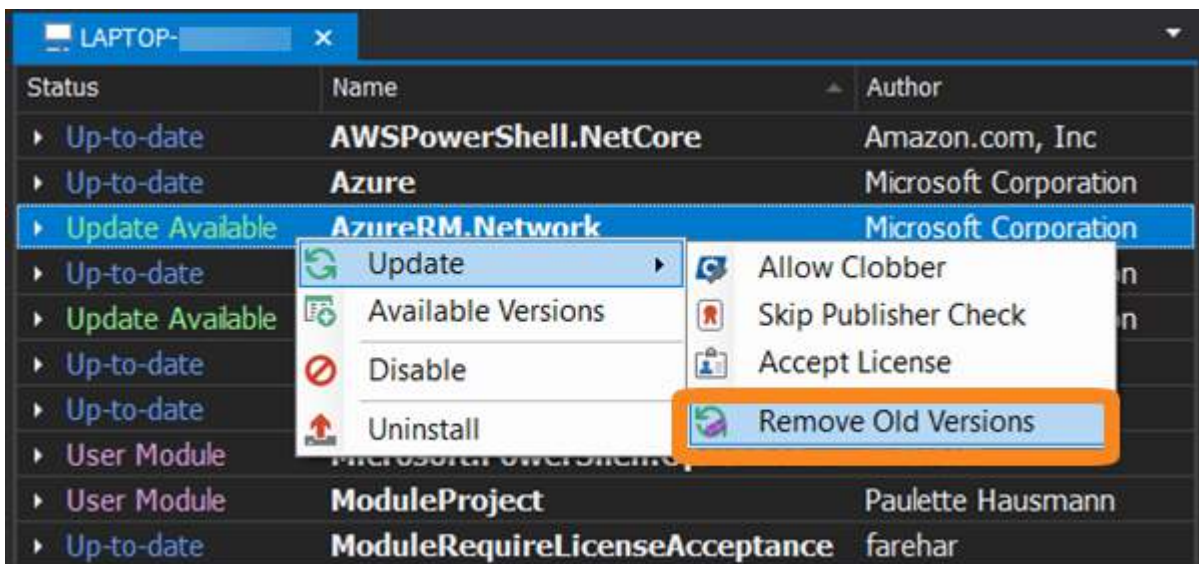
To update a module and remove old versions

- Select a module in the Machine tab, then in the Update drop-down list on the Home ribbon, select **Update, Remove Old**:



-OR-

- Right-click a module in the Machine tab, then select **Update > Remove Old Versions** from the context menu:



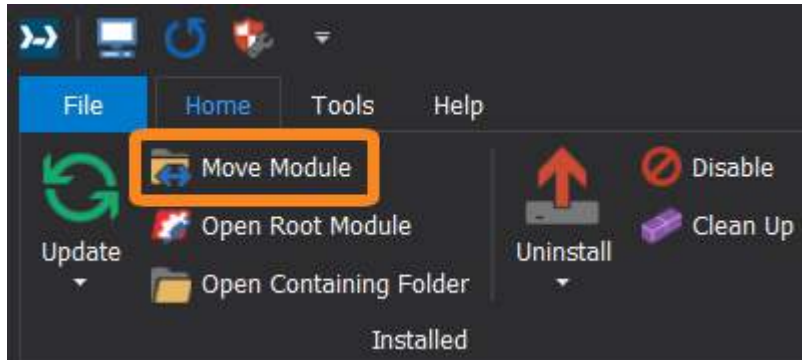
The latest version of the selected module will be installed, and all previous versions will be uninstalled.

4.3.4 Moving Modules

The default location when installing or updating modules is defined in the installation scope. You can also easily move a module after it has been installed. See [Installation Scope](#)^[44] for more details.

Move Module

The Move Module control on the Home ribbon allows you to move a module to a different location:

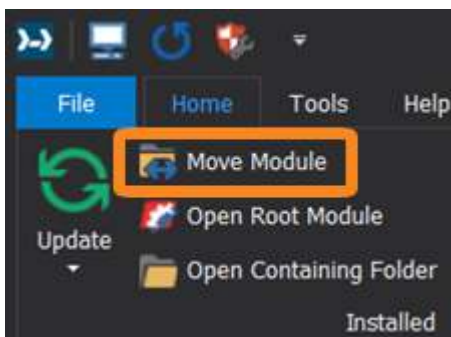


Home ribbon > Move Module

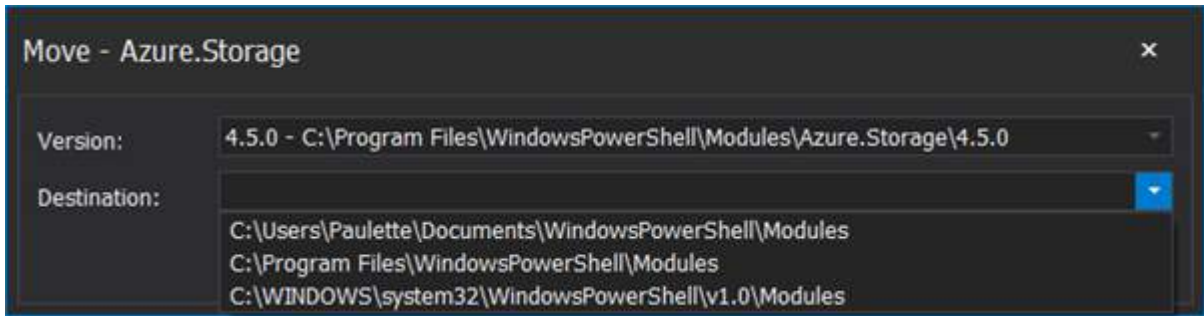
How to move a module

To move a module

1. Select a module in the Machine tab, then on the Home ribbon click **Move Module**:

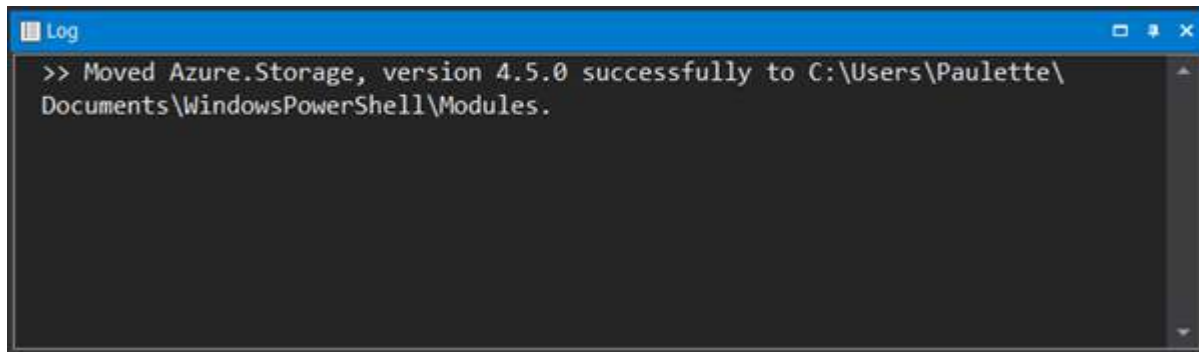


2. In the Move dialog:
 - a. Select a module **version**.
 - b. Select a **destination** folder.
 - c. Click **OK**.

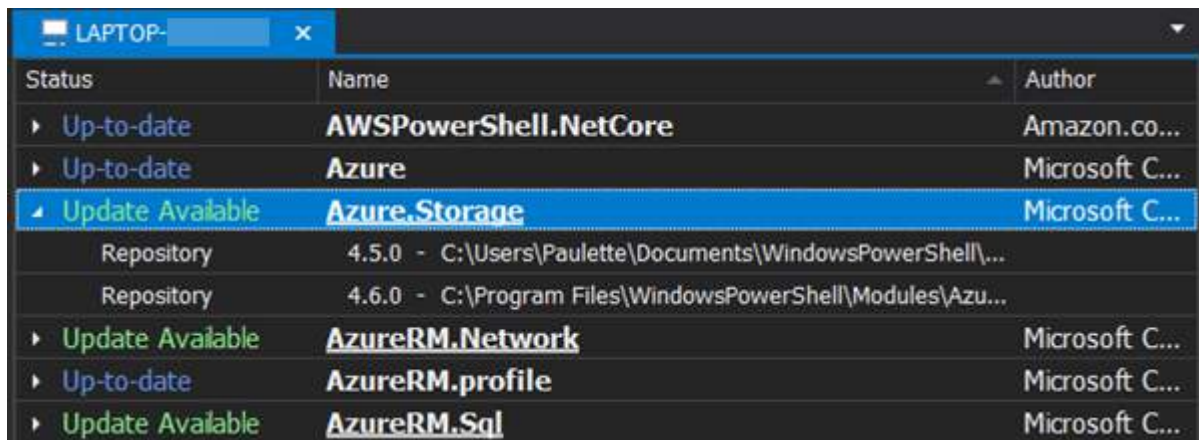


Move Dialog

The move activity will appear in the Log panel:



The location change will be updated in the Machine tab:



4.3.5 Publishing Modules

PowerShell ModuleManager allows you to share your PowerShell modules by publishing to online repositories like the PowerShell Gallery.

Take a moment to familiarize yourself with the version selection behavior of the Publish-Module command, and also the publishing prerequisites, and then learn how to publish modules.

Understanding the Publish-Module command's version selection behavior

The **Publish-Module** command—from the PowerShellGet module—allows you to share modules from your local computer to online or local galleries.

To run the Publish-Module command, you can either specify the module name or the path of the folder containing the module. When publishing by name, the first module found in the paths of the PSModulePath environment variable (\$env:PSModulePath) will be published—so if you have multiple versions, it is important to specify a minimum version.

Publishing prerequisites

- **PowerShell ModuleManager version**


You will need PowerShell ModuleManager version 2019 1.1.1 or later to use the publish functionality.

- **Registered repository**

The repository you want to publish to must be a [registered repository](#)^[84] in PowerShell.

- **Online repositories**

An API key is required to publish to NuGet-based online repositories, such as the PSGallery. To get an API key, you will need to create an account on the PowerShell Gallery and generate a new key for the module package.

 Local repositories do not require a key.

- **Windows 10 - PowerShellGet version**

Windows 10 includes an older version of PowerShellGet, which provides the [Publish-Module command](#)^[60]. If you are running Windows 10 with the default PowerShellGet module, you will need to update the module to recognize the API key from the latest gallery site.

- **Metadata**

Publishing a module requires metadata, most of which is taken from the module manifest (.psd1).

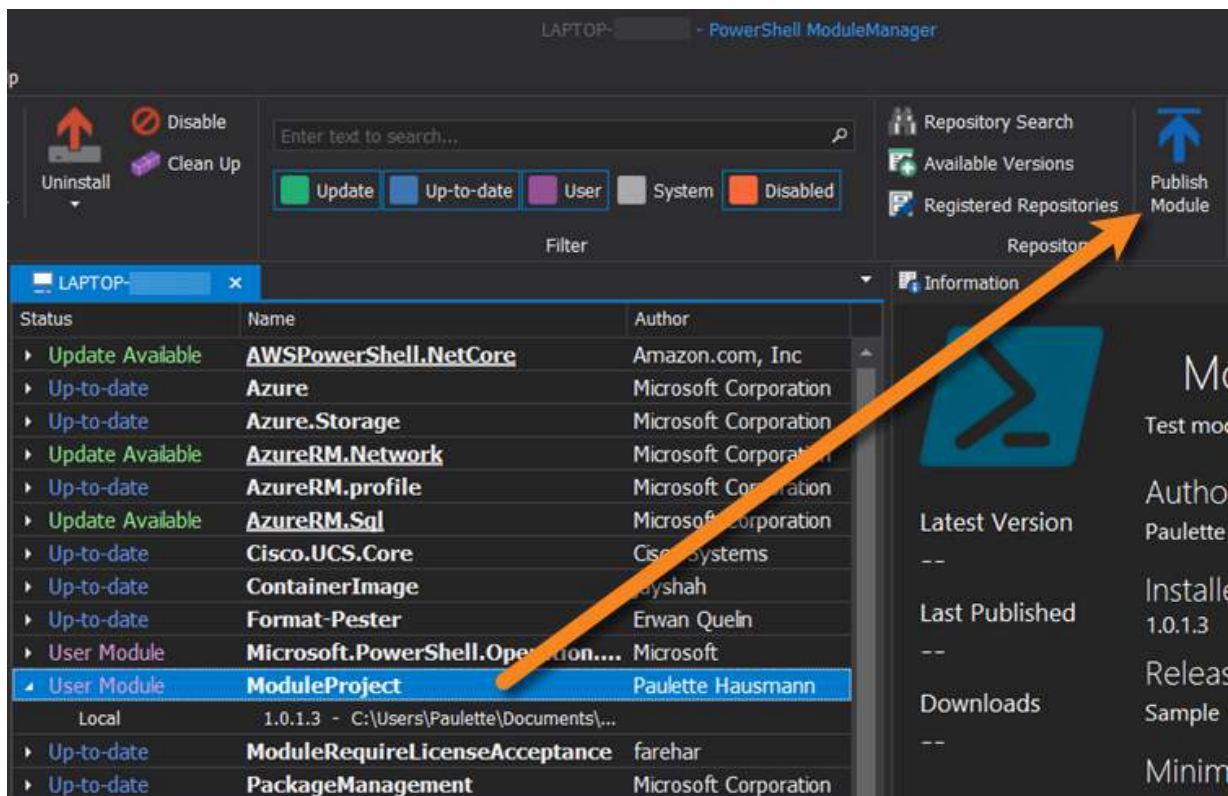
Metadata requirements:

Name:	A unique module name that does not already exist in the repository you intend to publish to.
Version:	The current version of the module, which must be higher than any existing module versions present on the repository.
Description:	A short description of the module.
Author:	The creator of the module.

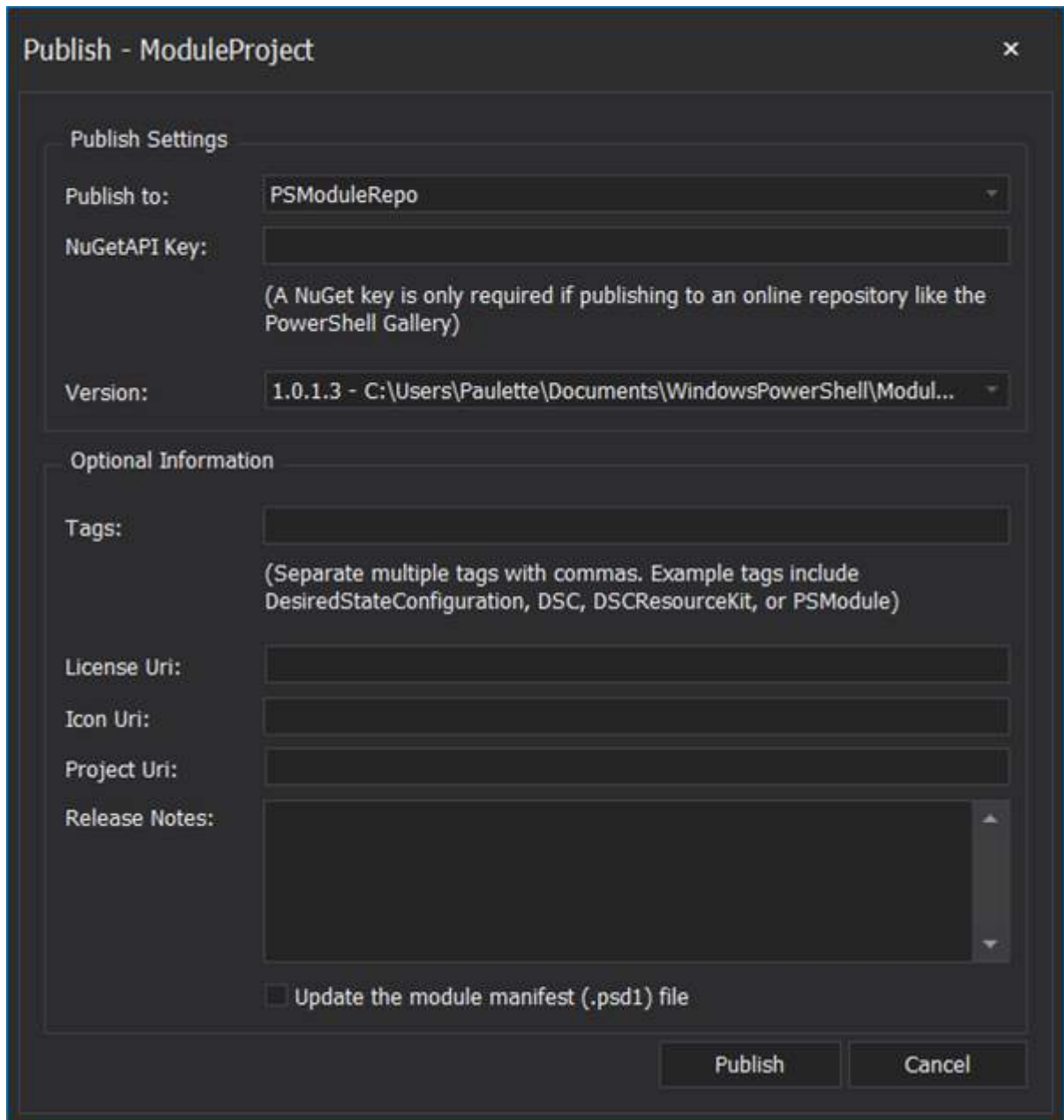
How to publish a module

To publish a module

1. Select a module in the Machine tab, then on the Home ribbon click **Publish Module**:



2. Enter the required data in the Publish dialog, and then click **Publish**:



The image shows a 'Publish - ModuleProject' dialog box with a dark theme. It is divided into two main sections: 'Publish Settings' and 'Optional Information'. The 'Publish Settings' section includes a 'Publish to:' dropdown menu set to 'PSModuleRepo', an empty 'NuGetAPI Key:' text box, and a 'Version:' dropdown menu showing '1.0.1.3 - C:\Users\Paulette\Documents\WindowsPowerShell\Modul...'. Below these is a note: '(A NuGet key is only required if publishing to an online repository like the PowerShell Gallery)'. The 'Optional Information' section contains a 'Tags:' text box, a 'License Uri:' text box, an 'Icon Uri:' text box, a 'Project Uri:' text box, and a 'Release Notes:' text area. Below the 'Release Notes' is a checkbox labeled 'Update the module manifest (.psd1) file'. At the bottom right are 'Publish' and 'Cancel' buttons.

Publish - ModuleProject

Publish Settings

Publish to: PSModuleRepo

NuGetAPI Key:

(A NuGet key is only required if publishing to an online repository like the PowerShell Gallery)

Version: 1.0.1.3 - C:\Users\Paulette\Documents\WindowsPowerShell\Modul...

Optional Information

Tags:

(Separate multiple tags with commas. Example tags include DesiredStateConfiguration, DSC, DSCResourceKit, or PSModule)

License Uri:

Icon Uri:

Project Uri:

Release Notes:

☐ Update the module manifest (.psd1) file

Publish Cancel

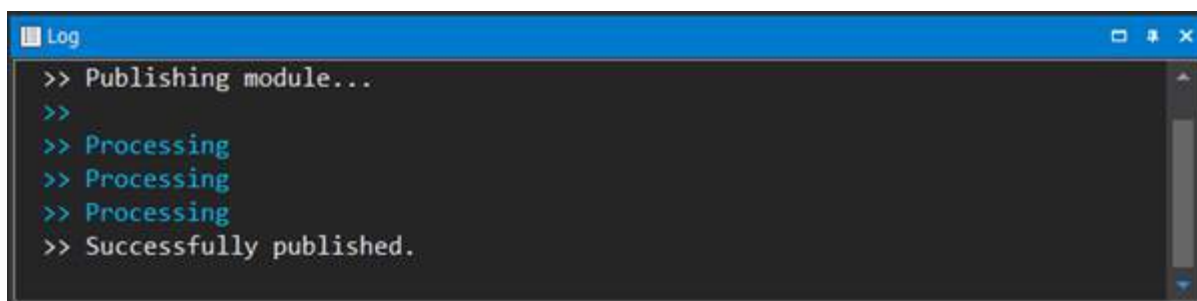
Publish Dialog

Publish <module> - Field Definitions:


Publish to*:	Select the registered repository from the drop-down list.
NuGetAPI Key:	If you are publishing to an online repository like the PowerShell Gallery, enter the NuGetAPI key.
Version*:	Select the module version.
Tags:	Keywords used to search for modules in PowerShell repositories. Separate multiple tags with commas.
License Uri:	The URL of licensing terms for the module.
Icon Uri:	The URL of an icon for the module.
Project Uri:	The URL of a website for the project.
Release Notes:	Comments about this version of the module.
Update the module manifest (.psd1) file:	<ul style="list-style-type: none">• If the data in the Optional Information section already exists in the module manifest, you do not need to enter it in the Publish dialog.• If you specify data in any of the Optional Information fields that is not already in the module manifest, you must select the <i>Update the module manifest (.psd1) file</i> checkbox to update the local module's manifest.

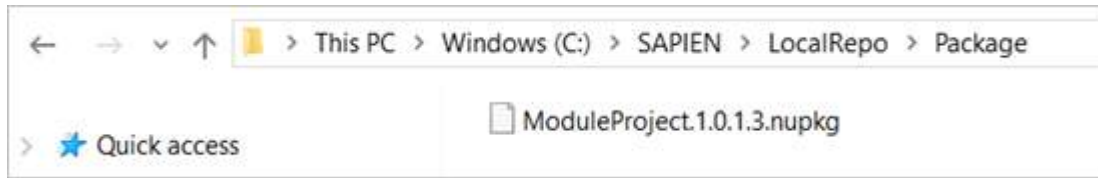
* Required fields

The publishing activity will appear in the Log panel:

A screenshot of a 'Log' window with a blue title bar. The log text is as follows:

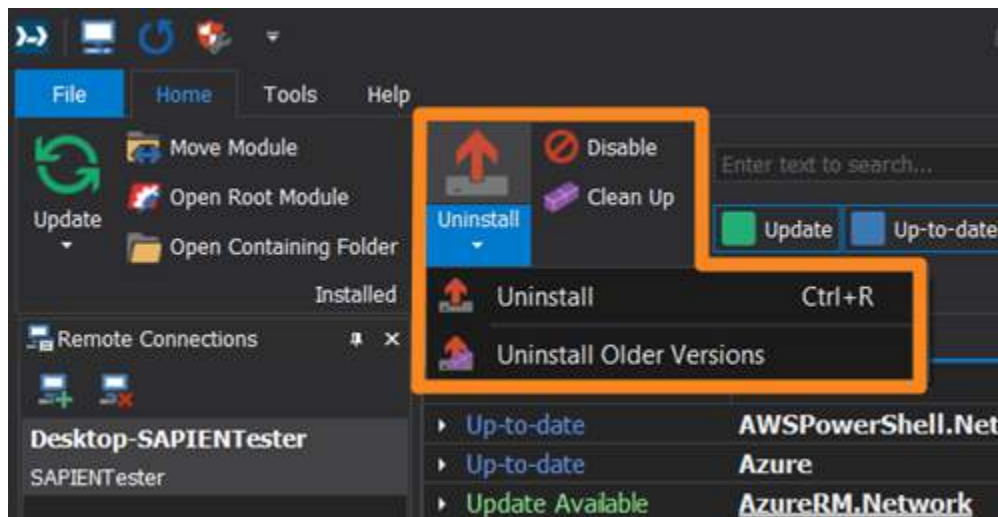
```
>> Publishing module...
>>
>> Processing
>> Processing
>> Processing
>> Successfully published.
```

 A .nupkg file is created when a module is published to a registered repository, allowing you to find the published module when performing a [search](#) 



4.3.6 Disabling and Uninstalling Modules

PowerShell ModuleManager provides a number of options for disabling and uninstalling modules.



Home ribbon > Uninstall and Disable options

In Windows PowerShell 5.0, you can have multiple versions of the same module on the same machine, and you can even have multiple versions of the same module imported into the same session. Understanding the difference between [disabling](#)^[66] and [uninstalling](#)^[74]—and specifically how to disable individual module versions—will allow you to control which module PowerShell will load into your scripts.

In addition to becoming familiar with PowerShell ModuleManager's [disable](#)^[66] and [uninstall](#)^[74] features, we recommend that you take a moment to review [PowerShell's module import behavior](#)^[65].

Understanding PowerShell's module import behavior

Beginning in Windows PowerShell 3.0, modules are imported automatically when any cmdlet or function in the module is used in a command. When importing a module into a session, PowerShell does not always import the latest version of a module; instead, it searches for the specified module in the paths defined in the value of the environmental variable, `PSModulePath`. For example, when the first path in `PSModulePath` is `$home\Documents\WindowsPowerShell\Modules`, and any version

of the module is installed in that directory, PowerShell then searches for module files in the following order:

- Module Manifest files (.psd1)
- Script Module files (.psm1)
- Binary Module files (.dll)

The **Import-Module** cmdlet has parameters to help explicitly import modules—such as `RequiredVersion`, `MinimumVersion`, and `MaximumVersion`—but it will import the first module found that meets the specified criteria.


Disabling Modules

Disabling specific versions of a module allows you to control which module PowerShell will load into your script. Disabling is also helpful when you are unsure if other modules are dependent on a particular module because disabling avoids having to reinstall the module. If it is later determined that the module is needed, simply enable it.

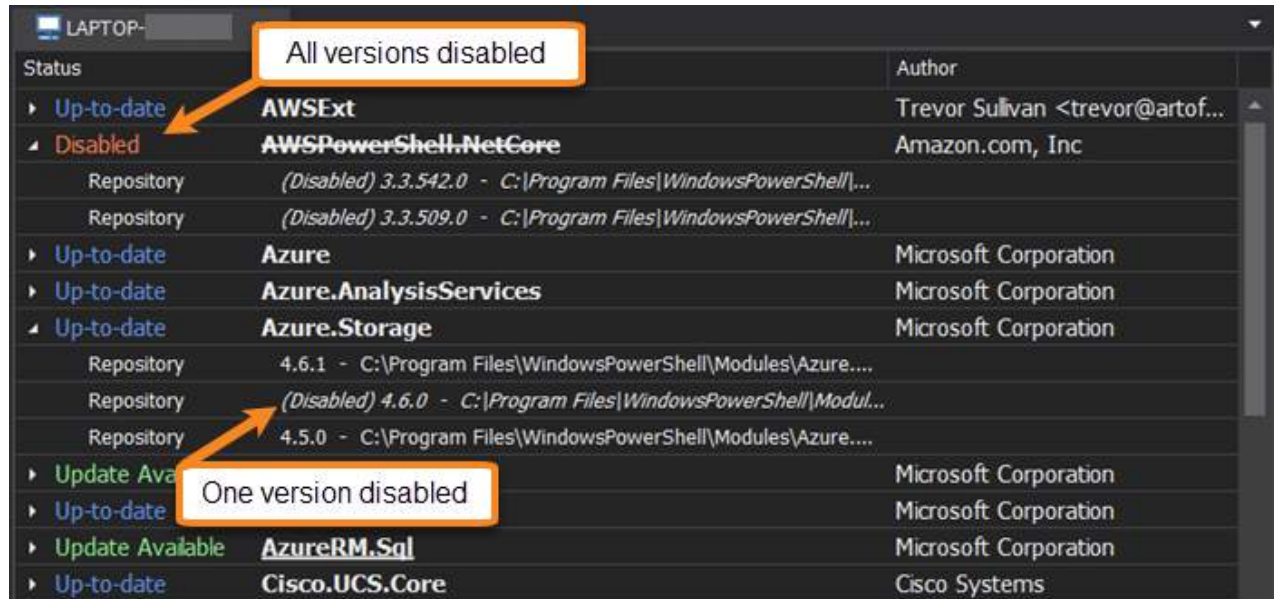
Understanding how the Disable feature works

The PowerShell ModuleManager **Disable** feature moves a module from the path saved in the value of the `PSModulePath` environment variable (`$env:PSModulePath`). Disabling a module is not the same as *uninstalling*. Disabling a module moves it to a user's AppData folder, and enabling moves it back to the previous location. If modules are no longer needed, they should be uninstalled.

Viewing disabled modules in the Machine tab

Disabled modules are displayed in the Machine tab or the [Disabled Modules panel](#)  ¹²⁹.

In the Machine tab, disabled modules appear with distinct characteristics:

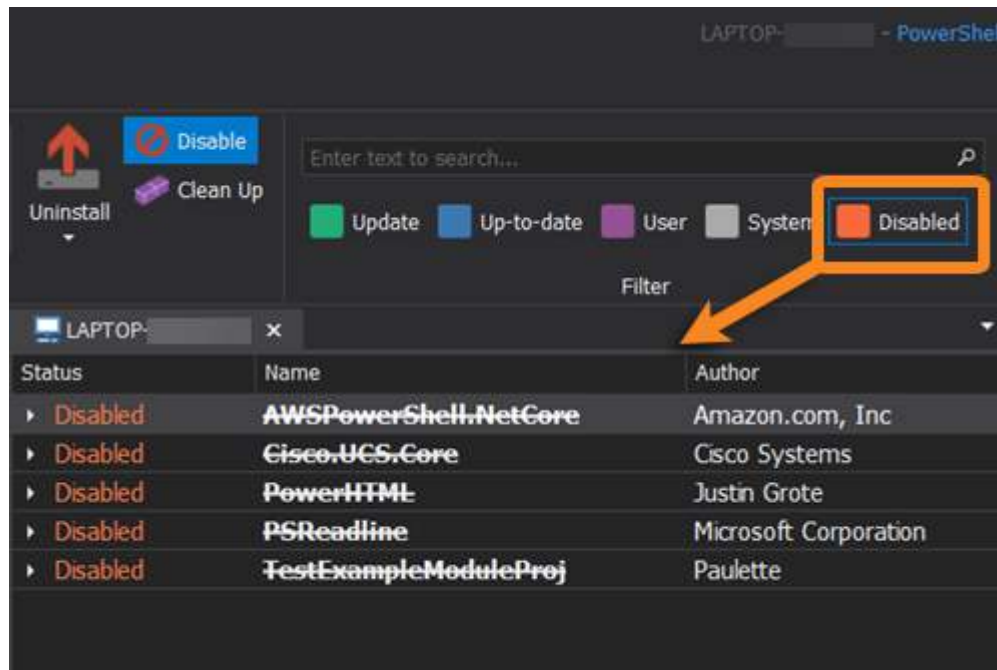


Status	Name	Author
Up-to-date	AWSExt	Trevor Sullivan <trevor@artof...
Disabled	AWSPowerShell.NetCore	Amazon.com, Inc
Repository	(Disabled) 3.3.542.0 - C:\Program Files\WindowsPowerShell\...	
Repository	(Disabled) 3.3.509.0 - C:\Program Files\WindowsPowerShell\...	
Up-to-date	Azure	Microsoft Corporation
Up-to-date	Azure.AnalysisServices	Microsoft Corporation
Up-to-date	Azure.Storage	Microsoft Corporation
Repository	4.6.1 - C:\Program Files\WindowsPowerShell\Modules\Azure....	
Repository	(Disabled) 4.6.0 - C:\Program Files\WindowsPowerShell\Modul...	
Repository	4.5.0 - C:\Program Files\WindowsPowerShell\Modules\Azure....	
Update Available	AzureRM.Sql	Microsoft Corporation
Up-to-date	Cisco.UCS.Core	Cisco Systems

i When all versions of a module are disabled, the parent node will display as **Disabled**.

To see only disabled modules in the machine tab

- Select only the **Disabled** filter on the Home ribbon:



👍 Disabled modules can also be viewed and managed from the [Disabled Modules panel](#) ¹²⁹.

For more information on locating and viewing installed modules, see [Filter, Sort and Search](#) ⁴⁵.

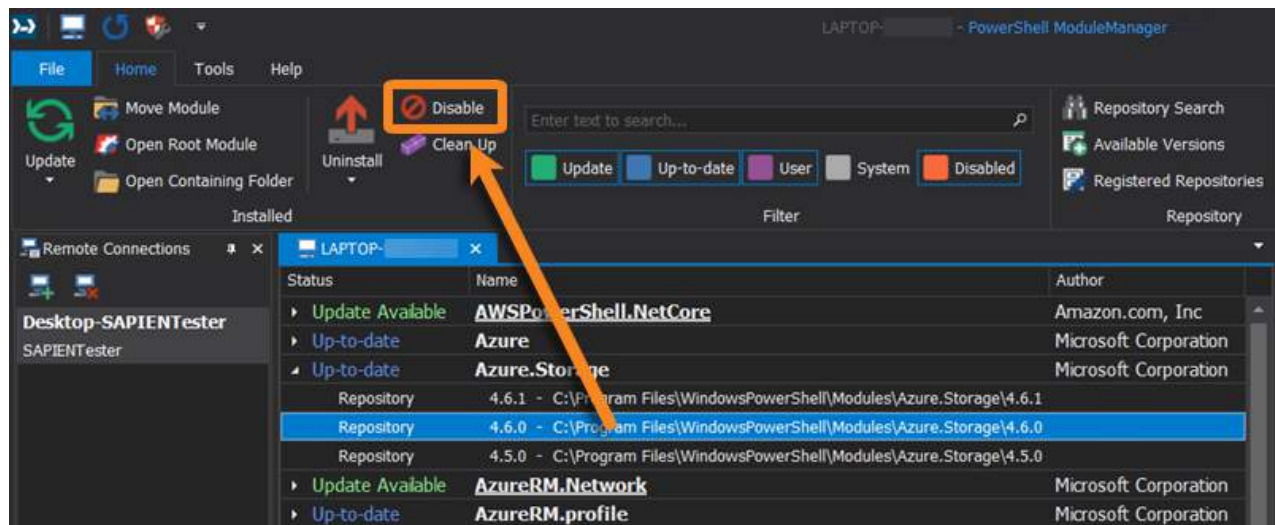
How to disable and enable modules

Disabling Modules

You can disable modules from the Home ribbon, or the context menu.

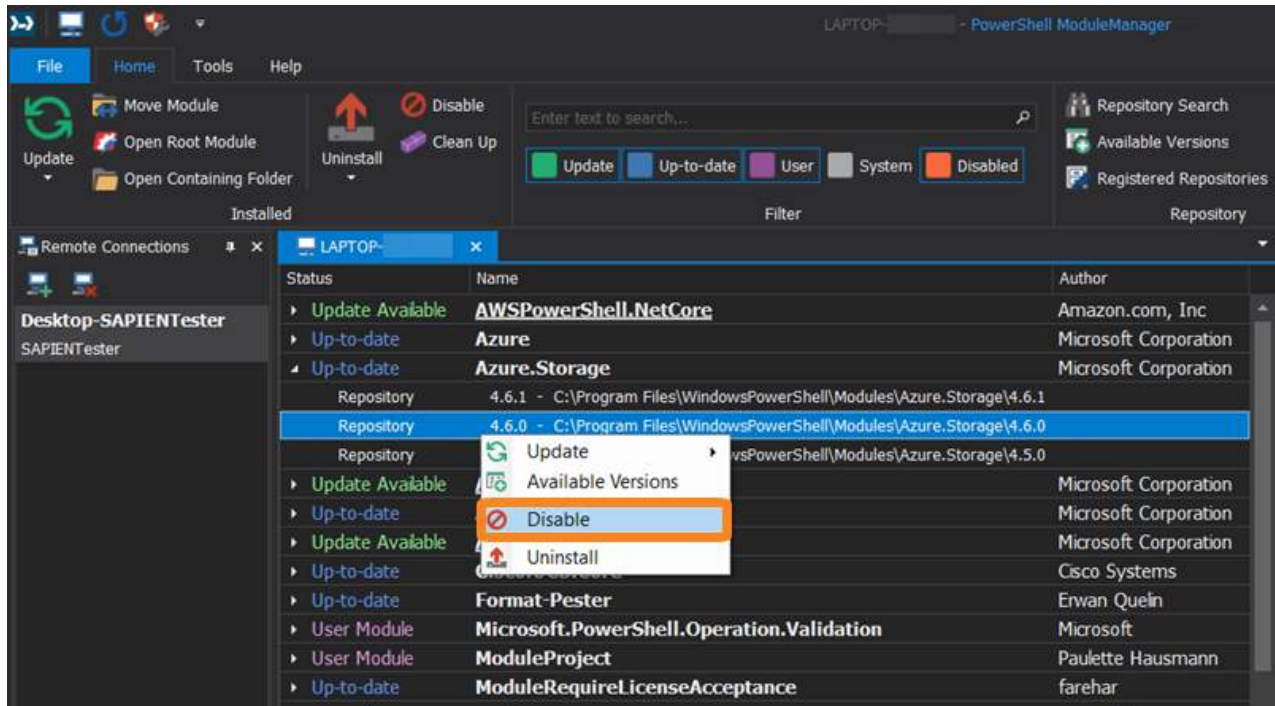
To disable a module

- Select a module, and then click **Disable** on the Home ribbon:



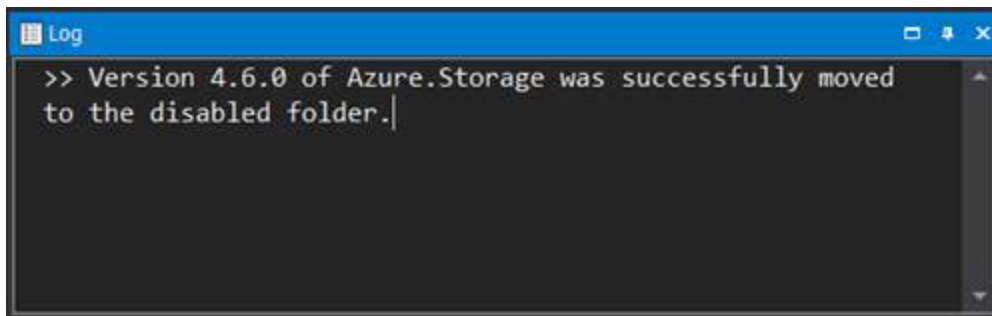
-OR-

- Right-click on a module, and then select **Disable** from the context menu:



👍 Selecting the parent node disables all versions; selecting a child node disables only that version. You can also *Ctrl-click* to select more than one module.

