Windowsの運用管理を快適にする
10の裏ワザ／表ワザ

山市 良 著

目次

免責事項

本書に記載された情報は、予告無しに変更される場合があります。セイ・テクノロジーズ株式会社は、本書に関していかなる種類の保証 (商用性および特定の目的への適合性の黙示の保証を含みますが、これに限定されません) もいたしません。
セイ・テクノロジーズ株式会社は、本書に含まれた誤謬に関しての責任や、本書の提供、履行および使用に関して偶発的または間接的に起こる損害に対して、責任を負わないものとします。

著作権

商標

文中の社名、製品名、サービス名等は各社の商標または登録商標である場合があります。
なお、本文および図表中では「™ (Trademark)」、「® (Registered Trademark)」を明記しておりません。

注記

本文にある下記は、コマンドラインを示し、コピーして利用する際は ↲ を削除してください。

> Get-ComputerInfo | Select WindowsProductName,WindowsVersion, OsBuildNumber ↲
>
> WMIC OS Get Caption, Version ↲

情報更新日

本書は2022年6月22日現在の情報です。

Windowsの運用管理を快適にする10の裏ワザ／表ワザ

企業のIT化をさらに進めるためには、システムの安定運用をどう実現するかが鍵になります。しかし、クラウドとの連携や仮想化、コンテナー化など複雑化するシステム環境、リモートワークやモバイル対応など多様化する利用形態、ますます高度化するセキュリティ脅威の中、Windows ServerとWindowsの安定運用に影響するさまざまな要因が増えています。本書では特別なツールを使わずに、可能な限りWindows標準の機能を利用した、運用管理に役立つ10のテクニックを紹介します。本書では、Windows 10およびWindows Server 2016以降を対象としていますが、テクニックの多くはそれ以前のバージョンにも利用できる場合があります。

1.更新管理に役立つバージョン、ビルド情報の取得

Windows 10およびWindows Server 2016以降は、半期チャネル（Semi-Annual Channel、SAC）と長期サービスチャネル（LTSC、旧称LTSB）の2つのサービスチャネルで提供されています（Windows ServerのSACはバージョン1709から）。SACは年に2回、LTSCは数年に1回に新バージョン（Windows 10では機能更新プログラムと呼びます）がリリースされ、SACのサービス期間は現在、原則18か月、下半期リリースのEnterpriseおよびEducationエディションについては30か月、LTSCは10年（メインストリーム5年＋延長サポート5年）となっており、サービス期間中、セキュリティ更新やバグ修正を含む品質更新プログラムが提供されます。

Windows 11 は、年に1回のリリースサイクルとなり、HomeおよびProエディションは24か月、EnterpriseおよびEducationエディションは36か月のサポートが提供されます。Windows 10についても、バージョン21H2以降、年に1回のリリースサイクルに緩和されます。

Windows 10、Windows 11、およびWindows Server 2016以降のバージョン番号は、OSビルド番号（10.0.は省略する場合があります）と1対1で対応します（表1）。

表1　Windows 10およびWindows Server 2016以降のバージョンとOSビルド、サービス期間

サービスチャネル	バージョン	OSビルド	サービス期間
SAC	なし（便宜上1507）	10.0.10240.x	既に終了
LTSC（LTSB）	2015	10.0.10240.x	Enterprise LTSB 10年
SAC	1511	10.0.10586.x	既に終了
SAC	1607	10.0.14393.x	既に終了
LTSC（LTSB）	2016（1607）	10.0.14393.x	Enterprise LTSB 10年 Server 10年
SAC	1703	10.0.15063.x	既に終了
SAC	1709	10.0.16299.x	既に終了
SAC	1803	10.0.17134.x	既に終了*3
SAC	1809	10.0.17763.x	既に終了*3
LTSC	2019（1809）	10.0.17763.x	Enterprise LTSC 10年 Server 10年
SAC	1903	10.0.18362.x	全エディション 18か月
SAC	1909	10.0.18363.x	Home/Pro/Server 18か月*3 Enterprise/Education 30か月
SAC	2004 （VB_RELEASE）	10.0.19041.x	全エディション 18か月
SAC	20H2 （VB_RELEASE）	10.0.19042.x	Home/Pro/~~Server~~ 18か月 Enterprise/Education 30か月 Server SAC廃止（2022/08/09 EoS）

サービスチャネル	バージョン	OSビルド	サービス期間
SAC	21H1 （VB_RELEASE）	10.0.19043.x	全エディション 18か月
10 SAC→GAC	21H2 （VB_RELEASE） 2022.10 GA	10.0.19044.x	Home/Pro 18か月 Enterprise/Education 30か月
10 LTSC (Windows 10 Enterprise LTSC 2021)	2021（21H2） （VB_RELEASE） 2022.10 GA	10.0.19044.x	Enterprise LTSC 5年*4 ~~Server 10年~~
11 GAC	21H2 （CO_RELEASE） 2022.10 GA	10.0.22000.x	Home/Pro 24か月 Enterprise/Education 36か月
Server LTSC （Windows Server 2022）	2022（21H2） （FE_RELEASE） 2022.09 GA	10.0.20348.x	Server 10年
11 GAC	22H2 （NI_RELEASE） 2022.秋	未定	Home/Pro 24か月 Enterprise/Education 36か月

*3　2021年5月11日ですべてのサービス期間が終了しました。

*4　Windows 10 Enterprise LTSC 2022はサービス期間が5年に短縮されることが発表されています（IoT Enterprise LTSC 2022は10年のまま）。

WindowsおよびWindows Serverの毎月の品質更新プログラムは、前月までの更新内容を累積した累積更新プログラムとして提供されます。OSビルド番号の最も左に位置するリビジョン番号（表1の.xの部分）は、Windowsの累積更新プログラムによってインクルメントされます。つまり、リビジョン番号を参照すれば、OSの更新状態を判断できるということです。最新の累積更新プログラムがインストールされたWindows 10、Windows 11、Windows Server 2016以降、Azure Stack HCI のOSビルド番号は、以下の release informationのページで確認することができます。また、最新状態にするための累積更新プログラムのKB番号を確認することもできます。

Windows 10 release information
https://docs.microsoft.com/en-us/windows/release-health/release-information

Windows 11 release information
https://docs.microsoft.com/en-us/windows/release-health/windows11-release-information

Windows Server release information
https://docs.microsoft.com/en-us/windows/release-health/windows-server-release-info

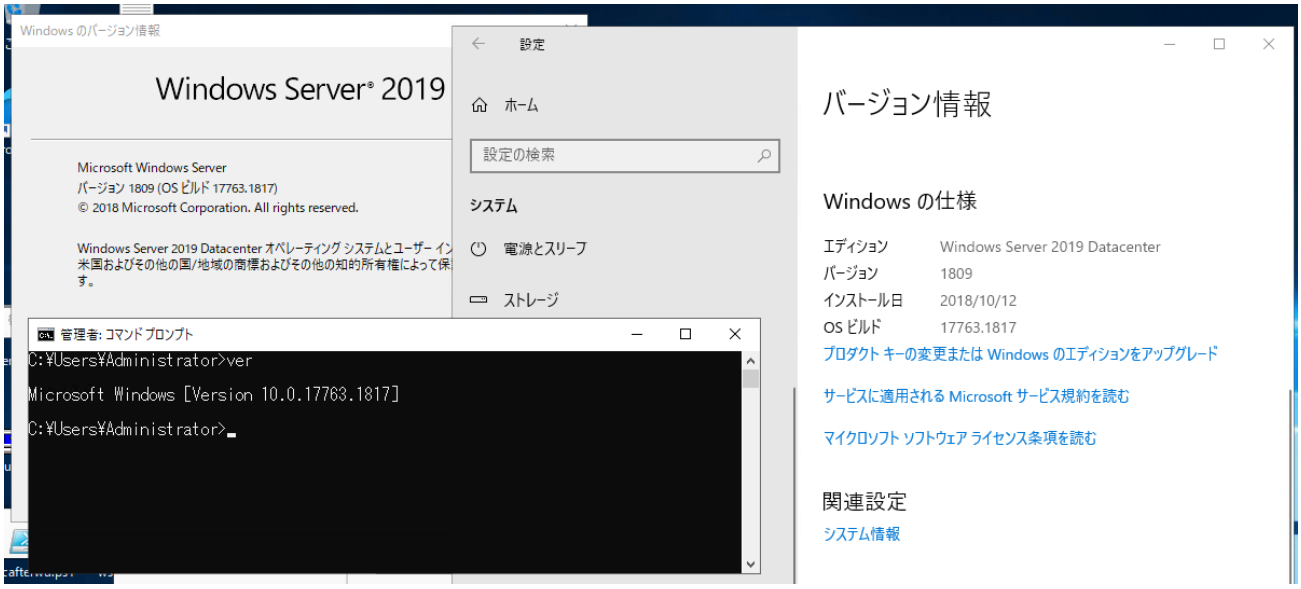
Azure Stack HCI release information
https://docs.microsoft.com/en-us/azure-stack/hci/release-information

製品名、バージョン情報、詳細なビルド番号の情報を取得するさまざまな方法

GUIツールやコマンドラインを使用して、Windowsの製品名、バージョン情報、ビルド番号を取得するいくつかの方法を紹介します。

Windows標準のバージョン情報

Windows 10およびWindows Server 2016以降のデスクトップエクスペリエンス（LTSCのみ）では、［設定］アプリの［システム］の［詳細情報］（バージョン1909以前は［バージョン情報］）やwinverコマンドで開く［Windowsのバージョン情報］から、OSの製品名、バージョン、およびリビジョン番号を含むOSビルド番号を確認することができます。Windows 10およびWindows Serverのバージョン1709からは、コマンドプロンプトのcmdコマンド（またはcmd /c "ver"）の実行結果にもリビジョン番号を含むOSビルド番号を確認できるようになりました（それ以前はリビジョン番号を含まないOSビルド番号）。

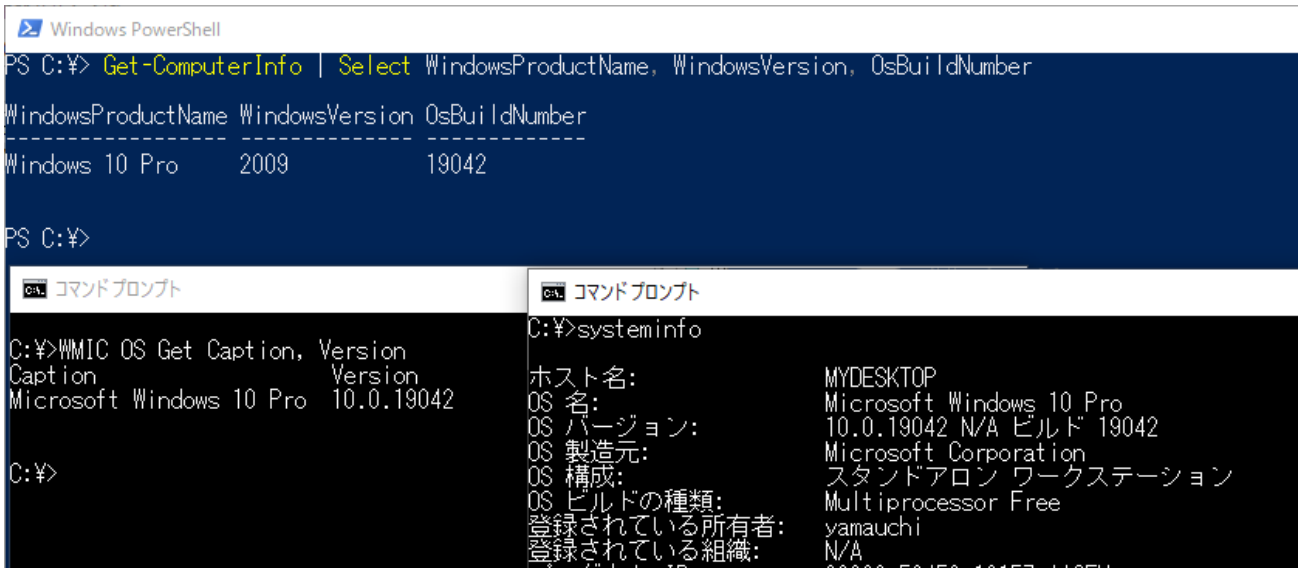


画面1　Windows標準のバージョン情報

その他のコマンドライン

コマンドラインで製品名やOSビルド番号を確認する他の方法としては、Get-ComputerInfoコマンドレット、WMICコマンド、およびSysteminfoコマンドを利用できます。次のコマンドラインを実行すると同等の情報を取得できます。ただし、これらのコマンドでリビジョン番号までは知ることはできません。

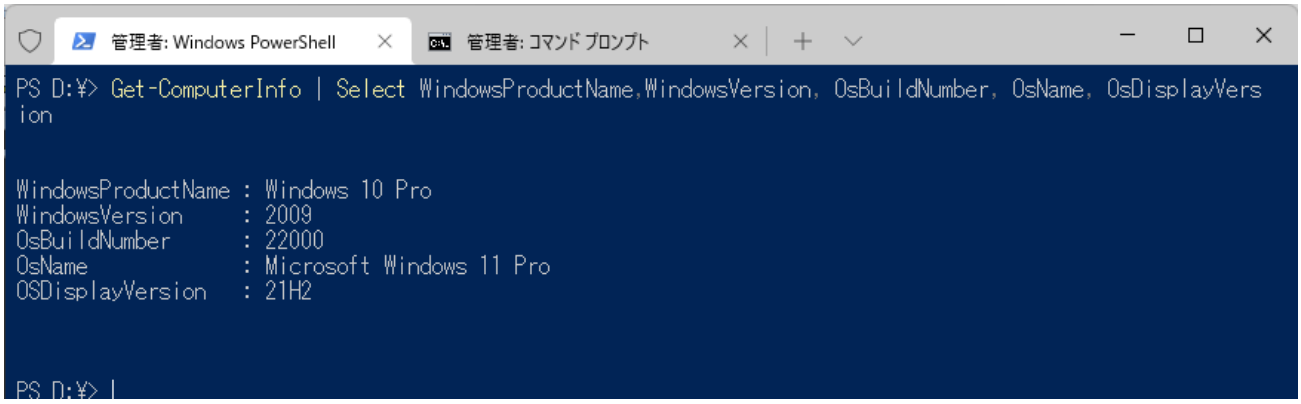
```
Get-ComputerInfo | Select WindowsProductName,WindowsVersion, OsBuildNumber ↲

WMIC OS Get Caption, Version ↲
```



画面2　Get-ComputerInfo、WMIC、Systeminfo コマンドによる製品名、バージョン情報、ビルド番号の取得

Windows 11の場合、Get-ComputerInfoでは OsName と OsDisplayVersion を使用する必要があります（※WindowsVersionは20H2のときの「2009」を最後に更新されていないか、無視されているようです）。

```
Get-ComputerInfo | Select OsName,OsDisplayVersion, OsBuildNumber ↲
```



レジストリから取得

Windowsの製品名、バージョン番号、ビルド番号、リビジョン番号は、レジストリキーHKEY_LOCAL_MACHINE（HKLM）¥SOFTWARE¥Microsoft¥Windows NT¥CurrentVersionにある表2に示すレジストリ値に格納されています。参考として、表2にはGet-ComputerInfo、Systeminfo、WMICコマンドとの対応を含めています。

表2　HKLM¥SOFTWARE¥Microsoft¥Windows NT¥CurrentVersionキー内のバージョン情報関連のレジストリ

	レジストリ値	Get-ComputerInfo	Systeminfo	WMIC
製品名	ProductName	WindowsProductName（11以外） OsName（11を含むすべて）	OS名	Caption
バージョン番号 （2004以前）	ReleaseId	WindowsVersion	なし	なし
バージョン番号 （20H2以降）	DisplayVersion	なし（10までなし、レジストリから取得↓） OsDisplayVersion（11/2022）	なし	なし

	レジストリ値	Get-ComputerInfo	Systeminfo	WMIC
OSビルド	CurrentBuild	OsBuildNumber	OSバージョン	Version
リビジョン番号	UBR	なし	なし	なし

例えば、次の1行のコマンドラインを実行すると、リビジョン番号を含む詳細なビルド番号（例：19042.867）を取得することができます。

```
(Get-ItemProperty  "HKLM:¥SOFTWARE¥Microsoft¥Windows NT¥CurrentVersion").CurrentBuild + "." +(Get-ItemProperty "HKLM:¥SOFTWARE¥Microsoft¥Windows  NT¥CurrentVersion").UBR ↲
```

なお、レジストリ値DisplayVersionは、YYH1/YYH2（YYは西暦下2桁）形式が採用されたWindows 10バージョン20H2から追加されたものであり、以前のバージョンには存在しません。次のコマンドラインを実行すると、DisplayVersionまたはReleaseIdの適切な方からバージョン番号（例：2004、20H2）を取得します。

```
$winver = (Get-ItemProperty  "HKLM:¥SOFTWARE¥Microsoft¥Windows NT¥CurrentVersion" -ErrorAction  SilentlyContinue).DisplayVersion ↲

if($winver -eq $null){ ↲

    $winver = (Get-ItemProperty "HKLM:¥SOFTWARE¥Microsoft¥Windows NT¥CurrentVersion").ReleaseId ↲

} ↲

Write-Host $winver ↲
```

リモートマシン、仮想マシンから情報を取得する方法

リモートのWindowsやWindows Serverに対話的にログオンすることなしに、ネットワーク経由で情報を取得するには、PowerShell RemotingのInvoke-Commandコマンドレットが便利です。

Invoke-Commandコマンドレットは次のように実行します。1行目のコマンドラインで資格情報の入力が要求されるので、リモートコンピューターの管理者ユーザーの資格情報を指定してください。なお、PowerShell Remotingを利用するには、リモート接続される側で事前にEnable-PSRemotingを実行し、PowerShell Remotingを許可しておく必要があります。

```
$cred = Get-Credential ↲

Invoke-Command -ComputerName <コンピューター名> -ScriptBlock {<リモートコンピューターで実行するコマンドライン>} -Credential $cred ↲ ↲
```

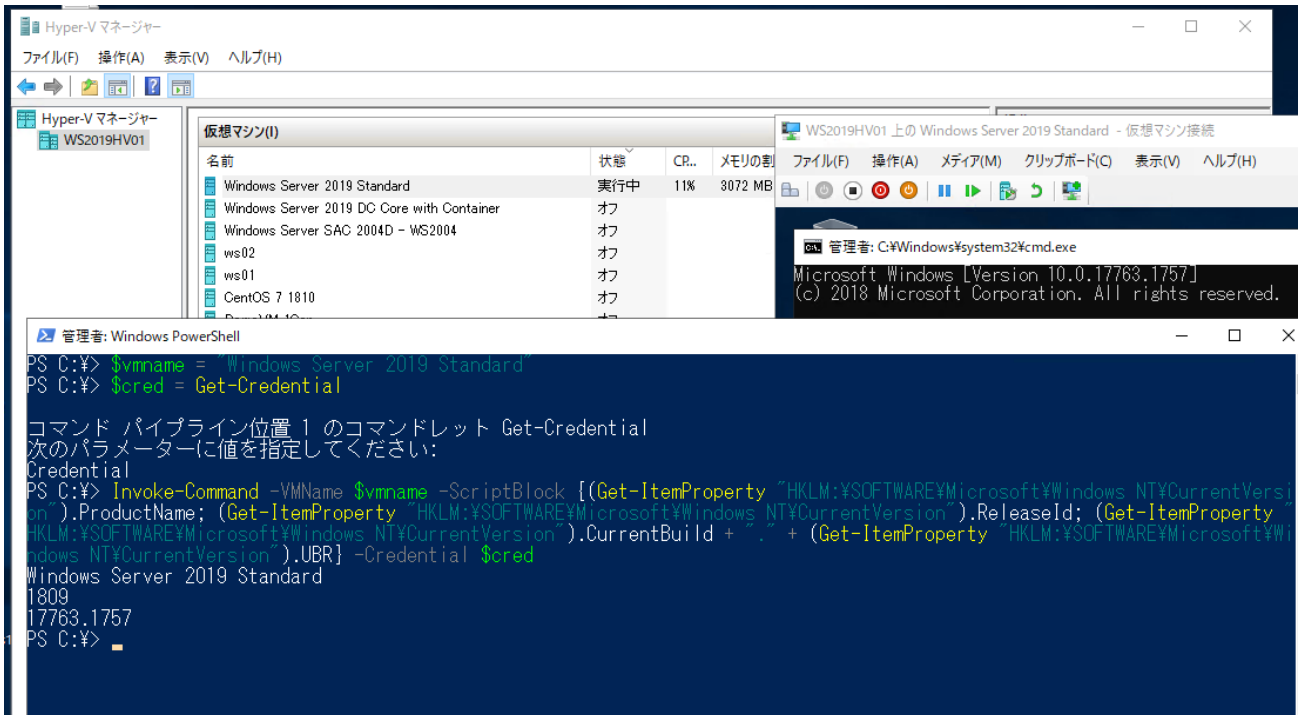
Windows 10またはWindows Server 2016以降のHyper-Vホストでは、Hyper-VホストからWindows 10またはWindows Server 2016以降を実行するWindows仮想マシンのゲストに対してリモート実行できるPowerShell Directを利用できます。PowerShell Remotingによく似ていますが、仮想マシンのネットワーク接続は使用しません。また、ゲスト側でPowerShell Remotingを許可する必要もありません。使い方もPowerShell Remotingによく似ており、-ComputerNameの代わりに-VMNameパラメーターに仮想マシン名を指定します（画面3）。

```
$cred = Get-Credential ↲
```

```
Invoke-Command -VMName <仮想マシン名> -ScriptBlock {<リモートコンピューターで実行するコマンドライン>} -Credential $cred ↲
```

次の画面3の実行例では、スクリプトブロックとして以下のコマンドラインを実行するように指定しています。実行結果として、Windowsの製品名、バージョン番号、およびリビジョン番号を含む詳細なビルド番号の3つの情報を取得できます。

```
(Get-ItemProperty  "HKLM:¥SOFTWARE¥Microsoft¥Windows NT¥CurrentVersion").ProductName;
(Get-ItemProperty "HKLM:¥SOFTWARE¥Microsoft¥Windows  NT¥CurrentVersion").ReleaseId;
(Get-ItemProperty  "HKLM:¥SOFTWARE¥Microsoft¥Windows NT¥CurrentVersion").CurrentBuild
+ "." + (Get-ItemProperty "HKLM:¥SOFTWARE¥Microsoft¥Windows  NT¥CurrentVersion").UBR ↲
```



画面3　Hyper-VホストからWindows仮想マシンのゲストに対するPowerShell Directを利用して、バージョン情報を取得しているところ

PowerShell DirectはWindows 10およびWindows Server 2016以降のHyper-Vゲスト用の統合コンポーネント（Hyper-V PowerShell Direct Service）とやり取りして結果を取得します。そのため、Windows 8.1やWindows Server 2012 R2以前を実行するWindows仮想マシンでは利用できません。また、Linux仮想マシンにクロスプラットフォーム対応のPowerShell 7.0をインストールしても、PowerShell Directで接続できることはありません。

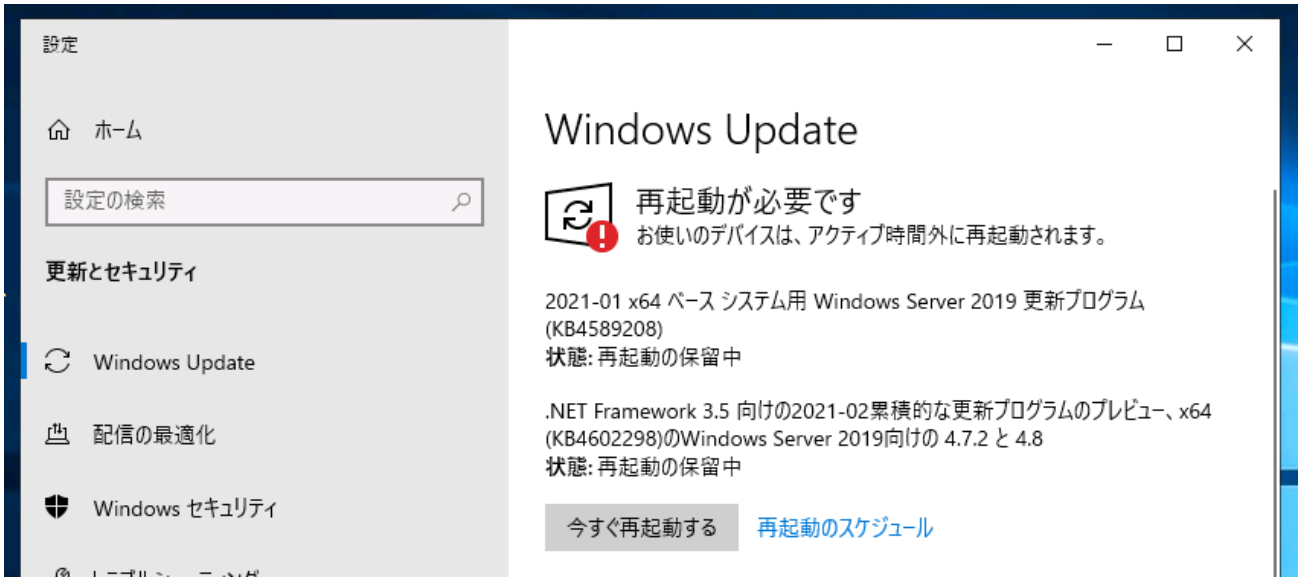
2.Windows Updateはお任せではなく、戦略的に

Windows 10になって、Windows Updateの仕組みは大幅に再設計され、Windows 8.1以前のようなユーザー側でコントロールできる部分が少なくなりました。Windows 10になってからも、新しいバージョンで次々に変更が加えられてきています。

Windows 10のWindows Updateの仕組みは、Windows Server 2016以降のデスクトップエクスペリエンスにも共通のものが搭載されています。中には、長期運用が前提のWindows Serverには余計とも思える機能もあります。例えば、再起動が必要な品質更新プログラムのインストールで再起動待ちになり、そのまま放置すると、アクティブ時間外に自動的に再起動が始まってしまう仕様がWindows Serverにも組み込まれています。また、［更新プログラムのチェック］をクリックすると、インストールが必須ではないオプションの更新プログラムのダウンロードとインストールが始まってしまうのも同じOSビルドのWindows 10と同じです。

WindowsおよびWindows Serverの品質更新プログラムは米国時間で毎月第2火曜日にセキュリティ更新を含む累積更新プログラムがリリースされ（Bリリースと呼ぶことがあります）、自動更新が有効な場合は自動配布されます。2021年4月時点では、Windows 10バージョン1809およびWindows Server 2019以降を対象に、その翌週、または翌々週にWindowsと.NET Frameworkのそれぞれに対してオプションの更新プログラムがリリースされます（Cリリース、累積更新プログラムのプレビューと呼ぶことがあります）。.NET FrameworkのBリリース、Cリリースについては、リリースされない月もあります。

オプションの更新プログラムは新たなセキュリティ更新を含まないため、インストールは任意です。オプションの更新プログラムで修正される問題を抱えていて、できるだけすぐに問題を解消したい、重要な変更点をテストしたいといった場合にのみインストールすればよいのです。しかし、［更新プログラムのチェック］をクリックしてしまうと、意図せず、オプションの更新プログラムを受け取ってしまいます（画面4）。



画面4　［更新プログラムのチェック］をクリックしてしまうと、意図せず、オプションの更新プログラムのダウンロードとインストールが始まってしまうことがある。アクティブ時間外の自動再起動もサーバーの安定運用を妨げる余計な機能

長く変わらず、信頼できるWindows Update Agent（WUA） API

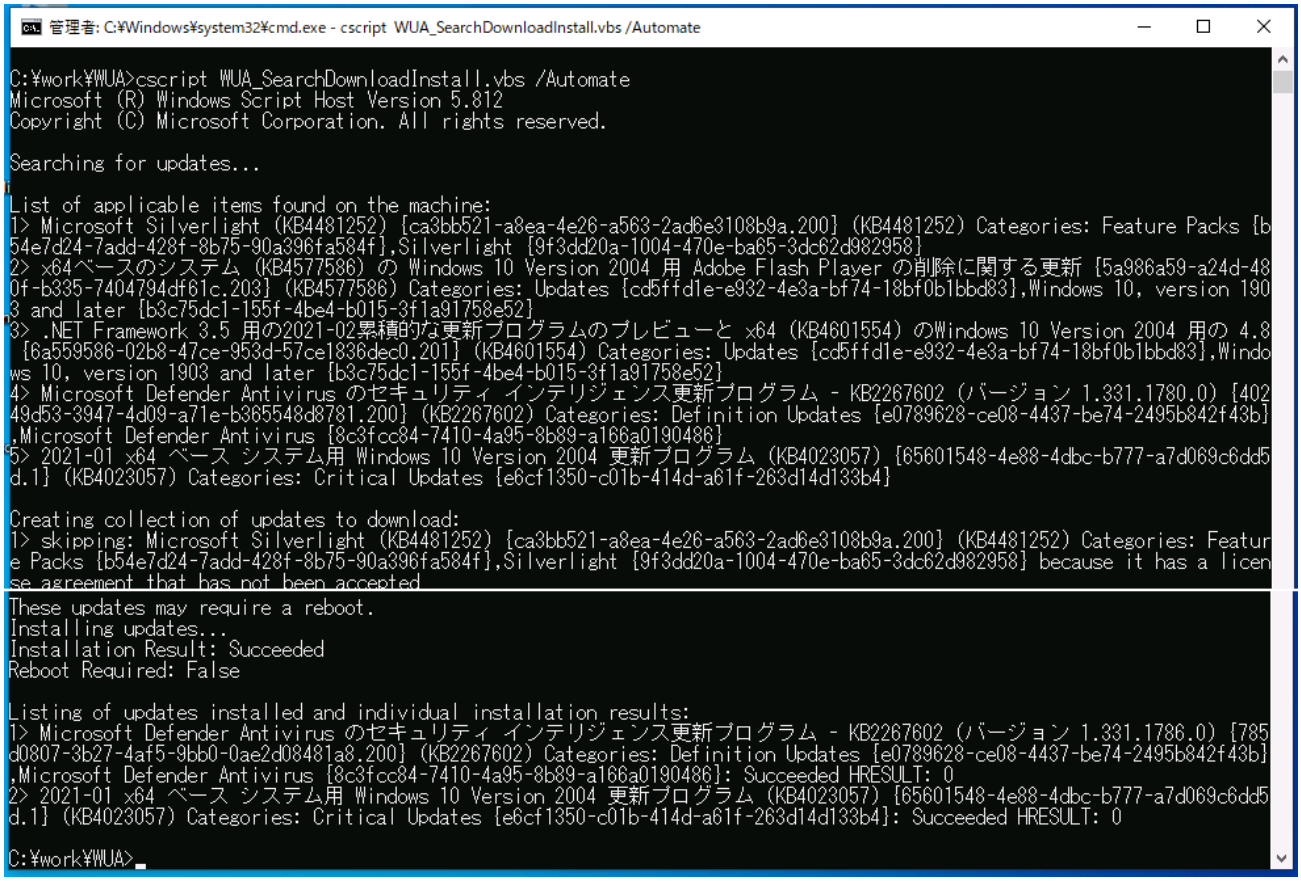
古いバージョンのWindowsから引き継いだ、変更されていない更新機能の1つに、Windows Update Agent（WUA）APIを利用する方法があります。

以下のサイトにWUA APIを利用して更新プログラムの確認、ダウンロード、インストールを行うサンプルスクリプトWUA_SearchDownloadInstall.vbsが公開されています。

