# 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 Partitioning Deployment Scenarios
- Accessing Partitioned Files
- Additional Partitioning Features

---

## 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 Partitioning 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.

## Accessing Partitioned Files

This section describes the access modes which can be used to access partitioned files.

### Access Modes

Adabas Vista supports the following modes of access to a partitioned Adabas file:

- *focused mode* is based on the partitioning field and allows direct access to a particular partition.
- *distributed mode* is not based on the partitioning field and requires access to all partitions of a file.

- *mixed mode* allows a partition to be addressed directly by its real file number, even while using the single file image.

- Focused Mode
- Distributed Mode
- Mixed Mode

## Focused Mode

A file often has a field which is used primarily for access to the file. In fact, many sites regularly re-order files according to their dominant key field. When using Adabas Vista, this key field is most commonly used as the *partitioning field*.

Applications generally access a file using search data that is based, in whole or in part, on the primary key field. In these cases, Adabas Vista is able to detect the specific partition(s) needed to satisfy the access. This is referred to as *focused mode access*.

For example, a Natural application contains the following:

```
0250   FIND EMPLOYEES WITH COUNTRY = 'USA'
0260      DISPLAY NAME EMP-NUMBER COUNTRY
0270   END-FIND
```

Adabas Vista

- detects access explicitly or implicitly based upon the partitioning field Country;

- directs the access to the corresponding partition (USA) based on the search argument.

Using this access mode, Adabas Vista can focus access to only one partition, which results in a low processing overhead..

## Performance Notes:

Selecting the correct partitioning field is very important when first partitioning a file. It is often the case that files are regularly re-ordered according to a key in order to achieve greater I/O performance. Such a key is generally suitable as the Adabas Vista partitioning field.

If certain groups of users focus on specific portions of a file, the partition restriction feature may be beneficial, especially if access can be completely restricted to a partition. Restricting access increases the amount of focused access achieved.

## Distributed Mode

Adabas Vista also provides *distributed mode access* using data that is not related to the partitioning field. For example:

```
0250   FIND EMPLOYEES WITH NAME = 'SMITH'
0260      DISPLAY NAME EMP-NUMBER COUNTRY
0270   END-FIND
```

Access by Name cannot be focused onto one particular partition. Adabas Vista distributes the access to all partitions before collating the results in the expected order.

### Mixed Mode

Adabas Vista also provides mixed mode, in which applications can not only access a single file image for all partitions, but also can access a partition directly using its actual Adabas file number. This is referred to as *mixed mode access*.

# Additional Partitioning Features

- Partition Outage
- Independent Partition Maintenance
- Partition Restriction
- Consolidation
- Partition Sharing (MultiPart)
- Distributed Lock Mode
- Extreme files

## Partition Outage

This section describes the Adabas Vista partition outage feature.

**Note:**
Earlier versions of Adabas Vista referred to the partition outage feature as partial data operation (PDO). This method of specifying this feature is no longer supported.

- Description and Use of Partition Outage
- Defining Outage Tolerance using the Critical Parameter
- Dynamic Modification of the Critical Parameter

### Description and Use of Partition Outage

The partition outage feature can be used to manage planned outages for a partitioned file.

Using this feature, tolerance levels can be defined which control the action to be taken when a partition becomes unavailable. Application access to the partitioned file can be maintained transparently even though all the partitions of the file may not be available. Data is returned from the available partitions.

Sensitivity to partition outage can be set unilaterally or overridden on a user basis by an application. This provides flexible control over what happens when a partition becomes unavailable.

The suitability of this feature should be assessed for each partitioned file. For suitable files, using this feature:

- maximizes Adabas Vista file availability (24 * 7);

- maintains partitions transparently; and
- recovers partitions transparently.

Partition unavailability is determined by the following Adabas response codes:

| Response Code | Description |
|---|---|
| 17 | partition locked |
| 48 | partition locked by an Adabas utility |
| 148 | database not active |

### Defining Outage Tolerance using the Critical Parameter

The partition parameter `Critical` is used to specify the outage tolerance level for a partition.

As an example, assume that a file is partitioned by country.

Using partition outage, it will be possible to take only the UK partition offline. Access may continue as required for the other countries, but UK data is unavailable until returned online.

In that users in the UK work predominantly on UK data, partition outage can be set so that only UK data is critical to UK users. UK users tend not to use or need data from the other countries.

Consequently, UK users are unaffected by USA outages. However, UK outages do interrupt UK users because their critical data is affected. Similar arrangements can be made for the users in other countries.

### Dynamic Modification of the Critical Parameter

Adabas Vista API function PARTOPTS can also be used to to update or extract a user’s current Critical parameter setting for a specific partition. The API function CRITREP can be used to list those partitions that are not currently available.

