-- Check existing temporary table

IF OBJECT\_ID('##XE\_Errors') IS NOT NULL DROP TABLE ##XE\_Errors;

DECLARE

@SQLDataRoot NVARCHAR(1024),

@filename NVARCHAR(1024),

@metadatafile NVARCHAR(1024),

@sql NVARCHAR(MAX)

EXEC master..xp\_instance\_regread

@rootkey='HKEY\_LOCAL\_MACHINE',

@key='SOFTWARE\Microsoft\MSSQLServer\Setup',

@value\_name='SQLDataRoot',

@value=@SQLDataRoot OUTPUT

SELECT @filename = @SQLDataRoot + N'\Log\what\_queries\_are\_failing.xel'

SELECT @metadatafile = @SQLDataRoot + N'\Log\what\_queries\_are\_failing.xem'

SELECT @sql =

'CREATE EVENT SESSION

what\_queries\_are\_failing

ON SERVER

ADD EVENT sqlserver.error\_reported

(

ACTION (sqlserver.sql\_text, sqlserver.tsql\_stack, sqlserver.database\_id, sqlserver.username)

WHERE ([severity]> 10 AND ([error\_number]<>2812 AND [message] NOT LIKE N''%CUSTOM''))

)

ADD TARGET package0.asynchronous\_file\_target

(SET filename = ''' + @filename + ''' ,

metadatafile = ''' + @metadatafile + ''',

max\_file\_size = 5,

max\_rollover\_files = 5)

WITH (MAX\_DISPATCH\_LATENCY = 5SECONDS)'

--Create an extended event session

IF EXISTS (SELECT TOP 1 1 FROM sys.server\_event\_sessions WHERE [name] = N'what\_queries\_are\_failing')

BEGIN

DROP EVENT SESSION what\_queries\_are\_failing ON SERVER

EXEC (@sql)

END

ELSE

BEGIN

EXEC (@sql)

END

-- Start the EX session

ALTER EVENT SESSION what\_queries\_are\_failing

ON SERVER STATE = START

-- Stop and drop the XE session

ALTER EVENT SESSION what\_queries\_are\_failing ON SERVER STATE = STOP

DROP EVENT SESSION [what\_queries\_are\_failing] ON SERVER;

-- Create temporary table with XE errors

SELECT @sql = '

;WITH events\_cte AS (

SELECT

DATEADD(mi,

DATEDIFF(mi, GETUTCDATE(), CURRENT\_TIMESTAMP),

xevents.event\_data.value(''(event/@timestamp)[1]'', ''datetime2'')) AS [err\_timestamp],

xevents.event\_data.value(''(event/data[@name="severity"]/value)[1]'', ''bigint'') AS [err\_severity],

xevents.event\_data.value(''(event/data[@name="error\_number"]/value)[1]'', ''bigint'') AS [err\_number],

xevents.event\_data.value(''(event/data[@name="message"]/value)[1]'', ''nvarchar(512)'') AS [err\_message],

xevents.event\_data.value(''(event/action[@name="sql\_text"]/value)[1]'', ''nvarchar(max)'') AS [sql\_text],

xevents.event\_data

FROM sys.fn\_xe\_file\_target\_read\_file

(''' + @SQLDataRoot + '\Log\what\_queries\_are\_failing\*.xel'',

''' + @SQLDataRoot + '\Log\what\_queries\_are\_failing\*.xem'',

NULL, NULL)

CROSS APPLY (SELECT CAST(event\_data AS XML) AS event\_data) AS xevents

)

SELECT \* INTO ##XE\_Errors FROM events\_cte ORDER BY err\_timestamp'

EXEC(@sql)

IF EXISTS (SELECT TOP 1 1 FROM ##XE\_Errors)

-- Send e-mail

BEGIN

EXEC msdb.dbo.sp\_send\_dbmail

@recipients = 'e-mail@domain', -- separate is ;

@subject = 'Attention! We have errors.'

END

ELSE

-- Drop XE files and table with XE errors

BEGIN

-- ALTER EVENT SESSION [what\_queries\_are\_failing] ON SERVER DROP TARGET package0.event\_file

IF OBJECT\_ID('##XE\_Errors') IS NOT NULL DROP TABLE ##XE\_Errors;

END