SQL Package Known Issues and Limitations

Last updated by | Vitor Tomaz | Jun 8, 2022 at 5:31 AM PDT

Contents

Frequently Asked Questions:

Compression rate

- A customer might complain that they don't think they're database got fully exported because the bacpac is so much smaller than the DB. They should try to import first and see if created DB really isn't the same size.
- Sqlpackage just uses zip compression, and it all depends on the data. We've seen 10GB compress to 10MB and a database with almost no compression since the database stored compressed zip files as data.

Permissions required for running sqlpackage:

- https://docs.microsoft.com/en-us/sql/relational-databases/data-tier-applications/upgrade-a-data-tier-application?redirectedfrom=MSDN&view=sql-server-ver15#Permissions ☒

- A DAC can only be upgraded by members of the sysadmin or serveradmin fixed server roles, or by logins that are in the dbcreator fixed server role and have ALTER ANY LOGIN permissions. The login must be the owner of the existing database. The built-in SQL Server system administrator account named SA can also upgrade a DAC.

Limitations:

Elastic Pool

- Import to an elastic pool is not supported. The workaround is to import to a single db, then move it into an elastic pool ☒

Elastic Jobs Database

- Unable to export Elastic Jobs Database

3 part naming

- It has always been a restriction of sqlpackage that it cannot deal with references to objects outside the current database. When using the resulting bacpac to recreate the objects, there is no guarantee that the external database/object will exist. It's easy to construct a situation with circular references which could never be reconstructed using a bacpac-style system.

Known issues:

- XML parsing: Document parsing required too much memory This is an engine failure on the server-side. The XML data that is trying to be uploaded is too complex/large, which is resulting in the server throwing this error.

Workarounds:

1. Try increasing the edition to a Premium tier.
2. Change the XML column datatype to [n]varchar(max). This bypasses the XML parsing step in the engine and data is imported as is.
3. Identify and remove the problematic XML data from the table so that it isn't exported.

How good have you found this content?