The disabled status is displayed in the Log panel:



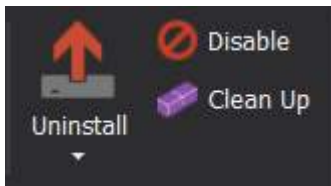
The Machine tab will refresh to display the disabled module:

Status	Name	Author
Update Available	AWSPowerShell.NetCore	Amazon.com, Inc
Up-to-date	Azure	Microsoft Corporation
Up-to-date	Azure.Storage	Microsoft Corporation
Repository	4.6.1 - C:\Program Files\WindowsPowerShell\Modules\Azure.Storage\4.6.1	
Repository	(Disabled) 4.6.0 - C:\Program Files\WindowsPowerShell\Modules\Azure.S...	
Repository	4.5.0 - C:\Program Files\WindowsPowerShell\Modules\Azure.Storage\4.5.0	
Update Available	AzureRM.Network	Microsoft Corporation
Up-to-date	AzureRM.profile	Microsoft Corporation

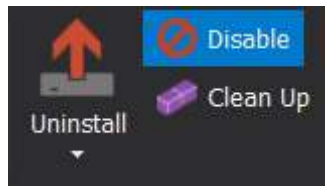
Enabling Modules

The **Disable** feature on both the Home ribbon and the Machine tab context menu behaves like a toggle. When a module is disabled, the **Disable** option is highlighted. Clicking the highlighted **Disable** option enables the module by moving it back to its original location, and the button is no longer highlighted:

Disable / Enable on the ribbon



Module can be Disabled



Module can be Enabled

Disable / Enable in the context menu



Module can be Disabled

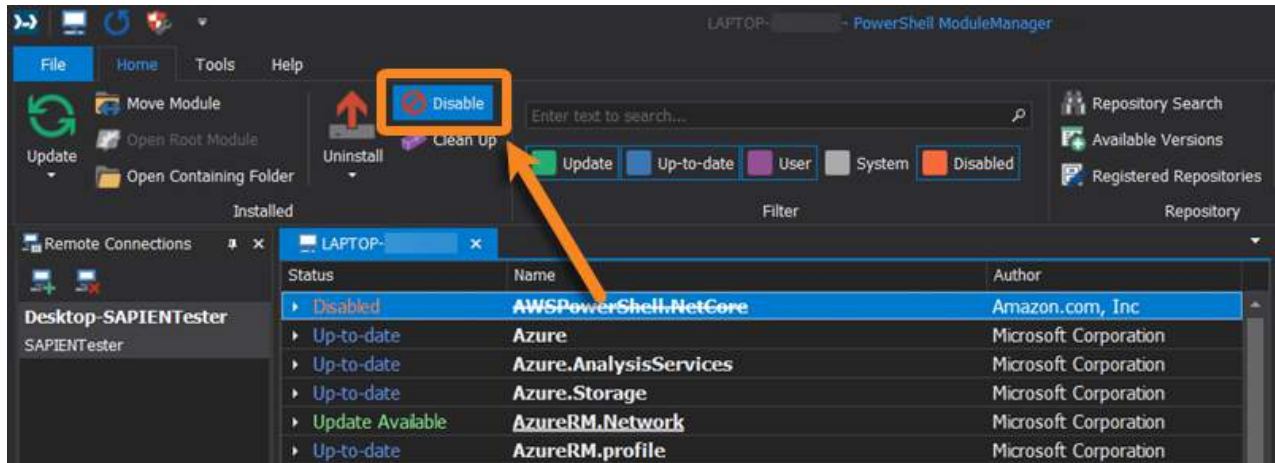


Module can be Enabled

To enable a module

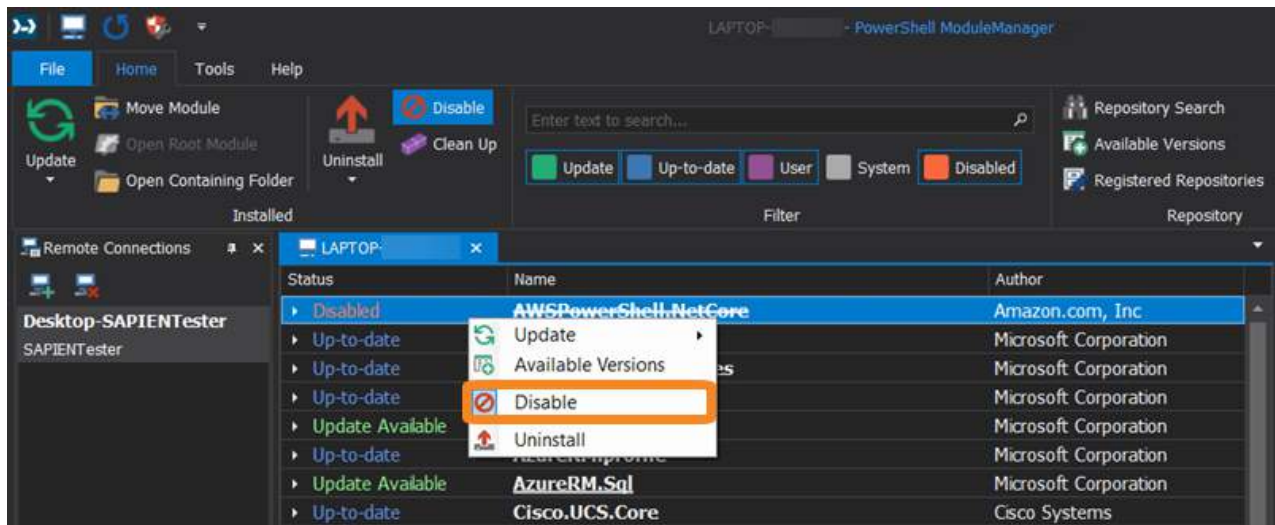
You can enable modules from the Home ribbon, or the context menu.

- Select a disabled module, and then click **Disable** (highlighted) on the Home ribbon:



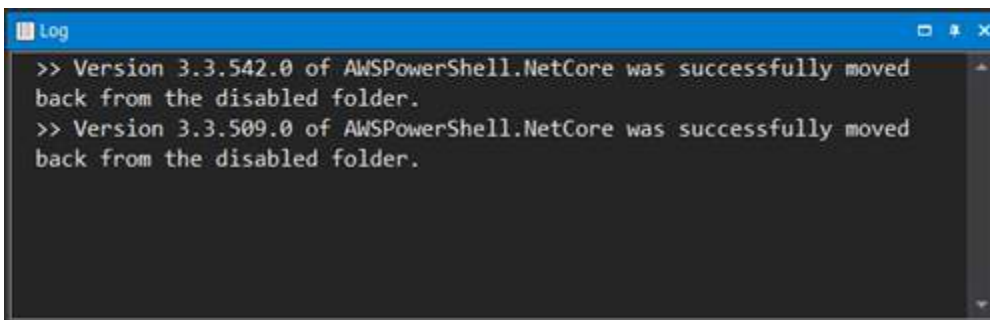
-OR-

- Right-click on a disabled module, and then select **Disable** (highlighted) from the context menu:



👉 Selecting the parent node enables all versions; selecting a child node enables only that version. You can also *Ctrl-click* to select more than one module.

The enabled status is displayed in the Log panel:



The Machine tab will refresh to display the enabled module(s):

Status	Name	Author
Update Available	AWSPowerShell.NetCore	Amazon.com, Inc
Repository	3.3.542.0 - C:\Program Files\WindowsPowerShell\Modules\...	
Repository	3.3.509.0 - C:\Program Files\WindowsPowerShell\Modules\...	
Up-to-date	Azure	Microsoft Corporation
Up-to-date	Azure.AnalysisServices	Microsoft Corporation
Up-to-date	Azure.Storage	Microsoft Corporation
Update Available	AzureRM.Network	Microsoft Corporation
Up-to-date	AzureRM.profile	Microsoft Corporation

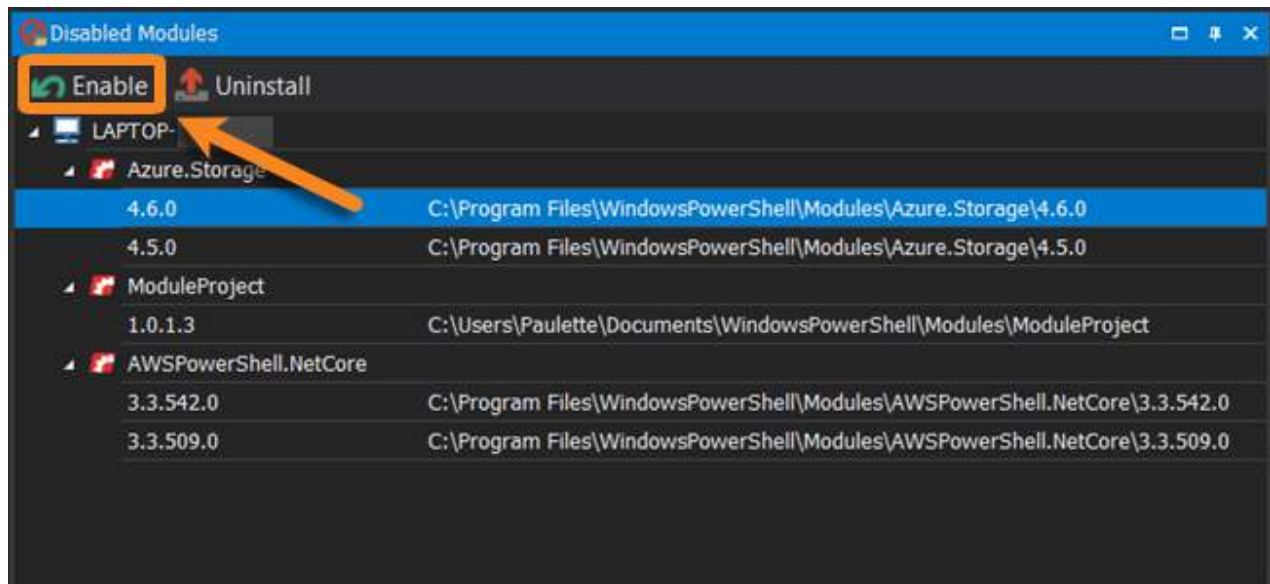
Modules can also be enabled or uninstalled from the [Disabled Modules panel](#).

How to enable modules from the Disabled Modules panel

Modules can be enabled or [uninstalled](#) from the Disabled Modules panel.

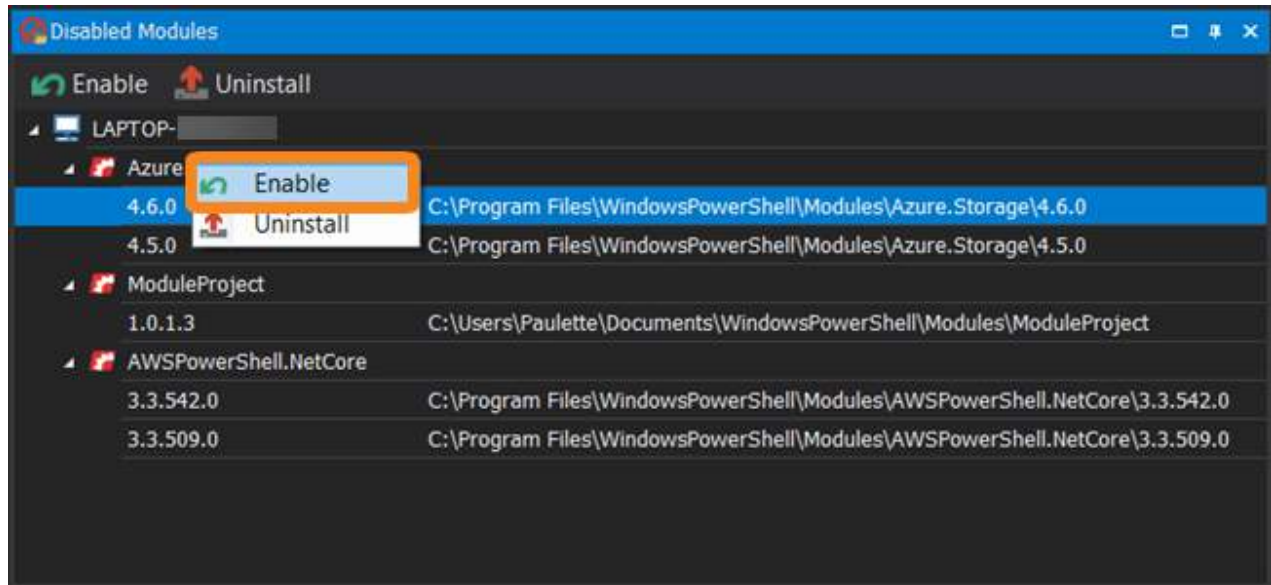
To enable a module

- Select a module, and then click the **Enable** button:



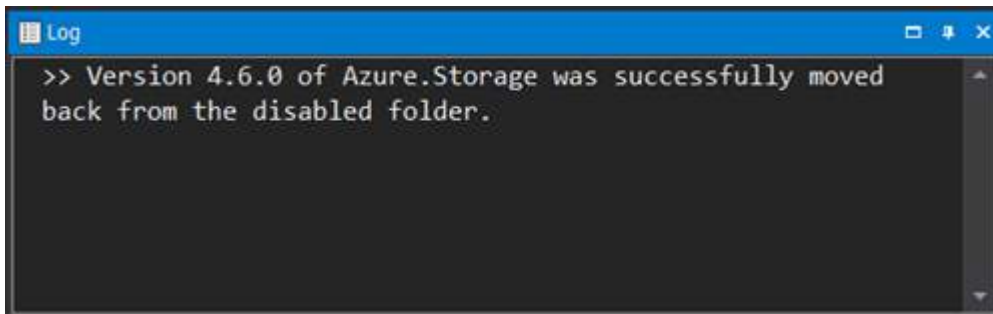
-OR-

- Right-click on a module, and then select **Enable** from the context menu:



👍 Selecting the parent node enables all versions; selecting a child node enables only that version.

The enabled status is displayed in the Log panel:



The Machine tab will refresh to display the enabled module(s):

Status	Name	Author
Disabled	AWSPowerShell.NetCore	Amazon.com, Inc
Up-to-date	Azure	Microsoft Corporation
Up-to-date	Azure.AnalysisServices	Microsoft Corporation
Up-to-date	Azure.Storage	Microsoft Corporation
Repository	4.6.1 - C:\Program Files\WindowsPowerShell\Modules\Azure.Stora...	
Repository	4.6.0 - C:\Program Files\WindowsPowerShell\Modules\Azure.Stora...	
Repository	(Disabled) 4.5.0 - C:\Program Files\WindowsPowerShell\Modules\A...	
Update Available	AzureRM.Network	Microsoft Corporation
Up-to-date	AzureRM.profile	Microsoft Corporation

Uninstalling Modules

Uninstall removes modules from your machine. You can remove one or more versions of a module, remove all older versions, or remove all versions.

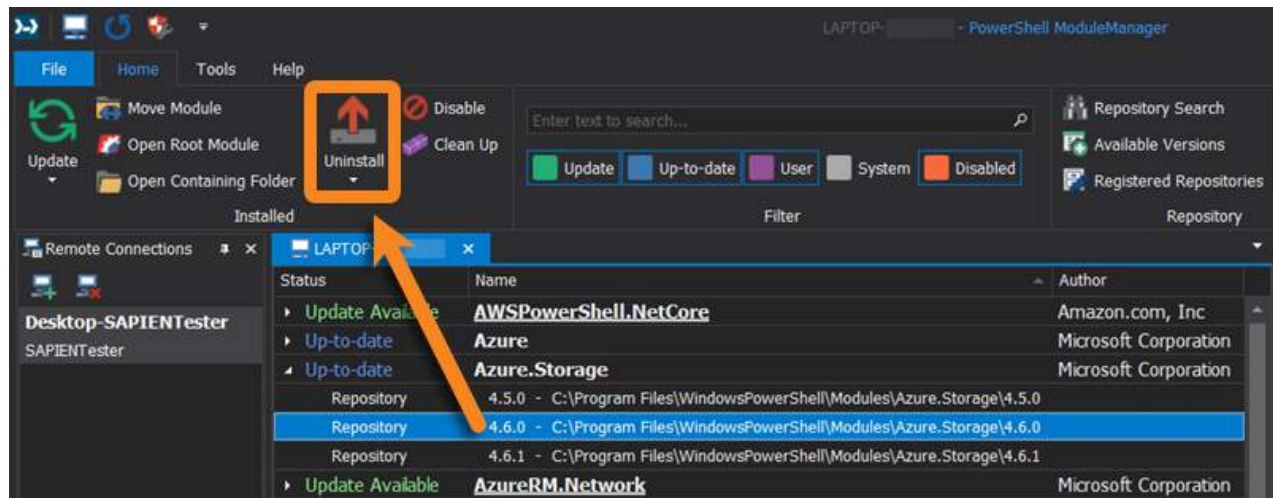
i Modules are uninstalled by the **Uninstall-Module** cmdlet, but if they fail to uninstall—or if they were not installed with the **Install-Module**—they are moved to the Recycle Bin.

👍 A good module management practice is to uninstall modules when they are no longer needed. If you are not sure if you want to uninstall a particular module version, then [disable](#) it.

How to uninstall a module version

To uninstall a module version

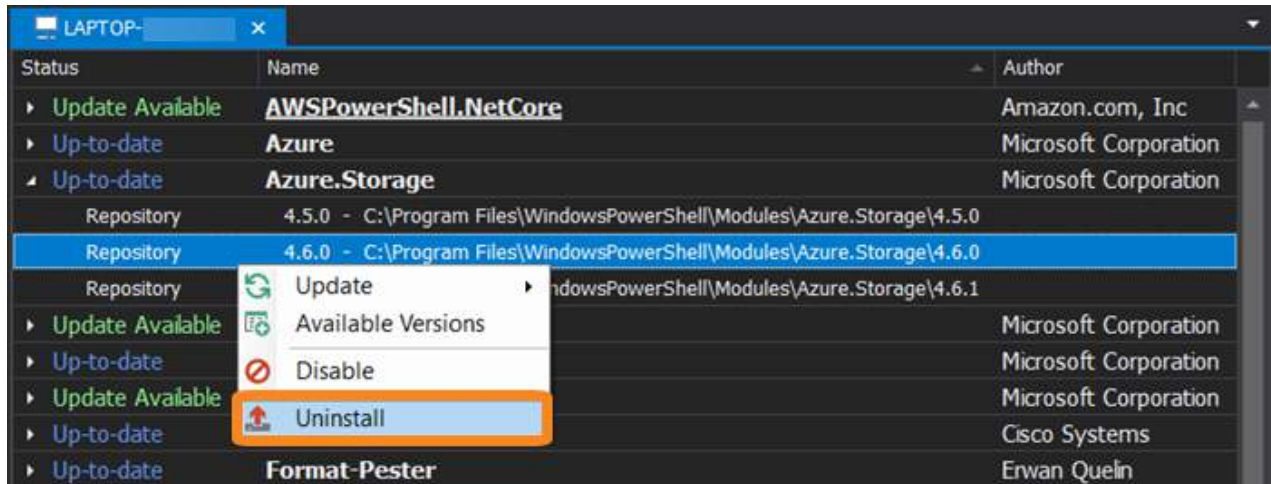
1. Select a module version in the Machine tab, and then click **Uninstall** (**Ctrl+R**) on the Home ribbon:



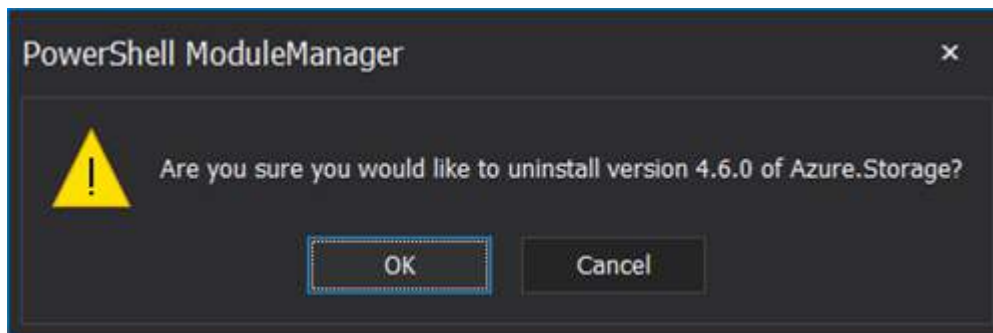
👍 Ctrl-click to select multiple versions.

-OR-

- Right-click on the selected module(s) in the Machine tab, and then select **Uninstall** from the context menu:

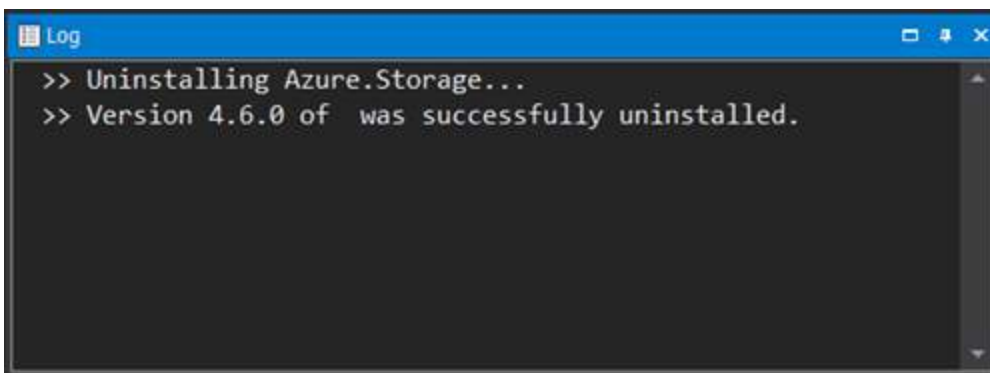


2. Click OK to confirm:



i If you selected more than one version in the Machine tab, you will be prompted to confirm each uninstall.

The uninstall progress and status is displayed in the Log panel:



The Machine tab will refresh when the module is uninstalled:

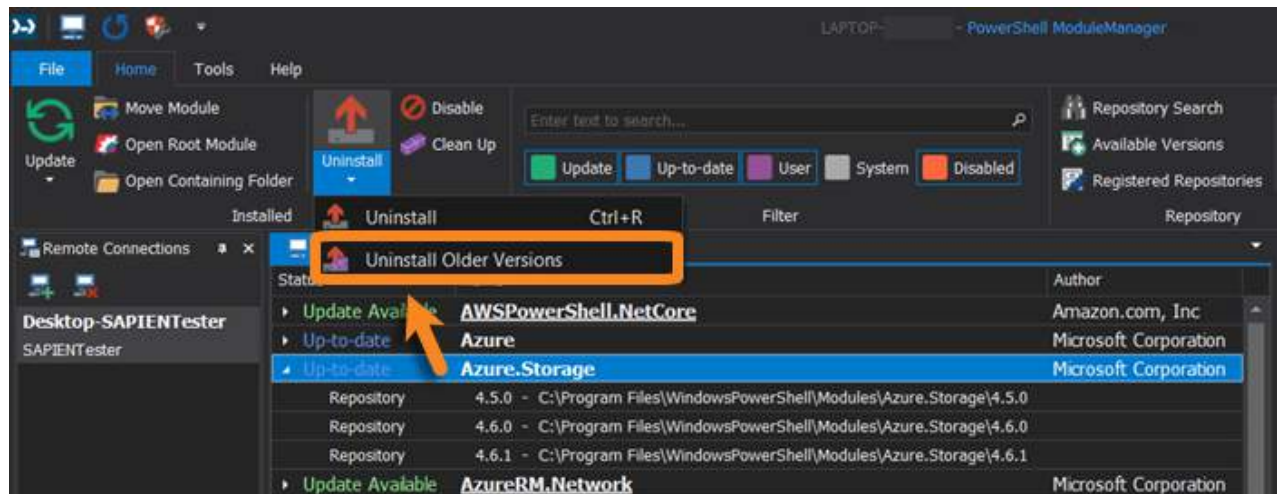
Status	Name	Author
Update Available	AWSPowerShell.NetCore	Amazon.com, Inc
Up-to-date	Azure	Microsoft Corporation
Up-to-date	Azure.Storage	Microsoft Corporation
	Repository 4.5.0 - C:\Program Files\WindowsPowerShell\Modules\Azure.Storage\4.5.0	
	Repository 4.6.1 - C:\Program Files\WindowsPowerShell\Modules\Azure.Storage\4.6.1	
Update Available	AzureRM.Network	Microsoft Corporation
Up-to-date	AzureRM.profile	Microsoft Corporation
Update Available	AzureRM.Sql	Microsoft Corporation

How to uninstall all older versions of a module

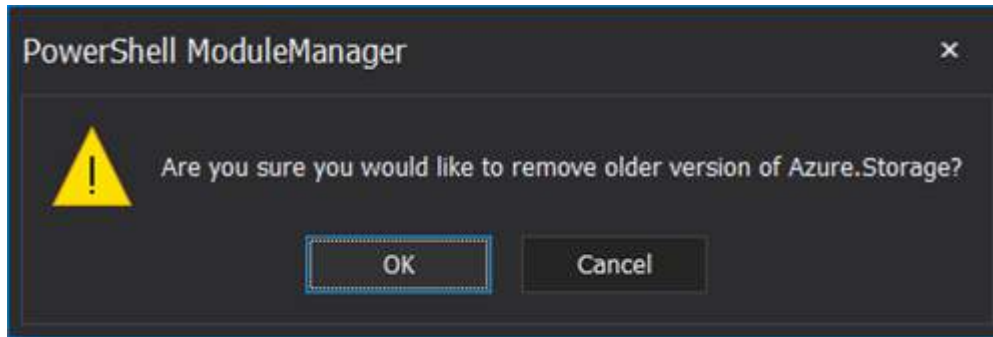
When PowerShell ModuleManager updates a module, the latest version is installed—but the previous versions remain. Use Uninstall Older Versions to keep the latest version, and remove all of the older versions.

To uninstall all older module versions

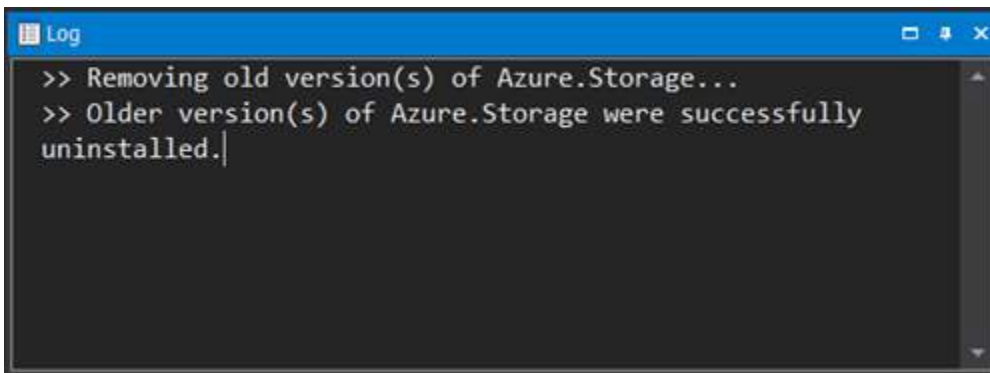
1. Select a module in the Machine tab, and then click **Uninstall Older Versions** from the Uninstall menu on the Home ribbon:



2. Click OK to confirm:



The uninstall activity will appear in the Log panel:



The Machine tab will refresh when all older versions of the module are removed:

Status	Name	Author
Update Available	AWSPowerShell.NetCore	Amazon.com, Inc
Up-to-date	Azure	Microsoft Corporation
Up-to-date	Azure.Storage	Microsoft Corporation
Repository	4.6.1 - C:\Program Files\WindowsPowerShell\Modules\Azure.Storage\4.6.1	
Update Available	AzureRM.Network	Microsoft Corporation
Up-to-date	AzureRM.profile	Microsoft Corporation

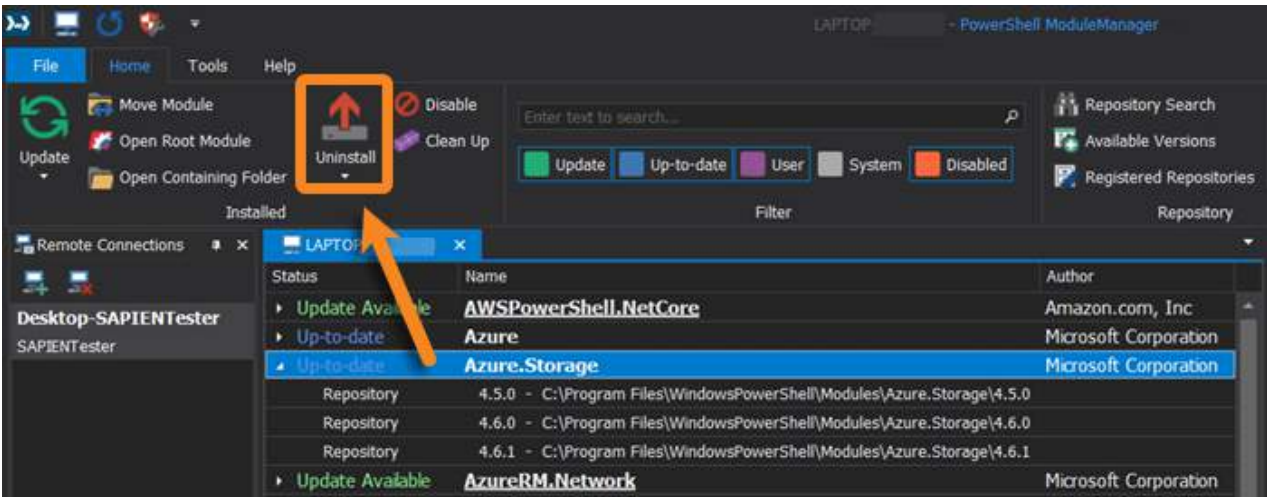
👍 When updating a module, you can remove all previous versions by selecting **Update, Remove Old Versions**.

How to uninstall all versions of a module

You can uninstall all versions of a module by selecting the module parent node in the Machine tab.

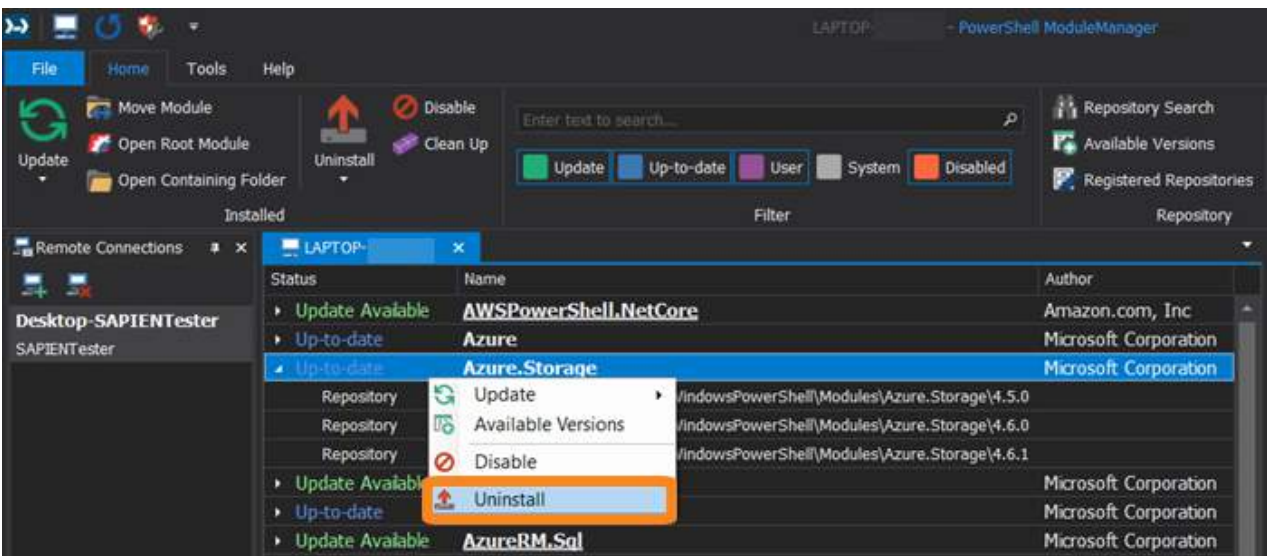
To uninstall all versions of a module

1. Select a module name in the Machine tab, and then click **Uninstall** (*Ctrl+R*) on the Home ribbon:

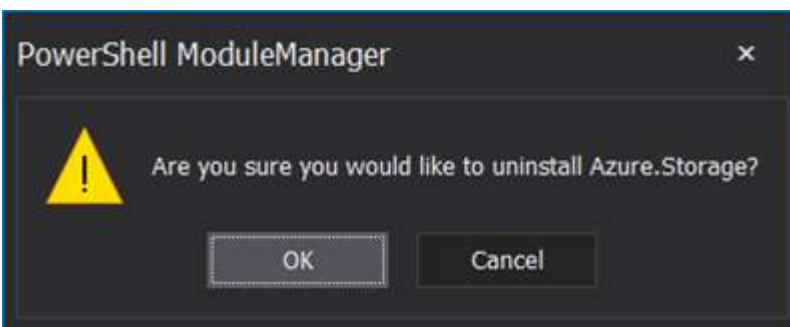


-OR-

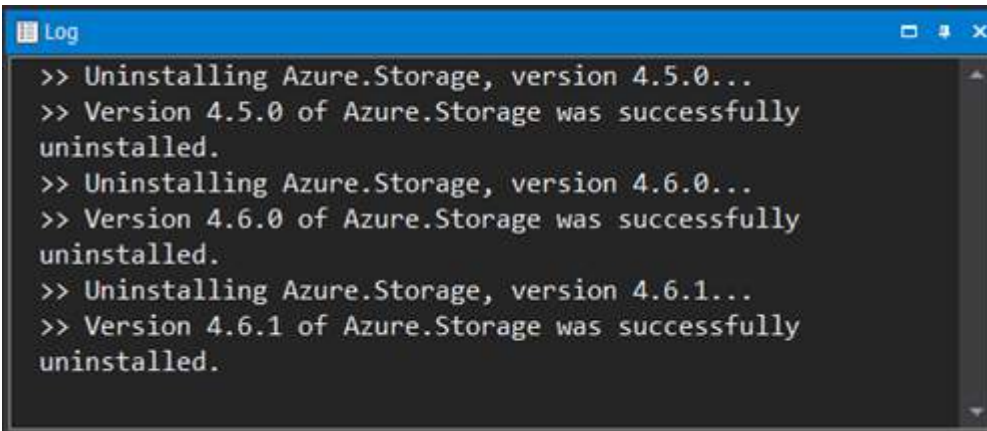
- Right-click a module in the Machine tab, and then select **Uninstall** from the context menu:



2. Click OK to confirm:

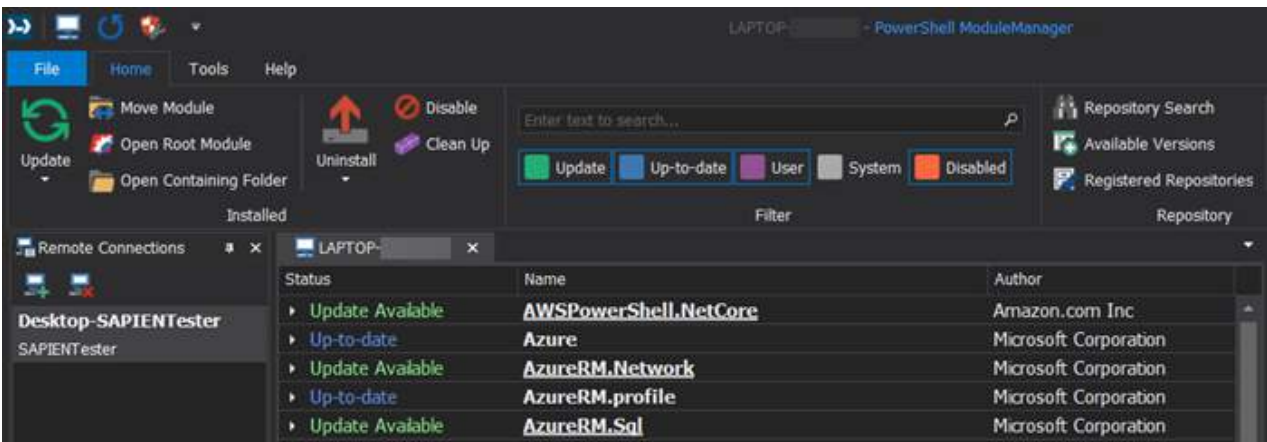


The uninstall activity will appear in the Log panel:



```
>> Uninstalling Azure.Storage, version 4.5.0...
>> Version 4.5.0 of Azure.Storage was successfully
uninstalled.
>> Uninstalling Azure.Storage, version 4.6.0...
>> Version 4.6.0 of Azure.Storage was successfully
uninstalled.
>> Uninstalling Azure.Storage, version 4.6.1...
>> Version 4.6.1 of Azure.Storage was successfully
uninstalled.
```

The Machine tab will refresh when all versions of the module are removed:

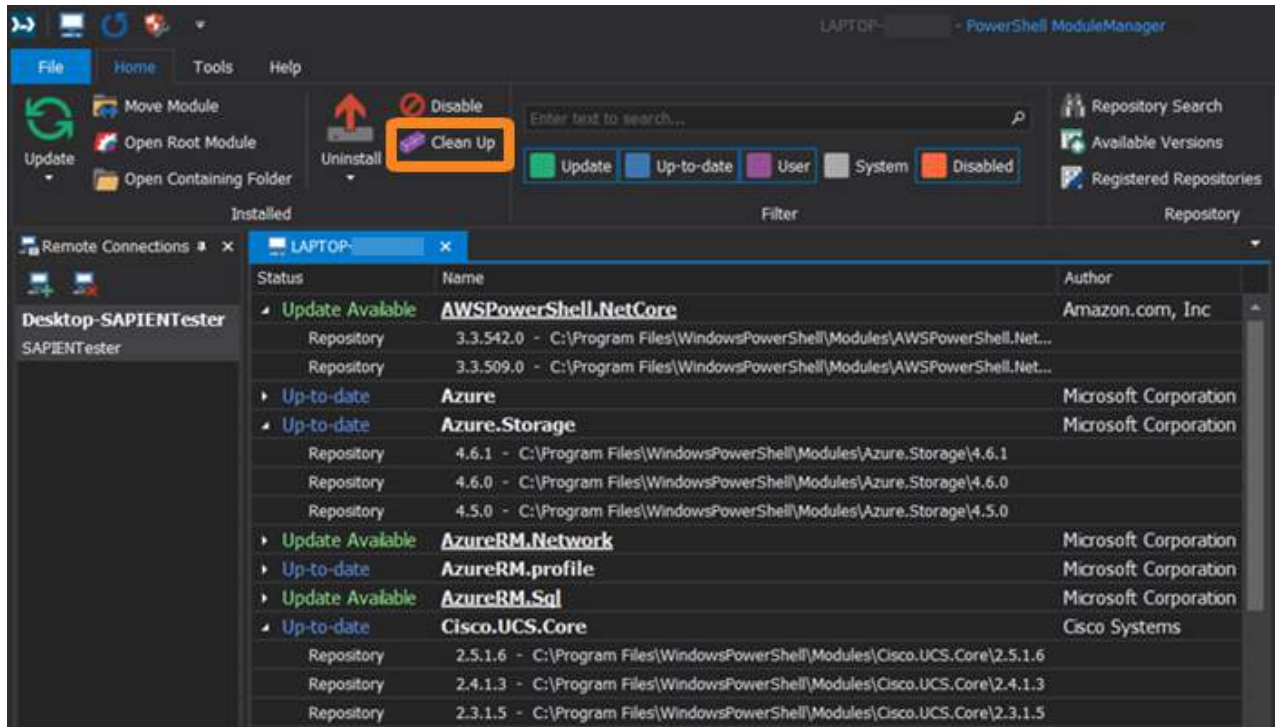


How to uninstall all older versions of 'all' modules (Clean Up)

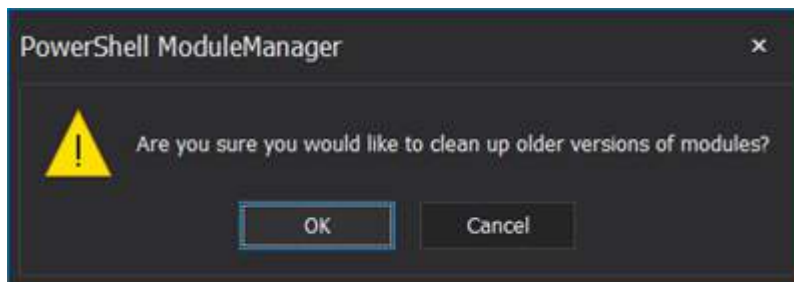
The **Clean Up** tool removes older versions of all modules, only keeping the latest versions installed.

To uninstall all older versions of all modules

1. Click **Clean Up** on the Home ribbon:



2. Click OK to confirm:



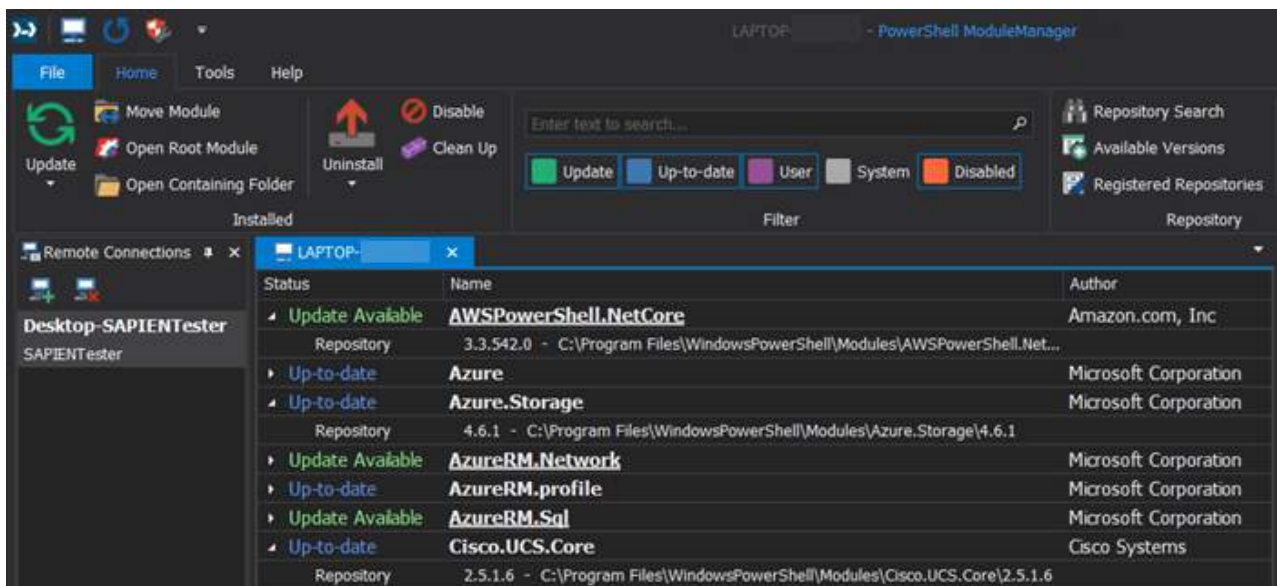
PowerShell ModuleManager will uninstall all older versions of all modules:



The uninstall activity will appear in the Log panel:

```
Log
>> Removing old version(s) of AWSPowerShell.NetCore...
>> Older version(s) of AWSPowerShell.NetCore were successfully uninstalled.
>> Removing old version(s) of Azure...
>> Older version(s) of Azure were successfully uninstalled.
>> Removing old version(s) of Azure.Storage...
>> Older version(s) of Azure.Storage were successfully uninstalled.
>> Removing old version(s) of AzureRM.Network...
```

The Machine tab will refresh when the older versions of all modules have been removed:

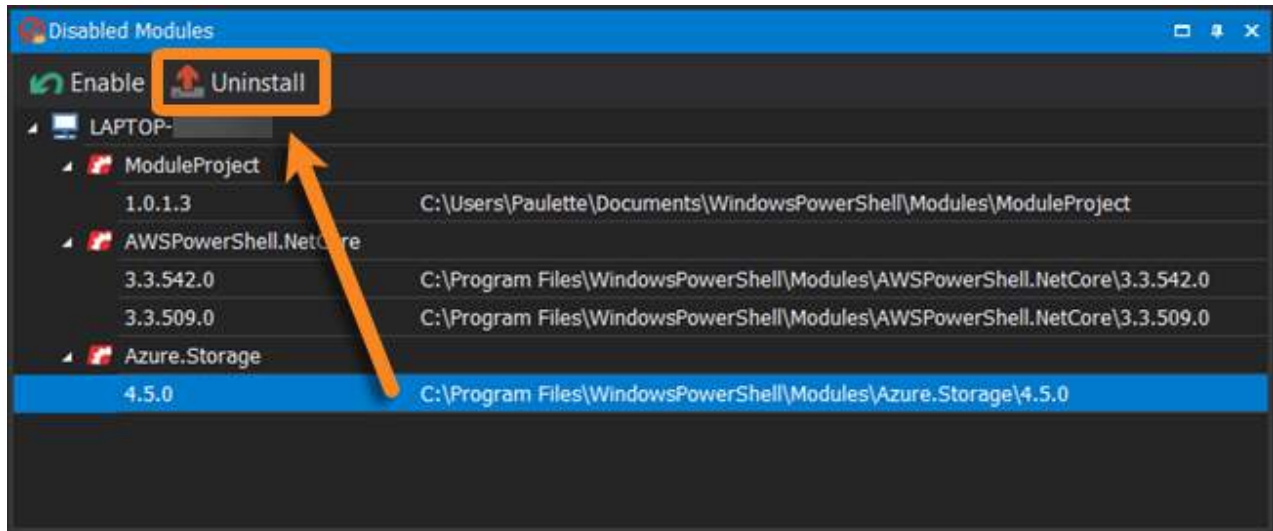


How to uninstall modules from the Disabled Modules panel

Modules can be uninstalled or [enabled](#)⁷² from the Disabled Modules panel.

