Going down the IT rabbit hole

blog.markic.org/2024/03/25/windows-admin-center-old-vs-new

March 25, 2024

Windows Admin Center installation – old vs new

March 25, 2024 jozemarkic Leave a comment Go to comments
Windows Admin Center (WAC) is a powerful tool for managing Windows Clients, Windows Servers and Windows Clusters. In December 2023, the **Windows Admin Center modernized gateway** (WACmg) was released to public preview through the Windows Server Insider program. This release comes as extra version to the latest generally available release of Windows Admin Center, version 2311. Some of the core differences between the old and modernized versions are:

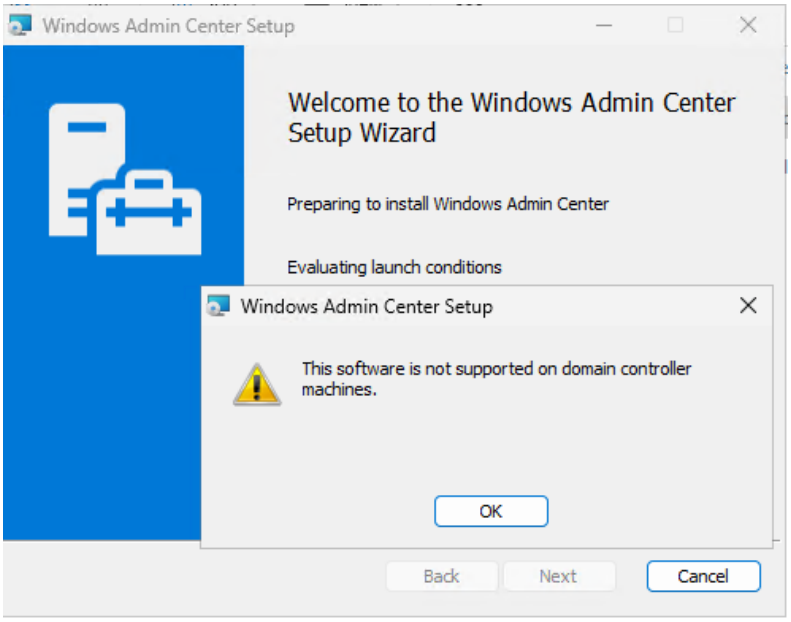
1. **Old Windows Admin Center**:
   o **Purpose**: The old WAC is the modern evolution of traditional "in-box" management tools like Server Manager and MMC (Microsoft Management Console).
   o **Functionality**: It provides control over various aspects of server infrastructure.
   o **Use Case**: Useful for managing servers on private networks that are not connected to the Internet.
   o **Backend**: Utilized the .NET Framework 4.6.2.
   o **Web Server**: Used Katana components for the backend web server.
   o **Extensions**: Supported extensions through gateway plug-ins.
2. **Modernized Windows Admin Center**:
   o **Purpose**: The modernized WAC is an upgraded version with enhanced performance, security, and extensibility.
   o **Backend Upgrade**: Migrated from .NET Framework 4.6.2 to .NET Core.
   o I**nstaller Flexibility**: The installer now offers customization options, including network access settings, trusted hosts, and specifying a fully qualified domain name (FQDN) for the gateway machine.
   o **Microservice Architecture**: Adopts a multi-process, micro-service based architecture. Windows Admin Center starts one process on application startup, which serves as a process manager. Subprocesses are spun up as needed for specific tasks. Gateway plug-ins also run their own subprocesses for better flexibility, scalability, and resilience.
   o **Kestrel Web Server**: Shifted to an ASP.NET Core Kestrel web server, supporting HTTP/2 for reduced latency and improved responsiveness.
   o **Impact on Extensions** : Gateway plug-in extensions are most impacted with the introduction of modernized gateway. If existing extensions use components written with .NET framework version 4.6.2, they will have to be updated to use the .NET Core framework. Official guidance with code samples for building compatible plugins is not yet available (more info here).

In summary, the modernized gateway brings significant improvements to Windows Admin Center, making it more efficient and adaptable for managing your server infrastructure. More information about the features and benefits of WAC modernized gateway is available here: What is the Windows Admin Center modernized gateway (preview)?