Searching, Downloading, and Installing Updates
https://docs.microsoft.com/ja-jp/windows/win32/wua_sdk/searching--downloading--and-installing-updates
以前は更新の各ステップをY/Nの入力で対話的に実行するシンプルなスクリプトでしたが、最近、いくつかの新しいパラメーターをサポートする高機能版に差し替えられました（古いスクリプトは別のURLのページにアーカイブされています*1）。新しいスクリプトをパラメーター無しで実行すると、以前のスクリプトと同様に対話的に更新の確認、インストールの選択、ダウンロード、およびインストールを行うことができます。このスクリプトを利用して更新プログラムをインストールするには、管理者権限で実行する必要があります。

次のように/Automateパラメーターを追加して実行すると、利用可能な更新プログラムを確認し、検出されたすべてをインストールできます（画面7）。ただし、EULA（ライセンス条項）に同意する必要がある更新プログラムについてはスキップされます。

> Cscript WUA_SearchDownloadInstall.vbs /Automate ↲



画面5　WUA_SearchDownloadInstall.vbsによる更新プログラムのインストールの自動化

*2　旧サンプルスクリプトのアーカイブ先 https://docs.microsoft.com/en-us/previous-versions/windows/desktop/aa387102(v%3dvs.85)

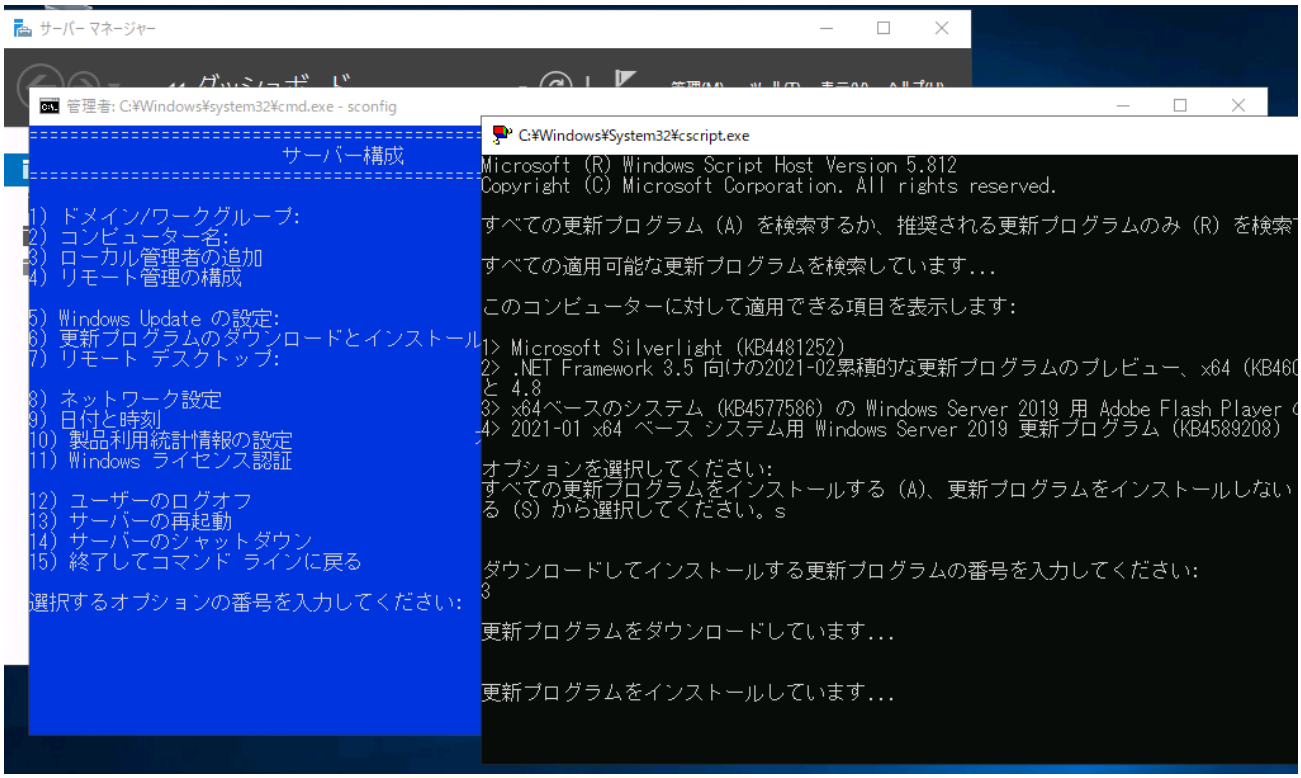
現在のWindows 10およびWindows Server 2019以降は、Windowsの累積更新プログラムのプレビューはWindows 10で新たに追加されたUpdateOrchestratorサービスによって検出されます。WUA APIは、Windowsの累積更新プログラムのプレビュー（Cリリース）や定例外のオプションの更新プログラム（Out-of-bandリリース）は検出しません。.NET Frameworkの累積更新プログラムのプレビュー（Cリリース）はWUA APIの検索対象に入ります。

SconfigユーティリティとWindows Admin CenterはWUA APIで動く

Windows ServerはSconfigというテキストベースのユーティリティを搭載しています。Windows Server 2012以降のSconfigユーティリティはWindows Serverのインストールの種類（GUI使用サーバー/デスクトップエクスペリエンスとServer Core）に関係なく、どちらでも利用できます。Sconfigユーティリティの［6 更新プログラムのダウンロードとインストール］は、WUA APIを利用するWSHスクリプト（C:¥Windows¥System32¥ja-jp¥WUA_SearchDownloadInstall.vbs）として実装されており、利用可能な更新プログラムの検索、インストールする更新プログラムの選択、およびダウンロードとインストールを実行できます（画面6）。本書で紹介したサンプルスクリプトと同じファイル名ですが、内容は異なります。現在の高機能版に差し替えられる前の古いバージョンのWUA_SearchDownloadInstall.vbsをベースとしています。

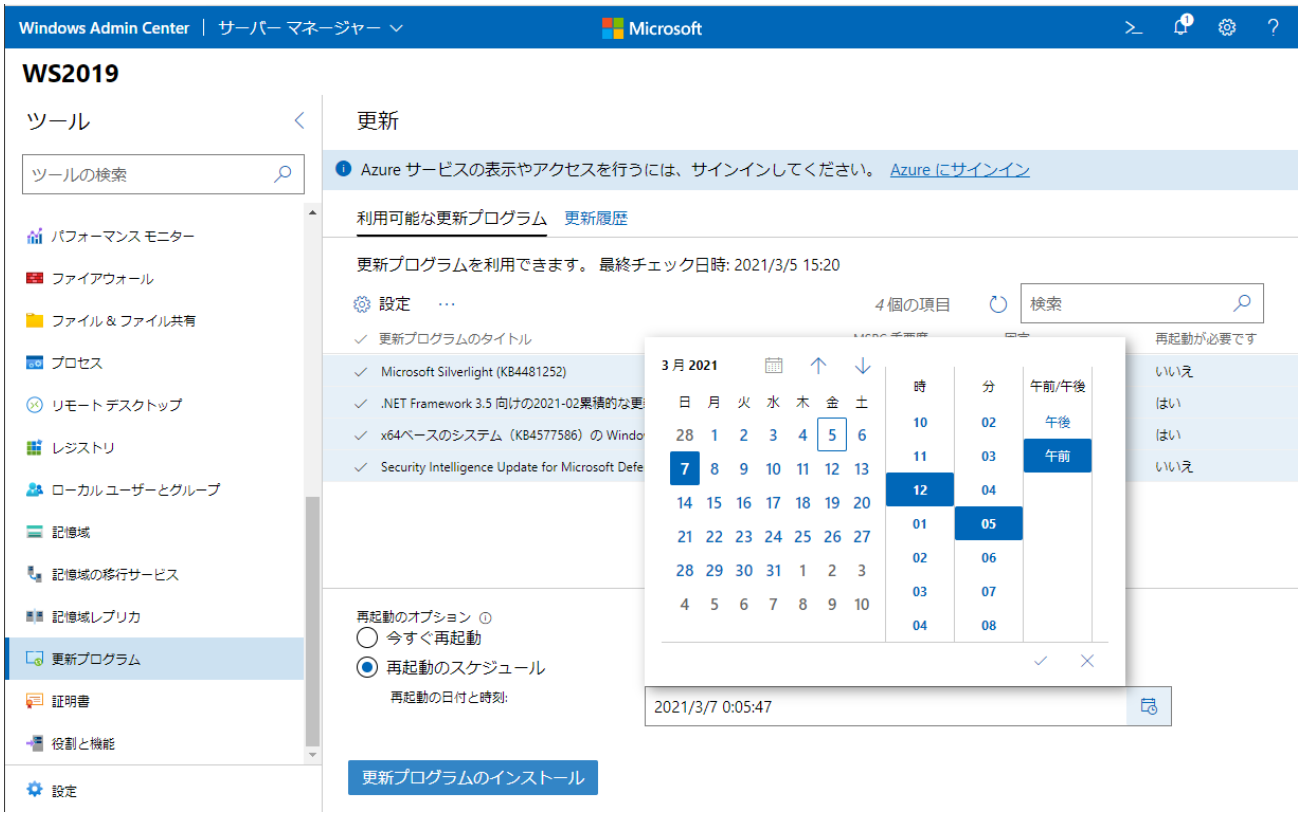
※Windows Server 2022/Azure Stack HCIからはPowerShellで Sconfig が構築されています。

前述したように、WUA APIはWindowsの累積更新プログラムのプレビューを検出しません。そのため、Sconfigを利用すればデスクトップエクスペリエンス環境で意図せずWindowsの累積更新プログラムのプレビューがインストールされるのを防止できます。なお、「推奨される更新プログラムのみ（R）」の検索は現在、期待どおりに機能しなくなりました。これは検索条件（Criteria）としての「AutoSelectOnWebSites=1」が推奨される更新プログラムに設定されない場合があるためです。



画面6　Windows Serverはインストールの種類に関わらず、Sconfigユーティリティを使用可能。［6 更新プログラムのダウンロードとインストール］はWUA APIを利用したスクリプトで実装されている

マイクロソフトは2018年4月、HTML5ベースのサーバー管理アプリWindows Admin Centerをリリースし、概ね半期に一度、更新バージョンをリリースしています。Windows Admin Centerは、Windows Server 2012 R2以降のWindows Server、フェールオーバークラスター、Azure Stack HCIクラスター、およびWindows 10のリモート管理に対応しています。Windows Admin Centerの［更新プログラム］ツールはWUA APIを利用して不足している更新プログラムの確認とインストールの選択、ダウンロードとインストールを行います。再起動が必要な更新プログラムがある場合は、再起動の日時をスケジューリングして実施することができるので、更新プログラムのインストールに起因する再起動が業務時間内に行われるのを回避できます（画面7）。

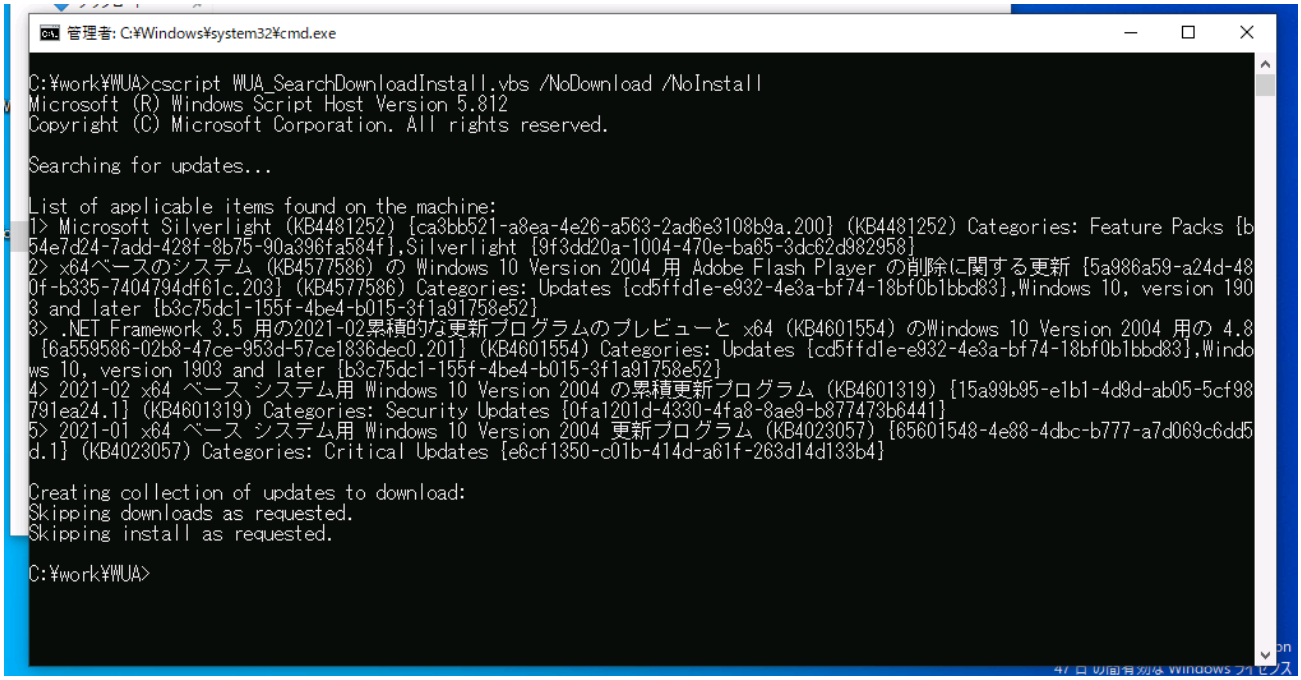


画面7　Windows Admin Centerを利用すると、更新のための再起動を計画的に実施できる

不足しているセキュリティ更新をレポートする（オンラインスキャン）

WUA_SearchDownloadInstall.vbsに/NoDownloadおよび/NoInstallパラメーターを指定して実行すると、ダウンロードとインストールは行わずに、スキャン結果だけをレポートしてくれます。この機能は稼働中のサーバーの更新状態を調査するのに便利です（画面8）。なお、スキャンの実行のみであれば、管理者権限は不要です。

> Cscript WUA_SearchDownloadInstall.vbs /NoDownload /NoInstall ↲



画面8　WUA_SearchDownloadInstall.vbsでスキャンだけを実施し、不足している更新プログラムを調査する

不足しているセキュリティ更新をレポートする（オフラインスキャン）

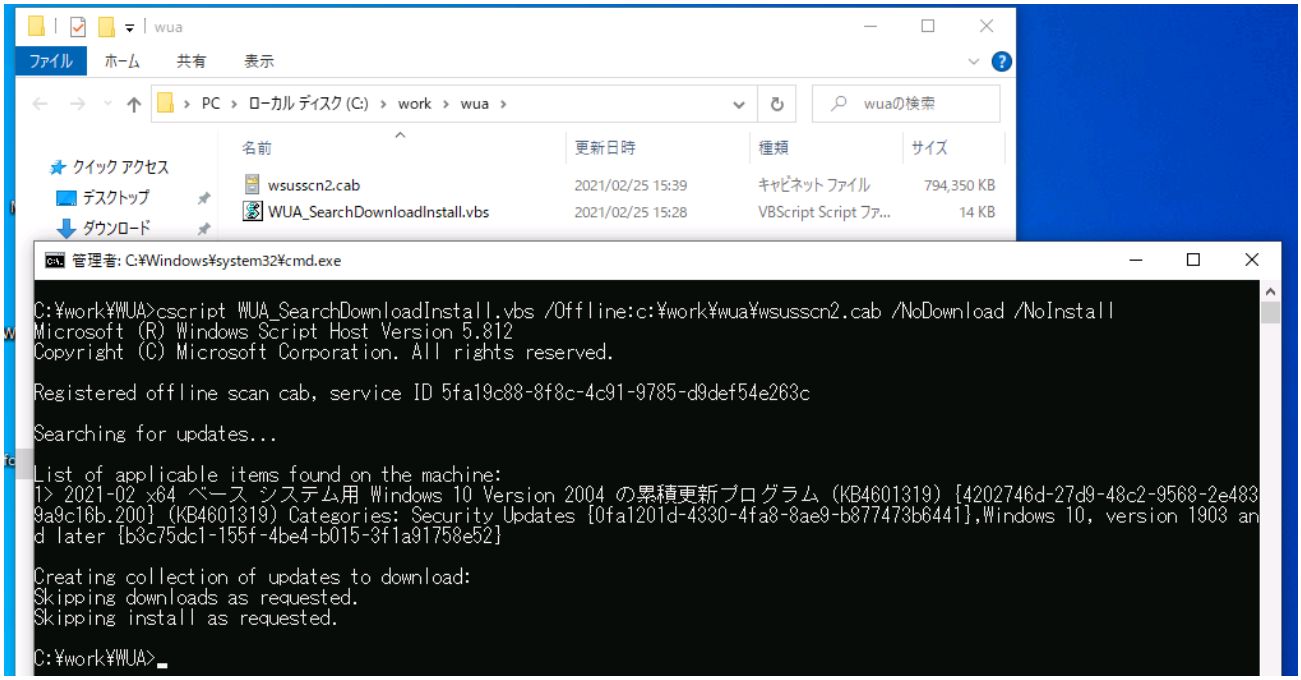
WUA_SearchDownloadInstall.vbsはオフラインスキャンにも対応しています。最新のオフラインスキャンファイルであるWSUSSCN2.cabを入手し、/OfflineパラメーターにWSUSSCN2.cabのパスを指定すると（C:¥work¥WSUSSCN2.cabのように、絶対パスで指定する必要があります）、インターネット接続の制限されたコンピューターに対してスキャンを実行することができます。オフラインスキャンもスキャンの実行のみであれば、管理者権限は不要です。

> Cscript WUA_SearchDownloadInstall.vbs /Offline:<絶対パス>¥WSUSSCN2.cab /NoDownload /NoInstall ↲

最新のWSUSSCN2.cabのダウンロード（米国時間毎月第2火曜日に更新）
http://download.windowsupdate.com/microsoftupdate/v6/wsusscan/wsusscn2.cab

オンラインスキャンは、オプションの更新プログラムをインストールが必要な更新プログラムとして検出しますが、オフラインスキャンは自動配布対象のセキュリティ更新および重要な更新のみを対象としています。次のようにオフラインスキャンと/Automateオプションを組み合わせることで、オフラインスキャンに基づいて本当に必要な更新プログラムだけを自動的にインストールすることができ便利です（画面9）。更新プログラムのダウンロードとインストールまで行う場合、管理者権限で実行する必要があります。

> Cscript WUA_SearchDownloadInstall.vbs /Offline:<絶対パス>¥wsusscn2.cab /Automate



画面9　最新のWSUSSCN2.cabを使用してオフラインスキャンを実施する。/NoDownload /NoInstallパラメーターの代わりに/Automateパラメーターを指定すると、オフラインスキャンに基づいてインストールを行うことも可能

更新履歴を表示する

WindowsおよびWindows Serverのデスクトップエクスペリエンスには、インストールの成功と失敗を含む更新の履歴を表示する機能があります。Windows ServerのServer Coreには更新の履歴を表示する機能はありません。

WUA APIを利用した次のサンプルスクリプトGet-WUHistory.vbsを使用すると、Windows標準の更新の履歴と同等の履歴情報を取得することができます（画面10）。Windows ServerのServer Coreでも利用できます。履歴情報を取得するために、管理者権限は必要ありません。

Get-WUHistory.vbsのスクリプト
注：UTF-8ではなくANSIコード（シフトJIS）で保存すること

```
Set objSession = CreateObject("Microsoft.Update.Session") ↲
Set objSearcher = objSession.CreateUpdateSearcher ↲
intCount = objSearcher.GetTotalHistoryCount ↲
Set colHistory = objSearcher.QueryHistory(0, intCount) ↲
For Each objHistory In colHistory ↲
If objHistory.HResult = 0 then ↲
WScript.Echo DateAdd("h",9,objHistory.Date) & ", " & objHistory.Title & ", 成功" ↲
ElseIf objHistory.HResult = -2145116140 then ↲
WScript.Echo DateAdd("h",9,objHistory.Date) & ", " & objHistory.Title & ", 成功（再起動が必要な更新）" ↲
Else ↲
WScript.Echo DateAdd("h",9,objHistory.Date) & ", " & objHistory.Title & ", 失敗(エラーコード：0x" &
Hex(objHistory.HResult) & ")" ↲
End If ↲
Next ↲
```

このスクリプトはインストールの結果コードに基づいて、成功（0）と失敗（0以外）を判断し、失敗の場合はエラーコードを出力します。エラーコードの一覧は以下のサイトで確認することができます。2つ目のIf文で評価している-2145116140（0x80242014）は再起動の保留状態を示しており、再起動されると結果コードに0がセットされます。しかし、筆者が確認した限り、Windows 10バージョン2004および20H2では再起動後も結果コードが0にセットされない不具合があるようです。そのため、「成功（再起動が必要な更新）」として出力するようにしました。

Windows Update の一般的なエラーと軽減策
https://docs.microsoft.com/ja-jp/windows/deployment/update/windows-update-errors

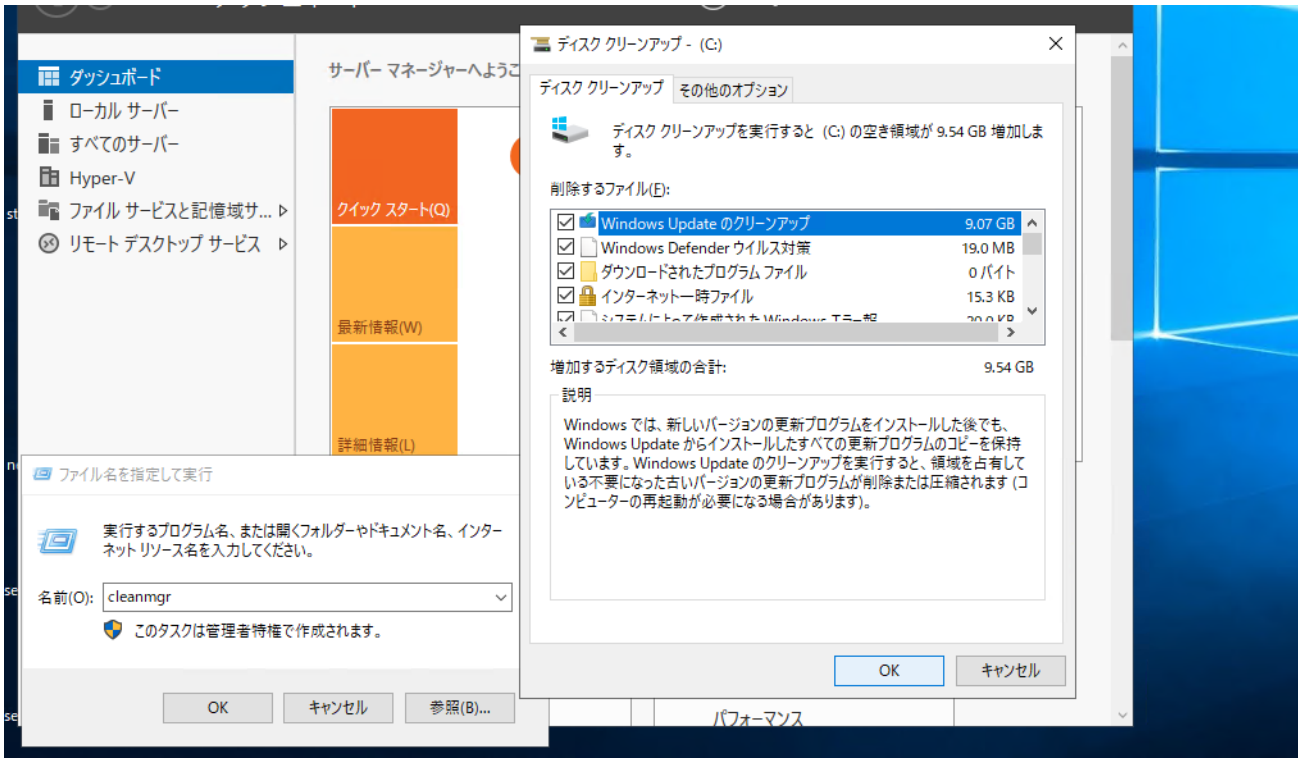
画面10　WUA APIを利用して成功／失敗を含む更新プログラムのインストール履歴を取得する

ディスク領域節約と更新時間の改善に効く、Windows Updateのクリーンアップ

Windows Updateを実施して、安定運用の状態を確認できている場合は、定期的に、例えば数か月に一度にWindows Updateのクリーンアップを実施することをお勧めします。Windows Updateのクリーンアップを実施すると、古いファイルの削除や圧縮が行われ、ディスクの使用領域を数GB単位で解放できる場合があります。また、次回のWindows Updateのパフォーマンス改善を期待できます。ただし、クリーンアップ後は古い更新プログラムをアンインストールできなくなるため、必ず安定運用の状態にあることを確認してから実施してください。

Windows Updateのクリーンアップを実施するには、［ディスククリーンアップ］（cleanmgr.exe）を管理者権限で実行し、［Windows Updateのクリーンアップ］を選択します（画面11）。クリーンアップで増加するディスク領域のサイズは不正確な場合があります（例：1MBや3.66 TBといったありえないサイズを報告）が、クリーンアップの実施には影響はありません。Windows 10の場合、［ディスククリーンアップ］ではなく、［設定］アプリの［システム｜記憶域｜一時ファイル］から［Windows Updateのクリーンアップ］を選択して削除することもできます。

コンピューターのスペックやインストールされている更新プログラムの数にもよりますが、通常、数時間かかかります。100%に近くなっても、それから数時間かかることもあるので、根気よく待ってください。



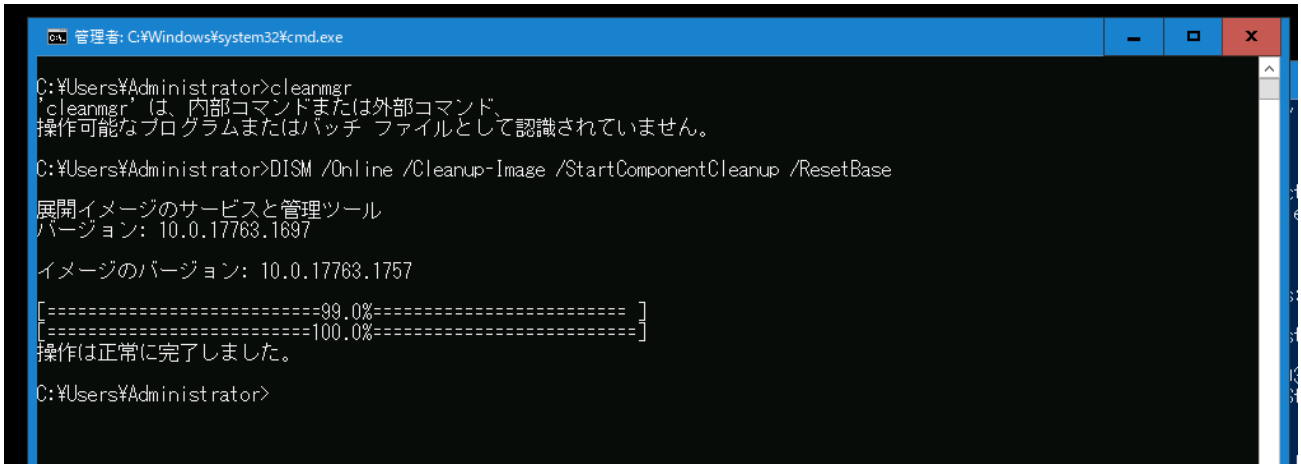
画面11　［ディスククリーンアップ］（cleanmgr.exe）を実行してWindows Updateのクリーンアップを選択

［ディスククリーンアップ］（cleanmgr.exe）は、WindowsおよびWindows Server 2016以降のデスクトップエクスペリエンスに標準で含まれます。Windows Server 2012 R2以前の場合は、サーバーの機能［デスクトップエクスペリエンス］を有効化することでインストールされます。Server Coreでは使用できません。また、理由は不明ですが、Windows Serverの場合は［Windows Updateのクリーンアップ］の項目が表示される場合とされない場合があるようです。

Windows Serverでのディスククリーンアップの使用について
https://docs.microsoft.com/ja-jp/windows-server/storage/file-server/disk-cleanup

Server Coreの場合、および［Windows Updateのクリーンアップ］の項目が表示されない場合は、次のコマンドラインを管理者権限で実行することで、同様の効果が期待できます。/ResetBaseパラメーターは省略できますが、指定することでより多くの領域を解放できます（画面12）。ただし、アンインストール用に保持されていた、更新で置き換えられたバイナリがすべて削除されることに注意してください。/ResetBaseパラメーターを指定しない場合、不要なコンポーネントだけがクリーンアップされ、コンポーネントストア（C:¥Windows¥WinSxS）のサイズが最適化されます。

```
DISM /Online /Cleanup-Image  /StartComponentCleanup /ResetBase ↲
```



画面12　Server CoreでWindows Updateの不要なコンポーネントと更新で置き換えられた古いバイナリをクリーンアップする

%WINDIR%¥SoftwareDistributionのリセット

Windows 10およびWindows Server 2016以降のWindows Updateが原因不明のエラーで失敗を繰り返す場合、更新プログラムのダウンロード先（¥Download）やデータベース（¥DataStore¥Datastore.edbなど）を含む%Windir%¥SoftwareDistributionディレクトリをリセットする（削除して再作成させる）ことで解消する場合があります。 SoftwareDistributionディレクトリの再作成を含むWindows Updateのコンポーネントのリセットの完全な手順については、以下のドキュメントで説明されています。

Windows Update - 追加リソース | Windows Update のコンポーネントをリセットする方法
https://docs.microsoft.com/ja-jp/windows/deployment/update/windows-update-resources#how-do-i-reset-windows-update-components

上記の手順は複雑ですが、SoftwareDistributionのリセットだけで多くの問題が解消します。それには、コマンドプロンプトを管理者として開き、次の一連のコマンドラインを実行してWindows Update関連のサービスを停止し（ほとんどの場合、wuauservの停止のみでよい）、%WINDIR%¥SoftwareDistributionディレクトリをリネーム後に削除します。

```
NET STOP wuauserv ↲
NET STOP bits ↲
NET STOP cryptsvc ↲
NET STOP msiserver ↲
NET STOP usosvc ↲
NET STOP dosvc ↲
REN %WINDIR%¥SoftwareDistribution    SoftwareDistibution.old ↲
RD %WINDIR%¥SoftwareDistibution.old /S ↲
```

> Windows Update（wuauserv）サービスが自動開始されてしまうとSoftwareDistributionディレクトリのリネームに失敗するので、その場合は再度停止後にすばやくリネームしてください。もしくは、Windows Updateサービスのスタートアップを“無効”に変更してコンピューターを再起動し、SoftwareDistributionディレクトリをリネーム後に“手動”に戻します。

3.メンテナンスタスクは確実に完了させること

通常、再起動が必要な更新プログラムをインストールする場合、コンピューターが再起動されることでその処理は完了します。しかし、次の更新プログラムがインストールされる前に、実行されていることが推奨されるメンテナンスタスクが残っています。

