# Change SQL Server Service Account Password using PowerShell Database Driven Solution

This solution provides additional flexibility to allow for DBA team to change service account passwords across multiple service accounts with different passwords on multiple computer accounts.

## Step #1: Setup the Database and Add Account Entries

1. First run the SvcAcctMgmt.sql to setup the database and required stored procedures.
2. Create all the entries required, review Example.HowToAddComputerAccount.sql for details.

## Step #2: Update Password

The PowerShell script uses SQL Server SMO and Win32\_Service class to update the password for one or more servers. I have tested the script against SQL Server 2016 on Windows 2016; please test the script for yourself before executing on production. It relies on the parent script ChangeServiceAccountPassword\_v3.ps1.

**Created by:** Mohit K. Gupta ([mogutpa@microsoft.com](mailto:mogutpa@microsoft.com))

**LastUpdated:** August 5, 2017

**Version:** 1.00

## Execution Method 1: Updating Password Information on a Single Server

|  |
| --- |
| **Command to Execute** |
| .\UpdateServiceAccountPasswords\_v1.ps1 -ComputerName SQL1 -ServiceAccountName "LAB\SQLSvc" -SvcAcctMgmtSQLInstance MOGUPTA-PC01 -PassKey “SuperComplicatedPassKeyToProtectAllPasswords” |
| **Output** |
| ComputerName : SQL1  OperatingSystem : Microsoft Windows Server2016  ServiceName : MSSQLSERVER  ServiceMode : Auto  ServiceState : Running  OperationStatus : Password Change Successfully |

## Execution Method 2: Updating Password Information on Multiple Servers for a Single Service Account

|  |
| --- |
| **Command to Execute** |
| .\UpdateServiceAccountPasswords\_v1.ps1 -ServiceAccountName "LAB\SQLSvc" -SvcAcctMgmtSQLInstance MOGUPTA-PC01 -PassKey “SuperComplicatedPassKeyToProtectAllPasswords” |

## Execution Method 3: Updating Passwords Information on a single servers with multiple service accounts

|  |
| --- |
| **Command to Execute** |
| .\UpdateServiceAccountPasswords\_v1.ps1 -ComputerName SQL1 -SvcAcctMgmtSQLInstance MOGUPTA-PC01 -PassKey “SuperComplicatedPassKeyToProtectAllPasswords” |

## Execution Method 4: Updating Passwords Information on all servers

|  |
| --- |
| **Command to Execute** |
| .\UpdateServiceAccountPasswords\_v1.ps1 -SvcAcctMgmtSQLInstance MOGUPTA-PC01 -PassKey “SuperComplicatedPassKeyToProtectAllPasswords” |

## Support for -Verbose

Any method can have the -Verbose added to get additional information from execution. If the script is failing due to WMI error or password change is failing; you’ll need to run in -Verbose to capture the error.

|  |
| --- |
| **Example Verbose Output** |
| OVERBOSE: 05/04/2017 00:02:25 Updating SQL Server Services' Service Accounts' Passwords  VERBOSE: 05/04/2017 00:02:25 Changing Password on Target \\SQL1\MSSQL$CLUSINST via Method Change (Win32\_Service)  VERBOSE: 05/04/2017 00:02:25 Changing Password on Target \\SQL1\SQLAgent$CLUSINST via Method Change (Win32\_Service)  VERBOSE: 05/04/2017 00:02:26 Changing Password on Target \\SQL2\MSSQL$CLUSINST via Method ChangePassword (SQLSMO)  VERBOSE: 05/04/2017 00:02:27 Changing Password on Target \\SQL2\SQLAgent$CLUSINST via Method ChangePassword (SQLSMO)  VERBOSE: 05/04/2017 00:02:27 Script Completed. |

## Support for -WhatIf

If you wish to see what services will be impacted by executing the command you can also supply -WhatIf switch. Notice when running with -WhatIf, you do not need to supply password information. In this example I am also outputting the results as a table for readability.

|  |
| --- |
| **Command to Execute** |
| .\UpdateServiceAccountPasswords\_v1.ps1 -SvcAcctMgmtSQLInstance MOGUPTA-PC01 -PassKey “SuperComplicatedPassKeyToProtectAllPasswords” -WhatIf | FT |
| **Output** |
| ComputerName OperatingSystem ServiceName ServiceMode ServiceState OperationStatus  ------------ --------------- ----------- ----------- ------------ ---------------  SQL1 Microsoft Windows Server 2016 Datacenter MSSQL$CLUSINST Manual Stopped Dry Run  SQL1 Microsoft Windows Server 2016 Datacenter SQLAgent$CLUSINST Manual Stopped Dry Run  SQL2 Microsoft Windows Server 2016 Datacenter MSSQL$CLUSINST Manual Running Dry Run  SQL2 Microsoft Windows Server 2016 Datacenter SQLAgent$CLUSINST Manual Running Dry Run |

## Outputting Results to CSV

If you are running it against a server list, you can capture the information to a CSV file via following command.

|  |
| --- |
| **Command to Execute** |
| .\UpdateServiceAccountPasswords\_v1.ps1 -SvcAcctMgmtSQLInstance MOGUPTA-PC01 -PassKey “SuperComplicatedPassKeyToProtectAllPasswords” | Export-Csv PasswordResult.csv -NoTypeInformation |
| **Output (PasswordResults.csv)** |
| "ComputerName","OperatingSystem","ServiceName","ServiceMode","ServiceState","OperationStatus"  "SQL1","Microsoft Windows Server 2016 Datacenter","MSSQL$CLUSINST","Manual","Stopped","Password Change Completed"  "SQL1","Microsoft Windows Server 2016 Datacenter","SQLAgent$CLUSINST","Manual","Stopped","Password Change Completed"  "SQL2","Microsoft Windows Server 2016 Datacenter","MSSQL$CLUSINST","Manual","Running","Password Change Completed"  "SQL2","Microsoft Windows Server 2016 Datacenter","SQLAgent$CLUSINST","Manual","Running","Password Change Completed" |