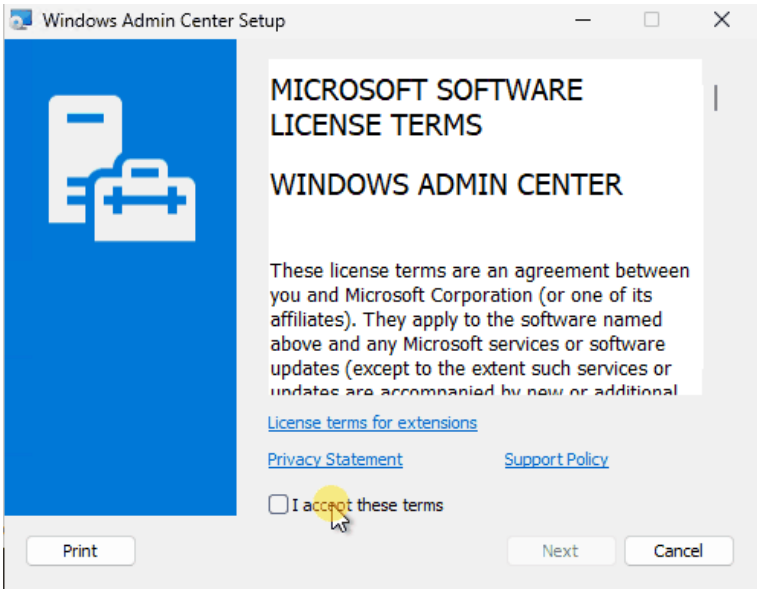
Download WAC:
– https://bit.ly/WAC-Eval (registration required)
– Direct link: https://aka.ms/wacdownload
Download WACmg:
– https://bit.ly/WACmg-Eval (Windows Server Insider registration required)
– Direct link: https://bit.ly/WACModern

WAC installation on domain controller

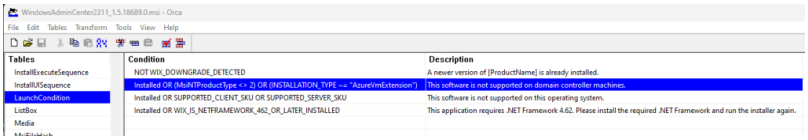
WAC installation on domain controller (DC) is not officially supported:



But that doesn't mean it doesn't work. It works nicely, just without the official support (good enough for testing in my lab environment):



If we want to install WAC on DC, we have to modify original MSI package:



To get the installation working on DC, we can drop one row with the following condition from the LaunchCondition MSI table:

```
Installed OR (MsiNTProductType <> 2) OR (INSTALLATION_TYPE ~= "AzureVmExtension")
```

or we can customize above condition to include all MsiNTProductType(s) with:

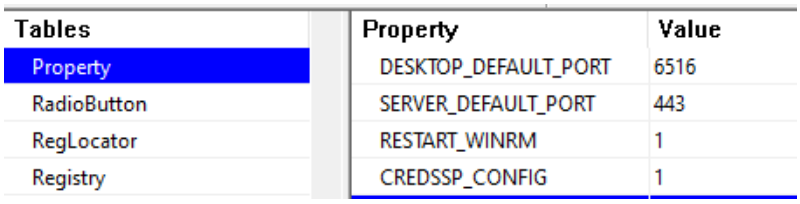
```
Installed OR (MsiNTProductType >= 1) OR (INSTALLATION_TYPE ~= "AzureVmExtension")
```

WAC installation customization

WAC was designed to be installed on desktop and/or server OS, including server core. If installation is done on desktop OS, WAC is installed in "desktop" mode:
– no remote connectivity to WAC service,
– local management of desktop OS,
– remote management of everything supported.
If installation is done on server OS, WAC is installed in "gateway" mode:

– WAC service is available for remote connectivity (ideal with Server Core as host OS),
– local management available only on server with desktop experience,
– remote management of local server and everything else.

