**Paul S. Randal**

**In Recovery...**

[](http://www.sqlskills.com/blogs/paul)

* [SQLskills Home](http://www.sqlskills.com)
* [Blog Home](http://www.sqlskills.com/blogs/paul)
* [About Paul](http://www.sqlskills.com/aboutpaulsrandal.asp)
* [Training](http://www.sqlskills.com/immersionevents.asp)
* [Services](http://www.sqlskills.com/audit.asp)
* [Site-Admin](http://www.sqlskills.com/blogs/paul/wp-admin)

You are here: [Home](http://www.sqlskills.com/blogs/paul) >> [Example Scripts](http://www.sqlskills.com/blogs/paul/category/example-scripts/) >> **Inside the Storage Engine: Ghost cleanup in depth**

[**Inside the Storage Engine: Ghost cleanup in depth**](http://www.sqlskills.com/blogs/paul/inside-the-storage-engine-ghost-cleanup-in-depth/)

By: [Paul Randal](http://www.sqlskills.com/blogs/paul)

Posted on: October 8, 2007 4:24 pm

Over the years I was in the Storage Engine team I saw a lot of concern on the various forums about the ghost cleanup task. There have been a few bugs with it in previous versions  (see these KB articles – [932115](http://support.microsoft.com/kb/932115) and [815594](http://support.microsoft.com/kb/815594)) and there's very little info available on it. For some reason I didn't get around to posting about it on my old blog but today I want to go into some depth on it.

So what is ghost cleanup? It's a background process that cleans up ghost records – usually referred to as the ghost cleanup task. What's a ghost record? As I described briefly in the [Anatomy of a record post](http://www.sqlskills.com/blogs/paul/inside-the-storage-engine-anatomy-of-a-record/) last week, a ghost record is one that's just been deleted *in an index on a table* (well, actually it gets more complicated if snapshot isolation of some form is enabled but for now, a record in an index is a good start). Such a delete operation never physically removes records from pages – it only marks them as having been deleted, or ghosted. This is a performance optimization that allows delete operations to complete more quickly. It also allows delete operations to rollback more quickly because all that needs to happen is to unmark the records as being deleted/ghosted, instead of having to reinsert the deleted records. The deleted record will be physically removed (well, its slot will be removed – the record data isn't actually overwritten) later by the background ghost cleanup task. The ghost cleanup task will leave a single record on the page to avoid having to deallocate empty data or index pages.

The ghost cleanup task can't physically delete the ghost records until after the delete transaction commits because the deleted records are locked and the locks aren't released until the transaction commits. As an aside, when ghost records exist on a page, even a NOLOCK or READ UNCOMMITTED scan won't return them because they are marked as ghost records.

When a record is deleted, apart from it being marked as a ghost record, the page that the record is on is also marked as having ghost records in one of the allocation maps – the PFS page (post coming soon!) – and in its page header. Marking a page as having ghost records in a PFS page also changes the database state to indicate that there are some ghost records to cleanup – somewhere. Nothing tells the ghost cleanup task to clean the specific page that the delete happened on - yet. That only happens when the next scan operation reads the page and notices that the page has ghost records.

The ghost cleanup task doesn't just start up when it's told to – it starts up in the background every 5 seconds and looks for ghost records to cleanup. Remember that it won't be told to go cleanup a *specific* page by a delete operation – it's a subsequent scan that does it, if a scan happens. When the ghost cleanup task starts up it checks to see if its been told to cleanup a page – if so it goes and does it. If not, it picks the next database that is marked as having some ghost records and looks through the PFS allocation map pages to see if there are any ghost records to cleanup. It will check through or cleanup a limited number of pages each time it wakes up – I remember the limit is 10 pages – to ensure it doesn't swamp the system. So – the ghost records will eventually be removed – either by the ghost cleanup task processing a database for ghost records or by it specifically being told to remove them from a page. If it processes a database and doesn't find any ghost records, it marks the database as not having any ghost records so it will be skipped next time.

How can you tell its running? On SQL Server 2005, you can use the following code to see the ghost cleanup task in sys.dm\_exec\_requests:

SELECT \* INTO myexecrequests FROM sys.dm\_exec\_requests WHERE 1 = 0;

GO

SET NOCOUNT ON;

GO

DECLARE @a INT

SELECT @a = 0;

WHILE (@a < 1)

BEGIN

INSERT INTO myexecrequests SELECT \* FROM sys.dm\_exec\_requests WHERE command LIKE '%ghost%'

SELECT @a = COUNT (\*) FROM myexecrequests

END;

GO

SELECT \* FROM myexecrequests;

GO

And on SQL Server 2000 you need to use sysprocesses (well, on SQL Server 2005 this works as well but its fake view derived from the DMVs):