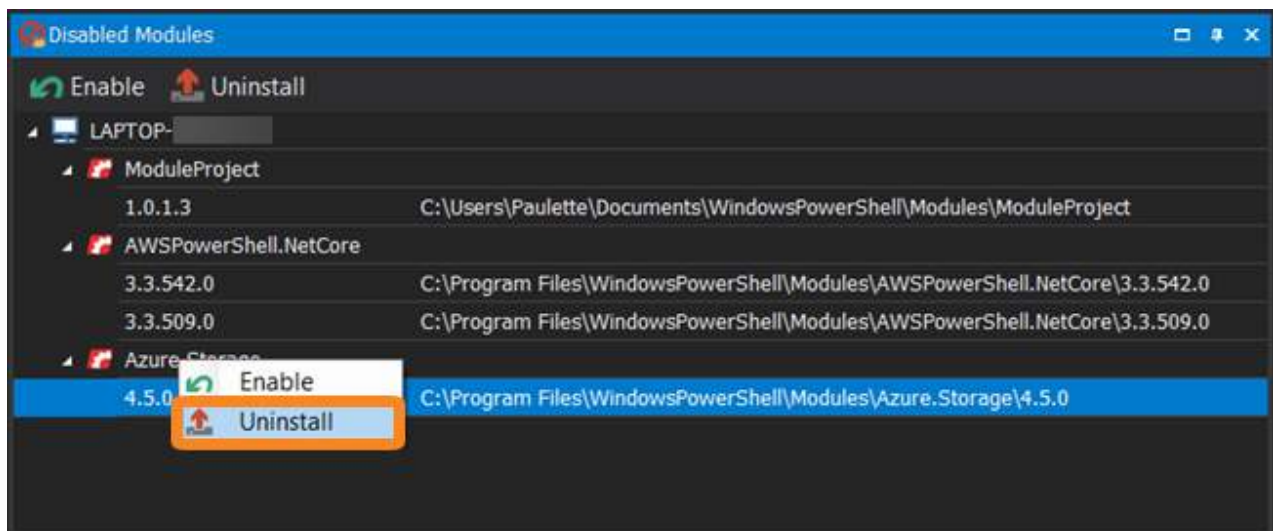
To uninstall a module

- Select a module, and then click the **Uninstall** button:



-OR-

- Right-click on a module, and then select **Uninstall** from the context menu:



👍 Selecting the parent node uninstalls all versions; selecting a child node uninstalls only that version.

The uninstall status is displayed in the Log panel:

```
Log
>> Uninstalling Azure.Storage...
>> Version 4.5.0 of Azure.Storage was successfully moved
back from the disabled folder.
>> Version 4.5.0 of Azure.Storage was successfully
uninstalled. |
```

The Machine tab will refresh, and the uninstalled module(s) will not be listed:

Status	Name	Author
Disabled	AWSPowerShell.NetCore	Amazon.com, Inc
Up-to-date	Azure	Microsoft Corporation
Up-to-date	Azure.AnalysisServices	Microsoft Corporation
Up-to-date	Azure.Storage	Microsoft Corporation
Repository	4.6.1 - C:\Program Files\WindowsPowerShell\Modules\Azur...	
Repository	4.6.0 - C:\Program Files\WindowsPowerShell\Modules\Azur...	
Update Available	AzureRM.Network	Microsoft Corporation
Up-to-date	AzureRM.profile	Microsoft Corporation

Possible errors during uninstall

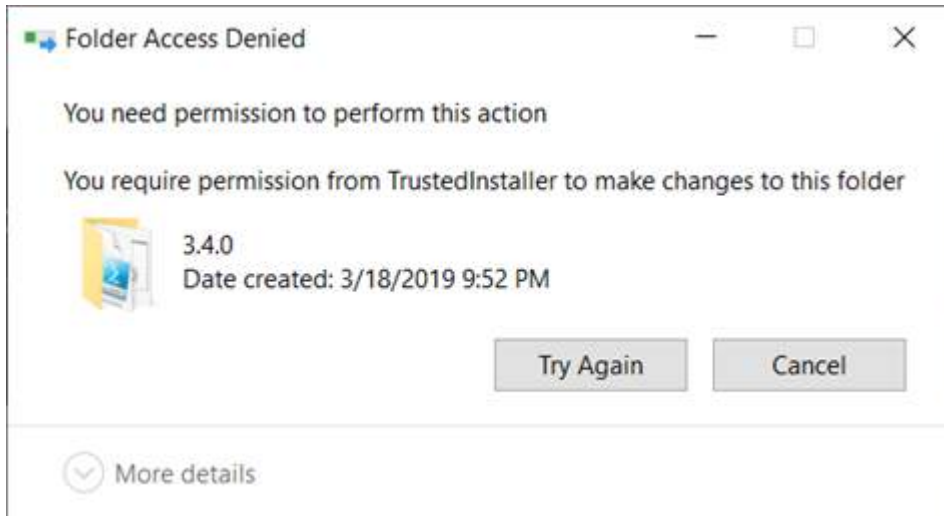
Some known errors may occur when uninstalling specific modules. If you encounter errors that are not listed here, please submit them in the [PowerShell ModuleManager support forum](#).

PackageManagement

PackageManagement and PowerShellGet are both required by PowerShell ModuleManager. Both modules are, by default, installed with Windows. If you uninstall and then update these modules, sometimes the runspace environment gets "confused," and errors can occur. We are working on implementing a way for PowerShell ModuleManager to handle this.

Pester 3.4

Pester 3.4 is installed with Windows and is by default "locked" by TrustedInstaller:



In this scenario, an Administrator account is required to uninstall Pester.

PowerShellGet

PackageManagement and PowerShellGet are both required by PowerShell ModuleManager. Both modules are, by default, installed with Windows. If you uninstall and then update these modules, sometimes the runspace environment gets "confused," and errors can occur. We are working on implementing a way for PowerShell ModuleManager to handle this.

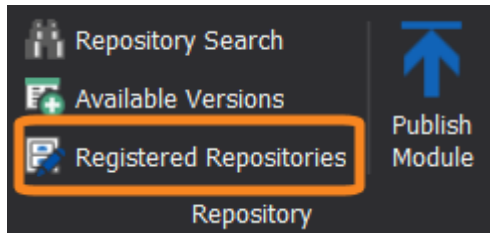
4.4 Managing Repositories

You can add, edit, or delete repositories from the **Registered Repositories** dialog.

👍 If you want to share modules you have written, the repository you wish to [publish](#)^[60] to must be a registered repository in PowerShell.

Registered Repositories

You can access the Registered Repositories dialog from the **Home** ribbon > **Registered Repositories**:



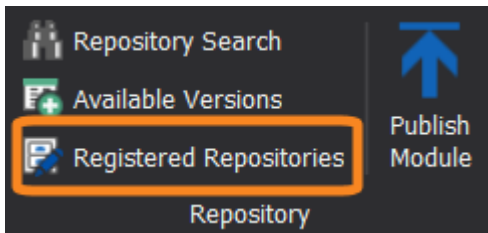
Home ribbon > Registered Repositories

i You can use the Registered Repositories dialog for repositories that are on network drives, but not for repositories that are on mapped drives.

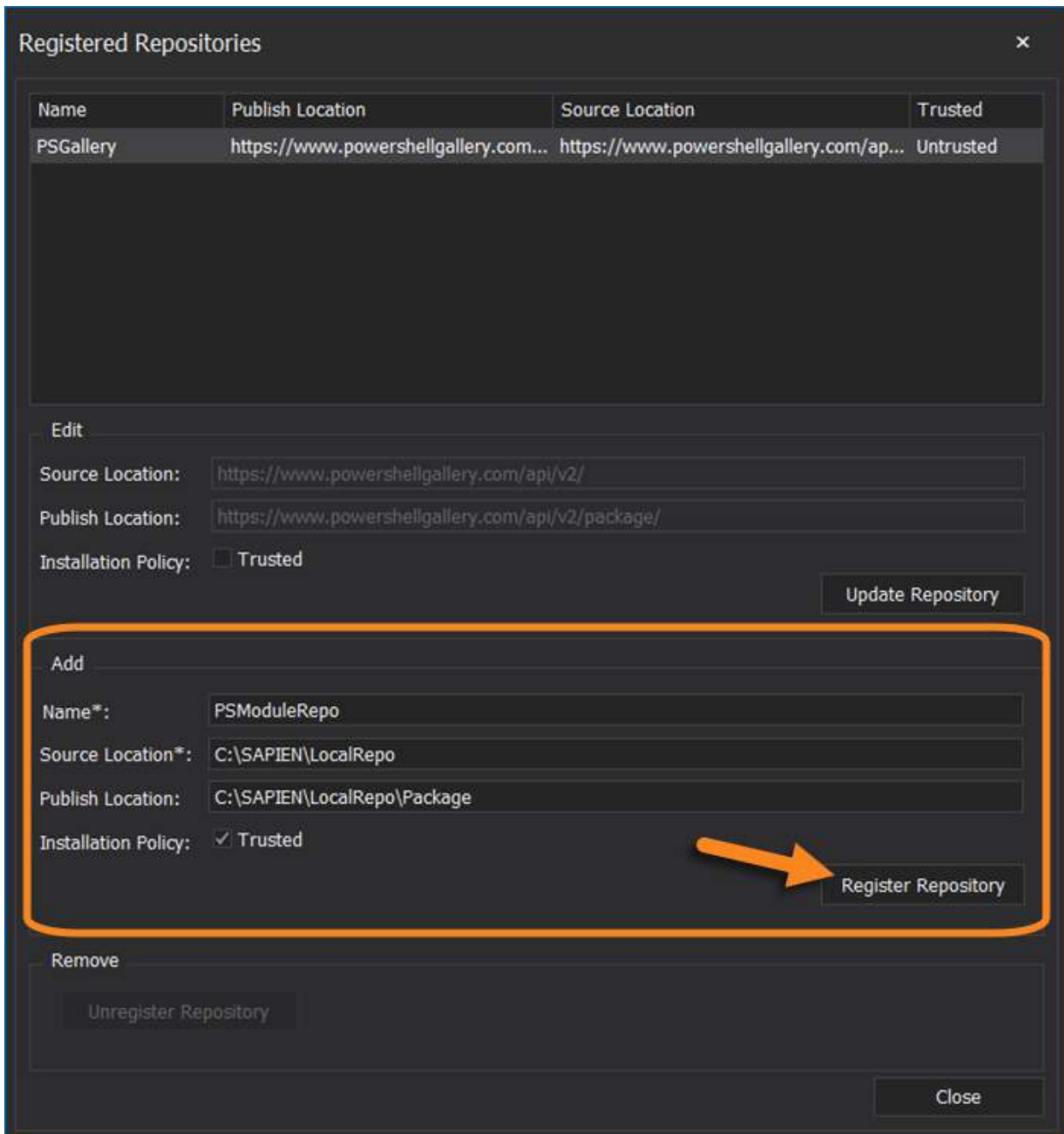
How to register a repository

To add a repository

1. On the **Home** ribbon > click **Registered Repositories**:



2. In the Registered Repositories dialog, enter the required data in the Add section and then click **Register Repository**:



Registered Repositories Dialog > Add

Add Repository - Field Definitions:

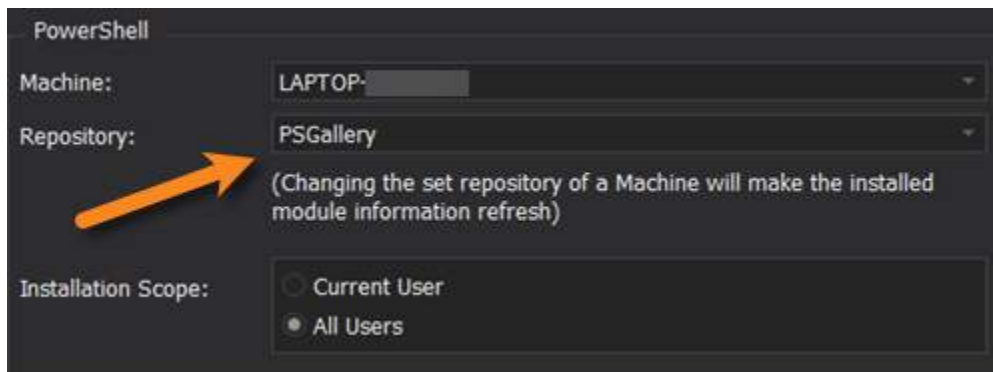
- Name***: The repository name.
- Source Location***: The URL or file path of the repository to search and install modules from.
- Publish Location**: The URL or file path of the repository where modules should be published.
- Installation Policy**: Specifies the PowerShell behavior when installing from the repository:
- **Untrusted**: Prompts for confirmation before installing.
 - **Trusted**: Does not prompt for confirmation before installing.

* Required fields

PowerShell ModuleManager will register the repository:

Name	Publish Location	Source Location	Trusted
PSGallery	https://www.powershellgallery.com...	https://www.powershellgallery.com/ap...	Untrusted
PSModuleRepo	C:\SAPIEN\LocalRepo\Package	C:\SAPIEN\LocalRepo	Trusted

The default registered repository is the PowerShell Gallery (PSGallery). This can be changed in **File > Options > General > PowerShell**:

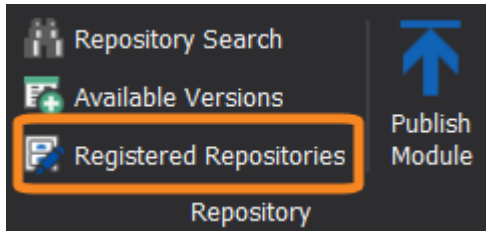


Options > General > Powershell

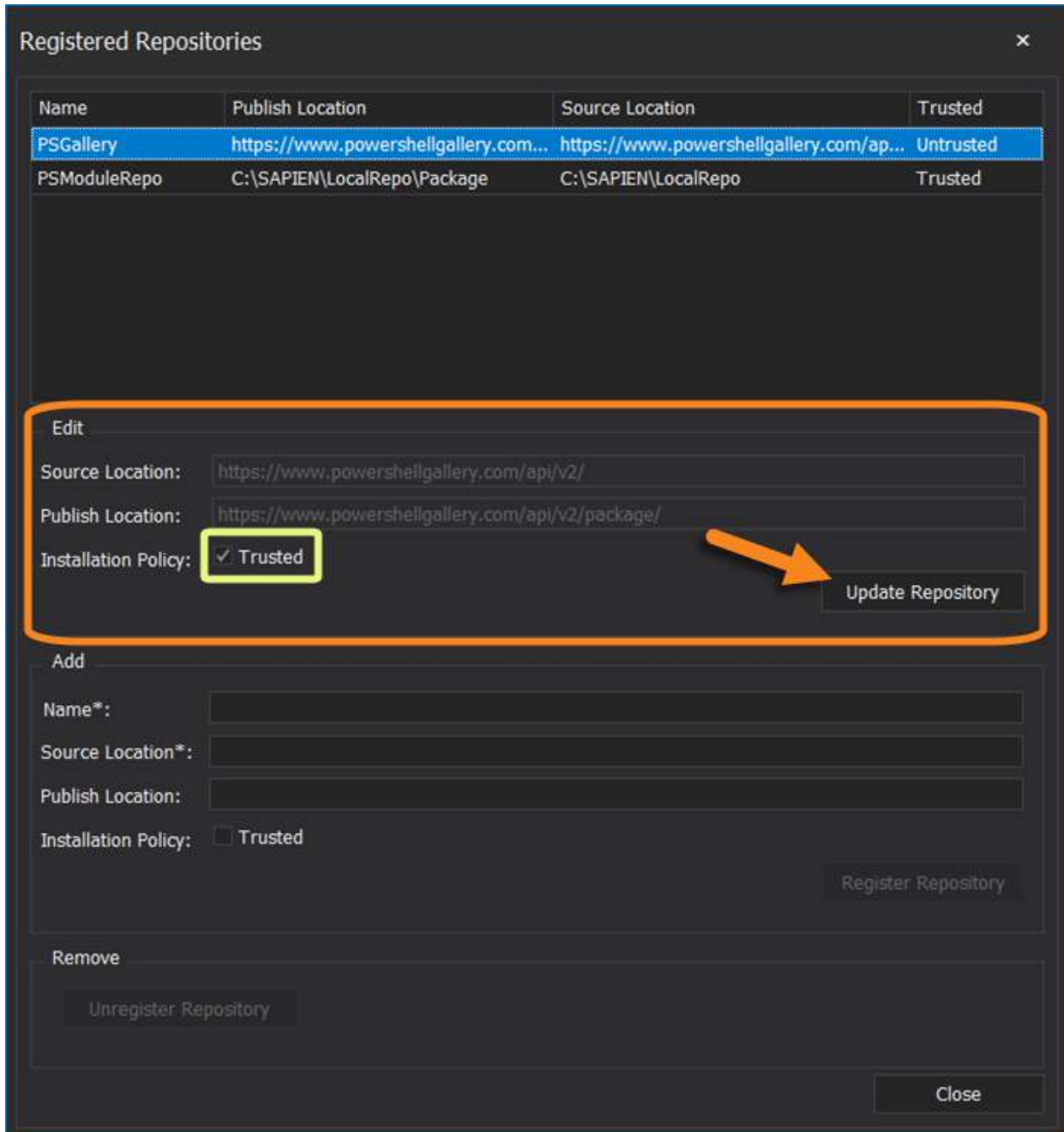
How to edit a repository

To edit a repository

1. On the **Home** ribbon > click **Registered Repositories**:



2. Select a repository and make any changes in the Edit section, then click **Update Repository**:



Registered Repositories Dialog > Edit

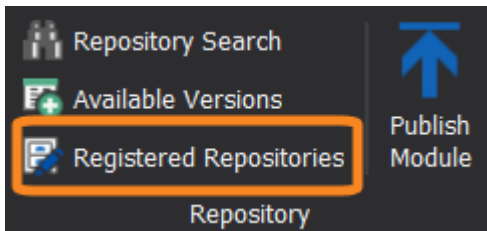
3. PowerShell ModuleManager will apply the changes to the repository:

Name	Publish Location	Source Location	Trusted
PSGallery	https://www.powershellgallery.com...	https://www.powershellgallery.com/ap..	Trusted
PSModuleRepo	C:\SAPIEN\LocalRepo\Package	C:\SAPIEN\LocalRepo	Trusted

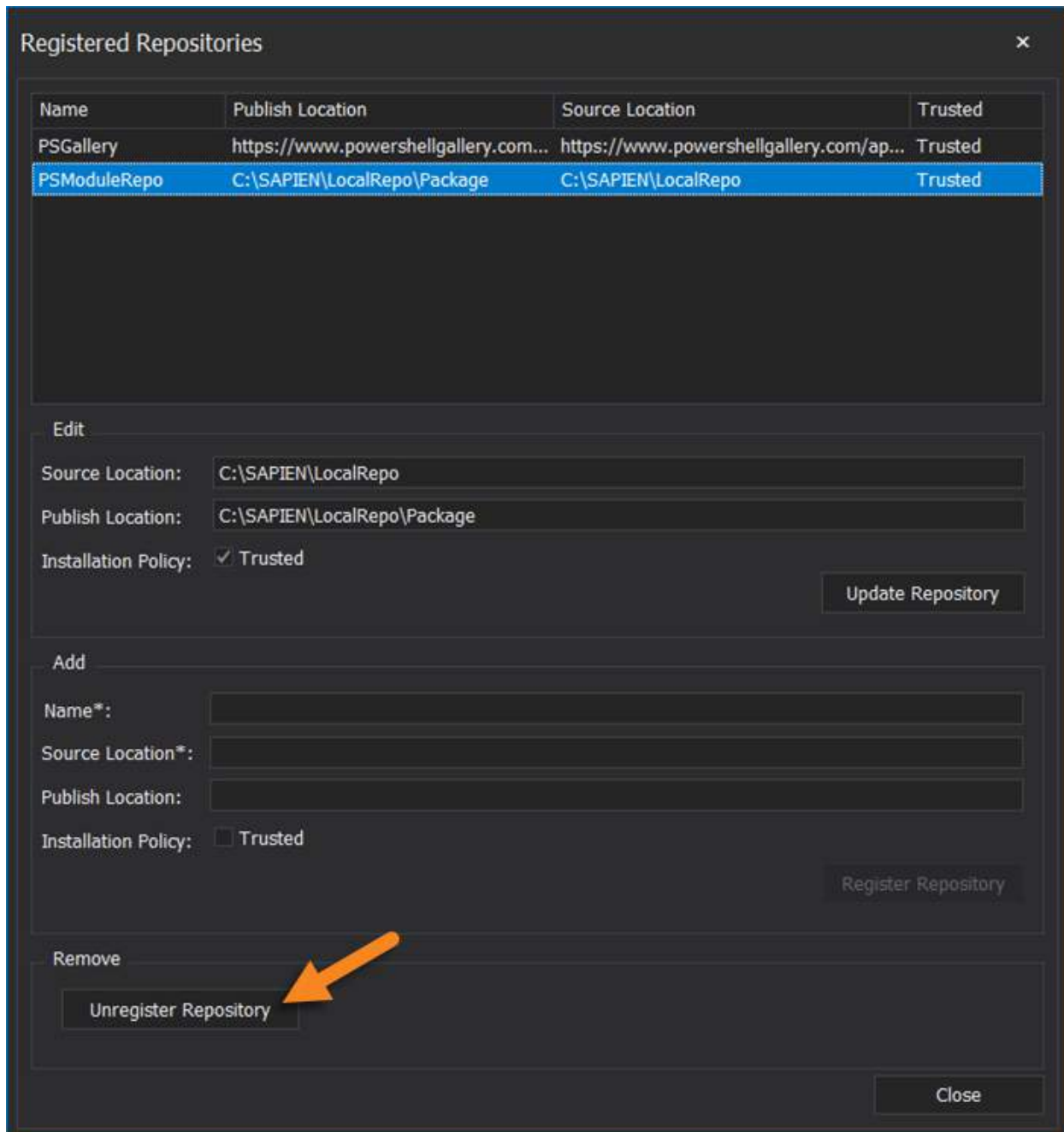
How to unregister a repository

To unregister a repository

1. On the **Home** ribbon > click **Registered Repositories**:

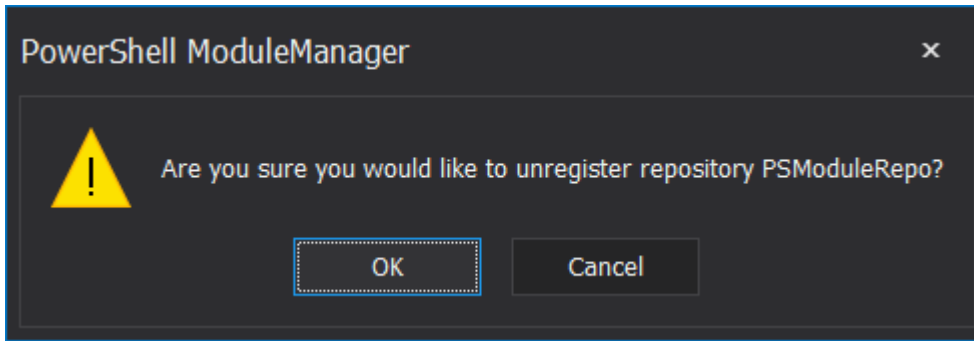


2. Select a repository, and then in the Remove section, click **Unregister Repository**:

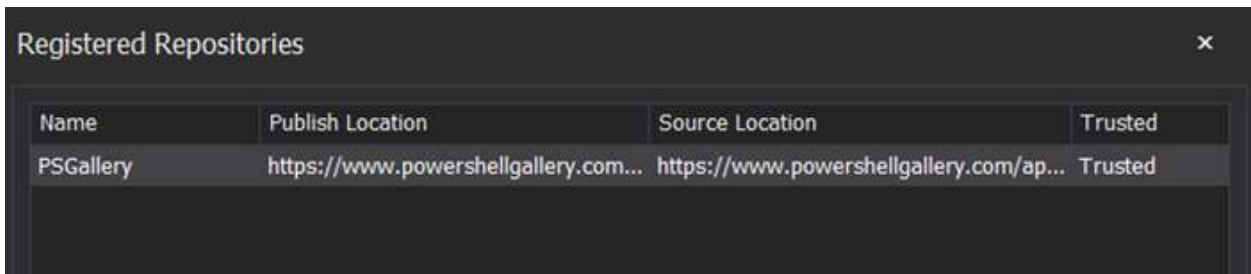


Registered Repositories Dialog > Remove

3. Click **OK** to confirm:

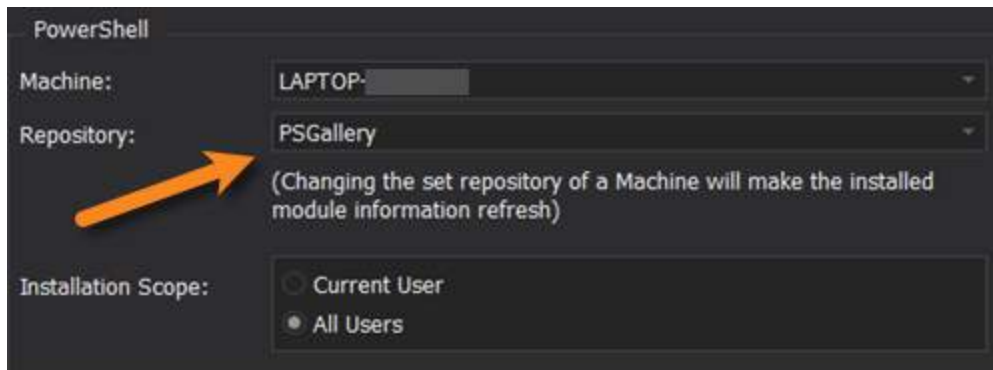


- PowerShell ModuleManager will remove the repository:



How to change the default registered repository

The default registered repository is the PowerShell Gallery (PSGallery). This can be changed in **File > Options > General > PowerShell**:



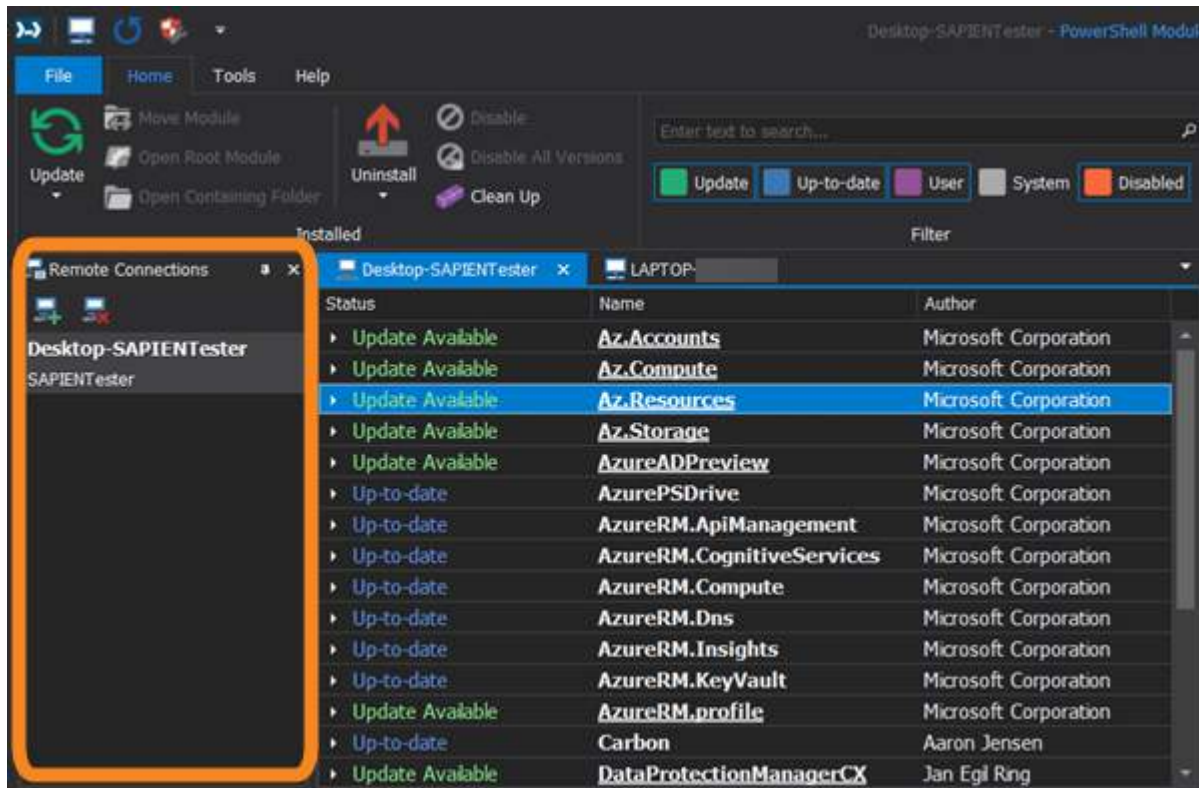
Options > General > Powershell

4.5 Managing Remote Connections

PowerShell ModuleManager allows you to manage modules on remote machines. The tasks in this section show you how to add, connect to, and remove a remote machine.

Remote Connections

You can manage remote machine connections in the **Remote Connections** panel, located in the middle-left of the PowerShell ModuleManager workspace:



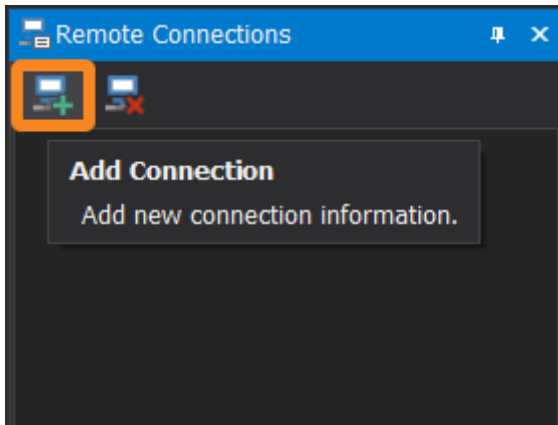
Remote Connections panel and remote machine open in Main Window

👍 Double-click on an existing connection in the Remote Connections panel to view the remote machine in the [main window](#) ¹⁶.

How to add a remote machine

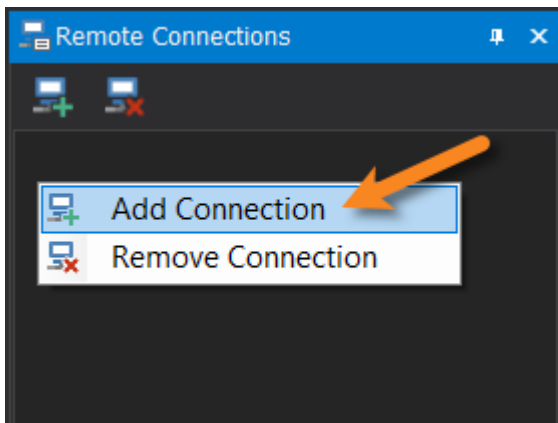
To add a remote connection

- In the Remote Connections panel, click the **Add Connection** button:

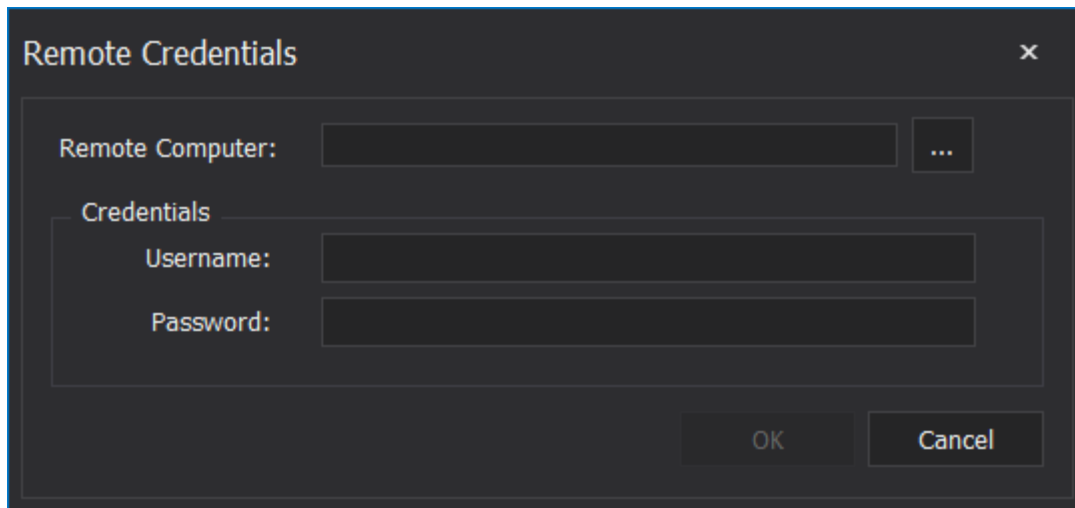


-OR-

- Right-click on the Remote Connections panel, then select **Add Connection**:



The Remote Credentials dialog will open:

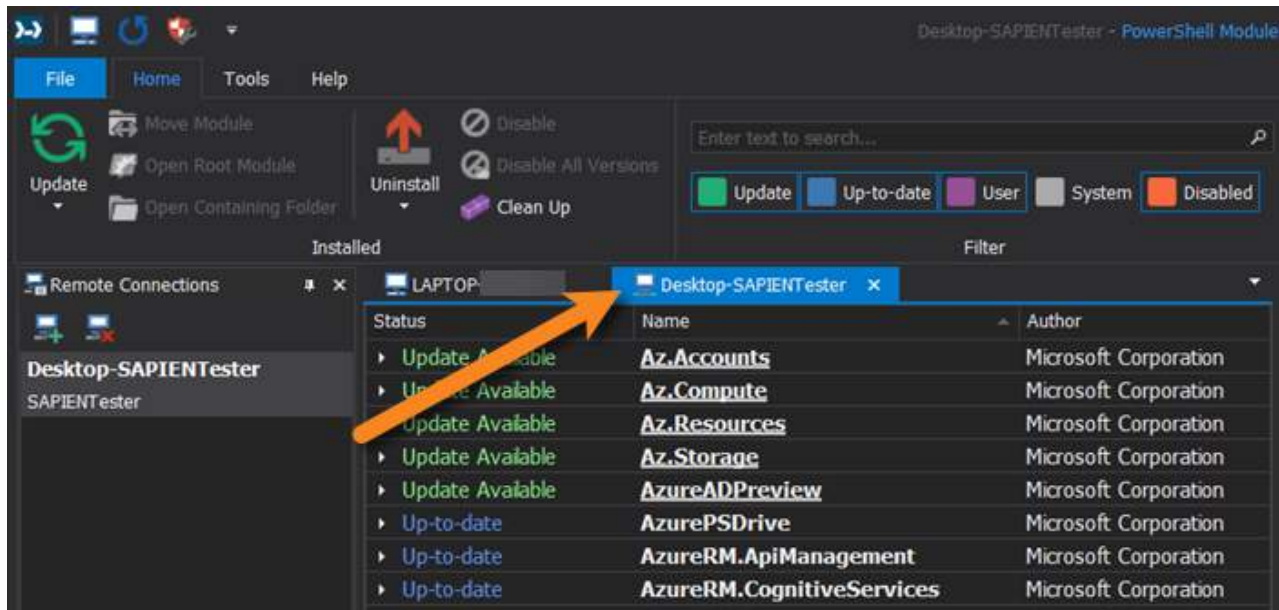


Remote Credentials Dialog

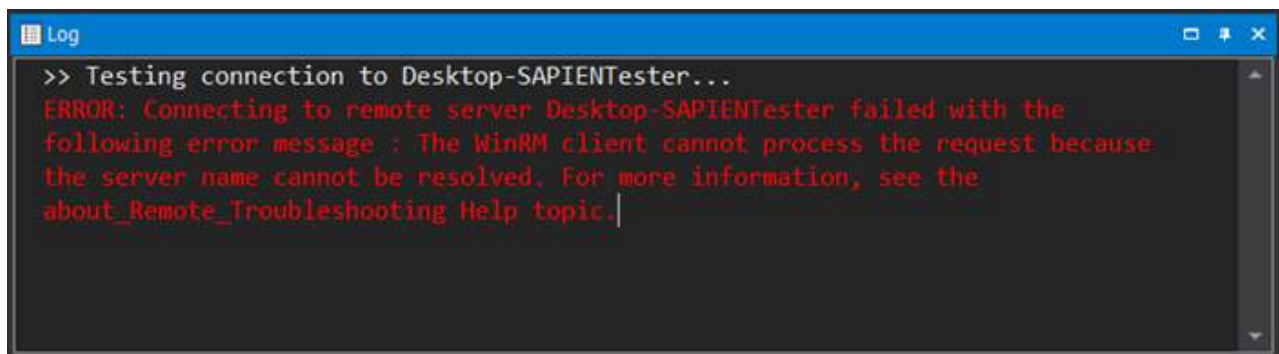
In the Remote Credentials dialog:

1. Enter the computer name in the Remote Computer field, or search connected computers (...).
2. Enter the credentials and click **OK** to add the remote computer.

