

SedonaOffice SEFT Log Truncation/DB Growth Post SedonaOffice 6.2.0.17 - 6.2.0.21

11/20/2025 11:47 am EST

00142997 / 00146413 / 00143478

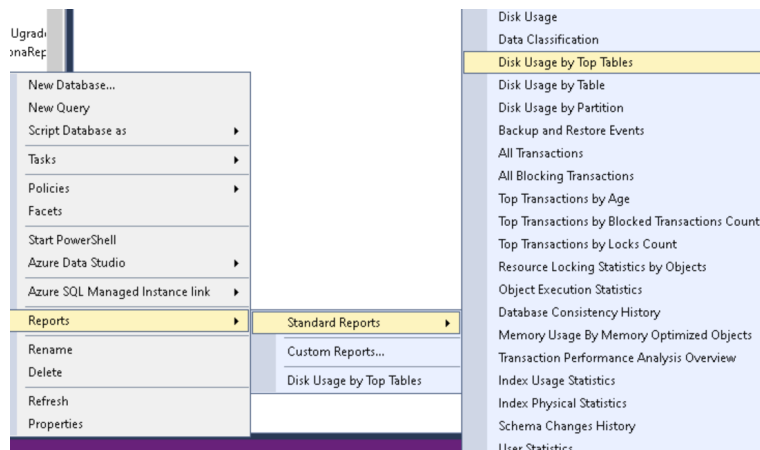
As of SedonaOffice ver 6.2.0.17 The SEFT log is aggregating a lot of data to compensate for several of the changes Forte has made around EFT/ACH Transactions, response codes, timing, reversals, and Z transactions. Accounting for the various conditions, states, and codes of transactions requires regular check ins with Forte to validate data. Development is working to scale the information down more and more in each future release.

Customers that have a lot of transactions on the SO versions above may see their database grow due and need assistance with shrinking this particular table. This guide goes over the steps to do so.

NOTE: Before truncating the SEFT Log Table, ensure the customer has nightly backups in the event they have any Payment Processing issues that require review.

You can access the information via SSMS UI by

Right-clicking on the production database > Reports > Standard Reports > Disk Usage by Top Tables



Disk Usage by To...AM - SEDONAOFFICE

SQL Server

on SEDONAOFFICE at 6/11/2025 8:55:15 AM

This report provides detailed data on the utilization of disk space by top 1000 tables within the Database. The report does not provide data for memory optimized tables.

Table Name	# Records	Reserved (KB)	Data (KB)	Indexes (KB)	Unused (KB)
dbo_SEFT_Service_Log	98,069,390	69,663,968	69,399,184	255,488	9,296
dbo_GL_Register	2,764,631	2,177,728	560,224	1,570,232	47,272
SedonaUser.GLRegBKUPTxt106652	2,533,208	507,784	507,736	8	40
dbo_bak_glregister_mjh_012924	2,496,007	500,104	500,088	8	8
dbo_AR_Invoice	530,968	645,840	384,272	226,024	35,544
dbo_tkt165725_gl4	1,703,203	339,656	339,592	8	56
dbo_SY_Edit_Log	2,607,386	903,040	335,536	559,040	8,464
dbo_GL_RegisterLog	1,436,416	331,808	330,104	1,392	312
dbo_AR_Customer_Notes	473,149	353,816	305,568	39,528	8,720
dbo_AR_Cycle_Detail	683,999	396,856	274,600	110,312	11,944
dbo_AR_Invoice_Item	808,649	506,464	254,272	234,464	17,728
dbo_ARInvBKUP20210302_1257pPT_SH	369,696	193,608	193,560	8	40
dbo_AR_Customer_Aging_RetroActive	1,015,066	360,832	178,928	181,736	168
dbo_tkt165725_inv4	329,583	172,232	172,168	8	56
dbo_AR_Collection_Event	608,833	226,064	160,848	56,456	8,760
dbo_AR_ACH	403,746	227,504	157,576	65,296	4,632
dbo_SS_Integration_Log	120,987	138,976	137,936	160	880
dbo_EM_Message	7,493	126,880	124,552	1,256	1,072
dbo_RM_Batch	137,104	95,192	91,352	208	3,632
dbo_tkt165725sach4	247,823	82,648	82,496	8	144
dbo_ARACH20200201	242,907	80,600	80,472	8	120
dbo_GL_Posting	916,425	72,160	70,072	160	1,928
dbo_xqCustInfo	36,004	96,072	69,728	8	26,336
dbo_AR_Deposit_Check	502,701	216,496	65,992	131,080	19,424