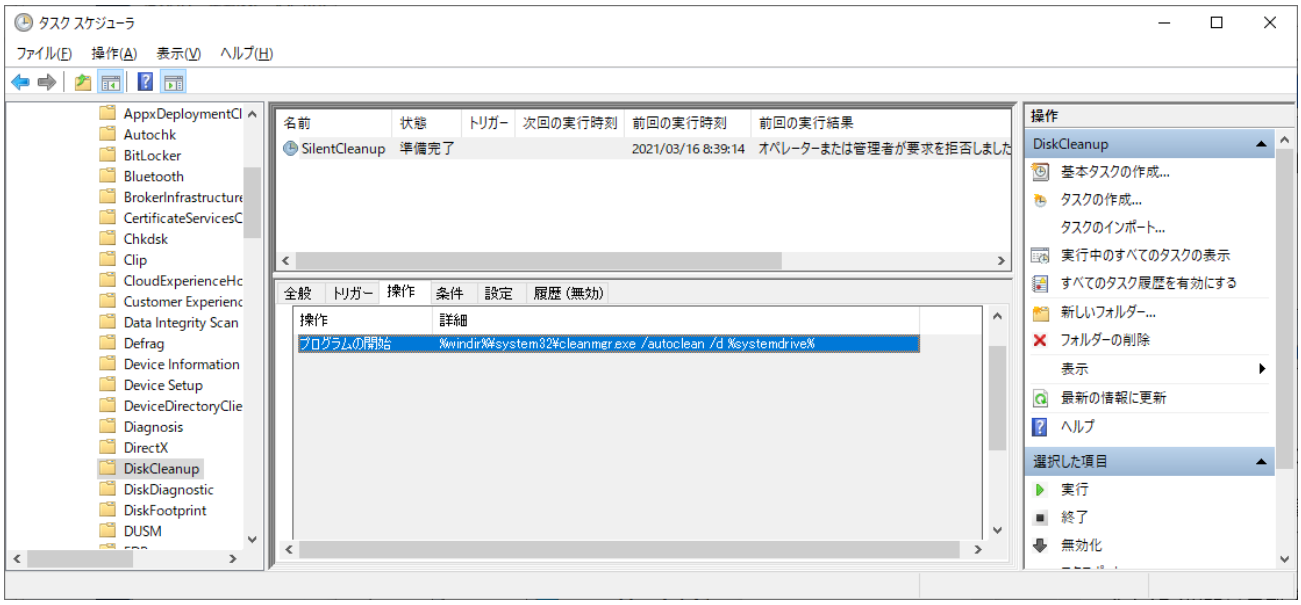
自動ディスククリーンアップタスク

Windows 8.1、Windows 10、デスクトップエクスペリエンス機能を追加したWindows Server 2012/2012 R2、およびWindows Server 2016以降のデスクトップエクスペリエンスでは、［タスクスケジューラ］（Taskschd.msc）の以下の場所にあるSilentCleanupタスクが定期的に実行されるようになっています（画面13）。

¥Microsoft¥Windows¥DiskCleanup¥SilentCleanup

SilentCleanupタスクは、コンピューターのアイドル時間にのみ実行され、アイドル状態でなくなると停止します。サーバーアプリケーションやインフラストラクチャサービスを実行するサーバーの場合、十分なアイドル時間が取れずにSlientCleanupタスクが完全に完了しない状態が続く可能性があります。

特に、更新プログラムのインストール後のSlientCleanupタスクは、Windows Updateの不要なコンポーネントの削除やコンポーネントストア（C:¥Windows¥WinSxS）の最適化（ファイルの圧縮など）が行われるため、完全に完了するためには余計に多くの時間がかかります。



画面13　SilentCleanupタスクは、Windows Update直後の実行には時間がかかるため、非アイドル状態になって途中で中止されることもある

更新プログラムのインストールのために業務時間外に集中して作業を行える場合は、SlientCleanupが実行するのと同じ以下のコマンドラインを、管理者として開いたコマンドプロンプトで実行します。コマンドプロンプトはすぐに帰ってきますが、cleanmgr.exeプロセスは実行中になります。［タスクマネージャー］（Taskmgr.msc）でcleamgr.exeプロセスが終了するまで待つとよいでしょう。

```
cleanmgr /autoclean /d C: ↲
```

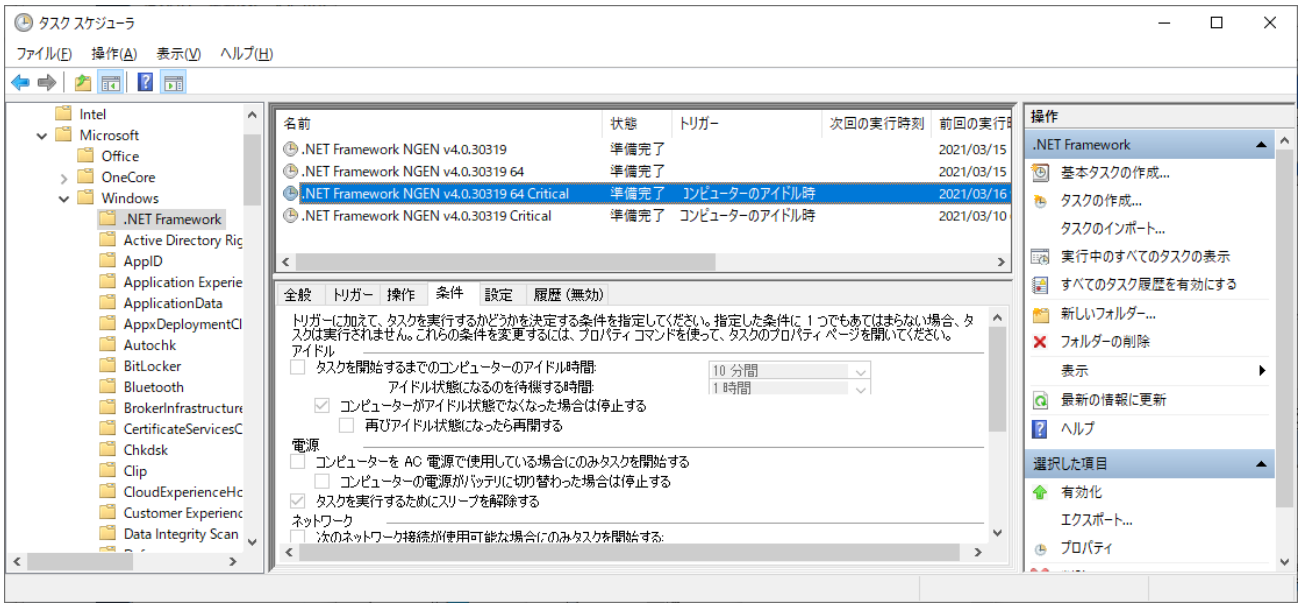
［ディスククリーンアップ］（cleanmgr.exe）を搭載しないServer Coreには、SlientCleanupタスクは存在しません。次のコマンドライン（/Resetbaseパラメーターは付けない）を実行することで、同様の効果を期待できます。

```
DISM /Online /Cleanup-Image  /StartComponentCleanup ↲
```

NGENタスク

.NET Frameworkの累積的な更新プログラム（注：Windows 10バージョン1803以前に標準搭載される.NET Frameworkバージョンの更新プログラムはWindowsの累積更新プログラムに含まれます）がインストールされると、以下の場所にある2つのタスク（32ビットOSの場合は2つ目のみ）が“準備完了”状態にセットされ、コンピューターのアイドル時間に自動実行されます（画面14）。

¥Microsoft¥Windows.NET Framework.NET Framework NGEN v4.0.30319 64 Critical
¥Microsoft¥Windows.NET Framework.NET Framework NGEN v4.0.30319 Critical



画面14　.NET Framework NGEN v4.0.30319 64 Critical（64ビット用）と.NET Framework NGEN v4.0.30319 Critical（32ビット用）は、.NET Frameworkの累積更新プログラムがインストールされると“準備完了”状態となり、実行が完了すると“無効”になる

これらのタスクは、.NET Frameworkのマネージドアプリケーションのパフォーマンスを向上するために、ネイティブイメージを生成して、ローカルコンピューターのネイティブイメージキャッシュに格納します。キャッシュが作成されることで、ランタイム時にJust-In-Time（JIT）コンパイラを使用してイメージをコンパイルする代わりに、キャッシュ内のイメージを使用できるようになります。

イメージキャッシュが存在しなければその時点でコンパイルされるため、これらのタスクが完了することは必須ではありません。しかし、これらのタスクの実行が次回のWindows Updateと重なると、システムの負荷が高まり、Windows Updateのインストール時間にも影響します。

更新プログラムのインストールのために業務時間外に集中して作業を行える場合は、タスクの状態を確認し、“準備完了”状態の場合は手動で集中的に実行するとよいでしょう。管理者として開いたコマンドプロンプトで次の4行のコマンドラインを実行します。なお、32ビットOSの場合は、3、4行目の実行は不要です。バッチファイルを作成しておくと（ngen.cmdなど）、コマンドラインを連続実行できて便利です。

```
C:¥Windows¥Microsoft.NET¥Framework¥v4.0.30319¥ngen.exe  executeQueuedItems ↲
C:¥Windows¥Microsoft.NET¥Framework¥v4.0.30319¥ngen.exe update ↲
C:¥Windows¥Microsoft.NET¥Framework64¥v4.0.30319¥ngen.exe  executeQueuedItems ↲
C:¥Windows¥Microsoft.NET¥Framework64¥v4.0.30319¥ngen.exe update ↲
```

サンプルスクリプト：PowerShellからプロセスの終了を監視する

Cleanmgrなど、バックグラウンドで実行されるプロセスの状態を監視するPowerShellスクリプトです。

checkproc.ps1 （使用法：.¥checkproc.ps1 <プロセス名（拡張子なし）>）

```
param($target)
while(1) {
$procid = (Get-Process -ProcessName $target -ErrorAction SilentlyContinue).Id
if ($procid.Length -eq 0) {
Write-Host $target "is not running currently"
break
} else {
Write-Host $target "(PID:"$procid") is alive"
}
Start-Sleep -seconds 1
}
```

checkproc.ps1の実行例

```
.¥checkproc.ps1  cleanmgr ↲
```

サンプルスクリプト：PowerShellからスケジュールされたタスクを実行する

以下は、指定したタスクが無効でなく、準備完了など実行中以外の状態であればすぐに実行を開始して、実行が終了するまで待つ汎用的なPowerShellスクリプトです。

runtask.ps1 （使用法：.¥runtask.ps1 "<タスク名>“ "<タスクパス> "）

```
param($taskname,$taskpath)
$taskfullname = $taskpath+$taskname
$taskstate = (get-scheduledtask -taskname $taskname -taskpath $taskpath -ErrorAction
SilentlyContinue).State
if ($taskstate -eq $null) {
Write-Host "Error : $taskname is not exist."
exit
}
if ($taskstate -eq "Disabled") {
Write-Host "$taskname is $taskstate : nothing to do."
} elseif ($taskstate -eq "Ready") {
Write-Host -NoNewLine "Start $taskname immediately... "
#Start-ScheduledTask -taskname $taskname -taskpath $taskpath
SCHTASKS /Run /I /TN $taskfullname
} elseif ($taskstate -eq "Running") {
Write-Host -NoNewLine "$taskname is already running."
} else {
Write-Host -NoNewLine "Try to start $taskname... "
SCHTASKS /Run /I /TN $taskfullname
Write-Host -NoNewLine "$taskname is already in progress."
}

while(1) {
$taskstate = (get-scheduledtask -taskname $taskname -taskpath $taskpath -ErrorAction
SilentlyContinue).State
if ($taskstate -eq "Ready") {
Write-Host "Task completed. (Ready)"
break
} elseif ($taskstate -eq "Disabled") {
Write-Host "Task completed. (Disabled)"
break
} else {
Write-Host -NoNewLine "."
}
Start-Sleep -seconds 1
}
```

NGENタスクの実行例

> .¥runtask.ps1 ".NET Framework NGEN v4.0.30319 Critical" "¥Microsoft¥Windows.NET Framework" ↲
>
> .¥runtask.ps1 ".NET Framework NGEN v4.0.30319 64 Critical" "¥Microsoft¥Windows.NET Framework" ↲

4.タスクスケジューラの使いこなし

Windows標準のタスク機能を利用すると、さまざまなタイミング、条件に基づいて、管理操作を自動化することができます。タスクは［タスクスケジューラ］（Taskschd.msc）、Schtasks.exeコマンド、およびPowerShellのScheduledTasksモジュールのコマンドレット（Get-Command -module ScheduledTasksで確認可能）を使用して作成および管理できます。ここでは［タスクスケジューラ］（Taskschd.msc）と一部コマンドラインを利用した活用例を紹介します。

週末に再起動をスケジューリングする

再起動保留状態の更新プログラムのインストールを完了させるため、サーバーで週末に再起動を実施するというシナリオを考えてみましょう。再起動は毎週必要なわけではなく、必要時にのみ簡単に再起動をスケジューリングすることにします。

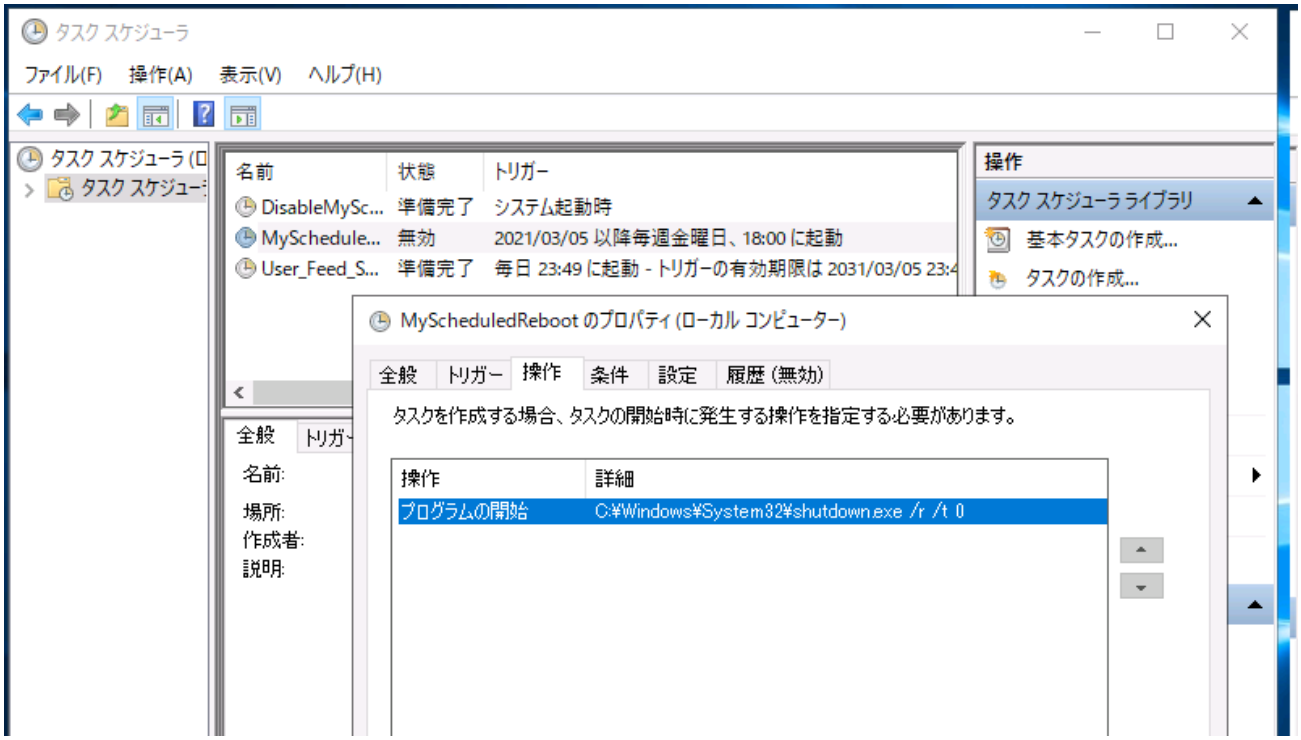
［タスクスケジューラ］（Taskschd.msc）を起動し、次の2つのタスクを作成します（画面15、画面16）。MyScheduledRebootタスクは毎週土曜日の23:59にshutdown.exe /r /t 0コマンドを実行して再起動するタスクです。もう1つのDisableMyScheduledRebootタスクは、システム起動時にMyScheduledRebootタスクの状態を“無効”にセットするタスクです。

全般	**名前**	**MyScheduledReboot**
	セキュリティオプション	✅ユーザーがログオンしているかどうかにかかわらず実行する ✅最上位の特権で実行する
トリガー	タスクの開始	スケジュールに従う、毎週、土曜日、23:59
操作	操作	プログラムの開始
	プログラム/スクリプト	%SystemRoot%¥System32¥shutdown.exe または C:¥Windows¥System32¥shutdown.exe
	引数の追加	/r /t 0

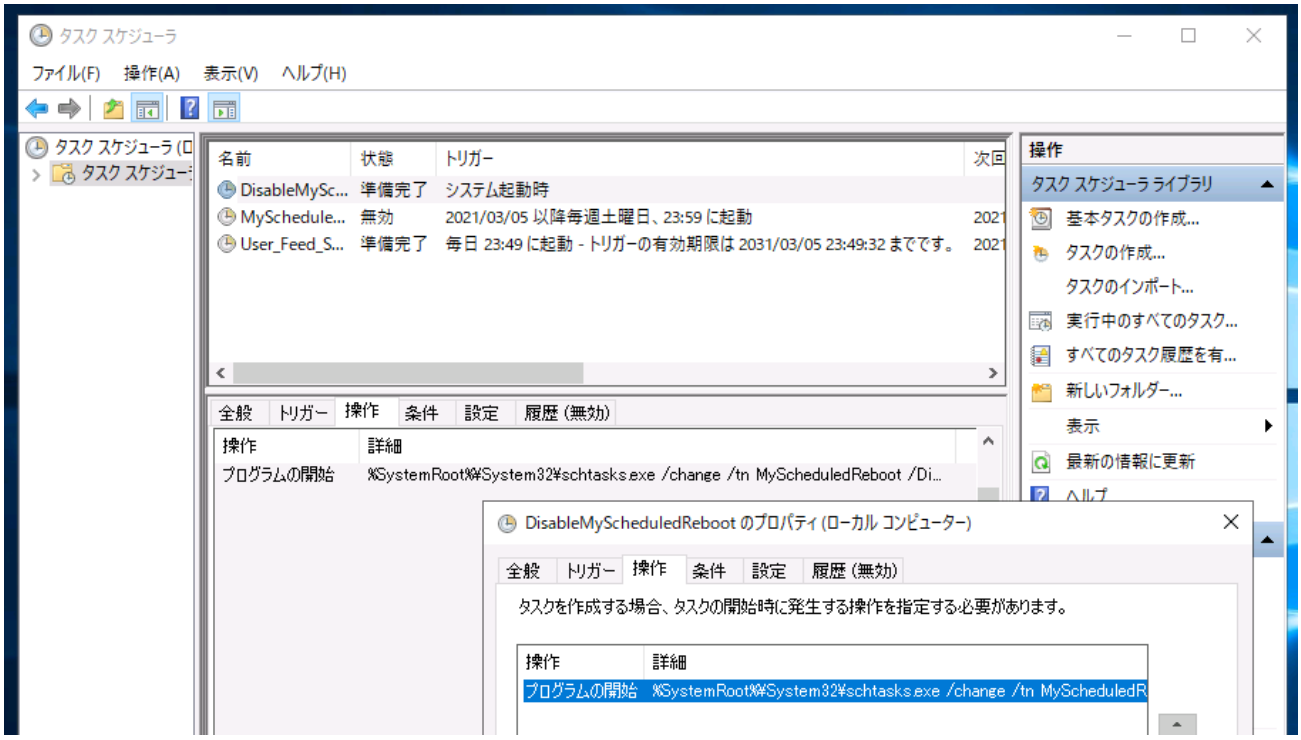
全般	**名前**	**DisableMyScheduledReboot**
	セキュリティオプション	✅ユーザーがログオンしているかどうかにかかわらず実行する ✅最上位の特権で実行する
トリガー	タスクの開始	スタートアップ時（システム起動時）
操作	操作	プログラムの開始

全般	名前	DisableMyScheduledReboot
	プログラム/スクリプト	%SystemRoot%¥System32¥schtasks.exe または C:¥Windows¥System32¥schtasks.exe
	引数の追加	/change /tn MyScheduledReboot /Disable

2つのタスクを作成したら、“準備完了”状態になっているMyScheduledTaskタスクを右クリックして［無効］を選択し、タスクを“無効”の状態に切り替えます。または、DisableMyScheduledRebootタスクを手動で一度実行します。



画面15　毎週末に再起動を実施するMyScheduledRebootタスクを作成し、“無効”状態にしておく



画面16　システム起動時にMyScheduledRebootタスクの状態を“無効”にセットするためのタスク

2つのタスクをこのように準備しておくことで、週末に再起動が必要な際に簡単かつすばやく再起動をスケジューリングできるようになります。PowerShellのEnable-ScheduledTaskコマンドレットまたはSchtasks.exeコマンドを次のように実行することで、MyScheduledRebootタスクが“準備完了”の状態にセットされます。すると、その週の週末に再起動が自動実行されます。そして、再起動後はDisableMyScheduledRebootタスクによって再びMyScheduledRebootタスクは“無効”の状態に戻ります。

```
Enable-ScheduledTask -TaskName MyScheduledReboot ↲
```

または

```
Schtasks.exe /change /tn  MyScheduledReboot /Enable ↲
```

PowerShellコマンドのスケジューリング

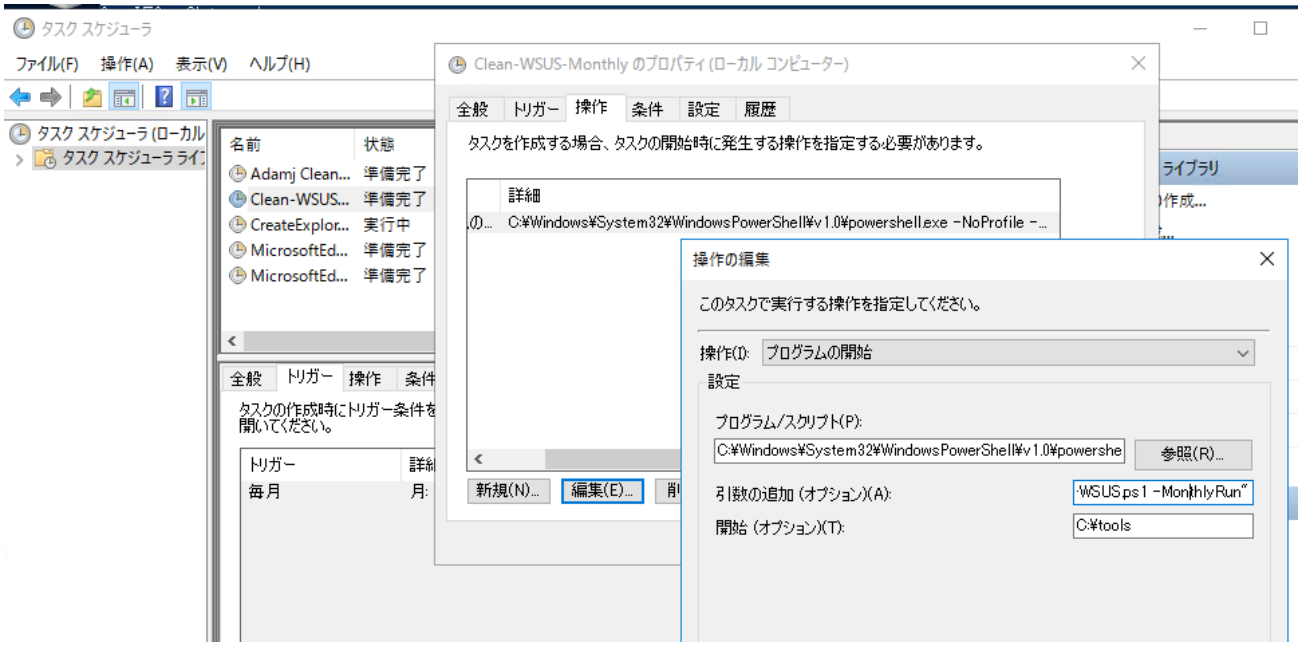
タスクの［プログラムの開始］操作にPowerShellコマンドレットのコマンドラインを指定したい場合は、次のように設定します。先ほどのDisableMyScheduledRebootタスクの［プログラムの開始］操作は、この方法で**<PowerShellのコマンドライン>**の部分に**Disable-ScheduledTask -TaskName MyScheduledReboot**と指定することで構成することもできます。

操作	操作	プログラムの開始
	プログラム/スクリプト	%SystemRoot%¥System32¥WindowsPowerShell¥v1.0¥powershell.exe または C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥powershell.exe
	引数の追加	-Command "<PowerShellのコマンドライン>" または -Command "& {<PowerShellのコマンドライン>}"
	開始	コマンドラインの実行を開始するカレントディレクトリ、省略すると%SystemRoot%¥System32

PowerShellスクリプト（.ps1）のスケジューリング

タスクの［プログラムの開始］操作にPowerShellのスクリプト（.ps1）をパラメーター付きで指定したい場合は、次のように設定します。スクリプトの実行ポリシーの現在の設定に関係なく、確実にスクリプトを実行させることが可能です（画面17）。スクリプトのパスには、［開始］に指定した場所からの相対パス（例： .¥script.ps1）で記述することもできます。

操作	操作	プログラムの開始
	プログラム/スクリプト	%SystemRoot%¥System32¥WindowsPowerShell¥v1.0¥powershell.exe または C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥powershell.exe
	引数の追加	-NoProfile -ExecutionPolicy unrestricted -Command "<.ps1ファイルのパス> <スクリプトに渡すパラメーター>" または -NoProfile -ExecutionPolicy unrestricted -Command "& {<.ps1ファイルのパス> <スクリプトに渡すパラメーター>}"
	開始	スクリプトの実行を開始するパス、省略すると%SystemRoot%¥System32



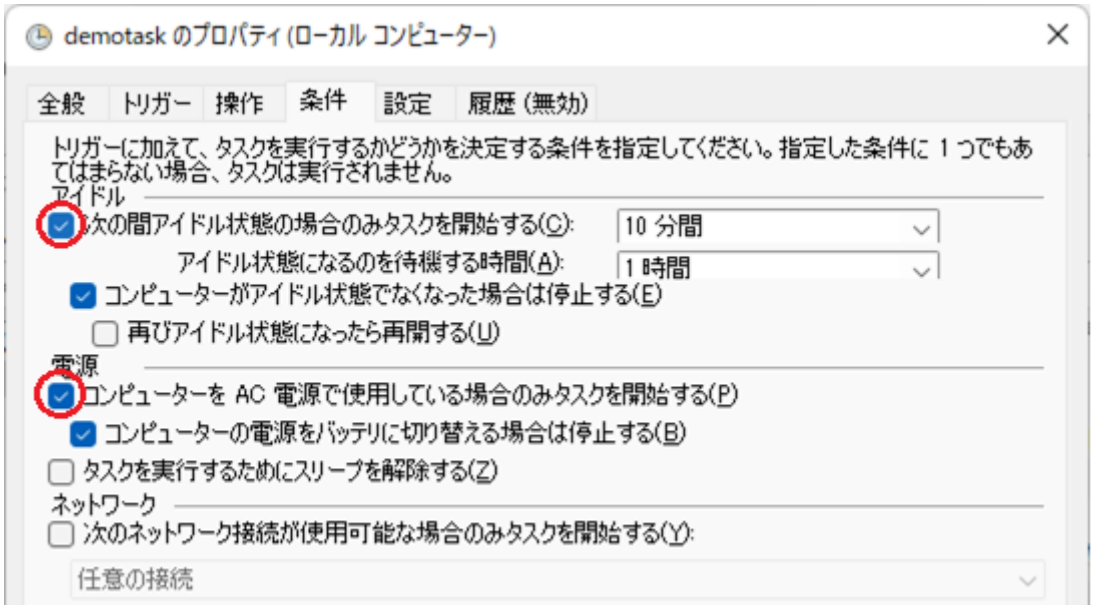
画面17　PowerShellスクリプト（.ps1）をタスクで実行させる場合の設定

タスクをコマンドラインから即時実行する

Start-ScheduledTaskまたはSchtasks.exe /RUNコマンドを使用すると、登録済みのタスク（Windowsに組み込みのタスクを含む）を即時に実行できます。

> Start-ScheduledTask -TaskName MyScheduledReboot ［-TaskPath “<タスクパス>”］↵
> または
> Schtasks.exe /RUN /I /TN MyScheduledReboot ↵

注：タスクのプロパティの［条件］タブで［次の間アイドル状態の場合のみタスクを開始する］がチェックされている場合、Start-ScheduledTaskはタスクをすぐに実行せずに、キューに挿入済み（Queued）にします。Schtasks.exe /RUNコマンドに/Iオプションを付けることでこの問題を回避できます。［コンピューターをAC電源で使用している場合のみタスクを開始する］がチェックされている場合も、バッテリ稼働時に同様の影響がありますが、/Iオプションで回避できます。


demotask のプロパティ (ローカル コンピューター)
全般
トリガー
操作
条件
設定
履歴 (無効)
トリガーに加えて、タスクを実行するかどうかを決定する条件を指定してください。指定した条件に 1 つでもあてはまらない場合、タスクは実行されません。
アイドル
次の間アイドル状態の場合のみタスクを開始する(C):
10 分間
アイドル状態になるのを待機する時間(A):
1 時間
コンピューターがアイドル状態でなくなった場合は停止する(E)
再びアイドル状態になったら再開する(U)
電源
コンピューターを AC 電源で使用している場合のみタスクを開始する(P)
コンピューターの電源をバッテリに切り替える場合は停止する(B)
タスクを実行するためにスリープを解除する(Z)
ネットワーク
次のネットワーク接続が使用可能な場合のみタスクを開始する(Y):
任意の接続

5.現在のセキュリティ設定を推奨基準と比較する

WindowsおよびWindows Serverのシステム設定は、パフォーマンスやネットワークスループットを最適化するため、セキュリティを強化するため、効率化するためなど、さまざまにカスタマイズすることができます。しかし、不適切なシステム設定や、意図せず設定が重複してしまい上書きされてしまったりすることがあります。特にセキュリティに関連する設定が不適切であると、たとえOSやアプリが最新状態に更新されていたとしても、自ら作り出してしまった脆弱性を突かれセキュリティ侵害されてしまうリスクもあります。ここでは、マイクロソフトが無料提供するMicrosoft Security Compliance Toolkit（SCT）1.0を利用したセキュリティ設定の評価と実装について紹介します。

Microsoft Security Compliance Toolkit（SCT）の入手と準備

Microsoft Security Compliance Toolkit（SCT）1.0は、企業のセキュリティ管理者がWindows、Windows Server、およびその他のマイクロソフト製品用の、マイクロソフト推奨のセキュリティ設定のベースラインをダウンロードし、現在の設定との比較評価、および実装するのに役立つ一連のツールです。

SCTを使用すると、Active Directoryドメインで展開している現在のグループポリシーとマイクロソフト推奨のベースラインを効率的に比較したり、マイクロソフト推奨のベースラインからグループポリシーオブジェクト（GPO）を作成して展開したりできます。また、ローカルコンピューターの現在のシステム設定とベースラインを比較したり、現在のシステム設定にベースラインの設定を上書きしたりすることもできます。

Microsoft Security Compliance Toolkit 1.0のダウンロード
https://www.microsoft.com/en-us/download/details.aspx?id=55319

Microsoft Security Compliance Toolkit 1.0のドキュメント
https://docs.microsoft.com/ja-jp/windows/security/threat-protection/windows-security-configuration-framework/security-compliance-toolkit-10

SCTのダウンロードサイトからは以下の表3のツールをダウンロードすることができます。SCTのバージョンは1.0のまま変わらなくても、ダウンロードコンテンツは不定期に更新されます。必ず最新版を入手して使用するようにしてください。なぜなら、マイクロソフト推奨のセキュリティ設定は変更される場合があります。以前の推奨設定が、現在では推奨されない設定になっていることもあるからです。

表3　SCT 1.0として提供されているツール群（2021年3月16日公開版）

ダウンロードファイル名	説明
PolicyAnalyzer.zip	Policy Analyzer v4.0（PolicyAnalyzer.exe）
LGPO.zip	LGPO v3.0（LGPO.exe）
SetObjectSecurity.zip	SetObjectSecurity v1.0 （SetObjectSecurity.exe）