A new tab with the computer's name will appear in the main window when the remote machine is connected:



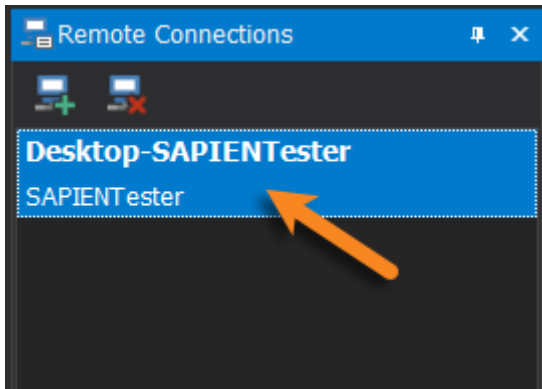
- i** If a connection is not established, an error will display in the Log panel:



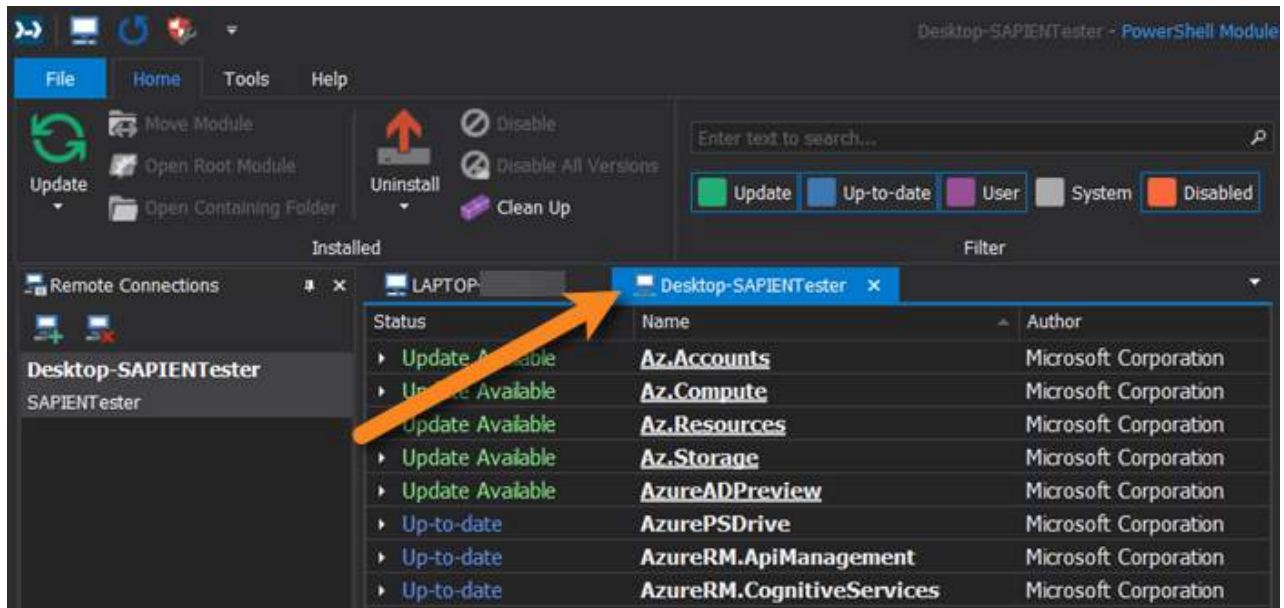
How to connect to a remote machine

To connect to an existing remote machine

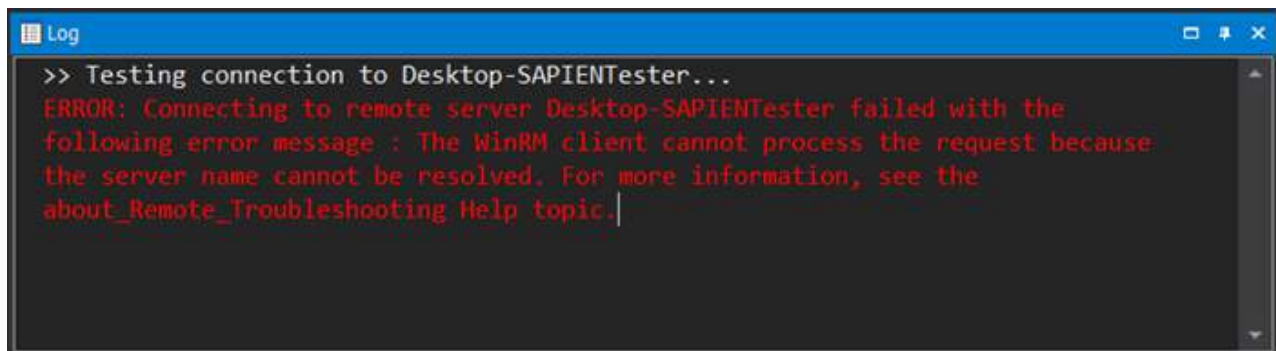
- In the Remote Connections panel, double-click on an existing connection:



A new tab with the computer's name will appear in the main window when the remote machine is connected:



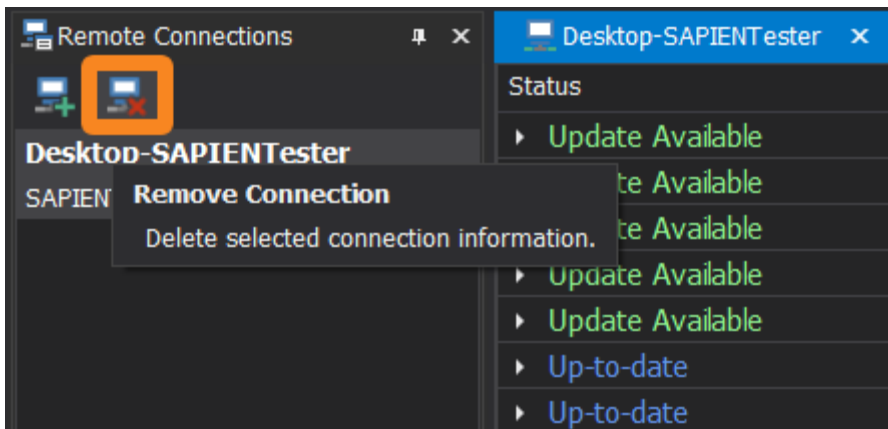
i If a connection is not established, an error will display in the Log panel:



How to remove a remote machine

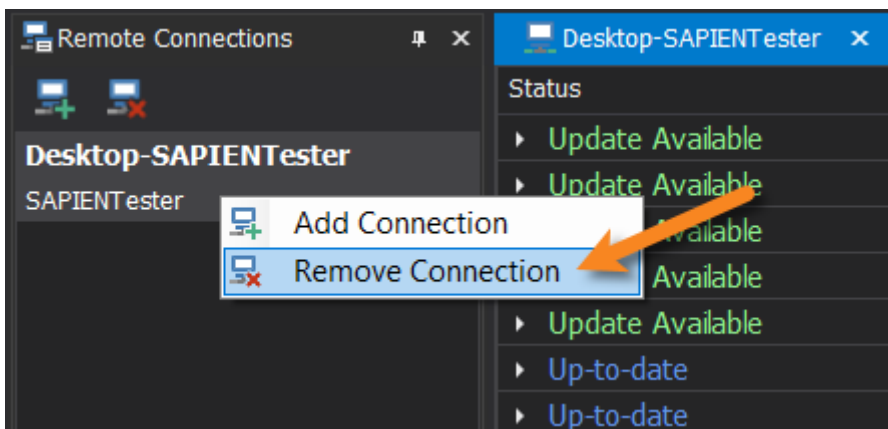
To remove a remote connection

- In the Remote Connections panel, select the remote machine and then click the **Remove Connection** button:



-OR-

- In the Remote Connections panel, right-click on the remote machine, then select **Remove Connection**:



The remote machine is removed from the Remote Connections panel.

5 Reference

The Reference section documents all of PowerShell ModuleManager's menu options and dialogs, and also covers the available program and customization options. Information about the SAPIEN Updates tool is also provided, as well as the Frequently Asked Questions.

There are extensive cross-references and links to the Managing Modules procedures sections, enabling you to find relevant task-based instructions for what is described in the Reference topics.

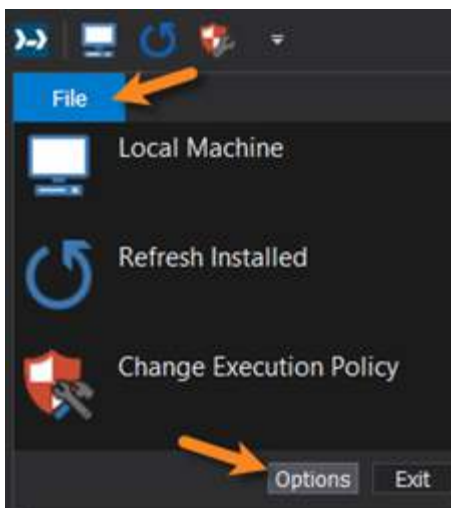
5.1 Program Options

The Options dialog contains program settings that are available for PowerShell ModuleManager. We recommend that you review these options to understand the settings that you can adjust and configure.

Accessing the program options

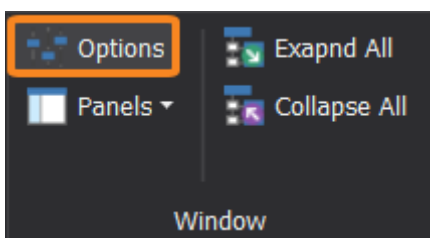
To access the program Options

- On the ribbon, select **File > Options**:



-OR-

- On the ribbon, select **Home > Options**:

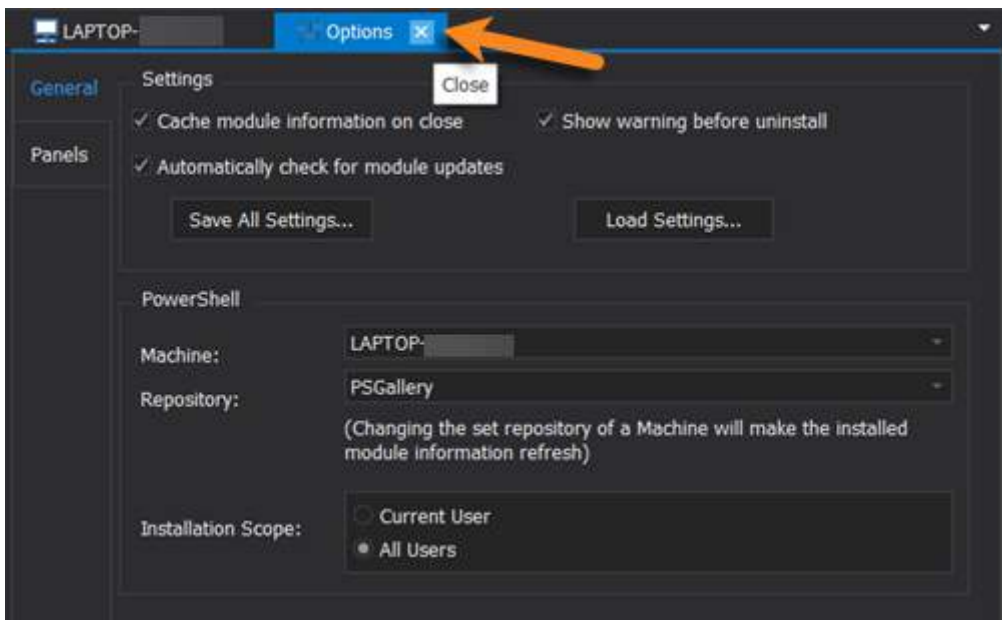


The Options open in the [main window](#)¹⁶, located in the center of the PowerShell ModuleManager workspace.

Closing the program options

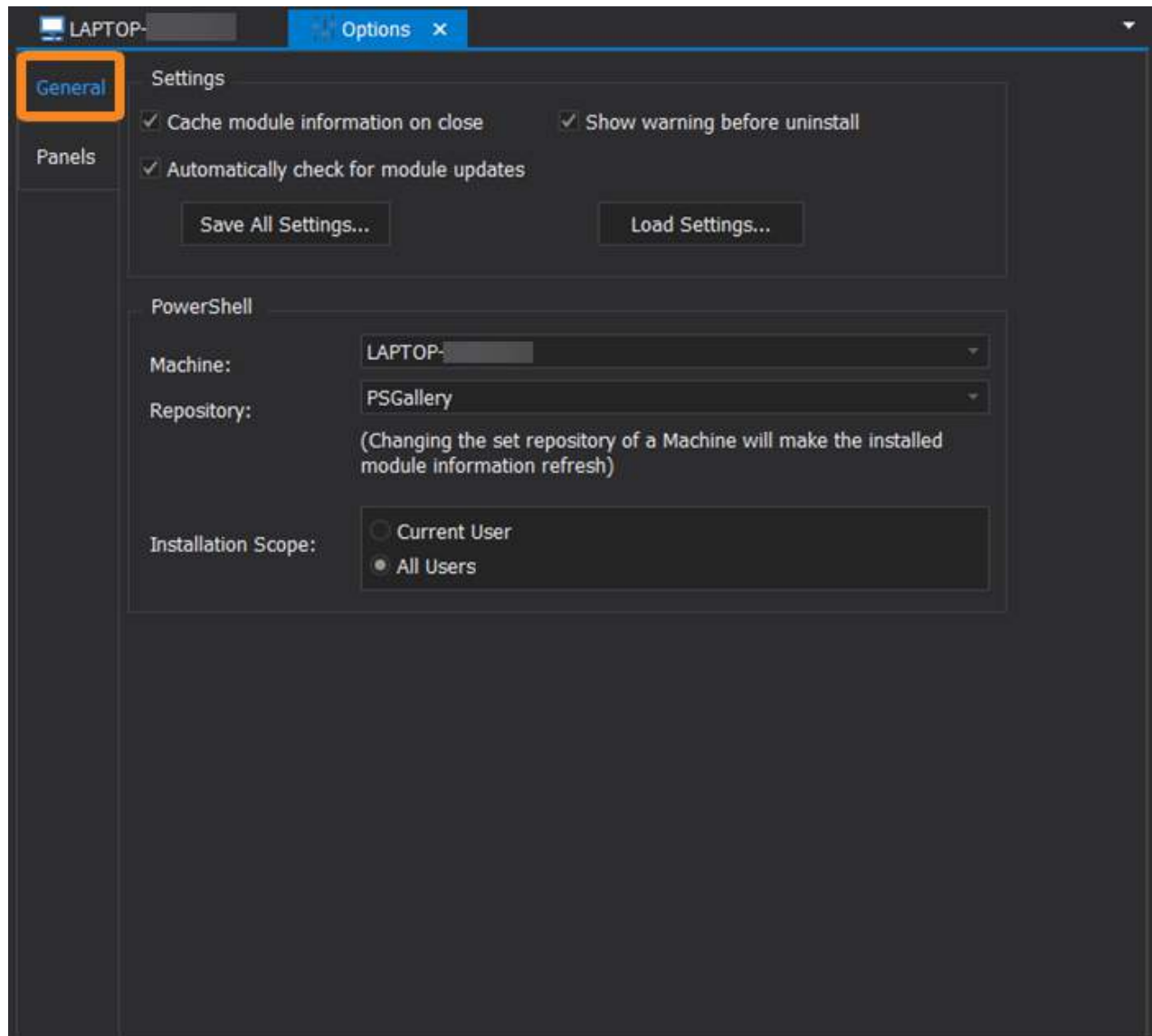
To close the program options

- Click the X on the right of the Options tab:



5.1.1 Program Options - General

The options on the **General** tab control various module settings such as the cache, uninstall, and update behaviors. They also allow you to save and load all program settings, as well as set the Repository and Installation Scope for a selected machine.



Program Options > General tab

Settings

These settings control the module cache, uninstall, and update behavior, and also allow you to save and load setting configurations.

Cache module information on close


Caches module information from the current session to load into the next session, allowing the application to load faster.

Show warning before uninstall

Prompts with a confirmation dialog before uninstalling a module.

Automatically check for module updates

During program startup, by default PowerShell ModuleManager automatically checks for updates by comparing the installed modules to the modules in the set repository.

 The set repository can be changes in File > Options > General > PowerShell.

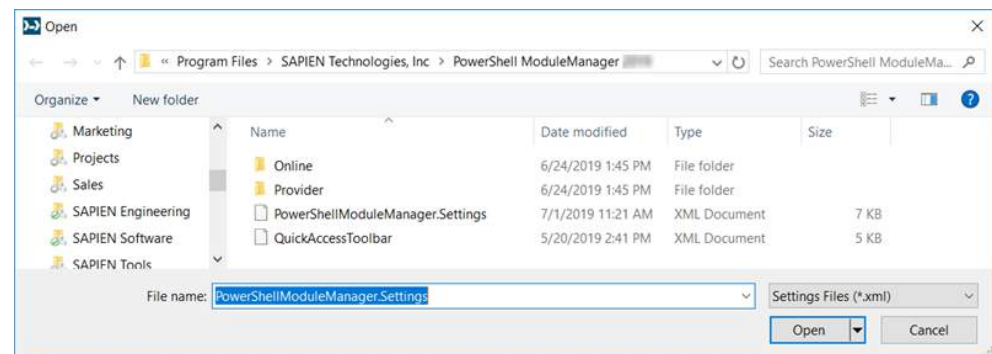
Save All Settings...

Saves all current program settings. Be default the settings are saved to:

`%systemdrive%\Program Files\SAPIEN Technologies, Inc\PowerShell ModuleManager<yyyy>\PowerShellModuleManager.Settings.xml`

Load Settings...

Opens File Explorer to load previously saved settings:



PowerShell

These settings allow you to set the Repository and Installation Scope for the selected machine.

Machine

Select the machine name from the drop-down list. The default is the local machine name.

Repository

Designates the repository used when searching for module information, updating modules, and browsing in Repository Search.

Installation Scope

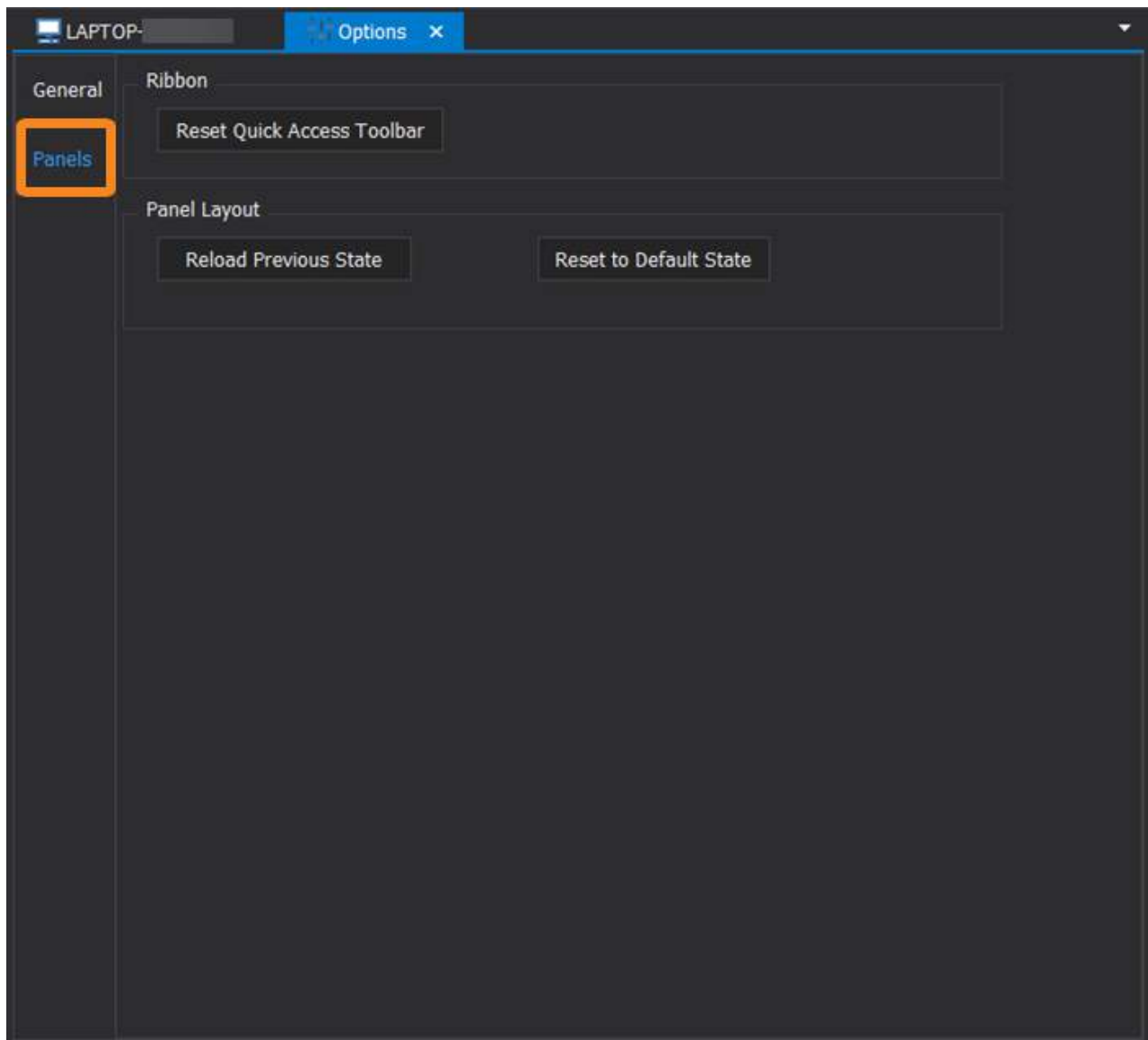
Specifies where modules will be installed when updating or installing:

- **Current User**

Modules will be installed in a location that is only available to the current user.

5.1.2 Program Options - Panels

The options on the **Panels** tab allow you to reset the Quick Access Toolbar, and reset or reload the Panel Layout.

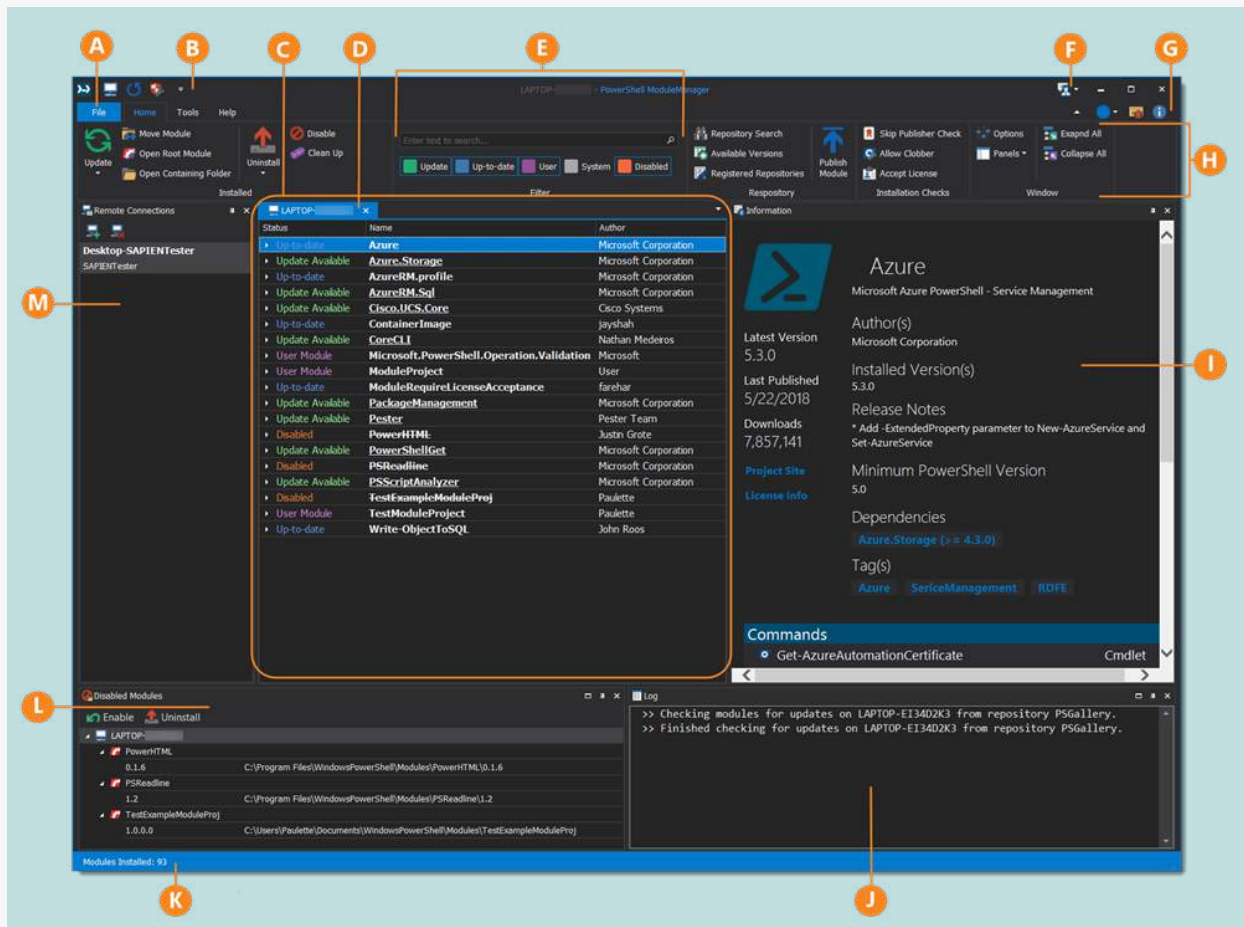


Program Options - Panels tab

Ribbon	Reset Quick Access Toolbar Resets the Quick Access Toolbar to its default state.
Panel Layout	Reload Previous State Reloads the panels to the same layout pattern they were in the last time you opened PowerShell ModuleManager.
	Reset to Default State Restores the PowerShell ModuleManager panels to the default layout.

5.2 Workspace, Ribbons and Dialogs

This section provides details for the PowerShell ModuleManager workspace and all of its elements.



PowerShell ModuleManager workspace

A. File Menu B. Quick Access Toolbar C. Main Window D. Local Machine E. Search and Filter F. Send Feedback G. Ribbon Header Buttons H. Ribbon Toolbar I. Information Panel J. Log Panel K. Status Bar L. Disabled Modules Panel M. Remote Connections Panel

File Menu

The [File Menu](#)¹⁰⁷ on the ribbon has options to view your local machine, refresh all installed modules (F5), change the PowerShell execution policy, and access the PowerShell ModuleManager program options.

Quick Access Toolbar

The [Quick Access Toolbar](#)^[117] on the top-left of the program window provides easy access to frequently used functions, and you can easily [add your own controls](#)^[155] to this toolbar.

Main Window

The [main window](#)^[123] is located in the center of the PowerShell ModuleManager workspace and is a central area for managing modules installed on the [Local Machine](#)^[120] and any [Remote Connections](#)^[137]. When the [Program Options](#)^[98] are accessed, they will also appear in the main window.

Local Machine

The [Local Machine](#)^[120] tab is located in the main window and is used to manage locally installed modules.

Search and Filter

The Filter section on the Home ribbon allows you to [search and filter](#)^[31] installed modules.

Send Feedback

The [Send Feedback](#)^[118] menu on the top-right of the program workspace provides links to our customer support forums where you can report a problem, ask a product-related question, and submit a suggestion or feature request.

Ribbon Header Buttons

The [Ribbon Header Buttons](#)^[119] on the top-right of the ribbon provide options to minimize the ribbon, change color-themed skins, view license information, and display product information.

Ribbon Toolbar

The Ribbon Toolbar, or [Ribbon](#)^[106], is organized in tabs, with each tab containing tools grouped logically by function. Some tools also provide access to dialogs with additional options.

Information Panel

The [Information panel](#)^[133] displays details for installed modules.

Log Panel

The [Log panel](#)^[135] displays PowerShell ModuleManager output messages.

Status Bar

The [Status Bar](#)^[120] displays the current number of installed modules.

Disabled Modules Panel

The [Disabled Modules panel](#)^[129] displays all disabled modules, where they can easily be enabled or uninstalled.

Remote Connections Panel

Use the [Remote Connections panel](#)^[137] to add, manage, and remove remote connections. When a remote connection is added and opened, it will appear as a tab in the [main window](#)^[123].

5.2.1 The Ribbon

PowerShell ModuleManager displays a Ribbon toolbar, or "Ribbon," at the top of the application window:

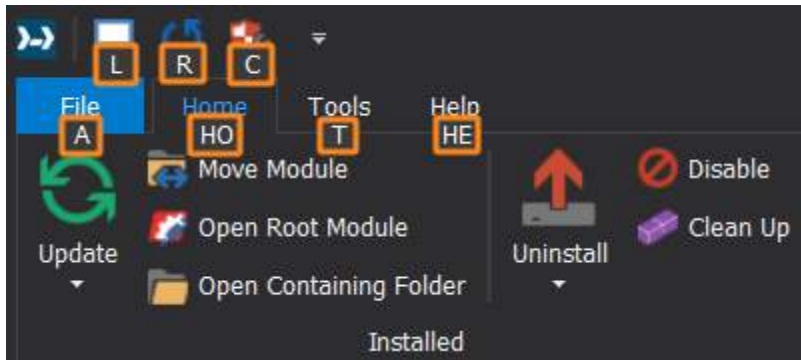


The Ribbon

The Ribbon Toolbar is where you access virtually all of PowerShell ModuleManager's functions. The ribbon is context-sensitive, automatically displaying the functions relevant to what you are doing at the moment. Functions that are not available are grayed out.

Accelerator Keys

Many of the ribbon controls are accessible from the keyboard; press and release the <Alt> key once, and the [accelerator keys](#)^[27] will be displayed as icons superimposed on the ribbon:



Accelerator Keys

Ribbon Tabs

Each tab on the ribbon contains tools grouped into sections based on related functions. Some controls also provide access to [dialogs](#)^[142] with additional options.

[File](#)^[107] A menu of frequently used functions.

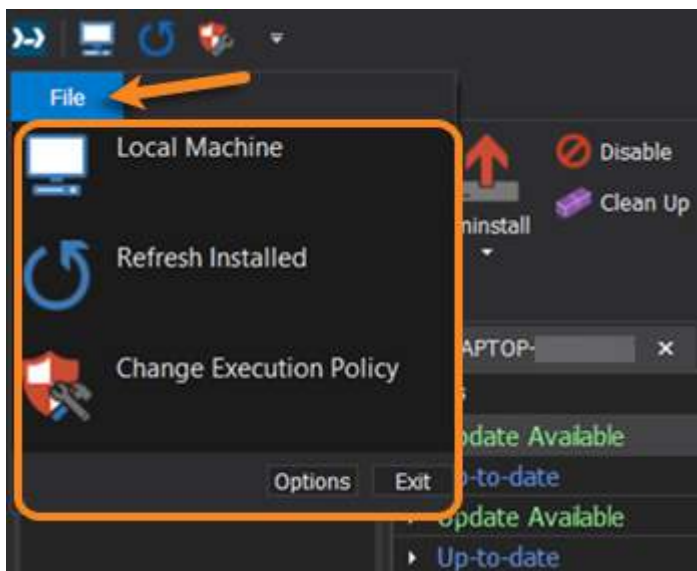
[Home](#)^[108] Controls for searching, installing, and managing modules.

[Tools](#)^[113] Links to external PowerShell and SAPIEN tools.

[Help](#)^[115] Links to product updates, the user manual, and online links to SAPIEN sites and support forums.

5.2.1.1 The File Tab

The File tab on the ribbon has a menu of frequently used functions:



The File Menu

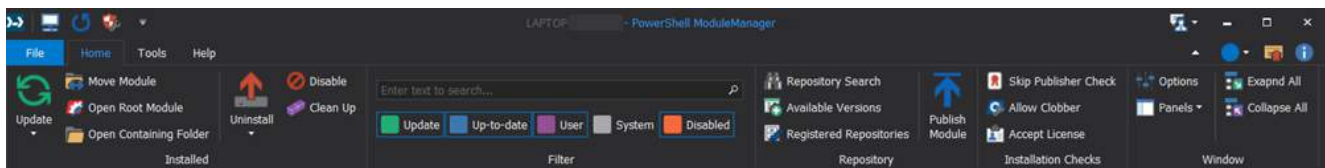
File Menu:

Local Machine	View modules installed on the local machine.
Refresh Installed (F5)	Refresh information for all installed modules.
Change Execution Policy	Opens the Script Security Center GUI interface, where you can change the PowerShell execution policy for both 32-bit and 64-bit shells.
Options	Access the PowerShell ModuleManager program options.
Exit	Close the PowerShell ModuleManager application.

👍 The File menu options are also available on the [Quick Access Toolbar](#)¹¹⁷ above the ribbon.

5.2.1.2 The Home Tab

The **Home** tab contains the functions used to search, install, and manage modules. It also provides access to the PowerShell ModuleManager program options and panels.

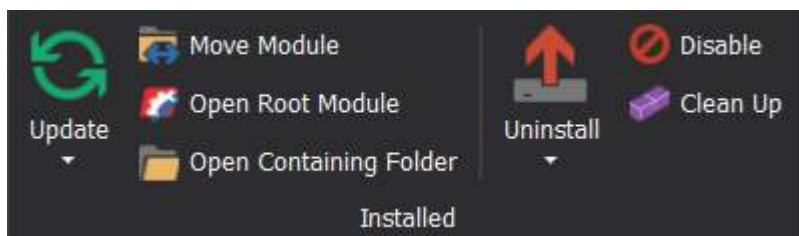


The Home Ribbon

The tools on the Home ribbon are grouped into sections based on related functions.


The Installed section


The **Installed** section on the Home tab has options for managing your installed modules:



Home > Installed

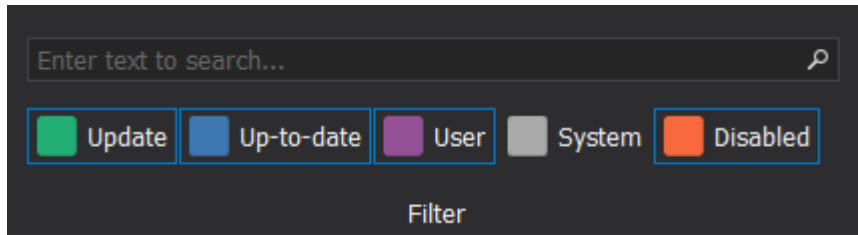
Installed section:

Update (<i>Ctrl+U</i>)	Checks if there is an update for the selected module. If there is, the module is updated.
Update All (<i>Ctrl+Shift+U</i>)	Installs the most recent version of all modules. The update time will vary, depending on the number of installed modules.
Update, Remove Old	Installs the latest version of the selected module, and all previous versions are uninstalled.
Move Module	Opens the Move <i><module></i> dialog where you can designate a destination folder and move the module.
Open Root Module	Opens the default editor allowing you to edit the root module source code.
Open Containing Folder (<i>Ctrl+O</i>)	Opens the folder containing the module.
Uninstall (<i>Ctrl+R</i>)	Uninstalls a specific module version.
Uninstall Older Versions	Keeps the latest installed module version and uninstalls all older versions.
Disable (<i>Ctrl+D</i>)	Disables the selected module.  A highlighted Disable button indicates that the module is <i>disabled</i> ; clicking the highlighted Disable button will <i>enable</i> the module.
Clean Up	Removes older versions of <i>all</i> modules, only keeping the latest versions installed.

 For more information about these controls, view [Updating Modules](#)^[53], [Moving Modules](#)^[59], and [Disabling and Uninstalling modules](#)^[65].

The Filter section

The **Filter** section on the Home tab allows you to search and filter modules installed on your machine:



Home > Filter

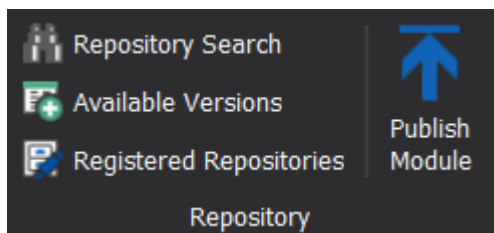
Filter section:

Enter text to search...	Search locally installed modules.
Update	Show modules that have an update available.
Up-to-date	Show modules that have the latest available version installed.
User	Show modules local to your machine, or modules that PowerShell ModuleManager cannot locate on the repository designated in Options ^[99] .
System	Show modules that are in the local system folder.
Disabled	Show modules that are disabled.

👍 For more information about these options, view [Finding Installed Modules](#)^[45].

The Repository section

The **Repository** section on the Home tab allows you to view all available module versions, and you can also search, register, and publish to repositories:



Home > Repository

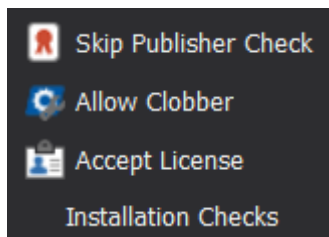
Repository Section:

Repository Search (<i>Ctrl+F</i>)	Search for modules to install from a repository.
Available Versions (<i>Ctrl+V</i>)	Install a specific version of a module from a repository.
Registered Repositories	Manage the PowerShell module repositories registered for the current user.
Publish Module	Publish a module to a repository.

👍 For more information about these options, view [Searching for Modules](#)^[31], [Installing Modules](#)^[35], [Publishing Modules](#)^[60], and [Managing Repositories](#)^[84].

The Installation Checks section

The **Installation Checks** section on the Home tab allows you to control some installation options when installing and updating modules:



*Home > Installation
Checks*

👍 The buttons act like toggles—turning the features on or off.

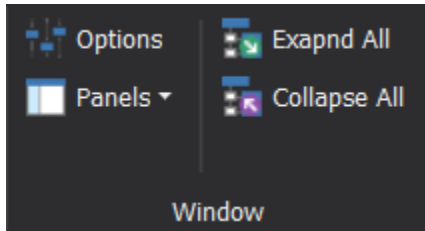
Installation Checks section:

Skip Publisher Check	Allows the installation of a newer module version that is digitally signed by a trusted publisher, replacing an existing, unsigned version.
Allow Clobber	Forces the installation of a module, even when another module contains an identical command.
Accept License	Accepts the license agreement during installation. i This option is not available in older versions of PowerShellGet.

👍 For more information about these options, view [Module Installation Options](#)^[43].

The Window section

The **Window** section on the Home tab allows you to expand or collapse the module information displayed, open the program options, and control the panels displayed in the program window:

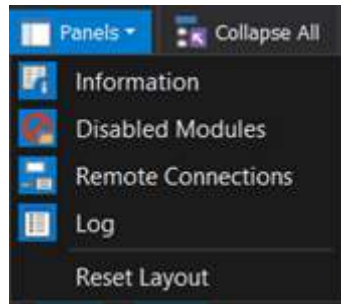


Home > Window

Window section:

Options View the PowerShell ModuleManager program options.

Panels Click the Panels drop-down to display the options:



Information

Click to show or hide the Information panel in the program window.

Disabled Modules

Click to display or hide the Disabled Modules panel in the program window.

Remote Connections

Click to display or hide the Remote Connections panel in the program window.

Log

Click to display or hide the Log panel in the program window.

Reset Layout

Click to restore the panel layout to the default state.

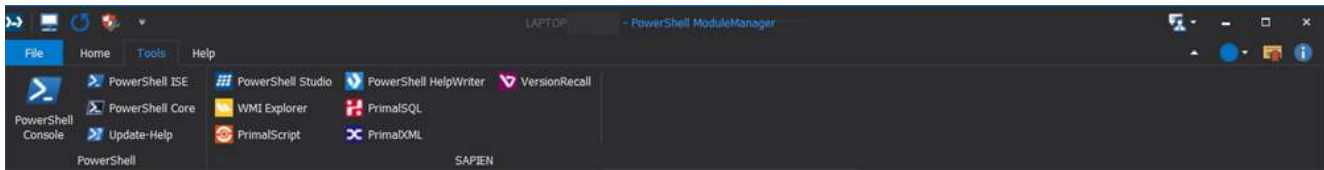
Expand All Expand all rows in the [machine tab](#)^[120] to view information for all installed module versions.

Collapse All Collapse all rows in the [machine tab](#)^[120] to view condensed information for installed modules.

👍 For more information about these options, see [Program Options](#)^[98], [Panels and Layout](#)^[159], and [View Installed Versions](#)^[49].

5.2.1.3 The Tools Tab

The **Tools** tab on the ribbon contains links to PowerShell tools and SAPIEN applications:

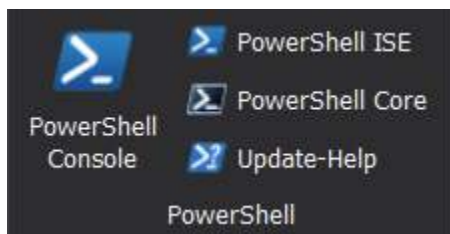


The Tools Ribbon

The Tools ribbon has two sections: PowerShell and SAPIEN.

The PowerShell section

The **PowerShell** section on the Tools tab has buttons to launch PowerShell tools:



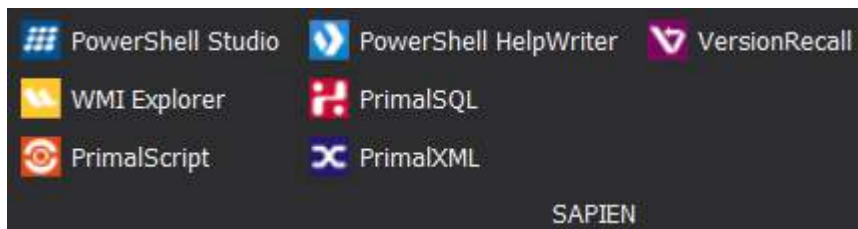
Tools > PowerShell

PowerShell section:

PowerShell Console	Opens the PowerShell Console in a separate window.
PowerShell ISE	Opens the PowerShell ISE in a separate window.
PowerShell Core	Opens the PowerShell Core Console in a separate window.
Update-Help	Updates PowerShell's Help content.

