# Run the steps below on each WSUS in the hierarchy.  When performing a cleanup and removing items from WSUS servers, you should start at the bottom of the hierarchy.

Steps 9-12 you will most likely need to run multiple times due to the large number of declined updates.  After each run, execute the Count of Declined Updates query to verify the number is going down and to monitor progress.  Step 8 may end in error each time you run it, hence the reason for steps 9-12 and running it again.  This is normal and to be expected.  Some of these steps, especially #8 could take several hours to run.

This is a long and repetitive, but I have seen it resolve many client issues with scanning.  Please let me know if you have any questions or issues!

Update Count

use SUSDB;

DECLARE @numberOfMatch INT

DECLARE @tmpTable TABLE (

    name VARCHAR(25)

)

INSERT INTO @tmpTable

EXEC spGetObsoleteUpdatesToCleanup

SELECT @numberOfMatch = @@ROWCOUNT

select

(Select count (\*) from vwMinimalUpdate ) 'Total Updates',

(Select count (\*) from vwMinimalUpdate where declined=0) as 'Live Updates',

(Select count (\*) from vwMinimalUpdate where IsSuperseded =1) as 'Superseded',

(Select count (\*) from vwMinimalUpdate where IsSuperseded =1 and declined=0) as 'Superseded but not declined',

(Select count (\*) from vwMinimalUpdate where declined=1) as 'Declined',

(Select count (\*) from vwMinimalUpdate where IsSuperseded =1 and declined=1) 'Superseded & Declined',

(select Count(\*)  From @tmpTable ) 'Obsolete Updates Needed to be cleaned'

Maintenance

* 1. Run this SQL first.  [Slow performance of the spDeleteUpdate procedure - Configuration Manager | Microsoft Docs](https://nam06.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdocs.microsoft.com%2Fen-US%2Ftroubleshoot%2Fmem%2Fconfigmgr%2Fspdeleteupdate-slow-performance&data=04%7C01%7Cbgonzalez%40microsoft.com%7C622722fcfda442d70ae508d9db815939%7C72f988bf86f141af91ab2d7cd011db47%7C1%7C0%7C637782170897172316%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000&sdata=zpWmuG1HXRslxKGVSevet7xHboEBy%2BhrofJ5crZYyZs%3D&reserved=0)
  2. Shrink Files
  3. Shrink Database
  4. **--Before starting cleanup, reindex and update stats on the SUSDB**

**--Run First to reindex:**

EXEC sp\_MSforeachtable @command1="SET QUOTED\_IDENTIFIER ON;ALTER INDEX ALL ON ? REBUILD;"

**--Run Second to update statistics:**

Exec sp\_msforeachtable "UPDATE STATISTICS ? WITH FULLSCAN, COLUMNS"

* 1. **First, cleanup of sync history. If there is a large number of syncs, it can crash the WSUS console.**

USE SUSDB

GO

DELETE FROM tbEventInstance WHERE EventNamespaceID = '2' AND EVENTID IN ('381', '382', '384', '386', '387', '389')

* 1. **--Second, cleanup of all superseded updates older than 30 days OR based on what you set it to cleanup**

USE SUSDB

GO

DELETE FROM tbEventInstance WHERE EventNamespaceID = '2' AND EVENTID IN ('381', '382', '384', '386', '387', '389')

DECLARE @thresholdDays INT = 90 -- Specify the number of days between today and the release date for which the superseded updates must not be declined. This should match configuration of supersedence rules in SUP component properties, if ConfigMgr is being used with WSUS.

DECLARE @testRun BIT = 0 -- Set this to 1 to test without declining anything.

-- There shouldn't be any need to modify anything after this line.