Manual SEFT Log Script Check and Truncation below

-- Get table Space sizes

SELECT

t.NAME AS TableName,

SUM(p.rows) AS RowCounts,

SUM(a.total_pages) * 8 / 1024 AS TotalSpaceMB,

SUM(a.used_pages) * 8 / 1024 AS UsedSpaceMB,

ROUND(SUM(a.total_pages) * 8.0 / 1024 / 1024, 3) AS TotalSpaceGB,

ROUND(SUM(a.used_pages) * 8.0 / 1024 / 1024, 3) AS UsedSpaceGB,

ROUND((SUM(a.total_pages) - SUM(a.used_pages)) * 8.0 / 1024 / 1024, 3) AS FreeSpaceGB,

CASE

WHEN SUM(a.total_pages) > 0

THEN ROUND((SUM(a.used_pages) * 100.0 / SUM(a.total_pages)), 2)

ELSE 0

END AS UsedSpacePercentage

```

FROM sys.tables t

JOIN sys.indexes i ON t.OBJECT_ID = i.object_id

JOIN sys.partitions p ON i.object_id = p.OBJECT_ID AND i.index_id = p.index_id

JOIN sys.allocation_units a ON p.partition_id = a.container_id

GROUP BY t.NAME

ORDER BY TotalSpaceGB DESC;

```

-- Get the Overall Databases Size summary

```

WITH DatabaseSpace AS (

SELECT

SUM(CAST(size AS BIGINT) * 8.0 / 1024) AS AllocatedSizeMB,

SUM(CAST(size AS BIGINT) * 8.0 / 1024 / 1024) AS AllocatedSizeGB

FROM sys.database_files

),

UsedSpace AS (

SELECT

SUM(a.total_pages) * 8.0 / 1024 AS UsedSizeMB,

SUM(a.total_pages) * 8.0 / 1024 / 1024 AS UsedSizeGB

FROM sys.allocation_units a

)

SELECT

DB_NAME() AS DatabaseName,

ROUND(ds.AllocatedSizeMB, 2) AS AllocatedSizeMB,

ROUND(ds.AllocatedSizeGB, 3) AS AllocatedSizeGB,

ROUND(us.UsedSizeMB, 2) AS UsedSizeMB,

ROUND(us.UsedSizeGB, 3) AS UsedSizeGB,

```

```
ROUND(ds.AllocatedSizeMB - us.UsedSizeMB, 2) AS FreeSizeMB,  
  
ROUND(ds.AllocatedSizeGB - us.UsedSizeGB, 3) AS FreeSizeGB,  
  
ROUND((us.UsedSizeMB / ds.AllocatedSizeMB) * 100, 2) AS UsedPercentage  
  
FROM DatabaseSpace ds, UsedSpace us;
```

```
--Action
```

```
--1. truncate SEFT_Service_Log table
```

```
--select * from ss_version --6.2.0.19
```

```
truncate table SEFT_Service_Log
```

If Customer has had issue occur more than once in 90 days they may benefit from the stored procedure Emeka D created

```
Create PROCEDURE dbo.Sp_TruncateSEFTLog
```

```
AS
```

```
BEGIN
```

```
SET NOCOUNT ON;
```

```
-- Clean up any existing temp tables
```

```
IF OBJECT_ID('tempdb..#LargestTable') IS NOT NULL DROP TABLE #LargestTable;
```

```
IF OBJECT_ID('tempdb..#DataBaseSummaryBefore') IS NOT NULL DROP TABLE #DataBaseSummaryBefore;
```

```
IF OBJECT_ID('tempdb..#DataBaseSummaryAfter') IS NOT NULL DROP TABLE #DataBaseSummaryAfter;
```

```
-- Get the Largest table info
```

```
SELECT TOP (1)
```

```

t.NAME AS TableName,

SUM(p.rows) AS RowCounts,

ROUND(SUM(a.total_pages) * 8.0 / 1024 / 1024, 3) AS TotalSpaceGB,

ROUND(SUM(a.used_pages) * 8.0 / 1024 / 1024, 3) AS UsedSpaceGB,

ROUND((SUM(a.total_pages) - SUM(a.used_pages)) * 8.0 / 1024 / 1024, 2) AS FreeSpaceGB,

CASE

    WHEN SUM(a.total_pages) > 0 THEN ROUND((SUM(a.used_pages) * 100.0 / SUM(a.total_pages)), 2)

    ELSE 0

END AS UsedSpacePercentage

INTO #LargestTable

FROM sys.tables t

JOIN sys.indexes i ON t.OBJECT_ID = i.object_id

JOIN sys.partitions p ON i.object_id = p.OBJECT_ID AND i.index_id = p.index_id

JOIN sys.allocation_units a ON p.partition_id = a.container_id

GROUP BY t.NAME

ORDER BY TotalSpaceGB DESC;

-- Check if the largest table is SEFT_Service_Log

IF NOT EXISTS (SELECT 1 FROM #LargestTable WHERE TableName = 'SEFT_Service_Log')

BEGIN

    SELECT 'Largest table is ' + TableName + '. Aborting!' AS Message FROM #LargestTable;

    RETURN;

END

-- Get the Overall Databases Size summary before Truncation

;WITH DatabaseSpace AS (

