Import Failed with 40632 -The password does not meet policy requirements

Last updated by | Keith Elmore | Apr 5, 2021 at 7:56 AM PDT

Contents

Import Failed with 40632 -The password does not meet policy requirements

Scenario

Unable to import from bacpac

Error

Exception Microsoft.SqlServer.Management.Dac.Services.ServiceException:Error encountered during the service operation.

at Microsoft.SqlServer.Management.Dac.Services.ImportJobHandler.Run(JobContext jobContext, CancellationToken jobCancellationToken)

at Microsoft.SqlServer.Management.Dac.Services.RequestProcessingHandler.Run()

Inner exception Microsoft.SqlServer.Dac.DacServicesException:Could not import package.

Warning SQL72012: The object [data_0] exists in the target, but it will not be dropped even though you selected the 'Generate drop statements for objects that are in the target database but that are not in the source' check box.

Warning SQL72012: The object [log] exists in the target, but it will not be dropped even though you selected the 'Generate drop statements for objects that are in the target database but that are not in the source' check box.

Error SQL72014: .Net SqlClient Data Provider: Msg 40632, Level 16, State 3, Line 1 Password validation failed. The password does not meet policy requirements because it is not complex enough.

Error SQL72045: Script execution error. The executed script:

CREATE USER [prod_ssrs_svc]

WITH PASSWORD = N'uHefunycsvc<dm1cgqusiaelmsFT7_&#$!~<ZkTu4ozwfIda';

Or

Message - (SQL000000) Creating [siku_E3A3B72B]...

Error - (SQL072014) .Net SqlClient Data Provider: Msg 40632, Level 16, State 1, Line 1 Password validation failed. The password does not meet policy requirements because it is not complex enough.

Error - (SQL072045) Script execution error. The executed script:

CREATE USER [siku_E3A3B72B]

WITH PASSWORD = N'yfq>dexuqvmjbenx9_anjtulmsFT7_&#$!~<eqftsikuzoog';

Cause :

--The password that is stored in a bacpac file is not the actual password or the actual password in encrypted form.

--For security reasons, bacpac files never contain the actual production passwords that are in your database. Instead, **a password is randomly generated and stored in the bacpac file for each password-secured object (e.g. each user).**

--This is by design and it means that after a bacpac file is imported to create a new database, you'll need to modify each user to set the password as desired.

Known Bug

--The randomly-generated password that is stored in the bacpac file is causing problems because it does not meet Azure SQL DB's password requirements. It turns out that the password that was randomly generated contains a 4-character substring match with the user name ("siku"), which violates Azure SQL DB's password complexity rules. This is unfortunately an oversight in the tool that performs the export operation, and we will correct it going forward.

--**The bug** is that the export service should check the username in order to prevent generating a random password that contains a substring that matches the username or part of it.

ID	Title	Assigned To	State	Work Item Type
9502072	DACFX: BACPAC export can generate password the don't meet Azure SQLDB complexity requirements			

Workaround

-It's possible to import the problem bacpac, but it will require unzipping the bacpac file and modifying the contents in order to replace this troublesome password.

Instructions:

1. A bacpac file is actually a ZIP archive with a different extension. Changing the extension to .zip lets you open the archive in Windows Explorer (or any archive utility that supports ZIP archives, in which case it may not be necessary to change the extension at all). Open the bacpac archive and copy the files **model.xml** and **Origin.xml** to a temporary location.
2. Open the **model.xml** file in a text editor (Visual Studio works well for this), and then find the User elements. Each one will look like this: <Element Type="SqlUser" Name="[user1]">
   <Property Name="AuthenticationType" Value="2" />
   <Property Name="Password" Value="kjlasfdiowerljk" />
   </Element>
3. Edit the highlighted part of the User elements, modifying the password to one that meets Azure SQL DB's password requirements. For reference, the password must:
   - Not contain any portion or the user name (3-characters in length or greater)
   - Must be 8 or more characters long
   - Must contain 3 out of 4 of:
     - Lower-case characters (a-z)
     - Upper-case characters (A-Z)
     - Numbers (0-9)
     - Special characters: (`~!@#$%^&*_-+=|\\{}[]:;"'<>,.?)/
4. A checksum is used to verify the model.xml file's contents. Compute a new checksum for the model.xml file using this PowerShell script: Copy the below into a file computeHash.ps1 and execute it

Please provide the path to the model.xml file, like so: C:\temp\model.xml

```
$modelXmlPath = Read-Host "model.xml file path"

$hasher = [System.Security.Cryptography.HashAlgorithm]::Create("System.Security.Cryptography.SHA256CryptoServiceProvider")

$fileStream = new-object System.IO.FileStream ` -ArgumentList @($modelXmlPath, [System.IO.FileMode]::Open)

$hash = $hasher.ComputeHash($fileStream)

$hashString = ""

Foreach ($b in $hash) { $hashString += $b.ToString("X2") }

$fileStream.Close()

$hashString
```

5. Open the **Origin.xml** file, find the line that contains the checksum value, and replace it with the newly computed checksum. The checksum line in Origin.xml looks like this:

```
    &lt;Checksum
Uri=&quot;/model.xml&quot;&gt;D3B38EF111CF3919FA042177D5251D5581B82BA453AC598FB1FB624B3A58B3758
```

6. Finally, copy the modified **Origin.xml** and **model.xml** files back into the bacpac archive, re-zip the files (if necessary), and change the archive's extension to .bacpac.
7. You can now import the bacpac file to Azure SQL DB.

Note:

- It's important when zipping the bacpac file that the contents are not inside of a folder. Instead, they must be in the top-level of the zip file.
- The file name of Origin.xml is case sensitive – please verify that the file name is exactly Origin.xml.

By the way, to expedite your testing you might find it convenient to test the import of the bacpac file using SqlPackage.exe. SqlPackage.exe uses the same library (the Data-Tier Application Framework) that is used by the Azure SQL Database Import/Export service in the Azure portal. You can download the latest version of SqlPackage.exe here: https://www.microsoft.com/en-us/download/details.aspx?id=54106

To perform an import operation using SqlPackage.exe you can run this command from a command prompt:

C:\Program Files (x86)\Microsoft SQL Server\130\DAC\bin\sqlpackage.exe /a:import /sf:C:\temp\mydb.bacpac /tcs:"Data Source=SERVERNAME;Initial Catalog=NEW_DB_NAME;UserId=USERNAME;Password=PASSWORD"

-- Finally, regarding the Azure SQL DB password complexity policy – the password policy is not user-configurable.

Classification

Root Cause: Azure SQL DB v2\Import/Export\User Issue/Error

How good have you found this content?