

Key Features

- Archive in Stages
 - Using Data Archiving for Adabas with Adabas Vista Partitioning
 - Using Data Archiving for Adabas with Adabas Vista Translation
 - Search the Archive Content
 - Recall from the Archive
 - Archiving is a Long-Term Activity
 - Alert for Contamination
 - Automated Discard
 - Restart/Recovery
 - Copy (only) Feature
 - Delete (only) Feature
-

Archive in Stages

You do not need to send archive data immediately to some far off, unreachable storage facility. In fact, you can archive from one Adabas file to another Adabas file. You do not have to go directly to an off-host storage facility. You can archive out to Adabas on another (less expensive) architecture for example. This means the data is in the same form and can be reached by your existing applications.

If you archive to another Adabas file you might use that file as an intermediate storage facility (for the most recently archived content) and then still archive from that intermediate file to a flat-file store at a later date. According to the data life cycle suggested earlier, you can have your present-day data in primary systems, your recent data in secondary (Adabas) systems and your historical data in a real flat-file store.

Obviously you do not need to use the intermediate Adabas file approach, you can go direct to a flat store. Conversely, you may only wish to archive to an Adabas store. It is your choice, on a file by file basis.

Using Data Archiving for Adabas with Adabas Vista Partitioning

It has long been known that Adabas Vista partitioning can be used to segregate data by criteria that is relevant to your business. In many cases this is in some way related to the data life cycle. For example, partitions may be organized by date. This is clearly a situation where the older partitions can be a) created by Data Archiving for Adabas as the age increases and/or b) by actually archiving away the very oldest data out of the Vista partitions altogether. At a later point it may be that older data is recalled into a temporary partition to allow old data to be included into your production systems as you need. This means you can use your existing systems to process data recalled from your archive at any time.

Using Data Archiving for Adabas with Adabas Vista Translation

You might also recall archived data into a standard Adabas file (not partitioned) but use your normal applications but with a Vista translation rule to redirect activity to the recalled file rather than to the actual production data. This is another way you can use your existing systems to process data recalled from your archive at any time.

Search the Archive Content

Once data is archived to a flat store you do not have to worry. It remains within easy reach. You can search it without having to recall it to Adabas. Obviously, you can perform full Adabas searches if the archive is an Adabas file. You can search the flat store too. Clearly, searching a flat store is not as flexible or fast as when searching with Adabas. However, the flat-file search is only needed to get you to the point where you identify the data you need. Then you recall this data into its original (Adabas) form, where once again you have full Adabas search capability.

Recall from the Archive

You can recall whole archive runs that were performed previously or you can recall the results of searches performed on the archive, it is up to you. You can recall into an empty Adabas file or into an existing Adabas file, provided the original FDT is supported by the destination Adabas file.

Archiving is a Long-Term Activity

Let's say we archive 1 million records per month to a flat-file store. After one year we have 12 million records. After ten years we have 120 million records. This example spans several years where there are (12*10=120) different operations to put information into the archive. In ten years, how many times does the data structure change? "Many times" is the likely answer.

Data Archiving for Adabas takes the meta-data (FDT) along with the data and manages change to data structures. If you wish to recall data to a new FDT that does not support all the fields that were included in the original archived data you can choose to ignore the extra data, for example. Recognition and tolerance of these changes over long periods of time is very important in the tooling you use.

Alert for Contamination

Once archived, the vast majority of archive content is never used again. However, that does not mean the data is unimportant. On the contrary, over time there are masses of content in the archive. At any time in the life cycle any part of the archive may be needed. But, if a particular piece of information is needed from (let's say) 7 years ago, how do you know it is still there? How do you know it has not been lost? Or damaged?

Data Archiving for Adabas will automatically validate the archive content at a rate you set (per day) and at a level that you decide, to make sure the whole content is both reachable and valid. If there is any problem you are alerted at the earliest possible opportunity so that a remedy can be found quickly.

Automated Discard

The end of the life cycle is automated too. You decide if expired data is automatically discarded from the archive. If legislation changes require data to be kept longer, you simply alter the expiration point, and the rest will take care of itself.

Restart/Recovery

Archive operations often involve a very large amount of data, especially if you decide that they should take place very infrequently. Data Archiving for Adabas allows you to determine the restart/recovery characteristics too. You can set pacing levels that cause an archive operation to be performed in small steps, so that recovery and restart are possible, or you can demand that the whole operation completes in one step. It is your choice

Copy (only) Feature

You can choose to archive data but without removing the original content. For example, you might set a regular schedule so that every month 25% of a production file is copied to another Adabas file for testing purposes (and obviously run a program to disguise the data from the original production values for privacy, legal reasons).

Delete (only) Feature

Similarly, you can decide to use Data Archiving for Adabas to regularly remove data that is not to be sent to the archive because it serves no further purpose, for example, the expiration date has been reached.