The SAPIEN section

The **SAPIEN** section on the Tools tab has buttons to launch SAPIEN applications:



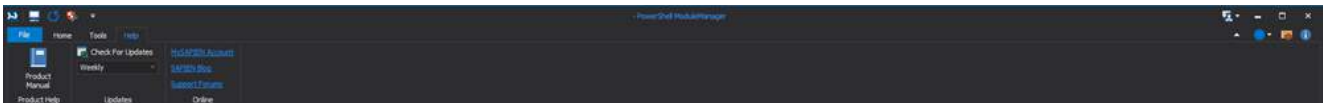
Tools > SAPIEN

SAPIEN section:

PowerShell Studio	Opens the SAPIEN PowerShell Studio application.
WMI Explorer	Opens the SAPIEN WMI Explorer application.
PrimalScript	Opens the SAPIEN PrimalScript application.
PowerShell HelpWriter	Opens the SAPIEN PowerShell HelpWriter application.
PrimalSQL	Opens the SAPIEN PrimalSQL application.
PrimalXML	Opens the SAPIEN PrimalXML application.
VersionRecall	Opens the SAPIEN VersionRecall application.

5.2.1.4 The Help Tab

The **Help** tab on the ribbon contains links to the PowerShell ModuleManager help manual, SAPIEN product updates, and SAPIEN online:

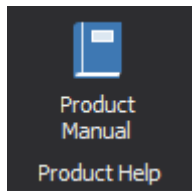


The Help Ribbon

The Help ribbon has three sections: Product Help, Updates, and Online.

The Product Help section

The **Product Help** section on the Help tab has a link to the PowerShell ModuleManager user manual:



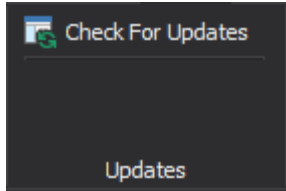
*Help >
Product Help*

Product Help section:

Product Manual	Opens the PowerShell ModuleManager help manual.
----------------	---

The Updates section

The **Updates** section on the Help tab contains product update options:



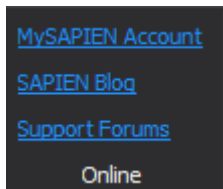
Help > Updates

Updates section:

Check For Updates	Opens the SAPIEN Updates  tool.
-------------------	---

The Online section

The **Online** section on the Help tab has links to SAPIEN support websites:



Help > Online

Online section:

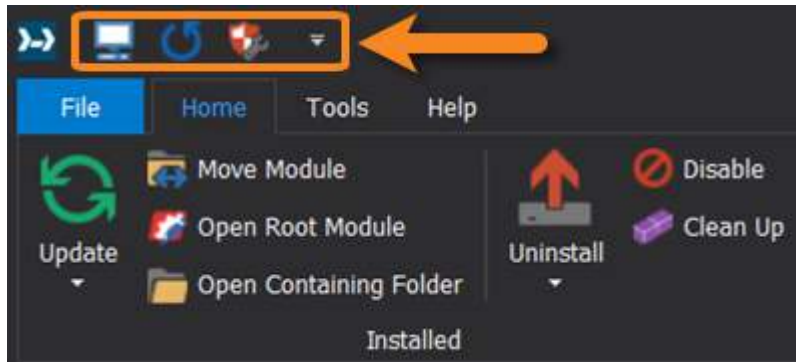
MySAPIEN Account	Opens your SAPIEN Account page in a web browser.
SAPIEN Blog	Opens the SAPIEN blog in a web browser.
Support Forums	Opens the SAPIEN Support page in a web browser.

5.2.2 Other Toolbars

This section provides a reference for the smaller ribbons and toolbars.

Quick Access Toolbar

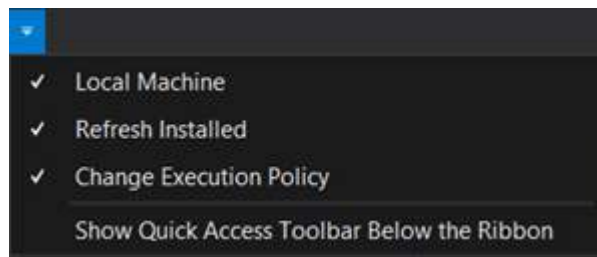
The configurable Quick Access Toolbar on the top-left of the program window provides direct access to frequently used functions:



Quick Access Toolbar

Button details (from left to right):

Local Machine	View modules installed on the local machine ¹²⁰ .
Refresh Installed (<i>F5</i>)	Refresh information for all installed modules.
Change Execution Policy	Opens the Script Security Center GUI interface, where you can change the PowerShell execution policy for both 32-bit and 64-bit shells.
Customize Quick Access Toolbar	Click the drop-down icon on the right of the toolbar to display the options:



Local Machine

Select or deselect to show the Local Machine button on the toolbar.

Refresh Installed

Select or deselect to display the Refresh Installed button on the toolbar.

Change Execution Policy

Select or deselect to display the PowerShell Execution Policy button on the toolbar.

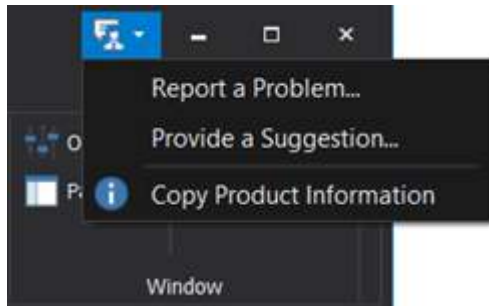
Show Quick Access Toolbar Below the Ribbon

Click to display the Quick Access Toolbar below the ribbon.

 See how to [customize the Quick Access Toolbar](#) ¹⁵⁵.

Send Feedback

The Send Feedback menu is on the top-right with the program window controls. The Send Feedback menu contains options to report a problem, submit a suggestion, or copy the product information:



Send Feedback Menu

Menu options (top to bottom):

Report a Problem...	Report a problem or ask a product-related question. <ul style="list-style-type: none">• Registered customers are directed to the Product Support forums.• Users of trial versions are directed to the Trial Software / Pre-sales Technical Questions forum.
Provide a Suggestion...	Submit a suggestion on the Feature Requests page.
Copy Product Information	Copy your PowerShell ModuleManager build number and OS information to the clipboard so that you can paste it into a new topic in the SAPIEN forums.

👍 When reporting a PowerShell ModuleManager problem or submitting a question, use the Copy Product Information option to copy the product build number and OS information to your clipboard, and then paste this information into your forum post.

Ribbon Header Buttons

There are four ribbon header buttons on the right above the ribbon:



Ribbon Header Buttons

Button details (from left to right):

Minimize the Ribbon (*Ctrl+F1*)

Toggle the ribbon toolbar:

Minimize:

Only show tab names on the ribbon.

Expand:

Show the ribbon so that it is always expanded even after you click a command.

Skins

Select color-themed [skins](#)^[25].

License Information


Display and edit your current license information.

About

Display production information.

Status Bar

The status bar on the bottom of the PowerShell ModuleManager workspace displays the number of installed modules:



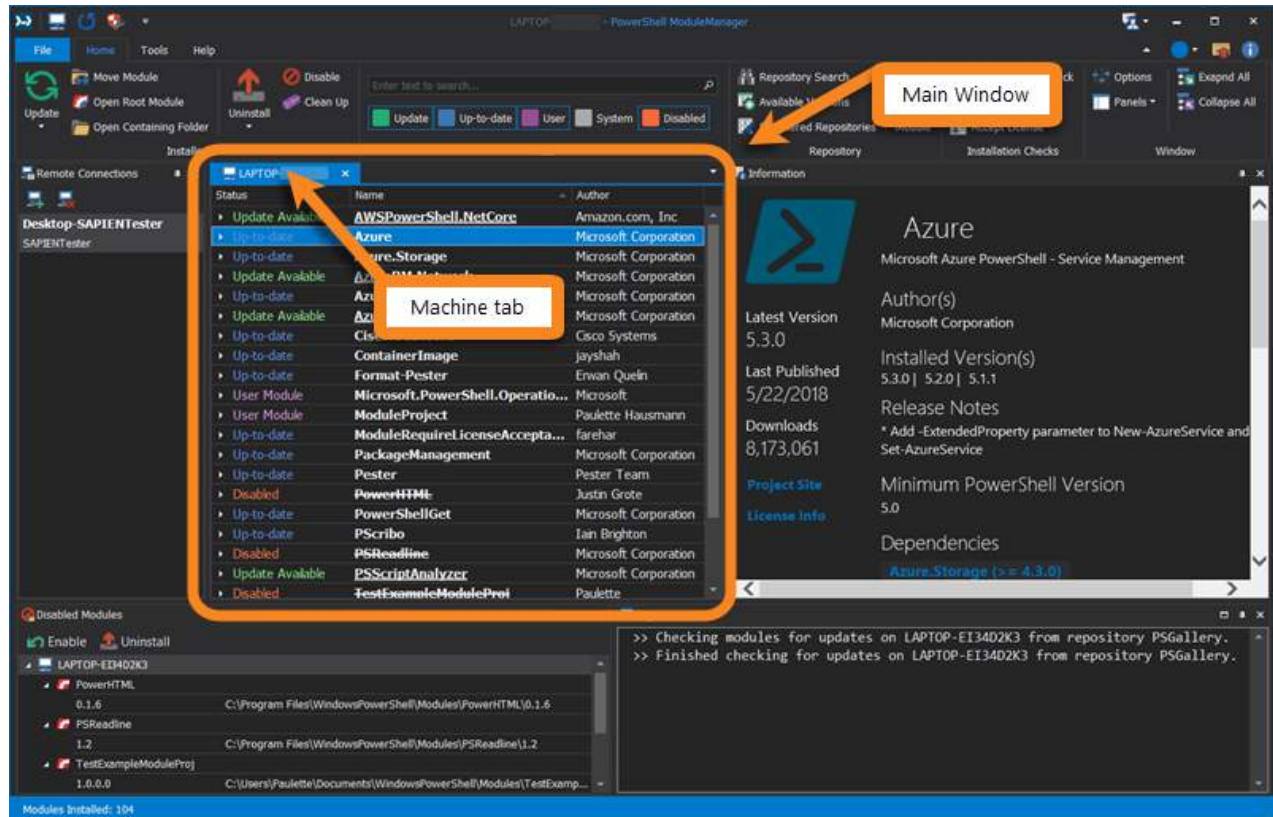
Modules Installed: 103

5.2.3 Machine Tab and Main Window

Installed modules are managed in the [Machine tab](#)^[120], which is centrally located in the [main window](#)^[123] of the PowerShell ModuleManager workspace. Both local and remote machines will appear as tabs with the name of the machine.

Machine Tab

A 'machine tab' is either a [local](#)^[15] or [remote machine](#)^[92], represented as a tab in the main window of the PowerShell ModuleManager workspace. When you launch PowerShell ModuleManager, by default the local machine appears as a machine tab:



Machine tab (Local Machine) in the Main Window

The machine tab is where you view and [manage installed modules](#) ⁴⁵.

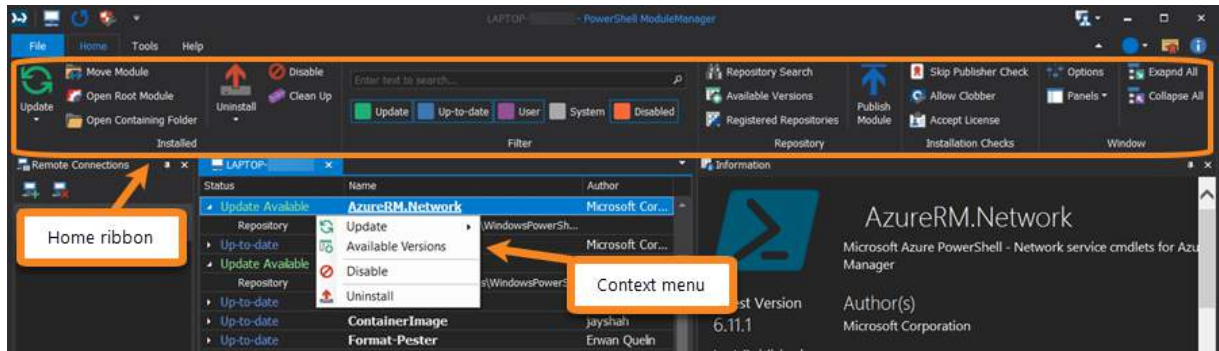
How to access module options

To access options for installed modules

1. Highlight a module in the machine tab
2. Select the desired control on the Home ribbon

-OR-

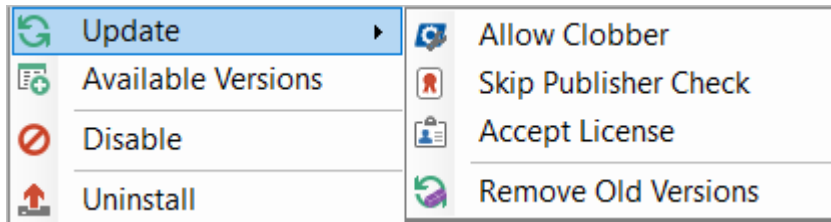
Right-click and select an option in the context menu



i The options available on the Home ribbon and the context menu will vary, depending on whether the tab selected is a local machine or a remote connection.

Machine tab context menu

Right-click a module in the machine tab to display the menu options:



Menu options

Update (*Ctrl+U*)

Update

Checks if there is an update for the selected module. If an update is available, the module is updated.

Update > Allow Clobber


Allows the installation of a newer module version that is digitally signed by a trusted publisher, replacing an existing, unsigned version.

Update > Skip Publisher Check

Forces the installation of a module, even when another module contains an identical command.

Update > Accept License

Accepts the license agreement during installation.

 This option is not available in older versions of PowerShellGet.

Update > Remove Old Versions

Installs the latest version and uninstalls all previous versions.


Available Versions (*Ctrl+V*)

Retrieves all available versions from the set repository.

 You can install modules directly from the [Versions dialog](#)  ¹⁵⁰.




Disable (*Ctrl+D*)

Disables the selected module.

 The Disable feature behaves like a toggle.

Uninstall (*Ctrl+R*)

Removes the module from the machine.

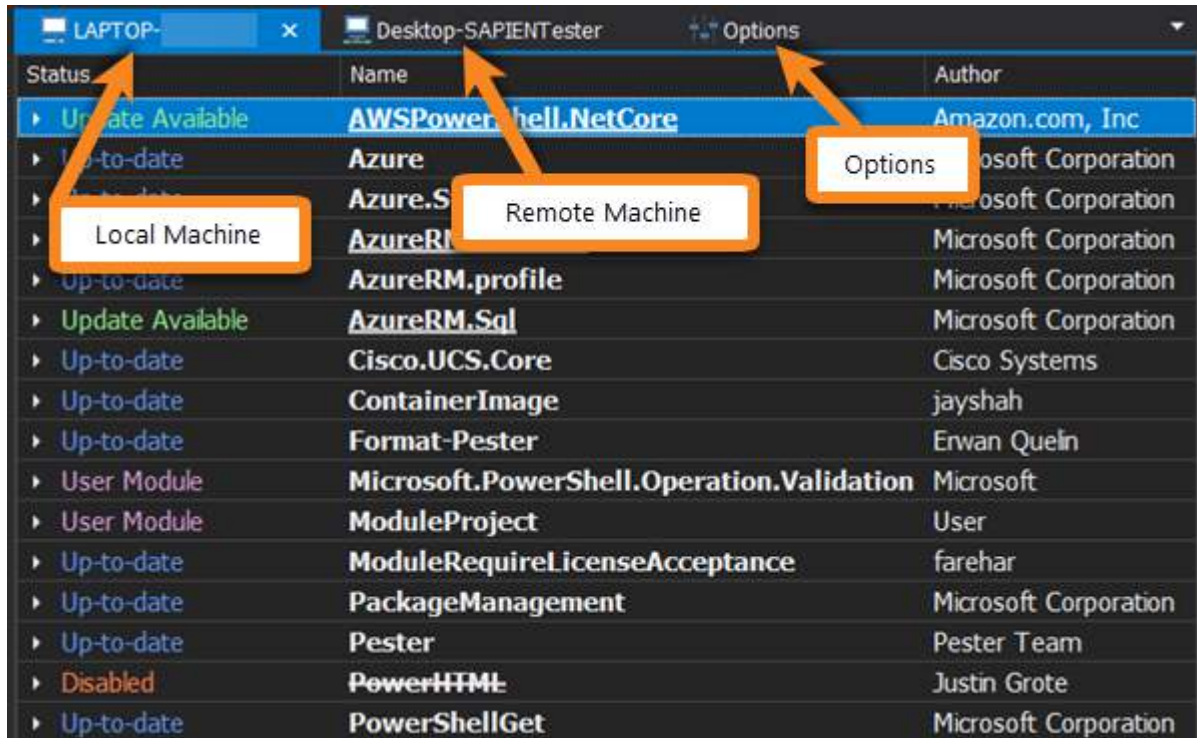
 The options available on the context menu will vary, depending on whether the tab selected is a [local machine](#)  ¹⁵¹ or a [remote connection](#)  ⁹².

See Finding Installed Modules for details on how to search, sort, filter, and view all modules in the machine tab.

Main Window

The **main window** provides a centralized location for managing modules installed on your local and remote machines. The main window also displays the program Options.

The local machine, remote machines, and program options display as tabs in the main window:

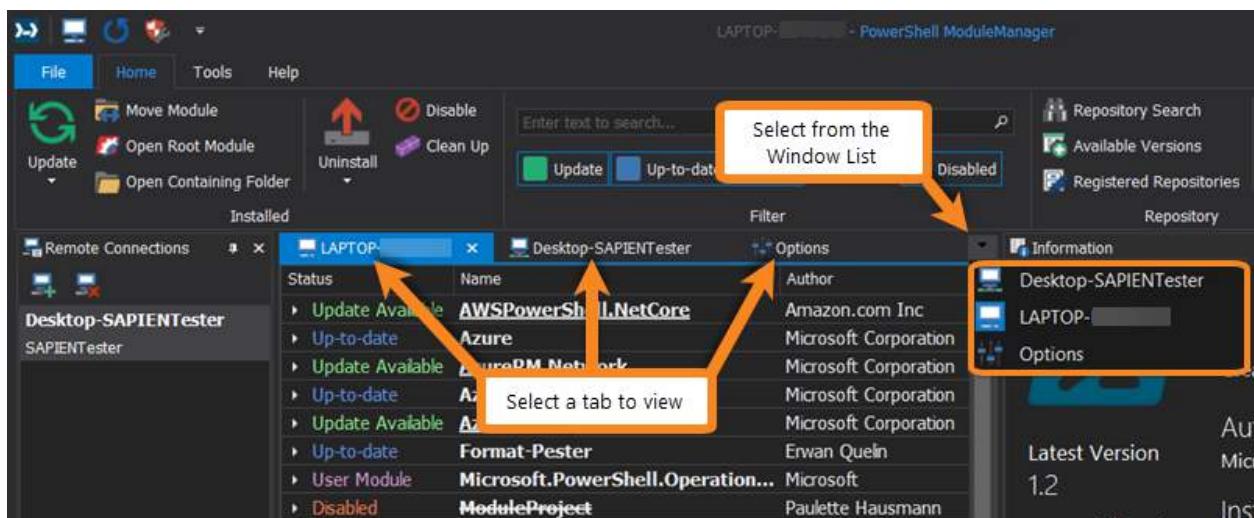


Main Window - Local Machine tab, Remote Machine tab, Options tab

How to navigate between tabs

To navigate between tabs in the main window


- Click the desired tab, or select an option in the Window List drop-down on the top-right:

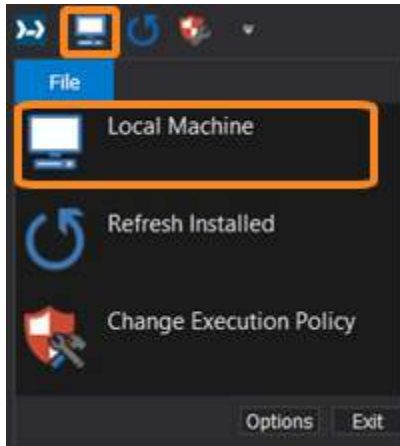


How to view the local machine

To view the local machine

- By default, the local machine is always visible in the main window.

👍 If the local machine tab is not visible in the main window you can reopen it by clicking the  button on either the Quick Access Toolbar or the File menu (**File > Local Machine**):

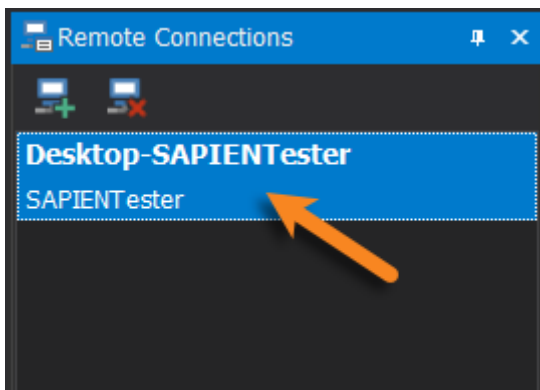


How to view a remote machine

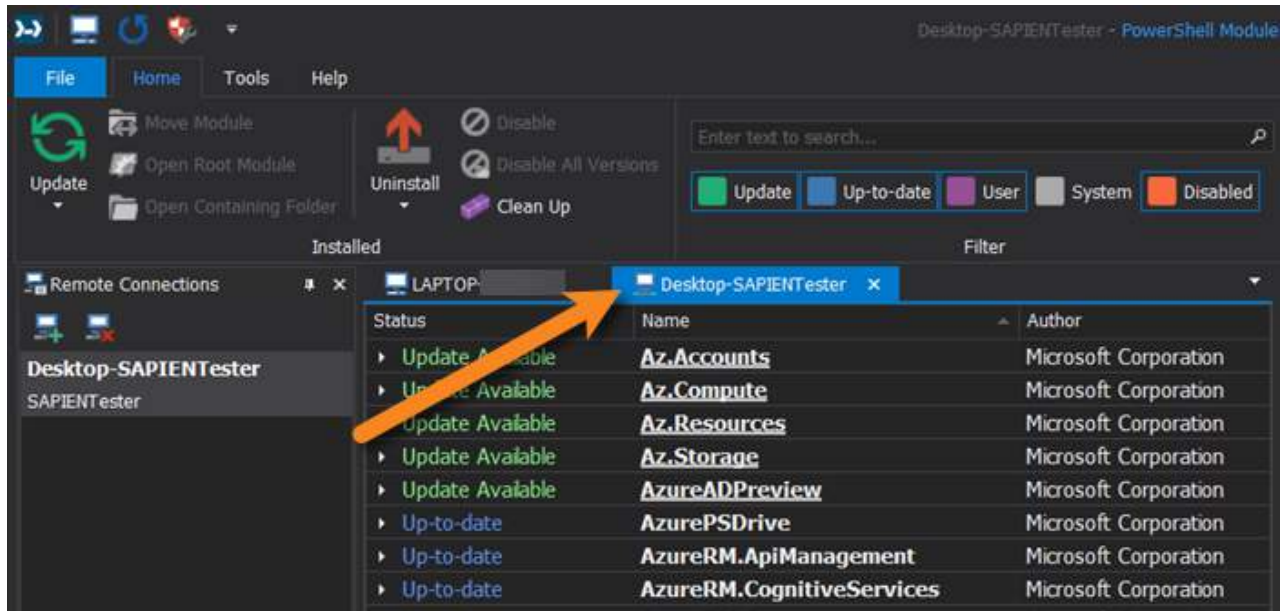
- To view a remote machine, you must first [add a remote connection](#) ⁹³.

To view a remote machine

- In the Remote Connections panel, double-click on an existing connection:



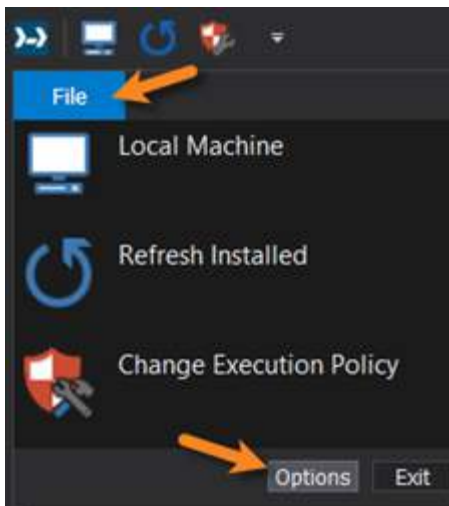
A new tab with the computer's name will appear in the main window when the remote machine is connected:



How to view the program options

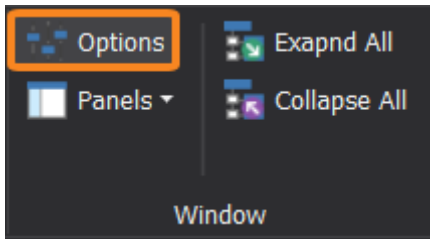
To see the program options

- On the ribbon, select **File > Options**:



-OR-

- On the ribbon, select **Home > Options**:

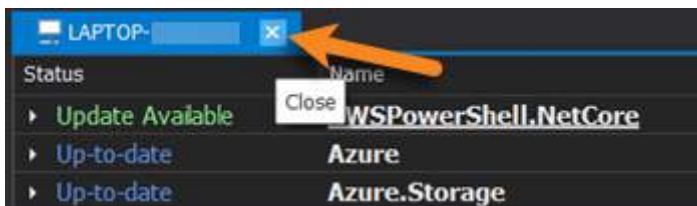



The program Options will appear as a tab in the main window.

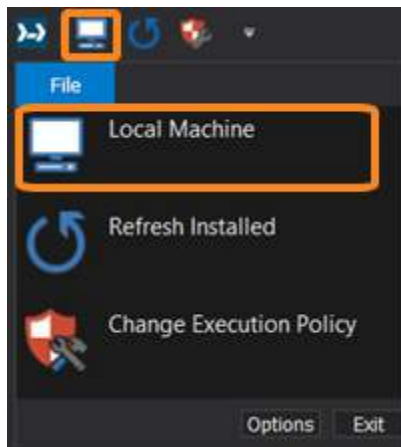
How to close a tab in the main window

To close a tab in the main window

- Click the X on the right of the tab:

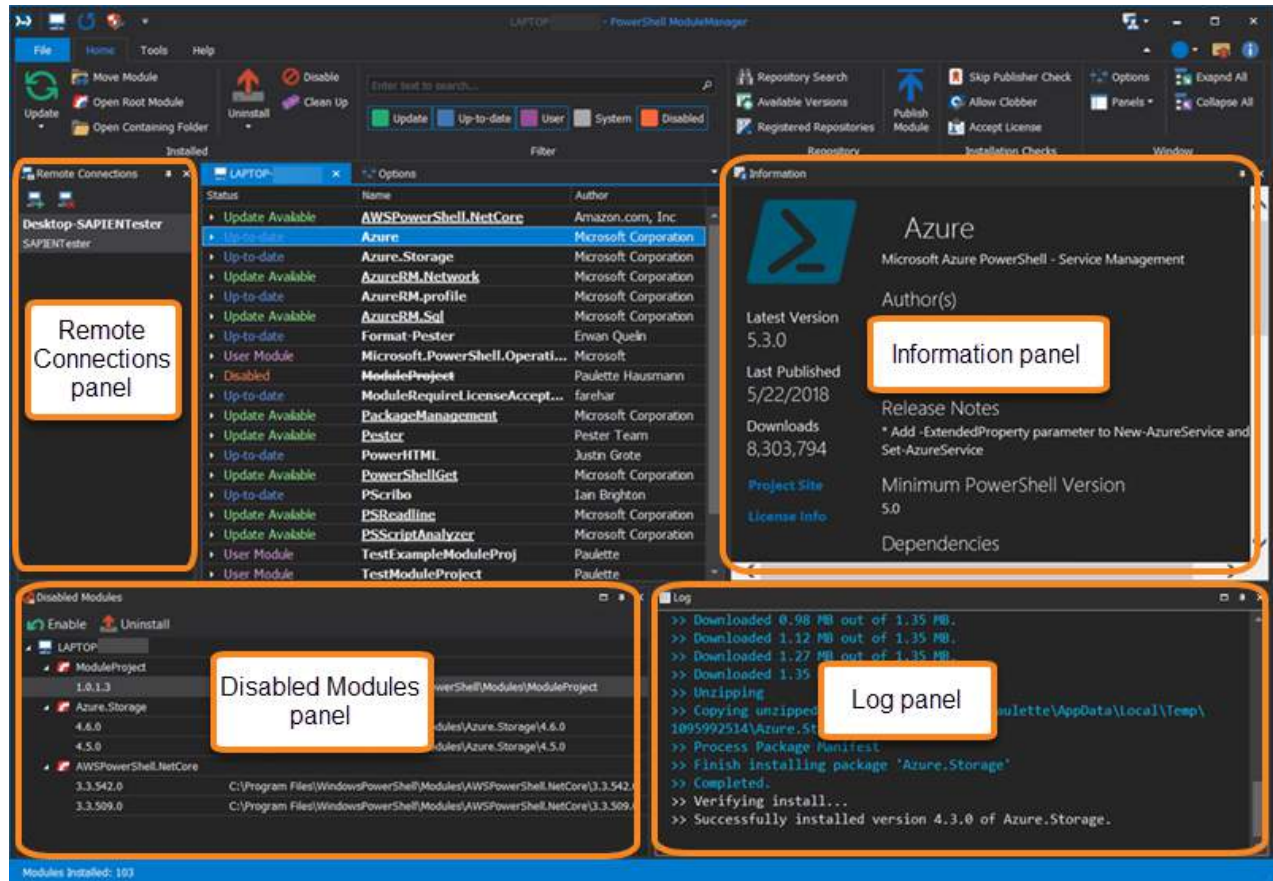


👍 If the local machine tab is not visible in the main window you can reopen it by clicking the  button on either the Quick Access Toolbar or the File menu (**File > Local Machine**):



5.2.4 Panels

PowerShell ModuleManager has dockable window panels that are used by dedicated features or to display output. This topic provides an overview of the available panels and their content.



PowerShell ModuleManager Panels

PowerShell ModuleManager panels

[Disabled Modules](#) ¹²⁹

Displays disabled modules. Use this panel to enable and uninstall modules.

[Information](#) ¹³³

Displays detailed information for modules highlighted in the machine tab.

[Log](#) ¹³⁵

Displays PowerShell ModuleManager output messages.

[Remote Connections](#) ¹³⁷

Displays remote connections. Use this panel to add, remove, and connect to remote machines.

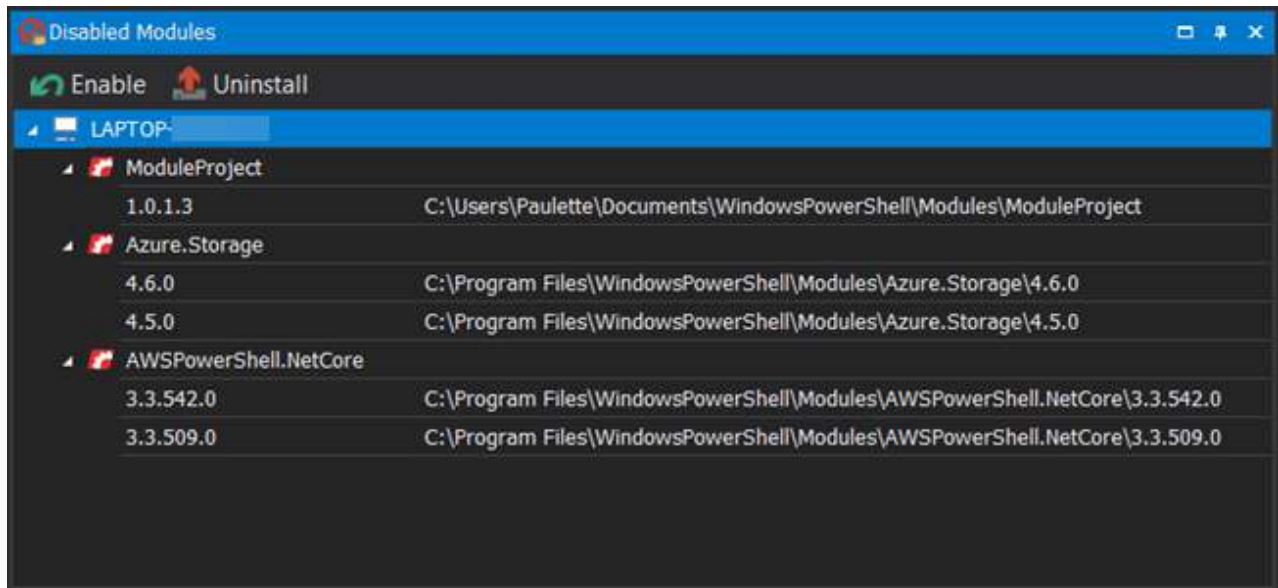
i For information about working with panels, see Customizing Your Workspace > [Panels and Layout](#) ¹⁵⁹.

5.2.4.1 Disabled Modules Panel



Disabled modules can be viewed and managed in the Disabled Modules panel.

The Disabled Modules Panel

The Disabled Modules panel is in the bottom-left of the PowerShell ModuleManager workspace:



Disabled Modules Panel

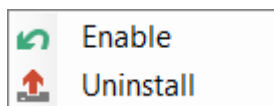
👍 You can [enable](#)^[130] or [uninstall](#)^[132] modules by selecting a module and clicking the  or  button in the top-left of the panel, or by right-clicking and selecting from the [context menu](#)^[129].

Understanding how the Disable feature works

The PowerShell ModuleManager **Disable** feature moves a module from the path saved in the value of the PSModulePath environment variable (\$env:PSModulePath). Disabling a module is not the same as *uninstalling*. Disabling a module moves it to a user's AppData folder, and enabling moves it back to the previous location. If modules are no longer needed, they should be uninstalled.

Disabled Modules panel context menu

Right-click a module in the Disabled Modules panel to display the following options:



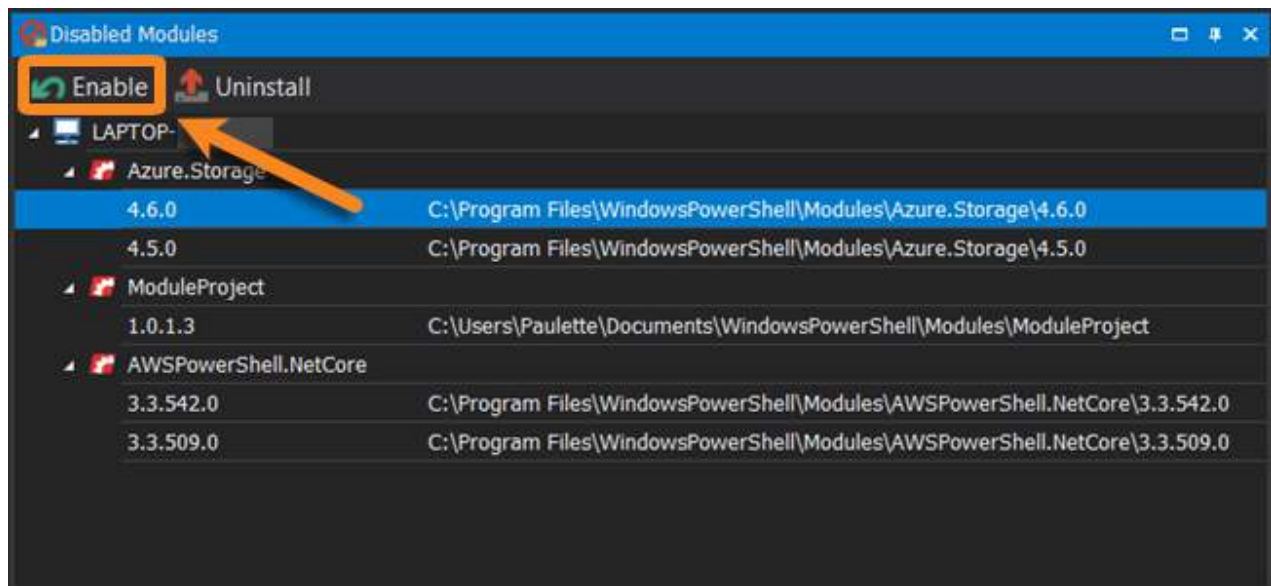
Menu options:

- | | |
|-----------|---|
| Enable | Move the module back to its original directory. |
| Uninstall | Remove the module from the machine. |

How to enable modules

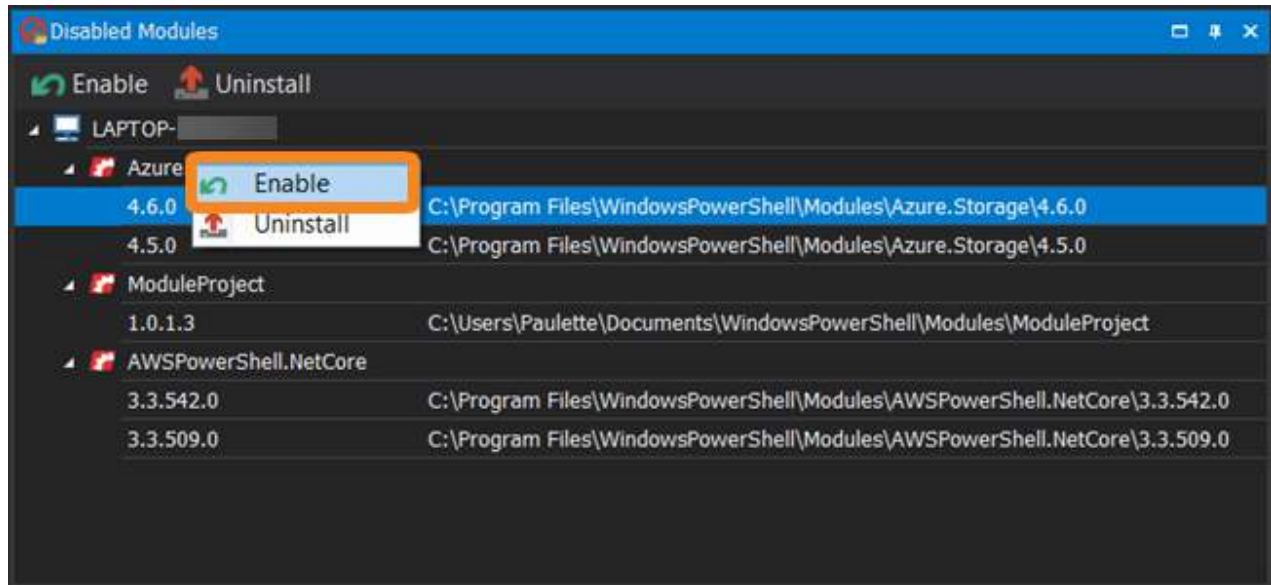
To enable a module

- Select a module, and then click the **Enable** button:



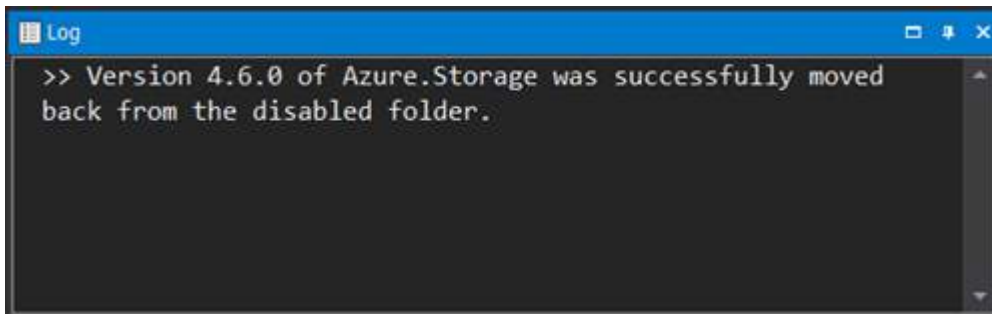
-OR-

- Right-click on a module, and then select **Enable** from the context menu:



👍 Selecting the parent node enables all versions; selecting a child node enables only that version.

