



USERS CONFERENCE 2017



Archiving

Presented By: Phill Addoriso





Archiving

- Archiving – a new twist?
- built to help with e-commerce
- built with DB2



In this session we're going to take a fresh look at Archiving – how we are enhancing it to take something old and making it new. We'll tell you about what we're doing and you can be the judge as to just how new this is.

As I mentioned earlier today, Archiving is something that last year had come up in internal discussions. We came up with the idea, we developed the concept to a point, but it was not really on our roadmap at the time. At least not enough for us to mention it - because we didn't. However more recently the importance of the idea began to drive it up in priority on our long to do list. What started to make this more important? A lot of this was due to e-commerce.

One other note before we jump in, the features being introduced are another example of a capability being built on DB2. These capabilities will be available to Maves systems using DB2 or those using PxPlus with DB2 mirroring.

Archiving

- what is the new archiving feature?
- what is the rationale behind the new Archiving functionality?
- what are the problems archiving is being used to solve?

So what is the new archiving feature?

Where I think I would like to start is with an explanation of the rationale behind Archiving and why we are doing this. And perhaps the best way to explain the rationale is to provide some background concerning the problems that we saw and how the archiving capability is being used to solve these problems. Also I'd like to add that I believe the archiving features could lead to additional benefits beyond the initial issues we needed to address now.

So let me begin by describing the problems. And I think perhaps you may not only recognize and agree with these issues, but possibly have your own related issues that you could add.

Archiving

- range of order numbers (and other transaction numbers) in the system are finite
- old orders are purged (IP94)
- the order numbers become available again and can be re-cycled
- problem solved?

As you all know, the order numbers in the Maves system get consumed and eventually run out. The range of order numbers in the system are finite. This is true of order numbers, receipt numbers, invoice numbers, and other numbers - not just order numbers.

The Maves system deals with this by taking old orders from the system and purging these using IP94, something I'm sure most of you are familiar with or at least aware of.

Once the old orders have been purged the order numbers become available again and can be re-cycled. The Maves system is green. This is a big deal in today's world of trying to be eco-friendly – and we're just trying to do our part. Again, this occurs with other transaction numbers in the Maves system, not just order numbers.

So using this approach, Maves largely avoids having transaction numbers run out. OK, so if we have a solution to this then what's the problem?

Archiving

- the problem is not solved because ...
- your clients are asking you to keep their data available longer and they want access to it
- clients are processing greater volumes of orders – especially clients doing e-commerce
- they have a large volume of small orders and consume many more order numbers as a result
- the problem is even greater when longer retention and e-commerce orders are combined

I'm not sure if any of the following applies to any of you in this room, but I have seen some of these issues with some of our clients. So here are some of the issues we've seen:

- 1) Some of your clients are asking that you keep their data available longer. They want access to data such as order details or receipt information for longer periods of time than is currently possible. What is currently possible, of course, varies depending on the number of transactions you do. But it is more difficult to recycle the order numbers before they run out when the client is asking to retain data longer. And again, this could apply to all transaction numbers, but we see the problem more so with orders.
- 2) In addition to your clients, some of you are looking to keep data longer for a variety of reasons that include analysis, reference, or other needs. ...
- 3) Clients are processing greater volumes of orders. This is true in general, almost across the board, but it is especially true for those clients involved with ecommerce. These clients typically have a much larger volume of small orders as opposed to a smaller volume of larger orders. This is important because it is not the volume of goods you are moving that affects your need for more order numbers, but rather the number of discrete individual orders. And in ecommerce scenarios that number tends to rise quite rapidly. This issue in particular was a concern to us as we are working to make the system better support the needs of the 3PL providing e-commerce fulfilment.



Archiving

- so what is the solution?
- order archiving!
- based around an archive database for orders
- the active or operational database can purge orders much more quickly
- orders are still available after they are removed from the active database
- active or operational database can purge orders much more quickly

4) The e-commerce scenario alone is problem enough, when both issues are combined together it really exacerbate the problem. Consuming a larger volume of orders, coupled with the need to keep the order history around longer really increases the issues tied to limited transaction numbers — in particular order numbers.

So what's the solution?

