Restore (or) Migrate Partition DB from Enterprise to Standard Edition

<https://gallery.technet.microsoft.com/scriptcenter/RestoreMigrate-partition-f61e4afc>

**Author**: [Mohammad Abdul Majeed](https://social.technet.microsoft.com/profile/mohammad%20abdul%20majeed/)

**Requirement:**

I got the requirement to restore/Migrate partition database from SQL 2005 Enterprise Edition to SQL2005/2008/2008R2 Standard edition to avoid the Enterprise edition license.

**But we have 10 tables with partition in that** Enterprise database and standard Edition won’t support the Partition function. Hence we have followed the below steps to achieve this task which is suitable to our requirement.

**Note:**

* If I have a database running on a (SQL Server 2008 R2 Developer Edition-Enterprise Edition) that has no enterprise features and take a backup of that database, I can restore it to a SQL Server 2008 R2 Standard edition instance.
* If I have a database that has enterprise features (data compression, transparent data encryption, partition,change data capture) and try to restore it to an SQL Server 2008 R2 Standard Edition instance I will fail.

**Steps Followed:**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Activity** | **Comment** |
| 1 | Stopped the application |  |
| 2 | Backup PartitionDB database and restore on Ent server with other name(PartitionDB\_Temp) |  |
| 3 | Identified the Partion tables | Script Attached |
| 4 | Dropped the partition tables fromPartitionDB\_Temp Note: We had three tables with Foreign Key relationship with other table hence we followed the below steps to drop those tables. | Script Attached |
| 5 | Identified the Partition tables Foreign Keys relationship for that three tables | Script Attached |
| 6 | Dropped that Foreign Keys | Script Attached |
| 7 | Now we dropped that 3 tables after removing the Foreign Keys relation ship | Script Attached |
| 8 | Script out all that 10 tables from PartitionDB database with constraints and related triggers | You can use generate script option with Script triggers true and script index True option |
| 9 | Remove the Partition Scheme from all the table scripts | You can use control find to identify the scheme name |
| 10 | Created these tables in PartitionDB\_Temp database without referring the partition scheme |  |
| 11 | Script out all the Foreign Keys constraints form PartitionDB which we deleted in step 6 |  |
| 12 | Created the dropped Foreign Keys constraints in PartitionDB\_Temp database |  |
| 13 | Exported the data of these 10 tables from PartitionDB database to PartitionDB\_Temp database | You can use Import/Export Wizard |
| 14 | Drop the partition Scheme from PartitionDB\_Temp database | Script Attached |
| 15 | Drop the partition Function from PartitionDB\_Temp database | Script Attached |
| 16 | Check & ensure the data should be available only in primary file group on PartitionDB\_Temp database for all tables | Script Attached |
| 17 | Drop all file groups except Primary from PartitionDB\_Temp database | Script Attached |
| 18 | Compared & ensured the dependent objects for 10 tables in between PartitionDB and PartitionDB\_Temp database are in sync | Script Attached |
| 19 | Took the backup of PartitionDB\_Temp on Ent DB server |  |
| 20 | Copied the backup to New SQL 2008 R2 standard edition server |  |
| 21 | Restored the PartitionDB\_Temp database backup on New SQL 2008 R2 Standard Edition |  |
| 22 | Compare the data between old PartitionDB DB and New PartitionDB DB | Script Attached |

**Identified the Partition tables:**

selectdistinct

p.[object\_id],

TbName =OBJECT\_NAME(p.[object\_id]),

index\_name = i.[name],

index\_type\_desc = i.type\_desc,

partition\_scheme = ps.[name],

data\_space\_id = ps.data\_space\_id,

function\_name = pf.[name],

function\_id = ps.function\_id

fromsys.partitions p

innerjoinsys.indexes i

on p.[object\_id] = i.[object\_id]

and p.index\_id = i.index\_id

innerjoinsys.data\_spaces ds

on i.data\_space\_id = ds.data\_space\_id

innerjoinsys.partition\_schemes ps

on ds.data\_space\_id = ps.data\_space\_id

innerJOINsys.partition\_functions pf

on ps.function\_id = pf.function\_id

-- WHERE p.[object\_id] = object\_id('JBMTest')

orderby TbName, index\_name ;

GO

**To Drop the Partition Tables from database:**

Droptabledbo.Tablename1

Droptable dbo.Tablename2

**Identified Foreign Keys relationship in tables:**

SELECT\*

FROMsys.foreign\_keys

WHERE referenced\_object\_id =object\_id('Tablename’)