The enabled status is displayed in the Log panel:



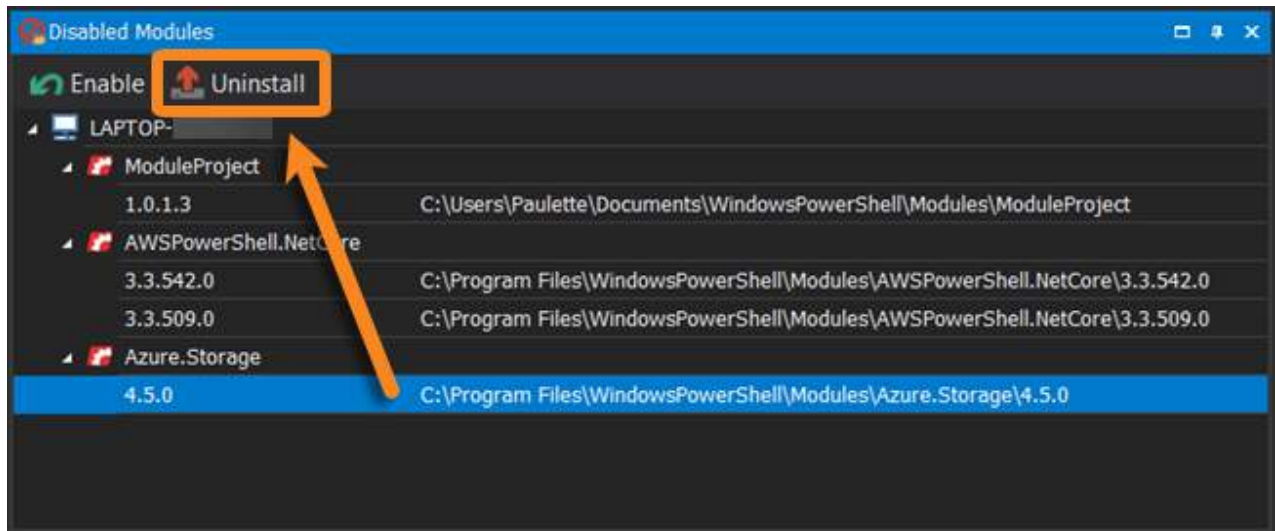
The Machine tab will refresh to display the enabled module(s):

Status	Name	Author
Disabled	AWSPowerShell.NetCore	Amazon.com, Inc
Up-to-date	Azure	Microsoft Corporation
Up-to-date	Azure.AnalysisServices	Microsoft Corporation
Up-to-date	Azure.Storage	Microsoft Corporation
Repository	4.6.1 - C:\Program Files\WindowsPowerShell\Modules\Azure.Stora...	
Repository	4.6.0 - C:\Program Files\WindowsPowerShell\Modules\Azure.Stora...	
Repository	(Disabled) 4.5.0 - C:\Program Files\WindowsPowerShell\Modules\A...	
Update Available	AzureRM.Network	Microsoft Corporation
Up-to-date	AzureRM.profile	Microsoft Corporation

How to uninstall modules

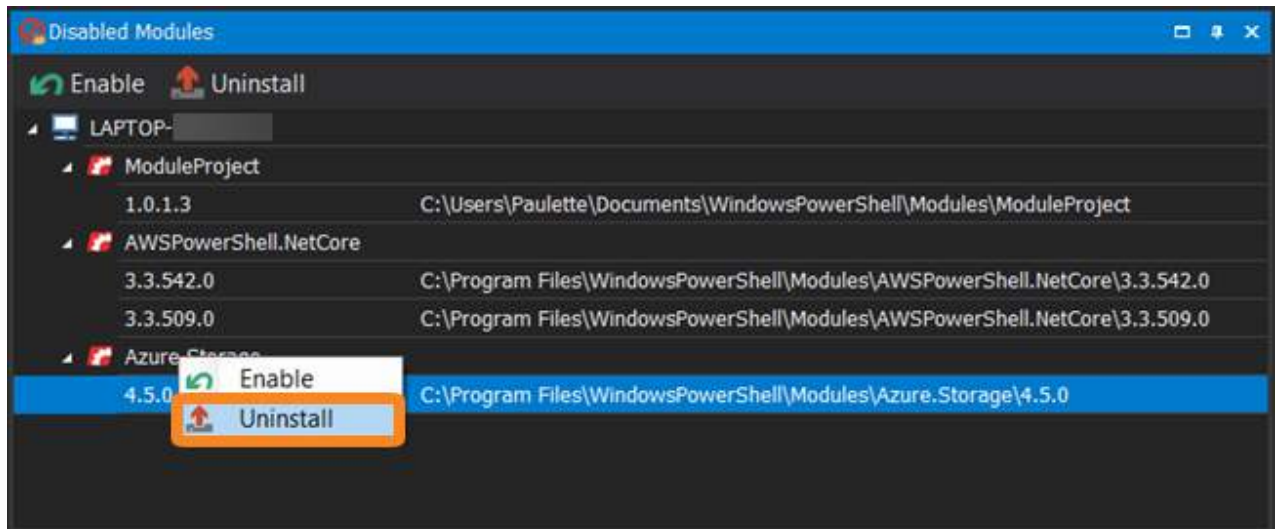
To uninstall a module

- Select a module, and then click the **Uninstall** button:



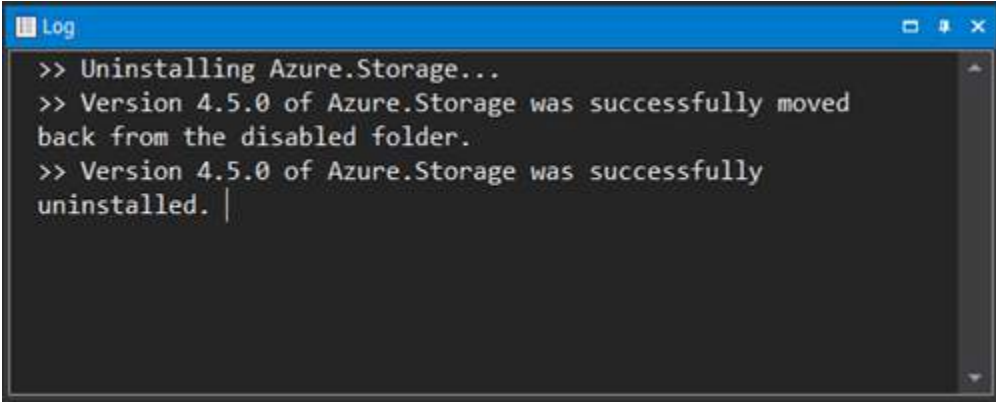
-OR-

- Right-click on a module, and then select **Uninstall** from the context menu:



👍 Selecting the parent node uninstalls all versions; selecting a child node uninstalls only that version.

The uninstall status is displayed in the Log panel:

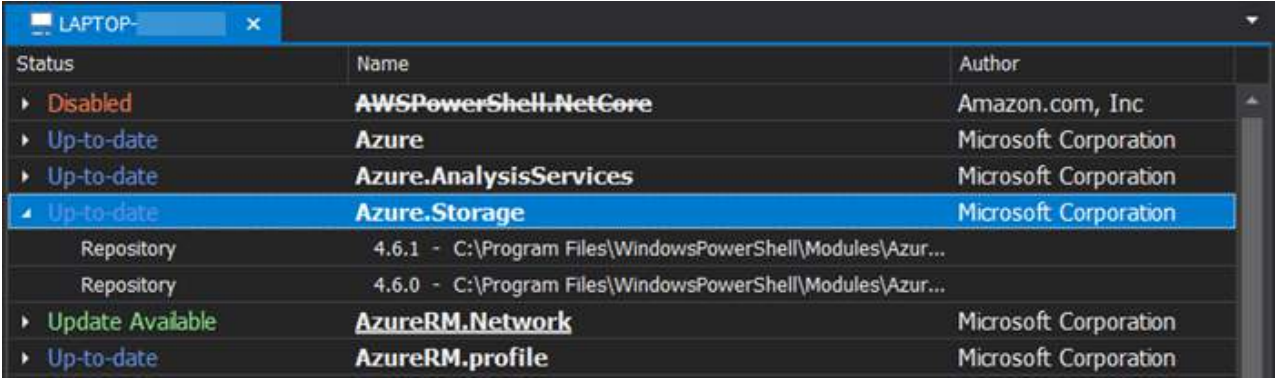


```


>> Uninstalling Azure.Storage...
>> Version 4.5.0 of Azure.Storage was successfully moved
back from the disabled folder.
>> Version 4.5.0 of Azure.Storage was successfully
uninstalled. |

```

The Machine tab will refresh, and the uninstalled module(s) will not be listed:



Status	Name	Author
Disabled	AWSPowerShell.NetCore	Amazon.com, Inc
Up-to-date	Azure	Microsoft Corporation
Up-to-date	Azure.AnalysisServices	Microsoft Corporation
Up-to-date	Azure.Storage	Microsoft Corporation
	Repository	4.6.1 - C:\Program Files\WindowsPowerShell\Modules\Azur...
	Repository	4.6.0 - C:\Program Files\WindowsPowerShell\Modules\Azur...
Update Available	AzureRM.Network	Microsoft Corporation
Up-to-date	AzureRM.profile	Microsoft Corporation

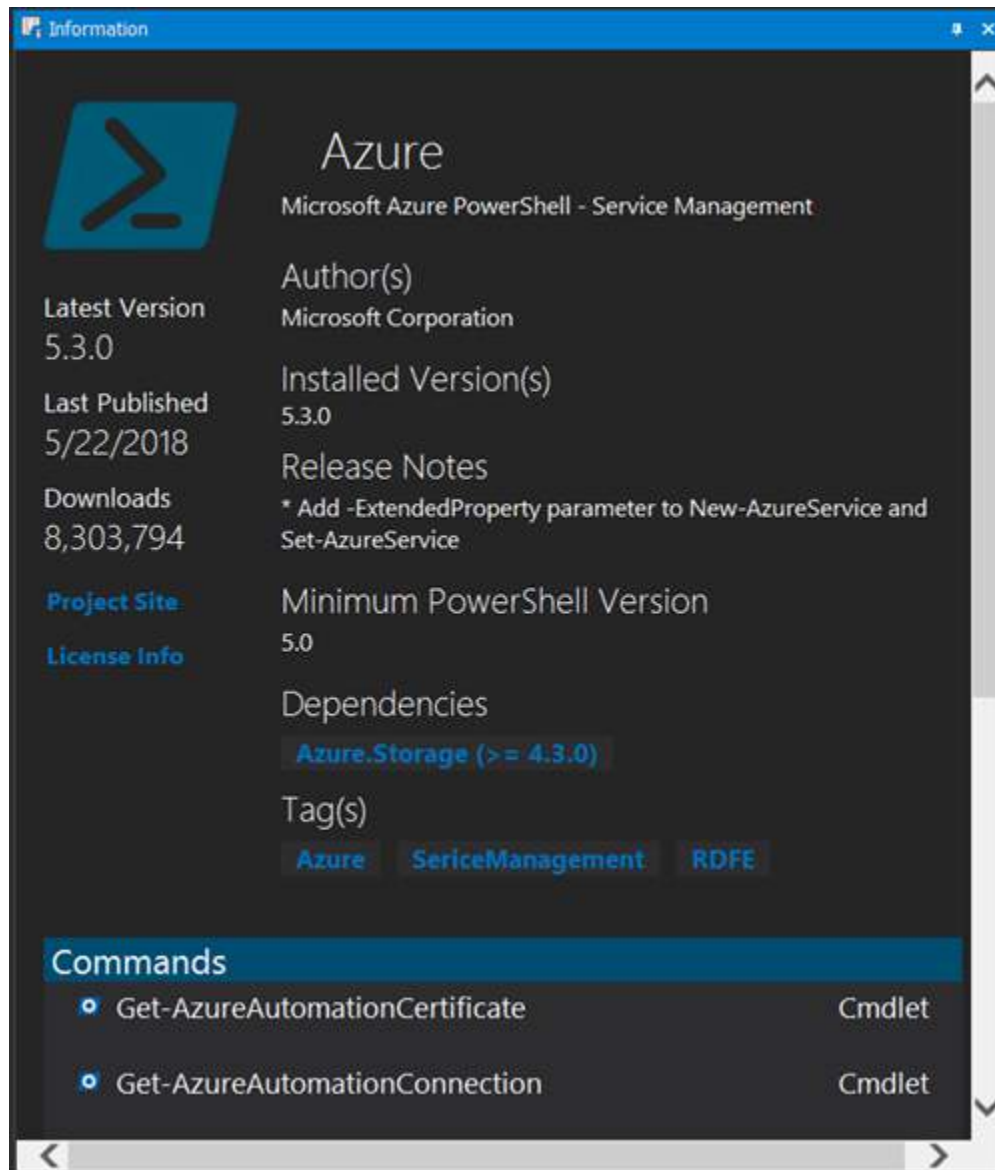
 To learn more, view Managing Modules > Managing Installed Modules > [Disabling and Uninstalling Modules](#) ⁶⁵.

5.2.4.2 Information Panel

Detailed information for installed modules appears in the Information panel.

The Information Panel


The Information panel is located in the middle of the PowerShell ModuleManager program window, on the right side:



Information panel

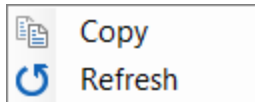
- i** Select a module in the [machine tab](#)^[15] to view the details in the Information panel.

Information panel contents

Description	A short description of the module.
Latest Version	The latest version of the module available in the repository.
Last Published	The last date the module was updated in the repository.
Installed Version(s)	The version(s) of the module installed on your machine.
Minimum PowerShell Version	The minimum PowerShell version required to run the module.
Dependencies	If dependent module(s) are listed, the dependent module version(s) will also be installed.
Tags	Keywords used to search for modules in PowerShell repositories.  Click on a Tag to open a search for modules with the same tag.
Commands	Cmdlets, functions, or aliases contained in the module.

Information panel context menu

Right-click in the Information panel to display the following options:



Menu options:

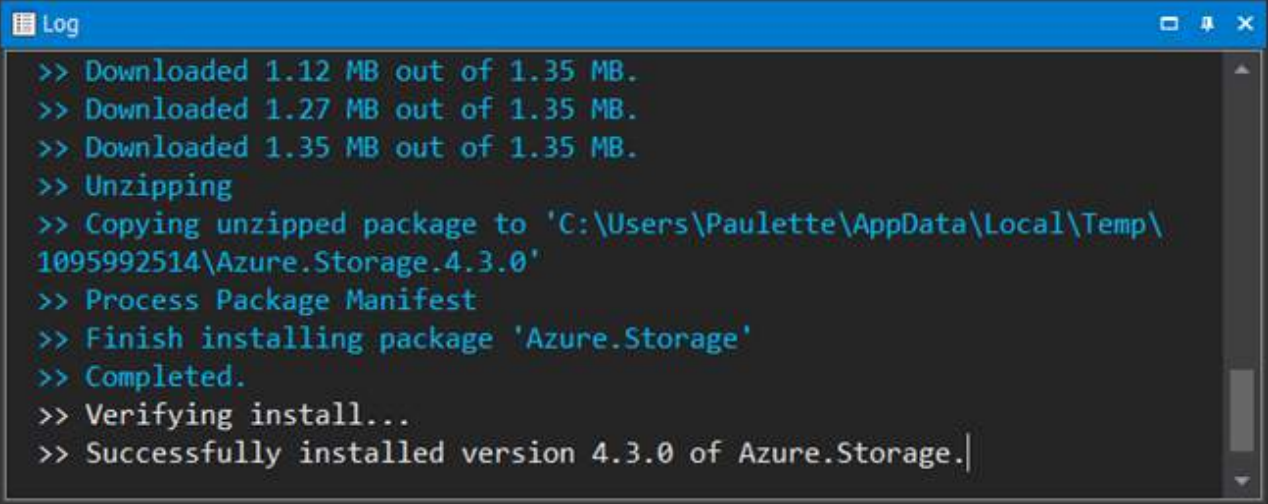
Copy	Copy text highlighted in the Information panel to the clipboard.
Refresh	Refresh the content displayed in the Information panel.

5.2.4.3 Log Panel

The Log panel displays all of the PowerShell ModuleManager status and output messages.

The Log Panel

The Log panel is located on the bottom-right of the PowerShell ModuleManager workspace:



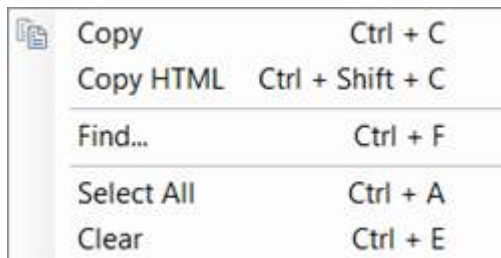
The screenshot shows a window titled 'Log' with a blue header bar. The log text is as follows:

```
>> Downloaded 1.12 MB out of 1.35 MB.
>> Downloaded 1.27 MB out of 1.35 MB.
>> Downloaded 1.35 MB out of 1.35 MB.
>> Unzipping
>> Copying unzipped package to 'C:\Users\Paulette\AppData\Local\Temp\1095992514\Azure.Storage.4.3.0'
>> Process Package Manifest
>> Finish installing package 'Azure.Storage'
>> Completed.
>> Verifying install...
>> Successfully installed version 4.3.0 of Azure.Storage.
```

Log Panel

Log panel context menu

Right-click on the Log panel to display the following options:

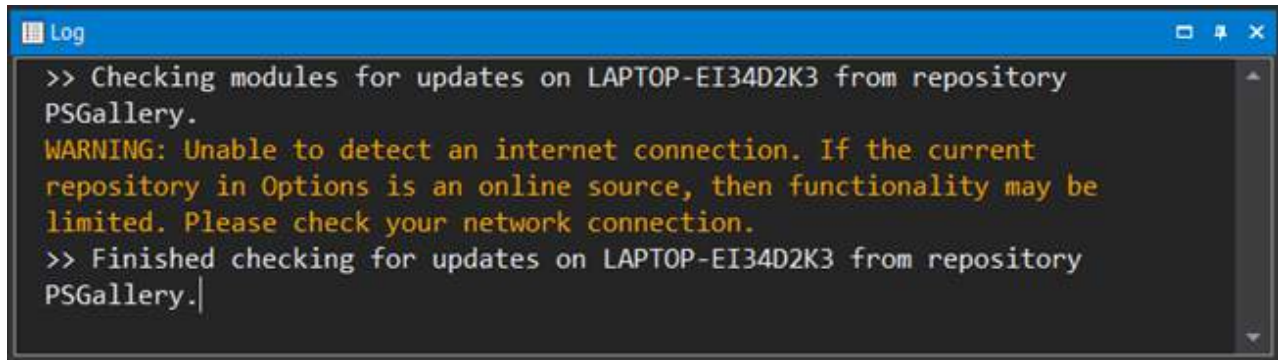


Menu options:

Copy (<i>Ctrl+C</i>)	Copy text highlighted in the Log panel to the clipboard.
Copy HTML (<i>Ctrl+Shift+C</i>)	Copy highlighted text as HTML, including color-coding and formatting.
Find... (<i>Ctrl+F</i>)	Search the text in the Log panel.
Select All (<i>Ctrl+A</i>)	Select all of the text in the Log panel.
Clear (<i>Ctrl+E</i>)	Clear the text in the Log panel.

Internet Connection Warning

! A warning will be displayed in the Log panel if the PowerShell repository designated in [Options](#)^[99] is an online source and an internet connection is not detected:



```
Log
>> Checking modules for updates on LAPTOP-EI34D2K3 from repository
PSGallery.
WARNING: Unable to detect an internet connection. If the current
repository in Options is an online source, then functionality may be
limited. Please check your network connection.
>> Finished checking for updates on LAPTOP-EI34D2K3 from repository
PSGallery.
```

5.2.4.4 Remote Connections Panel



You can manage remote machine connections in the Remote Connections panel.

The Remote Connections Panel

The Remote Connections panel is in the middle-left of the PowerShell ModuleManager program window:

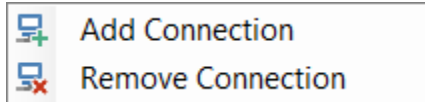


Remote Connections Panel

👍 You can add or remove remote connections by clicking the  or  button in the top-left of the panel, or by right-clicking and selecting from the [context menu](#) ¹³⁸.

Remote Connections panel context menu

Right-click on the Remote Connections panel to display the following options:



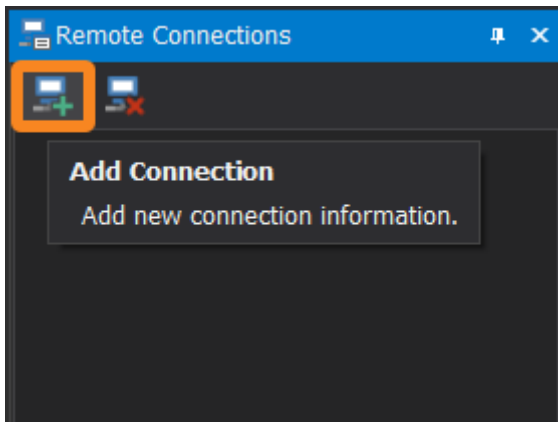
Menu options:

Add Connection	Add a new connection.
Remove Connection	Delete the selected connection.

How to add a remote machine

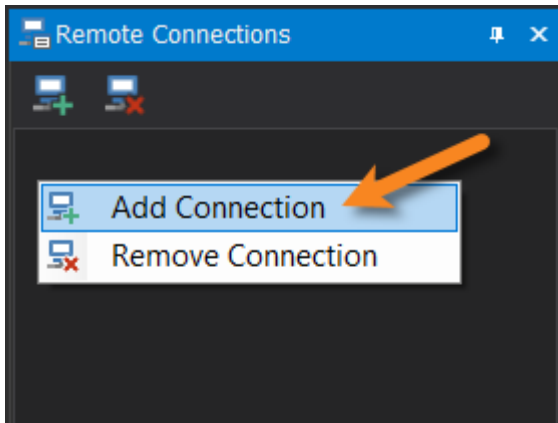
To add a remote connection

- In the Remote Connections panel, click the **Add Connection** button:

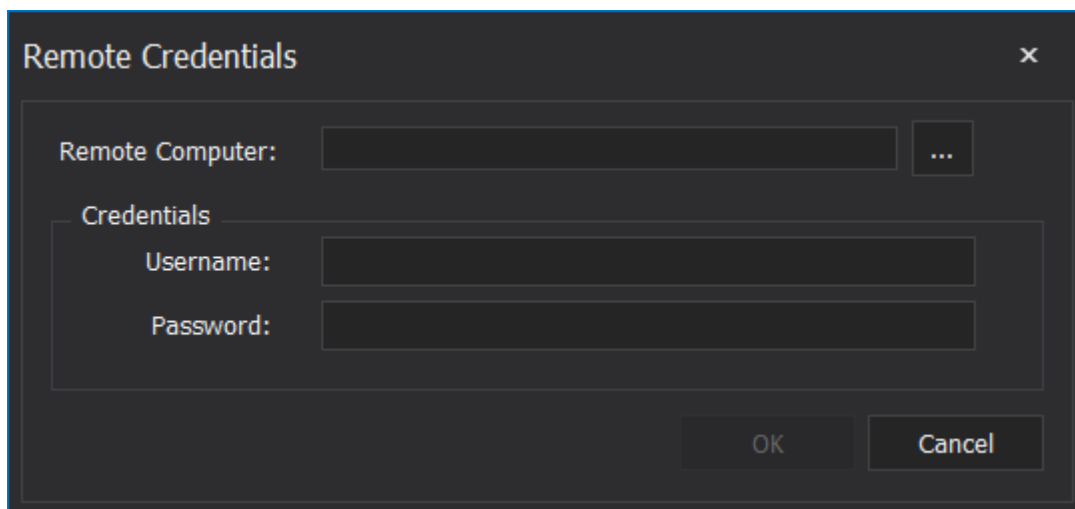


-OR-

- Right-click on the Remote Connections panel, then select **Add Connection**:



The Remote Credentials dialog will open:

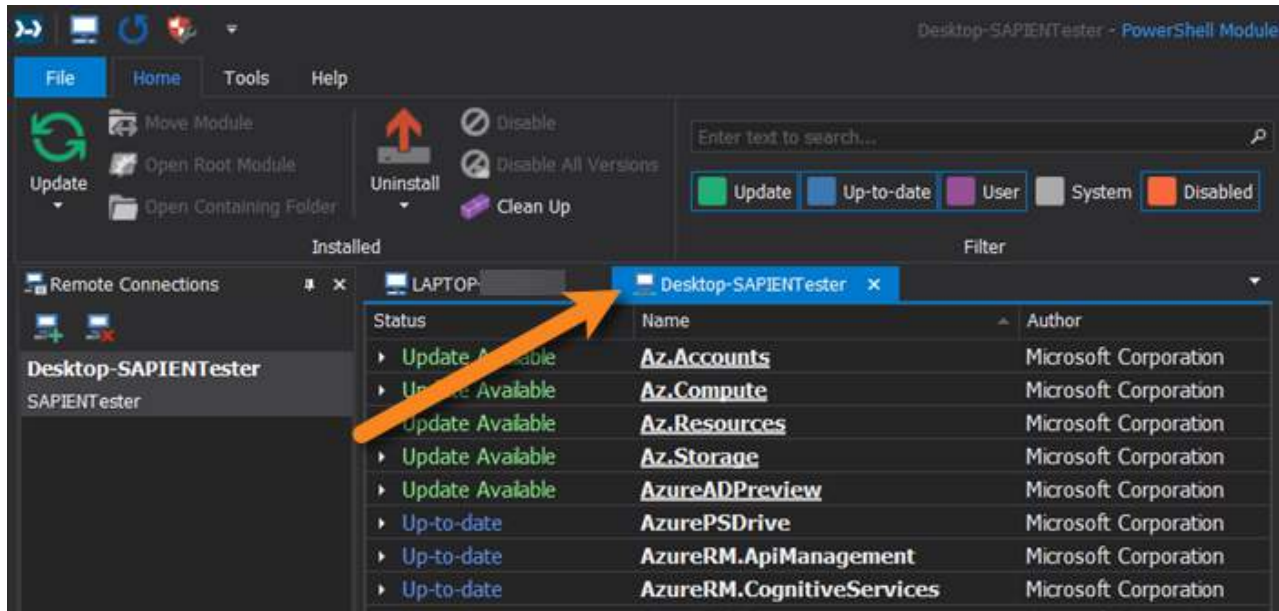


Remote Credentials Dialog

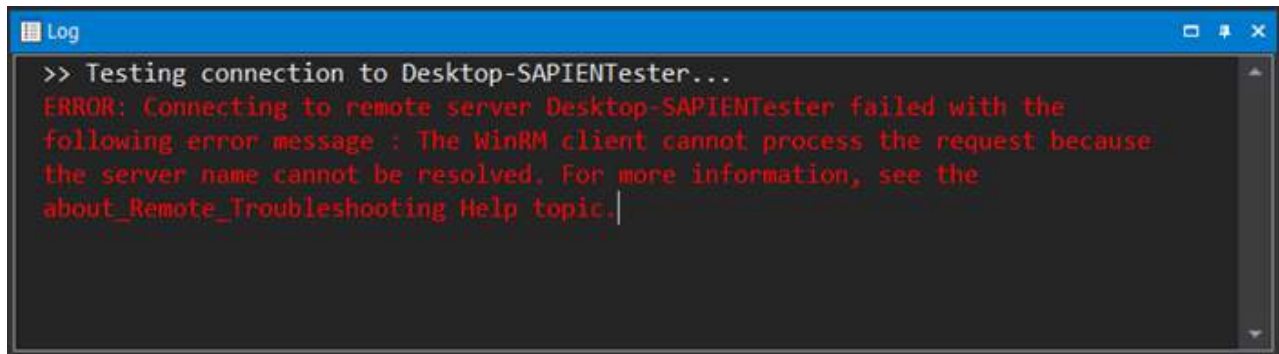
In the Remote Credentials dialog:

1. Enter the computer name in the Remote Computer field, or search connected computers (...).
2. Enter the credentials and click **OK** to add the remote computer.

A new tab with the computer's name will appear in the main window when the remote machine is connected:



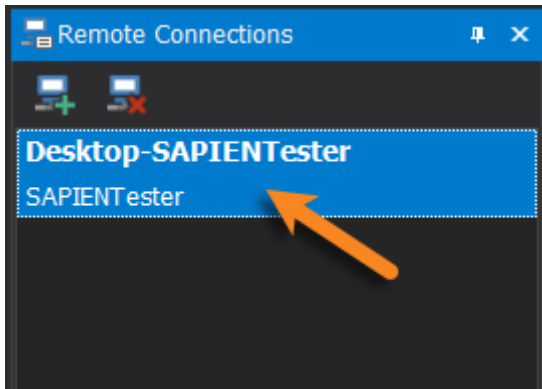
i If a connection is not established, an error will display in the Log panel:



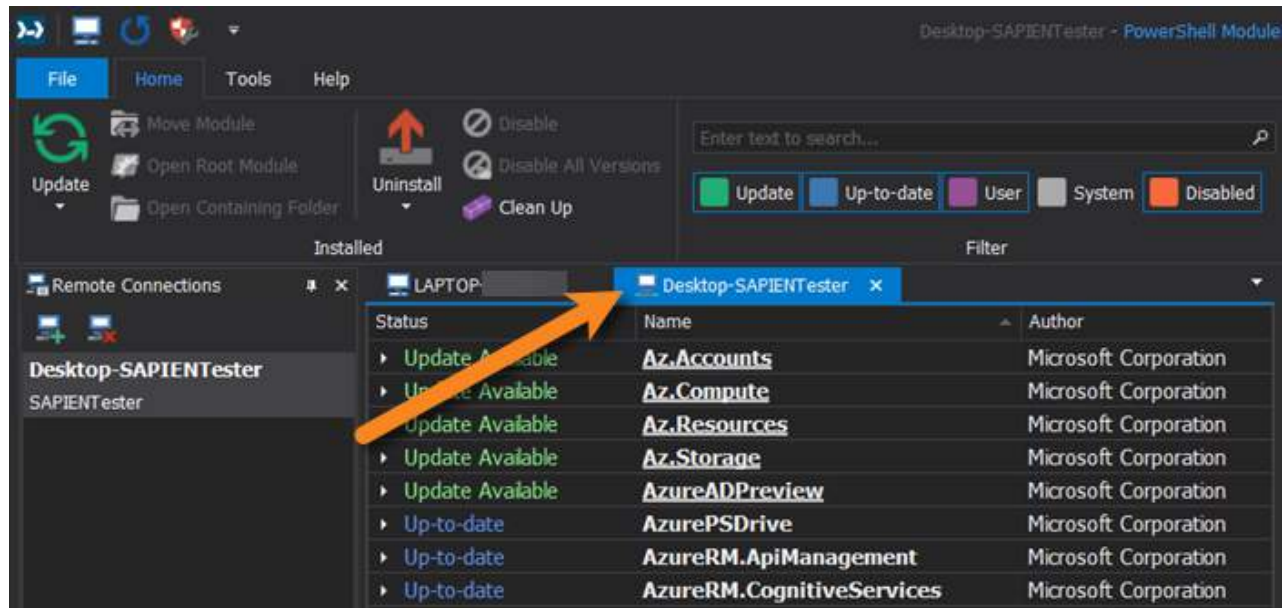
How to connect to a remote machine

To connect to an existing remote machine

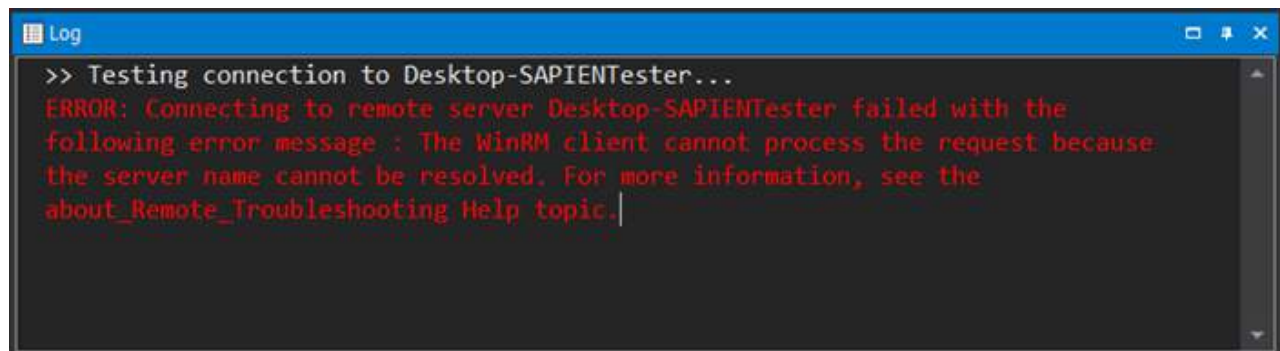
- In the Remote Connections panel, double-click on an existing connection:



A new tab with the computer's name will appear in the main window when the remote machine is connected:



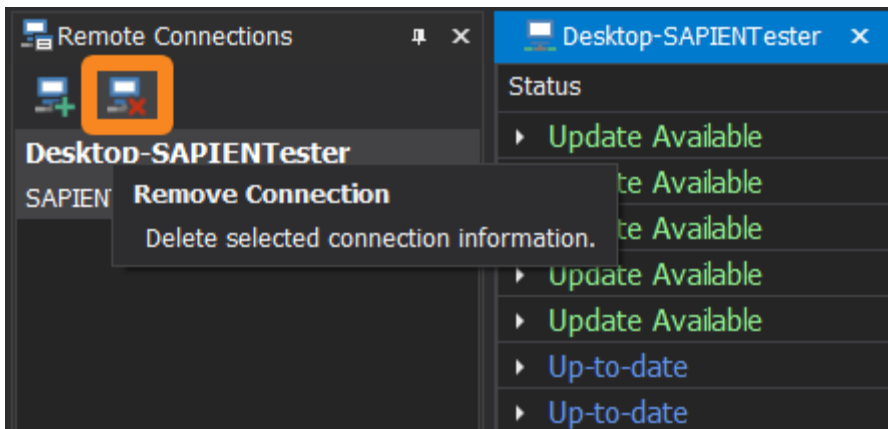
i If a connection is not established, an error will display in the Log panel:



How to remove a remote machine

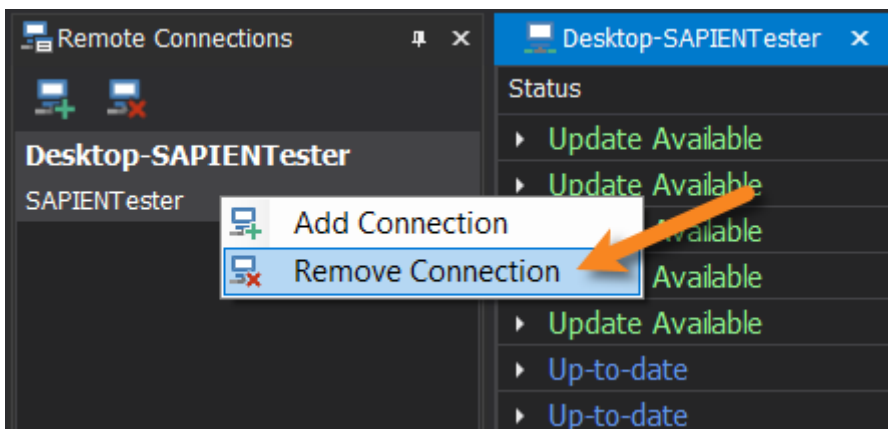
To remove a remote connection

- In the Remote Connections panel, select the remote machine and then click the **Remove Connection** button:



-OR-

- In the Remote Connections panel, right-click on the remote machine, then select **Remove Connection**:



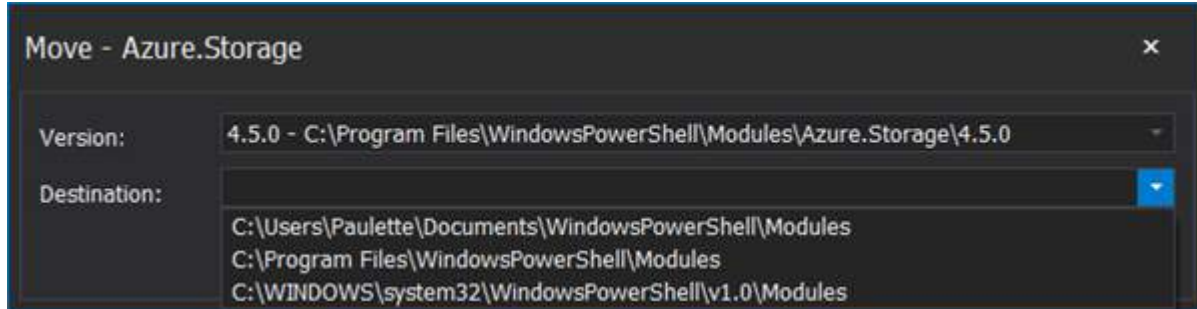
The remote machine is removed from the Remote Connections panel.

5.2.5 Dialogs

This topic provides an overview of the PowerShell ModuleManager dialogs and their options.

Move Dialog

The Move dialog (**Home** > **Move Module**) allows you to move a module's location.



Move Dialog

Move <module> - Field Definitions:

Version	A drop-down list of installed module versions.
Destination	A drop-down list of available destination folders.

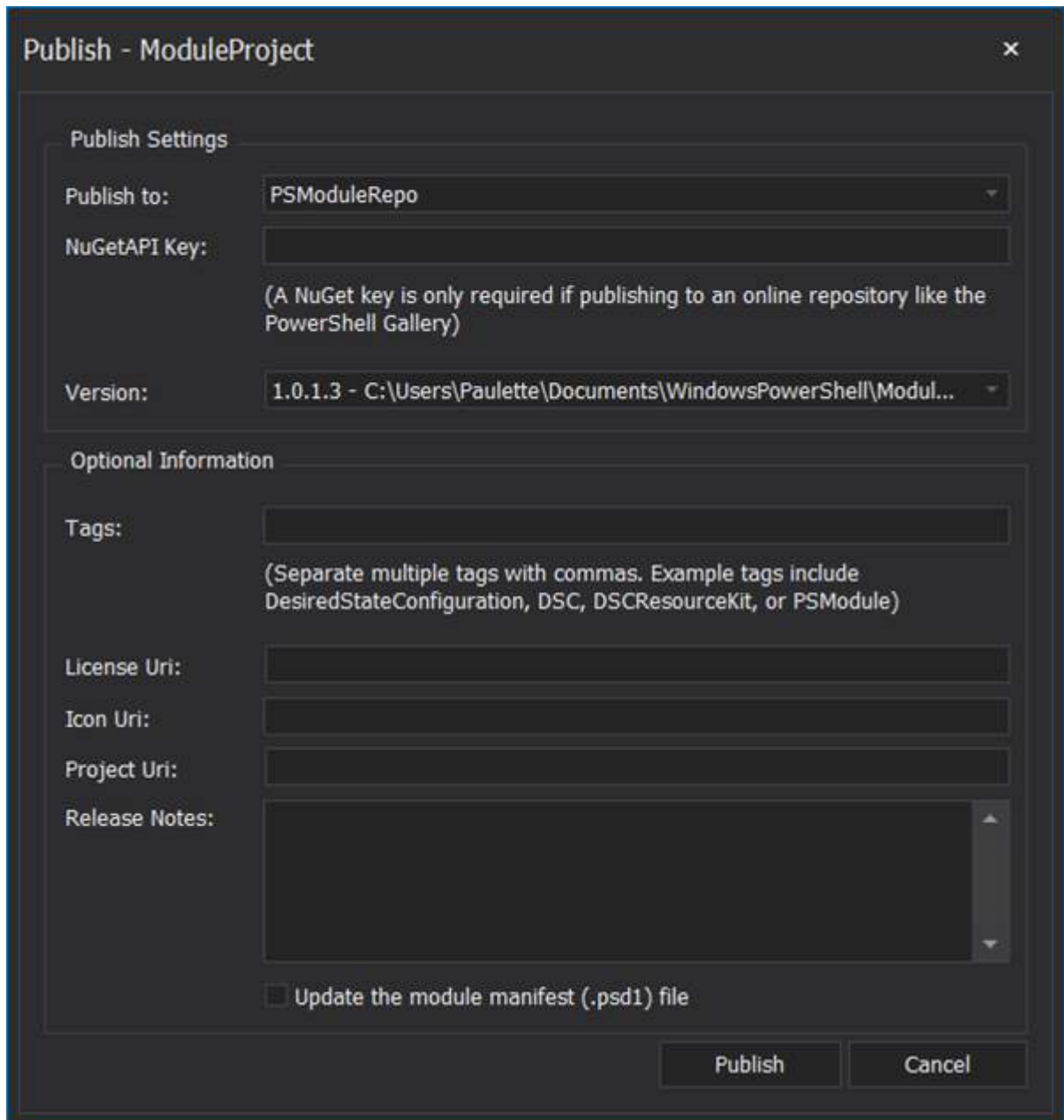
👍 See [Installation Scope](#)^[44] for information about where modules are installed when installing or updating.

ℹ For information about moving modules, see Managing Modules > Managing Installed Modules > [Moving Modules](#)^[59].

Publish Dialog

The Publish dialog allows you to publish modules to registered repositories on your local machine, or online repositories like the PowerShell Gallery.

ℹ We recommend that you review the [version selection behavior of the Publish-Module command](#)^[60] and the [publishing prerequisites](#)^[60] before publishing a module.



The image shows a 'Publish - ModuleProject' dialog box with a dark theme. It is divided into two main sections: 'Publish Settings' and 'Optional Information'. The 'Publish Settings' section includes a 'Publish to:' dropdown menu set to 'PSModuleRepo', a 'NuGetAPI Key:' text box, and a 'Version:' dropdown menu set to '1.0.1.3 - C:\Users\Paulette\Documents\WindowsPowerShell\Modul...'. Below these is a note: '(A NuGet key is only required if publishing to an online repository like the PowerShell Gallery)'. The 'Optional Information' section includes a 'Tags:' text box, a 'License Uri:' text box, an 'Icon Uri:' text box, a 'Project Uri:' text box, and a 'Release Notes:' text area. Below the 'Release Notes' is a checkbox labeled 'Update the module manifest (.psd1) file'. At the bottom right are 'Publish' and 'Cancel' buttons.