Based on host version, setup defaults change during the installation. One good example is default port for WAC service:



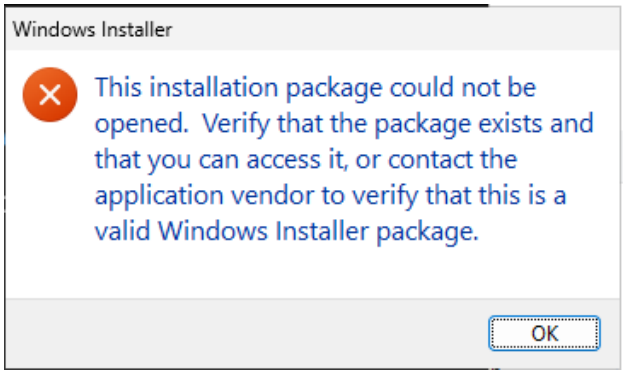
During WAC installation we can customize:
– diagnostic settings (**required** or required+optional)
– use Microsoft Update for updates (yes or **no**)
– modify trusted hosts settings (**yes** or no)
– WinRM over HTTPS only (yes or **no**)
– automatically update WAC (**yes** or no)
– custom port (server: **443**, desktop: **6516** or any)
– SSL certificate (**generate with expiration 60 days** or existing cert)
– HTTP port 80 redirection to HTTPS (yes or **no**)

If we want to change defaults, using GUI installer is easy. If we want to automate customization so that we can run installer remotely or silently, we have to go and read installation documentation. Installation parameters mentioned in the official documentation:

- **RESTART_WINRM**=0 -> do not restart WinRM service during installation (WinRM must be restarted for Windows Admin Center to function)
- **SME_PORT**=443
- **SSL_CERTIFICATE_OPTION**=generate
- **SSL_CERTIFICATE_OPTION**=installed
- **SME_THUMBPRINT**=<thumbprint>

One installation warning, that is useful to know:
Don't invoke `msiexec` from PowerShell using dot-slash relative path notation (like, `.\<WindowsAdminCenterInstallerName>.msi`). That notation isn't supported, the installation will fail. Remove the `.\` prefix or specify the full path to the MSI.

If we don't follow the official documentation and still try to run the installer using dot-slash relative path notation, below is the error that will stop the installation:



If we want to find some additional parameters, we can use to customize and automate installation, we can check official documentation for high availability deployment of WAC. There we can find link to Windows Admin Center HA Script zip file. Script is using two extra msi parameters when it finds existing WAC installation:
– REINSTALLMODE=amus
– REINSTALL=ALL

If we want to find every possible parameter that WAC MSI supports, we have to go back to MSI tables and look at Property table:

Property	Value
UpgradeCode	{AF3E4932-2D63-46F8-A37F-B6ACFD5378CD}
SME_TELEMETRY_PRIVACY_TYPE_INTERNAL	Required
WEBSOCKET_VALIDATION_OVERRIDE_INTERNAL	*
SET_TRUSTED_HOSTS_CLIENT	true
SET_TRUSTED_HOSTS	true
SSL_CERTIFICATE_OPTION	generate
MS_UPDATE_OPT_IN	no
[illegible]	[illegible]
DESKTOP_DEFAULT_PORT	6516
SERVER_DEFAULT_PORT	443
RESTART_WINRM	1
CREDSSP_CONFIG	1
[illegible]	[illegible]

There is one property in this table: **SecureCustomProperties**. This one has few extra properties that could be used to customize the behaviour of the installer:
CHK_REDIRECT_PORT_80;CORS_ORIGINS;CSP_FRAME_ANCESTORS;DEV_MODE;INSTALLATION_TYPE;LEGACY_CERT;NETFRAM

WAC installation automation

We can find some nice PowerShell examples for automated WAC installation in MSLab scenario: Windows Admin Center and Enterprise CA:

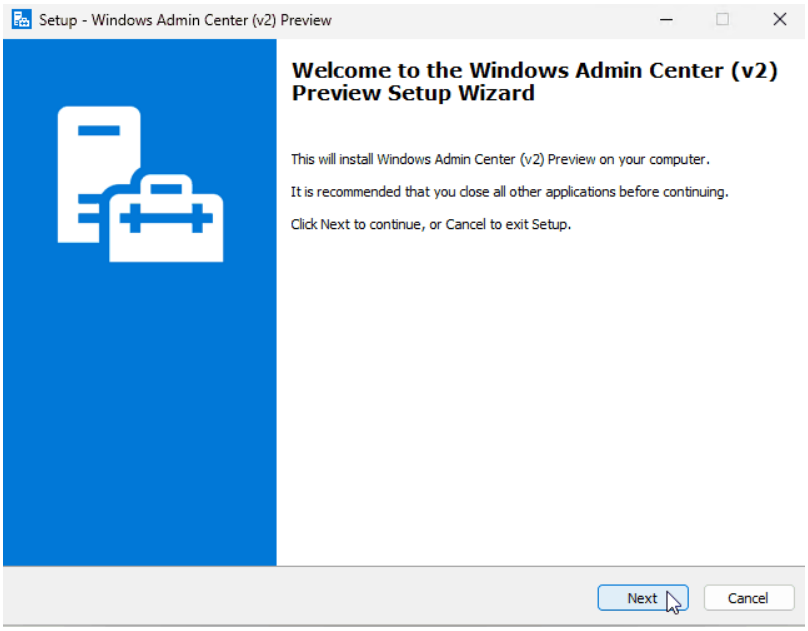
 # Desktop mode installation on Windows Client (with self-signed certificate)
```
Start-Process msiexec.exe -Wait -ArgumentList "/i C:\WAC\WindowsAdminCenter.msi /qn /L*v
C:\WAC\WindowsAdminCenter.log SME_PORT=6516 SSL_CERTIFICATE_OPTION=generate"
```

 # Gateway mode installation (with self-signed certificate)
```
Start-Process msiexec.exe -Wait -ArgumentList "/i C:\WAC\WindowsAdminCenter.msi /qn /L*v C:\WAC\WindowsAdminCenter.log
REGISTRY_REDIRECT_PORT_80=1 SME_PORT=443 SSL_CERTIFICATE_OPTION=generate"
```

 # Gateway mode installation (with custom certificate)
```
Start-Process msiexec.exe -Wait -ArgumentList "/i C:\WAC\WindowsAdminCenter.msi /qn /L*v
C:\WAC\WindowsAdminCenter.log REGISTRY_REDIRECT_PORT_80=1 SME_PORT=443 SSL_CERTIFICATE_OPTION=installed
SME_THUMBPRINT=$Thumbprint"
```

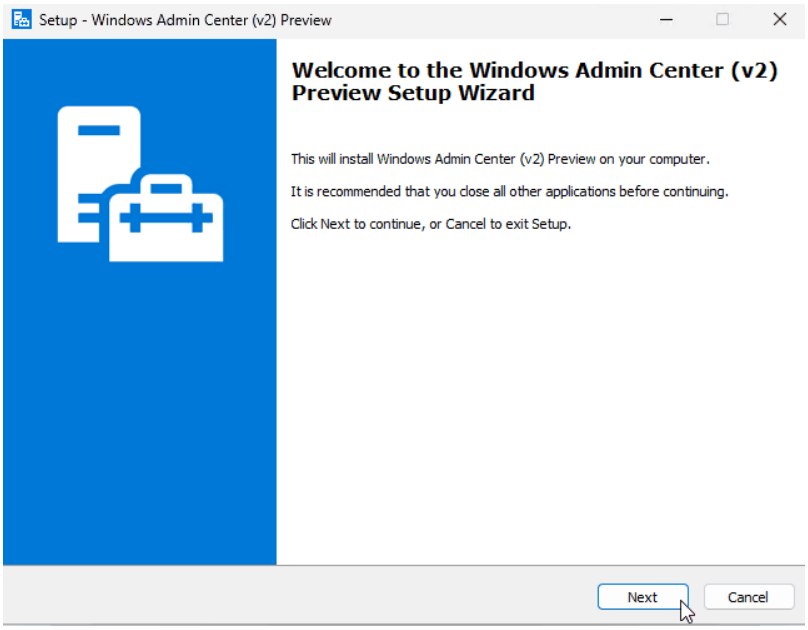
WACmg installation customization

Everything we learned about WAC installation and customization is no good to us with **Windows Admin Center modernized gateway** (WACmg). WACmg installer doesn't use MSI anymore – now the installer is build with Inno Setup. Installer was designed to offer more flexibility to the admin doing the installation (more info here and here):



During the installation we can choose one of the two express options (local or remote) or custom setup. Both express options will skip customization of:
– Sign-in authentication mode (**FormLogin**, WindowsAuthentication, or AadSso)
– Host access network names
– Internal and external network ports