```

```

SELECT SUM(CAST(size AS BIGINT) * 8.0 / 1024) AS AllocatedSizeMB,

        SUM(CAST(size AS BIGINT) * 8.0 / 1024 / 1024) AS AllocatedSizeGB

FROM sys.database_files

),

UsedSpace AS (

SELECT SUM(a.total_pages) * 8.0 / 1024 AS UsedSizeMB,

        SUM(a.total_pages) * 8.0 / 1024 / 1024 AS UsedSizeGB

FROM sys.allocation_units a

)

SELECT

DB_NAME() AS DatabaseName,

ROUND(ds.AllocatedSizeGB, 3) AS AllocatedSizeGB,

ROUND(us.UsedSizeGB, 3) AS UsedSizeGB,

ROUND(ds.AllocatedSizeGB - us.UsedSizeGB, 3) AS FreeSizeGB,

ROUND((us.UsedSizeMB / ds.AllocatedSizeMB) * 100, 2) AS UsedPercentage

INTO #DataBaseSummaryBefore

FROM DatabaseSpace ds, UsedSpace us;


-- Truncate the SEFT_Service_Log table

TRUNCATE TABLE SEFT_Service_Log;


-- Introduce a 10-second delay to allow database space to update

WAITFOR DELAY '00:00:10';


-- Get the Overall Databases Size summary after Truncation

WITH DatabaseSpace AS (

```

```

SELECT SUM(CAST(size AS BIGINT) * 8.0 / 1024) AS AllocatedSizeMB,

        SUM(CAST(size AS BIGINT) * 8.0 / 1024 / 1024) AS AllocatedSizeGB

FROM sys.database_files

),

UsedSpace AS (

SELECT SUM(a.total_pages) * 8.0 / 1024 AS UsedSizeMB,

        SUM(a.total_pages) * 8.0 / 1024 / 1024 AS UsedSizeGB

FROM sys.allocation_units a

)

SELECT

DB_NAME() AS DatabaseName,

ROUND(ds.AllocatedSizeGB, 3) AS AllocatedSizeGB,

ROUND(us.UsedSizeGB, 3) AS UsedSizeGB,

ROUND(ds.AllocatedSizeGB - us.UsedSizeGB, 3) AS FreeSizeGB,

ROUND((us.UsedSizeMB / ds.AllocatedSizeMB) * 100, 2) AS UsedPercentage

INTO #DataBaseSummaryAfter

FROM DatabaseSpace ds, UsedSpace us;

-- Gather Stats report

-- Truncated table information

SELECT

'Truncated Table' AS Information,

TableName,

RowCounts,

RTRIM(CAST(TotalSpaceGB AS FLOAT)) AS TotalSpaceGB,

```

```
RTRIM(CAST(UsedSpaceGB AS FLOAT)) AS UsedSpaceGB,  
  
RTRIM(CAST(FreeSpaceGB AS FLOAT)) AS FreeSpaceGB,  
  
RTRIM(CAST(UsedSpacePercentage AS FLOAT)) AS UsedSpacePercentage  
FROM #LargestTable;
```

-- Database Summary before and after truncation

```
SELECT  
  
DatabaseName,  
  
RTRIM(CAST(AllocatedSizeGB AS FLOAT)) AS AllocatedSizeGB,  
  
RTRIM(CAST(UsedSizeGB AS FLOAT)) AS UsedSizeGB,  
  
RTRIM(CAST(FreeSizeGB AS FLOAT)) AS FreeSpaceGB,  
  
RTRIM(CAST(UsedPercentage AS FLOAT)) AS UsedPercentage,  
  
(SELECT 'SedonaOffice Version: ' + version FROM ss_version) AS [Version],  
  
'Before Truncation' AS Status  
FROM #DataBaseSummaryBefore
```

UNION ALL

```
SELECT  
  
DatabaseName,  
  
RTRIM(CAST(AllocatedSizeGB AS FLOAT)) AS AllocatedSizeGB,  
  
RTRIM(CAST(UsedSizeGB AS FLOAT)) AS UsedSizeGB,  
  
RTRIM(CAST(FreeSizeGB AS FLOAT)) AS FreeSpaceGB,  
  
RTRIM(CAST(UsedPercentage AS FLOAT)) AS UsedPercentage,  
  
(SELECT 'SedonaOffice Version: ' + version FROM ss_version) AS [Version],  
  
'After Truncation' AS Status
```


FROM #DataBaseSummaryAfter;

END

GO
