

Re: WRITETEXT – I'm utterly confused By BOL

Source:

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.programming/2004-07/4058.html>

From: Dan Guzman (*danguzman_at_nospam-earthlink.net*)

Date: 07/18/04

Date: Sun, 18 Jul 2004 13:05:25 -0500

- > *What should I do given the*
- > *requirements as described above – should I create a default value of null*
- > *for my Text or NText column? Would that take care of this 'valid pointer'*
- > *issue?*

No – a default constraint of NULL will not initialize the text pointer. If you insert a row with an initial value of NULL for text data, the pointer will not be initialized (allowing WRITETEXT) until an explicit UPDATE of the text column is performed.

- > *What value should I store in the column when the user decides to*
- > *remove all text content for a given row? If I receive a zero-length string*
- > *from the application, should I insert a null into the Text column?*

It seems to me the application code should specify the desired value, NULL or otherwise. The text pointer will remain valid regardless of the value specified so there is no need to translate empty strings to NULL or visa-versa.

Note that you can make your life a lot easier by using standard INSERT and UPDATE statements rather than using WRITETEXT. This should work in your situation since you are replacing the entire text value:

```
CREATE TABLE MyTable
(
  MyId int NOT NULL
  CONSTRAINT PK_MyTable PRIMARY KEY,
  MyTextData text NULL DEFAULT NULL
)
GO
```

```
CREATE PROC InsertMyTable
  @MyId int,
  @MyTextData text
AS
SET NOCOUNT ON
```

```
INSERT INTO MyTable VALUES(@MyId, @MyTextData)
GO
```

```
CREATE PROC UpdateMyTable
    @MyId int,
    @MyTextData text
AS
SET NOCOUNT ON
UPDATE MyTable
SET MyTextData = @MyTextData
WHERE MyID = @MyId
GO
```

```
--
Hope this helps.
Dan Guzman
SQL Server MVP
"Frank Thomas" <No@Spam.org> wrote in message
news:urV6%23WDbEHA.1048@tk2msftngpl3.phx.gbl...
> I just want to avoid the possibility of creating an application that works
> most of the time, but periodically chokes because the planets aligned in a
> way that resulted in an invalid text pointer.
>
> Specifically, my objective is simply to enable users of an application to
> store large strings of text in a SS2K database. By "large" I mean more
text
> than VarChar allows (will typically range from a few hundred bytes to
1MB).
> I don't want to split the data amongst multiple VarChar columns - which
> leaves me with Text or NText (AFAIK). Per business logic, any/all updates
to
> the text will entail a wholesale replacement of any/all text that exists
> prior to the update. That is, these updates will not require updating a
> subset of the text - the whole thing gets replaced. It will also be
possible
> for users of the application to remove all text for a given row (i.e., the
> column for the row would contain a null value). In my research I found
> WRITETEXT. Per BOL, this looks like what I need to accomplish what I want.
>
> Rather than taking the development approach of "let's throw it on the wall
> and see if it sticks" - I'm trying to understand what WriteText is all
about
> so that I implement the business requirements in a way that makes sense,
> performs reliably and quickly, blah blah blah. But I'm confused by what
BOL
> describes...
>
> Can someone explain this quote from BOL/SP3?
>
> <BEGIN QUOTE>
> For WRITETEXT to work properly, the column must already contain a valid
text
> pointer. {okay, I'm with you so far...}
>
> If the table does not have in row text, SQL Server saves space by not
> initializing text columns when explicit or implicit null values are placed
> in text columns with INSERT, and no text pointer can be obtained for such
> nulls. To initialize text columns to NULL, use the UPDATE statement. If
the
```

microsoft.public.sqlserver.programming: Re: WRITETEXT – I'm utterly confused By BOL

```
> table has in row text, there is no need to initialize the text column for
> nulls and you can always get a text pointer.
> <BEGIN QUOTE>
>
> Am I supposed to initialize text columns to NULL or not? I know about null
> values - but never heard of explicit null vs implicit null... Theory
> aside,
> I have a system to build and deliver. What should I do given the
> requirements as described above - should I create a default value of null
> for my Text or NText column? Would that take care of this 'valid pointer'
> issue? What value should I store in the column when the user decides to
> remove all text content for a given row? If I receive a zero-length string
> from the application, should I insert a null into the Text column?
>
> Thanks!
>
>
```