

Re: Lengthy merge code

Source:

<http://www.tech-archive.net/Archive/Word/microsoft.public.word.mailmerge.fields/2007-01/msg00309.html>

- *From:* "Peter Jamieson" <pjj@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Wed, 24 Jan 2007 22:59:27 -0000
-

Do you think bumping the virtual memory might help? We're running the merges on a dedicated machine that's faster and better than our other desktop jobbies. All it does is the merges. It lives for merging. ;)

Shouldn't think so, but in this kind of situation I would try anything I was convinced I could back out of. However, since you've dedicated a system to it

- a. it's probably going to be worth taking a fairly systematic approach to performance monitoring, using the Windows performance monitoring facilities, Task manager, and so on to try to discover what is going on.
- b. if there's plenty of free (real) memory even before Word starts its merge, the chances are that the problem has nothing to do with memory.
- c. if not, I'd probably try to remove/disable anything I didn't need. Not always easy to discover, of course.

Yes – the merge code seems rather insane. We've converted to a proprietary software package – I don't dare name it – and the merge seems to be the best way to handle the types of communications we send out.

From what you said earlier it sounds as if you're now committed to this approach, but are there other options? In particular, is there any way to do more of the decision-making work when exporting from your package (if that's what you're doing) and less using field coding?

Peter Jamieson

"Heather" <heather.c@xxxxxxxxxx> wrote in message
<news:1169672285.780756.99660@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>

Yes – the merge code seems rather insane. We've converted to a proprietary software package – I don't dare name it – and the merge seems to be the best way to handle the types of communications we send out. I haven't come across much information on merge code of this

Re: Lengthy merge code

magnitude.

Do you think bumping the virtual memory might help? We're running the merges on a dedicated machine that's faster and better than our other desktop jobbies. All it does is the merges. It lives for merging. ;)

Heather

On Jan 24, 12:16 pm, "Peter Jamieson" <p...@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote:

the programming for this particular letter is about 90 pages
of merge
code,Rather you than me :-) I've come across some complex
merge
documents over

the years but I think yours beats the lot !

Reports suggest that there are "boundaries" at which things start to slow significantly, but I've never really had a good go at pinning down any specific causes. One possibility in many cases is that even with quite large amounts of RAM in a system, software (i.e. code/programs) can still take so much of it that an application may run into a RAM limitation quite quickly when it is loading or generating data, at which point Windows will spend more time playing with its virtual memory. You /might/ be able to check that by removing as many programs and inessential services from your system and having a look at real/virtual memory usage. The trouble is that performance testing of this kind can be very time-consuming, and if the real scenario involved real printing, your tests aren't necessarily realistic unless you're actually using up paper or using something like the full Acrobat or Microsoft Office Document Imaging as your output printer.

Peter Jamieson

"Heather" <heathe...@xxxxxxxx> wrote in
message<news:1169664132.110199.76040@xx>

I'm with you – it ain't broke (yet) and I'd rather not have to
fix it.
:)

Re: Lengthy merge code

I'm not sure if the SKIPIF would be a good solution unless I create multiple templates for different kinds of students and run the merges using the same data. I have to run my merges in groups of 500 or less, because of our internal document imaging software limitations, but our daily files are about 50–100 records, with a few huge files in the summer (3000+).

With regards to Word getting progressively slower – is there an upper limit, do you think? 30+ records? I have a LOT of includetext fields – the programming for this particular letter is about 90 pages of merge code, broken down into five documents.

Heather

On Jan 24, 10:52 am, "Peter Jamieson"
<p...@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
wrote:

The only type of code that does anything like that is a { SKIPIF } code and as far as I can tell that's not appropriate for your situation – i.e. it's there to stop processing of the current record altogether when the condition specified in the SKIP is found.

Not much helps when you're dealing with large numbers – you can't convert

Re: Lengthy merge code

your data to Access, for example, because it has a 255-column maximum, and that's why you probably wouldn't be able to use SQL even if you wanted to.

The merge takes a long time, and I assume it's because word is ticking through each line of code and saying to itself 'yes/no.' It could well be that. It could also be because Word takes quite a long time

even to read a long text record and split it up into separate fields.

(It shouldn't but as far as I can tell, it does).

Doing lots of INCLUDETEXTS

is

likely to have a significant impact as well.

There are some unauual text/formatting features that can slow processing down (e.g.

"formatted

bullets"), but I think you would be able to tell if you have any of

those

by

taking out most of your fields and seeing if the merge is still slow.

You

may also find that Word gets progressively slower as it works its way through your data file, and if that is the case, it /might/ make sense

to

do

the merge in multiple chunks.

Re: Lengthy merge code

so I'm stuck with my pages
and pages of
merge
code.

To try to improve
performance, I have code
stored in different
files...so with the main
letter, if the individual
record meets
certain
criteria, there's an
includetext function for a
document. That
document, in turn, has
includetext functions from a
master document
that inserts appropriate
paragraphs. It works – it's
not pretty, but
it
works.

The merge takes a long
time, and I assume it's
because word is
ticking
through each line of code
and saying to itself 'yes/no.'

My question is: is there
something I can insert into
my code to say
that if the condition is found
(positive result), to stop
searching
though the rest of the code
in that linked document and
move on?

Thank you again for your
help. This group has been an

Re: Lengthy merge code

invaluable
resource.

Heather