ITPro with deep vertical Knowledge

RECENT

PRACTICAL SQL SERVER

# SQL Server Database Corruption, Part V: Storage Problem Alerts

overachiever | Feb 24, 2012

DOWNLOAD the CODE

Continuing on from our last post where we looked at setting up CHECKSUM verification as one means of helping more readily detect corruption early-on, we'll now take a look at how to have SQL Server notify you of any instances where it runs into problems with the IO subsystem.

## Configuring Alerts for Storage Problems

In addition to being able to quickly and easily detect corruption problems once CHECKSUM verification has been enabled, SQL Server is also robust enough to be able to watch out for other kinds of disk errors. In this regard, SQL Server can be set up to pro-actively send alerts when it encounters simple errors such as read-write failures, or even when it proactively attempts to issue multiple re-read attempts against 'sticky' or problematic data (that is typically a clear sign of impending problems at the IO subsystem level).

Therefore, to take advantage of SQL Server's innate ability to track these kinds of errors, all you need to do is:

1. Ensure that you've got SQL Server's Database Mail functionality set up and properly configured. (Setting this up is outside the scope of this document, but is easy enough to do by right-clicking on the node and walking through a wizard where you establish connectivity and settings that enable SQL Server to 'talk' to an SMTP server within your organization).

PartV_Figure1

PartV_Figure1

2. Create at least one SQL Server Agent Operator – as per the screenshot below:

PartV_Figure2

PartV_Figure2

And, again, the creation of operators is outside the scope of this document – but is easy enough to do by right-clicking the Operators node and walking through the process of creating at least a 'catch all' operator such as 'Admins' or 'General' that can be mapped to something like alerts@yourcompany.com or some other monitored email address where alerts can be sent.

3. Configure SQL Server to forward instances of common IO errors 'up' to a SQL Server Agent Operator so that you can be notified of errors AS they occur.

To implement this last step, execute the following code on your SQL Server to create the initial alerts needed to forward SQL Server errors 824, 825, and 826 as they occur:

```
USE msdb
GO

EXEC msdb.dbo.sp_add_alert @name = N'823 - Read/Write Failure',
    @message_id = 823,
    @severity = 0,
    @enabled = 1,
    @delay_between_responses = 0,
    @include_event_description_in = 1
GO

EXEC msdb.dbo.sp_add_alert @name = N'824 - Page Error',
    @message_id = 824,
    @severity = 0,
    @enabled = 1,
    @delay_between_responses = 0,
    @include_event_description_in = 1
GO

EXEC msdb.dbo.sp_add_alert @name = N'825 - Read-Retry Required',
    @message_id = 825,
```

```
    @message_id = 825,
    @severity = 0,
    @enabled = 1,
    @delay_between_responses = 0,
    @include_event_description_in = 1
GO
```

Then, once these alerts are configured, refresh the Alerts node and then right click on each of them, select Properties and then set up the Response tab to Notify operators when an error occurs – as per the following screen shot:

PartV_Figure3

PartV_Figure3

By means of this approach, you'll be alerted any time SQL Server encounters disk failures – which are almost always related to the potential for corruption or associated with processes and situations that create corruption.

### Up Next

Now that you've configured SQL Server to alert you when it encounters storage problems, you're now better able to be alerted to problems with corruption (or other disk subsystem issues) when the happen – instead of letting them effectively go unnoticed until it becomes harder (or even too late) to deal with them. But, there's much more that you can do to stay apprised of the potential for corruption, such as schedule regular checks – which we'll look at in our next post.

Part VI: Regular Corruption Checks

TAGS: MICROSOFT SQL SERVER CLOUD & DATA CENTER

0 COMMENTS

```
SELECT STRING_AGG(CHAR(n), '')
  WITHIN GROUP (ORDER BY CHAR(n) COLLATE Latin1_General_CS_
FROM dbo.GetNums(0, 255)
WHERE CHAR(n) COLLATE Latin1_General_CS_AS LIKE '[A-Z]';


SELECT STRING_AGG(CHAR(n), '')
  WITHIN GROUP (ORDER BY CHAR(n) COLLATE Latin1_General_BIN
FROM dbo.GetNums(0, 255)
WHERE CHAR(n) COLLATE Latin1_General_BIN LIKE '[A-Z]';
```

BUSINESS INTELLIGENCE > MICROSOFT SQL SERVER

## The Challenge Is on: Maintain a Custom Letter-Based Sequence

If you need to implement a custom sequence solution, and you want it to perform as well as possible, you need to ask yourself which guarantees the solution needs to provide versus which it doesn't.