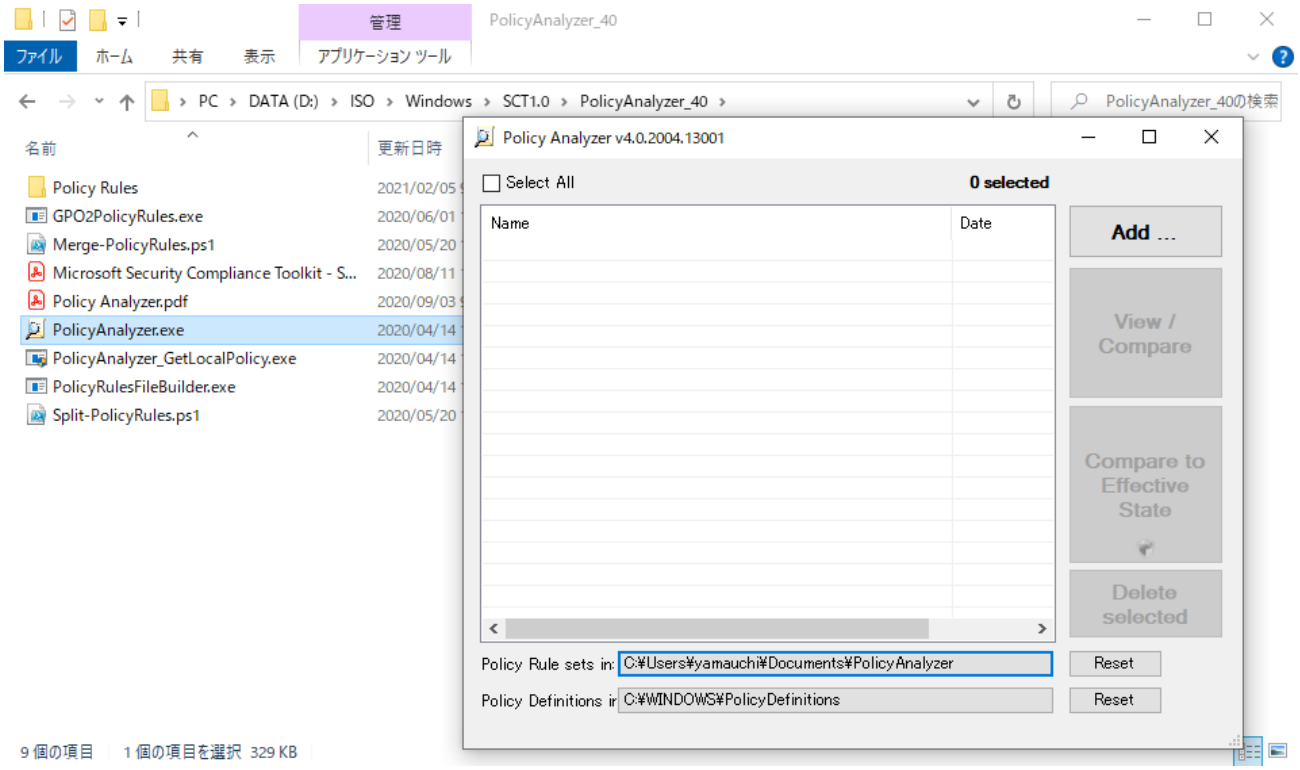
ダウンロードファイル名	説明
Windows Server 2012 R2 Security Baseline.zip	WindowsおよびWindows Serverの各バージョンに対応したセキュリティベースライン
Windows 10 Version 1507 Security Baseline.zip	
Windows 10 Version 1607 and Windows Server 2016 Security Baseline.zip	
Windows 10 Version 1803 Security Baseline.zip	
Windows 10 Version 1809 and Windows Server 2019 Security Baseline.zip	
Windows 10 Version 1903 and Windows Server Version 1903 Security Baseline - Sept2019Update.zip	
Windows 10 Version 1909 and Windows Server Version 1909 Security Baseline.zip	
Windows 10 Version 2004 and Windows Server Version 2004 Security Baseline.zip	
Windows 10 Version 20H2 and Windows Server Version 20H2 Security Baseline.zip	
Windows 10 Update Baseline.zip	Windows 10のWindows Updateのベースライン
Microsoft Edge v88 Security Baseline.zip	Chromium版Microsoft Edge Stableバージョン88のセキュリティベースライン
Office365-ProPlus-Sept2019-FINAL.zip	Microsoft 365 Apps（旧称、Office 365 ProPlus）のセキュリティベースライン

各ツールの機能については以下の公式ブログを参照してください。

New & Update Security Tools
https://techcommunity.microsoft.com/t5/microsoft-security-baselines/new-amp-updated-security-tools/ba-p/1631613

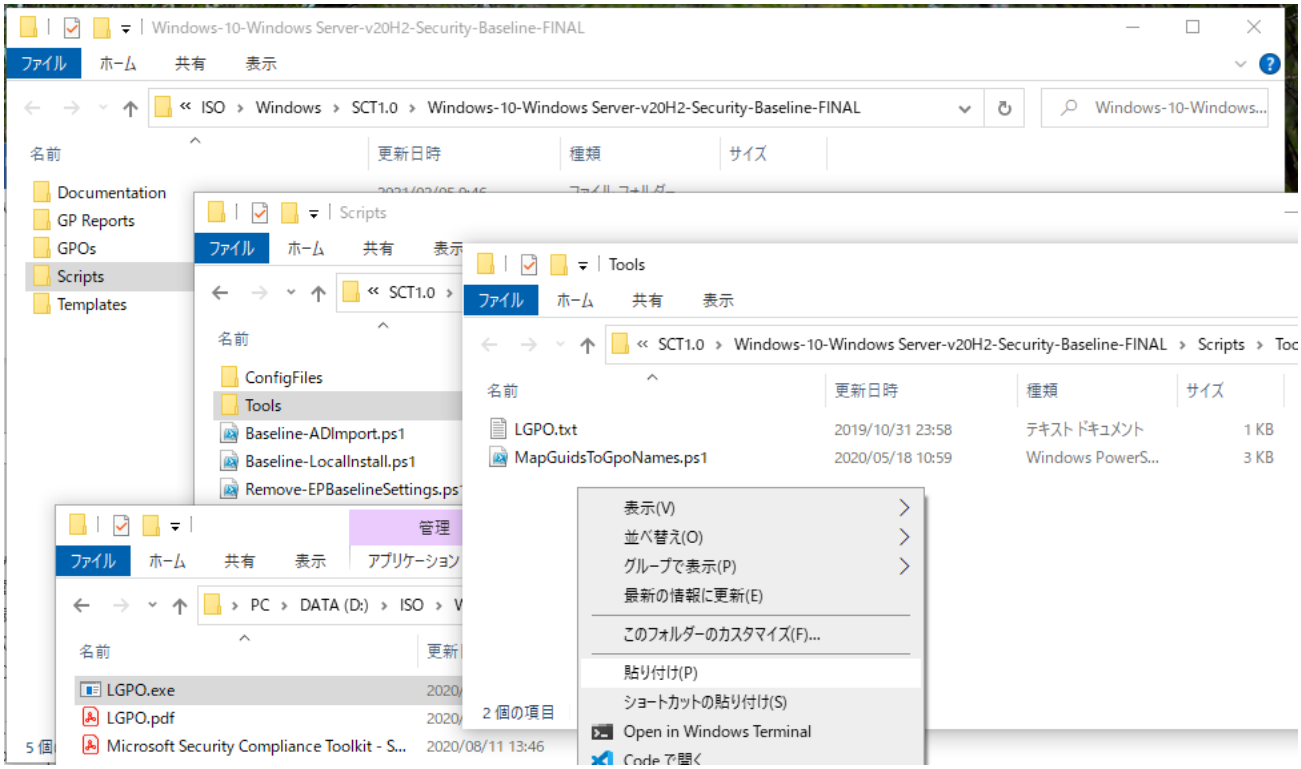
※ 現在は、Windows 11、最新の Microsoft 365アプリ、Windows Server 2022、最近の Microsoft Edge 用のテンプレートを入手できます。

セキュリティベースラインとの比較を行う場合は、Policy Analyzer.zipと、評価したいWindowsやWindows Server、その他のマイクロソフト製品のベースラインの.zipファイルをダウンロードして任意の場所に展開します。Policy Analyzer（PolicyAnalyzer.exe）は、1つ以上のセキュリティベースラインとGPO、またはローカルコンピューターの現在の設定を比較するための分析ツールです（画面18）。



画面18　SCT 1.0の分析ツールであるPolicy Analyzer（PolicyAnalyzer.exe）

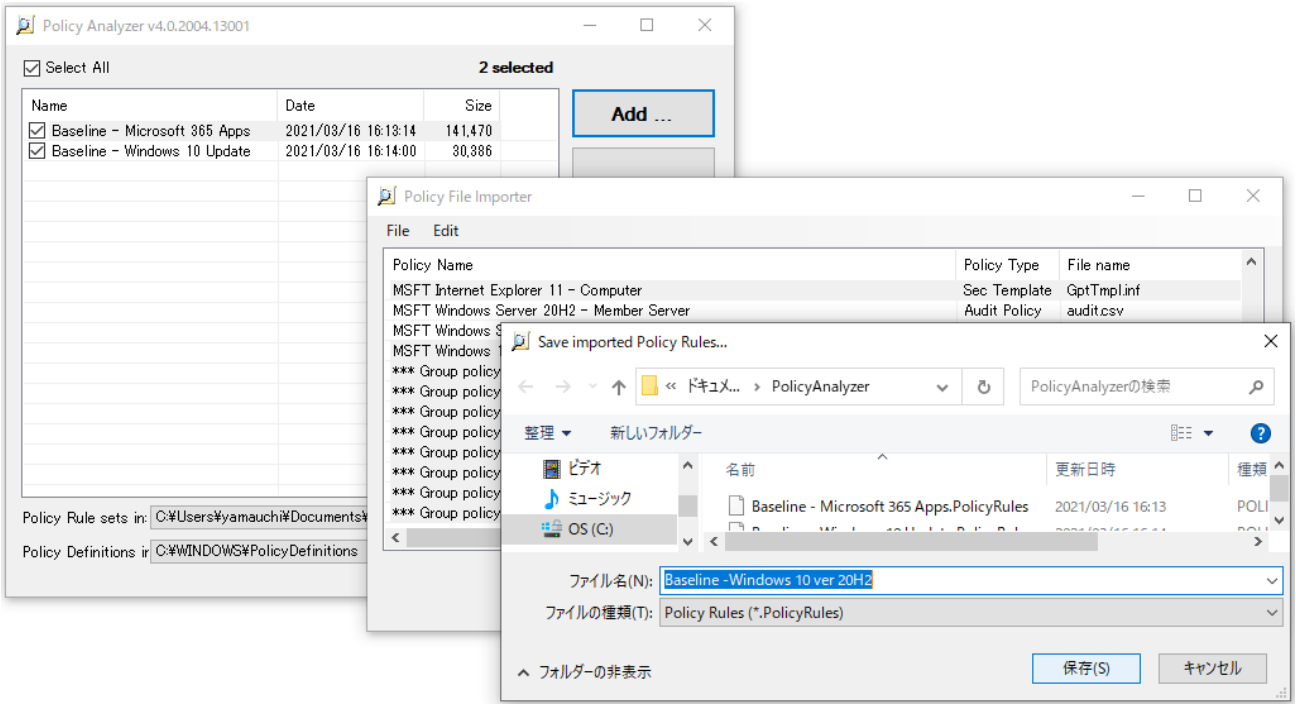
推奨のベースラインからGPOを作成したり、ローカルコンピューターを設定する場合はLGPO.zipをダウンロードして展開し、LGPO.exeを各セキュリティベースラインのスクリプト用ディレクトリのToolsに格納しておきます（Scripts¥ToolsまたはLocal_Scripts¥Toolsディレクトリが存在する場合、画面19）。GPOの作成や設定のインストールには、スクリプト用ディレクトリにあるスクリプト（.ps1）またはバッチ（.cmd）を使用します。ベースラインの評価目的だけであれば、この準備作業は不要です。



画面19　ベースラインのScripts¥ToolsディレクトリにLGPO.exeをコピーしておく

SCTによる現在のローカル設定の評価

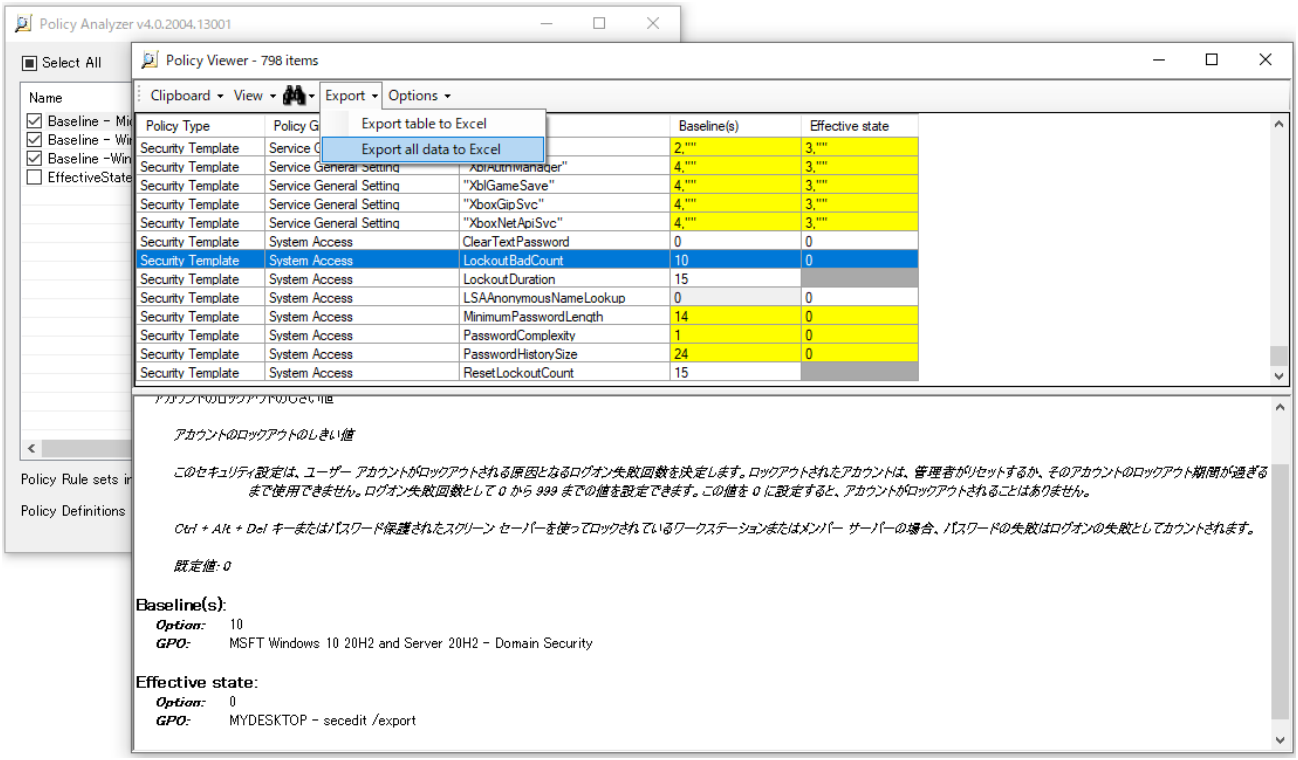
SCT を使用してローカルコンピューターのシステム設定をセキュリティベースラインと比較します。Policy Analyzer（PolicyAnalyzer.exe）を開き、［Add］ボタンをクリックし、さらに［File］から［Add Files from GPO(s)］を選択して、調査したいセキュリティベースラインの展開先ディレクトリにあるGPOsサブディレクトリを選択してポリシーをインポートします。インポートする際、［Save imported policy Rules］の場所と名前を要求されるので、ベースラインであることが分かるように分かりやすい名前を入力してインポートします（画面20）。



画面20　ベースラインのGPOsサブディレクトリからポリシー設定をインポートする

1つ以上のベースラインをインポートしたら、調査したいベースラインを選択した状態で［Compare to Effective State］ボタンをクリックします。すると、［Policy Viewer］が開き、ローカルコンピューターのレジストリ設定とベースラインのポリシー設定が比較され、違いをハイライト表示されます。

［Policy Viewer］で1つの行を選択すると、設定の意味やどのベースラインに設定されている推奨設定なのか、および現在ローカルコンピューターで有効になっている設定を確認できます（画面21）。［Export］メニューを使用すると、表形式のレポート、またはすべてのデータをExcelワークシートにエクスポートすることができます。



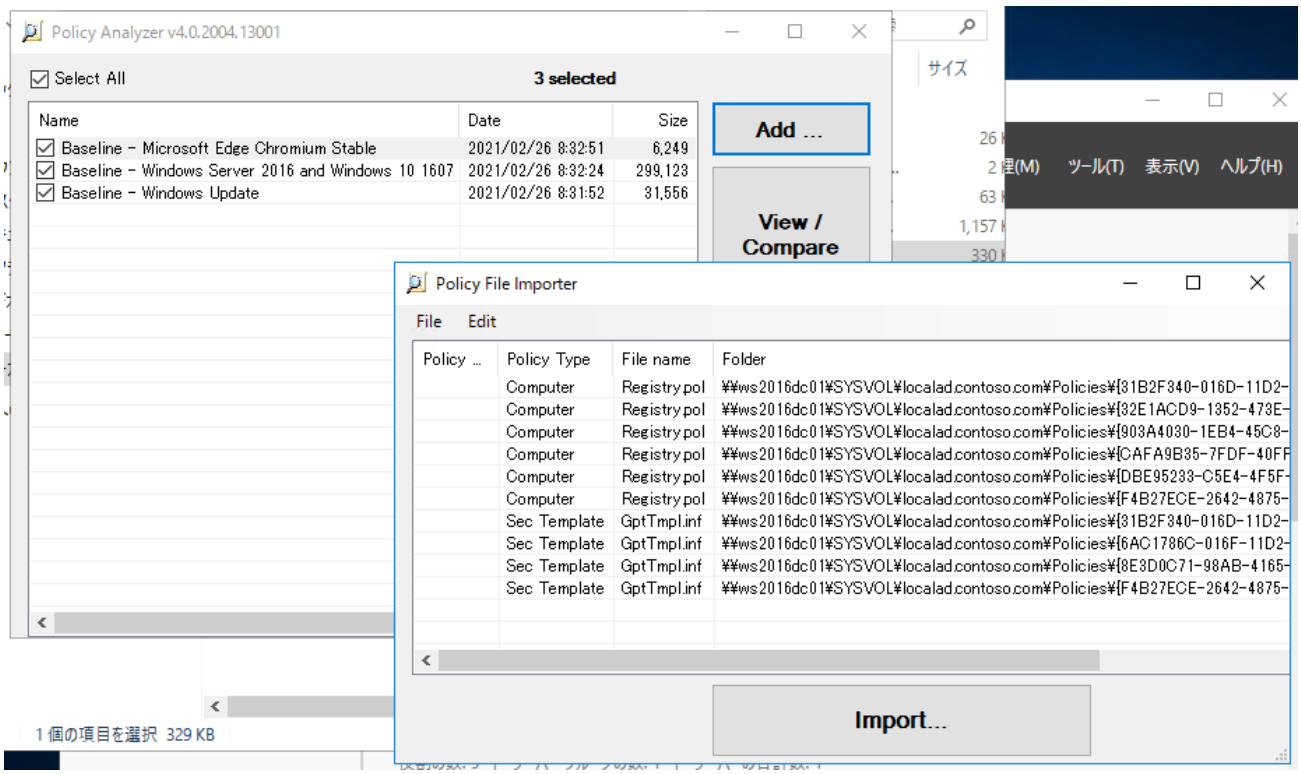
画面21　［Policy Viewer］で現在のシステム設定と推奨のベースライン設定を比較する

SCTによるグループポリシー設定の評価

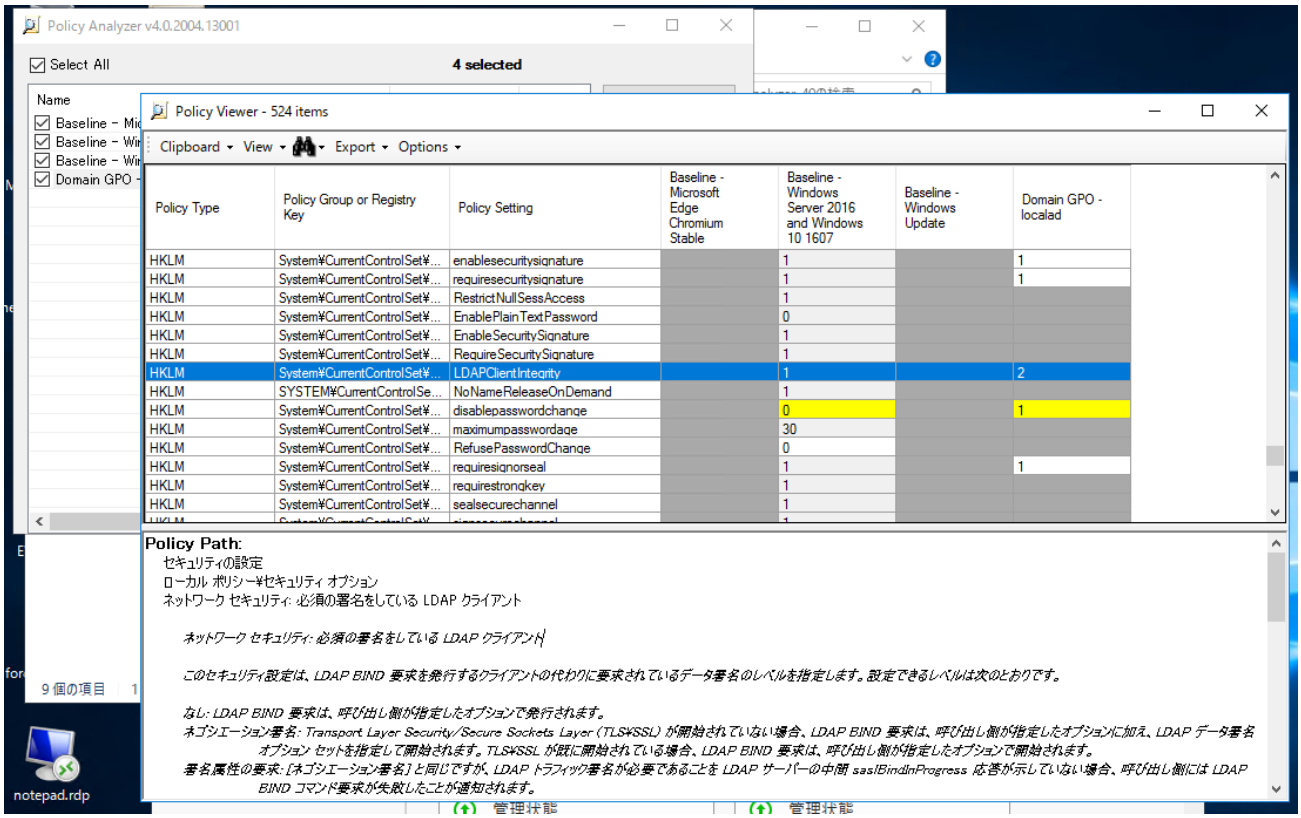
SCTを使用してActive Directoryドメインの現在のグループポリシー設定を調査したい場合は、ドメインコントローラーまたはActive Directoryの管理ツールを利用できるメンバーコンピューターでPolicy Analyzer（PolicyAnalyzer.exe）を実行します。

［Add］ボタンをクリックし、ベースラインをインポートしたら、さらに続けてドメインのSYSVOL共有パス内のGPO（¥<ドメインコントローラー名>¥SYSVOL<DNSドメイン名>¥Policies）からポリシー設定をインポートします（画面22）。SYSVOL共有のGPOをインポートする代わりに、［グループポリシーの管理］スナップインを使用してバックアップしたGPOを、バックアップ先（またはそのコピー）からインポートすることもできます。

調査したいベースラインとドメインのGPOからのインポートしたポリシー設定のすべてを選択し、［View/Compare］ボタンをクリックします。すると、［Policy Viewer］が開き、現在のドメインで有効なGPOとベースラインとの違いをハイライト表示で調査することができます（画面23）。ローカル設定の評価の場合と同様に、［Export］メニューを使用すると、表形式のレポート、またはすべてのデータをExcelワークシートにエクスポートすることができます。



画面22　ドメインのSYSVOL共有からGPOをインポートする



画面23　ドメインで有効なGPOとベースラインのポリシー設定を比較する

ベースラインからのGPOの作成

各セキュリティベースラインはActive DirectoryドメインのGPOとして簡単にインポートできます。例えば、Windows 10 Update Baselineの場合はScriptsディレクトリにあるbaseline-ADImport.ps1をドメインコントローラー上のPowerShellウィンドウで実行するだけです。

> .¥baseline-ADImport.ps1 ↲

インポートされたGPOは、［グループポリシーの管理］の［グループ ポリシー オブジェクト］の下で確認できます。GPOをドメインのコンテナーや組織単位（OU）に関連付ければ、推奨セキュリティ設定をドメイン内に展開できます。

なお、スクリプトのファイル名や使用方法はベースラインにより異なる場合があります。パラメーターの指定が必要なものもありますし、バッチファイル（例：Domain_Controller_Install.cmd）として提供されている場合もあります。スクリプト用ディレクトリ（ScriptsやLocal_Scripts）内にあるスクリプトやバッチの内容を確認して適切に使用してください。

ベースラインのローカルインストール（非推奨）

推奨セキュリティ設定のベースラインをローカルコンピューターのローカル設定（レジストリ設定）として直接的に上書きするローカルインストール用スクリプトやバッチ、パラメーターが利用可能な場合もあります（例：Baseline_LocalInstall.ps1、Client_Install.cmd）。しかし、ベースラインのローカルインストールはお勧めしません。アンインストール用のスクリプトやバッチが用意されて場合もありますが（例：Remove-EPBaselineSettings.ps1）、設定を完全に元の状態に戻すことができないからです。

もしローカルインストール機能を利用したいのであれば、簡単にロールバックできる仮想マシン環境などで十分にテストした上で実施してください。

6.Windowsファイアウォールの健全性チェック

WindowsおよびWindows Serverでは、エンドポイントのファイアウォール機能として［Windows Defenderファイアウォール］（旧称、Windowsファイアウォール）が既定で有効になっており、送受信トラフィックの許可／禁止を制御しています。Windows Defenderファイアウォールは不要な、あるいは許可されていないネットワークトラフィックをブロックすることで攻撃面を縮小することが主な目的です。Windows Defenderファイアウォールが適切に構成されていないと、セキュリティを低下につながるばかりか、パフォーマンスに影響することもあります。

ネットワークの場所とファイアウォールプロファイルの関係

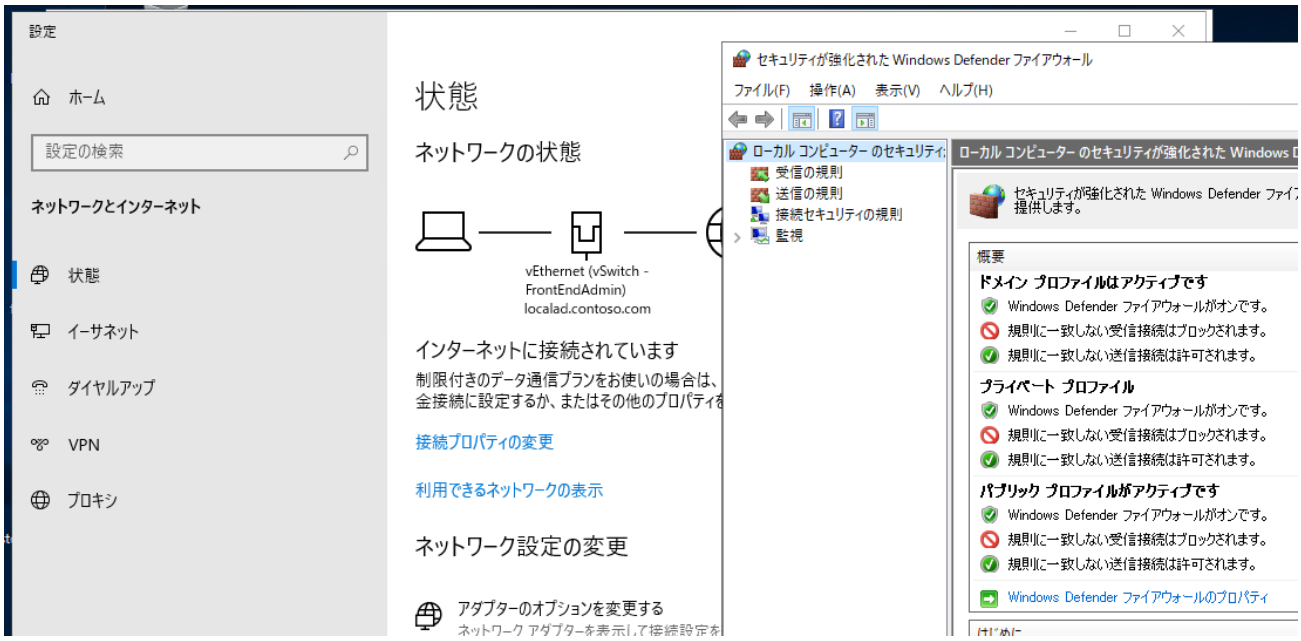
コンピューターにインストールされている物理ネットワークアダプターには、アダプターごとに**ネットワークプロファイル（ネットワーク接続プロファイル）**が関連付けられます。

Windowsは新しいネットワーク接続を検出すると、それを識別し、ネットワークの場所（Network Category）を**パブリックネットワーク（Public）**または**プライベートネットワーク（Private）**に設定します。既定ではパブリックに設定しようとしますが、ユーザーに対してプライベートネットワークに切り替えるか問い合わせますし、後から［設定］アプリの［ネットワークとインターネット］のGUIで変更することもできます。プライベートに設定されたネットワークがActive Directoryドメインとして認証されると（コンピューターがドメインメンバーである場合）、ネットワークの場所を**ドメインネットワーク（DomainAuthenticated）**に設定します。デフォルトゲートウェイが存在せず、インターネット接続を確認できない接続については、**識別されていないネットワーク**として**パブリックネットワーク（Public）**が設定されます。

一方、Windows Defenderファイアウォールには、**パブリック（Public）**、**プライベート（Private）**、**ドメイン（Domain）**の3つの**ファイアウォールプロファイル**が存在します。**パブリック（Public）**は信頼できないネットワーク接続に対して最小限の通信を許可するファイアウォールルールが、**プライベート（Private）**にはLAN（ローカルエリアネットワーク）上でさまざまなネットワーク機能（ファイルとプリンター共有など）を可能にする緩和されたファイアウォールルールが、**ドメイン（Domain）**にはActive Directoryドメインのネットワーク機能に不可欠なファイアウォールルールが定義されています。

そして、**ネットワークプロファイル**の**パブリックネットワーク（Public）**、**プライベートネットワーク（Private）**、**ドメインネットワーク（DomainAUthenticated）**の設定には、**パブリック（Public）**、**プライベート（Private）**、**ドメイン（Domain）**の**ファイアウォールプロファイル**が適用され、それぞれにファイアウォールのルールが適用されることになります。