SELECT \* INTO mysysprocesses FROM master.dbo.sysprocesses WHERE 1 = 0;

GO

SET NOCOUNT ON;

GO

DECLARE @a INT

SELECT @a = 0;

WHILE (@a < 1)

BEGIN

INSERT INTO mysysprocesses SELECT \* FROM master.dbo.sysprocesses WHERE cmd LIKE '%ghost%'

SELECT @a = COUNT (\*) FROM mysysprocesses

END;

GO

SELECT \* FROM mysysprocesses;

GO

The output from sys.dm\_exec\_requests is (with most unused and uninteresting columns stripped off):

session\_id request\_id  start\_time              status       command  
———- ———– ———————– ———— —————-  
15         0           2007-10-05 16:34:49.653 background   GHOST CLEANUP

So how can you tell if a record is ghosted? Let's engineer some and look at it with DBCC PAGE – I've stripped out the uninteresting bits of the output and highlighted the interesting ghost parts:

CREATE TABLE t1 (c1 CHAR(10))

CREATE CLUSTERED INDEX t1c1 on t1 (c1)

GO

BEGIN TRAN

INSERT INTO t1 VALUES ('PAUL')

INSERT INTO t1 VALUES ('KIMBERLY')

DELETE FROM t1 WHERE c1='KIMBERLY';

GO

DBCC IND ('ghostrecordtest', 't1', 1);

GO

DBCC TRACEON (3604);

GO

DBCC PAGE ('ghostrecordtest', 1, 143, 3);

GO

<snip>

m\_freeData = 130    m\_reservedCnt = 0   m\_lsn = (20:88:20)  
m\_xactReserved = 0  m\_xdesId = (0:518)  **m\_ghostRecCnt = 1**  
m\_tornBits = 0

<snip>

Slot 0 Offset 0×71 Length 17

Record Type = **GHOST\_DATA\_RECORD**      Record Attributes =  NULL\_BITMAP  
Memory Dump @0x6256C071

00000000:   1c000e00 4b494d42 45524c59 20200200 †….KIMBERLY  ..  
00000010:   fc†††††††††††††††††††††††††††††††††††.  
UNIQUIFIER = [NULL]

Slot 0 Column 1 Offset 0×4 Length 10

c1 = KIMBERLY

Slot 1 Offset 0×60 Length 17

Record Type = PRIMARY\_RECORD         Record Attributes =  NULL\_BITMAP  
Memory Dump @0x6256C060

00000000:   10000e00 5041554c 20202020 20200200 †….PAUL      ..  
00000010:   fc†††††††††††††††††††††††††††††††††††.  
UNIQUIFIER = [NULL]

Slot 1 Column 1 Offset 0×4 Length 10

c1 = PAUL

Let's see what goes on the transaction log during this process (remember this is undocumented and unsupported – do it on a test database) – I've stripped off a bunch of the columns in the output:

DECLARE @a CHAR (20)

SELECT @a = [Transaction ID] FROM fn\_dblog (null, null) WHERE [Transaction Name]='PaulsTran'

SELECT \* FROM fn\_dblog (null, null) WHERE [Transaction ID] = @a;

GO

Current LSN              Operation         Context             Transaction ID  
———————— —————– ——————- ————–  
00000014:00000054:0011   LOP\_BEGIN\_XACT    LCX\_NULL            0000:00000206  
00000014:0000005a:0012   LOP\_INSERT\_ROWS   LCX\_CLUSTERED       0000:00000206  
00000014:0000005a:0013   LOP\_INSERT\_ROWS   LCX\_CLUSTERED       0000:00000206  
00000014:0000005a:0014   LOP\_DELETE\_ROWS   LCX\_MARK\_AS\_GHOST   0000:00000206  
00000014:0000005a:0016   LOP\_DELETE\_ROWS   LCX\_MARK\_AS\_GHOST   0000:00000206

So there are the two inserts followed by the two deletes – with the rows being marked as ghost records. But where's the update to the PFS page? Well, changing the ghost bit in a PFS page is not done as part of a transaction. We'll need to look for it another way (apart from just dumping everything in the transaction log and searching manually):

SELECT Description, \* FROM fn\_dblog (null, null) WHERE Context like '%PFS%' AND AllocUnitName like '%t1%';

GO

Description               Current LSN              Operation        Context   Transaction ID  
————————- ———————— —————- ——— —————-  
Allocated 0001:0000008f   00000014:00000054:0014   LOP\_MODIFY\_ROW   LCX\_PFS   0000:00000208  
                          00000014:0000005a:0015   LOP\_SET\_BITS     LCX\_PFS   0000:00000000

The first one is just allocating a page but the second one is the one we're looking for – it's changed the bit for the page to say it has ghost records on. Let's commit the transaction and see what happens, filtering out all the previous transaction log:

SELECT MAX ([Current LSN]) FROM fn\_dblog (null, null);

GO

– 00000014:0000005e:0001

COMMIT TRAN

GO

SELECT [Page ID], \* FROM fn\_dblog (null, null) WHERE [Current LSN] > '00000014:0000005e:0001';

GO

Page ID         Current LSN              Operation          Context         Transaction ID  
————— ———————— —————— ————— ————–  
NULL            00000014:0000005f:0001   LOP\_COMMIT\_XACT    LCX\_NULL        0000:00000206  
0001:0000008f   00000014:00000060:0001   LOP\_EXPUNGE\_ROWS   LCX\_CLUSTERED   0000:00000000

We see that almost as soon as the transaction has commited, the ghost cleanup task goes in and process the page. Let's check a page dump to make sure the record is gone, and show that the contents of the record are still on the page (again, with non-relevant bits snipped out):

DBCC PAGE ('ghostrecordtest', 1, 143, 3);

GO

<snip>

m\_freeData = 130         m\_reservedCnt = 0        m\_lsn = (20:94:1)  
m\_xactReserved = 0       m\_xdesId = (0:518)       **m\_ghostRecCnt = 0**  
m\_tornBits = 0

<snip>

Record Type = PRIMARY\_RECORD         Record Attributes =  NULL\_BITMAP  
Memory Dump @0x6212C060

00000000:   10000e00 5041554c 20202020 20200200 †….PAUL      ..  
00000010:   fc†††††††††††††††††††††††††††††††††††.  
UNIQUIFIER = [NULL]

Slot 0 Column 1 Offset 0×4 Length 10

c1 = PAUL

DBCC PAGE ('ghostrecordtest', 1, 143, 2);

GO

<snip>

6212C040:   01000000 00000000 00000000 00000000 †…………….  
6212C050:   00000000 00000000 00000000 00000000 †…………….  
6212C060:   10000e00 5041554c 20202020 20200200 †….PAUL      ..  
6212C070:   fc1c000e 004b494d 4245524c 59202002 †…..**KIMBERLY**  .  
6212C080:   00fc0000 00000000 00000000 01000000 †…………….  
6212C090:   00000000 13000000 01000000 00000000 †…………….

<snip>

So even though the record no longer exists, all that happened was that the slot was removed from the slot array at the end of the page – the record contents will remain on the page until the space is reused.

In the next post I'll go into details of the PFS and other allocation maps. Btw – please let me know if this stuff is interesting – I'd like to know where to spend blogging time. Thanks!

**Related Posts**

* [Lock logging and fast recovery](http://www.sqlskills.com/blogs/paul/lock-logging-and-fast-recovery/)
* [New script: When were the sp\_configure options last changed?](http://www.sqlskills.com/blogs/paul/new-script-when-were-the-sp_configure-options-last-changed/)
* [Ghost cleanup redux](http://www.sqlskills.com/blogs/paul/ghost-cleanup-redux/)
* [Turning off the ghost cleanup task for a performance gain](http://www.sqlskills.com/blogs/paul/turning-off-the-ghost-cleanup-task-for-a-performance-gain/)
* [Initial VLF sequence numbers and default log file size](http://www.sqlskills.com/blogs/paul/initial-vlf-sequence-numbers-and-default-log-file-size/)

Posted in: [Example Scripts](http://www.sqlskills.com/blogs/paul/category/example-scripts/), [Inside the Storage Engine](http://www.sqlskills.com/blogs/paul/category/inside-the-storage-engine/), [On-Disk Structures](http://www.sqlskills.com/blogs/paul/category/on-disk-structures/), [Transaction Log](http://www.sqlskills.com/blogs/paul/category/transaction-log/), [Undocumented commands](http://www.sqlskills.com/blogs/paul/category/undocumented-commands/)

[9 Comments](http://www.sqlskills.com/blogs/paul/inside-the-storage-engine-ghost-cleanup-in-depth/#comments)

### 9 Responses to Inside the Storage Engine: Ghost cleanup in depth

1. http://0.gravatar.com/avatar/ad516503a11cd5ca435acc9bb6523536?s=44GarbageCollector says:

[January 4, 2008 at 2:33 pm](http://www.sqlskills.com/blogs/paul/inside-the-storage-engine-ghost-cleanup-in-depth/#comment-3663)

Yes, this stuff is interesting. Please keep posting!!

[Reply](http://www.sqlskills.com/blogs/paul/inside-the-storage-engine-ghost-cleanup-in-depth/?replytocom=3663#respond)

1. http://0.gravatar.com/avatar/ad516503a11cd5ca435acc9bb6523536?s=44Praveen Voleti says:

[October 7, 2008 at 6:18 pm](http://www.sqlskills.com/blogs/paul/inside-the-storage-engine-ghost-cleanup-in-depth/#comment-3664)

Absolutely Awesome explanation. Thank you very much.

