Citrix Workspace Environment Management 1912 SDK

Contents

Citrix Workspace Environment Management 1912 SDK

May 25, 2023

Citrix Workspace Environment Management uses intelligent resource management and Profile Management technologies to deliver the best possible performance, desktop log on, and application response times for XenApp and XenDesktop deployments.

The following SDKs are provided to let you customize and manage various aspects of Workspace Environment Management:

Workspace Environment Management SDK PowerShell Modules

Citrix Workspace Environment Management provides various Microsoft Windows PowerShell versions 3.0. The cmdlets in these modules allow you to create and upgrade Workspace Environment Management databases, and perform tasks on the infrastructure service.

Citrix Workspace Environment Management 1912 SDK

May 25, 2023

Citrix Workspace Environment Management uses intelligent resource management and Profile Management technologies to deliver the best possible performance, desktop log on, and application response times for XenApp and XenDesktop deployments.

The following SDKs are provided to let you customize and manage various aspects of Workspace Environment Management:

Workspace Environment Management SDK PowerShell Modules

Citrix Workspace Environment Management provides various Microsoft Windows PowerShell versions 3.0. The cmdlets in these modules allow you to create and upgrade Workspace Environment Management databases, and perform tasks on the infrastructure service.

Understanding the SDK

May 26, 2023

The Citrix Workspace Environment Management SDK allows you to perform various configuration operations.

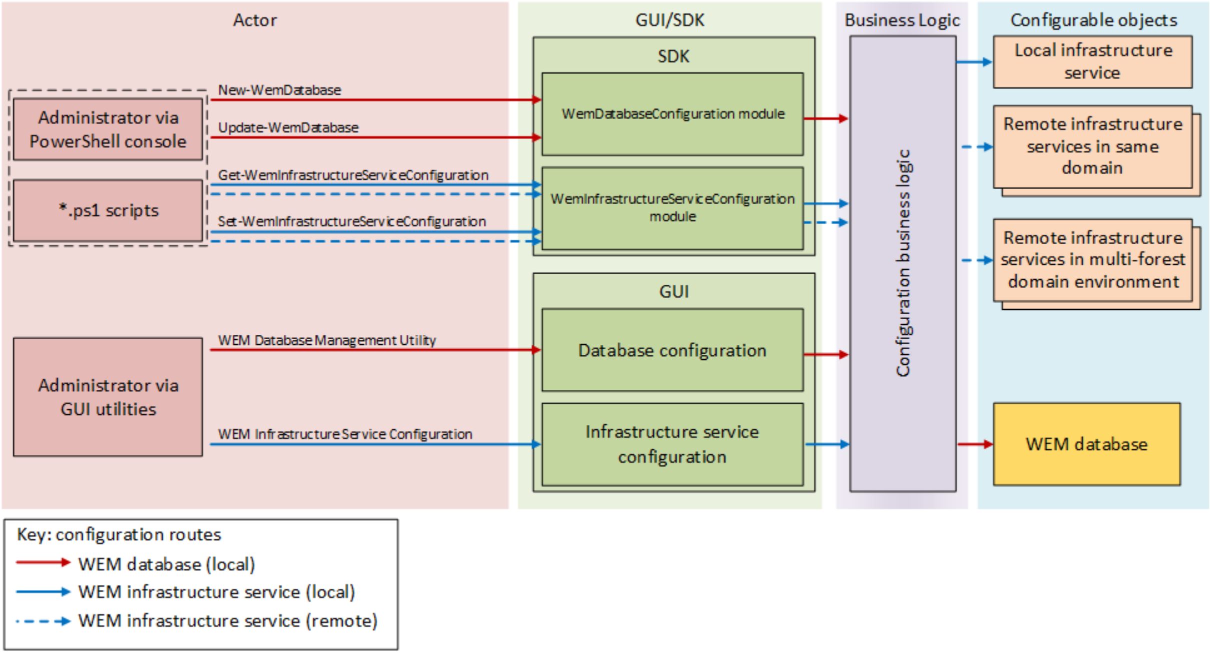
PowerShell modules

The Citrix Workspace Environment Management PowerShell modules allow you to:

- create and upgrade local and remote Workspace Environment Management databases
- perform administrative operations on local and remote infrastructure services, including those in multi-forest domains.

You can use run the provided cmdlets directly from the PowerShell console, or from PowerShell scripts.

The following illustration summarizes how configuring Workspace Environment Management using the PowerShell modules compares with using the standard GUIs.


Actor
GUI/SDK
Business Logic
Configurable objects
SDK
Administrator via PowerShell console
*.ps1 scripts
New-WemDatabase
Update-WemDatabase
Get-WemInfrastructureServiceConfiguration
Set-WemInfrastructureServiceConfiguration
WemDatabaseConfiguration module
WemInfrastructureServiceConfiguration module
GUI
Administrator via GUI utilities
WEM Database Management Utility
WEM Infrastructure Service Configuration
Database configuration
Infrastructure service configuration
Configuration business logic
Local infrastructure service
Remote infrastructure services in same domain
Remote infrastructure services in multi-forest domain environment
WEM database
Key: configuration routes
WEM database (local)
WEM infrastructure service (local)
WEM infrastructure service (remote)

Getting started with the SDK

May 24, 2023

Using the PowerShell modules

Installation

The PowerShell modules are installed automatically when you install the Workspace Environment Management infrastructure services.

Permissions

To use the PowerShell modules, in addition to the privileges required to use the local GUI utilities, you need the following:

- Microsoft Windows PowerShell version 3.0 or later installed on your machine
- Privileges to run PS scripts on the machine
- Access to the target machine
- Privileges for configuring registry, windows services, folders security properties and databases on the target machines. (When these machines are remote, you need these privileges on the remote machine.)

To access and run the cmdlets

Start a shell in PowerShell.

Using scripts

You can write your own scripts and include Workspace Environment Management PowerShell cmdlets.

To use PowerShell cmdlets within scripts, set the execution policy in PowerShell. For more information about PowerShell execution policy, see your Microsoft documentation.

> **Note:**
>
> Filepaths—If you use **New-WemDatabase**, ensure that the filepaths you specify for the DataFilePath and LogFilePath properties are valid and fully formed.

Citrix.WEM.SDK.Configuration.Database

May 23, 2023

Property Commandlets.BaseWemDatabaseCommand‘1.SqlServerCredential PSCredential for connecting to the SQL instance for database creation. Leave empty to use Windows Authentication for current user.

Property Commandlets.BaseWemDatabaseCommand‘1.DatabaseServerInstance SQL Server on which the database will be hosted. (serveraddress,port\instancename).

Property Commandlets.BaseWemDatabaseCommand‘1.DatabaseName Name of the WEM database to create.

Property Commandlets.BaseWemDatabaseCommand‘1.PSDebugMode Debug mode displays extra exception information. Specify ‘None’to leave the current value unchanged. This is equivalent to omitting this parameter.

Type Commandlets.NewWemDatabase

Create a WEM database.

The New-WemDatabase cmdlet creates one Workspace Environment Management (WEM) database. The database is created on the SQL Server.

Example: code

```
$passwd = ConvertTo-SecureString "[Password]" -AsPlainText -Force
$sqlServerCred = New-Object System.Management.Automation.PSCredential("sa", $passwd);
$DBname = "WEM_DB";
New-WemDatabase -DatabaseServerInstance "10.10.10.10" -DatabaseName $DBname -DefaultAdministratorsGroup "[Domain]\[GroupName]" -SqlServerCredential $sqlServerCred
```

Create a database instance on the remote SQL Server (10.10.10.10) by using SQL Server authentication.