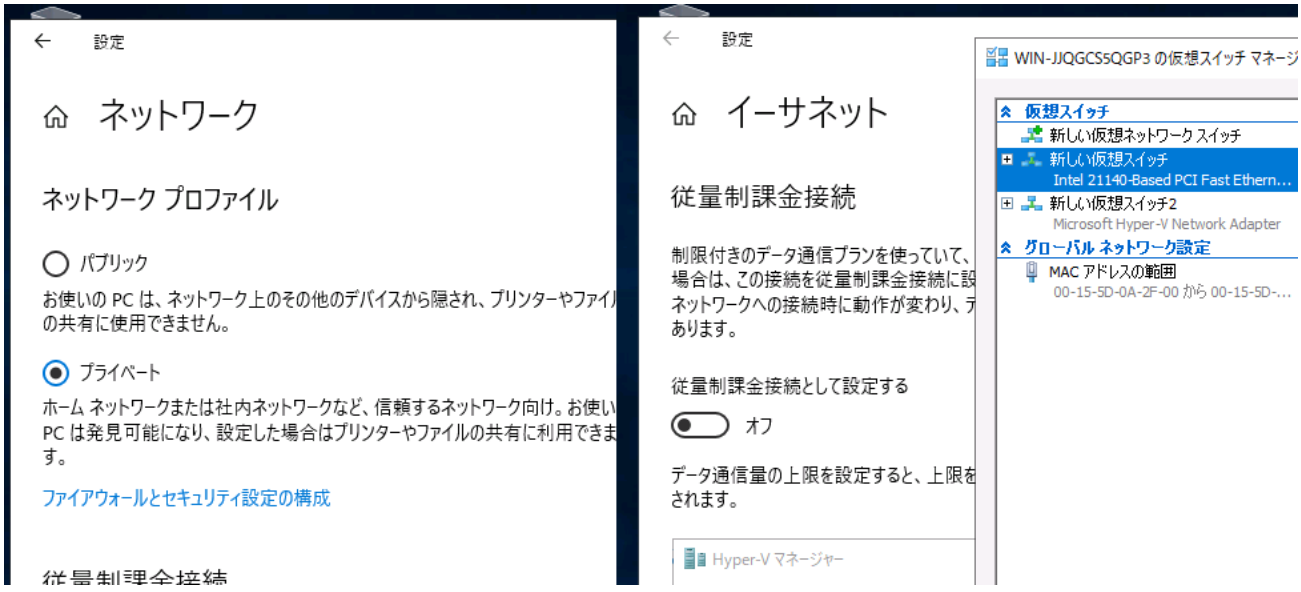
Windows 10およびWindows Server 2016以降のデスクトップエクスペリエンスの場合、現在のネットワークの識別状態は、［設定］アプリの［ネットワークとインターネット］、およびコントロールパネルの［ネットワークと共有センター］で確認することができます。また、Windows Defenderファイアウォールの状態は［セキュリティが強化されたWindows Defenderファイアウォール］（WF.msc）スナップインで確認することができます。ネットワークを安全な状態を維持するためには、最低限、ネットワークの場所が適切に識別されており、対応するファイアウォールプロファイルが有効になっていることが重要です。



画面1　ネットワークの識別と対応するファイアウォールプロファイルが有効になっていることを確認する

コマンドラインによるプロファイルの確認と変更

物理ネットワークアダプターが共有タイプのHyper-V仮想スイッチとして使用されている場合や、ドメインネットワークとして識別されている場合など、Windows標準のGUIでは**ネットワークプロファイル**の**パブリック（Public）**と**プライベート（Private）**を切り替えする場所がなくなります（画面2、注：**ドメインネットワーク（DomainAUthenticated）**として識別されているのであればネットワークの識別は適切です）。Windows ServerのServer Coreはもともと GUIを提供しません。*1

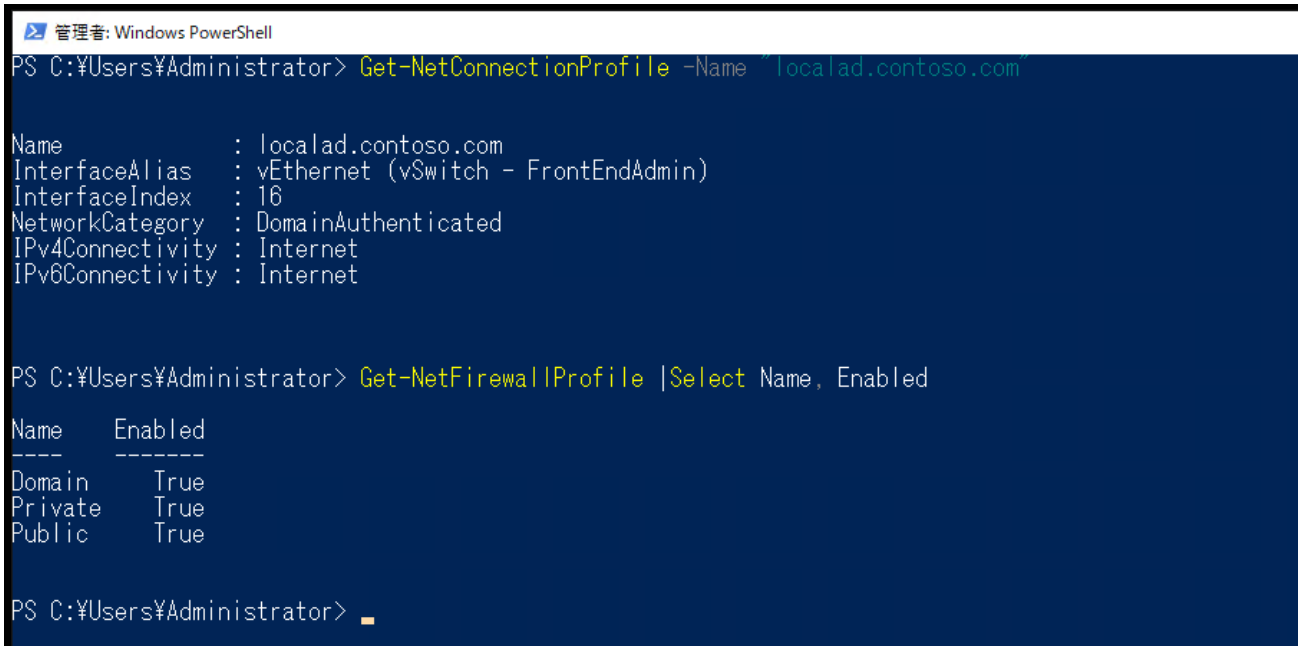


画面2　同じコンピューターでHyper-V仮想スイッチを作成する前（画面左）と作成した後（画面右）

*1　Windows Server 2019/2018以降はServer Coreアプリ互換性FoDをインストールすることで、［セキュリティが強化されたWindows Defenderファイアウォール］（WF.msc）スナップインを含む一部の管理ツールを利用可能ですが、ネットワークプロファイルの確認や切り替えのためのGUIはありません。

現在の**ネットワークプロファイル**における**ネットワークの場所（Network Category）**の識別と、**ファイアウォールプロファイル**の状態をコマンドラインで確認には、PowerShellのGet-NetConnectionProfileおよびGet-NetFirwewallProfileコマンドレット（↲ は Enter キーまたは改行位置、以下同じ*2）を利用できます（画面3）。

> Get-NetConnectionProfile ↲
>
> Get-NetFirewallProfile ↲



画面3　Get-NetConnectionProfileとGet-NetFirewallProfileで現在の状態を確認する

ネットワークの場所（NeworkCategory）が適切でない場合は、Set-NetConnectionProfileコマンドレットを次のように実行することで適切なものに切り替えることが可能です（画面4）。-Nameパラメーターの他、-InterfaceAliasや-InterfaceIndexパラメーターでネットワーク接続を指定することもできます。

> Set-NetConnectionProfile -Network "<ネットワーク名>" -NetworkCategory PublicまたはPrivateまたはDomainAuthenticated ↲

画面4　Set-NetConnectionProfileでネットワークの場所をパブリックからプライベートに切り替える

ファイアウォールのルールが肥大化していないか

Windows Defenderファイアウォールには既定のファイアウォールルールが定義済みとなっており、既定で有効になっているもの、Windows Serverのサーバーの役割や機能の有効化、WindowsやWindows Serverのリモート管理の設定やリモートデスクトップ接続の設定によって自動で有効化されるもの、サーバーアプリケーションやデスクトップアプリケーション、ストアアプリ（ユニバーサルWindowsプラットフォームアプリ、UWPアプリ）のインストールや使用によって自動またはユーザーの許可により追加されるものがあります。

Windows Server 2019のインストール直後の状態で確認すると、デスクトップエクスペリエンスで104、Server Coreで55のファイアウォールルールが有効でした。Windows 10 Enterpriseバージョン20H2のインストール直後の状態で確認すると、202のファイアウォールルールが有効でした。

ファイアウォールルールはレジストリに格納されるため、ルールが肥大化すると、レジストリハイブも肥大化します。大きすぎるレジストリハイブは、システムの応答停止やエラー、パフォーマンスの低下など、さまざまな問題を引き起こす可能性があります。*3

*3　レジストリ64ビット版Windows XP SP2およびWindows Server 2003 SP2からレジストリハイブの最大サイズは2GBになりました。しかし、2GBを超えた場合に再起動できなくなる問題があり、Windows 8およびWindows Server 2012以降については更新プログラムによって最大4GBまで拡張されました（Computer cannot be restarted if the registry hives are larger than 2 GB https://support.microsoft.com/help/2978366）。Windows 10およびWindows Server 2016以降については最初から4GB（4095MB）です。現在のロードされているレジストリハイブのサイズと最大サイズは、PowerShellでGet-WmiObject Win32_registryを実行して確認できます。32ビット版Windowsにはより小さな上限が設定されます。

ルールの数に特に上限はなく、明確な基準を示すことはできませんが、異常な数のルール、あるいは同じ名称や似た名称のルールが大量に存在する場合は、早めに対処したほうがよいでしょう。ただし、Windowsに既定で用意されている定義済みのルールについては削除しないでください。

PowerShellで以下のコマンドラインを実行すると、現在、有効になっているファイアウォールルールの数を確認することができます。

```
(Get-NetFirewallRule | where {$_.Enabled -eq $true}).Count ↲
```

同じ名称のルールが重複して登録されている状況は、PowerShellで次のコマンドラインを実行することで、ルールの表示名でグループ化し、降順にソートして調査することができます。

```
Get-NetFirewallRule | Group-Object -Property DisplayName | Sort-Object -Property Count -Descending ↲
```

次に実際のルールの肥大化の例と対処方法を示します。肥大化したルールを安易に削除することはせず、ルールの詳細な設定を確認し、削除による影響をよく検討してください。また、できるだけ現在のルールをバックアップしてから対処しましょう。次のコマンドラインを実行すると、現在のルール設定をファイル（.wfw）にエクスポートしてバックアップ、およびインポートして復元することができます。［セキュリティが強化されたWindows Defenderファイアウォール］（WF.msc）スナップインでの［ポリシーのエクスポート］と［ポリシーのエクスポート］の操作と同じことをコマンドラインからすばやく実行する方法です。

```
Netsh advfirewall export <エクスポート先パス>¥ファイル名.wfw" ↲
```

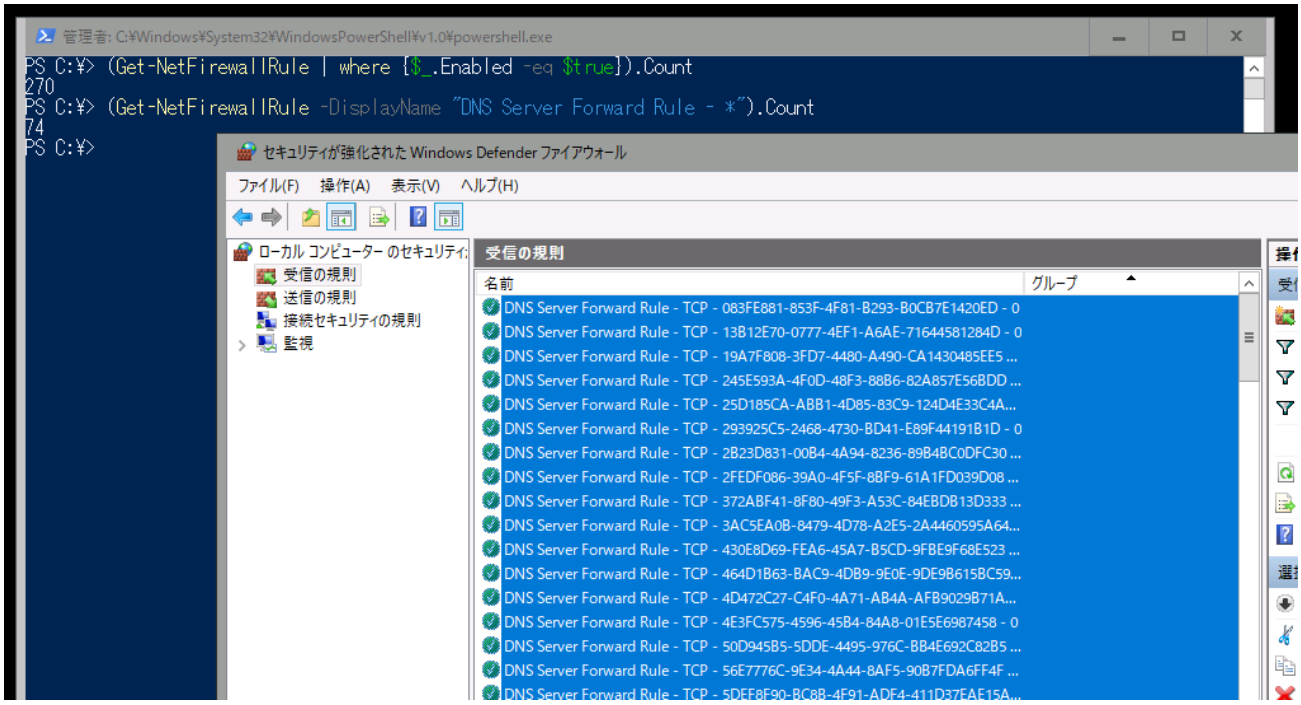
> Netsh advfirewall import <エクスポート先パス>¥ファイル名.wfw" ↲

・事例1

Windows 10バージョン1809およびWindows Server 2019のHyper-Vホストやコンテナーホストでは、受信の規則に次のルールのペアが大量に登録されることを確認しています（画面5）。ルールの名前は完全に同一ではなく、GUID（XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX）の部分はペアごとに異なります。

DNS Server Forward Rule - TCP - GUID - 0

DNS Server Forward Rule - UDP - GUID - 0



画面5　Windows 10バージョン1809やWindows Server 2019のHyper-Vホストに大量に登録されるDNS Server Forward Rule。削除しても次回起動時に再登録されるので問題ない

この問題は、既定で作成されるDefault Switchという名前のHyper-V仮想スイッチが、ホストの起動のたびにリセットされることが原因と思われます。ルールの数が多くなっている場合は、以下のコマンドラインを実行して既に作成されているルールを削除し、コンピューターを再起動すると、新たなペアが自動的に登録されます。

> Remove-NetFirewallRule -Displayname "DNS Server Forward Rule -*" ↲
>
> Restart-Computer ↲

なお、重複して登録される問題が解決されるわけではないので、定期的にルールをクリーンアップするとよいでしょう。Windows 10およびWindows Serverのより新しいバージョンでは、次のルール名が次のペアに変更され、極端に重複登録されることはなくなったようです。Windows 10の機能更新プログラムによるアップグレードを繰り返していると、過去に自動登録され、現在は利用されていないルールとして旧名称のルールが残っていることがあるかもしれません。

HNS Container Networking - ICS DNS (TCP-In) - GUID - 0

HNS Container Networking - DNS (UDP-In) - GUID - 0

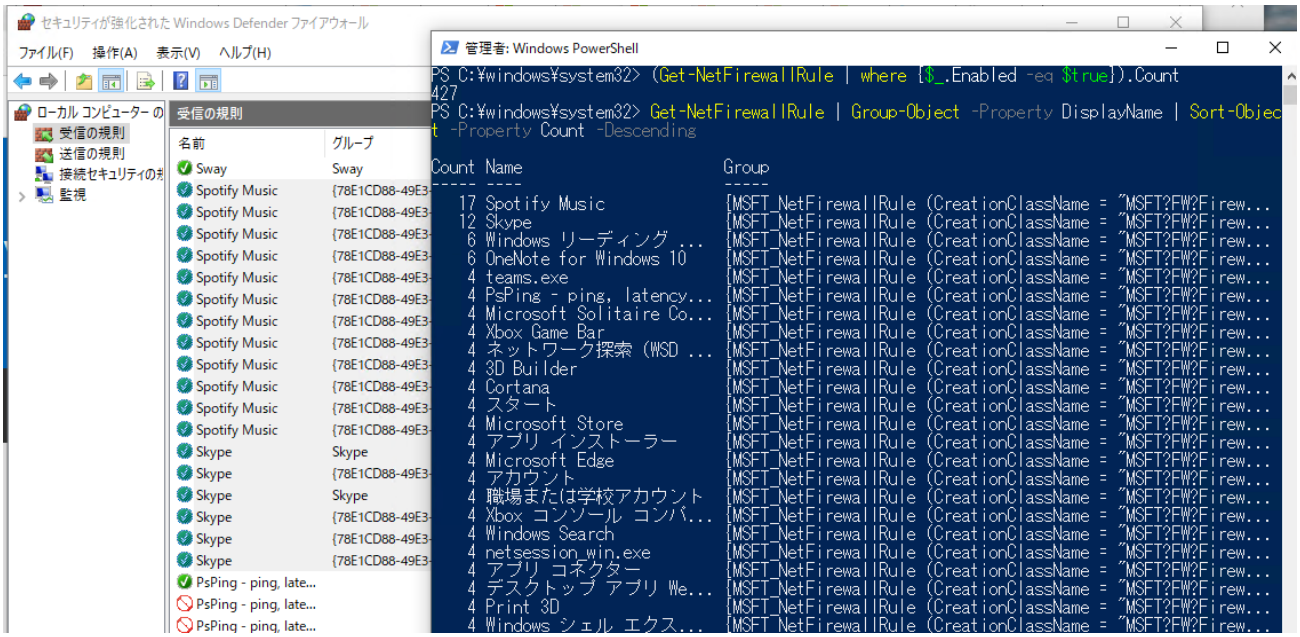
・事例2

Mirantis Container Runtime（旧称、Docker Enterprise）のコンテナーホストでは受信の規則と送信の規則に、次の名前のルール（XXXXは数字）が大量に登録される場合があることを確認しています。これらのルールが存在しなくても、コンテナーのネットワーク機能に影響がないようなので、大量に登録されている場合は削除してもよいでしょう。

```
Get-NetFirewallRule -Displayname "docker*in" ↲

Get-NetFirewallRule -Displayname "docker*out" ↲

Remove-NetFirewallRule -Displayname "docker*in" ↲

Remove-NetFirewallRule -Displayname "docker*out" ↲
```

・事例3

Windows 10ではストアアプリ（UWPアプリ）のインストールや使用により、同じ名前のルールが重複して登録される場合があります。次の例は、Spotify MusicアプリとSkypeアプリのルールが重複して登録されている様子です。



画面6　重複登録されたSpotify MusicアプリとSkypeアプリのルール

実は、このコンピューターではこれらのアプリを使用したことがありません。使用しないアプリの登録を削除しても何の問題もないでしょう。また、削除したとしても必要時に自動的に再登録されるはずです。

Windows 10の機能更新プログラムによるアップグレードを繰り返していると、廃止やアンインストールなどで既に存在しないストアアプリのためのルールが残っている場合もあります。不要なストアアプリ用のルールは削除して問題ないでしょう。

・事例4

リモートデスクトップセッションホストであるWindows Server 2016、Windows Server 2019、および仮想デスクトップインフラストラクチャ（VDI）のWindows 10仮想デスクトップでは、移動ユーザープロファイルやユーザープロファイルディスクの問題で、ファイアウォールルールが肥大化し、デスクトップの機能不全（スタートメニューが開かないなど）、パフォーマンスの劣化、サーバーのハングアップなどの症状を引き起こすことが知られています。該当する場合は、以下の対処方法に従ってください。

Windows Server 2016 および 2019 における Windows ファイアウォール規則の肥大化について

https://jpwinsup.github.io/blog/2020/10/08/RemoteDesktopService/wnf-leak-issue-on-windows-server-2016-and-windows-server-2019/

7.重大障害につながるイベントログを見逃さない

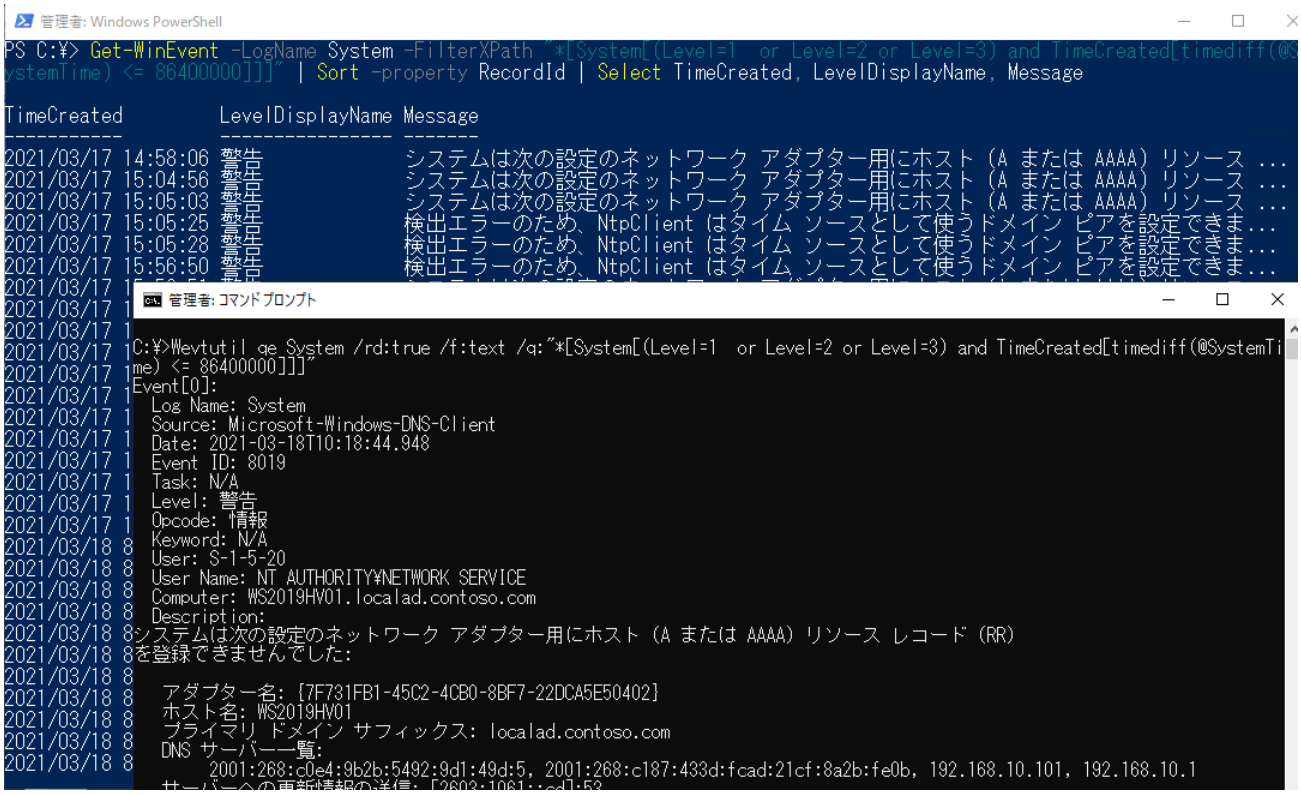
Windowsイベントログは、システムで発生するシステム、アプリケーション、セキュリティに関する膨大なイベントを記録し、提供するサービスです。Windowsイベントログを使用すると、ハードウェアの障害の予兆、容量不足、システムやアプリケーションの重大なエラー、セキュリティ侵害の痕跡などを調査することができます。Windowsに標準搭載されている［イベントビューアー］（Eventvwr.msc）スナップインを使用すると、イベントログを参照したり、フィルター条件を設定して効率的に調査したりできます。しかしながら、［イベントビューアー］（Eventvwr.msc）スナップインは現在進行中の問題を調査するのに快適なパフォーマンスを提供できるとは言えません。また、膨大なイベントに埋もれて、重大なイベントの発生を見逃してしまうこともあるでしょう。

過去24時間に発生した重大／エラー／警告イベントをすばやく確認する

WindowsはWindowsイベントログにコマンドラインからアクセスする複数の方法を用意しています。その中で共通のフィルター条件で利用できるPowerShellのGet-WinEventコマンドレットとWevtutilコマンドの2つを紹介します。

次のコマンドラインの例はいずれも、過去24時間（86400000ミリ秒）にシステム（System）ログに記録された、警告レベル以上のイベント、すなわち警告（レベル3）、エラー（レベル2）、重大（レベル1）イベントをフィルターし、記録された順番に出力するものです（画面9）。

```
Get-WinEvent -LogName System -FilterXPath "*[System[(Level=1 or Level=2 or Level=3) and TimeCreated[timediff(@SystemTime) <= 86400000]]]" | Sort -property RecordId | Select TimeCreated, LevelDisplayName, Message ↲

または

Wevtutil qe System /rd:true /f:text /q:"*[System[(Level=1 or Level=2 or Level=3) and TimeCreated[timediff(@SystemTime) <= 86400000]]]" ↲
```

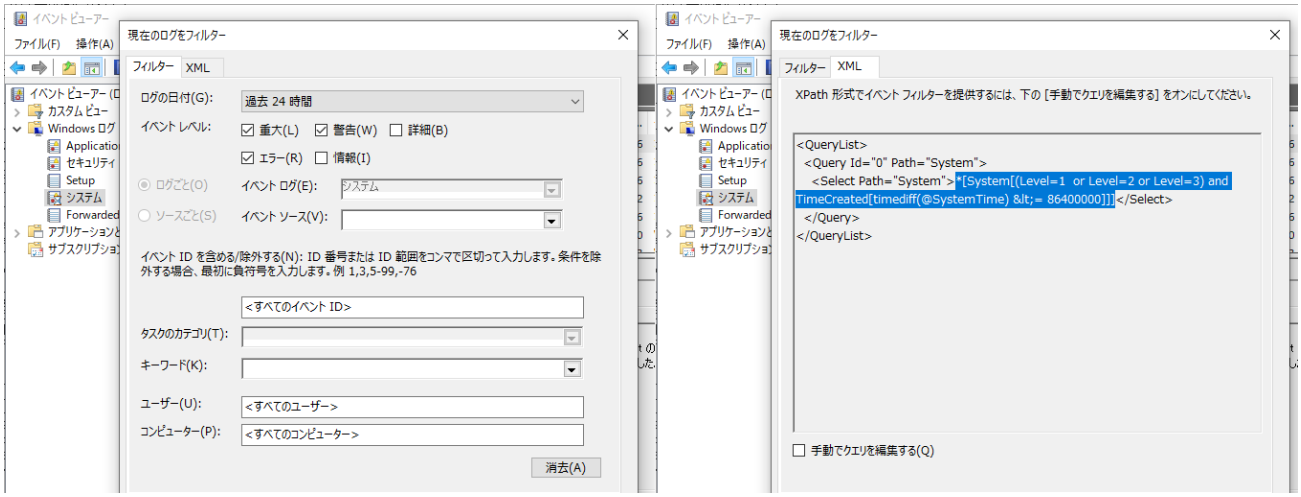


画面9　過去24時間に発生した警告レベル以上のイベントを確認する

構造化（XPath）クエリ式の簡単な作り方

Get-WinEventコマンドレットとWevtutilコマンドはどちらも、構造化（XPath）クエリ式でフィルター条件を設定します。クエリ式を作成するのが難しそうと思うかもしれませんが、実は非常に簡単です。

［イベントビューアー］（Eventvwr.msc）スナップインを開いて、目的のログを選択し、［現在のログをフィルター…］をクリックします。［フィルター］タブのGUIでフィルター条件を設定したら、［XML］タブに切り替えます。すると、<Select>セクション内に目的のクエリ式が見つかります。なお、クエリの中にXMLの中でエスケープされている>と<が含まれる場合は、それぞれ>と<に置き換えて使用してください。目的のクエリ式を取り出したら、フィルター設定は破棄してかまいません。



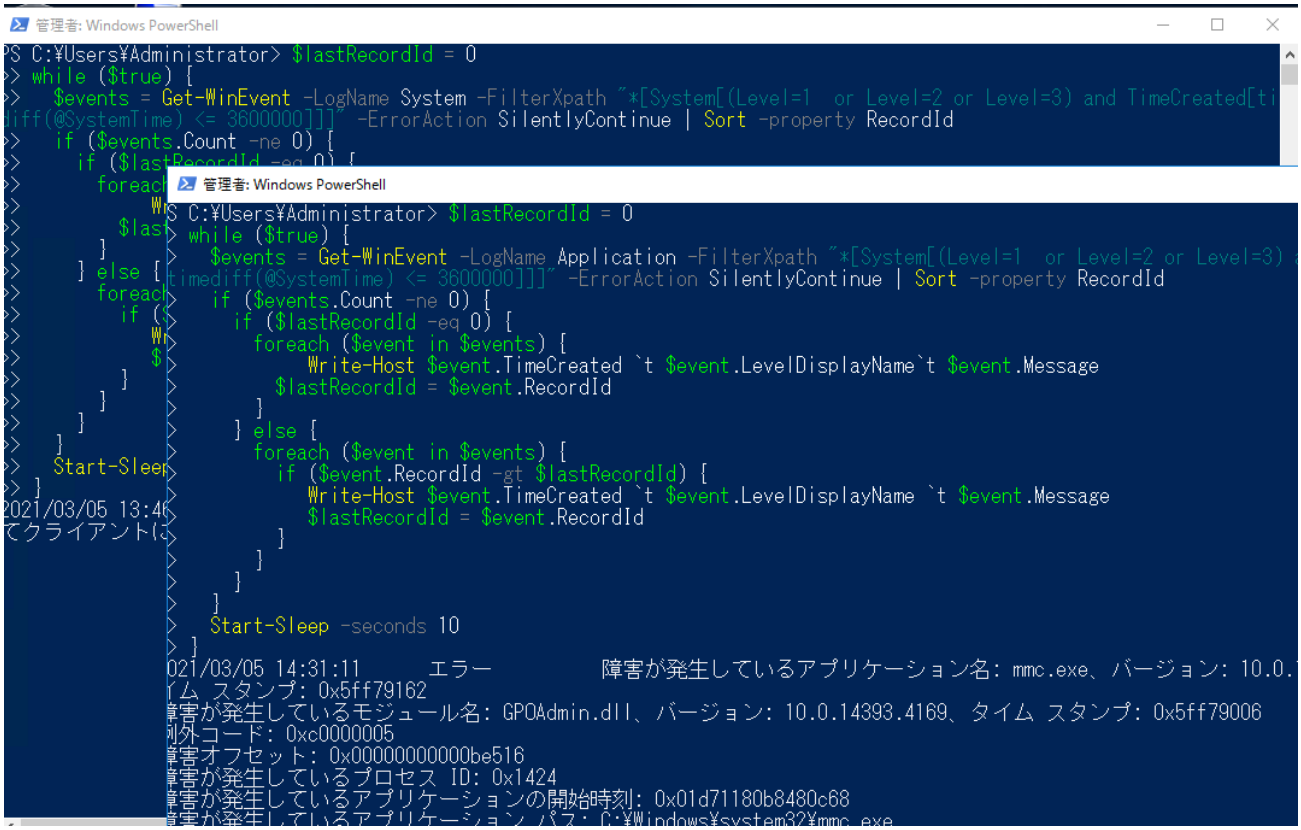
画面10　GUIでフィルター条件を設定すると、XPathクエリが完成する

システムやアプリケーションの重大／エラー／警告イベントをリアルタイムに監視する

現在進行中の問題をリアルタイムに追跡したい、あるいは再現テストをしながらのイベント発生をリアルタイムに監視したいという場合は、PowerShellで簡単なスクリプトを書くことで対応できます。例えば、次のコードは、過去1時間（3600000ミリ秒）にシステムログに記録された警告以上のイベントがあればそれを出力し、その後、10秒間隔で新たに記録される警告以上のイベントを出力し続けます。-LogNameパラメーターのSystemをApplicationに書き換えることで、アプリケーションログに対応させることができます（画面10）。

```
$lastRecordId = 0 ↲

while ($true) { ↲

$events = Get-WinEvent -LogName System -FilterXpath "*[System[(Level=1 or Level=2 or Level=3) and TimeCreated[timediff(@SystemTime) <= 3600000]]]" -ErrorAction SilentlyContinue | Sort -property RecordId ↲

if ($events.Count -ne 0) { ↲

if ($lastRecordId -eq 0) { ↲

  foreach ($event in $events) { ↲

    Write-Host $event.TimeCreated `t $event.LevelDisplayName `t $event.Message ↲

    $lastRecordId = $event.RecordId ↲

} ↲

} else { ↲

foreach ($event in $events) { ↲

   if ($event.RecordId -gt $lastRecordId) { ↲

     Write-Host $event.TimeCreated `t $event.LevelDisplayName `t $event.Message ↲

     $lastRecordId = $event.RecordId ↲

   } ↲

 } ↲

} ↲

} ↲
```

```
    Start-Sleep -seconds 10 ↲

    } ↲
```



