

Data Archiving for Adabas

Introduction to Data Archiving for Adabas

Version 1.7.1

November 2016

This document applies to Data Archiving for Adabas Version 1.7.1.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2008-2016 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software AG.

Document ID: ADR-INTRO-171-20161124

Table of Contents

Preface	v
1 Data Life Cycle	1
2 Benefits	3
Managing a Growing Problem	4
Healthy Primary Systems	10
Cheaper Ancillary Systems	4
Long-Term Automation is Key	4
3 Key Features	5
Archive in Stages	11
Using Data Archiving for Adabas with Adabas Vista Partitioning	6
Using Data Archiving for Adabas with Adabas Vista Translation	11
Search the Archive Content	7
Recall from the Archive	7
Archiving is a Long-Term Activity	7
Adabas File Transfer	7
Delete (only) Feature	12
4 Components of Data Archiving for Adabas	9
Administration UI (User Interface)	10
Repository	11
Adabas Archive or Transfer Files	11
Vaults	11
Services	11
Extractors	11
Accumulators	12
Searches	12
Validations	12
5 Summary	13

Preface

Data Archiving for Adabas (ADR) provides the tooling to implement a well-managed, automated, accountable, secure place to store, search and recall data archived from Adabas.

In addition, this software provides a convenient way to perform Adabas file transfer within a single database, across databases within the same platform and across databases of different platforms.

Most, if not all enterprises experience database content growth at an increasing rate. Studies show as much as 85% of database content is inactive. This pattern emerges across all types of industry. There are many reasons for the growth in data content, too many to cover here. The fact is that growth is a continuing phenomenon which pressures primary production services. At the same time, there are increasing legislative reasons why information must be kept for longer, creating even more pressure.

Data Archiving for Adabas relieves the pressure by making it easy to archive and recall data on a large scale on a long-term basis.

[Data Life Cycle](#)

[Benefits Provided](#)

[Key Features](#)

[Components](#)

[Summary](#)

1 Data Life Cycle

Although 85% of database content is inactive this does not mean it is possible to discard it immediately. Inactive means the data is not a part of present-day business activity but it will be needed at some indeterminate time in the future AND it has to be kept for a minimum period of time before it can be discarded. In short, inactive does not mean unnecessary. There is a life cycle for data, a simple definition for the data life cycle might be:

- Active
- Recent
- Historical
- Expired

There are many examples of this life cycle. An everyday example is activity in your bank account. As follows:

- Active: A withdrawal is noted in a transaction ledger by your bank.
- Recent: The withdrawal appears in your statement.
- Historical: The bank makes sure the transaction is remembered (legal requirements) for a guaranteed amount of time.
- Expired: The details of the withdrawal no longer need to be kept for legal reasons, they can be discarded.

A similar overall life cycle applies to different sorts of data. For example, closing a bank account will pass through the same sort of life cycle, the timings may be different but the overall process is the same.

2 Benefits

- Managing a Growing Problem 4
- Healthy Primary Systems 10
- Cheaper Ancillary Systems 4
- Long-Term Automation is Key 4

Managing a Growing Problem

Large amounts of data bring large operational challenges. For example, planned or unplanned outages take longer because more disk space means more time to back-up, recover, reorder, etc. This all has a cost in one way or another. Data Archiving for Adabas implements a data life cycle to manage this problem, allowing the production database content to be focused on present-day activities.

Healthy Primary Systems

If up to 85% of your data can be managed in a different way to present-day data, you can make some clear economic savings. You can also heighten the level of service provision for your primary data services. By focusing your core systems on your primary data you will reduce outages, increase responsiveness, consume less primary disk space and you will be able to focus on increasing primary systems availability and capacity.

Cheaper Ancillary Systems

At the same time as attaining focus on your primary systems you can use Data Archiving for Adabas to make historical data available in a less expensive infrastructure. You can archive your inactive database content to less expensive computers, architectures and disk storage without losing the ability to search the content later or indeed recall it to its original form.

Long-Term Automation is Key

It is certainly feasible for database content to be archived manually using standard utilities, perhaps mixed with some specially written programs. Manual mechanisms are high-risk in terms of data consistency and legal compliance, especially because the amount of data to be archived is growing and the period the data must remain archived is increasing. What is needed is software tooling that delivers a well-managed infrastructure that provides a consistent, automated, systematic process that can be relied upon for a very long period. Data Archiving for Adabas delivers the automation needed to operate a valuable, reliable archiving system asset over a long period of time.

3 Key Features

- Archive in Stages 11
- Using Data Archiving for Adabas with Adabas Vista Partitioning 6
- Using Data Archiving for Adabas with Adabas Vista Translation 11
- Search the Archive Content 7
- Recall from the Archive 7
- Archiving is a Long-Term Activity 7
- Adabas File Transfer 7
- Delete (only) Feature 12

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 $(12 \times 10 = 120)$ 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.