Example: code

```
$DBname = "WEM_DB";
$fileFolder = "C:\Program Files\Microsoft SQL DatabaseServerInstance\MSSQL11.MSSQLSERVER\MSSQL\DATA\";
New-WemDatabase -DatabaseServerInstance "[Server\Instance]" -DatabaseName $DBname -DataFilePath($fileFolder+$DBname+"_Data.mdf") -LogFilePath($fileFolder+$DBname+"_Log.ldf") -DefaultAdministratorsGroup "[Domain]\[GroupName]"
```

Create a database instance on the remote SQL Server (10.10.10.10) by usingWindows authentication.

Example: code

```
$DBname = "WEM_DB";
$fileFolder = "C:\Program Files\Microsoft SQL DatabaseServerInstance\MSSQL11.MSSQLSERVER\MSSQL\DATA\";
New-WemDatabase -DatabaseServerInstance "[Server\Instance]" -DatabaseName $DBname -DataFilePath($fileFolder+$DBname+"_Data.mdf") -LogFilePath($fileFolder+$DBname+"_Log.ldf") -DefaultAdministratorsGroup "[Domain]\[GroupName]" -WindowsAccount "[Domain]\[UserName]
```

Create a database instance on the remote SQL Server (10.10.10.10) by using Windows authentication. You can add extra database users by using the “WindowsAccount”attribute.

Example: code

```
$fileFolder = "C:\Program Files\Microsoft SQL DatabaseServerInstance\MSSQL11.MSSQLSERVER\MSSQL\DATA\";
$DBname = "WEMDB_1_Obj";
$cfg = New-Object Citrix.WEM.SDK.Configuration.Database.SDKNewDatabaseConfiguration;
$cfg.DatabaseServerInstance = "[Server\Instance]";
$cfg.DatabaseName = $DBname;
$cfg.DataFilePath = ($fileFolder+$DBname+"_Data.mdf");
$cfg.LogFilePath =  ($fileFolder+$DBname+"_Log.ldf") ;
$cfg.DefaultAdministratorsGroup = "[Domain]\[GroupName]";
$cfg.WindowsAccount = "[Domain]\[UserName]";
New-WemDatabase -Configuration $cfg;
```

Create a new database instance on the remote SQL Server (10.10.10.10) by using a single configuration object to connect to the server and to configure the database.

Update-WemDatabase

Property Commandlets.NewWemDatabase.WindowsAccount Windows account granted access to WEM database.

Property Commandlets.NewWemDatabase.DataFilePath Path to the .mdf file location on the SQL Server. Leave empty to auto-populate the Data file field with the correct path of the SQL being used. If auto-population fails, the path of the SQL Server 2008 R2 is used by default.

Property Commandlets.NewWemDatabase.LogFilePath Path to the .ldf file location on the SQL Server. Leave empty to auto-populate the Log file field with the correct path of the SQL being used. If auto-population fails, the path of the SQL Server 2008 R2 is used by default.

Property Commandlets.NewWemDatabase.DefaultAdministratorsGroup Default group of WEM administrators with Full Access to the administration console.

Property Commandlets.NewWemDatabase.VuemUserSqlPassword Specific password for the WEM vuemUser SQL user account. Leave empty to create a default password.

Property Commandlets.NewWemDatabase.CommandTimeout Timeout period for connection attempts to the WEM database. After this time an error message is displayed. Leave empty to use default timeout of 300 seconds.

Property Commandlets.NewWemDatabase.Configuration Configuration set to save settings in.

Type Commandlets.NewWemDatabaseOnCloud

Create a WEM database on Cloud.

The New-WemDatabaseOnCloud cmdlet creates one Workspace Environment Management (WEM) database. The database is created on the SQL server.

Example: code

```
$passwd = ConvertTo-SecureString "[Password]" -AsPlainText -Force
$sqlServerCred = New-Object System.Management.Automation.PSCredential("sa", $passwd);
$DBname = "WEM_DB";
$elasticPool = "elasticPoolName";
New-WemDatabase -DatabaseServerInstance "10.111.12.145" -DatabaseName $DBname -SqlServerCredential $sqlServerCred -ElasticPool $elasticPool
```

Create a new database instance on the remote SQL Server (10.111.12.145) by using SQL Server authentication.

Example: code

```
$elasticPool = "elasticPoolName";
$DBname = "WEMDB_1_Obj";
$sqlServerCred = New-Object System.Management.Automation.PSCredential("name", $passwd);
$cfg = New-Object Citrix.WEM.SDK.Configuration.Database.SDKNewDatabaseConfigurationOnCloud;
$cfg.DatabaseServerInstance = "[Server\Instance]";
$cfg.SqlServerCredential = $$sqlServerCred;
$cfg.DatabaseName = $DBname;
$cfg.ElasticPool = $elasticPool;

New-WemDatabaseOnCloud -Configuration $cfg;
```

Create a new database instance on the remote SQL Server (10.111.12.145) by using a single configuration object to connect to the server and to configure the database.

Property Commandlets.NewWemDatabaseOnCloud.ElasticPool Elastic Pool of the WEM database to create.

Property Commandlets.NewWemDatabaseOnCloud.VuemUserSqlPassword Specific password for the WEM vuemUser SQL user account. Leave empty to create a default password.

Property Commandlets.NewWemDatabaseOnCloud.Configuration Configuration set save settings in.

Type Commandlets.UpdateWemDatabase

Update an existing WEM database.

The Update-WemDatabase cmdlet updates an existing Workspace Environment Management (WEM) database instance on the SQL server.

Example: code

```
Update-WemDatabase -DatabaseServerInstance "NK_SQL" -DatabaseName "WEM_DB"
```

Update an existing database to the latest version by using Windows authentication.

Example: code

```
    $password = ConvertTo-SecureString "[Password]" -AsPlainText -Force;
    $sqlServerCred = New-Object System.Management.Automation.PSCredential("sa", $password);
    Update-WemDatabase -DatabaseServerInstance "NK_SQL" -DatabaseName "WEM_DB" -SqlServerCredential $sqlServerCred;
```

Update an existing database to the latest version by using SQL Server authentication.

Example: code

```
    $password = ConvertTo-SecureString "[Password]" -AsPlainText -Force;
    $sqlServerCred = New-Object System.Management.Automation.PSCredential("sa", $password);
    Update-WemDatabase -DatabaseServerInstance "NK_SQL" -DatabaseName "WEM_DB" -SqlServerCredential $sqlServerCred -WindowsAccount "[Domain]\[UserName]";
```

Update an existing database to the latest version by using SQL Server authentication. You can add extra database users by using the “WindowsAccount”attribute.

Example: code

```
$cfg_obj = New-Object Citrix.WEM.SDK.Configuration.Database.SDKDatabaseConfiguration
$cfg_obj.DatabaseServerInstance = "10.10.10.10";
$cfg_obj.DatabaseName = "WEM_DB";
$cfg_obj.WindowsAccount = "[Domain]\[UserName]";
Update-WemDatabase -Configuration $cfg_obj;
```

Update an existing database instance on the remote SQL Server (10.10.10.10) by using a single configuration object to connect to the server and to configure the database.