画面11　システムとアプリケーションログの警告以上のイベント発生をリアルタイムに監視する

Sysmonで詳細なアクティビティを徹底的にログに記録し、調査する

マイクロソフトが無料提供しているWindows Sysinternalsは、WindowsおよびWindows Serverのシステムを高度に監視、診断、およびトラブルシューティングするのに役立つユーティリティ群です。Windowsイベントログ関連のユーティリティとしては、System Monitor（Sysmon）があります。

Sysmonは、個々のコンピューターやネットワーク上の、潜在的で悪質なアクティビティを追跡するために作成されたものですが、悪質なアクティビティだけでなく、トラブルシューティングの目的にも使用できます。

SysmonはWindows SysinternalsのサイトからZIP（sysmon.ziip）としてダウンロードできるほか、Live Sysinternalsのサイトから実行可能ファイル（sysmon.exe）を直接ダウンロードすることもできます。

Windows Sysinternals｜Sysmon
https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon

Live Sysinternals
https://live.sysinternals.com/sysmon.exe

Sysmonは多数のルールを用いて、表2に示すイベントを検出します。Sysmonが検出したアクティビティはイベントログ（Microsoft-Windows-Sysmon/Operations）に記録され、［イベントビューアー］（Eventvwr.msc）スナップインやその他の標準的なコマンドラインツールから参照できます。

表2　Sysmonのルールと検出可能なイベント（Sysmon v13.02、SchemaVersion 4.50の場合）

ルール名	イベント	イベントID	既定	説明
ProcessCreate	Process Create	1	include	プロセスの作成
FileCreateTime	File creation time changed	2		ファイル作成日時の変更
NetworkConnect	Network connection detected	3		ネットワーク接続の検出
ProcessTerminate	Process terminated	5	include	プロセス終了
DriverLoad	Driver loaded	6		ドライバーの読み込み
ImageLoad	Image loaded	7		実行可能イメージの読み込み
CreateRemoteThread	CreateRemoteThread detected	8		リモートスレッド作成の検出
RawAccessRead	RawAccessRead detected	9		RAWリードアクセスの検出

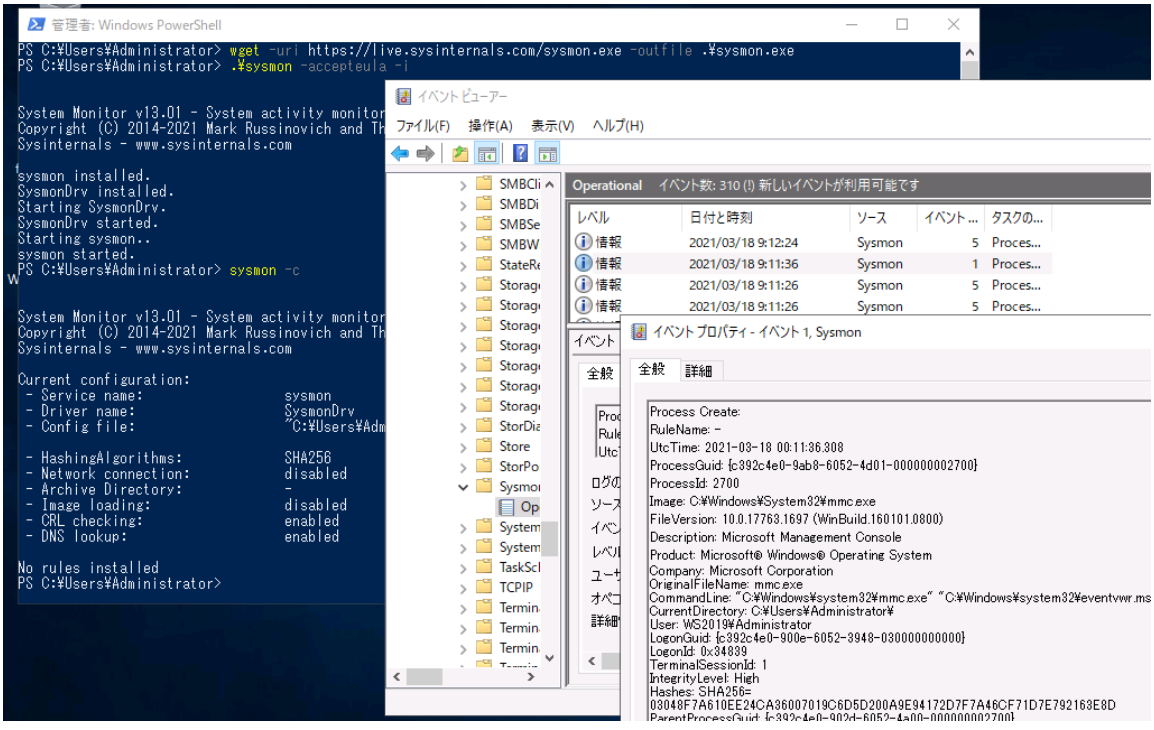
ルール名	イベント	イベントID	既定	説明
ProcessAccess	Process accessed	10		別のプロセスからアクセスの検出
FileCreate	File created	11	exclude	ファイルの作成、上書き検出
RegistryEvent	Registry object added or deleted	12	exclude	レジストリの作成、削除の検出
	Registry value set	13	exclude	レジストリ値の設定の検出
	Registry object renamed	14	exclude	レジストリのリネーム検出
FileCreateStreamHash	File stream created	15	exclude	ファイルストリームの作成検出
PipeEvent	Pipe Created	17	exclude	名前付きパイプの作成検出
	Pipe Connected	18	exclude	名前付きパイプへの接続検出
WmiEvent	WmiEventFilter activity detected	19	exclude	WMIイベントフィルター登録の検出、マルウェアの挙動の可能性あり
	WmiEventConsumer activity detected	20	exclude	WMIコンシューマーの登録検出
	WmiEventConsumerToFilter activity detected	21	exclude	WMIコンシューマーのフィルターバンドの検出
DnsQuery	Dns query	22	exclude	DNSクエリの検出
FileDelete	File Delete	23		ファイルの削除検出
ClipboardChange	Clipboard changed	24		RDP接続でクリップボード共有を検出
ProcessTampering	Process Tampering	25		プロセスのイメージの外部からの変更の検出
–	Sysmon service state changed	4	–	Sysmonサービスの状態変更（開始、停止）の検出

ルール名	イベント	イベントID	既定	説明
–	Sysmon config state changed	16	–	Sysmon構成の変更の検出、フィルタールールの更新など
–	Error report	255	–	Sysmonのエラーの検出

インストールなしで利用できるWindows Sysinternalsの大多数のユーティリティとは異なり、Sysmonはシステムにインストールすることでサービスとして動作し、システムの起動直後からシャットダウンまで、長期にわたって（Sysmonをアンインストールするまで）監視を継続します。Sysmonは構成によっては膨大なイベントを記録し続けるため、フィルター条件を設定して、短期間に集中的に監視することをお勧めします。

Sysmonをインストールするには-iパラメーターを指定して管理者として実行します。すると既定の構成でインストールされ、監視がスタートします。既定の構成では、表2のincludeとなっているプロセスの作成と終了イベント、およびSysmon関連のイベントです。また、ネットワーク接続の検出とイメージの読み込みイベントの監視機能は無効化されます。Sysmonのインストール後、-cパラメーターを指定して実行すると、現在の構成をダンプ出力できます（画面12）。

```
sysmon.exe -acceteula -i ↲

sysmon -c
```



画面12　Sysmonを既定の構成でインストールし監視をスタート。既定の構成では、すべてのプロセスの作成と終了がログに記録される

Sysmonによる監視を停止するには-uパラメーターを指定して実行してSysmonをアンインストールします。アンインストールすると記録済みのイベントログを解釈できなくなります。アンインストール後もイベントログを参照するためには-mパラメーターを指定して実行し、ログの解釈に必要なイベントマニュフェストのみをインストールします。

```
sysmon -u ↲
```

```
sysmon -m ↲
```

特定のプロセスに対象を絞って、イメージの読み込みイベント、およびネットワーク接続（TCP/UDPポートへの接続）イベントを含めて監視するには、次のようなコマンドラインでインストールします。既にSysmonをインストールしてある場合は、いったんアンインストールしてから、再インストールしてください。次の例は、app.exeというプロセスに関連するイベントのみをログに記録します。複数のプロセスを指定する場合は、app.exe, app2.exeのようにカンマで区切って指定してください。

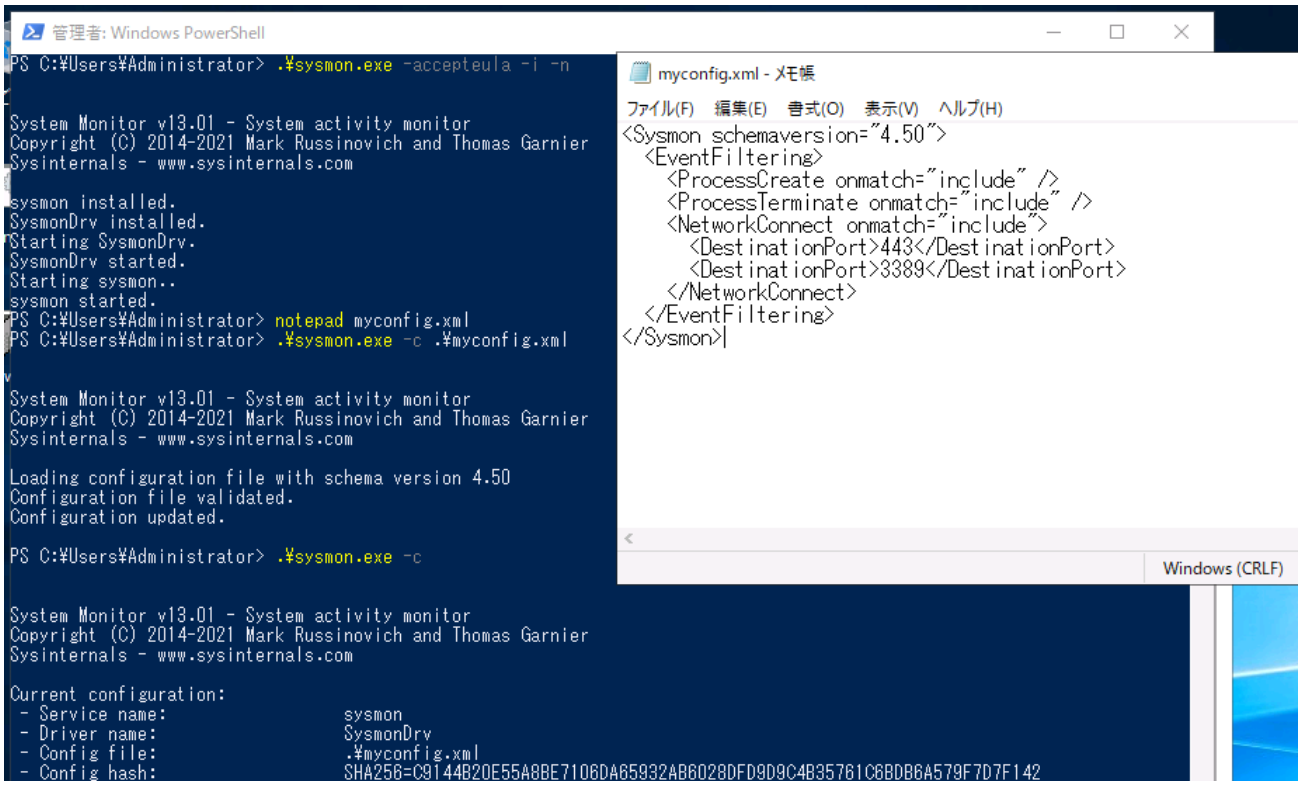
```
sysmon.exe -accepteula -i -l -n app.exe ↲
```

SysmonはXML構成ファイルを使用して複雑なフィルター条件を設定できます。例えば、HTTPS（443）およびRDP（3389）へのネットワーク接続のみを検出させたい場合は、次のようなmyconfig.xmlファイルを記述し、Sysmonをネットワーク接続イベントの検出を有効（-nパラメーター）にしてインストールしたあと、-cパラメーターを付けてXML構成ファイルを読み込ませます（画面13、画面14）。

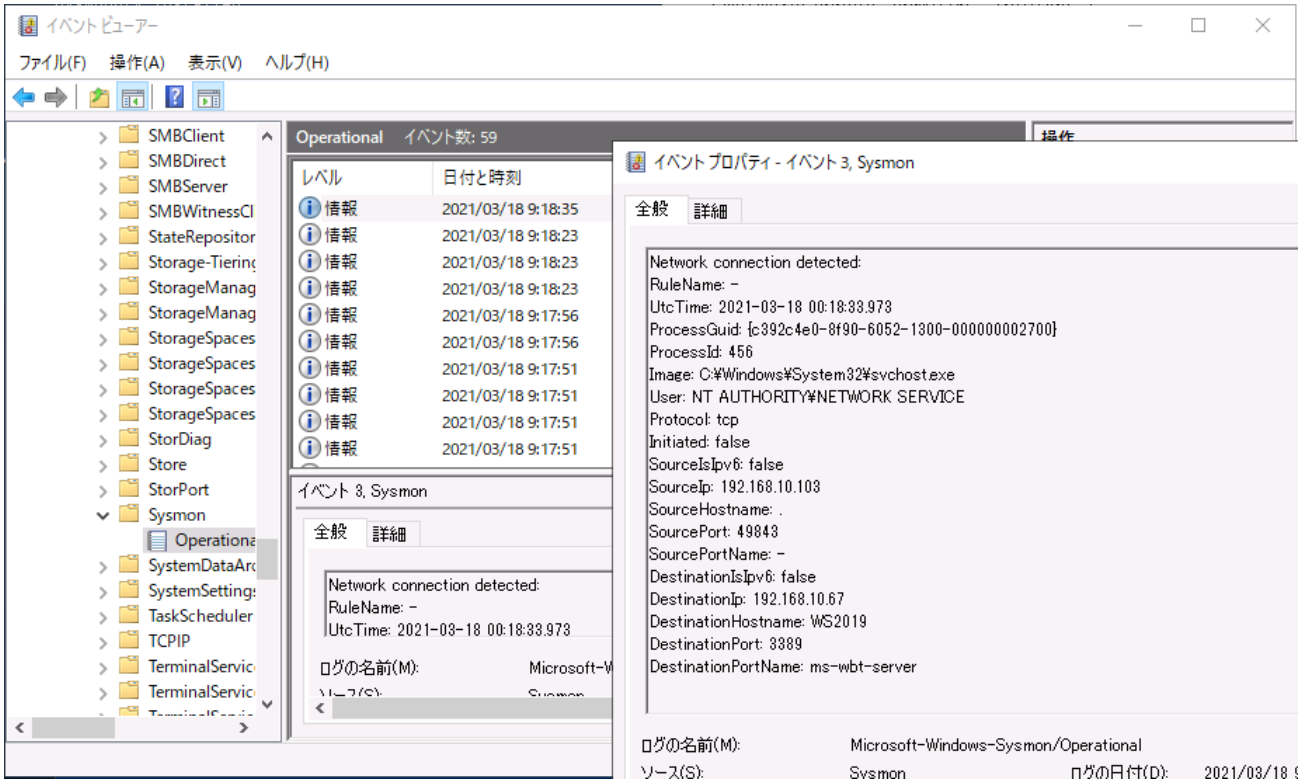
```
sysmon.exe -accepteula -i -n ↲

sysmon -c myconfig.xml
```

myconfig.xml
※SchemaVersionはSysmonのバージョンによって更新されることがあります。適切なSchemaVersionはSysmon -sを実行して確認、変更してください。

```
<Sysmon schemaversion="4.50"> ↲
<EventFiltering> ↲
<ProcessCreate onmatch="include" /> ↲
<ProcessTerminate onmatch="include" /> ↲
<NetworkConnect onmatch="include"> ↲
<DestinationPort>443</DestinationPort> ↲
<DestinationPort>3389</DestinationPort> ↲
</NetworkConnect> ↲
</EventFiltering> ↲
</Sysmon> ↲
```



画面13　XML構成ファイルを使用してSysmonをインストール、構成する



画面14　Sysmonを使用してHTTPS（443）およびRDP（3389）ポートへの着信を監視する

Sysmonをうまく使いこなすことで、知りたい情報に絞って効率よく監視することができます。しかし、複雑な条件を設定するのは難しいかもしれません。Sysmonのダウンロードサイトにある使用例や、以下の書籍の解説が参考になるでしょう。ただし、現在利用できるバージョンは、さらに機能が拡充されており、使用方法のすべてが説明されているわけではないことに留意してください。

Windows Sysinternals徹底解説 改訂新版（日経BP社）

https://www.nikkeibp.co.jp/atclpubmkt/book/17/P98960/

8.突然調子がおかしくなった!? そんなとき頼りになる信頼性モニター

Windowsの標準ツールである信頼性モニターは、システムのパフォーマンスと信頼性に影響する可能性があるエラーや警告イベント、およびアプリケーションのインストールなどの情報イベントの発生状況に基づいて、システムの安定性を日別（過去20日）または週別（過去20週）に1～10の安定性インデックス（インデックス10が最も安定）で評価します。最近、システムの調子が悪い、あるいはあるタイミングで突然、エラーが多発するようになったという場合、信頼性モニターを利用することでトラブルの予兆や原因に迫ることができます。

信頼性モニターを起動するには

信頼性モニターはWindows 10以降レガシな扱いとなったコントロールパネルの深い場所にあるため、信頼性モニターの存在すら知らず、利用したことがないというユーザーも多いようです。

信頼性モニターにアクセスするには、コントロールパネル（Control.exe）の［システムとセキュリティ¥セキュリティとメンテナンス］（Windows Server 2012 R2以前は［システムとセキュリティ¥アクションセンター］）を開き、［メンテナンス］の項目に含まれる［信頼性履歴の表示］をクリックします（画面15）。

画面15　WindowsやWindows Serverで利用できる信頼性モニターの安定性のレポート

信頼性モニターが初めて搭載されたWindows VistaおよびWindows Server 2008（フルインストール）ではパフォーマンスモニター（信頼性とパフォーマンスモニター）の機能の一部として提供され、コントロールパネルには含まれていませんでした。その名残というわけではありませんが、パフォーマンスモニターのコマンド（Perfmon.exe）に/rel（信頼性を示す**Rel**iavilityの略）パラメーターを指定して実行すると、すばやく信頼性モニターを開くことができます。また、［パフォーマンスモニター］（Perfmon.msc）スナップインを開き［モニターツール］を右クリックして［システム信頼性の表示］をクリックすることで開くこともできます。

> Perfmon /rel ↵

Windows Server 2012 R2以前のレガシなWindows Serverバージョンでは、信頼性モニターに何も記録されない場合があります。以下のサポート情報の解決策に従って対処することで、信頼性モニターによる評価を開始することができます（過去に発生したイベントや安定性インデックスは作成されません）。

Reliability Monitor displays no information in Windows Server
https://docs.microsoft.com/en-us/troubleshoot/windows-server/performance/reliability-monitor-shows-no-information

トラブルシューティング事例

信頼性モニターを使用したトラブルシューティング事例を紹介します。

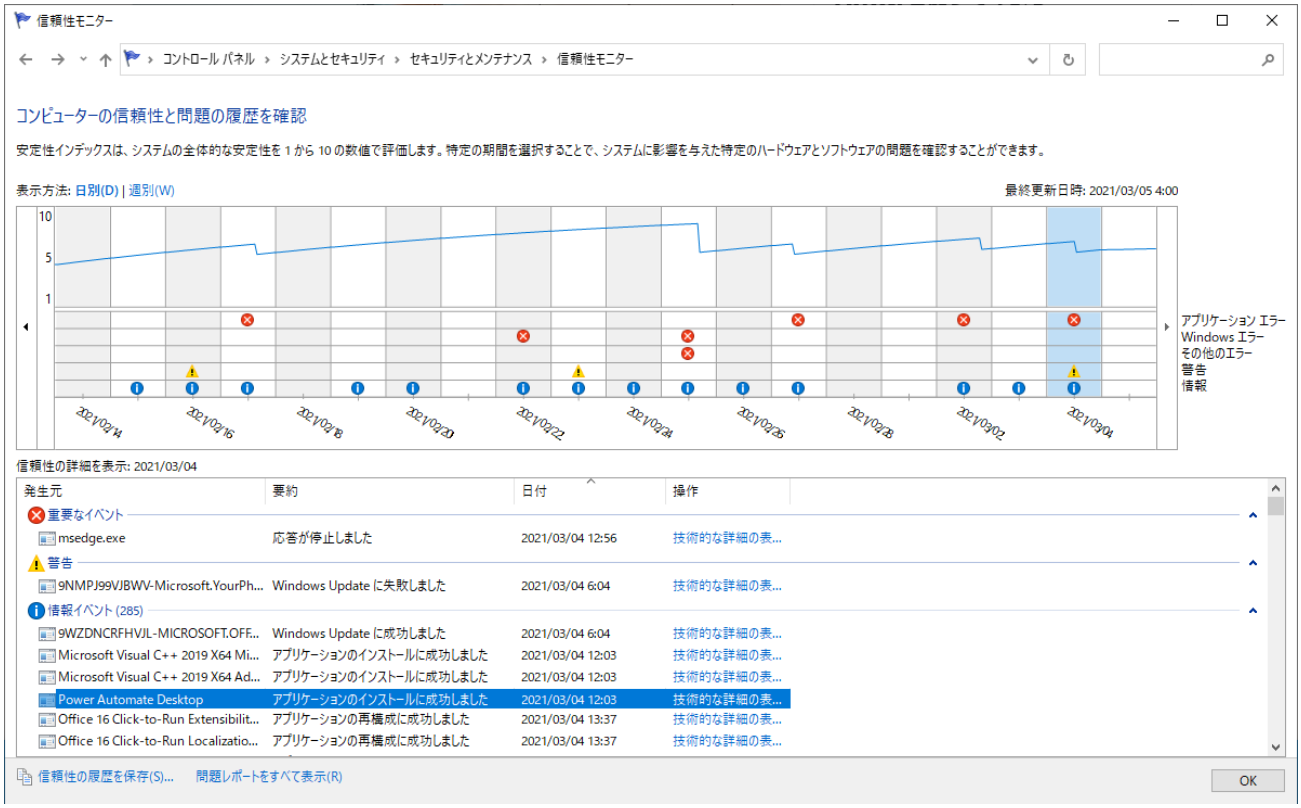
先日、私はWindows 10にインストールしたWindows Admin Centerを利用して、複数台のWindows Serverをリモート管理していました。ふと、ローカルのWindows 10の管理画面はどうなっているのだろうと接続先を切り替えようとしたところ、ローカルのWindows 10にだけ応答なしになったり、エラーで接続できなくなっていることに気付きました（画面16）。数時間前までは確かに問題なく接続できていました。

画面16　Windows 10にインストールしたWindows Admin Centerで、ある日、どこかのタイミングで突然、ローカルのWindows 10にだけ接続できなくなった

思い当たることがあるとすれば、このエラーが発生する少し前、マイクロソフトが無料公開したばかりの、あるデスクトップアプリケーションをインストールしたことです。しかし、それが直接的な原因かどうかは定かではありませんし、インストールした時間まで正確には覚えていません。

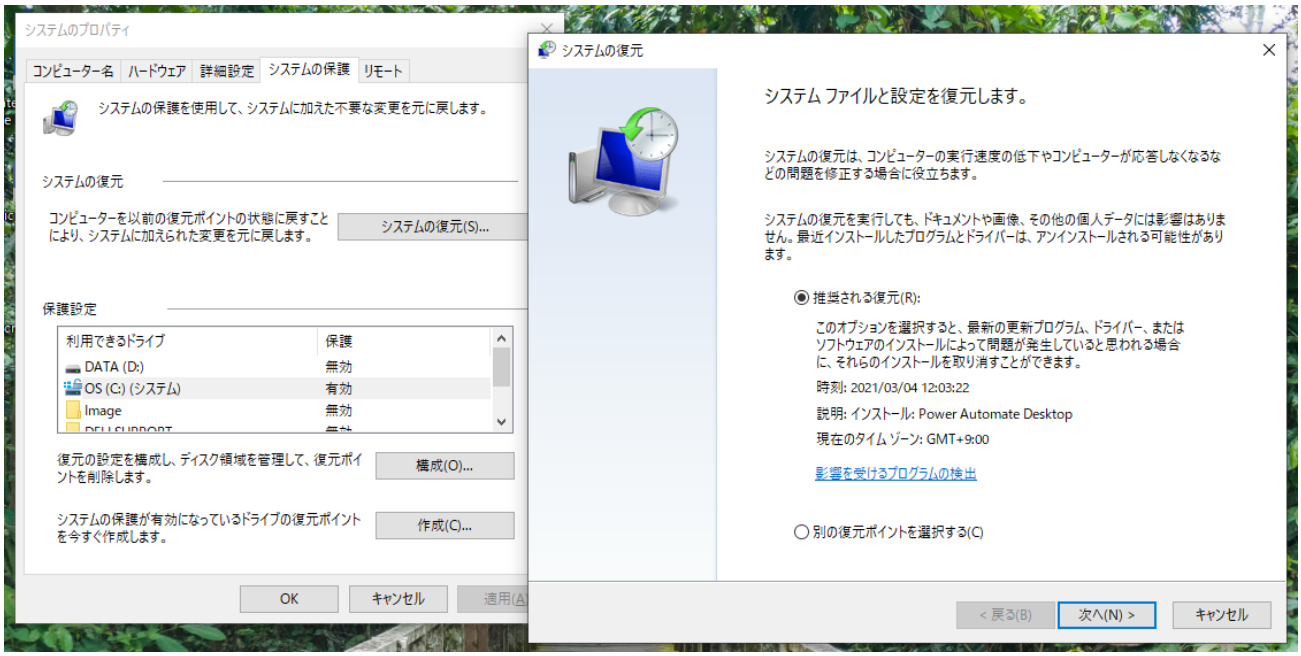
そこで、信頼性モニターを開いてここ数時間のイベントを確認してみました。すると、Windows Admin Centerを開いていたMicrosoft Edgeで応答なしのエラーが発生するまでの1時間の間に、疑っていたアプリケーション（Power Automate Desktop）とその関連コンポーネント（Microsoft Visual C++ 2019 Runtime）インストールが情報イベントとして記録されていました（画面17）。

別のWindows 10の仮想マシン環境にWindows Admin Centerをインストールし、ローカルのWindows 10に接続できることを確認した上で、問題のアプリケーションをインストールしてみると、同じ問題が再現されました。問題の原因は特定できたので、問題のアプリケーションと関連コンポーネントをアンインストールし、コンピューターを再起動してみましたが、問題は解消されませんでした。Windows Admin Centerのアンインストールと再インストールも実施しましたが、それでも問題は解消されませんでした。**（注：この問題は Power Automate Desktop の当時のバージョンで確認しましたが、現在のバージョンでは再現しません。バージョン固有の問題あるいは筆者の環境固有の問題の可能性もあります。）**

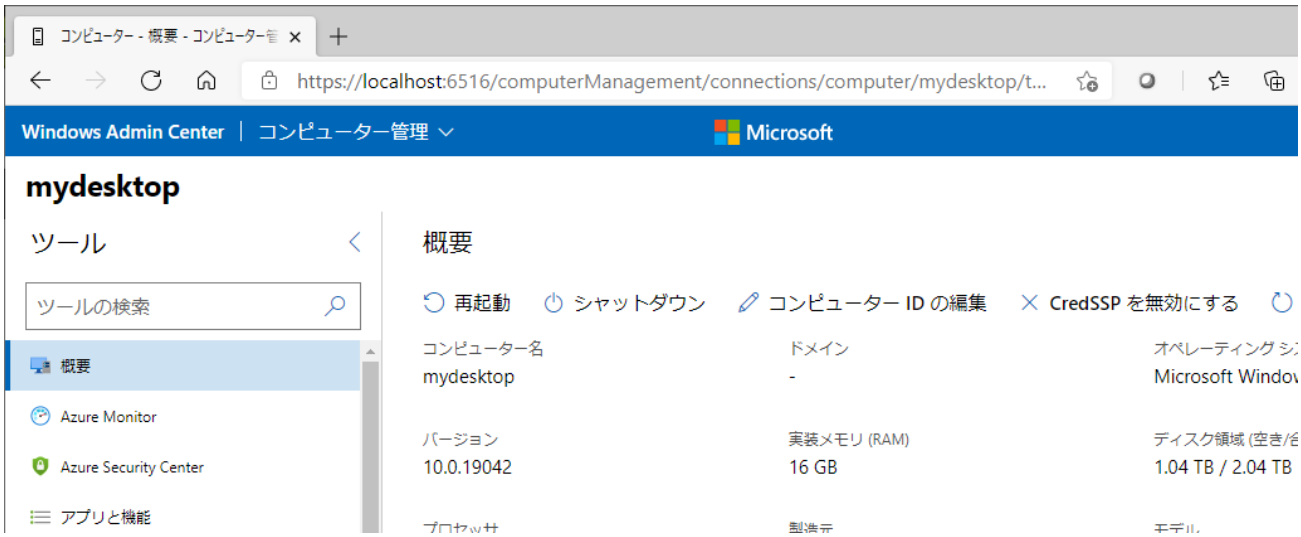


画面17　Microsoft Edgeのエラーが発生する少し前に、疑っていたアプリケーションをインストールしていたことを確認

Windowsは更新プログラムのインストール、ドライバーのインストール、アプリケーションのインストールといったタイミングでシステムの保護の復元ポイントを自動作成します（仮想マシンの場合は保護設定が既定で無効になっている場合があります）。問題のアプリケーションのインストール直前に作成された復元ポイントを復元することで、Windows Admin Centerの問題は完全に解消しました（画面18、画面19）。



画面18　問題のアプリケーションのインストール直前に自動作成された復元ポイントを使ってシステムを復元する



画面19　システムの復元によりWindows Admin Centerのローカル接続問題は解消

なお、オンラインの状態からシステムの復元を実行できない（OSが起動しない、あるいは復元が失敗する）場合は、Windows回復環境（WinRE）からシステムの復元を開始（トラブルシューティング｜詳細オプション｜システムの復元）できる場合があります（画面20）。また、システムの復元が無効になっており、利用可能な復元ポイントがない場合は、過去に取得済みのシステムのフルバックアップからシステム全体またはシステム状態を復元することができます。

画面20　Windowsからシステムの復元ができない場合は、Windows回復環境から実行する

問題を解決しようと、関係のないシステム設定を変更したりすると、別の問題を引き起こす、原因の特定が困難になるなど、問題が悪化することがあります。システム設定を意図的に変更していない限り、突然の問題発生の原因はシステム設定ではありません。オンラインのサービス関連の問題であれば、ローカル側のシステム設定は全く関係なく、サービス提供者側の障害や一時的な問題かもしれません。