## Independent Partition Maintenance

Since Adabas Vista partitions are actually independent Adabas files, it is possible to maintain them individually.

Independent partition maintenance can be used to size, order, and restore according to the needs of the individual partition. It is not necessary that all partitions adhere to the same physical constraints.The Adabas Associator space required by the partition can be individually fine tuned as well.

Separating partitions into independent Adabas files also allows you to perform parallel maintenance.

## Partition Restriction

**Note:**
Earlier versions of Adabas Vista referred to this feature as restricted segment option (RSO).

The partition restriction feature provides access control for each partition of a partitioned file. Partition restriction can either be implemented for all of the partition users or it can be defined so that certain applications are permitted to override the restriction at runtime.

Partition restriction:

- dynamically restricts user views of partitioned files,
- provides access based on user role controlled by the application,
- maximizes focused access.

This feature therefore provides the following benefits:

- Security: the availability of data can be restricted to authorized users
- Performance: the amount of focused access can be increased by limiting the number of partitions available to each user based upon role, location, or other criteria.

The type of access to a partition (none, read-only, read/write, or only) is specified using the parameter `Access`.

## Consolidation

Adabas Vista partitions do not have to be identical. Provided all the partitions support all of the views to be used by Adabas Vista, the files can operate with different physical layouts (FDTs). Of course, the Adabas source fields that are common to all partitions must be defined identically in each FDT.

For example, if an organization combines with a similar trading company, it is likely that the two companies hold similar data. In this case, a new file for another country could be added to the partitioned definition.

## Partition Sharing (MultiPart)

The partition sharing (multipart) feature can be used to define multiple partitions of the same partitioned file in order to share the same Adabas file while preserving the collating sequence of the single file image. In this way, specific portions of a file can be split away from the main file with minimal data manipulation.

This feature is activated using the parameter `Shared Partition`.

## Distributed Lock Mode

The distributed lock mode feature provides greater control of record hold logic in a partitioned environment.

When multiple partitions are targeted for distributed access and hold processing is involved, it is possible for successive accesses within the same process to be satisfied from the same partition. However, the other partitions still have another record within the same process on hold because that record is taking part in collating-sequence checking to ensure that the user receives the data in the correct sequence across partitions. If as a result, records are held for longer periods than expected, the frequency of the following response codes may increase:

- response code 145 because of record contention; and
- response code 9, subcode 2 because of Adabas transaction timeout (`TT`) expiration.

The `GET` statement should be used to hold records immediately before modification. This minimizes hold queue sizes and contention. This is even more relevant when using partitioned files.

If this recommendation cannot be followed, it may be useful to adjust the runtime control `Distributed Lock Mode`.

## Extreme files

Extreme files are a new option; standard partitioned files remain the default. Extreme files allow your applications to use files that are essentially infinite in size within the constraints of today's technologies. This feature is implemented in a way to make it as close to 100% transparent to your applications as possible; hopefully leaving few or no changes needed to adopt extreme files. You can set extreme partitioned files in one of two styles depending upon the needs of your application systems. You choose either field style or ISN style. The field style allows you to use extreme files without demanding wholesale changes to your applications. The ISN approach demands your applications always use the ACBX style of commands for the extreme file.

- Extreme files by field
- Extreme files by ISN

### Extreme files by field

The *field* approach allows you to use the traditional ACB mode of commands which may be a good way to proceed, initially at least. The field style usually demands you add a special field to the FDT dedicated for use by Vista (4-byte packed). You should understand that no data is ever stored by Vista in the file for this field; it is simply a placeholder for Vista processing. However, some applications can even avoid this small change to the FDT. In cases where the field is needed (because the nature of the application logic in these cases prevents Vista from automatically working out what to do) the field must be identified as the first field in the format buffer view used by the application. This allows Vista to invisibly manipulate this field to accomplish partition sensitivity for each record.

Most application programs are well-formed and so do not require the special field to be used at the start of the format buffer. If all your programs are well-formed it is possible the field is not needed in the FDT at all, making the introduction of extreme files absolutely transparent. There may be areas of your application that need specific control over the partition being affected by a command. The PARTID API is introduced to accommodate this requirement where it arises.

For more information see: Using extreme files and API function overview.

### Extreme files by ISN

The ISN style demands your application uses ACBX commands for extreme files. In this style Vista will use the senior portion of the appropriate 8-byte ISN field in ACBX to allow it to control partition processing similar to the way it uses the senior portion of the 4-byte ISN for standard partitioned files. This demands your application uses 8-byte ISN too.

**Note:**
It is intended that Natural will use this feature transparently in a future release, thereby avoiding wholesale application changes.

There may be areas of your application that need specific control over the partition being affected by a command. The PARTID API is introduced to accommodate this requirement where it arises.

For more information see: Using extreme files and API function overview.