Itzik Ben-Gan | Apr 01, 2018

Recently I delivered an advanced T-SQL class in Europe, and one of my students, Nicolas Barreau of Avanade France SAS, presented a nice little puzzle. The task wasn't very complex, but it was interesting and fun to work on, and I hope you will enjoy working on it, too.

### The Challenge

The challenge Nicola presented was to maintain a custom, letter-based, cycling, gap-allowing, sequence, with values in the form LLL, where L is an alphabet letter in the range A through Z (26 symbols). Nicola's company uses the sequence value as a match flag for settlement, for instance,

between payment and invoice. The first value produced should be AAA, and it should advance as if the letters were digits in base 26, like so:

AAA

AAB

…

AAY

READ MORE

TAGS: SOFTWARE DEVELOPMENT

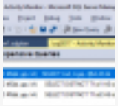
0 COMMENTS

RELATED

The Challenge Is on: Multi-Way Number Partitioning with T-SQL and SQL CLR
FEB 21, 2018

SQL Server: Avoiding a Sort with Descending Order
JAN 19, 2018

SQL Server: Avoiding a Sort with Merge Join Concatenation
DEC 18, 2017



T-SQL Puzzle Challenge: Grouping Connected Items
NOV 27, 2017

Sign up for the ITPro Update newsletter.

Email address

United States

SIGN UP ▸

SECURITY > ENDPOINT SECURITY

# Apple Is Ready for the Tech Backlash

The only real guarantee that companies won't go overboard in invading customers' privacy is that they have large revenue streams which make it an unnecessary risk.

Bloomberg | Mar 30, 2018

(Bloomberg View) --It's easy to understand Apple CEO Tim Cook's somewhat self-righteous comments on the Facebook-Cambridge Analytica scandal: His company has never emphasized customer data monetization, and privacy has been one of its persistent selling points. Apple still collects our data — and reserves the right to share it — but the company has a much better track record than newer tech titans.

"The truth is, we could make a ton of money if we monetized our customer — if our customer was our product," Cook said on Wednesday. "We've elected not to do that." Asked what he'd do if he

were in Facebook CEO Mark Zuckerberg's shoes, he replied, "I wouldn't be in this situation."

In reality, Apple collects more information about its customers than Facebook because it offers more products and services. It knows everything from an iPhone's location to its owner's taste in movies and music. It's all in the company's privacy statement, including the frank admission that Apple will share the data with "strategic partners that work with Apple to provide products and services, or that help Apple market to customers" and with law enforcement. Apple also lets applications that run on its platform (including Facebook) hoover up data. And it does have a small ad platform that allows targeted advertising.

Apple's closest equivalent in terms of data collection, sharing and "pass-through" to app developers is Google, which offers roughly the same array of services (plus search) and also owns mobile and desktop operating systems. The difference is that Apple has imposed some voluntary restrictions on itself: For example, it anonymizes location data (unless you're using the "Find My iPhone" feature) and generally employs "differential privacy" — a cryptography-based practice of obtaining usage and preference data without linking it to specific users. Google and Microsoft also use that to some degree, and a co-inventor of differential privacy, Cynthia Dwork, has even worked for Microsoft Research, but Apple was the first to market the technique as an advantage for its customers.

Cook clearly thinks of these self-restrictions as a good way of staving off external regulation. On Wednesday, he scolded Facebook for not being as far-sighted. "I think the best regulation is no regulation, is self-regulation," he said. "However, I think we're beyond that here."

As a user, however, I wouldn't put much trust in any company's ability to regulate itself. Cook has been wary of monetizing Apple's user base, but he won't be CEO forever. And people do change their mind, especially when it comes to picking up money that's there for the taking. Regulation — such as the EU's General Data Protection regulation, launching in May — is a more reliable way to ensure a level playing field. That's why Google, Apple and Microsoft all have good, convenient privacy controls which allow the user to turn off various kinds of data collection, erase data that companies already have and opt out of targeted advertising. Facebook, too, promises to put such controls in place soon.

But even external regulation isn't a reliable guarantee of privacy. Most people won't fiddle with the controls because they may not even realize what all the fuss is about. And by default, companies with advertising-based business models will want to collect as much information as they can and