

File Partitioning

This section provides an overview of the benefits and features provided when using file partitioning as provided by Adabas Vista.

- When is Partitioning Required?
 - Implementing Partitioning within Application Systems
 - Implementing Partitioning using Adabas Vista
 - Typical Adabas Vista Deployment Scenarios
-

When is Partitioning Required?

File partitioning may be useful for any of the following reasons:

- data volume is too large to be managed within a single file;
- data partitions are needed for separate groups of users who use the same application but usually access different data;
- separate files need a consolidated view as well as an individual view;
- data archiving is needed with an efficient, high-performance recall capability without the need to develop a special recall application with a specific recall data source.

Implementing Partitioning within Application Systems

A partitioned file never provides the same performance levels as a standard file, whether the data management software is Adabas or some other system. The reason is that some form of navigation must occur to determine the location and relative content of partitions.

A number of performance options are available for computer systems. Some require application changes on a massive scale. The cost of these changes is not normally measured in the day-to-day operation of the computer, but they do exist and they are significant.

The burden of partitioning has fallen traditionally on application systems, making them extremely complex. For example, the application has often been required to decide the partition targeted by an access or update. The bigger problem is, however, that the usual level of application independence from the physical data model is lost. The cost of development and ongoing maintenance in these situations can be enormous.

Implementing Partitioning using Adabas Vista

- Benefits

- Data Partitioning Example

Benefits

Using Adabas Vista, a partitioned file can be spread across databases and computers, providing excellent scope for load balancing based on application requirements. In addition, navigation processing is performed in the client process, which increases the opportunity for parallel processing.

Adabas Vista can be used to partition data into separate Adabas files without having to re-construct the client applications which use these files. The applications continue to refer to one (simple) Adabas file entity that is accessed and updated as a whole. The physical data model is partitioned and can be distributed across a wide-ranging computer complex.

Adabas Vista addresses the problem of managing and productively using the massive physical capacity of an Adabas file, given commercial constraints on the amount of data that can be put in a single file. For example, a commercial requirement to limit outage to 15 minutes focuses attention on restore times and thus on the amount of data maintained in each file.

Because Adabas Vista processing is performed predominantly in the client process, any overheads associated with partitioning have minimal impact on the database service. Adabas is usually the busiest process by far within the computer. Normally, many thousands of clients concurrently use the same Adabas service. Adabas has proven its worth in these situations, providing excellent response times. The limiting factor for Adabas is often the processing capacity of the CPU. New ways are continually being sought to minimize database CPU consumption and to spread processing over as many CPU engines or even computers as possible.

Generally, the main overhead for partitioning is monitoring database requests to determine whether the target is a partitioned file and if so, which partition(s) needs to be used. Overall, this is referred to as navigation .

The cost of navigation

- is the same for 1 or 100 partitioned files because database requests must be monitored in any case; and
- does not increase with the addition of new partitioned files and the processing does not affect the capacity of the database.

In summary, using Adabas Vista to partition data

- ensures the independence of programs and data; and
- saves the time and money required for program changes.

Data Partitioning Example

The following example shows a typical use of data partitioning.

A company which has offices in various countries maintains a single file which contains information about all employees regardless of location. Each country is represented in the file by a 3-character code (D: Germany, UK: United Kingdom, USA: United States).

Country	Employee No.	Name
USA	100	Smith
D	200	Kenji
USA	300	Barker
UK	125	Smith
D	350	Matsui
USA	150	Wills
D	175	Smith
UK	325	Adams
UK	425	Fern

Using Adabas Vista, the same data may be partitioned into three separate files, organized by a partitioning field, in this case Country:

Country	Employee No.	Name
UK	125	Smith
UK	325	Adams
UK	425	Fern

Country	Employee No.	Name
USA	100	Smith
USA	300	Barker
USA	150	Wills

Country	Employee No.	Name
D	200	Kenji
D	350	Matsui
D	175	Smith

The organization may also have details of all the vehicles it has produced. Access to this data is usually for the current year, with occasional access to previous years. New records are only added for the current year and updates to the data are only permitted for the current year.

If this Vehicles file is partitioned on the field Production Year, updates (new records and updates to existing records) will only take place against a single partition. Regular backup and recovery procedures can be targeted to this single file, with an occasional backup for the entire data.

Typical Adabas Vista Deployment Scenarios

This section shows typical deployments of Adabas Vista:

Very Large Files

Very large files present an obvious use for partitioning. Although Adabas can hold a massive number of records within a file, the difficulties of handling volume remain. The key to managing mass data is the ability to divide the data into more easily managed units.

By using Adabas Vista to partition a large file across multiple databases, the processing load can be spread across the computer service. If the computer has more than one CPU engine, advantages are gained for all users by making greater use of the parallel availability of the CPU engines.

More and more sites are clustering computers to operate as one large service. Partitions can be placed on specific computers within the overall computer complex, thereby localizing the majority of data usage while maintaining the overall large file availability for management information.

The Adabas Vista partition outage feature can also be used to greatly increase the overall availability of data. This feature provides tolerance levels for partition availability. Application access to the partitioned file can be maintained transparently even though all the partitions of the file may not be available.

Archiving

Many organizations have a requirement to keep data for a specified period of time; often, many years. The sheer volume of data forces many organizations to archive, just to manage the data. In either case, an archiving and a recall process must be established and maintained.

Archiving is often a case of managing date-related data. However, types of data that are not related to date may also require archive processing.

A date-ordered file eventually contains old data that may not be used by most or all users. The old data can have an adverse effect on outage time because of additional data volumes, or perhaps because index sizes are too large making search times longer. Archiving is needed.

Most archiving systems access old data, copy it to archive, and delete it from the current file. This can take a considerable amount of time.

Adabas Vista offers a more flexible approach. Partitioning by date means that new data can be directed towards new partitions. Partition criteria (and sizing) might, for example, be based on handling a complete year, or month. Using the Adabas Vista partition restriction feature, old partitions can be left intact but made unavailable to all but a few users who require archived data.

The need for an archiving operation that interferes with normal operation as it searches through masses of data is thereby greatly reduced.

Should an emergency arise where older data is required urgently, the old database or file can simply be restored and the partition definition altered to make it available to the application again.

Using this approach, it is much easier to keep more old data available for longer, perhaps on remote (connected) computers.

Merger

Some organizations merge their operations for efficiency as well as growth. The merged units are often in the same business and therefore have similar systems and data, although perhaps not identical.

Using the Adabas Vista consolidation feature, a single file image may be imposed upon multiple, previously unrelated files. Although the files may be different, they support the same consolidated view.

For example, two organizations merge and each has its own accounting applications and files. Both support a common view. The new management requires information that is consolidated for the overall operation. However, the individual operations still need their old systems to operate as they did prior to the merge.

Using the Adabas Vista feature mixed mode access, both the individual and the consolidated views can be used by all applications. A new management information system can therefore be developed using the consolidated view without losing the operability of the previous systems.

Service Bureau

Smaller organizations often use a central computing service. Sometimes a common application such as payroll or accounts is used, but the independence of the data is mandatory. More often, a complete, separate copy of the application is made. This means, of course, that maintenance has to be rolled out to multiple services or repeated.

Adabas Vista makes it possible for multiple, separate parties to simultaneously share an application with separate data files managed as individual partitions of a simple, single file image. This can be accomplished using the previously mentioned features partition restriction and partition outage.

The cost saving alone is enough to justify using a single application for multiple, separate user groups.

The same effect can be achieved by using Adabas Vista definitions to partition a single, massive file without physically splitting the data. One physical file is used transparently as multiple, discreet entities.