What Is the Difference Between a MESSAGE and a PAGE?

Q: If I send a message to someone, they get it. Why is there a "message" count and a "page" count that are often different?

A square is a rectangle but a rectangle isn't (necessarily) a square; a page is a message, but a message isn't a page. A message is the letter you write to someone, but it isn't mail until the post office delivers it (or can't deliver it, or loses it). A message is what's created when you enter the text in Smart Console or eNotify or SmartWeb or whatever. But it isn't a page until the paging process handles it (and succeeds or fails).

Therefore, a page is a message that was delivered (or attempted to be delivered) to a pager (in Smart Suite, a device set up as a pager in the pager form, which can also be an email address). You can't have a page without a message, but a message doesn't have to go to a pager (a device set up as a pager in the Smart Suite system).

When a message is created (in Smart Console, from a phone call to the VRU, via SmartWeb, etc) it is first stored in the MESSAGE table.

It is important to understand that a message is not SENT. One could argue that the programs "send" the message to the database to be stored, but a message is not SENT to an end-user / device. A message is STORED.

If the "Send to Page" flag is set to true for the given message, a row is created in the QUEUED_PAGE table for that message (see also Smart Suite Messaging Priorities article). If the target recipient has more than one device, then more than one row (one per device) is created in the QUEUED_PAGE table.
If the "Send to Email" flag is set to true for the given message, a row is created in the QUEUED_EMAIL table for that message.

Also important (or unfortunate depending on your point of view) is the dual behavior of the word "archive" as used in Smart Suite. The message table has a column for its time of entry. When a message is stored, that date becomes its DATE_ENTERED. Assuming that the ARCHIVE_FLAG for the message is set to T, once it has successfully been paged its DATE_ARCHIVED column is populated. This also occurs when a user manually "archives" it (via Smart Console, for example). A message with no archived date is an "Active Message".

The word "archive" is also used to mean that a row from any defined table is moved from the live user (atms) tables to the archive user (atms_archive) tables. When dealing with paging, archiving a message does not mean it's moved to the atms_archive schema (see Data Retention article), it means its DATE_ARCHIVED column was populated.

The paging processes that are logged into the database take the information from the row in the QUEUED_PAGE table and deal with it according to their own specifications.

If the page is successful, then the PAGE is SENT and the information is posted to the PAGE_LOG table. If the message's archive flag was set to T, then the system will post the date of the success to the message's date archive column. If the page flag was F and the email flag was T, then the row is posted to the EMAIL_LOG table instead. The MESSAGE_EVENT table is updated along the way from storage to processing.

If the page was not successful, then the archive doesn't happen; the archived date does not get posted and the message remains in the MESSAGE table. The PAGE_LOG table is still updated but the ERROR_NUMBER based on the paging error is posted to the table.

If the page flag was false, the message will never have been posted to the QUEUED_PAGE table and therefore the paging process won't archive it.

If an operator cancels a page before it goes out, or it has to be removed from the QUEUED_PAGE table before it can be paged, then the same situation occurs.

There is a cron (a scheduled task) on the database servers that runs daily and posts an archival date to all messages with NO archival date that were entered three days prior to the cron's running. This way the archive process (for tables) can move those rows to the archive (atms_archive) storage. If the message had no archive date then it would stay in the live tables.

At most sites there will always be more rows in the PAGE_LOG table than the MESSAGE table. Sites that use paging very little but use the console suite as more of an answering service would see that flipped.

Messages vs Pages
https://knowledge.spok.com/Spok_Products/Smart_Suite/Smart_Suite_Knowledge_Articles/Messages_vs_Pages
© 2019 Spok
Here is a process flow chart of the message to page process.

Here are some examples of messages and pages, adding up the counts as we go. In these examples we’ll assume that the paging was successful unless otherwise noted.

Start with five message groups with 20 listing records each. One message per group is stored, becoming 20 pages per group. Each member of the group gets a message stored in the MESSAGE table, but the message sequence number is the same for 20 different listing ids.

Although there are 20 rows, only five distinct message sequence numbers have been created.

Total:
- 100 messages stored
- 100 pages sent (100 rows in the page log table)

Next add in records with multiple devices.
For each message stored, there will be one entry in the message table and two, three, etc in page_log.

We post 10 messages to 10 people with multiple devices (one message for each of 10, that is), and let’s say they all have only two devices. That’s 10 more messages and 20 more pages.