Adabas File Transfer

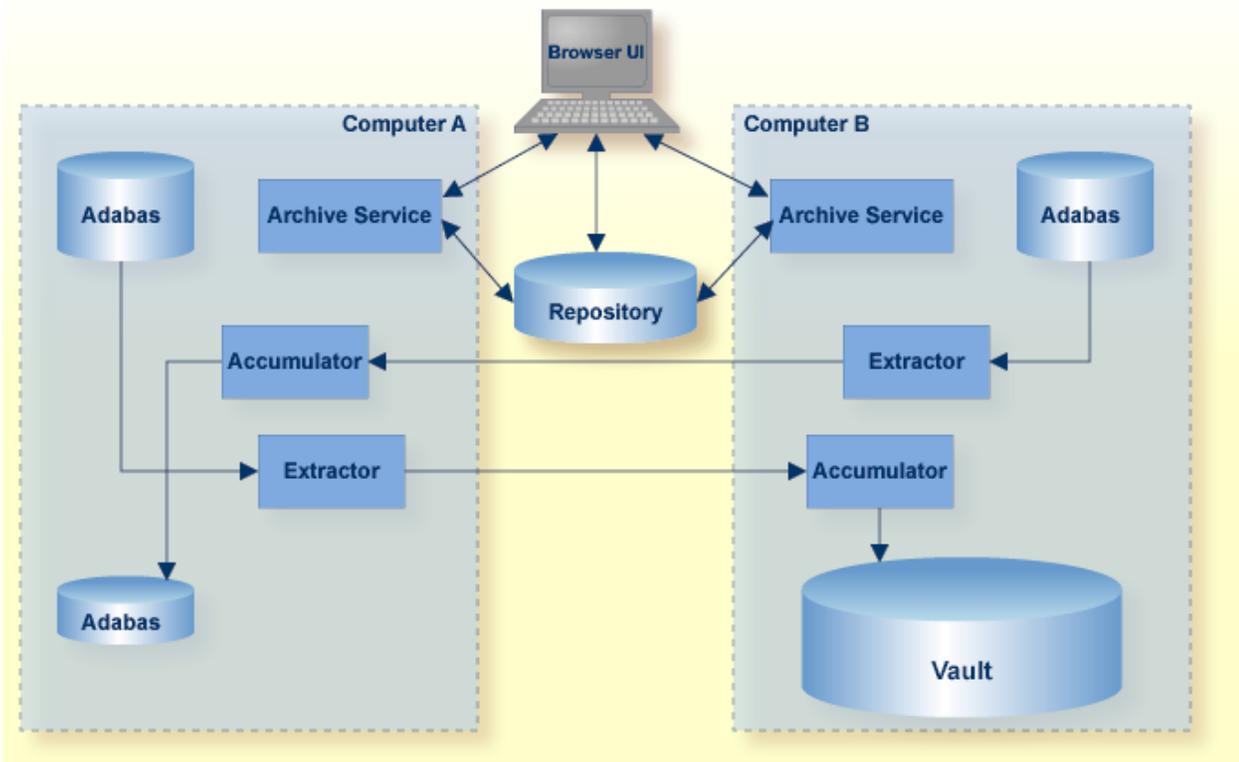
You can choose to transfer data to another file instead of archiving to the vault. In doing so you can choose to move the data or just copy it, your choice. This can be performed adhoc or to a regular schedule. For example, you might set a regular schedule so that every month 25% of a production file is transferred (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.

4 Components of Data Archiving for Adabas

▪ Administration UI (User Interface)	10
▪ Repository	11
▪ Adabas Archive or Transfer Files	11
▪ Vaults	11
▪ Services	11
▪ Extractors	11
▪ Accumulators	12
▪ Searches	12
▪ Validations	12



Administration UI (User Interface)

The UI is a browser-based graphical sibling of the Adabas Manager (AMA). The UI helps you to:

- configure the archive activities throughout your enterprise
- operate archive activities, assisted by automation
- search archive history and content
- recall archive content
- observe archive activities across your enterprise

Repository

All configuration data is stored in an Adabas file that you nominate.

Adabas Archive or Transfer Files

If you choose to archive or transfer Adabas files you must nominate the files that are used.

Vaults

You define the names of the files that form the flat-file store for the archive. Collectively they are referred to as the vault. You may decide to define more than one vault, for different business divisions for example, in order to keep them totally separate from each other.

Services

An archive service runs in each computer where archive operations are to be run in your enterprise. Archive services:

- run unattended
- manage archive operations according to the rules you set in the Repository
- launch extractors and accumulators according to the schedules you set
- launch searches, recalls, discard operations that you initiate
- record status/recovery/progress information to assist you in the UI

Extractors

Extractors extract the information you desire from Adabas according to the criteria you set in the Repository. Extractors:

- are launched by the Archive service according to your schedule
- run unattended
- communicate progress information to the Archive service

- communicate restart/recovery information to the Archive service
- transmit the extracted data to the peer accumulator according to the rules you state

Accumulators

Accumulators receive the data transmitted by the peer Extractor and place it into the destination you set in the Repository. Accumulators:

- are launched by the Archive service according to your schedule
- run unattended
- communicate progress information to the Archive service
- communicate restart/recovery information to the Archive service
- transmit the data to the destination you desire

Searches

You initiate searches through the Administration UI. Searches:

- are launched by the Archive service when you request them through the UI
- run unattended
- communicate progress information to the Archive service
- return search results to the UI
- retain search results for subsequent reuse

Validations

You define the validation of vault content that is to take place and when. Validations:

- are launched by the Archive service according to your schedule
- run unattended
- communicate progress information to the Archive service

5

Summary

Archiving is a data management issue on an extreme scale of time and volume. Archiving as an IT service must deal in decades as opposed to days and terabytes as opposed to gigabytes. The scale over long periods of time is significantly beyond what a database management system can comfortably handle. Data Archiving for Adabas enables you to provide a highly professional, modern and comprehensive archiving service to your organization as an integrated extension to your Adabas service.