[Reply](http://www.sqlskills.com/blogs/paul/inside-the-storage-engine-ghost-cleanup-in-depth/?replytocom=3664#respond)

1. http://0.gravatar.com/avatar/ad516503a11cd5ca435acc9bb6523536?s=44Vaibhav Bapat says:

[January 28, 2009 at 5:14 am](http://www.sqlskills.com/blogs/paul/inside-the-storage-engine-ghost-cleanup-in-depth/#comment-3665)

Hi all,  
The above article extremely excellent, it clear all doubt in ghost cleanup process without doing any question. This article focus on different angle to solve and understand different question raise against it

Thank a lot  
Vaibhav Bapat

[Reply](http://www.sqlskills.com/blogs/paul/inside-the-storage-engine-ghost-cleanup-in-depth/?replytocom=3665#respond)

1. http://0.gravatar.com/avatar/ad516503a11cd5ca435acc9bb6523536?s=44ladnar luap says:

[March 25, 2009 at 4:00 pm](http://www.sqlskills.com/blogs/paul/inside-the-storage-engine-ghost-cleanup-in-depth/#comment-3666)

"The ghost cleanup task will leave a single record on the page to avoid having to deallocate empty data or index pages." – this is not true in SQL Server 2005 or later version. Unless this is the first page maintained in system table, a page could be de-allocated by ghost cleanup.

[Reply](http://www.sqlskills.com/blogs/paul/inside-the-storage-engine-ghost-cleanup-in-depth/?replytocom=3666#respond)

1. http://0.gravatar.com/avatar/ad516503a11cd5ca435acc9bb6523536?s=44[*paul*](http://www.sqlskills.com/) says:

[March 25, 2009 at 4:07 pm](http://www.sqlskills.com/blogs/paul/inside-the-storage-engine-ghost-cleanup-in-depth/#comment-3667)

Yes, I think you’re right. That was certainly the case in 2000. But why post the comment with the obvious fake name?

[Reply](http://www.sqlskills.com/blogs/paul/inside-the-storage-engine-ghost-cleanup-in-depth/?replytocom=3667#respond)

1. http://0.gravatar.com/avatar/ad516503a11cd5ca435acc9bb6523536?s=44Amy says:

[October 12, 2009 at 3:05 pm](http://www.sqlskills.com/blogs/paul/inside-the-storage-engine-ghost-cleanup-in-depth/#comment-3668)

Very well articulated!!! (I know you must get that a lot )  
But thoroughly enjoyed reading this

[Reply](http://www.sqlskills.com/blogs/paul/inside-the-storage-engine-ghost-cleanup-in-depth/?replytocom=3668#respond)

1. [*Turning off the ghost cleanup task for a performance gain | Paul S. Randal*](http://www.sqlskills.com/blogs/paul/turning-off-the-ghost-cleanup-task-for-a-performance-gain/) says:

[January 1, 2013 at 9:17 pm](http://www.sqlskills.com/blogs/paul/inside-the-storage-engine-ghost-cleanup-in-depth/#comment-3863)

[...] Inside the Storage Engine: Ghost cleanup in depth [...]

[Reply](http://www.sqlskills.com/blogs/paul/inside-the-storage-engine-ghost-cleanup-in-depth/?replytocom=3863#respond)

1. [*An XEvent a Day (15 of 31) – Tracking Ghost Cleanup | Jonathan Kehayias*](http://www.sqlskills.com/blogs/jonathan/an-xevent-a-day-15-of-31-tracking-ghost-cleanup/) says:

[January 2, 2013 at 10:58 pm](http://www.sqlskills.com/blogs/paul/inside-the-storage-engine-ghost-cleanup-in-depth/#comment-4111)

[...] know anything about Ghost Cleanup, I recommend highly that you go read Paul Randal’s blog posts Inside the Storage Engine: Ghost cleanup in depth, Ghost cleanup redux, and Turning off the ghost cleanup task for a performance gain.  To my [...]

[Reply](http://www.sqlskills.com/blogs/paul/inside-the-storage-engine-ghost-cleanup-in-depth/?replytocom=4111#respond)

1. [*Lock logging and fast recovery | Paul S. Randal*](http://www.sqlskills.com/blogs/paul/lock-logging-and-fast-recovery/) says:

[January 3, 2013 at 7:26 am](http://www.sqlskills.com/blogs/paul/inside-the-storage-engine-ghost-cleanup-in-depth/#comment-4183)

[...] Inside the Storage Engine: Ghost cleanup in depth [...]

[Reply](http://www.sqlskills.com/blogs/paul/inside-the-storage-engine-ghost-cleanup-in-depth/?replytocom=4183#respond)

### Leave a Reply