Publish - ModuleProject

Publish Settings

Publish to: PSModuleRepo

NuGetAPI Key:

(A NuGet key is only required if publishing to an online repository like the PowerShell Gallery)

Version: 1.0.1.3 - C:\Users\Paulette\Documents\WindowsPowerShell\Modul...

Optional Information

Tags:

(Separate multiple tags with commas. Example tags include DesiredStateConfiguration, DSC, DSCResourceKit, or PSModule)

License Uri:

Icon Uri:

Project Uri:

Release Notes:

☐ Update the module manifest (.psd1) file



Publish Cancel

Publish Dialog

Publish <module> - Field Definitions:


Publish to*:	Select the registered repository from the drop-down list.
NuGetAPI Key:	If you are publishing to an online repository like the PowerShell Gallery, enter the NuGetAPI key.
Version*:	Select the module version.
Tags:	Keywords used to search for modules in PowerShell repositories. Separate multiple tags with commas.
License Uri:	The URL of licensing terms for the module.
Icon Uri:	The URL of an icon for the module.
Project Uri:	The URL of a website for the project.
Release Notes:	Comments about this version of the module.
Update the module manifest (.psd1) file:	<ul style="list-style-type: none"> • If the data in the Optional Information section already exists in the module manifest, you do not need to enter it in the Publish dialog. • If you specify data in any of the Optional Information fields that is not already in the module manifest, you must select the <i>Update the module manifest (.psd1) file</i> checkbox to update the local module's manifest.

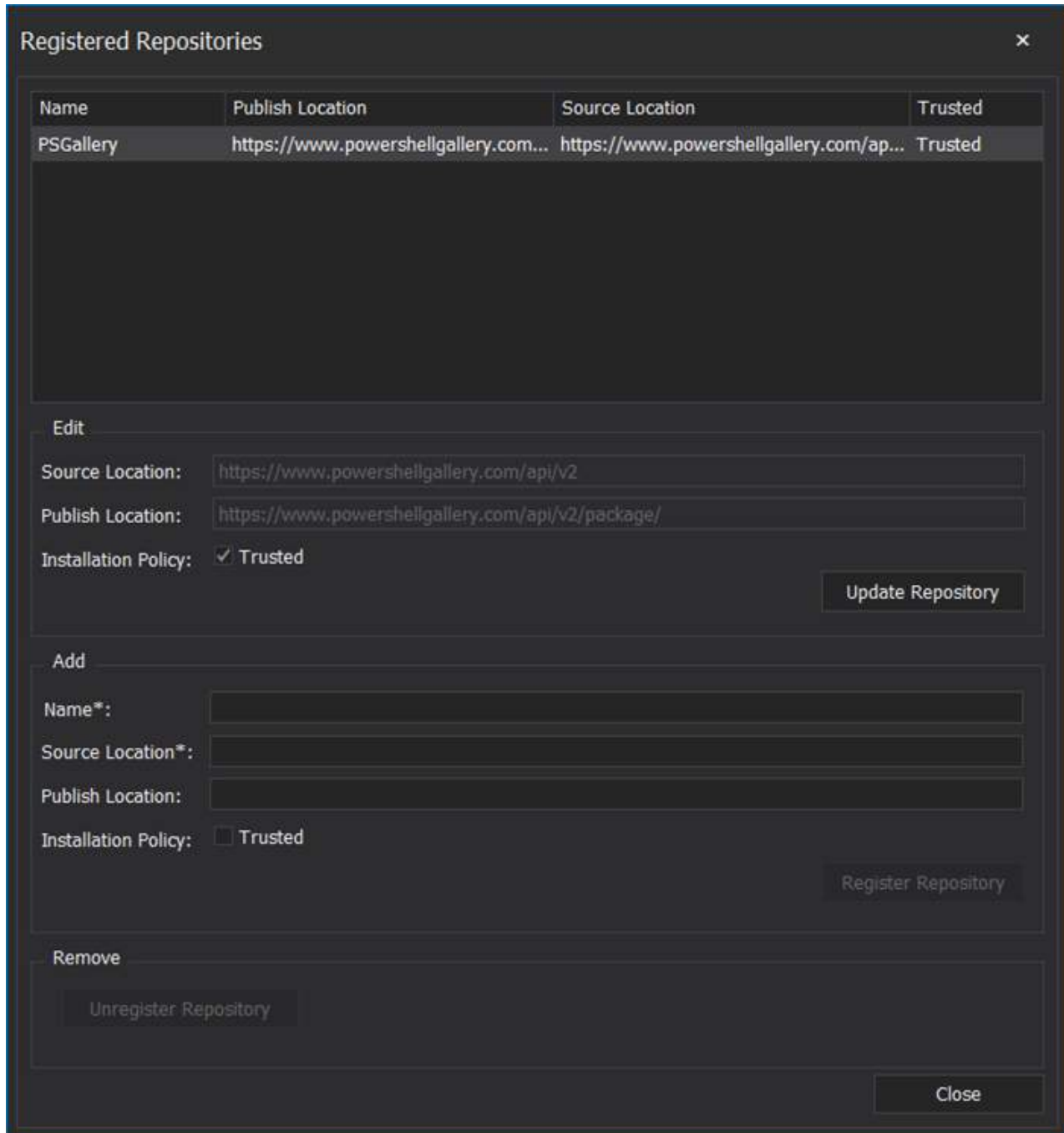
* Required fields

 For information about publishing modules, see Managing Modules > Managing Installed Modules > [Publishing Modules](#) .

Registered Repositories Dialog

Registered Repositories (**Home** > **Registered Repositories**) allows you to view all of the repositories registered to your machine. You can also register, edit, and unregister repositories in the dialog.

 You can use the Registered Repositories dialog for repositories that are on network drives, but not for repositories that are on mapped drives.

*Registered Repositories Dialog*

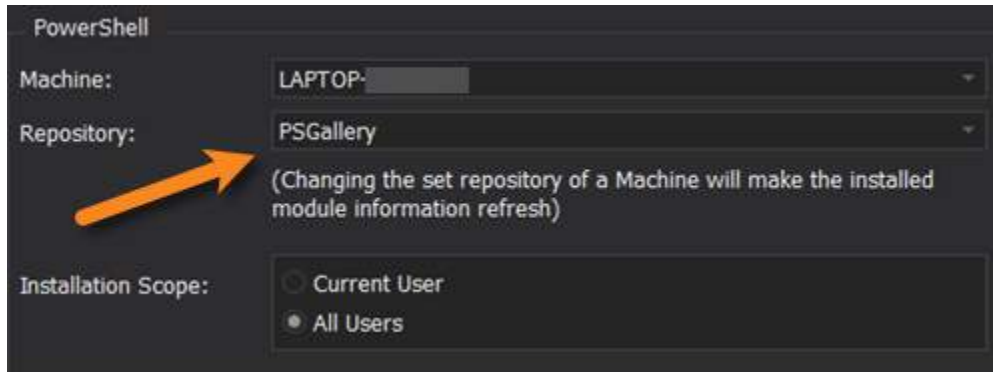
Registered Repositories - Field Definitions:

Edit	<p>These fields allow you to edit the selected repository.</p> <p>Source Location The URL or file path of the repository to search and install modules from.</p> <p>Publish Location The URL or file path of the repository where modules should be published.</p> <p>Installation Policy Specifies the PowerShell behavior when installing from the repository:</p> <ul style="list-style-type: none">• Untrusted: Prompts for confirmation before installing.• Trusted: Does not prompt for confirmation before installing.
Add	<p>These fields allow you to register a repository.</p> <p>Name* The repository name.</p> <p>Source Location* The URL or file path of the repository to search and install modules from.</p> <p>Publish Location The URL or file path of the repository where modules should be published.</p> <p>Installation Policy Specifies the PowerShell behavior when installing from the repository:</p> <ul style="list-style-type: none">• Untrusted: Prompts for confirmation before installing.• Trusted: Does not prompt for confirmation before installing.

* Required fields

Set Repository:

The default registered repository is the PowerShell Gallery (PSGallery). This can be changed in **File > Options > General > PowerShell:**

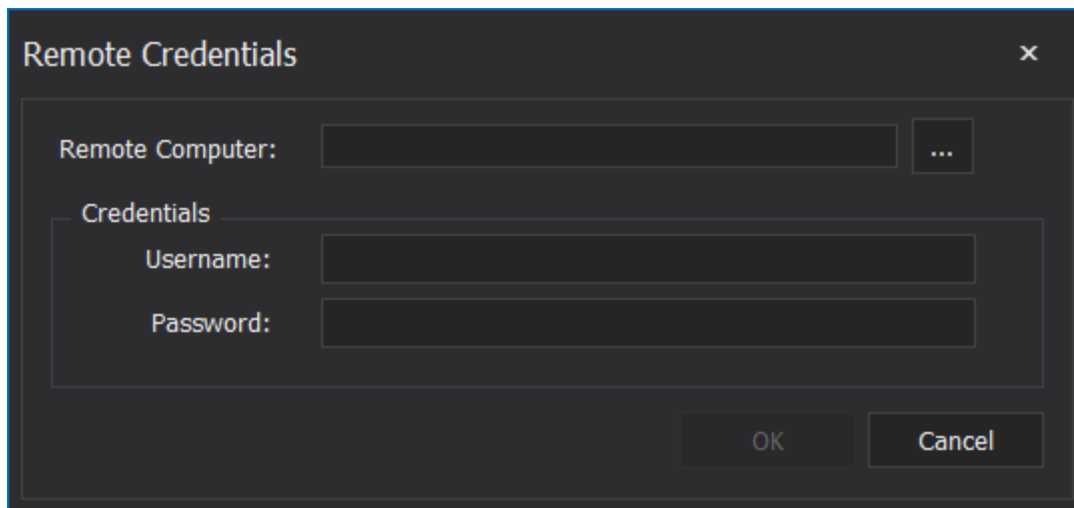


Options > General > Powershell

i For information about registered repositories, see Managing Modules > [Managing Repositories](#)

Remote Credentials Dialog


Remote Credentials (Remote Connections panel > **Add Connection**) allows you to connect to a remote machine.



Remote Credentials Dialog

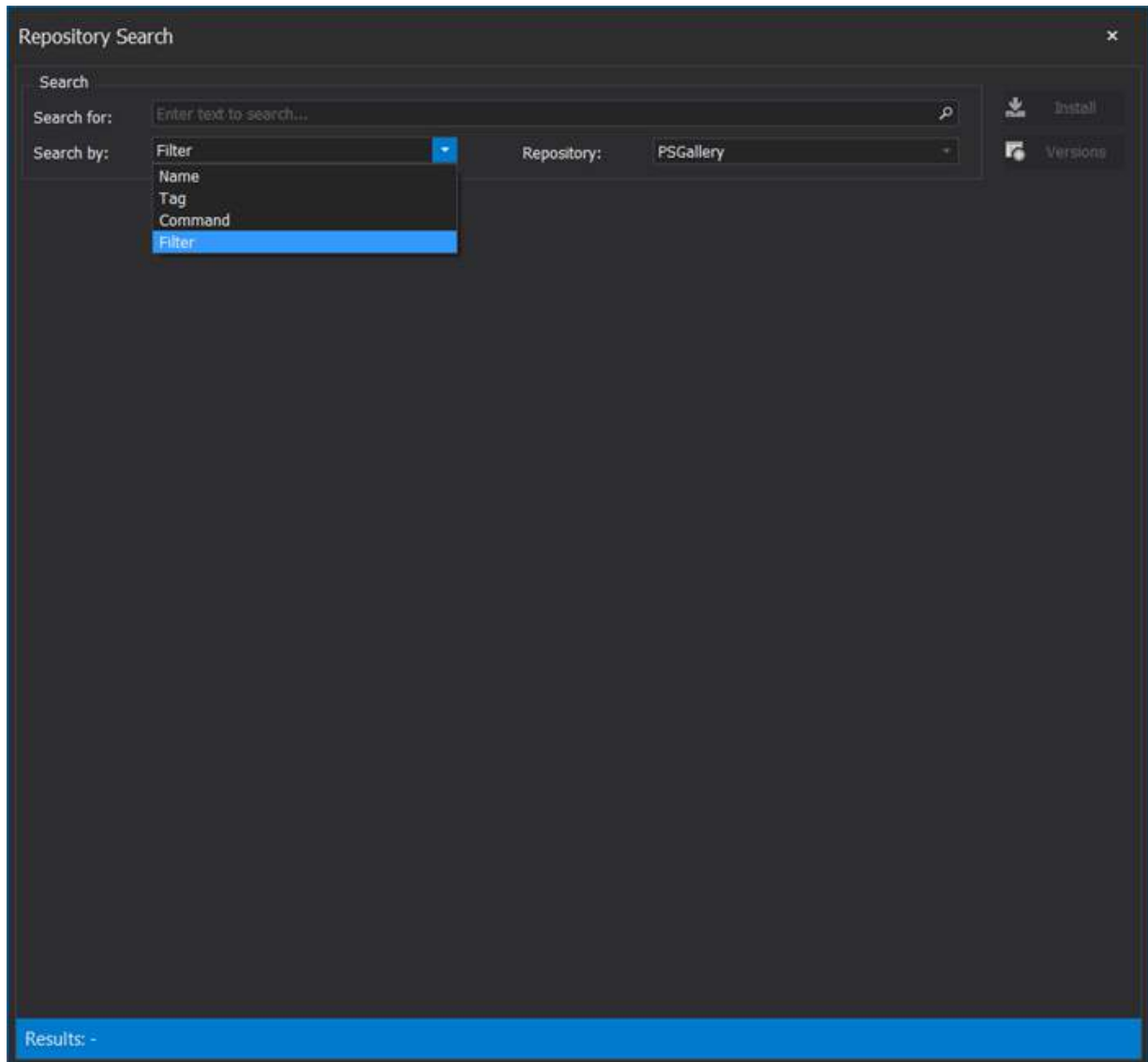
Remote Credentials - Field Definitions:

Remote Computer	The computer name, or search connected computers (...).
Credentials	<p>Username The username for the remote computer user account.</p> <p>Password The password for the remote computer user account.</p>

 For information about connecting to remote machines, see Managing Modules > [Managing Remote Connections](#)^[92].

Repository Search Dialog

Repository Search (*Ctrl+F*) displays module information from registered repositories and allows you to add modules to your system.



Repository Search Dialog


Repository Search - Field Definitions:

Search for Enter the search term(s).

Search by Select an option from the drop-down list.

Name

Specifies the names of one or more modules to search for.

 This is the only parameter that supports wildcard characters.

Tag

Keywords used to help find modules.


Command



Cmdlets, functions, or aliases found in the module.

Filter

Filters through the name, description, and tags for the search argument based on the PackageManagement module provider-specific search syntax. For NuGet modules, this is the equivalent of searching by using the Search bar on the PowerShell Gallery website.

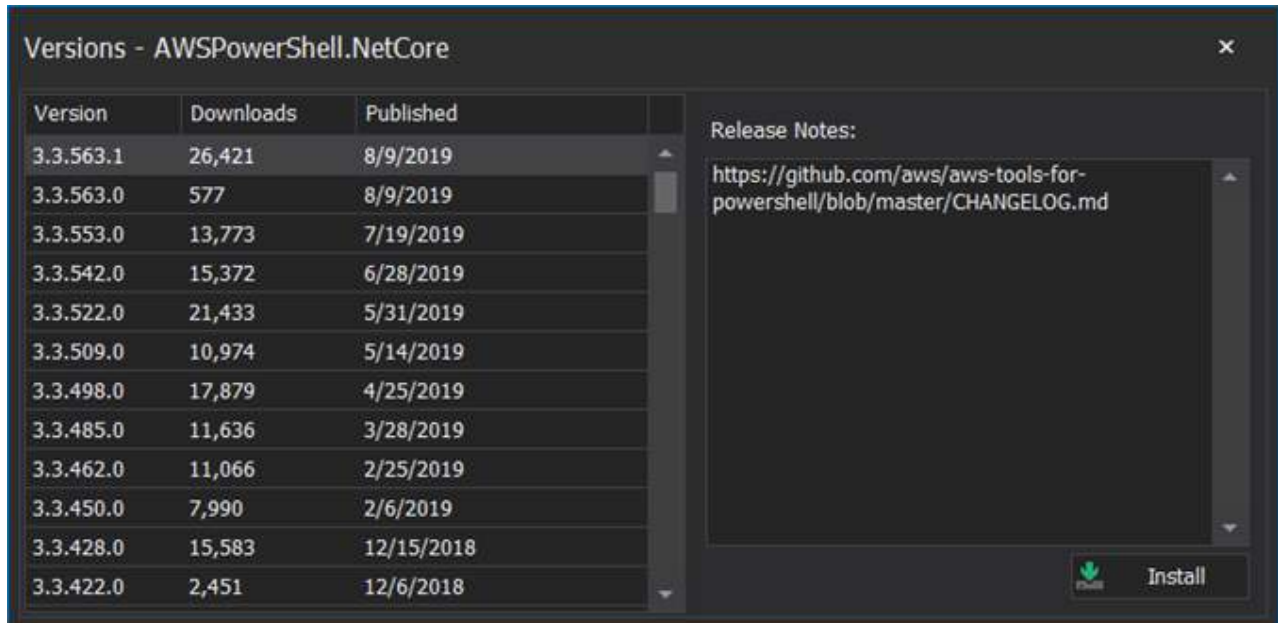
Repository Select a repository from the drop-down list.

 The default registered repository is the PowerShell Gallery (PSGallery). This setting can be changed in **File > Options > General > PowerShell**.

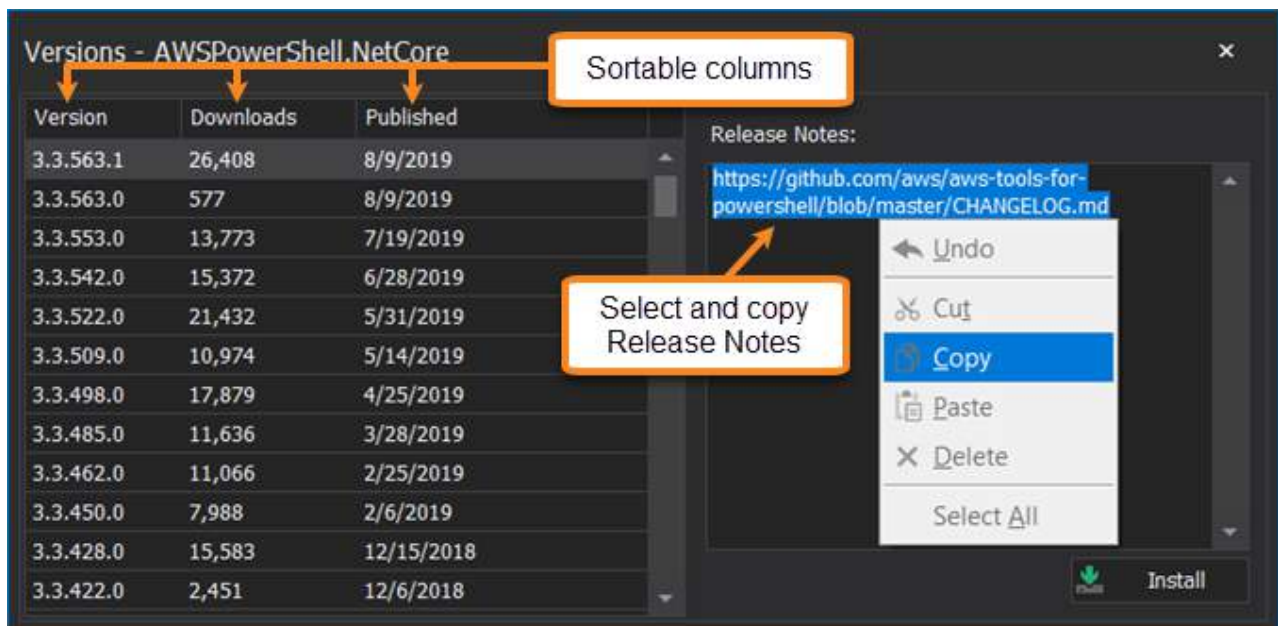
 For information about searching for modules, see Managing Modules > [Searching for Modules](#) 

Versions Dialog

The Versions (*Ctrl+V*) dialog allows you to view and install all available versions of a module.

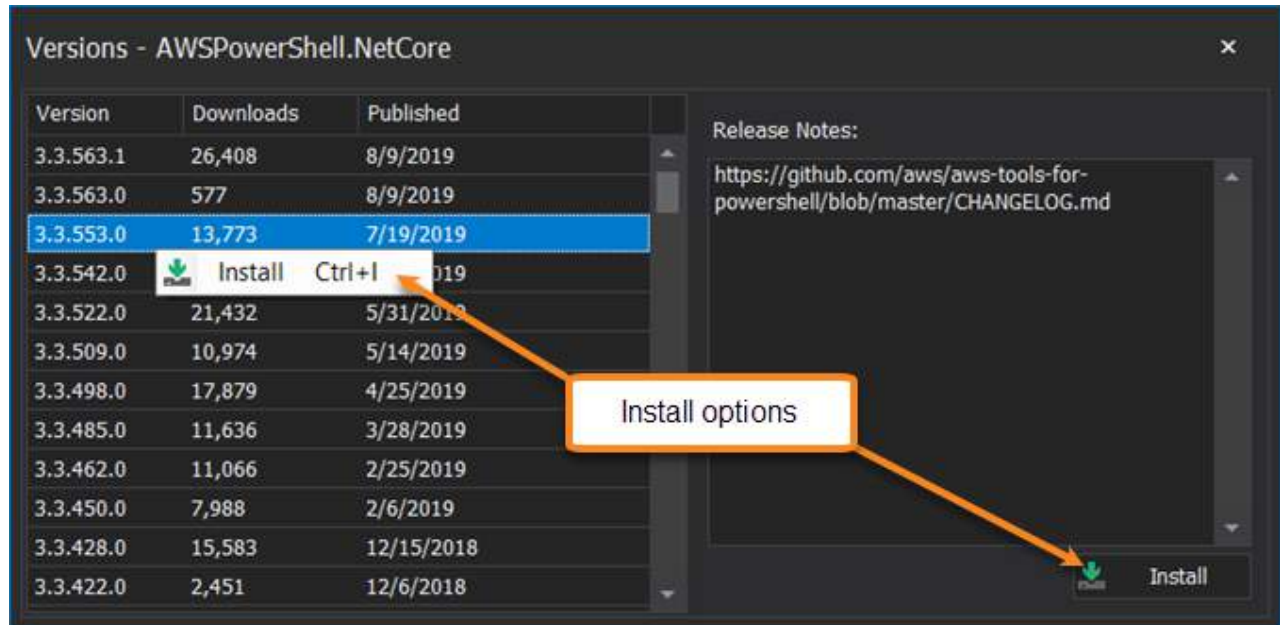
*Versions Dialog*

You can sort the Version, Downloads, and Published columns, and also right-click and copy the Release Notes:



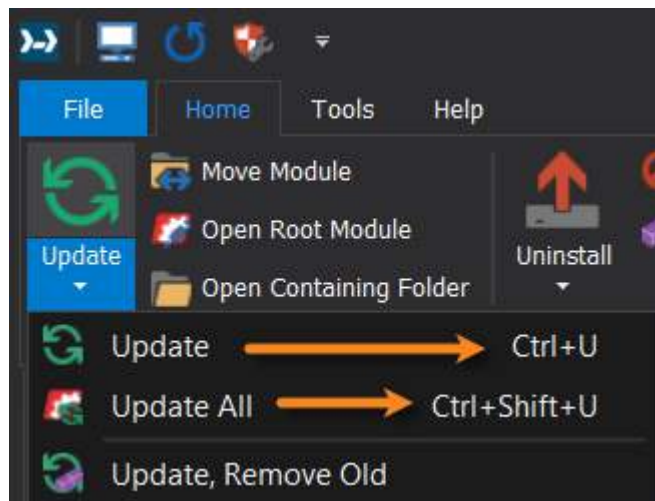
There are two ways to install a selected version:

- Right-click and select **Install** (*Ctrl+I*).
- Click the **Install** button.

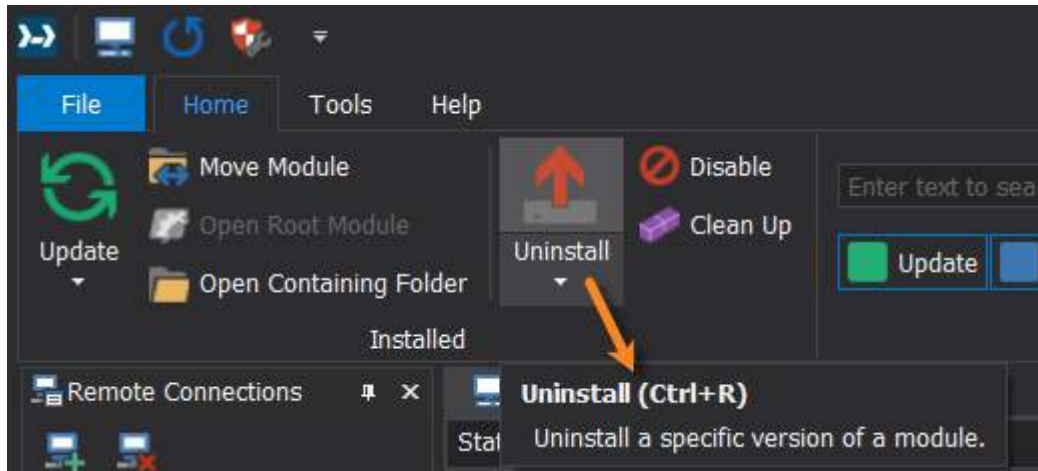


5.2.6 Keyboard Shortcuts

Some PowerShell ModuleManager functions have keyboard shortcuts which appear in the menu options and button tooltips:



Keyboard shortcuts in menu options



Keyboard shortcuts in ribbon button tooltips

Keyboard Shortcuts

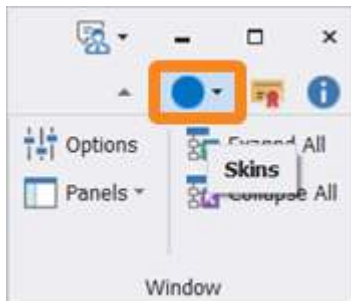
Help	<i>F1</i>
Repository Search	<i>Ctrl + F</i>
Available Versions	<i>Ctrl + V</i>
Install	<i>Ctrl + I</i>
Refresh Installed	<i>F5</i>
Open Containing Folder	<i>Ctrl + O</i>
Update	<i>Ctrl + U</i>
Update All	<i>Ctrl + Shift + U</i>
Disable	<i>Ctrl + D</i>
Uninstall	<i>Ctrl + R</i>
Minimize Ribbon / Expand Ribbon	<i>Ctrl + F1</i>
Access Ribbon Key Shortcuts	<i>Alt</i>

5.3 Customizing Your Workspace

PowerShell ModuleManager provides options for customizing the program workspace. You can change the color theme, add frequently used functions to the Quick Access Toolbar, and reconfigure the panel layout.

5.3.1 Changing the color theme

Color themes let you modify the colors in PowerShell ModuleManager's user interface to suit your preference and work environment.



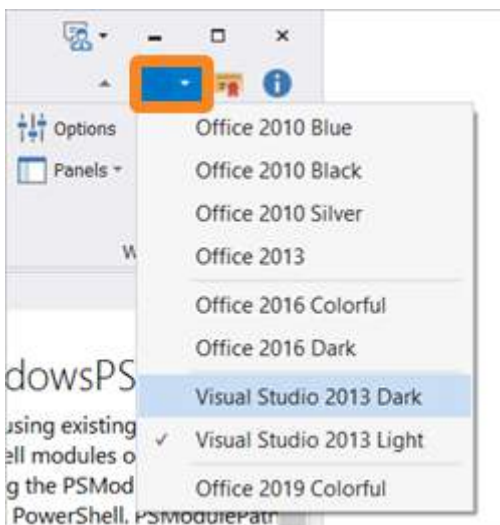
Skins button

How to change the color-themed skin

You can change PowerShell ModuleManager's user interface color theme from the **Skins** menu above the Ribbon.

To change the color-themed skin

- Click the **Skins** button and select a theme from the menu:

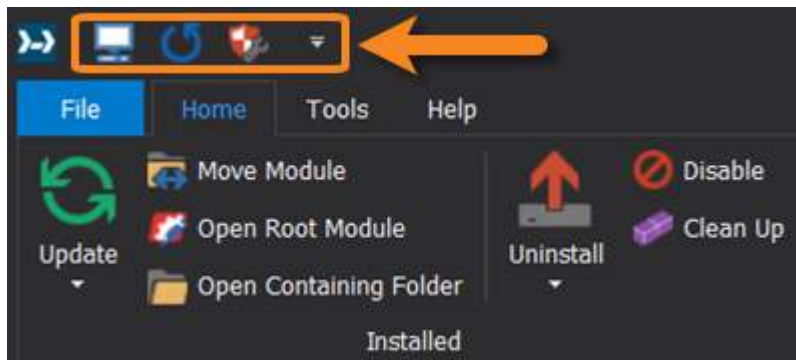


5.3.2 Customizing the Quick Access Toolbar

The **Quick Access Toolbar** at the top-left of the program window provides access to your most frequently used tools. This topic explains how to add or remove default buttons, how to add new buttons, and how to show the Quick Access Toolbar above or below the ribbon.

Default Buttons

The default controls on the Quick Access Toolbar provide instant access to view your local machine, refresh all installed modules (F5), or change the PowerShell execution policy:



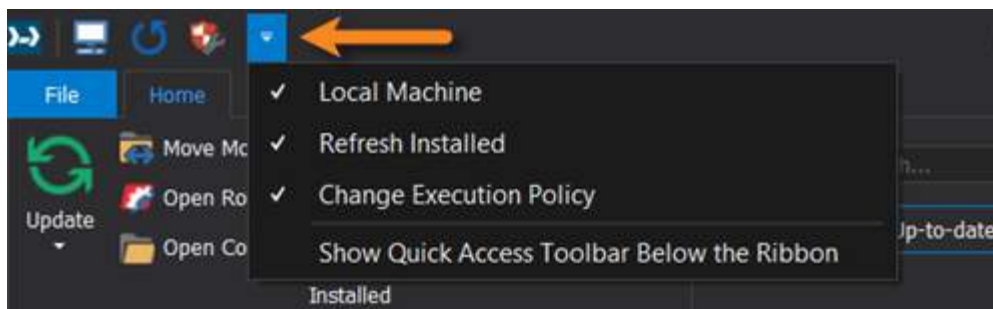
Quick Access Toolbar

-  The default options on the Quick Access Toolbar are also available on the [File Menu](#) ¹⁰⁷.

How to add or remove default buttons to the Quick Access Toolbar

To add or remove default controls

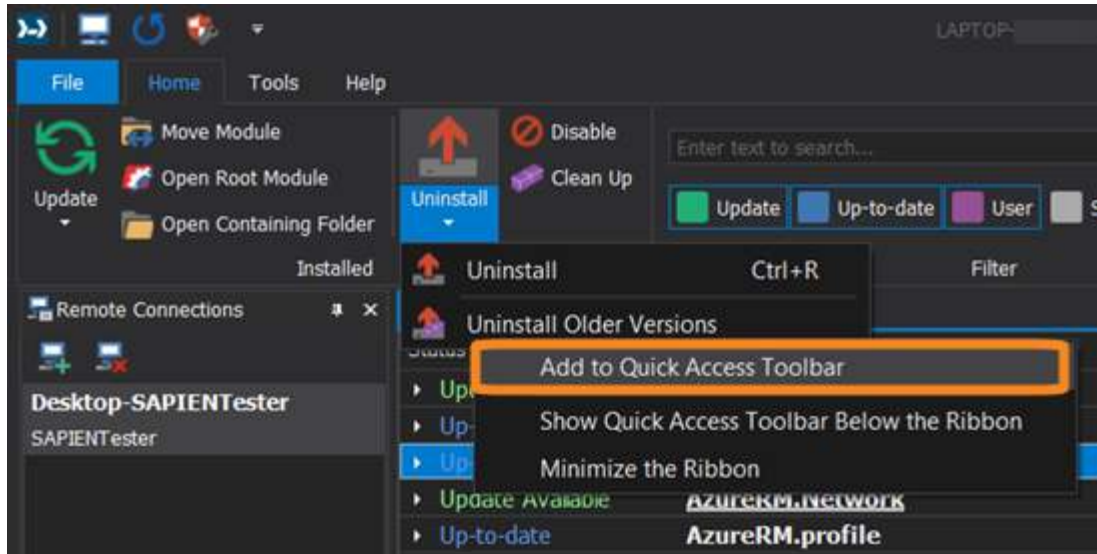
- Click the drop-down icon to the right of the Quick Access Toolbar and select or deselect options:



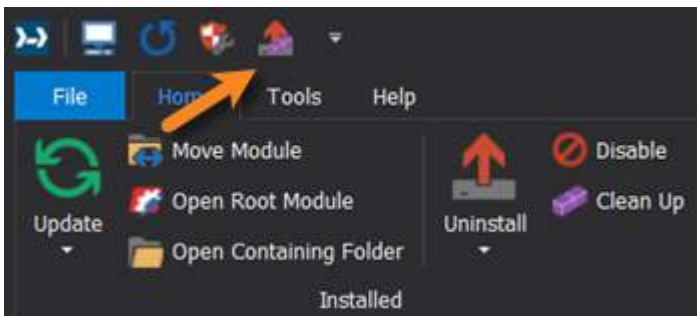
How to add new buttons to the Quick Access Toolbar

To add a button to the toolbar

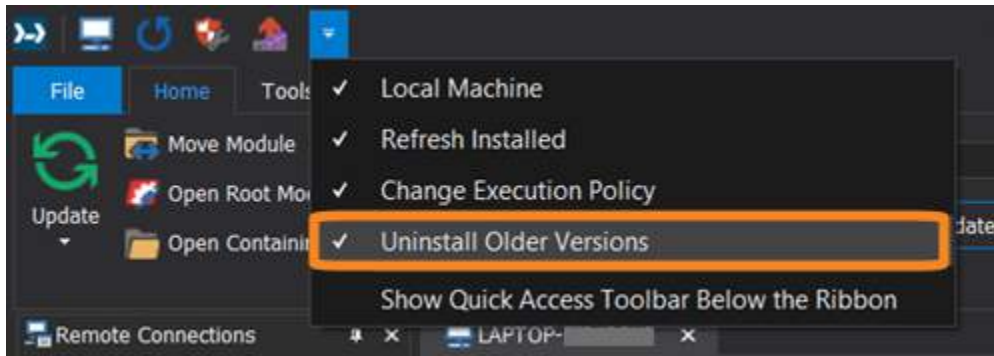
- Right-click on a ribbon control > select **Add to Quick Access Toolbar**:



The selected control is added to the Quick Access Toolbar:



👍 To remove the new button, deselect the control from the drop-down icon to the right of the toolbar:

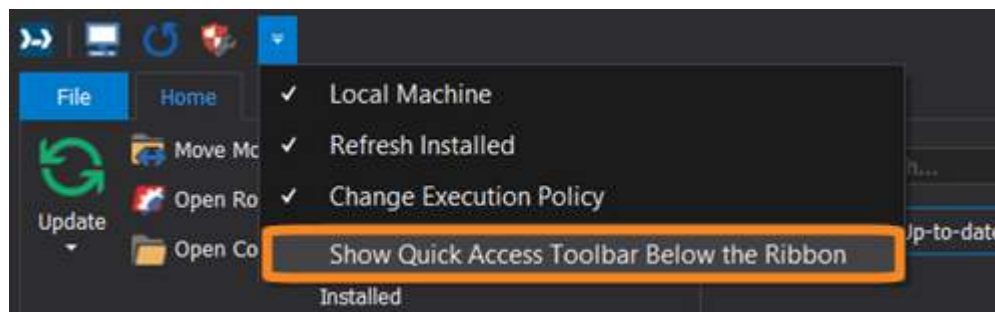


i You can reset the Quick Access Toolbar to the default options from **File > Options > Panels > Reset Quick Access Toolbar**.

How to show the Quick Access Toolbar below the ribbon

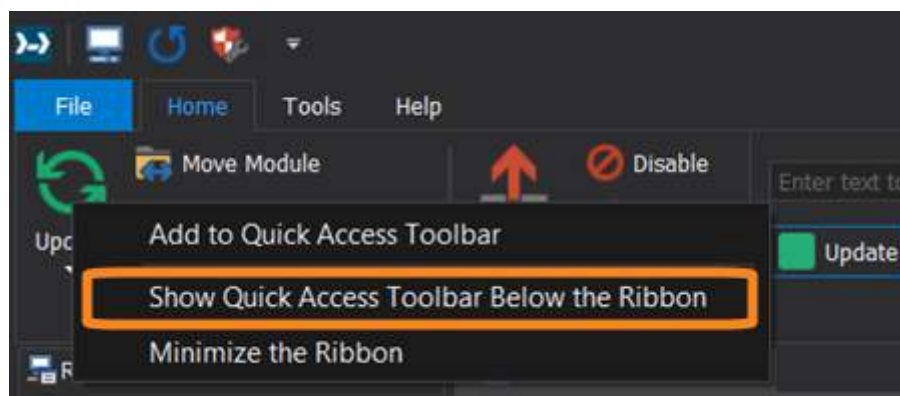
To show the toolbar below the ribbon

- Click the drop-down icon to the right of the Quick Access Toolbar > then select **Show Quick Access Toolbar Below the Ribbon**:



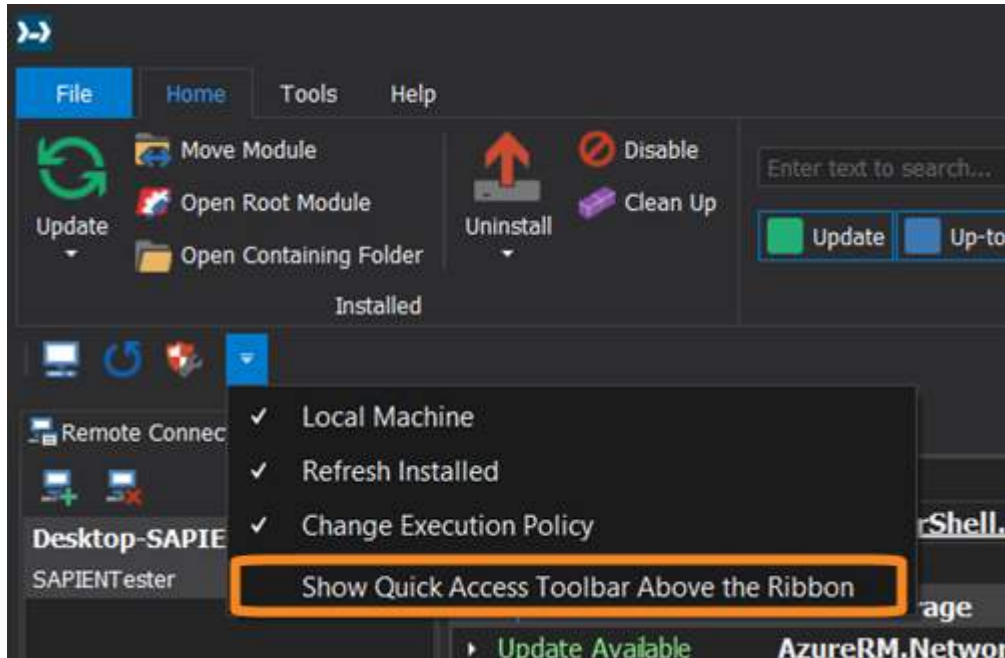
-OR-

Right-click any ribbon control > select **Show Quick Access Toolbar Below the Ribbon**:



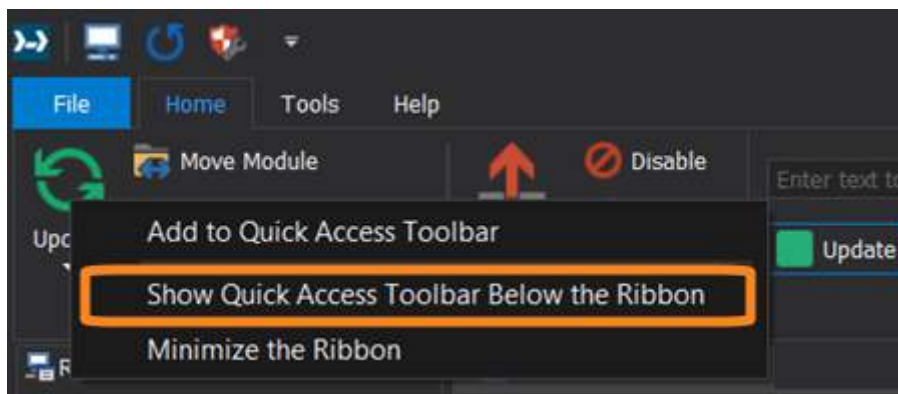
To show the toolbar above the ribbon

- Click the drop-down icon to the right of the Quick Access Toolbar below the ribbon > then select **Show Quick Access Toolbar Above the Ribbon**:



-OR-

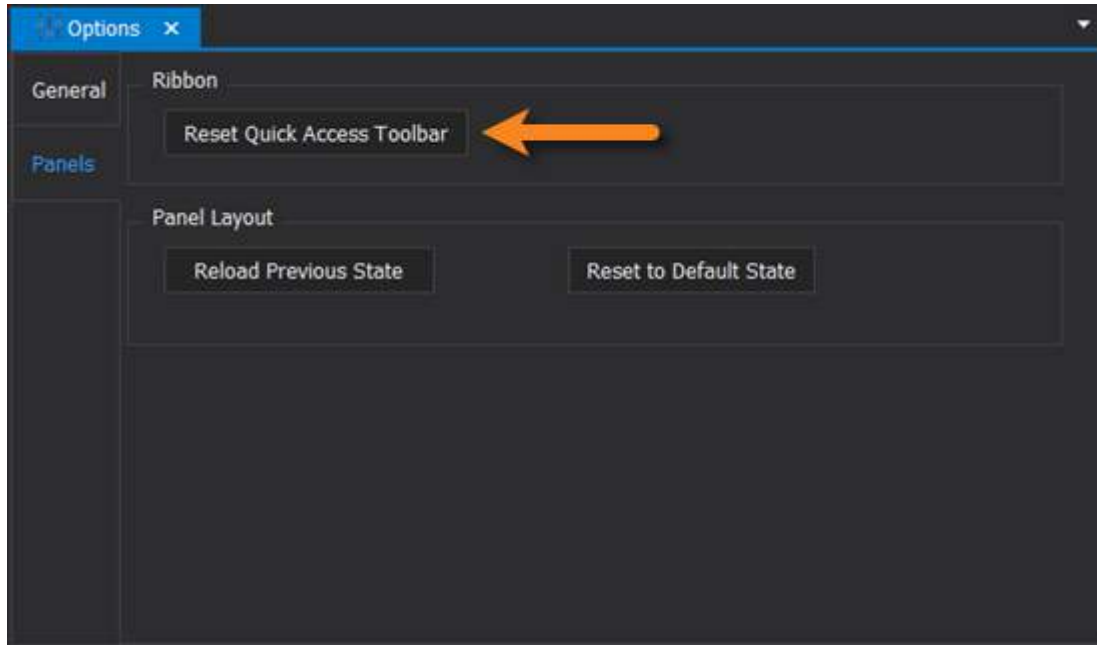
Right-click any ribbon control > select **Show Quick Access Toolbar Above the Ribbon**:



How to restore the Quick Access Toolbar default options

To reset the Quick Access Toolbar

- Select File > Options > Panels > Reset Quick Access Toolbar:

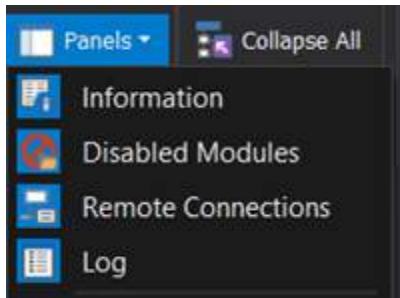


5.3.3 Panels and Layout

This section explains how to work with [panels](#)^[22], and provides an overview of the most common tool panel tasks.