New-WemDatabase

Property Commandlets.UpdateWemDatabase.WindowsAccount Windows account granted access to WEM database.

Property Commandlets.UpdateWemDatabase.Configuration Configuration set.

Type SDKDatabaseConfiguration

SDK Database Configuration object.

Property SDKDatabaseConfiguration.SqlServerCredential PSCredential for connecting to the SQL instance for database creation. Leave empty to use Windows Authentication for current user.

Property SDKDatabaseConfiguration.DatabaseServerInstance SQL Server on which the database will be hosted (serveraddress,port\instancename).

Property SDKDatabaseConfiguration.DatabaseName Name of the WEM database to create or update.

Type SDKNewDatabaseConfiguration

SDK new database Configuration object.

Property SDKNewDatabaseConfiguration.WindowsAccount Windows account granted access to WEM database.

Property SDKNewDatabaseConfiguration.DataFilePath Path to the .mdf file location on the SQL Server. Leave empty to auto-populate the Data file field with the correct path of the SQL being used. If auto-population fails, the path of the SQL Server 2008 R2 is used by default.

Property SDKNewDatabaseConfiguration.LogFilePath Path to the .ldf file location on the SQL Server. Leave empty to auto-populate the Log file field with the correct path of the SQL being used. If auto-population fails, the path of the SQL Server 2008 R2 is used by default.

Property SDKNewDatabaseConfiguration.DefaultAdministratorsGroup Default group of WEM administrators with Full Access to the Administration Console.

Property SDKNewDatabaseConfiguration.VuemUserSqlPassword Specific password for the WEM vuemUser SQL user account. Leave empty to create a default password.

Property SDKNewDatabaseConfiguration.CommandTimeout Timeout period for connection attempts to the WEM database. After this time an error message is displayed. Leave empty to use default timeout of 300 seconds.

Type SDKNewDatabaseConfigurationOnCloud

Configuration new database class

Property SDKNewDatabaseConfigurationOnCloud.VuemUserSqlPassword Specific password for the WEM vuemUser SQL user account. Leave empty to create a default password.

Property SDKNewDatabaseConfigurationOnCloud.ElasticPool ELASTIC POOL of the WEM database to join.

Type SDKUpdateDatabaseConfiguration

Configuration update database class

Property SDKUpdateDatabaseConfiguration.WindowsAccount Windows account granted access to WEM database.

Citrix.WEM.SDK.Configuration.Database

May 23, 2023

Property Commandlets.BaseWemDatabaseCommand‘1.SqlServerCredential PSCredential for connecting to the SQL instance for database creation. Leave empty to use Windows Authentication for current user.

Property Commandlets.BaseWemDatabaseCommand‘1.DatabaseServerInstance SQL Server on which the database will be hosted. (serveraddress,port\instancename).

Property Commandlets.BaseWemDatabaseCommand‘1.DatabaseName Name of the WEM database to create.

Property Commandlets.BaseWemDatabaseCommand‘1.PSDebugMode Debug mode displays extra exception information. Specify ‘None’to leave the current value unchanged. This is equivalent to omitting this parameter.

Type Commandlets.NewWemDatabase

Create a WEM database.

The New-WemDatabase cmdlet creates one Workspace Environment Management (WEM) database. The database is created on the SQL Server.

Example: code

```
     $passwd = ConvertTo-SecureString "[Password]" -AsPlainText -Force
     $sqlServerCred = New-Object System.Management.Automation.
        PSCredential("sa", $passwd);
     $DBname = "WEM_DB";
     New-WemDatabase -DatabaseServerInstance "10.10.10.10" -DatabaseName
         $DBname -DefaultAdministratorsGroup "[Domain]\[GroupName]" -
        SqlServerCredential $sqlServerCred
```

Create a database instance on the remote SQL Server (10.10.10.10) by using SQL Server authentication.

Example: code

```
     $DBname = "WEM_DB";
     $fileFolder = "C:\Program Files\Microsoft SQL
        DatabaseServerInstance\MSSQL11.MSSQLSERVER\MSSQL\DATA\";
     New-WemDatabase -DatabaseServerInstance "[Server\Instance]" -
        DatabaseName $DBname -DataFilePath($fileFolder+$DBname+"_Data.
        mdf") -LogFilePath($fileFolder+$DBname+"_Log.ldf") -
        DefaultAdministratorsGroup "[Domain]\[GroupName]"
```

Create a database instance on the remote SQL Server (10.10.10.10) by usingWindows authentication.

Example: code

```
     $DBname = "WEM_DB";
     $fileFolder = "C:\Program Files\Microsoft SQL
        DatabaseServerInstance\MSSQL11.MSSQLSERVER\MSSQL\DATA\";
     New-WemDatabase -DatabaseServerInstance "[Server\Instance]" -
        DatabaseName $DBname -DataFilePath($fileFolder+$DBname+"_Data.
        mdf") -LogFilePath($fileFolder+$DBname+"_Log.ldf") -
        DefaultAdministratorsGroup "[Domain]\[GroupName]" -
        WindowsAccount "[Domain]\[UserName]
```

Create a database instance on the remote SQL Server (10.10.10.10) by using Windows authentication. You can add extra database users by using the “WindowsAccount”attribute.

Example: code

```
     $fileFolder = "C:\Program Files\Microsoft SQL
        DatabaseServerInstance\MSSQL11.MSSQLSERVER\MSSQL\DATA\";
     $DBname = "WEMDB_1_Obj";
     $cfg = New-Object Citrix.WEM.SDK.Configuration.Database.
        SDKNewDatabaseConfiguration;
     $cfg.DatabaseServerInstance = "[Server\Instance]";
```

```
    $cfg.DatabaseName = $DBname;
    $cfg.DataFilePath = ($fileFolder+$DBname+"_Data.mdf");
    $cfg.LogFilePath =  ($fileFolder+$DBname+"_Log.ldf") ;
    $cfg.DefaultAdministratorsGroup = "[Domain]\[GroupName]";
    $cfg.WindowsAccount = "[Domain]\[UserName]";
    New-WemDatabase -Configuration $cfg;
```

Create a new database instance on the remote SQL Server (10.10.10.10) by using a single configuration object to connect to the server and to configure the database.

Update-WemDatabase

Property Commandlets.NewWemDatabase.WindowsAccount Windows account granted access to WEM database.

Property Commandlets.NewWemDatabase.DataFilePath Path to the .mdf file location on the SQL Server. Leave empty to auto-populate the Data file field with the correct path of the SQL being used. If auto-population fails, the path of the SQL Server 2008 R2 is used by default.

Property Commandlets.NewWemDatabase.LogFilePath Path to the .ldf file location on the SQL Server. Leave empty to auto-populate the Log file field with the correct path of the SQL being used. If auto-population fails, the path of the SQL Server 2008 R2 is used by default.

Property Commandlets.NewWemDatabase.DefaultAdministratorsGroup Default group of WEM administrators with Full Access to the administration console.

Property Commandlets.NewWemDatabase.VuemUserSqlPassword Specific password for the WEM vuemUser SQL user account. Leave empty to create a default password.

Property Commandlets.NewWemDatabase.CommandTimeout Timeout period for connection attempts to the WEM database. After this time an error message is displayed. Leave empty to use default timeout of 300 seconds.

Property Commandlets.NewWemDatabase.Configuration Configuration set to save settings in.

Type Commandlets.NewWemDatabaseOnCloud

Create a WEM database on Cloud.