それ以外の問題、WindowsのシステムエラーやSTOPエラー（ブルースクリーン、BSoD）、アプリケーションのエラーは、Windows Updateの更新プログラムのインストール、ドライバーのインストールや更新、アプリケーションのインストールや更新、アンインストールのタイミングで発生することがあります。信頼性モニターを使用すると、これらの履歴が情報イベントとして記録され、その前後のエラーの発生状況から関連性を調査することができます。

9.WSUSが遅くなった？ 長期運用WSUSのメンテナンス

Windows Serverのサーバーの役割の1つであるWindows Server Update Services（WSUS）は、企業内ネットワークでWindows、Windows Server、およびその他のマイクロソフト製品の更新プログラムの管理の一元化と配布の自動化を可能にする、Windowsネットワークにおける重要なインフラストラクチャサービスです。

しかし、WSUSサーバーを長期に運用していると、WSUSの管理コンソールでエラーが発生して管理画面にアクセスできなくなったり、パフォーマンスが低下したり、WSUSのコンテンツの格納先のディスク領域が不足したりといった、さまざまなトラブルを発症することがあります。

特に、WSUS管理コンソールのエラーに関する問題については、マイクロソフトの公式ドキュメントおよびWSUSサポートチームの情報が役に立つでしょう。また、WSUSの安定運用に関するホワイトペーパー『WSUS正常性監視のポイント　Windows 10時代の重要インフラWSUS、安定運用の勘所』も参考にしてください。

WSUS と Configuration Manager SUP のメンテナンスに関する完全なガイド

https://docs.microsoft.com/ja-jp/troubleshoot/mem/configmgr/wsus-maintenance-guide

「WSUS 管理コンソールにつながらない！」を解消するための 7 つのワザ（JAPAN SCCM & WSUS Support Team）
https://social.msdn.microsoft.com/Forums/ja-JP/0dc69153-1d4e-4e91-bf91-df311424a8be/wsus-7-?forum=jpsccmwsus

WSUS正常性監視のポイント　Windows 10時代の重要インフラWSUS、安定運用の勘所
https://www.say-tech.co.jp/yamaichi/key_point_of_wsus_monitoring

ここでは、WSUSを長期的に安定運用するのに役立つ、一般的な考慮点やメンテナンスタスクの定期実行について説明します。既にパフォーマンスの低下や容量不足に陥っているWSUSサーバーについても、これらの手順を実施することで劇的に改善する可能性があります。

SUSDB用WID/SQL Serverの最小／最大メモリの調整

WSUSデータベース（SUSDB）をホストするWindows Internal Database（WID、Windows Server付属のサーバーの機能の1つ）またはSQL Serverのサーバーメモリオプションは、WSUSサーバー全体のパフォーマンスを最適化できる重要な調整ポイントです。

WIDおよびSQL Serverのサーバーメモリオプションは、最小サーバーメモリと最大サーバーメモリがあり、既定値は最小0MB、最大2,147,483,647MB（約2ペタバイト）で、SQL Serverがメモリを動的に自動管理します。最大サーバーメモリを既定から変更していない場合、SQL Serverは最大で、システムやSQL Server以外のアプリケーションに必要なメモリ容量を物理メモリ全体量から差し引いた容量を割り当てます。SQL Serverの負荷が高く、既に利用可能な最大容量を使用している場合、新たに起動するサービスやアプリケーションや、既存のプロセスが追加で要求するリソースを確保できないという状況が発生します。

動的メモリを使用する仮想マシン仮想環境では、最小サーバーメモリをメモリを予約しておく目的で重要です。SUSDBをホストするSQL Serverでは、最小サーバーメモリを1024MBに設定することで安定したという実績があるそうです。

物理サーバーメモリを指定すると、サーバーの物理メモリ容量に余裕がない場合に、システムやSQL Server以外のアプリケーションのためにメモリを予約できます。

最小サーバーメモリおよび最大サーバーメモリは、SQL Server用の管理ツールであるSQL Server Management Studio（SSMS）を使用して簡単に変更できます。SUSDBのホストにWIDを使用している場合は、以下のURLから最新のSSMSをダウンロードしてWSUSサーバーにインストールしてください。

SQL Server Management Studio (SSMS) のダウンロード
🌏https://docs.microsoft.com/ja-jp/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15

SSMSを起動したらSQL Serverのインスタンスまたは WIDをホストするサーバーのデータベースエンジンに接続します。WIDを使用している場合はサーバー名として以下の名前を入力して接続してください。

¥.¥pipe¥MICROSOFT##WID¥tsql¥query

サーバーに接続したらサーバー名を右クリックして［プロパティ］を選択し、［メモリ］を選択し、［最小サーバーメモリ］および必要に応じて［最大サーバーメモリ］を変更し、［OK］をクリックします（画面21）。設定の変更は即座に反映されます。

画面21　SQL Serverのサーバーのプロパティを開き、［最小サーバーメモリ］および必要に応じて［最大サーバーメモリ］を変更する

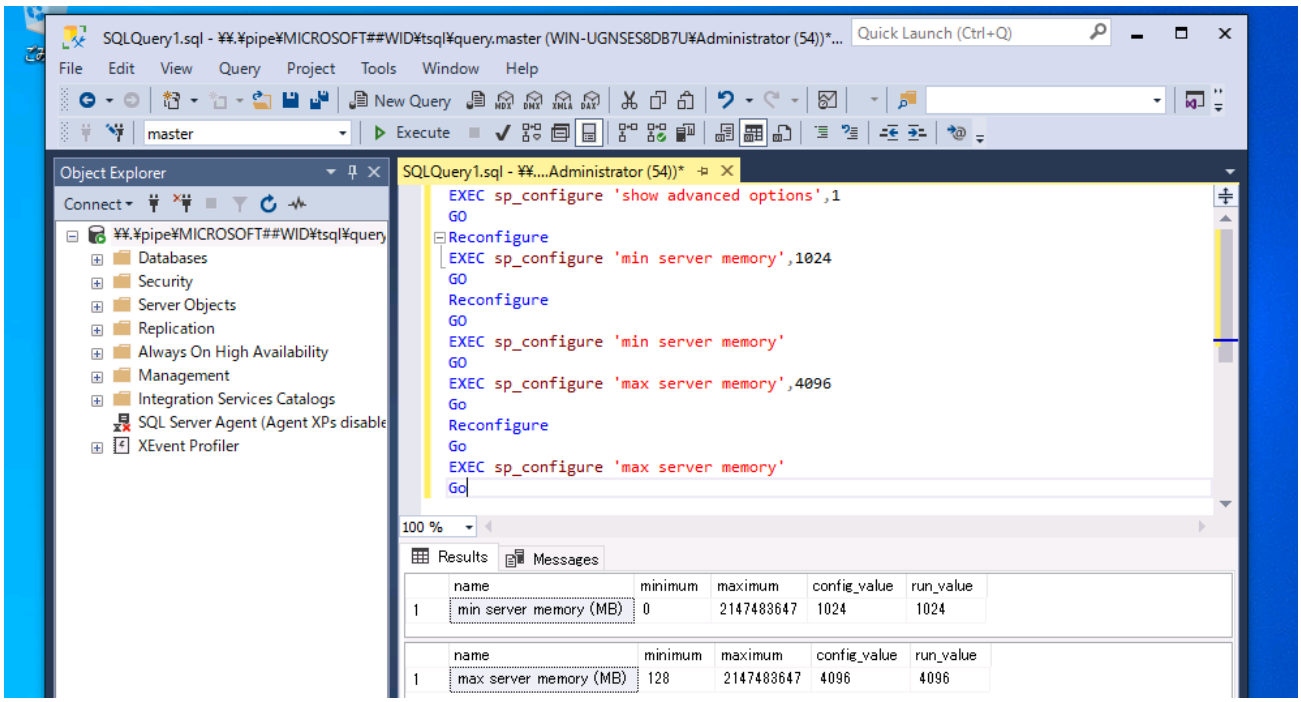
WIDを使用している場合、SSMSでデータベースサーバーのプロパティを開くとエラーが発生する場合があります。その場合は［新しいクエリ（New Query）］を開き、次のようなクエリを入力して実行（Execute）し、設定値の変更を確認します（画面22）。次の例は最小サーバーメモリを1024MB（1GB）に変更します。

```
sp_configure 'show advanced options', 1 ↲

GO ↲

RECONFIGURE ↲

GO ↲

sp_configure 'min server memory', 1024 ↲

GO ↲

RECONFIGURE ↲

GO ↲

EXEC sp_configure 'min server memory' ↲

GO ↲
```

次の例は最大サーバーメモリを4096MB（4GB）に変更します。

```
sp_configure 'show advanced options', 1 ↲

GO ↲

RECONFIGURE ↲

GO ↲

sp_configure 'max server memory', 4096 ↲
```

```
GO ↲

RECONFIGURE ↲

GO ↲

EXEC sp_configure 'max server memory' ↲

GO ↲
```



画面22　WIDの最小サーバーメモリと最大サーバーメモリをTransactSQLクエリで変更する

高速インストールオプションの必要性の検討

Windows 10バージョン1809およびWindows Server 2019では、累積更新プログラム（Cumulative Updates）と呼ばれる毎月の品質更新プログラムのWindows Updateでの配布方法とパッケージ形式に大きな変更が加えられました。

Windows 10バージョン1803以前の従来の累積更新プログラムは、以前にリリースされたすべての修正プログラムが含まれています。半期チャネル（SAC）ではあまり問題になることはありませんが、複数年にわたって品質更新プログラムが提供される長期サービスチャネル（LTSC）の場合、毎月の累積更新プログラムに累積される以前の修正プログラムが増える影響で、更新プログラムのパッケージサイズが肥大化し続け、ダウンロードのためのネットワーク帯域幅に直接的な影響を与えてしまうことがあります。例えば、Windows Server 2016用の累積更新プログラムは半年ほどで1GBを超え、リリースから4年以上経過した2021年春には1.6GBを超えるまでになっています。

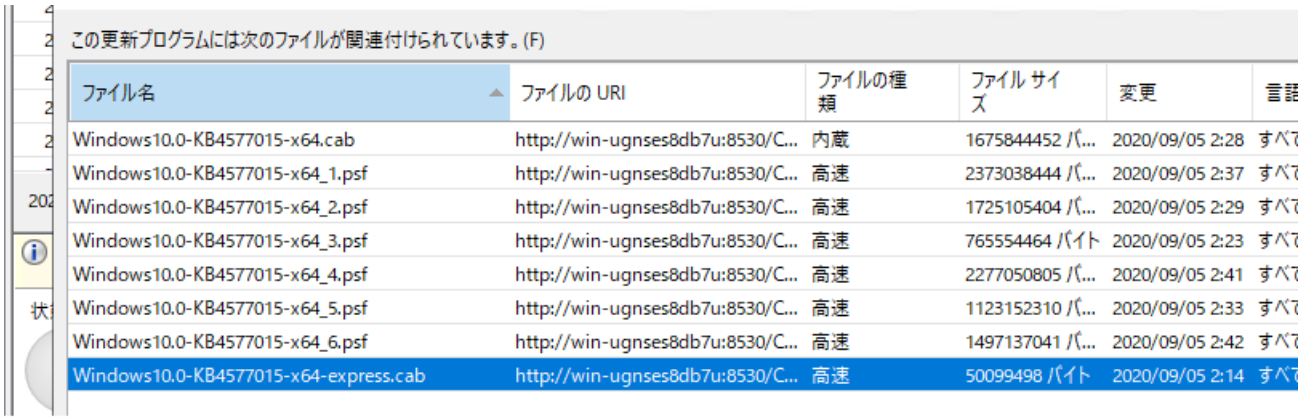
累積更新プログラムのサイズの肥大化に対して、Windows Updateにおけるダウンロード時間とネットワーク帯域幅の圧迫を改善するため、高速インストール（Express Install）または高速ダウンロード（Express Download）という仕組みが利用されてきました。高速インストール／高速ダウンロードでは、フルパッケージ（例：Windows10.0-KBXXXXXXX-x64.cab）をダウンロードする代わりに、カタログを含むExpressパッケージ（例：Windows10.0-KBXXXXXXX-x64_express.cab）がまずダウンロードされ、Expressパッケージの内容に基づいて、更新が必要なものだけを差分でダウンロードするようになっています。

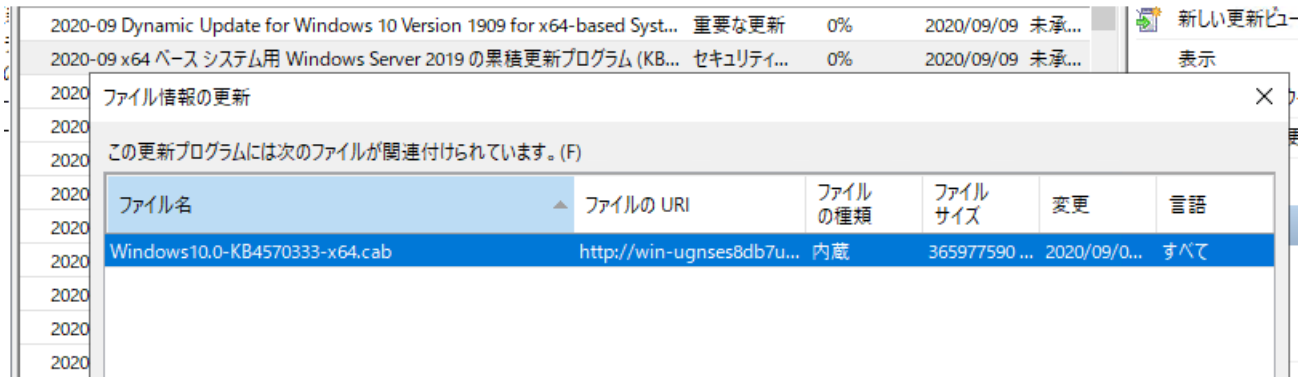
Windows 10バージョン1809およびWindows Server 2019では、Windows Updateでの累積更新プログラムのダウンロードに高速インストール／高速ダウンロードは原則として使用しなくなり、フルパッケージをダウンロードするようになりました（サービススタックの更新プログラムなど一部の更新プログラムについては引き続きExpressパッケージが使用されます）。新しい累積更新プログラムは以前にリリースされたすべての修正プログラムを含むのではなく、前方差分と後方差分を含む形で更新履歴を追跡可能にしています。これにより、フルパッケージのサイズは小さく抑えられ、年月とともに肥大化するということはなくなりました。詳しくは、以下のホワイトペーパーで説明されています。

前方差分と後方差分を使用したWindowsの更新プログラム（Windows Updates using forward and reverse differentials）
https://docs.microsoft.com/ja-jp/windows/deployment/update/psfxwhitepaper

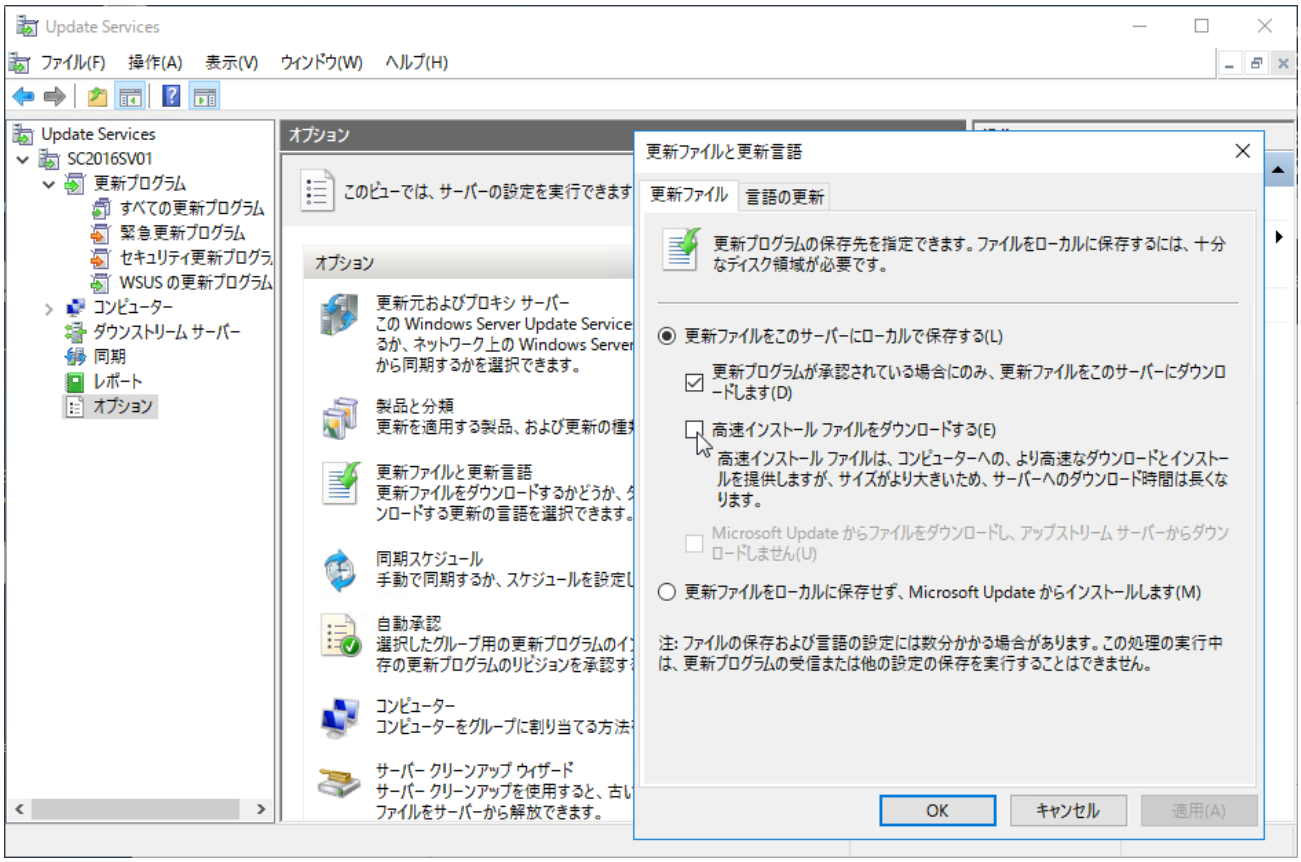
Windows 10バージョン1803以前（Windows Server 2016を含む）のWindows Updateで利用される高速インストール／高速ダウンロードの仕組みは、WSUSでも［高速インストールファイルをダウンロードする］というオプションを有効化することで対応できます。このオプションを有効化した場合、企業内ネットワーク内での高速インストール／高速ダウンロードを提供できますが、WSUSサーバーにMicrosoft Updateからダウンロードされるコンテンツは大幅に増加します。そのため、サーバーへのダウンロード時間が長くなり、WSUSコンテンツのディスク領域をより多く消費するようになります。

もし現在、WSUSで［高速インストールファイルをダウンロードする］オプションを有効にして運用している場合、その必要性を検討してください。企業内ネットワークで稼働中のクライアントやサーバーの、Windows 10バージョン1809およびWindows Server 2019以降への移行が完了し、以前のバージョンのWindows 10やWindows Server 2016が存在しなくなったのであれば、［高速インストールファイルをダウンロードする］オプションを有効にしているメリットはもうありません（画面23、画面24）。





画面23　WSUSに同期された累積更新プログラムのファイル情報の比較。Windows 10バージョン1803以前向けはフルパッケージ（.cab）と高速インストール用（_express.cabおよび.psf）の両方が用意されているが（画面上）、Windows 10バージョン1809以降はフルパッケージ（.cab）の提供のみ



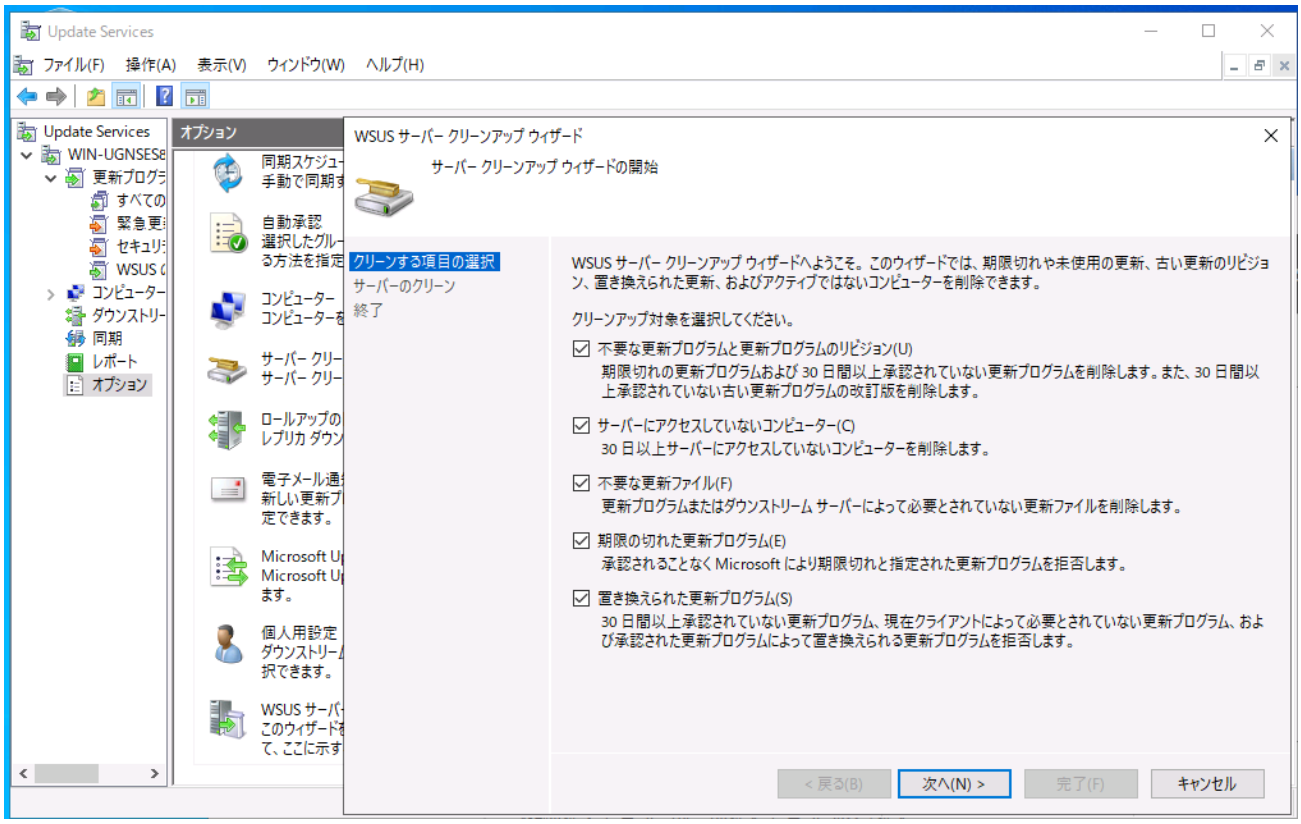
画面24　企業内のWindowsバージョンによっては［高速インストールファイルをダウンロードする］オプションを有効化するメリットはほとんどない

WSUSサーバークリーンアップウィザードの定期的な実行

毎月リリースされる更新プログラムおよび数か月ごとにリリースまたは改訂される機能更新プログラムの情報はWSUSデータベース（SUSDB）に蓄積され、承認に基づいてダウンロードされたWSUSコンテンツ（WSUSContent）ディレクトリに格納される更新プログラムの本体はドライブのディスク領域を使用し続けます。

WSUSはデータベースのデータとコンテンツディレクトリからのファイルの削除に関しては自動管理を行いません。そのため、数か月に1回、あるいは毎月（同期完了後）など定期的にメンテナンスを実施することが重要です。それには、［Update Services］（wsus.msc）スナップインの［オプション］から［WSUSサーバークリーンアップウィザード］を開始して、すべてのクリーンアップ対象を選択して実行します（画面25）。

なお、WSUSをダウンストリームサーバーとアップストリームサーバーの多階層構成で展開している場合は、階層の下部から順番にクリーンアップを実施するようにしてください。同じ階層の場合は複数のサーバーで同時にクリーンアップを実行できます。



画面25　WSUSサーバーを安定的に運用するためには、［WSUSサーバークリーンアップウィザード］を定期的に実行してデータベースとコンテンツディレクトリを最適化する

クリーンアップを定期的に実施していれば短時間で終了しますが、そうでない場合はすべての対象をクリーンアップするのに数時間単位の時間を要する場合があります。また、クリーンアップの実行中に複数回、WSUS Serviceサービスが再起動されます。長時間かかる場合、管理コンソールへのアクセスやパフォーマンス、クライアントアクセスに影響する可能性があります。影響を最小限にするためには、クライアントに影響しない業務時間外の日時に、クライアントの更新スケジュールと重複しない形で実施することをお勧めします。

PowerShellのInvoke-WsusServerCleanupコマンドレット（Windows Server 2012以降のWSUSで利用可能）を使用すると、パラメーターで1つ以上のクリーンアップ対象を指定することでコマンドラインからクリーンアップを実施することができます。クリーンアップ対象とパラメーターの対応を表3にまとめました。コマンドラインを記述したPowerShellスクリプト（.ps1）をタスクとして登録（**Windowsの運用管理を快適にする10の裏ワザ／表ワザ**　『[4. タスクスケジューラの使いこなし』を参照してください）することで、毎月1回、あるいは数か月に1回といったサイクルで自動実行することができます。

表3　クリーンアップ対象とInvoke-WsusServerCleanupコマンドレットのパラメーターの対応

WSUSサーバークリーンアップウィザードでのクリーンアップ対象	Invoke-WsusServerCleanupのパラメーター
不要な更新プログラムと更新プログラムのリビジョン	-CompressUpdate -CleanupObsoleteUpdates
サーバーにアクセスしていないコンピューター	-CleanupObsoleteComputers
不要な更新ファイル	-CleanupUnneededContentFiles
期限の切れた更新プログラム	-DeclineExpiredUpdates
置き換えられた更新プログラム	-DeclineSupersededUpdates

次のWsusmaintenance.ps1は、ホワイトペーパー『WSUS正常性監視のポイント　Windows 10時代の重要インフラWSUS、安定運用の勘所』で紹介したサンプルスクリプトです。すべてのクリーンアップ対象にクリーンアップを実行し、実行結果をログファイルにリダイレクトします。

Wsusmaintenance.ps1

```
$starttime = Get-Date -Format "yyyyMMdd-HHmm" ↲
$result = (Invoke-WsusServerCleanup -CompressUpdates ↲
    -CleanupObsoleteUpdates ↲
    -CleanupObsoleteComputers ↲
    -CleanupUnneededContentFiles ↲
    -DeclineExpiredUpdates ↲
    -DeclineSupersededUpdates) ↲
$endtime = Get-Date -Format "-HHmm" ↲
$result > C:¥work¥wsusmaintenance_$starttime$endtime.logs ↲
```

その他のクリーンアップスクリプト

［WSUSサーバークリーンアップウィザード］と同等、あるいはそれ以上の機能を持つメンテナンスタスクをタスクスケジューラで自動実行することを目的とした有償／無償のツールや製品*4がいくつか存在します。その中で、Samer Sultan氏がGitHubで無償で公開しているWSUS Cleanup PowerShell Scriptを紹介します。

WSUS Cleanup PowerShell Script（GitHub samersultan/wsus-cleanup）
https://github.com/samersultan/wsus-cleanup

*4　筆者は以前、Adam Marshall氏（https://www.adamj.org/）が作成し、無料公開していたClean_WSUS.ps1スクリプト（通称、Adamj Clean-WSUS）をWSUSのメンテナンスに優れたツールとして利用および紹介していましたが、現在、数年前にこのスクリプトの提供は終了しています。Adamj Clean-WSUSは後継のWSUS Automated Mantenanceという有償ツールに置き換えられました。その早期のバージョンであるAdamj Clean-WSUSは使用しないことが推奨されています。

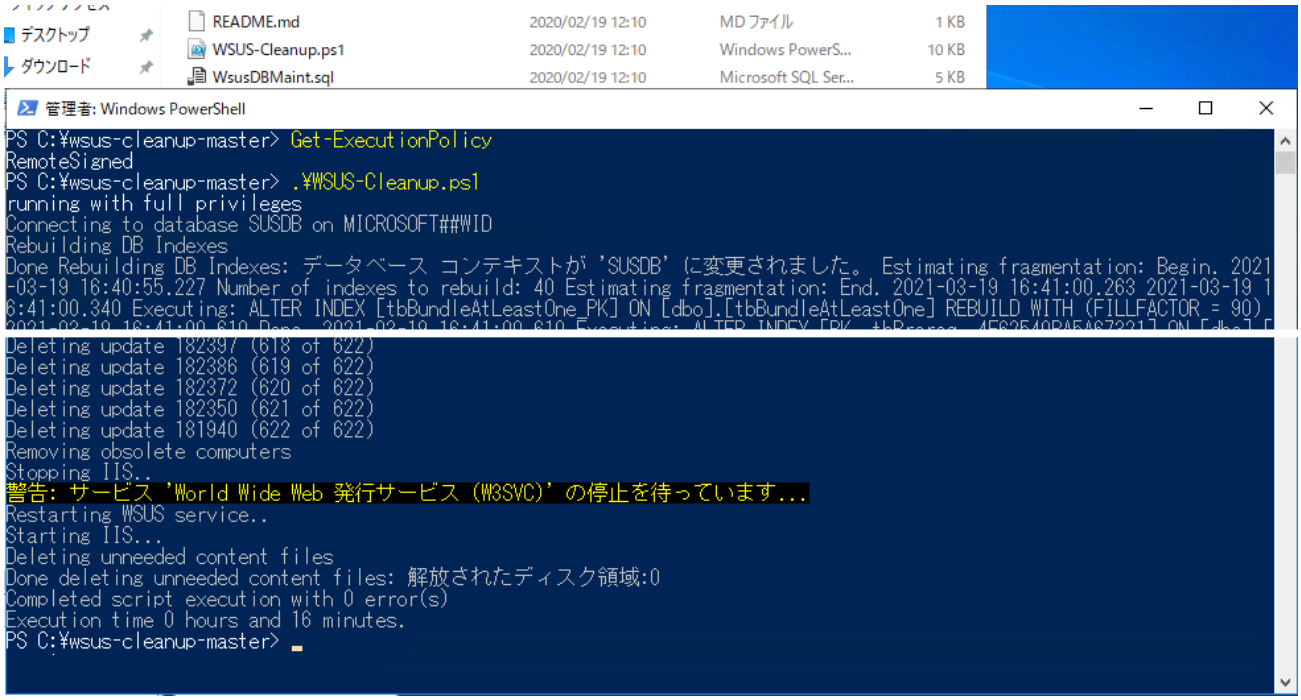
WSUS Cleanup PowerShell Scriptは、［WSUSサーバークリーンアップウィザード］やInvoke-WsusServerCleanupコマンドレットが行うのと同じメンテナンスタスクを、SQL Serverのストアドプロシージャを直接的に実行することで、タイムアウトエラーなどを回避しながらメンテナンスタスクを確実に実行するように作られています。さらに、データベースのインデックスの再構築という追加のデータベースメンテナンスタスクも実施してくれます。しかも、最後にIISとWSUSサービスの再起動は発生するものの、実行中にWSUSへのクライアントアクセスや管理アクセスが制限されることはありません。

このスクリプトはマイクロソフトが提供するものではなく、コミュニティベースのものですが、通常の［WSUSサーバークリーンアップウィザード］がエラーで失敗する場合は、試してみる価値はあります。