PowerShell ModuleManager - Available panels

The panels available in PowerShell ModuleManager are, by default, all displayed in the program [workspace](#)^[12]. The **Panels** drop-down list (**Home** ribbon > **Window** section) shows the available panels:



i Panels that are *open* in PowerShell ModuleManager have a blue highlight in the Panels drop-down list, which behaves like a toggle; click a panel on the list to open or close the panel (see below for more ways to close a panel).

How to close and open a panel

By default, all available panels are open when the application starts. There are a number of ways to close and open panels.

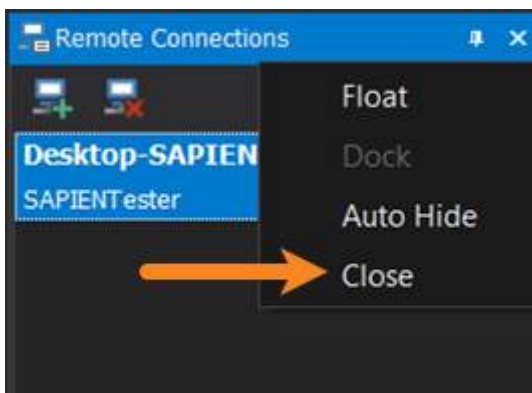
To close a panel

- Click the X on the title bar:



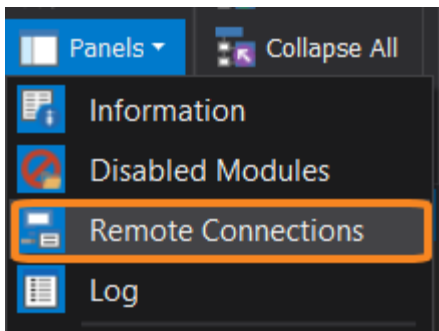
-OR-

- Right-click on the title bar and select **Close**:

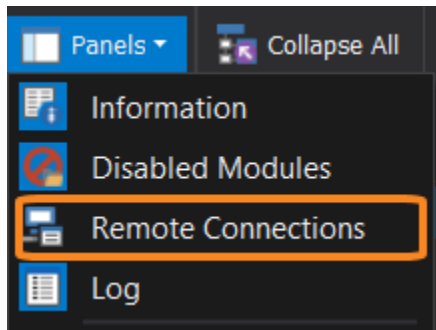


-OR-

- From the Panels drop-down list, **select the panel name** to close:

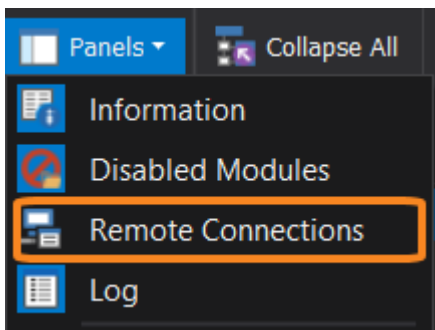


The panel will close, and the panel name on the drop-down list will not be highlighted:

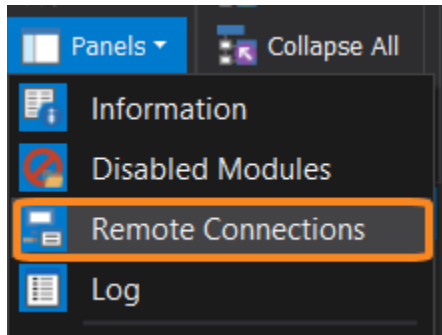


To open a panel

- From the Panels drop-down list, **select the panel name** to open:



The panel will open, and the panel name on the drop-down list will be highlighted:

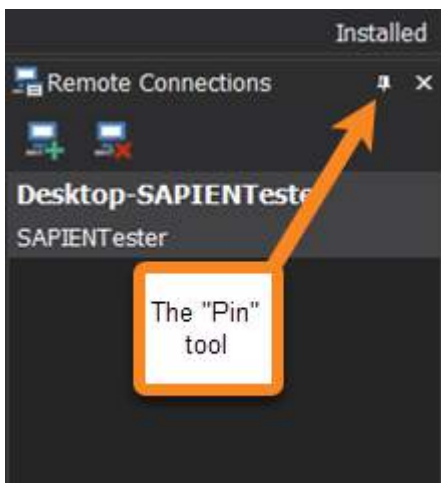


How to pin a panel to the margin

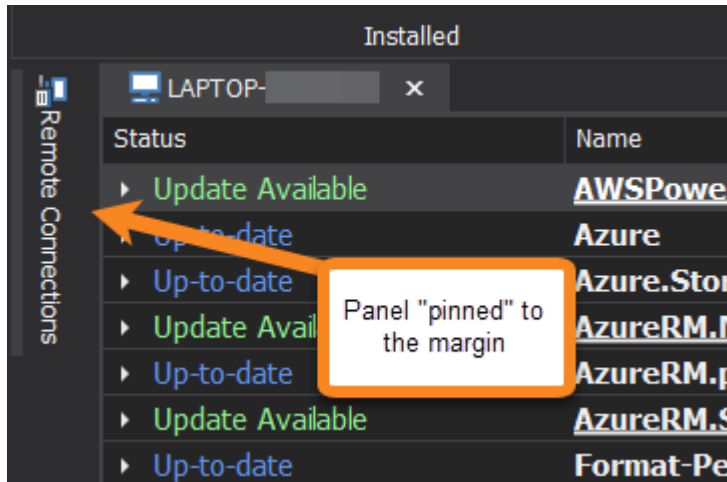
You can "pin" a panel to the margin, which causes the panel to "hide" when you click away from the panel and "reveal" when you mouseover the panel title.

To pin a panel to the margin

- Click the "pin" tool on the top-right of the panel title bar (or right-click the title bar and select Auto-Hide):

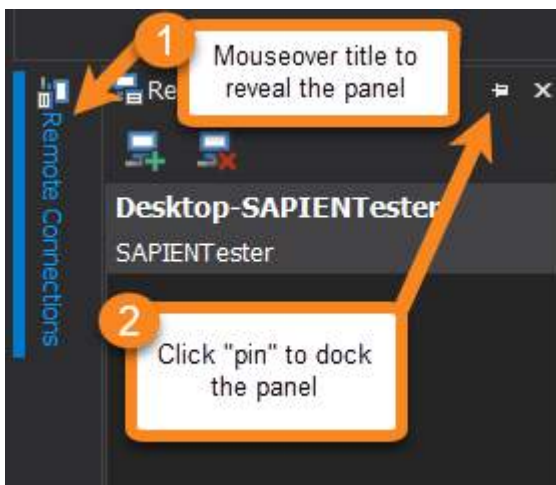


The panel will be hidden and will only display when you mouseover the panel title in the margin:



To unpin a panel from the margin

- Mouseover the title in the margin to reveal the panel > then click on the "pin" tool in the title bar:



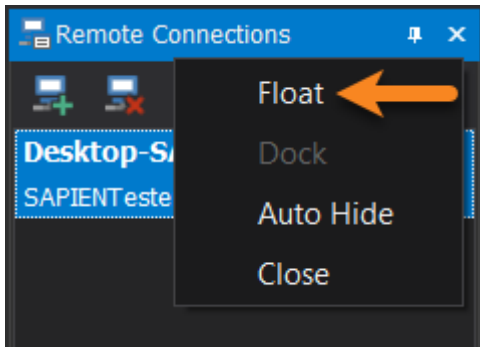
The panel will return to its most recent docked position.

How to float, size, move, and dock a panel

Panels can be released, or "floated" from their docked position. Floating panels can be moved, sized, or docked.

To float a panel

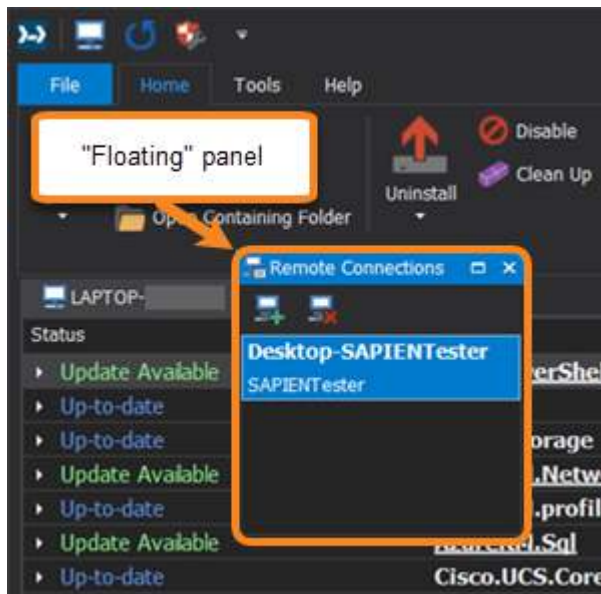
- Right-click on the title bar and select **Float**:



-OR-

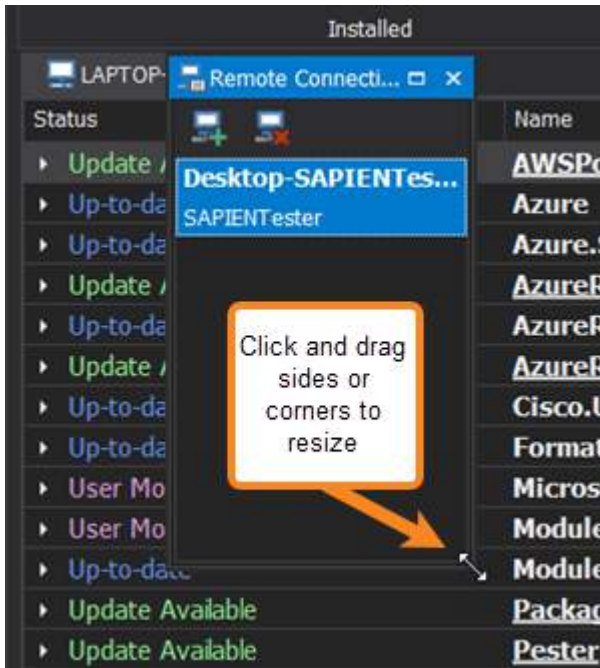
- Drag the panel title bar and drop it in the program window.

The panel will then "float" in the program window:

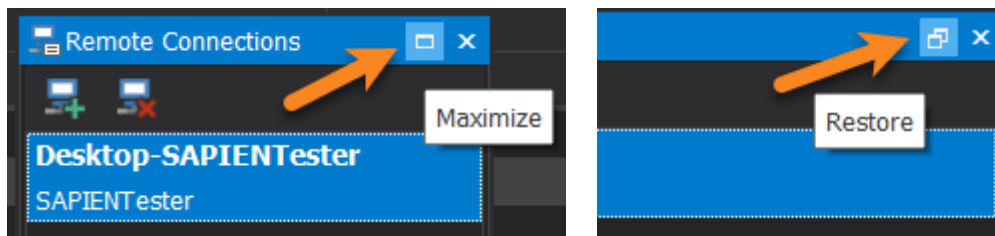


To size a panel

- Click and drag the pointer arrows on the sides and corners of the panel:

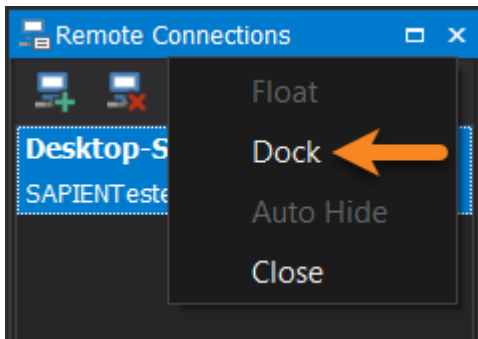


Panels can also be **maximized** to the full size of the display, and **restored** to the most recent size by clicking the respective button on the top-right of the title bar:

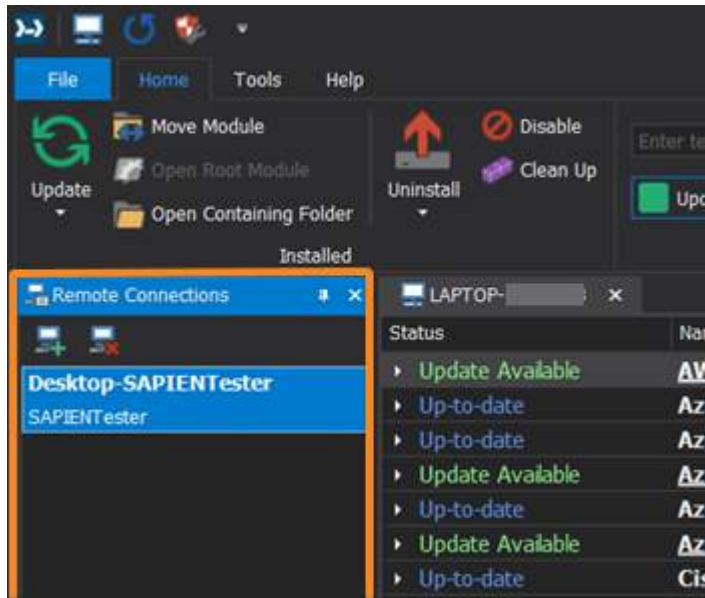


To dock a panel

- Right-click on the title bar and select **Dock**:



The panel is returned to its most recent docked position:

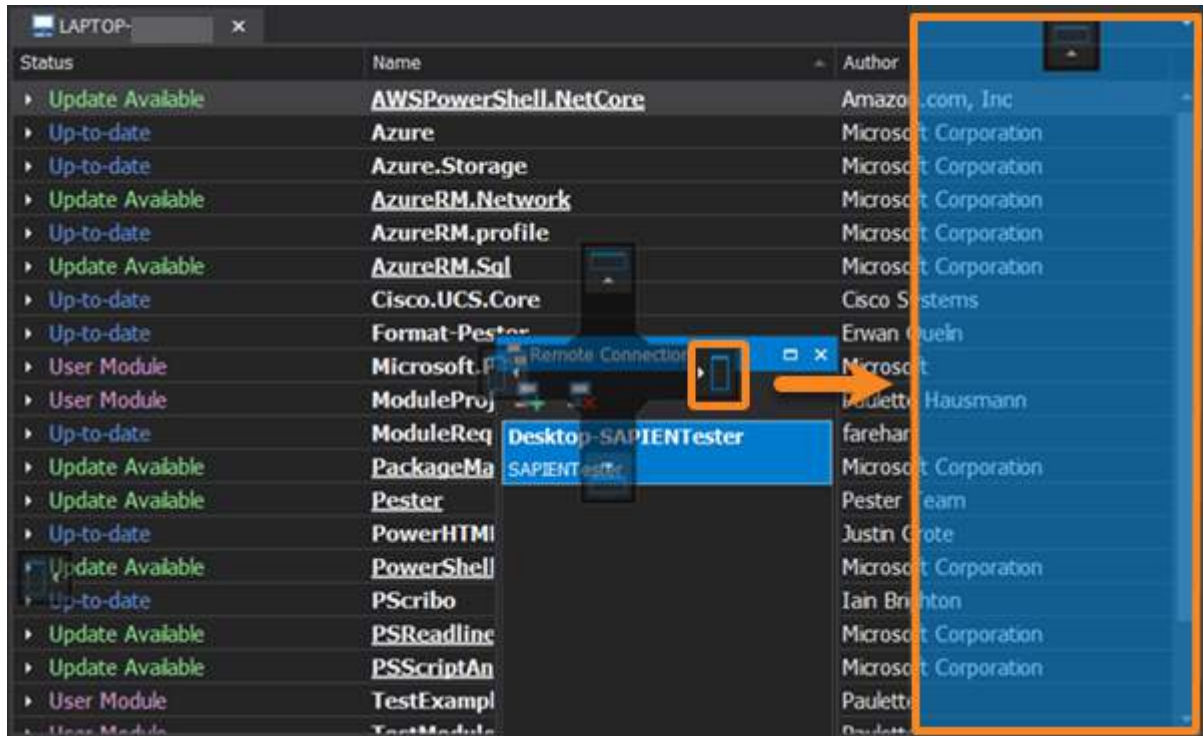


To move a panel

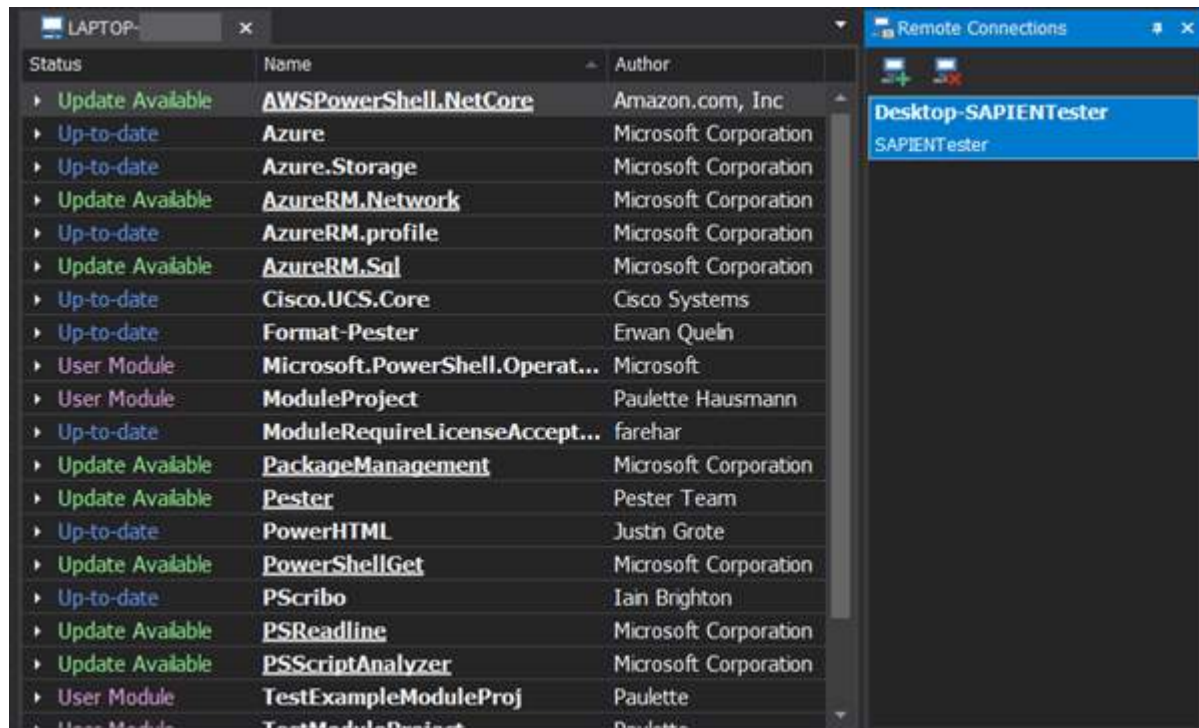
- Click and drag a panel title bar to reveal the docking location indicator:



- Drag the panel and then drop it when the desired location is highlighted in the docking indicator. In the example below, the Remote Connections panel is moved to the right of the Local Machine tab:

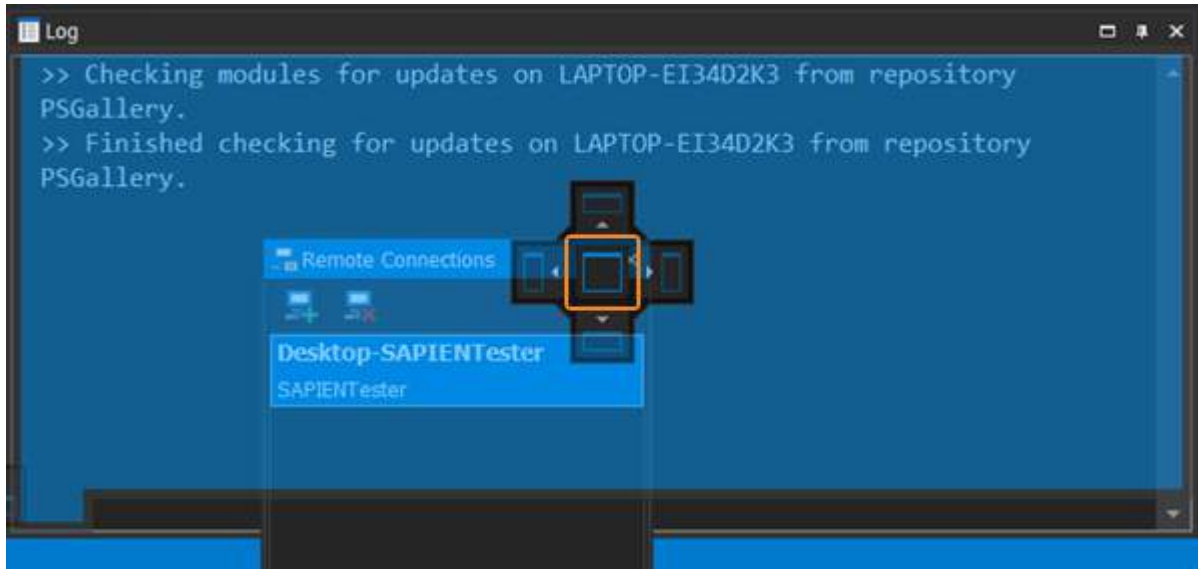


The Remote Connections panel is moved to the right of the Local Machine tab:



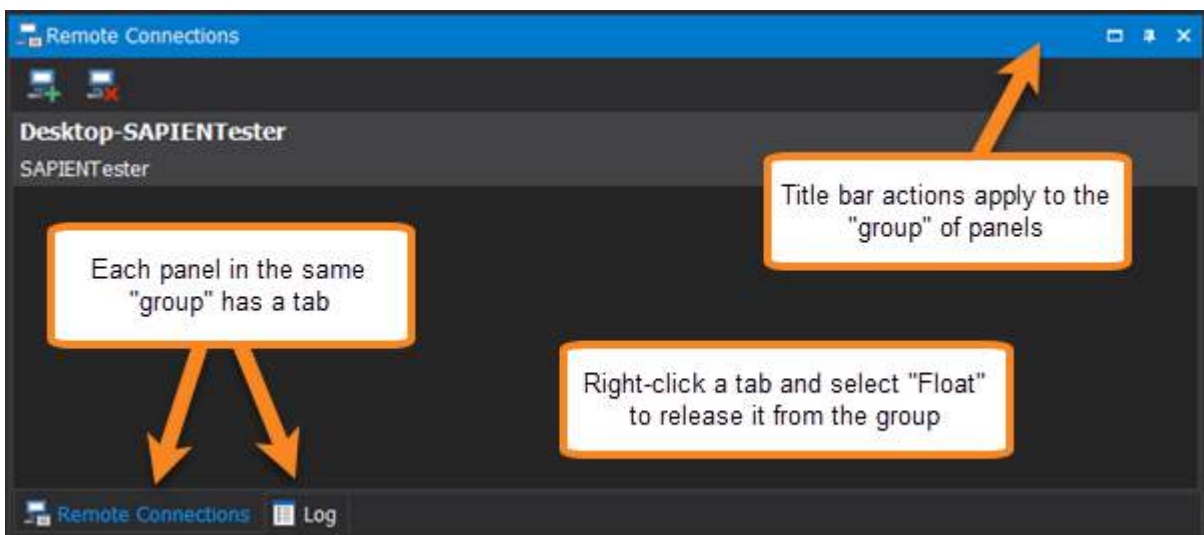
Grouped Panels

If you drop a panel in the center docking indicator, the panel will become part of a "group" with the other panel(s) in that location. In the example below, the Remote Connections panel will be grouped with the Log panel:



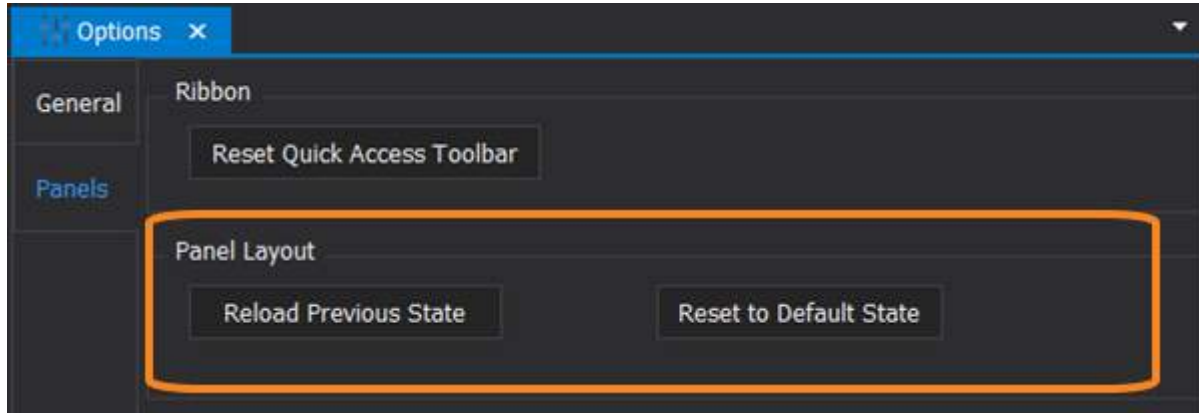
When panels are grouped:

- Each panel in the group will have a tab.
- Dragging the title bar will move the entire group of panels.
- To release a panel from a group, right-click the panel tab and click Float.



How to reload or reset the panel layout

A layout defines the position and visibility of the various PowerShell ModuleManager panels. If you need to revert to a previous layout, there are some options available in the Options dialog (**Home > Options > Panels > Panel Layout**):



Options > Panels > Panel Layout

Panel Layout - Options:

Reload Previous State	Reloads the panels to the same layout pattern they were in the last time you opened PowerShell ModuleManager.
Reset to Default State	Restores the PowerShell ModuleManager panels to the default layout.

👍 To maximize the entire PowerShell ModuleManager program window in your display, double-click the top title bar. Double-click the title bar again to restore the window down to the previous size.

5.4 SAPIEN Updates

We are continually updating our software, both to remove bugs and to add and improve product features. We recommend always staying current with the most recent versions to ensure that you are taking advantage of the latest features, functionality, and product stability.

Every SAPIEN product has a built-in update tool—**SAPIEN Updates**—which will check for updates on all current activations and unexpired trial versions of our products. Available product updates are indicated in the SAPIEN Updates tool and also in the [Notifications dialog](#)^[170] (see below).

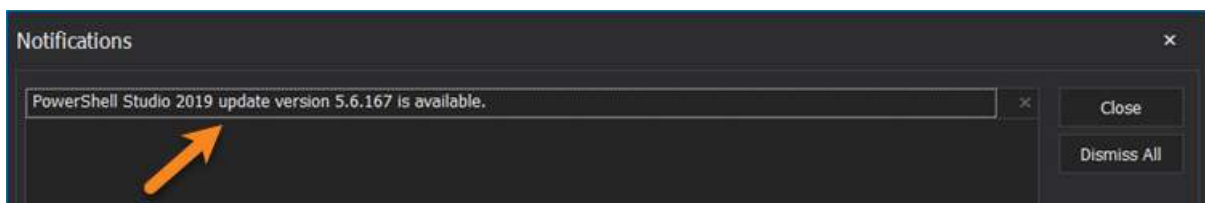
SAPIEN Notifications

SAPIEN products provide automatic notifications when there is a software update available, or when your maintenance is about to expire. Notifications are indicated by a 'flag' icon in the top-right of the program window:



How to view SAPIEN notifications

- Click the notification flag icon above the ribbon to open the Notifications dialog:



- If a product update is available, click the update notification to open the SAPIEN Updates tool.

👍 Click the X button to dismiss individual notifications or select **Dismiss All**. Dismissed notifications will not be shown again.

SAPIEN Updates - Tool Overview

The SAPIEN Updates tool indicates when an update is available for any SAPIEN program installed on your computer.

i To minimize the impact on your system, the tool does not run during Windows startup or continuously in the system tray.

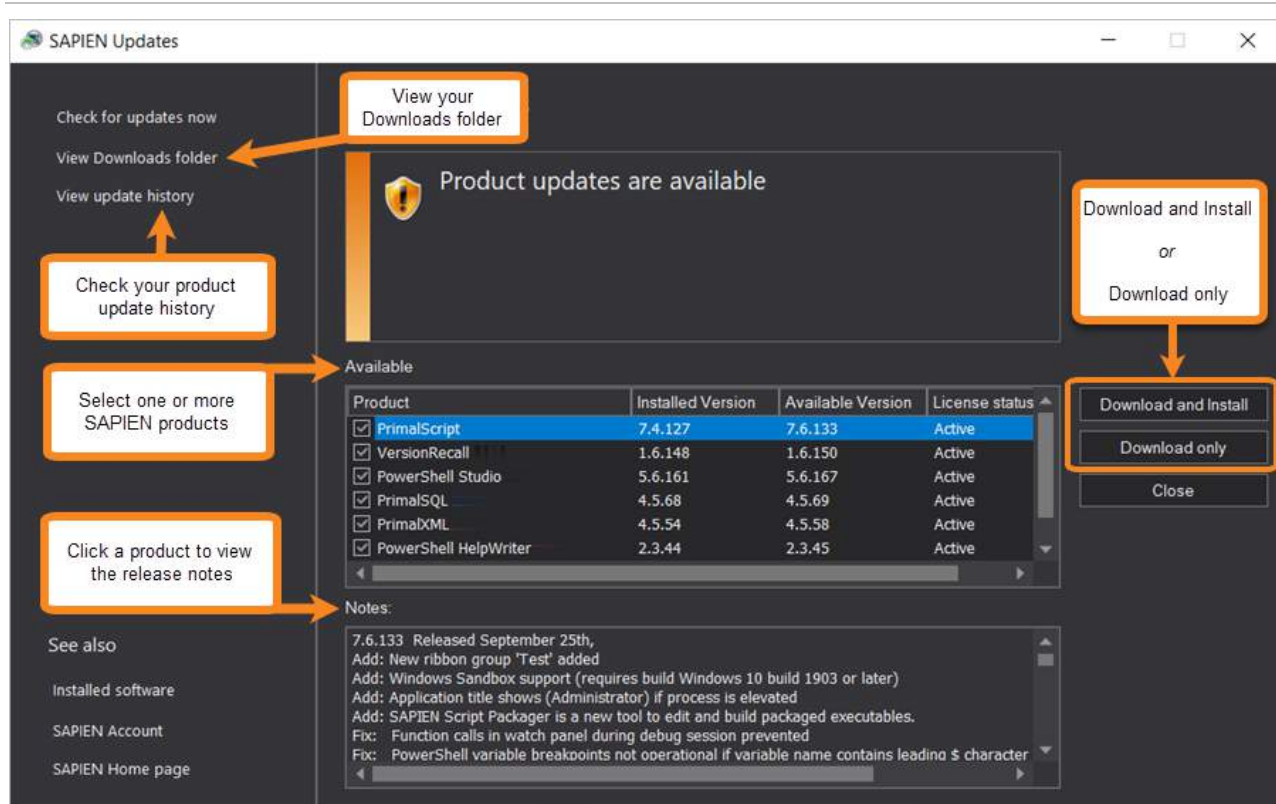
How to access the SAPIEN Updates tool

- On the **Help** or **Tools** ribbon > click **Check Now** or **Check For Updates** (Updates section).

-OR-



- Click the [notification icon](#)¹⁷⁰ above the ribbon > then in the Notifications dialog, click the update notification.

SAPIEN Updates Tool



SAPIEN Updates Tool

SAPIEN Updates - Options

Check for updates now	Immediately checks to see if additional product updates are available.
View Downloads folder	Displays the Downloads folder in File Explorer.
View update history	Displays the history of all downloaded and installed product updates.
Available	<p>Displays a selectable list of available product updates.</p> <p> Select one or more products to Download or Download and Install.</p>
Download and Install	Downloads and installs the updates for the product(s) selected in the Available updates list.
Download only	Downloads the updates for the product(s) selected in the Available updates list.
Close	Closes the SAPIEN Updates tool.
Notes	<p>Displays a brief synopsis of what was changed, added, or fixed for the products selected in the Available window.</p> <p> The build history for all SAPIEN products is available here.</p>

Update On-Demand

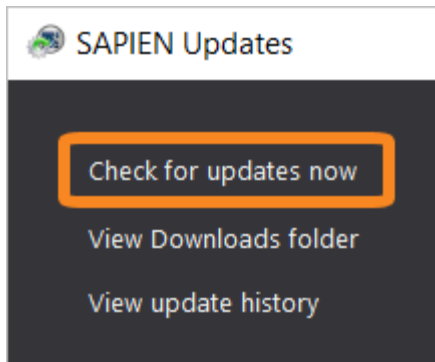
You don't need to wait to be notified when an update is available; you can check for updates at any time. This is particularly useful if you've heard about a new update and want to install it immediately, or if you are ready to start a new project and want to complete all updates before you begin.

How to check for updates on-demand

- On the **Help** or **Tools** ribbon > select **Check Now** or **Check For Updates** to open the SAPIEN Updates tool.

 These instructions may vary between SAPIEN products.

- In the SAPIEN Updates tool, select **Check for updates now**:



The latest product updates are displayed in the SAPIEN Updates **Available** window.

Security and Permissions

Installing updates to programs in a Program Files directory requires the permissions of a member of the Administrators group on the computer. When you click **Download and Install** in the [update tool](#)^[171], or install after downloading, you will be prompted for administrator credentials.

The update tool requires a functioning internet connection and unimpeded access through your internet firewall. For some installations, you might need to create a firewall rule to allow access or make some accommodations.

5.5 Appendices

Appendices for PowerShell ModuleManager Help Manual

[Appendix A: Manual and Product Version](#)^[174]

[Appendix B: Icon License Attribution](#)^[174]

5.5.1 Appendix A: Manual and Product Version

Appendix A Manual and Product Version

This help manual covers the features and functions included in PowerShell ModuleManager version 1.1.3 and earlier.

5.5.2 Appendix B: Icon License Attribution

Appendix B Icon License Attribution

Icons used in this manual are licensed under [Creative Commons Attribution 3.0 Unported \(CC BY 3.0\)](#).

Icons made by [Icomoon](#) from [www.flaticon.com](#) are licensed under [CC 3.0 BY](#):



[Thumb up](#)



[Information](#)



[Warning](#)

6 Frequently Asked Questions

SAPIEN - Frequently Asked Questions

For answers to questions about product licenses, activations, support, and more, view the [SAPIEN Frequently Asked Questions](#).

PowerShell ModuleManager - Frequently Asked Questions

For answers to common questions about PowerShell ModuleManager, view the frequently asked questions below.

What are the PowerShell ModuleManager features?

View the product features on the [PowerShell ModuleManager product page](#).

How can I buy PowerShell ModuleManager?

You can buy PowerShell ModuleManager online with all major credit cards. As soon as your transaction completes, you will be able to [download and install](#) the program.

Online orders:

<https://www.sapien.com/store/powershell-modulemanager>

Worldwide authorized resellers:

<https://www.sapien.com/company/resellers>

Is there a trial version of PowerShell ModuleManager?

Want to try before you buy? You can [download a trial version here](#).

How can I get answers to my questions before I buy the software?

For answers to your pre-order questions, check out the [SAPIEN Frequently Asked Questions](#) or post in the [Trial Software / Pre-sales Technical Questions](#) forum.

How do I install PowerShell ModuleManager?

To get started using PowerShell ModuleManager, follow [these instructions to download and install](#) the program.

Does PowerShell ModuleManager need to run with elevated permissions?

PowerShell ModuleManager needs to run with elevated permissions to update modules stored in the Program Files directory.

The following prompt appears when the program is installed or launched: "*Do you want to allow*

this app to make changes to your device?"

Select **Yes**.

How do I get help troubleshooting installation?

If you encounter problems installing PowerShell ModuleManager, please report them in the [Installation Issues support forum](#).

To output to a log file use these Installer Log parameters: `Installer.exe /exenoui /qn /L*v .\PMG_Install.log`

What version of PowerShell is required?

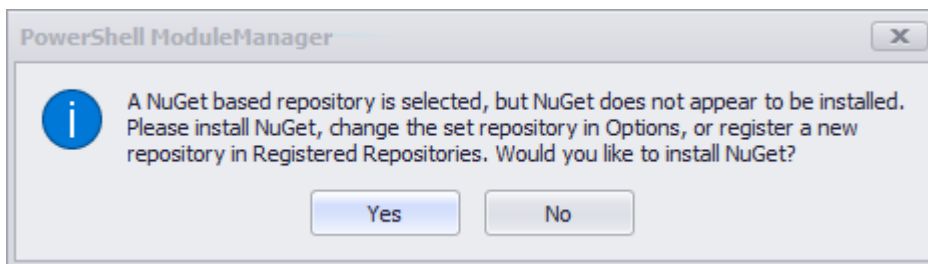
PowerShell v5.0 is required and is, by default, installed with Windows 10. Depending on demand, in the future, PowerShell ModuleManager may support older PowerShell versions.

Are any specific PowerShell modules required?

The PowerShellGet and PackageManagement modules are required. These modules are, by default, installed with Windows 10; we recommend installing the latest versions.

Are there any special requirements to install modules from the PowerShell Gallery?

To install modules from the PowerShell Gallery, [NuGet](#) must be installed via PowerShell. If NuGet is not already installed, PowerShell ModuleManager will prompt to install NuGet for you:



Is an internet connection required to use PowerShell ModuleManager?

An internet connection is required if the PowerShell repository designated in Options is an online source (Options > General > PowerShell > Repository).

How do I update to the latest version of PowerShell ModuleManager?

- See the [SAPIEN Updates](#) ¹⁶⁹ topic in this user manual for software update information.

How can I learn more about the latest PowerShell ModuleManager updates?

- Check out the PowerShell ModuleManager articles on the [SAPIEN blog](#).
- View the PowerShell ModuleManager [version history](#).

How can I submit a product suggestion or feature request?

Submit your PowerShell ModuleManager suggestions and product enhancement ideas on the [Feature Requests](#) page.

How can I get support for PowerShell ModuleManager?

See the [Getting Help](#)^[28] topic in this user manual for support options.