Total:
- 110 messages
- 120 pages

Now add in rows where the messages’ PAGE_FLAGs are false. Perhaps the person to be messaged has an unpageable status -- in surgery, generally unavailable, etc. or the site in question is also running an answering service (where the messages are stored and then someone from the location calls in for their messages later, or they're faxed / emailed at the end of the day).
For this example, 40 messages with page flag F.

Total:
- 150 messages
- 120 pages

Messages vs Pages
https://knowledge.spok.com/Spok_Products/Smart_Suite/Smart_Suite_Knowledge_Articles/Messages_vs_Pages

© 2019 Spok
As stated above, if the PAGE_FLAG is T but the page errored (could not be sent), then the message won't have an archive date posted.

In this next example, the row for the message in the QUEUED_PAGE table was picked up by the paging process but an error occurred (bad modem, bad network connection, invalid pager id on the provider side, etc etc). This message is stored but the page failed, so the message is not archived by the paging process.

We'll say there were 20 messages to 20 listing records (one message to each of 20 records) with page flag T, one device each (not multiple devices), but 12 failures.

Total:
170 messages
140 pages

BUT if we're only concerned with successful pages -- those with error code 0 -- we have

Total:
170 messages
128 pages

When the PAGE_FLAG is T but the message / transaction never made it to the paging process (stuck in queue), the paging process does not archive the message. If the paging process is turned off or if the class of service for a pager is set up to go to a process that doesn't exist, the message will remain in queue and never be posted to the PAGE_LOG table.

Another 20 messages intended for 20 records, one device each.

Total:
190 messages
140 pages (including errors)
128 pages (no errors)

Now add in five new message groups with 20 records each, each record sending to multiple devices, each with 3 devices. One message for each group becomes 20 messages per group. At five groups, this becomes 100 messages (with five distinct message sequence numbers).

There were 190 messages, and now there are 290 messages.

The additional 100 messages go to page three devices each, which becomes 300 pages.
There were 140 or 128 pages, which now becomes 440 pages (including errors) or 428 (no errors).

Total:
290 messages
440 pages (including errors)
428 pages (no errors)

In addition to the QUEUED_PAGE and PAGE_LOG tables there are the QUEUED_EMAIL and EMAIL_LOG tables. The path is the same, although the process that sends the email uses the QUEUED_EMAIL.

Take 5 listing records that have a LISTING_EMAIL_ADDRESS set up, but no pager. One message is emailed to each of these.

Total:
295 messages
440 pages (including errors)
428 pages (no errors).

The other 5 rows are in the EMAIL_LOG table.

Here is a real world example (one month) from a site.

Count of rows from the MESSAGE table where the page flag is T from Dec 2018:
121658

Count of distinct message sequence numbers for the same:
99951

How many of those rows made it to the PAGE_LOG table?

Count of rows from PAGE_LOG for the given messages (above) for Dec 2018 (including errors):
152671

How many distinct message sequence numbers (including errors):
99951

But what about just the successful ones?
Count of rows from Dec 2018 for successful pages:
149734

Distinct message sequence numbers from Dec 2018 for successful pages:
99947

----------------------------------------
These small amounts add up quickly. For the same site, over the course of 7 months (from Jun 2018 to (through) Dec 2018):

Rows in the message table with page_flag T:
867858

Distinct message sequence numbers with page_flag T:
717995

(Since this article is only about messages and pages, these examples are only looking at those messages with page flag T. But here are the numbers for this site just for transparency)

Rows in the message table total (page flag T and F):
869363, for a difference of 1505. That's 1505 messages in the seven month span with a page_flag of F, which is about 7 a day.

Distinct message sequence numbers total in the MESSAGE table:
719500

which is again the difference of 1505 rows.

Rows in the page log table (including errors):
1045353

Rows in the page log table (successful only):
1032233

Distinct message sequence numbers in the page log table:
717993

Distinct message sequence numbers in the page log table (successful only):
717981

From the difference of two rows from the total number of messages that were supposed to be paged (717995) to those that made it to the page log (717993) we can infer that those rows never made it
to the page log table in the first place (by way of an operator cancelling the page, the row being removed from the queue, or whatever else it could have been). Digging a little deeper for this particular site, one of those was emailed only (and went to the EMAIL_LOG table) and one was queued up and didn't get paged (and then stalemsg archived it three days later).

KB52038