The New-WemDatabaseOnCloud cmdlet creates one Workspace Environment Management (WEM) database. The database is created on the SQL server.

Example: code

```
$passwd = ConvertTo-SecureString "[Password]" -AsPlainText -Force
$sqlServerCred = New-Object System.Management.Automation.
   PSCredential("sa", $passwd);
$DBname = "WEM_DB";
$elasticPool = "elasticPoolName";
New-WemDatabase -DatabaseServerInstance "10.111.12.145" -
   DatabaseName $DBname -SqlServerCredential $sqlServerCred -
   ElasticPool $elasticPool
```

Create a new database instance on the remote SQL Server (10.111.12.145) by using SQL Server authentication.

Example: code

```
$elasticPool = "elasticPoolName";
$DBname = "WEMDB_1_Obj";
$sqlServerCred = New-Object System.Management.Automation.
   PSCredential("name", $passwd);
$cfg = New-Object Citrix.WEM.SDK.Configuration.Database.
   SDKNewDatabaseConfigurationOnCloud;
$cfg.DatabaseServerInstance = "[Server\Instance]";
$cfg.SqlServerCredential = $$sqlServerCred;
$cfg.DatabaseName = $DBname;
$cfg.ElasticPool = $elasticPool;

New-WemDatabaseOnCloud -Configuration $cfg;
```

Create a new database instance on the remote SQL Server (10.111.12.145) by using a single configuration object to connect to the server and to configure the database.

Property Commandlets.NewWemDatabaseOnCloud.ElasticPool Elastic Pool of the WEM database to create.

Property Commandlets.NewWemDatabaseOnCloud.VuemUserSqlPassword Specific password for the WEM vuemUser SQL user account. Leave empty to create a default password.

Property Commandlets.NewWemDatabaseOnCloud.Configuration Configuration set save settings in.

Type Commandlets.UpdateWemDatabase

Update an existing WEM database.

The Update-WemDatabase cmdlet updates an existing Workspace Environment Management (WEM) database instance on the SQL server.

Example: code

```
Update-WemDatabase -DatabaseServerInstance "NK_SQL" -DatabaseName "WEM_DB"
```

Update an existing database to the latest version by using Windows authentication.

Example: code

```
    $password = ConvertTo-SecureString "[Password]" -AsPlainText -Force;
    $sqlServerCred = New-Object System.Management.Automation.PSCredential("sa", $password);
    Update-WemDatabase -DatabaseServerInstance "NK_SQL" -DatabaseName "WEM_DB" -SqlServerCredential $sqlServerCred;
```

Update an existing database to the latest version by using SQL Server authentication.

Example: code

```
$password = ConvertTo-SecureString "[Password]" -AsPlainText -Force
    ;
$sqlServerCred = New-Object System.Management.Automation.
    PSCredential("sa", $password);
Update-WemDatabase -DatabaseServerInstance "NK_SQL" -DatabaseName "
    WEM_DB" -SqlServerCredential $sqlServerCred -WindowsAccount "[
    Domain]\[UserName]";
```

Update an existing database to the latest version by using SQL Server authentication. You can add extra database users by using the “WindowsAccount”attribute.

Example: code

```
$cfg_obj = New-Object Citrix.WEM.SDK.Configuration.Database.
    SDKDatabaseConfiguration
$cfg_obj.DatabaseServerInstance = "10.10.10.10";
$cfg_obj.DatabaseName = "WEM_DB";
$cfg_obj.WindowsAccount = "[Domain]\[UserName]";
Update-WemDatabase -Configuration $cfg_obj;
```

Update an existing database instance on the remote SQL Server (10.10.10.10) by using a single configuration object to connect to the server and to configure the database.

New-WemDatabase

Property Commandlets.UpdateWemDatabase.WindowsAccount Windows account granted access to WEM database.

Property Commandlets.UpdateWemDatabase.Configuration Configuration set.

Type SDKDatabaseConfiguration

SDK Database Configuration object.

Property SDKDatabaseConfiguration.SqlServerCredential PSCredential for connecting to the SQL instance for database creation. Leave empty to use Windows Authentication for current user.

Property SDKDatabaseConfiguration.DatabaseServerInstance SQL Server on which the database will be hosted (serveraddress,port\instancename).

Property SDKDatabaseConfiguration.DatabaseName Name of the WEM database to create or update.

Type SDKNewDatabaseConfiguration

SDK new database Configuration object.

Property SDKNewDatabaseConfiguration.WindowsAccount Windows account granted access to WEM database.

Property SDKNewDatabaseConfiguration.DataFilePath Path to the .mdf file location on the SQL Server. Leave empty to auto-populate the Data file field with the correct path of the SQL being used. If auto-population fails, the path of the SQL Server 2008 R2 is used by default.

Property SDKNewDatabaseConfiguration.LogFilePath Path to the .ldf file location on the SQL Server. Leave empty to auto-populate the Log file field with the correct path of the SQL being used. If auto-population fails, the path of the SQL Server 2008 R2 is used by default.

Property SDKNewDatabaseConfiguration.DefaultAdministratorsGroup Default group of WEM administrators with Full Access to the Administration Console.

Property SDKNewDatabaseConfiguration.VuemUserSqlPassword Specific password for the WEM vuemUser SQL user account. Leave empty to create a default password.

Property SDKNewDatabaseConfiguration.CommandTimeout Timeout period for connection attempts to the WEM database. After this time an error message is displayed. Leave empty to use default timeout of 300 seconds.

Type SDKNewDatabaseConfigurationOnCloud

Configuration new database class

Property SDKNewDatabaseConfigurationOnCloud.VuemUserSqlPassword Specific password for the WEM vuemUser SQL user account. Leave empty to create a default password.

Property SDKNewDatabaseConfigurationOnCloud.ElasticPool ELASTIC POOL of the WEM database to join.

Type SDKUpdateDatabaseConfiguration

Configuration update database class

Property SDKUpdateDatabaseConfiguration.WindowsAccount Windows account granted access to WEM database.

Citrix.WEM.SDK.Configuration.InfrastructureService

May 23, 2023

Property Commandlets.BaseInfrastructureServiceConfigurationCommand.InfrastructureServer Remote infrastructure service machine name or IP address.

Property Commandlets.BaseInfrastructureServiceConfigurationCommand.InfrastructureServerCredentia
PSCredential that will be used on the remote machine for getting data.

Property Commandlets.BaseInfrastructureServiceConfigurationCommand.PSDebugMode Enable verbose logging of the infrastructure service. Specify ‘None’to leave the current value unchanged. This is equivalent to omitting this parameter.

Type Commandlets.GetWemInfrastructureServiceConfiguration

Get the current infrastructure service configuration.

The Get-WemInfrastructureServiceConfiguration cmdlet gets the current infrastructure service configuration from the local or remote infrastructure server machine. Remote machines can be either in the same domain, or can be in a multi-forest domain environment.

–To return the current configuration on the local infrastructure server, run the cmdlet without the InfrastructureServer parameter and without the InfrastructureServerCredential parameter. All the following parameter values are applied.

–To return the current configuration from a remote server in the same domain, you must specify the InfrastructureServer parameter.

–To return the current configuration from a remote server in a multi forest Active Directory environment, you must specify the InfrastructureServer parameter (to identify the target machine) and the InfrastructureServerCredential parameter(to provide access credentials).