使い方は非常に簡単で、ダウンロードしたWSUS-Cleanup.ps1およびWsusDBMaint.sqlを格納したディレクトリから、WSUS-Cleanup.ps1を管理者として実行するだけです（画面26）。タスクスケジューラに登録して（**Windowsの運用管理を快適にする10の裏ワザ／表ワザ**　『4.タスクスケジューラの使いこなし』を参照してください）、毎月1回、あるいは数か月に1回といったサイクルで自動実行すればよいでしょう。スクリプトの作成者によると、週次または月次での実行を薦めています。

```
.¥WSUS-Cleanup.ps1 ↲
```

なお、リモートのSQL Serverインスタンスを使用している場合はWSUS-Clean.ps1の$SqlServerにコンピューター名を設定する必要があります。



画面26　WSUS-Cleanup.ps1を利用したWSUSサーバーのメンテナンス

データベースのインデックスの再構築のためには、SQLCMDユーティリティが利用可能であることが必要です。WIDを使用している場合は、以下のURLからSQLCMDユーティリティ（本稿執筆時点のバージョンはMicrosoft Command Utilites 15 for SQL Server）をダウンロードしてWSUSサーバーにインストールしてから実行してください。

sqlcmd ユーティリティ
https://docs.microsoft.com/ja-jp/sql/tools/sqlcmd-utility

SQL Serverを使用している場合、スクリプトの問題によりインデックスの再構築でエラーが発生し、スキップされるかもしれません。筆者が使用したスクリプトのバージョン（Last updated 11-26-2019、Version 4）の場合は、WSUS-Cleanup.ps1のfunction RebuildDBIndexes{} 内のコードを以下のように変更することでローカルのWIDとSQL Serverの両方に対応させることができます。なお、WIDを使用している場合は変更の必要はありません。

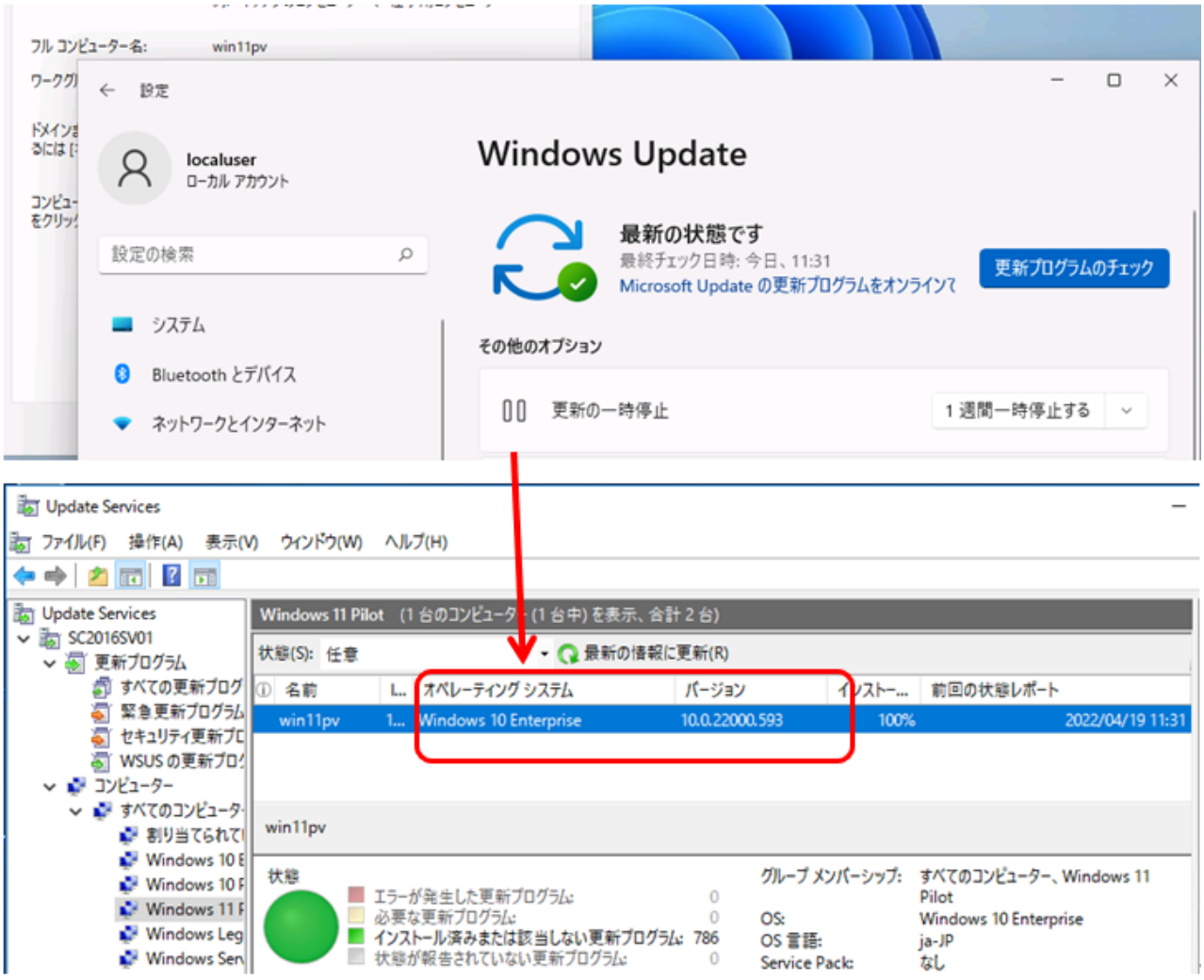
変更前	$status = SQLCMD -S ¥¥.¥pipe¥Microsoft##WID¥tsql¥query -i $SQLPath -I ↲
変更後	if ($script:SqlServer -match "microsoft##"){ ↲ 　$status = SQLCMD -S ¥¥.¥pipe¥Microsoft##WID¥tsql¥query -i $SQLPath -I ↲ } else { ↲ 　$status = SQLCMD -S $script:SqlServer -i $SQLPath -I ↲ } ↲

WSUSとWindows11

WSUSはWindows Server 2016でWindows 10に対応して以降（Windows 10対応はWindows Server 2012/2012 R2 WSUSにも更新プログラムでバックポートされました）、実質、変更されていません。製品の分類でWindows 11を選択するだけで、Windows 11のパッチ管理に対応できます。

ただし、WSUS では、検出したWindows 11クライアントの「オペレーティングシステム」列の情報を「Windows 10」と表示することに注意してください。これは、WSUSがクライアントの「HKLM¥SOFTWARE¥Microsoft¥Windows NT¥CurrentVersion¥ProductName」レジストリ値から取得した情報を基に、「オペレーティングシステム」列の情報を表示しているからです。これは表示上の問題で、インストール済みのWindows 11向け更新プログラムの状態やインストール対象としてのWindows 11向け更新プログラムの承認には影響しません。

似たような問題に、Windows 10やWindows 11の実際の詳細なビルド番号とWSUSのコンソールで確認できるクライアントの「バージョン」列の情報が一致しないことがあります。WSUSは、ここにWindowsのバージョンではなく、Windows Update Agentコンポーネント（wuaueng.dll）のファイルバージョン情報を表示する仕様だからです。



10.マルウェア対策の最適化

Windows 10およびWindows Server 2016以降には、Microsoft Defenderウイルス対策（旧称、Windows Defender）が標準搭載されており、サードベンダーのウイルス対策製品がインストールされていない限り既定で有効です。サードベンダーのウイルス対策製品がインストールされている場合は、自動的に無効化されます。

マルウェア対策としてMicrosoft Defenderウイルス対策を利用している場合、リアルタイムスキャンが有効であること、エンジンと定義ファイル（セキュリティインテリジェンス更新プログラム）が最新版に更新されていること、および定期的にスキャンが実行されていることが高いセキュリティを維持する上で重要です。

その上で、特にサーバーのパフォーマンスを最適化するためには、適切な除外設定を行うことが重要です。除外設定なしで運用する場合、パフォーマンスが劣化したり、サービスやアプリケーションでエラーが発生して正常に動作しなくなったりすることがあります。ここではMicrosoft Defenderウイルス対策の除外設定の具体的な方法、およびHyper-Vの役割とWindowsコンテナー用コンテナーホストのための除外設定について説明します。

Microsoft Defenderの自動除外

Windows Server 2016以降のMicrosoft Defenderウイルス対策では、自動除外（Automatic Exclusions）という機能を備えており、既定で有効になっています。この機能はWindows 10にもあり既定で有効ですが、特にさまざまな役割やサーバーアプリケーションを実行するWindows Serverでこの機能が有効になっていることが重要です。

Microsoft Defenderウイルス対策の自動除外が有効になっているかどうかは、PowerShellの以下のコマンドラインの実行結果がFalseになっているかどうかで分かります。

> (Get-MpPreference).DisableAutoExclusions ↲
>
> False ･･･ 自動除外が有効

Windows Serverで自動除外が有効になっている場合、有効になっているサーバーの役割に応じて定義済みの除外設定が適用されます。

具体的にはWindows Updateのデータベースやログファイル、Windowsのセキュリティデータベース、グループポリシー関連ファイル、WINSデータベース、ファイルレプリケーションサービス（FRS）データベース、分散ファイルシステムレプリケーション（DFSR）のデータベースとプロセス、Hyper-V関連のファイルの種類、ディレクトリ、およびプロセス、Active DirectoryのNTDSデータベース、SYSVOL共有、DHCPサーバーのデータベース、DNSサーバーのデータベース、クラスター共有ストレージ、プリンタースプーラーのディレクトリおよびプロセス、IIS Webサーバーのディレクトリとプロセス、Windows Server Update Services（WSUS）のコンテンツとデータベースなどが除外されます。

詳しくは、以下のドキュメントで説明されています。

Windows Server で Microsoft Defender ウイルス対策の除外を構成する
https://docs.microsoft.com/ja-jp/windows/security/threat-protection/microsoft-defender-antivirus/configure-server-exclusions-microsoft-defender-antivirus

例えば、Hyper-Vの役割が有効になっている場合、表1に示す除外設定が適用されます。

表1　Hyper-Vの役割の自動除外設定

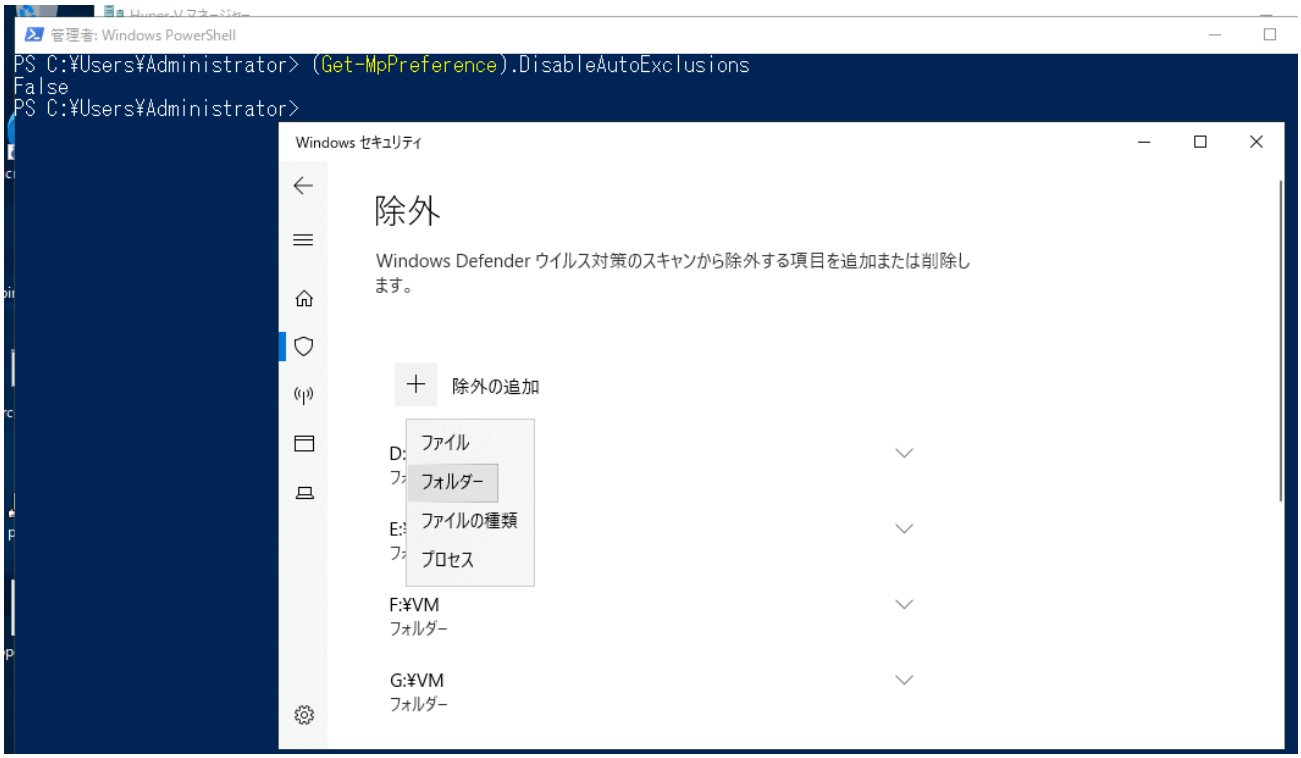
ファイルの種類の除外	.vhd、.vhdx、.avhd、.avhdx、.iso、.rct、.vsv、.vmcx、.vmrs
ディレクトリの除外	%ProgramData%¥Microsoft¥Windows¥Hyper-V %ProgramFiles%¥Hyper-V %Public%¥Documents¥Hyper-V¥Virtual Hard Disks %SystemDrive%¥ProgramData¥Microsoft¥Windows¥Hyper-V¥Snapshots
プロセスの除外	%systemroot%¥System32¥Vmms.exe（Virtual Machine Management Service） %systemroot%¥System32¥Vmwp.exe（Virtual Machine Worker Process）

Microsoft Defenderの自動除外＋カスタム設定

自動除外設定でカバーされない場所に除外するべきデータベースやファイルを配置している場合は、自動除外に加えて、カスタム設定を追加することを推奨します。また、自動除外設定の対象外の役割やアプリケーションを利用している場合もカスタム設定を検討してください。除外設定によりパフォーマンスを改善できる場合があります。一方で、除外対象からはマルウェアが検出されなくなるというデメリットにも留意してください。

カスタム除外設定の追加

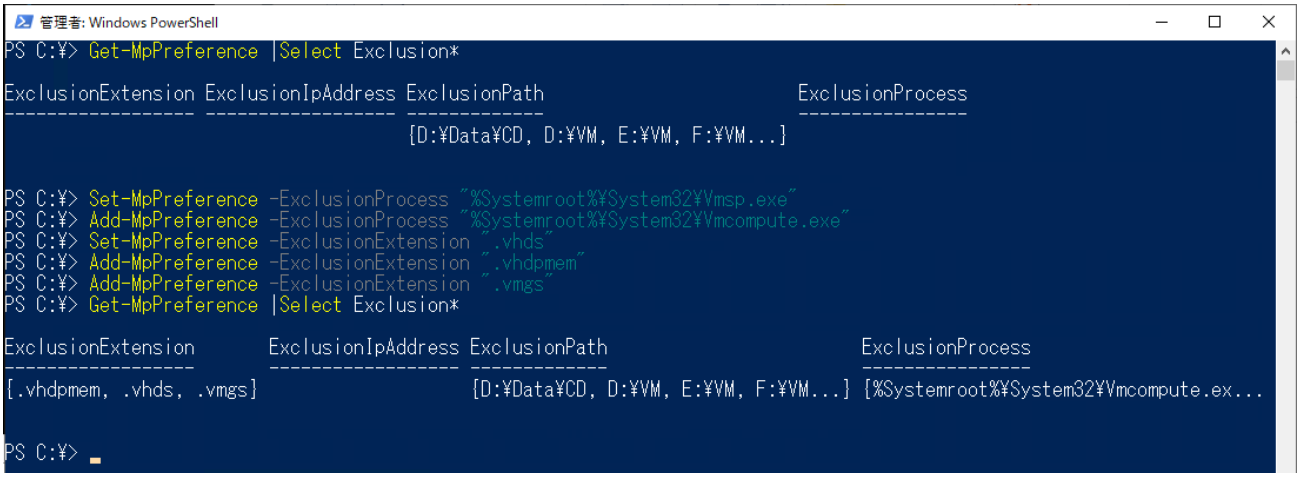
Microsoft Defenderウイルス対策の除外設定は、Windows 10やWindows Server 2016以降のバージョンによって異なります。Windows Server 2016デスクトップエクスペリエンスの場合、［設定］アプリの［更新とセキュリティ］－［Windows Defender］－［除外］から除外を編集します。Windows Server 2019デスクトップエクスペリエンスの場合は、［Windowsセキュリティ］の［ウイルスと脅威の防止］－［ウイルスと脅威の防止の設定］の［設定の管理］から除外設定を編集します（画面7）。



画面7　Windows Server 2019デスクトップエクスペリエンスでのカスタム除外設定

PowerShellのDefenderモジュールのコマンドレットを使用すると、Windowsのバージョンによる設定の場所の違いや、インストールの種類（デスクトップエクスペリエンスとServer Core）に関係なく、一貫性のある方法で除外設定が行えます（画面8）。

> Get-MpPreference |Select Exclusion* ↲
>
> Set-MpPreference -ExclusionExtension ".拡張子" ↲（新たに設定）
>
> Add-MpPreference -ExclusionExtension ".拡張子" ↲（設定を追加）
>
> Remove-MpPreference -ExclusionExtension ".拡張子" ↲（設定を削除）
>
> Set-MpPreference -ExclusionPath "ディレクトリパス" ↲（新たに設定）
>
> Add-MpPreference -ExclusionPath "ディレクトリパス" ↲（設定を追加）
>
> Remove-MpPreference -ExclusionPath "ディレクトリパス" ↲（設定を削除）
>
> Set-MpPreference -ExclusionProcess "実行可能ファイルのパス" ↲（新たに設定）
>
> Add-MpPreference -ExclusionProcess "実行可能ファイルのパス" ↲（設定を追加）
>
> Remove-MpPreference -ExclusionProcess "実行可能ファイルのパス" ↲（設定を削除）



画面8　PowerShellのDefenderモジュールのコマンドレットを使用したカスタム除外設定。なお、自動除外の設定内容は表示されない

Hyper-Vホストのためのカスタム設定

以下の「Hyper-V ホストに推奨されるウイルス対策の除外」のドキュメントの説明によると、自動除外に含まれない以下のファイルの種類、ディレクトリ、およびプロセスを除外することが推奨されています。Hyper-V環境ではホストとゲストの両方でMicrosoft Defenderウイルス対策またはサードベンダーのマルウェア対策製品の保護が働きます。スキャンが重複しないようにホスト側での除外設定は重要です。

- VHDセット（.vhds）（Windows Server 2016以降）
- VMゲスト状態ファイル（.vmgs）（Windows Serverバージョン1709以降の仮想マシン構成バージョン8.2以降）
- メモリ状態ファイル（.bin）（Windows Server 2012 R2の古い仮想マシン構成バージョン5.0）
- 仮想マシン構成ファイルが配置されているカスタムディレクトリ
- 仮想ハードディスクファイルが配置されているカスタムディレクトリ
- Hyper-Vレプリカを使用している場合はレプリケーションデータのディレクトリ
- SMB 3.0共有に仮想マシンを配置している場合は、共有しているディレクトリ（ファイルサーバー側のウイルス対策で除外）
- %SystemRoot%¥System32¥Vmsp.exe（Virtual Machine Security Process、Windows Server 2016以降）
- %SystemRoot%¥System32¥Vmcompute.exe（Hyper-V Host Compute Service、Windows Server 2019以降）

Hyper-V ホストに推奨されるウイルス対策の除外
https://docs.microsoft.com/ja-jp/troubleshoot/windows-server/virtualization/antivirus-exclusions-for-hyper-v-hosts

Windowsコンテナーのためのカスタム設定

Windows Server 2016以降の新しい機能である、Mirantis Container Runtime（旧称、Docker Engterprise）対応のWindowsコンテナーに関しては自動除外の対象として明記されていません。

Microsoft Defenderウイルス対策は、Windows Serverベースのコンテナーホストの保護をサポートしますが、Windowsコンテナー内ではMicrosoft Defenderウイルス対策はサポートされません。WindowsコンテナーのベースOSイメージであるservercoreおよびnanoserverにはMicrosoft Defenderウイルス対策のバイナリ（C:¥Program Files¥Windows Defender）やサービス（WinDefend）は含まれません。言い換えると、Windowsコンテナー内の実行環境でMicrosoft Defenderによる保護は提供できないため、コンテナーのイメージやプロセスをコンテナーホスト側で保護することが重要です。

Microsoft Defenderを対象としたものではありませんが、コンテナーホストにおけるマルウェア対策の最適化については、マイクロソフトおよびDockerのドキュメントで説明されています。

Windows コンテナー用のウイルス対策最適化
https://docs.microsoft.com/ja-jp/windows-hardware/drivers/ifs/anti-virus-optimization-for-windows-containers

Antivirus software and Docker
https://docs.docker.com/engine/security/antivirus/

Dockerドキュメントによると、Dockerが使用中のファイルをマルウェア対策ソフトウェアがスキャンすると、対象のファイルがロックされることにより、dockerコマンドのハングが発生する可能性があるそうです。この問題を緩和するには、Dockerのデータディレクトリ（Windows Serverベースのコンテナーホストの場合の既定はC:¥ProgramData¥docker）を保護の対象から除外します。しかし、除外により、イメージ内にウイルスやマルウェアが含まれる場合やコンテナーボリュームに書き込まれた場合、それを検出できません。なお、除外設定なしでも通常は問題なく機能しますが、スキャンの負荷がコンテナーの開始と実行に影響する可能性はあります。

Dockerのデータディレクトリ（既定のパスまたはdaemon.jsonのgraphパスで指定されたカスタムパス）を保護から除外する場合は、Dockerサービスの停止、除外の解除、データディレクトリのスキャン、除外の再設定、Dockerサービスの起動を定期的なタスクをスケジューリングするとよいでしょう。Microsoft Defenderの場合は、PowerShellの次の一連のコマンドラインをタスクとして実行することで実現できます。PowerShellコマンドやPowerShellスクリプト（.ps1）のタスク登録については、**Windowsの運用管理を快適にする10の裏ワザ／表ワザ**　『4.タスクスケジューラの使いこなし』で説明しました。

```
Stop-Service -Name Docker ↲

Remove-MpPreference -ExclusionPath "C:¥ProgramData¥docker" ↲

Start-MpScan -ScanPath "C:¥ProgramData¥docker" -ScanType CustomScan ↲

Add-MpPreference -ExclusionPath "C:¥ProgramData¥docker" ↲

Start-Service -Name Docker ↲
```

SQL Serverのためのカスタム設定

SQL Serverを実行しているコンピューターのマルウェア対策の一般的なガイドラインは以下のドキュメントに記されています。

How to choose antivirus software to run on computers that are running SQL Server
https://support.microsoft.com/en-us/topic/feda079b-3e24-186b-945a-3051f6f3a95b

ガイドラインでは、SQL Serverのパフォーマンスの改善に寄与するディレクトリ、拡張子、プロセスの除外設定が詳細に記されています。適切な除外設定を行うことで、SQL Serverのサービスが関連ファイルを使用する際にスキャンツールがロックするのを回避できます。ただし、除外設定したファイルがマルウェアに感染した場合、マルウェア対策ソフトウェアはそれを検出できないというデメリットがあります。

SQL Server用に除外設定を行う場合は、Windowsコンテナーの場合と同様に、業務時間外などにSQL Serverのサービスを停止した上で、除外設定を解除してカスタムスキャンを実施することをお勧めします。

他社マルウェア対策製品を利用する場合の考慮事項

Windows Serverでサードベンダーのマルウェア対策製品を利用する場合は、Microsoft Defenderウイルス対策の除外設定、およびHyper-Vホストに推奨されるウイルス対策の除外のドキュメントを参考に、サーバーの役割に適切なすべての除外設定をカスタム設定で実装してください。

Microsoft Defenderオフラインの注意事項

Windows 10とWindows 11のMicrosoft Defenderマルウェア対策には、スキャンオプションの1つとしてMicrosoft Defenderオフラインがあります。この機能は、WinPEベースで動作し、オフラインのWindowsのファイルシステムをスキャンして駆除することで、オンライン中にスキャンエンジンから身を隠している（またはスキャンエンジンが既に侵害されていて信頼できない可能性がある）場合に、脅威を取り除くことができるというものです。ただし、この機能、スキャンを実行し、検出や駆除が終了すると自動的に再起動してしまい、結果をレポートしてくれないという問題があります（本来であれば保護の履歴に残るはずですが機能していない可能性があります）。また、Windows 11では検出しても駆除してくれないという不具合も確認しています。詳しくは、以下の記事をご覧ください。

疑惑を検証！ Windows 11の「Microsoft Defenderオフライン」は“いざ”というときに役に立つのか？
https://atmarkit.itmedia.co.jp/ait/articles/2204/08/news004.html

セイ・テクノロジーズからのお知らせ

「Windowsの運用管理を快適にする10の裏ワザ／表ワザ」を活用したシステム安定運用テンプレートのご紹介

本書のなかで山市氏はWindowsの運用管理を快適にするためには、

- バージョン番号やビルド番号の把握や脆弱性への対応や既存環境への適用計画といった戦略（テクニック1）
- Windows Updateを戦略的に行い、Windows Updateの適用状況を正確に把握すること（テクニック2）
- Windowsでは自動実行するメンテナンスタスクの該当するタスクを実行することで、パフォーマンスを向上させること（テクニック3）

などが必要だと述べられています。自立分散型サーバー監視ソフト『BOM for Windows Ver.8.0』を活用することで、これらの対策を簡単に設定することができます。
『BOM for Windows Ver.8.0』には、簡単に監視をはじめられる監視テンプレートが同梱されており、その中には『Windowsの運用管理を快適にする10の裏ワザ/表ワザ』に沿った「システム安定運用テンプレート」を2つご用意しております。

1. 「システム安定運用-パフォーマンス改善テンプレート」

ディスククリーンアップタスク実行・NGEN監視・FireWall監視・Microsoft Defenderの除外設定監視などのパフォーマンスの改善に関する項目が含まれています。

2.「システム安定運用-セキュリティテンプレート」

Windows バージョン情報取得・Windows Update 未適用リスト取得監視・Windows Update成否リスト取得などセキュリティに関する項目が含まれています。

本書に沿った内容の『BOM for Windows Ver.8.0』のシステム安定運用テンプレートの詳細説明は山市どっとこむ内の記事で掲載していますので、下記URLからご覧ください。
https://www2.say-tech.co.jp/special/ryo-yamaichi/template

自立分散型サーバー監視ソフト『BOM for Windows Ver.8.0』

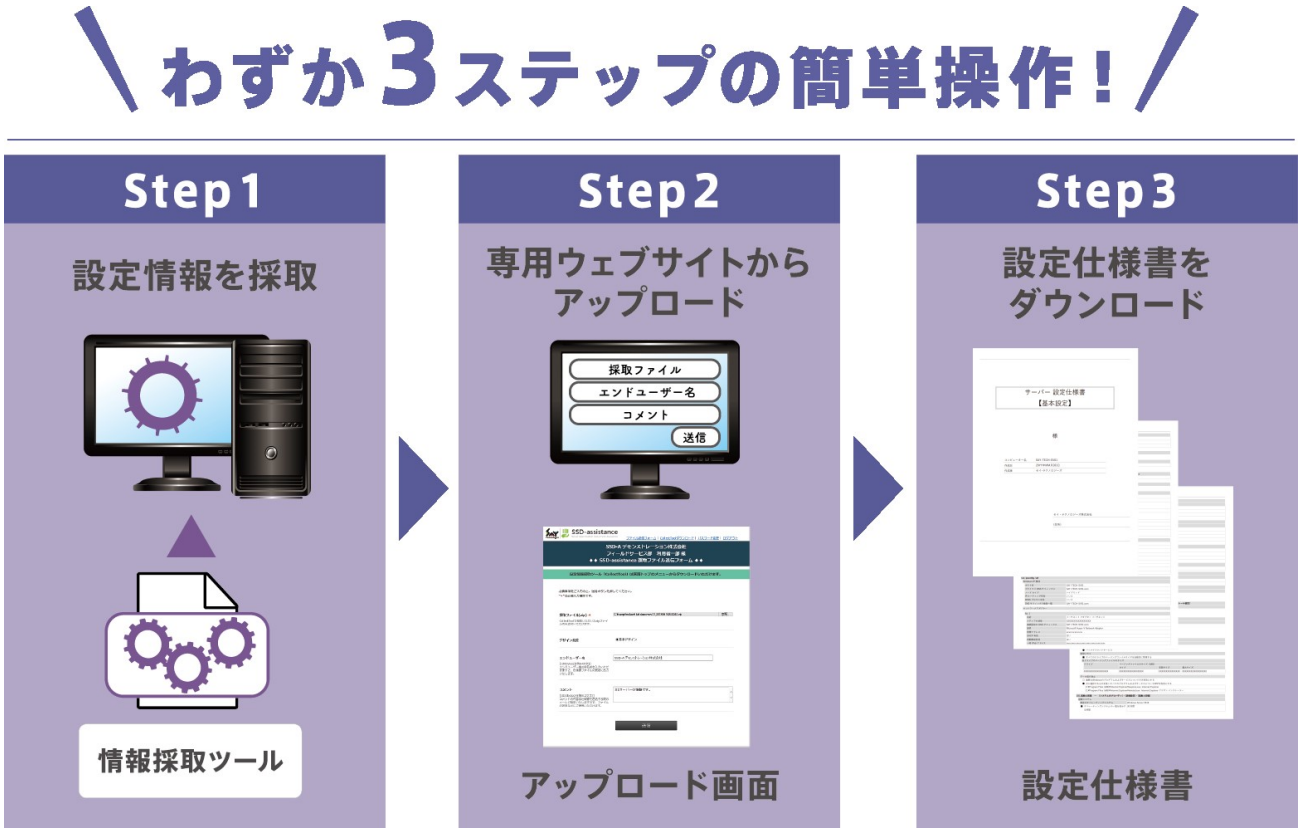
自立分散型サーバー監視ソフト『BOM for Windows Ver.8.0』製品詳細
https://www.say-tech.co.jp/product/bomwin80

サーバー設定仕様書自動生成サービス『SSD-assistance』

テクニック5のテーマである現在のセキュリティ設定と推奨基準を比較する時や、テクニック9のテーマであるWSUSの設定を把握したい時には、サーバー設定仕様書自動生成サービス「SSD-assistance」をご活用ください。

サーバー設定仕様書自動生成サービス「SSD-assistance」は、システムの納品、運用保守、リプレースといった一連の業務に欠かすことのできない設定仕様書を、たった3ステップで自動生成するクラウドサービスです。SSD-assistanceを利用することにより、設定仕様書の作成/更新に伴う課題を解決することができます。

異なる2つの環境を比較することはもちろん、同じ環境で採取した異なるタイミングの変更前後の差分がひと目で把握できる差分比較仕様書やWSUSに特化した専用のWSUS設定仕様書、その他OSの様々な設定情報を簡単にドキュメント化できます。



サーバー設定仕様書自動生成サービス『SSD-assistance』製品詳細
https://www.say-tech.co.jp/product/ssda

山市良どっとこむとは？

Windowsに関する数多くの著作や連載記事等でご活躍されており、本ホワイトペーパーの著者でもある山市 良氏(2008年からMicrosoft Most Valuable Professionalを連続受賞)によるお役立ちコンテンツをご紹介するページです。
本ホワイトペーパー以外にも、山市氏のコンテンツを活用したテクニックもご紹介しており、Windowsに関わるシステム管理者や運用者に役立つ情報をお届けします
https://www2.say-tech.co.jp/special/ryo-yamaichi

お問い合わせ

本書に関するお問い合わせは、以下のフォームをご利用ください。

https://www.say-tech.co.jp/contact

初版：2021年 7月 9日

更新：2023年 8月28日

作成：セイ・テクノロジーズ株式会社