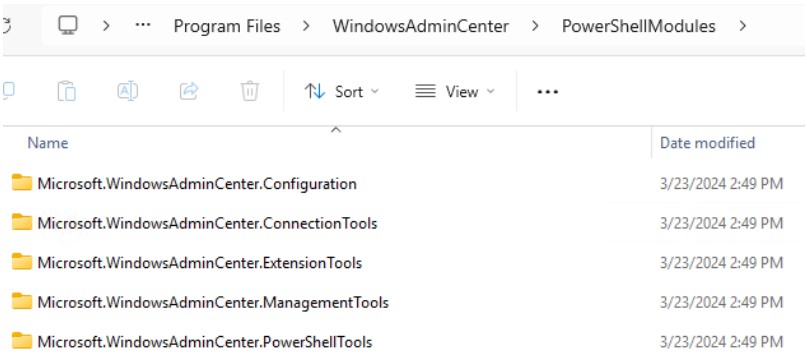
– Endpoint FQDN
– Trusted hosts mode (**allow access to any computer** or only trusted domain computers)
– WinRM over HTTPS (**HTTP** or WinRM over HTTPS)
If we want to configure any of the above options, we have to select custom setup:



AadSso Sign-in authentication mode is only available with PowerShell customization post-installation.

Good to know about the installer:

- We can install a modernized gateway build of Windows Admin Center side-by-side with a legacy gateway build if we don't choose the same ports for both installations.
- We can install WACmg in both modes (local and remote setup) on desktop and server OS.
- WACmg installation not supported on DC but works out of the box – requires some post installation customization.
- Any and all install parameters can be changed post installation with PowerShell module Microsoft.WindowsAdminCenter.Configuration (by default available in C:\Program Files\WindowsAdminCenter\PowerShellModules\Microsoft.WindowsAdminCenter.Configuration)
- Detailed installation log with all PowerShell functions executed during the installation available in %localappdata%\temp.
- New PowerShell modules with 86 functions for WAC customization:



If we want to see the details of the installation, we can check Setup Log file ("Setup Log yyyy-MM-dd #001") available in %localappdata%\temp. We can also unpack original installer and find install_script.iss inside of the unpacked folder.

My custom (default) installation executed PowerShell functions (in order):

```
Set-WACNetworkServiceAccess
Enable-WACPSRemoting
Remove-WACSelfSignedCertificates
New-WACSelfSignedCertificate
Copy-WACTempUpdaterProcessFiles
Register-WACUpdaterScheduledTask
Register-WACService -Automatic
Set-WACInstallDate
Set-WACFileVersion -FileVersion 2.0.1.0
Set-WACNuGetVersion -NuGetVersion 2.0.0
Set-WACWinRmTrustedHosts -TrustAll
Set-WACWinRmOverHttps -Enable:$false
Set-WACSoftwareUpdateMode -Mode Automatic
```

```
Set-WACTelemetryPrivacy -Mode Required
Set-WACLoginMode -Mode FormLogin
Set-WACSubjectName
Set-WACCertificateAcl
Set-WACHttpsPorts -WacPort 443 -ServicePortRangeStart 6601 -ServicePortRangeEnd 6610
Register-WACFirewallRule -Port 443
Register-WACLocalCredSSP
Set-WACServiceSecurityDescriptor
Initialize-WACDatabase
Import-WACExistingExtensions
Import-WACExistingPlugins
New-WACEventLog
```

WACmg installation automation

Because WACmg installer (WindowsAdminCenterModernizedGWPublicPreview.exe) is build with Inno Setup, we can list all supported installer parameters with:

```
WindowsAdminCenterModernizedGWPublicPreview.exe /HELP
```

or

```
WindowsAdminCenterModernizedGWPublicPreview.exe /?
```

To run silent installation (no interactive prompts, use setup defaults) we can run:

```
Start-Process "C:\WAC\WindowsAdminCenterModernizedGWPublicPreview.exe" -wait -ArgumentList
"/log=C:\WAC\WindowsAdminCenterModernizedGWPublicPreview.log /silent"
```

To run very silent installation (no interactive prompts and no progress interface, use setup defaults) we can run:

```
Start-Process "C:\WAC\WindowsAdminCenterModernizedGWPublicPreview.exe" -wait -ArgumentList
"/log=C:\WAC\WindowsAdminCenterModernizedGWPublicPreview.log /verysilent"
```