Here is how we are approaching this through the use of archiving. Note that in the first release the archiving will address orders only. We've put our focus here because, as I mentioned, this is where the problem seems to hit the most. We will be addressing other transactions in a subsequent release.

I will focus my discussion on how the order archiving will work. When it comes to other transactions there are, of course, some differences, but the concepts are the same. The feature is based around an archive database for orders. That means the active or operational database can purge orders much more quickly because they are not lost once they are removed from the active database. They will continue to be available in the archive database.



Archiving

- once the order is confirmed as shipped and the EDI has been sent the order is no longer needed from an operational standpoint
- we provide a grace period and then remove the orders from the active database
- order numbers are freed up much more quickly

Once an order is confirmed as shipped and the associated EDI, if any, has been sent the order is no longer needed from an operational standpoint. It is only needed for reference purposes.

Our intent is that we will provide a grace period of say 90 days after shipment (can be user configured). That should provide more than sufficient time in the event any corrective action is needed after the order is confirmed. This recognizes that some activities, in particular correcting any issue that might not be recognized until a few days later, is easier done while the order is still in the active database.

Once that grace period has passed, the orders can be removed from the active database and stored in the archive database. That allows order numbers to be freed up much more quickly, avoiding the risk of running out before they are purged.

So that's an oversimplification. There are other important details. Let me touch on the main ones.



Archiving

- designed to store historical order information indefinitely
- you will want to consider storage usage when deciding how long to keep archived data
- the same order number will be archived more than once in the archive database
- extended the order number series within the archive database – using a prefix

The archive database is designed to store historical order information indefinitely.

That doesn't mean you will. It only means that theoretically you could. You will need to balance this with practical considerations — for example how much storage your willing to provide. Clearly storing more data for a longer period of time will require additional storage. And while storage is relatively cheap, it is still a factor that you will want to consider when deciding how long to keep archived data (ie this is user configurable).

Given the information may be stored indefinitely, it means that as order numbers are recycled in the operational database and archived, the same order number will be archived more than once in the archive database.

Given probabilities, it is quite likely that the same order number, if it appears in the archive database 3 times, will be for 3 different clients. But it is not guaranteed. The same order number can be reused for the same client. As a result we need to handle the same order number being archived multiple times, including the possibility of multiple times for the same client.

We've handled this by extending the order number series within the archive database. At the time the orders are archived, a prefix is added to the order number to ensure that all orders added to the archive at a given time will have unique numbers that are different from those previously archived. Aside from this, all other order information is kept as is making it easy to refer to.



Archiving

- how will you and your clients access the archived orders?
- using Highviews and the MyLogistics portal archived information is easily accessible for both internal and external users
- Order Query Action Grid displays all orders in the active database and the archive database
- this was key for usability

That covers how we handle the archiving and deal with the transaction number issues.

Now that the order are in the archive database, how will you be able to access all of these archived orders, and how will your clients be able to access this information?

Archived data will be accessible for internal users via the Highviews, and for your clients, they will have access via the MyLogisitcs portal. This makes the archived information easily accessible for both internal and external users.

Currently there are two Action Grids you can use to view the orders: Orders WIP, and Order Query. These will continue to be used and the Order WIP Action Grid is completely unaffected since it only deals with Orders currently in process.

The Order Query Action Grid will display all orders whether in the active database or the archive database.



Archiving

- how will you and your clients access the archived orders?
- using Highviews and the MyLogistics portal archived information is easily accessible for both internal and external users
- Order Query Action Grid displays all orders in the active database and the archive database
- this was key for usability

Notes page 2

We believed this was very important from a usability perspective. We did not want to introduce a new Action Grid for archived orders as this would require that the user knows when an order has crossed the threshold and been archived, so they can choose the correct Action Grid to find the data they are looking for. We felt this would make the use quite clumsy for internal users and very confusing for external users.

And if you wanted a list of orders that spanned over the threshold - so that some orders on the list were archived and some were not, then you would not be able to get all the data you are interested in on a single screen.

So the Query action grid will be capable of finding the data whether archived or not and display all the orders that match your query in a single Action Grid. The only difference you will see on archived orders vs. those that are still in the active database is that the order numbers for archived orders will be longer due to the prefix assigned when the order is archived.



Archiving

Q & A

