Steve Stedman
Freelance SQL Server Consultant
http://stevestedman.com

SQL Server Row Data Linking to Off Row Data

The question came up as how to find a link from blog storage that is corrupt back to the table and row that contains that data.

The is no link from the blob storage back to the table and row, but this is a link from the data page containing the table and row off to the blob data.

First let's start with row data and off row data. When SQL Server stores a row that contains variable size data like a VARCHAR, if all the lengths of all the values in the row are less than about 8k, SQL Server stores that entire row in a single data page. If a row contains a set of data larger than 8K, then some if it can be moved off row and stored in blob storage.

Picture the following table:

If you were to put small values into the varchar columns then the entire row would fit in a single data page (8k). But if you put 4k of data into one varchar(max) and 8k of data into the next varchar(max) then it will not fit in a page. One of the varchar values will be moved off page into a page that contains blob storage. No rocket science here yet, but when you want to find the relationship between these page it gets difficult.

Using the undocumented DBCC IND command you can see that the table (clustered index) is comprised of 2 data pages, the first one of type 10 which is the IAM page or Index Allocation Map, and the secon page of page type 1 which is a standard data row.

Next we add another row.

And we can see that both rows are still only using the same 2 pages.

So now what happens if we insert a row that is larger than the page?

We get two more pages added as shown by DBCC IND.

We will skip the new row of page type 10 for the IAM page, and take a look at the new row of page type 3 for LOB Data.

We use the undocumented DBCC PAGE command, which must first have trace flag 3604 turned on to be able to use it. The output shows us the contents of the page, which contains the 8000 b's inserted in our insert statement above.

Which doesn't contain any links back to page 328 or the page that contains our row.

Ok, now let's take a look at page 328 to see if there are links from that page to page 336 containing blob data.

Looking through the several pages of output we see several things. First the page header which looks like a normal page header for a data page.

The scrolling down we see the test row data that we inserted for the first row, and for the second row.

Followed by the third row that was had one column too big to fit into the page.

And if we page all the way to the end of the third row of data we can see the link to the page with the lob data.

The RowID of 1:336:0 references file 1, page 336, offset 0, which is where you will find the large block of bbbbbb data that we inserted as part of our 3rd row.

With the above example you can see that you can jump from row data to the page that contains the blob data, but not the other direction.

Or can you?

Here is how I would do it.

Dump the output of DBCC IND into a temp table. The have a cursor loop through all the page of type 1 (data page).

For each of those rows run something similar to this to dump the output into a temp table and find the off row data links

And there you go, you can find the source object associated with a blob page.

I hope that nobody ever has to use this because that may mean that you are in a heap of trouble. If you are and you need help with your database corruption repair, please reach out to me for help. http://StedmanSolutions.com