Setup programs build with Inno setup can support automation of installer parameters by specifying custom INF file. We can generate this INF file with /SAVEINF parameter:

```
Start-Process "C:\WAC\WindowsAdminCenterModernizedGWPublicPreview.exe" -wait -ArgumentList
"/log=C:\WAC\WindowsAdminCenterModernizedGWPublicPreview.log
/SAVEINF=C:\WAC\WindowsAdminCenterModernizedGWPublicPreview.inf"
```

Sadly, this does not work for custom parameters that are used in `WindowsAdminCenterModernizedGWPublicPreview.exe` installer (most of the installation wizard parameters are custom parameters):

WACmg INF with default values:

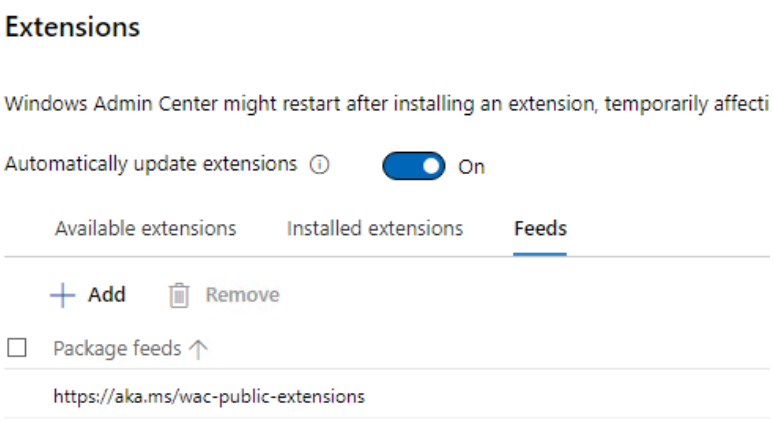
```
[Setup]
Lang=en
Dir=C:\Program Files\WindowsAdminCenter
Group=Windows Admin Center (v2) Preview
NoIcons=0
Tasks=desktopshortcut
```

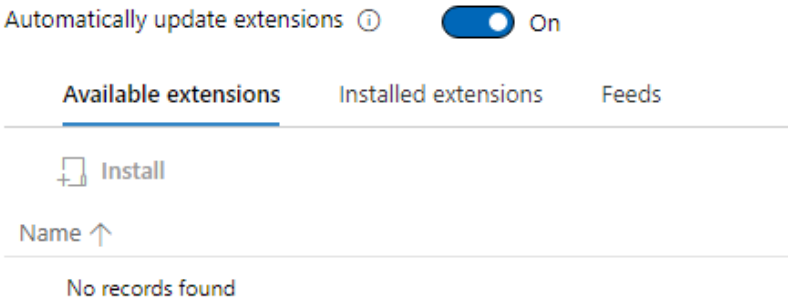
Custom parameters are supported with Inno setup installers and can be used if the author (developer) decides to implement them (as far as I can tell, this is not yet the case for WACMg installer – maybe in GA version of installer)…

To automate custom installation of Windows Admin Center modernized gateway, we have to run silent installer and do all the needed customization post-installation with provided PowerShell functions (functions listed above (for default express installation) are sorted in order they are executed during the WindowsAdminCenterModernizedGWPublicPreview.exe installation).

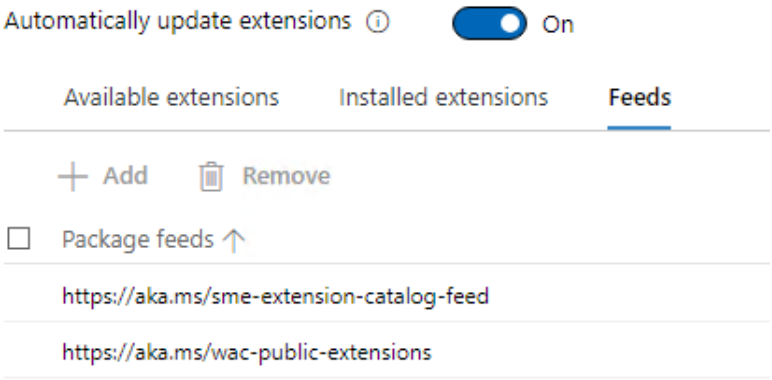
WACmg customization

Currently the installer doesn't configure the extension feed for the modernized gateway. Extensions not included in the Windows Admin Center installer, including external partner extensions, aren't available unless you add an extension feed.
Default feed configuration:





Add custom feed https://aka.ms/sme-extension-catalog-feed



Name ↑
advantech.ipmi-tool
biitops.changes
biitops.inventory
cm.control-panel
DataON.MUST
DataON.MUSTPro
dell-emc.openmanage-integration
dell.apex-cp
dell.powerstore-manager
dellemc.dellemcpowerpath
fujitsu.sme.infrastructure-manager
fujitsu.sme.serverview.health-extension
fujitsu.sme.serverview.raid-extension
hpe.hci.snap-in
hpe.oneview.wac
hpe.storage-manager
hpe.wac
intel.sdptool
lnvgy_sw_xclarity_integrator_for_wac
marvell.qlogicfcqconvergeconsole
microsoft.diagnostic
microsoft.remote-support
microsoft.security
msft.azure.cloud-shell
msft.azure.security-center
msft.azurestack.1p-monitoring
msft.iis.iis-management
msft.sdn.load-balancers
msft.sdn.public-ip-address
msft.sme.active-directory
msft.sme.containers
msft.sme.dhcp
msft.sme.dns
msft.sme.event-viewer
msft.sme.gpu
msft.sme.netatc
msft.sme.sdn.route-tables
msft.sme.subnet-stretch
nec.esmpro.wac-extension
purestorage

QCT-Management-Suite

sccm.control-panel

secureguard.hardware-update

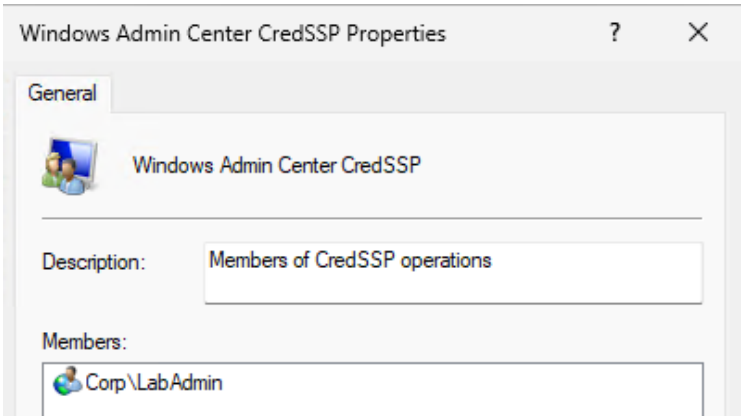
thomas-krenn.hardware-update

thomas-krenn.s2d-micro-cluster

WACmg installation on domain controller

WACmg installation works on DC but then exits with an error. In log files we can find error message:
Register-WACLocalCredSSP: Failed to register CredSSP session configuration.

If we go and check Register-WACLocalCredSSP function, we'll see that it fails because it tries to create new local group (which is not possible on DC):



Because this group is not created, function can't create PS Session Configuration file and due to this failure, it can't register new PS Session Configuration. To get this working, we can update function to use domain group or remove this part from function.

WACmg remote installation

If we try to run WACmg installation remotely (/silent or /verysilent), the installer will stop at the Register-WACLocalCredSSP function. After installer exits with an error, we can open new remote PowerShell session and manually run function `Register-WACLocalCredSSP`. This will drop existing PS sessions so we need to reconnect and finalize WACmg configuration with few more functions:

```
Set-WACServiceSecurityDescriptor
Initialize-WACDatabase
Import-WACExistingExtensions
Import-WACExistingPlugins
New-WACEventLog
Test-WACInstallationFailure 'C:\ProgramData\WindowsAdminCenter\Logs\Configuration.log'
Start-WACService
```

WAC and WACmg with custom certificate

If we want to use custom certificate with WAC service, we need to allow Network Service account access to private key of the certificate used. With WACmg installer we got new PowerShell function we can use to set the required permissions: `Set-WACCertificateAcl`
Set-WACCertificateAcl function modifies the access control list of the certificate identified by the given subject name to grant full control permissions to the Network Service account.

Categories: , Solving Windows Problems, Windows, Windows 10, Windows 11, Windows Admin Center, Windows Client, Windows Server, Windows Server 2025 Tags: Microsoft, PowerShell, WAC, Windows