**Automate Alerting for SQL Server Suspect Database Pages**

By: [Ben Snaidero](https://www.mssqltips.com/sqlserverauthor/74/ben-snaidero/)

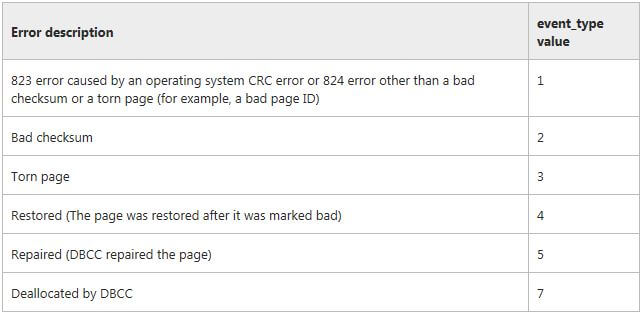
FROM: <https://www.mssqltips.com/sqlservertip/4166/automate-alerting-for-sql-server-suspect-database-pages/>

**Problem**

We suspect there are some issues with our disk subsystem and would like to be alerted by email if SQL Server encounters any corruption in any of the data files that reside on the disk in question. How can we add this automated monitoring?

**Solution**

Luckily for us, SQL Server 2005 introduced a new table in the msdb database called [suspect\_pages](https://msdn.microsoft.com/en-us/library/ms174425.aspx) which logs any pages which it suspects might be bad. Any time the database engine comes across a page id that it thinks might be bad, whether through a query reading a page, a DBCC operation or a backup/restore, the page id is added/updated in this table along with some details about the event. The details can be found in the event\_type column in this table and a description of these types is below. Notice that event\_types 4, 5 and 7 actually show that the issue has been resolved.



It's important to note that suspect\_pages has to be maintained by a DBA or someone with permissions to make updates in the msdb database. This table is limited to 1000 rows, so if there are records in the table that are no longer needed they should be deleted so new rows can be added if required. More on managing this table can be found [here](https://msdn.microsoft.com/en-us/library/ms191301.aspx).

Monitoring Setup

The first thing we'll need is a query to pull information from the suspect\_pages table. Since this table mainly contains non-descriptive ids, let's join it with [sys.databases](https://msdn.microsoft.com/en-us/library/ms178534.aspx) and [sys.master\_files](https://msdn.microsoft.com/en-us/library/ms186782.aspx) so our alert will contain everything we need to track down and resolve the issue. Below is the TSQL for this query.

select sp.database\_id "Database ID",

d.name "Database",

sp.file\_id "File ID",

mf.physical\_name "File",

sp.page\_id "Page ID",

case when sp.event\_type = 1 then '823 or 824 error other than a bad checksum or a torn page'

when sp.event\_type = 2 then 'Bad checksum'

when sp.event\_type = 3 then 'Torn Page'

when sp.event\_type = 4 then 'Restored (The page was restored after it was marked bad)'

when sp.event\_type = 5 then 'Repaired (DBCC repaired the page)'

when sp.event\_type = 7 then 'Deallocated by DBCC'

end as "Event Desc",

sp.error\_count "Error Count",

sp.last\_update\_date "Last Updated"

from msdb.dbo.suspect\_pages sp

inner join sys.databases d on d.database\_id=sp.database\_id

inner join sys.master\_files mf on mf.database\_id=sp.database\_id and mf.file\_id=sp.file\_id

Now that we have a query with all the information we need we just have to wrap this query with some logic to send out an email only when it actually finds data. Using database mail and some html formatting, samples of each can be found [here](https://msdn.microsoft.com/en-us/library/ms190307.aspx), we can produce a nicely formatted email alert. The TSQL code for this is below and can be added to a new or existing SQL job to monitor your instance.

declare @count integer

declare @tableHTML nvarchar(MAX);

declare @subj nvarchar(100)

select @count=count(1)

from msdb.dbo.suspect\_pages;

set @subj = 'Suspect Pages Found in ' + @@SERVERNAME;

set @tableHTML =

N'<H1>Suspect Pages Found in ' + @@SERVERNAME + ', details are below.</H1>' +

N'<table border="1">' +

N'<tr><th>Database ID</th><th>Database</th>' +

N'<th>File ID</th><th>File</th><th>Page ID</th>' +

N'<th>Event Desc</th><th>Error Count</th><th>Last Updated</th></tr>' +

cast ( ( select td = sp.database\_id, '',

td = d.name, '',

td = sp.file\_id, '',

td = mf.physical\_name, '',

td = sp.page\_id, '',

td = case when sp.event\_type = 1 then '823 or 824 error other than a bad checksum or a torn page'

when sp.event\_type = 2 then 'Bad checksum'

when sp.event\_type = 3 then 'Torn Page'

when sp.event\_type = 4 then 'Restored (The page was restored after it was marked bad)'

when sp.event\_type = 5 then 'Repaired (DBCC repaired the page)'

when sp.event\_type = 7 then 'Deallocated by DBCC'

end, '',

td = sp.error\_count, '',

td = sp.last\_update\_date

from msdb.dbo.suspect\_pages sp

inner join sys.databases d on d.database\_id=sp.database\_id

inner join sys.master\_files mf on mf.database\_id=sp.database\_id and mf.file\_id=sp.file\_id

for xml path('tr'), TYPE

) as nvarchar(max) ) +

N'</table>' ;

IF @count > 0

exec msdb.dbo.sp\_send\_dbmail

@recipients=N'testuser@test.com',

@body= @tableHTML,

@subject = @subj,

@body\_format = 'HTML',

@profile\_name ='testprofile'

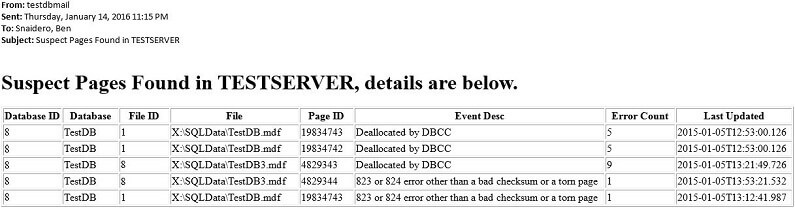
As an added benefit, assuming you would be running this at some interval whether it be daily/weekly/monthly whatever you decide is best for your environment, we could also piggy back on this job and do the cleanup of this table at the same time. Adding another step to this job that contains the following TSQL we can purge anything in this table that is older than 90 days. Note that you can change this purge logic however you see fit.

delete from msdb.dbo.suspect\_pages

where last\_update\_date < getdate()-90

Sample Output

Hopefully you check the suspect\_pages table in all of your instances and you don't find anything. Rather than have you go through the process of corrupting a page in one of your databases to see what the report looks like I'll include a sample of one below for reference.



**Next Steps**

* Read more on ways to fix corrupted pages:
  + [Minimize performance impact of SQL Server DBCC CHECKDB](https://www.mssqltips.com/sqlservertip/2399/minimize-performance-impact-of-sql-server-dbcc-checkdb/)
  + [SQL Server database mirroring automatic page repair](https://www.mssqltips.com/sqlservertip/3660/sql-server-database-mirroring-automatic-page-repair/)
* Read [more tips on monitoring](https://www.mssqltips.com/sql-server-tip-category/54/monitoring/) SQL Server
* Read [more tips on database mail](https://www.mssqltips.com/sql-server-tip-category/44/database-mail/)

Last Update: 1/25/2016