**Generate script to drop Foreign Keys relationship for tables:**

SELECT

'ALTER TABLE '+OBJECT\_NAME(parent\_object\_id)+

' DROP CONSTRAINT '+ name

FROMsys.foreign\_keys

WHERE referenced\_object\_id =object\_id('TM\_TransactionMaster')

**Drop the partition Scheme:**

DROPPARTITION SCHEME [SchemaName]

GO

**Drop the partition Function:**

DROPPARTITIONFUNCTION [FunctionName]

GO

**Check & ensure the data should be available only in primary file group on PartitionDB\_Temp database for all tablesList allObjectsandIndexes per Filegroup /Partition**

-------------------------------------------------------------------

SELECT DS.name AS DataSpaceName

,AU.type\_desc AS AllocationDesc

,AU.total\_pages / 128 AS TotalSizeMB

,AU.used\_pages / 128 AS UsedSizeMB

,AU.data\_pages / 128 AS DataSizeMB

,SCH.name AS SchemaName

,OBJ.type\_desc AS ObjectType

,OBJ.name AS ObjectName

,IDX.type\_desc AS IndexType

,IDX.name AS IndexName

FROMsys.data\_spacesAS DS

INNERJOINsys.allocation\_unitsAS AU

ON DS.data\_space\_id = AU.data\_space\_id

INNERJOINsys.partitionsAS PA

ON (AU.typeIN(1, 3)

AND AU.container\_id = PA.hobt\_id)

OR

(AU.type= 2

AND AU.container\_id = PA.partition\_id)

INNERJOINsys.objectsAS OBJ

ON PA.object\_id= OBJ.object\_id

INNERJOINsys.schemasAS SCH

ON OBJ.schema\_id= SCH.schema\_id

LEFTJOINsys.indexesAS IDX

ON PA.object\_id= IDX.object\_id

AND PA.index\_id = IDX.index\_id

ORDERBY DS.name

,SCH.name

,OBJ.name

,IDX.name

**Drop file groups:**

USE [DBName]

GO

ALTERDATABASE [DBName] REMOVE FILE [Filegroupname]

GO

**Compare the data between old PartitionDB DB and New PartitionDB DB:**

* Execute the below script in both the old and new database and copy the output to Excel sheet and compare the row count and it will be same.
* Don’t compare the size as it may be differ for partition tables as we created them again in destination.

SELECTSCHEMA\_NAME(tbl.schema\_id)as [Schema]

, tbl.Name

,Coalesce((Select pr.name

Fromsys.database\_principals pr

Where pr.principal\_id = tbl.principal\_id)

,SCHEMA\_NAME(tbl.schema\_id))as [Owner]

, tbl.max\_column\_id\_used as [Columns]

,CAST(CASE idx.index\_id WHEN 1 THEN 1 ELSE 0 ENDASbit)AS [HasClusIdx]

,Coalesce((Selectsum(spart.rows)fromsys.partitions spart

Where spart.object\_id= tbl.object\_idand spart.index\_id < 2), 0)AS [RowCount]

,Coalesce((SelectCast(v.low/1024.0 asfloat)

\*SUM(a.used\_pages -CASEWHEN a.type<> 1 THEN a.used\_pages WHEN p.index\_id < 2 THEN a.data\_pages ELSE 0 END)

FROMsys.indexesas i

JOINsys.partitionsas p ON p.object\_id= i.object\_idand p.index\_id = i.index\_id

JOINsys.allocation\_unitsas a ON a.container\_id = p.partition\_id

Where i.object\_id= tbl.object\_id)

, 0.0)AS [IndexKB]

,Coalesce((SelectCast(v.low/1024.0 asfloat)

\*SUM(CASEWHEN a.type<> 1 THEN a.used\_pages WHEN p.index\_id < 2 THEN a.data\_pages ELSE 0 END)

FROMsys.indexesas i

JOINsys.partitionsas p ON p.object\_id= i.object\_idand p.index\_id = i.index\_id

JOINsys.allocation\_unitsas a ON a.container\_id = p.partition\_id

Where i.object\_id= tbl.object\_id)

, 0.0)AS [DataKB]

, tbl.create\_date, tbl.modify\_date

FROMsys.tablesAS tbl

INNERJOINsys.indexesAS idx ON (idx.object\_id= tbl.object\_idand idx.index\_id < 2)

INNERJOINmaster.dbo.spt\_values v ON (v.number=1 and v.type='E')orderby name

**Source**: <https://gallery.technet.microsoft.com/scriptcenter/RestoreMigrate-partition-f61e4afc>