Example: code

```
Get-WemInfrastructureServiceConfiguration
```

Get the current configuration of the infrastructure service from the local machine.

Example: code

```
Get-WemInfrastructureServiceConfiguration – InfrastructureServer “ [
    Server] ”
```

Get the current configuration of the infrastructure service from the remote machine in the same domain by using Windows authentication.

Example: code

```
    $passwd = ConvertTo-SecureString "[Password]" -AsPlainText -Force;
    $cred = New-Object System.Management.Automation.PSCredential ("[
        Domain\UserName]", $passwd)
    Get-WemInfrastructureServiceConfiguration – InfrastructureServer “
        [Server] ” – InfrastructureServiceAccountCredentials $cred
```

Get the current configuration of the infrastructure service from the remote machine in multi-forest environments. For authentication, this cmdlet uses the PSCredential type object.

Set-WemInfrastructureServiceConfiguration

Type Commandlets.SetWemInfrastructureServiceConfiguration

Set the infrastructure service configuration on a local or remote machine.

The Set-WemInfrastructureServiceConfiguration cmdlet sets the infrastructure service configuration on a local or remote infrastructure server machine. Remote machines can be either in the same domain, or can be in a multi-forest domain environment. You can set the full configuration, or a subset of it.

–To return the current configuration on the local infrastructure server, run the cmdlet without the InfrastructureServer parameter and without the InfrastructureServerCredential parameter. All the following parameter values are applied.

–To set the current configuration to a remote server in the same domain, you must specify the InfrastructureServer parameter.

–To set the current configuration to a remote server in a multi forest Active Directory environment, you must specify the InfrastructureServer parameter (to identify the target machine) and the InfrastructureServerCredential parameter(to provide access credentials).

Example: code

```
Set-WemInfrastructureServiceConfiguration -DatabaseName "WEM_DB";
```

Set a single configuration option (DatabaseName) on the local machine.

Example: code

```
    $Enable = [Norskale.Utilities.Common.SwitchState]::Enable;
    Set-WemInfrastructureServiceConfiguration -DatabaseName "WEM_DB" -MonitoringPort 8084 -DebugMode $Enable;
```

Set multiple configuration options (DatabaseName, MonitoringPort, and EnableDebug) on the local machine.

Example: code

```
    $Enable = [Norskale.Utilities.Common.SwitchState]::Enable;
    Set-WemInfrastructureServiceConfiguration -InfrastructureServer "[Server]" -DatabaseName "WEM_DB" -MonitoringPort 8084 -DebugMode $Enable;
```

Set multiple configuration options (DatabaseName, MonitoringPort, and EnableDebug) on the remote machine in the same domain by using Windows authentication.

Example: code

```
    $passwd = ConvertTo-SecureString "[Password]" -AsPlainText -Force;
    $cred = New-Object System.Management.Automation.PSCredential("[Domain]\[UserName]", $passwd);
    Set-WemInfrastructureServiceConfiguration -InfrastructureServer "[Server]" -InfrastructureServiceAccountCredential $cred -DatabaseName "WEM_DB" -MonitoringPort 8084;
```

Set multiple configuration options (DatabaseName and MonitoringPort) on the remote machine in multi-forest domain environments.

Example: code

```
$Enable = [Norskale.Utilities.Common.SwitchState]::Enable;
$Disable = [Norskale.Utilities.Common.SwitchState]::Disable;
$config = New-Object Citrix.WEM.SDK.Configuration.InfrastructureService.SDKInfrastructureServiceConfiguration
$config.DatabaseServerInstance = "SQLServer_machine";
$config.DatabaseName = "WEM_DB";
$config.AdminServicePort = 8284;
$config.DebugMode = $Disable;
$config.SendGoogleAnalytics = $Enable
...
Set-WemInfrastructureServiceConfiguration -Configuration $config
```

Configure the infrastructure service through a single configuration object. You can also use this approach to configure the local and remote machine (in the same domain or in multi-forest domain environments).

Warning! If you use a single configuration object, make sure that you configure all required properties of the configuration object. Otherwise, infrastructure service is configured with empty values by default.

Get-WemInfrastructureServiceConfiguration

Property Commandlets.SetWemInfrastructureServiceConfiguration.DebugMode Enable WEM debug mode. Specify ‘None’to leave the current value unchanged. This is equivalent to omitting this parameter.

Property Commandlets.SetWemInfrastructureServiceConfiguration.SendGoogleAnalytics Enable collection of statistics. Specify ‘None’to leave the current value unchanged. This is equivalent to omitting this parameter.

Property Commandlets.SetWemInfrastructureServiceConfiguration.UseCacheEvenIfOnline Enable infrastructure service to always reading site settings from its cache. Specify 'None'to leave the current value unchanged. This is equivalent to omitting this parameter.

Property Commandlets.SetWemInfrastructureServiceConfiguration.DatabaseServerInstance SQL Server instance on which the WEM database is hosted. (serveraddress,port\instancename).

Property Commandlets.SetWemInfrastructureServiceConfiguration.DatabaseName WEM database name.

Property Commandlets.SetWemInfrastructureServiceConfiguration.DatabaseFailoverServerInstance Database failover server instance.

Property Commandlets.SetWemInfrastructureServiceConfiguration.SetSqlUserSpecificPassword Allow vuemUser SQL user account password to be set. Specify 'None'to leave the current value unchanged. This is equivalent to omitting this parameter.

Property Commandlets.SetWemInfrastructureServiceConfiguration.SqlUserSpecificPassword vuemUser SQL user account password.

Property Commandlets.SetWemInfrastructureServiceConfiguration.AdminServicePort Administration port for administration console to connect to the infrastructure service.

Property Commandlets.SetWemInfrastructureServiceConfiguration.AgentServicePort Agent service port for agent to connect to the infrastructure server.

Property Commandlets.SetWemInfrastructureServiceConfiguration.AgentSyncPort Cache synchronization port for agent cache synchronization process to connect to the infrastructure service.

Property Commandlets.SetWemInfrastructureServiceConfiguration.MonitoringPort WEM monitoring port.

Property Commandlets.SetWemInfrastructureServiceConfiguration.InfrastructureServiceAccountCredent
PSCredential for running the infrastructure service.

Property Commandlets.SetWemInfrastructureServiceConfiguration.EnableInfrastructureServiceAccountC
Use Windows authentication for infrastructure service database connection. Specify ‘None’to leave the current value unchanged. This is equivalent to omitting this parameter.

Property Commandlets.SetWemInfrastructureServiceConfiguration.CacheRefreshDelay Time (in minutes) before the infrastructure service refreshes its cache.

Property Commandlets.SetWemInfrastructureServiceConfiguration.SQLCheckDelay Time (in seconds) between each infrastructure service attempt to poll the SQL server.

Property Commandlets.SetWemInfrastructureServiceConfiguration.InfrastructureServiceSQLConnectionT Time (in seconds) which the infrastructure service waits when trying to establish a connection with the SQL server.

Property Commandlets.SetWemInfrastructureServiceConfiguration.EnableScheduledMaintenance Enable deletion of old statistics records from the database at periodic intervals. Specify 'None'to leave the current value unchanged. This is equivalent to omitting this parameter.

Property Commandlets.SetWemInfrastructureServiceConfiguration.StatisticsRetentionPeriod Retention period for user and agent statistics (in days).

Property Commandlets.SetWemInfrastructureServiceConfiguration.SystemMonitoringRetentionPeriod Retention period for system optimization statistics (in days).

