Partition Compression Issue

# Bug Scenario

* Partitioned table in source control using data compression
* Data moves on and an outside process creates extra partitions
* SSDT Shows data compression on every partition in source control
* SSDT will now constantly see this data compression on a new partition as a discrepancy
* Its not feasible to change source control every time a new partition is created as this is fluid and automated at different times for different tables.
* This causes Sqlpackage to do whole table rebuilds to “fix” when nothing is actually wrong. This can block up our app as it tries to do it on 4tb tables

# Replication Steps

1. Deploy PartitionCompressionIssue Database project to your local instance
2. Go to Tools > New Schema Comparison
3. Use PartitionCompressionIssue project as source
4. Use localhost. PartitionCompressionIssue as target
5. You should have no differences
6. Now run the below script to add a new partition into your table. This mimics the actions of a partition maintenance job which runs daily/every other day.

ALTER PARTITION SCHEME ps\_PartitionCompressionIssue

NEXT USED [PRIMARY];

ALTER PARTITION FUNCTION pf\_PartitionCompressionIssue ()

SPLIT RANGE (1000);

1. We now get a table discrepancy and partition discrepancy

Graphical user interface, application

Description automatically generated

1. Partition discrepancy we can just remove by selecting Ignore partition schemes which is fine as we’d do that anyway because we know partitions will always be moving onwards.
2. The partition compression we are totally blocked. It will try to rebuild the table on every deploy after a new partition is there

## Source Control

WITH (DATA\_COMPRESSION = PAGE ON PARTITIONS (1), DATA\_COMPRESSION = PAGE ON PARTITIONS (2), DATA\_COMPRESSION = PAGE ON PARTITIONS (3), DATA\_COMPRESSION = PAGE ON PARTITIONS (4)) ON [ps\_PartitionCompressionIssue] ([UserAccountID])

## Database

WITH (DATA\_COMPRESSION = PAGE ON PARTITIONS (1), DATA\_COMPRESSION = PAGE ON PARTITIONS (2), DATA\_COMPRESSION = PAGE ON PARTITIONS (3), DATA\_COMPRESSION = PAGE ON PARTITIONS (5), DATA\_COMPRESSION = PAGE ON PARTITIONS (4)) ON [ps\_PartitionCompressionIssue] ([UserAccountID])

See highlighted difference above. This is the issue. SSDT is classing this as a discrepancy, when in our source control table we set DATA\_COMPRESSIONS=PAGE for all so this doesn’t need to be split out into data compression for each partition.

If we were to deploy this in production it would try and move our Terabyte tables to new tables to fix this “discrepancy”