DECLARE @uid UNIQUEIDENTIFIER

DECLARE @title NVARCHAR(500)

DECLARE @date DATETIME

DECLARE @userName NVARCHAR(100) = SYSTEM\_USER

DECLARE @count INT =0

DECLARE DU CURSOR FOR

SELECT MU.UpdateID, U.DefaultTitle, U.CreationDate FROM vwMinimalUpdate MU

JOIN PUBLIC\_VIEWS.vUpdate U ON MU.UpdateID = U.UpdateId

WHERE MU.IsSuperseded = 1 AND MU.Declined = 0 AND MU.IsLatestRevision = 1

AND MU.CreationDate < DATEADD(dd,-@thresholdDays,GETDATE())

ORDER BY MU.CreationDate

PRINT 'Declining superseded updates older than ' + CONVERT(NVARCHAR(5),@thresholdDays) + ' days.' + CHAR(10)

OPEN DU

FETCH NEXT FROM DU INTO @uid, @title, @date

WHILE (@@FETCH\_STATUS > - 1)

BEGIN

SET @count = @count +1

PRINT 'Declining update ' + CONVERT(NVARCHAR(50),@uid) + ' (Creation Date ' + CONVERT(NVARCHAR(50),@date) + ') - ' +@title + ' ...'

IF @testRun = 0

EXEC spDeclineUpdate @updateID = @uid, @adminName = @userName, @failIfReplica = 1

FETCH NEXT FROM DU INTO @uid, @title, @date

END

CLOSE DU

DEALLOCATE DU

PRINT CHAR(10) + 'Attempted to decline ' + CONVERT(NVARCHAR(10), @count) + ' updates.'

* 1. **--Third, cleanup of all obsolete updates**

DECLARE @var1 INT

DECLARE @msg nvarchar(100)

CREATE TABLE #results (Col1 INT)

        INSERT INTO #results(Col1) EXEC spGetObsoleteUpdatesToCleanup

DECLARE WC Cursor

        FOR

        SELECT Col1 FROM #results

OPEN WC

        FETCH NEXT FROM WC

        INTO @var1

        WHILE (@@FETCH\_STATUS > -1)

        BEGIN SET @msg = 'Deleting' + CONVERT(varchar(10), @var1)

        RAISERROR(@msg,0,1) WITH NOWAIT EXEC spDeleteUpdate @localUpdateID=@var1

        FETCH NEXT FROM WC INTO @var1 END

CLOSE WC

        DEALLOCATE WC

        DROP TABLE #results

* 1. **Run PowerShell ISE As Admin**

**WSUS Cleanup Wizard via PowerShell**

**Save the below as a Cleanup.PS1 then you can later setup to run on a regular basis via a Scheduled Task**

[reflection.assembly]::LoadWithPartialName("Microsoft.UpdateServices.Administration") | out-null

$wsus = [Microsoft.UpdateServices.Administration.AdminProxy]::GetUpdateServer();

$cleanupScope = new-object Microsoft.UpdateServices.Administration.CleanupScope;

$cleanupScope.DeclineSupersededUpdates = $true

$cleanupScope.DeclineExpiredUpdates = $true

$cleanupScope.CleanupObsoleteUpdates = $true

$cleanupScope.CompressUpdates = $true

$cleanupScope.CleanupObsoleteComputers = $true

$cleanupScope.CleanupUnneededContentFiles = $true

$cleanupManager = $wsus.GetCleanupManager();

$cleanupManager.PerformCleanup($cleanupScope);

* 1. **Save the below as a Cleanup\_Declined.PS1 then you can later setup to run on a regular basis via a Scheduled Task**

**Run PowerShell ISE As Admin**

[reflection.assembly]::LoadWithPartialName("Microsoft.UpdateServices.Administration")

$wsus = [Microsoft.UpdateServices.Administration.AdminProxy]::GetUpdateServer();

$wsus.GetUpdates() | Where {$\_.IsDeclined -eq $true} | ForEach-Object {$wsus.DeleteUpdate($\_.Id.UpdateId.ToString()); Write-Host $\_.Title removed }

* 1. Shrink Files
  2. Shrink Database

* 1. **Once cleanup is run, reindex and update stats on the SUSDB**

**--Run First to reindex:**

EXEC sp\_MSforeachtable @command1="SET QUOTED\_IDENTIFIER ON;ALTER INDEX ALL ON ? REBUILD;"

**--Run Second to update statistics:**

Exec sp\_msforeachtable "UPDATE STATISTICS ? WITH FULLSCAN, COLUMNS"