Property Commandlets.SetWemInfrastructureServiceConfiguration.AgentRegistrationsRetentionPeriod Retention period for agent registration logs (in days).

Property Commandlets.SetWemInfrastructureServiceConfiguration.DatabaseMaintenanceExecutionTime The time at which the database maintenance action is performed (HH:MM).

Property Commandlets.SetWemInfrastructureServiceConfiguration.GlobalLicenseServerOverride Override any Citrix License Server information already in the WEM database. Specify 'None'to leave the current value unchanged. This is equivalent to omitting this parameter.

Property Commandlets.SetWemInfrastructureServiceConfiguration.LicenseServerName Citrix License Server name.

Property Commandlets.SetWemInfrastructureServiceConfiguration.LicenseServerPort Citrix License Server port.

Property Commandlets.SetWemInfrastructureServiceConfiguration.Configuration Configuration set to save the settings in.

Property Commandlets.SetWemInfrastructureServiceConfiguration.BrokerMinimumWorkerThreads Minimum number of worker threads.

Property Commandlets.SetWemInfrastructureServiceConfiguration.BrokerMinimumCompletetionPortThr Minimum number of asynchronous I/O threads.

Property Commandlets.SetWemInfrastructureServiceConfiguration.EnablePerformanceSettings Enable configuration of WCF performance settings. Specify ‘None’to leave the current value unchanged. This is equivalent to omitting this parameter.

Property Commandlets.SetWemInfrastructureServiceConfiguration.CachedDataSyncPort Cache synchronization port for agent cache synchronization process to connect to the infrastructure service.

Type SDKInfrastructureServiceConfiguration

SDK Infrastructure service Configuration object.

Property SDKInfrastructureServiceConfiguration.DebugMode Enable WEM debug mode. Specify 'None'to leave the current value unchanged. This is equivalent to omitting this parameter.

Property SDKInfrastructureServiceConfiguration.SendGoogleAnalytics Enable collection of statistics. Specify 'None'to leave the current value unchanged. This is equivalent to omitting this parameter.

Property SDKInfrastructureServiceConfiguration.UseCacheEvenIfOnline Enable infrastructure service to always reading site settings from its cache. Specify 'None'to leave the current value unchanged. This is equivalent to omitting this parameter.

Property SDKInfrastructureServiceConfiguration.DatabaseServerInstance SQL Server instance on which the WEM database is hosted. (serveraddress,port\instancename).

Property SDKInfrastructureServiceConfiguration.DatabaseName WEM database name.

Property SDKInfrastructureServiceConfiguration.DatabaseFailoverServerInstance Database failover server instance.

Property SDKInfrastructureServiceConfiguration.SetSqlUserSpecificPassword Allow vuemUser SQL user account password to be set. Specify ‘None’to leave the current value unchanged. This is equivalent to omitting this parameter.

Property SDKInfrastructureServiceConfiguration.SqlUserSpecificPassword vuemUser SQL user account password.

Property SDKInfrastructureServiceConfiguration.AdminServicePort Administration port for administration console to connect to the infrastructure service.

Property SDKInfrastructureServiceConfiguration.AgentServicePort Agent service port for agent to connect to the infrastructure server.

Property SDKInfrastructureServiceConfiguration.AgentSyncPort Cache synchronization port for agent cache synchronization process to connect to the infrastructure service.

Property SDKInfrastructureServiceConfiguration.MonitoringPort WEM monitoring port.

Property SDKInfrastructureServiceConfiguration.CachedDataSyncPort Cached data synchronization port for agent cached data synchronization process to connect to the infrastructure service.

Property SDKInfrastructureServiceConfiguration.EnablePerformanceSettings Enable configuration of WCF performance settings. Specify 'None'to leave the current value unchanged. This is equivalent to omitting this parameter.

Property SDKInfrastructureServiceConfiguration.BrokerMinimumWorkerThreads Minimum number of worker threads.

Property SDKInfrastructureServiceConfiguration.BrokerMinimumCompletetionPortThreads Minimum number of asynchronous I/O threads.

Property SDKInfrastructureServiceConfiguration.InfrastructureServiceAccountCredentialLogin Login for running the infrastructure service.

Property SDKInfrastructureServiceConfiguration.InfrastructureServiceAccountCredentialPassword Password for running the infrastructure service.

Property SDKInfrastructureServiceConfiguration.EnableInfrastructureServiceAccountCredential Use Windows authentication for infrastructure service database connection. Specify 'None'to leave the current value unchanged. This is equivalent to omitting this parameter.

Property SDKInfrastructureServiceConfiguration.CacheRefreshDelay Time (in minutes) before the infrastructure service refreshes its cache.

Property SDKInfrastructureServiceConfiguration.SqlCheckDelay Time (in seconds) between each infrastructure service attempt to poll the SQL server.

Property SDKInfrastructureServiceConfiguration.InfrastructureServiceSQLConnectionTimeout Time (in seconds) which the infrastructure service waits when trying to establish a connection with the SQL server.

Property SDKInfrastructureServiceConfiguration.EnableScheduledMaintenance Enable deletion of old statistics records from the database at periodic intervals. Specify 'None'to leave the current value unchanged. This is equivalent to omitting this parameter.

Property SDKInfrastructureServiceConfiguration.StatisticsRetentionPeriod Retention period for user and agent statistics (in days).

Property SDKInfrastructureServiceConfiguration.SystemMonitoringRetentionPeriod Retention period for system optimization statistics (in days).

Property SDKInfrastructureServiceConfiguration.AgentRegistrationsRetentionPeriod Retention period for agent registration logs (in days).

Property SDKInfrastructureServiceConfiguration.DatabaseMaintenanceExecutionTime The time at which the database maintenance action is performed (HH:MM).

Property SDKInfrastructureServiceConfiguration.GlobalLicenseServerOverride Override any Citrix License Server information already in the WEM database. Specify 'None'to leave the current value unchanged. This is equivalent to omitting this parameter.

Property SDKInfrastructureServiceConfiguration.LicenseServerName Citrix License Server name.

Property SDKInfrastructureServiceConfiguration.LicenseServerPort Citrix License Server port.

Citrix.WEM.SDK.Configuration.Client

May 23, 2023

Property BaseCmdlet.PSDebugMode Determines whether to display debugging information. Default value is False.

Property BaseCmdlet.InfrastructureServer The infrastructure server host name or IP address. Default value is "localhost".

Property BaseCmdlet.Port The port used to connect to the infrastructure service. Default value is 8284.

Type ExportAdObject

Exports all AD objects in a specified configuration set.

The Export-AdObject cmdlet exports all AD objects (user-level and machine-level) in a specified configuration set.

Example: code

```
Export-AdObject -InfrastructureServer "10.10.10.10" -FolderName "C:\backup" -SiteId 3
```

Exports all AD objects in the configuration set whose ID is 3 to the folder "C:\backup". The IP address of the remote infrastructure server is "10.10.10.10".

Example: code

```
Export-AdObject -FolderName "C:\backup" -SiteName "Default Site"
```

Exports all AD objects (on the local infrastructure server) in "Default Site"to the folder "C:\backup".

Property ExportAdObject.FolderName The name of the folder to which all AD objects are exported. Note: The folder is created automatically if it does not exist.

Property ExportAdObject.SiteName The name of the configuration set whose AD objects are to be exported. Note: This parameter does not work if the parameter "SiteId"is specified.

Property ExportAdObject.SiteId The ID of the configuration set whose AD objects are to be exported.

Type ExportSite

Exports a specified configuration set.

The Export-WemSite cmdlet exports a WEM configuration set.

Example: code

```
Export-WemSite -InfrastructureServer "10.10.10.10" -FolderName "C:\backup-site" -SiteId 3
```

Exports the configuration set whose ID is 3 to the folder "C:\backup-site". The IP address of the remote infrastructure server is "10.10.10.10".

Example: code

```
Export-WemSite -FolderName "C:\backup-site" -SiteName "Default Site"
```

Exports "Default Site"(on the local infrastructure server) to the folder "C:\backup-site".

Property ExportSite.FolderName The name of the folder to which the configuration set is exported. Note: The folder is created automatically if it does not exist.

Property ExportSite.SiteName The name of the configuration set to be exported. Note: This parameter does not work if the parameter "SiteId"is specified.

Property ExportSite.SiteId The ID of the configuration set to be exported.

Type ImportAdObject

Imports all AD objects to a specified configuration set.

The Import-AdObject cmdlet imports all AD objects (user-level and machine-level) to a specified configuration set.

Example: code

```
Import-AdObject -InfrastructureServer "10.10.10.10" -FolderName "C:\backup" -SiteId 3
```

Imports all AD objects from the folder “C:\backup”to the configuration set whose ID is 3. The IP address of the remote infrastructure server is “10.10.10.10”.

Example: code

```
Import-AdObject -FolderName "C:\backup" -SiteName "Default Site"
```

Imports AD objects from the folder “C:\backup”to “Default Site”. By default, this operation is performed on the local infrastructure server unless otherwise specified.

Property ImportAdObject.FolderName The name of the folder from which AD objects are to be imported.

Property ImportAdObject.SiteName The name of the configuration set to which AD objects are to be imported. Note: This parameter does not work if the parameter “SiteId”is specified.

Property ImportAdObject.SiteId The ID of the configuration set to which AD objects are to be imported.

Type ImportSite

Imports a WEM configuration set.

The Import-WemSite cmdlet imports a WEM configuration set from a folder.

Example: code

```
Import-WemSite -InfrastructureServer "10.10.10.10" -FolderName "C:\
   backup-site"
```

Imports a WEM configuration set from the folder "C:\backup-site". The IP address of the remote infrastructure server is "10.10.10.10".

Property ImportSite.FolderName The name of the folder from which the configuration set is to be imported.

Type CreateMachineAdObject

Creates a machine-level AD object.

The New-MachineAdObject cmdlet creates a machine-level AD object.

Example: code

```
$Machine = New-Object Citrix.DeviceMgmt.Agent.Windows.Sdk.
   MachineModel
$Machine.Name = "CN=YourComputerName,CN=Computers,DC=domain,DC=
   local"
$Machine.Type = "Computer"
$Machine.Enabled = $True
$Machine.Priority = 100
New-MachineAdObject -InfrastructureServer "10.10.10.10" -
   MachineAdObject $Machine
```

Creates a computer named "YourComputerName"with a priority of 100. The IP address of the remote infrastructure server is "10.10.10.10".

Example: code

```
$Machine = New-Object Citrix.DeviceMgmt.Agent.Windows.Sdk.
   MachineModel
$Machine.Name = "OU=YourOUName,DC=domain,DC=local"
$Machine.Type = "OU"
$Machine.Enabled = $False
$Machine.Priority = 80
New-MachineAdObject -MachineAdObject $Machine
```

Creates a disabled OU named "YourOUName"with a priority of 80 on the local infrastructure server.

Property CreateMachineAdObject.MachineAdObject The machine-level AD object to be created. Note: Use "distinguished name"or "SID"to specify this machine-level AD object. You do not need to specify the "Id"property when creating this object because the ID is generated automatically.

Type DeleteMachineAdObject

Deletes a machine-level AD object.

The Remove-MachineAdObject cmdlet deletes a machine-level AD object.

Example: code

```
    Remove-MachineAdObject -InfrastructureServer "10.10.10.10" -Id 3
```

Deletes a machine-level AD object whose ID is 3. The IP address of the remote infrastructure server is "10.10.10.10".

Example: code

```
    $Machine = (Get-MachineAdObject) | Where-Object {
  $_.Name.ToLower().Contains("cn=your-computer-name,")   }
 )
    Remove-MachineAdObject -MachineAdObject $Machine.Id
```

Deletes a computer named "your-computer-name"on the local infrastructure server.

Property DeleteMachineAdObject.Id The ID of the machine-level AD object to be deleted.

Type GetMachineAdObject

Queries machine-level AD objects.

The Get-MachineAdObject cmdlet queries machine-level AD objects.

Example: code

```
Get-MachineAdObject -InfrastructureServer "10.10.10.10"
```

Queries all machine-level AD objects. The IP address of the remote infrastructure server is “10.10.10.10”.

Example: code

```
Get-MachineAdObject -SiteName "Default Site"
```

Queries all machine-level AD objects in “Default Site”on the local infrastructure server.

Example: code

```
Get-MachineAdObject -Id 10
```

Queries the machine-level AD object whose ID is 10 on the local infrastructure server.

Example: code

```
Get-MachineAdObject -Sid "S-1-5-21-1375803180"
```

Queries the machine-level AD object whose SID is “S-1-5-21-1375803180”on the local infrastructure server.

Property GetMachineAdObject.SiteName The name of the configuration set that is used to filter machine-level AD objects. Only the AD objects that belong to this configuration set are to be displayed. If this parameter is not specified, the filtering operation does not work. Note: This parameter does not work if the parameter “SiteId”is specified.

Property GetMachineAdObject.SiteId The ID of the configuration set that is used to filter machine-level AD objects. Only the AD objects that belong to this configuration set are to be displayed. If this parameter is not specified, the filtering operation does not work.

Property GetMachineAdObject.Id If this parameter is specified, only the machine-level AD object with the specified ID is to be displayed.

Property GetMachineAdObject.Sid If this parameter is specified, only the machine-level AD object with the specified SID is to be displayed.

Type UpdateMachineAdObject

Updates a machine-level AD object.

The Update-MachineAdObject cmdlet updates a machine-level AD object.

Example: code

```
    $Machine = (Get-MachineAdObject -InfrastructureServer "10.10.10.10") | Where-Object {
  $_.Name.ToLower().Contains("cn=your-computer-name,")  }

    $Machine.Enable = $False
    $Machine.Priority += 10
    $Machine.Description = "Modify the description"
    Update-MachineAdObject -InfrastructureServer "10.10.10.10" -MachineAdObject $Machine
```

Updates a machine whose name is “your-computer-name”. The IP address of the remote infrastructure server is “10.10.10.10”. These commands disable the machine status, change the description, and increase the priority by 10.

Property UpdateMachineAdObject.MachineAdObject The machine-level AD object to be updated. Note: This cmdlet updates the machine-level AD object according to the property “Id”. The properties “Sid”, “Name”, and “Type”are read-only, and they remain unchanged even if you specify values for them.

Type CreateSite

Creates a configuration set.

The New-WemSite cmdlet creates a WEM configuration set.

Example: code

```
$Site = New-Object Citrix.DeviceMgmt.Agent.Windows.Sdk.SiteModel
$Site.Name = "New Configuration Set"
$Site.Description = "This is a new configuration set created by
   Powershell SDK"
New-WemSite -InfrastructureServer "10.10.10.10" -Site $Site
```

Creates a new configuration set named "New Configuration Set". The IP address of the remote infrastructure server is "10.10.10.10".

Property CreateSite.Site The configuration set to be created. Note: You do not need to specify the "Id"property when creating this object because the ID is generated automatically.

Type DeleteSite

Deletes a configuration set.

The Remove-WemSite cmdlet deletes a WEM configuration set.

Example: code

```
Remove-WemSite -InfrastructureServer "10.10.10.10" -Id 12
```

Deletes a configuration set whose ID is 12. The IP address of the remote infrastructure server is "10.10.10.10".

Example: code

```
Remove-WemSite -SiteName "New Configuration Set"
```

Deletes a configuration set whose name is "New Configuration Set". By default, this operation is performed on the local infrastructure server unless otherwise specified.

Property DeleteSite.SiteId The ID of the configuration set to be deleted. Note: You cannot delete the built-in configuration set named “Default Site”.

Property DeleteSite.SiteName The name of the configuration set to be deleted. Note: This parameter does not work if the parameter “SiteId”is specified. You cannot delete the built-in configuration set named “Default Site”.

Type GetSite

Queries WEM configuration sets.

The Get-WemSite cmdlet queries WEM configuration sets.

Example: code

```
Get-WemSite -InfrastructureServer "10.10.10.10"
```

Queries all WEM configuration sets. The IP address of the remote infrastructure server is “10.10.10.10” .

Example: code

```
Get-WemSite -SiteName "Default Site"
```

Queries the configuration set whose name is “Default Site”. By default, this operation is performed on the local infrastructure server unless otherwise specified.

Example: code

```
Get-WemSite -SiteId 2
```

Queries the configuration set whose ID is 2 on the local infrastructure server.

Property GetSite.SiteName The name of the configuration set to be queried. All configuration sets are to be displayed if this parameter is not specified. Note: This parameter does not work if the parameter “SiteId”is specified.

Property GetSite.SiteId The ID of the configuration set to be queried. All configuration sets are to be displayed if this parameter is not specified.

Type UpdateSite

Updates a WEM configuration set.

The Update-WemSite cmdlet updates a WEM configuration set.

Example: code

```
$Site = Get-WemSite -InfrastructureServer "10.10.10.10" -SiteName "Default Site"
$Site.Name = "New Name"
$Site.Description = "Modify the description"
Update-WemSite -InfrastructureServer "10.10.10.10" -Site $Site
```

Updates the name and description of “Default Site”. The IP address of the remote infrastructure server is “10.10.10.10”.

Property UpdateSite.Site The configuration set to be updated. Note: This cmdlet updates the configuration set according to the property “Id”.

Type CreateUserAdObject

Creates a user-level AD object.

The New-UserAdObject cmdlet creates a user-level AD object.

Example: code

```
$User = New-Object Citrix.DeviceMgmt.Agent.Windows.Sdk.UserModel
$User.Name = "CN=Domain Users,CN=Users,DC=domain,DC=local"
$User.Type = "Group"
$User.Enabled = $True
$User.Priority = 100
New-UserAdObject -InfrastructureServer "10.10.10.10" -UserAdObject $User
```

Creates a group named "Domain Users"with a priority of 100. The IP address of the remote infrastructure server is "10.10.10.10".

Example: code

```
$User = New-Object Citrix.DeviceMgmt.Agent.Windows.Sdk.UserModel
$User.Name = "CN=User1,CN=Users,DC=domain,DC=local"
$User.Type = "User"
$User.Enabled = $False
$User.Priority = 80
New-UserAdObject -UserAdObject $User
```

Creates a disabled user named "User1"with a priority of 80 on the local infrastructure server.

Property CreateUserAdObject.UserAdObject The user-level AD object to be created. Note: Use "distinguished name"or "SID"to specify this user-level AD object. You do not need to specify the "Id" property when creating this object because the ID is generated automatically.

Type DeleteUserAdObject

Deletes a user-level AD object.

The Remove-UserAdObject cmdlet deletes a user-level AD object.

Example: code

```
Remove-UserAdObject -InfrastructureServer "10.10.10.10" -Id 12
```

Deletes a user-level AD object whose ID is 12. The IP address of the remote infrastructure server is "10.10.10.10".

Example: code

```
    $User = (Get-UserAdObject) | Where-Object {
  $_.Name.ToLower().Contains("cn=user1,")  }
 )
    Remove-UserAdObject -UserAdObject $User.Id
```

Deletes a user whose name is “user1”on the local infrastructure server.

Property DeleteUserAdObject.Id The ID of the user-level AD object to be deleted. Note: You cannot delete the built-in user-level AD objects (Everyone and BUILTIN\Administrators) using this cmdlet.

Type GetUserAdObject

Queries user-level AD objects.

The Get-UserAdObject cmdlet queries user-level AD objects.

Example: code

```
    Get-UserAdObject -InfrastructureServer "10.10.10.10"
```

Queries all user-level AD objects. The IP address of the remote infrastructure server is “10.10.10.10”

.

Example: code

```
    Get-UserAdObject -SiteName "Default Site"
```

Queries all user-level AD objects in “Default Site”on the local infrastructure server.

Example: code

```
    Get-UserAdObject -Id 10
```

Queries the user-level AD object whose ID is 10 on the local infrastructure server.

Example: code

```
      Get-UserAdObject -Sid "S-1-1-0"
```

Queries all user-level AD objects whose SIDs are “S-1-1-0”on the local infrastructure server.

Property GetUserAdObject.SiteName The name of the configuration set that is used to filter user-level AD objects. Only the AD objects that belong to this configuration set are to be displayed. If this parameter is not specified, the filtering operation does not work. Note: This parameter does not work if the parameter “SiteId”is specified.

Property GetUserAdObject.SiteId The ID of the configuration set that is used to filter user-level AD objects. Only the AD objects that belong to this configuration set are to be displayed. If this parameter is not specified, the filtering operation does not work.

Property GetUserAdObject.Id If this parameter is specified, only the user-level AD object with the specified ID is to be displayed.

Property GetUserAdObject.Sid The SID used to filter user-level AD objects. Only the AD objects with the specified SID are to be displayed. If this parameter is not specified, the filtering operation does not work.

Type UpdateUserAdObject

Updates a user-level AD object.

The Update-UserAdObject cmdlet updates a user-level AD object.

Example: code

```
    $User = (Get-UserAdObject -InfrastructureServer "10.10.10.10") | Where-Object {
  $_.Name.ToLower().Contains("cn=user1,")  }

    $User.Enable = $False
    $User.Priority += 10
    $User.Description = "Modify the description"
    Update-UserAdObject -InfrastructureServer "10.10.10.10" -UserAdObject $User
```

Updates a user whose name is “user1”. The IP address of the remote infrastructure server is “10.10.10.10”. These commands disable the user status, change the description, and increase the priority by 10.

Property UpdateUserAdObject.UserAdObject The user-level AD object to be updated. Note: This cmdlet updates the user-level AD object according to the property “Id”. The properties “Sid”, “Name” , and “Type”are read-only, and they remain unchanged even if you specify values for them.

Property Norskale.Administration.Console.Configuration.IAuthorizationInfo.CurrentAuthorizationLevel Current user authorization level.
