

Using Adabas Vista with Adabas

This section provides programming guidelines for using Adabas Vista with Adabas:

- Adabas Vista ISN
 - Adabas Commands
 - Adabas Parameters
 - Adabas Security and Ciphering
 - ETID
 - Transaction Directives
 - CID Handling
 - ET Data
 - Collation Descriptors
 - Read Sequential Direction Changes
-

Adabas Vista ISN

If an Adabas file is defined as a standard Adabas Vista partitioned file, each Adabas ISN is modified with a partition identifier before returning the ISN to the application. Such a modified ISN is called an Adabas Vista ISN. Correspondingly, any ISN received from an application program for such a file is interpreted as an Adabas Vista ISN.

At the time the partition is defined, the space for a partition identifier is reserved within the 4-byte ISN field. The size of this area is based on the Adabas Vista file parameter `Maximum Number of Partitions`.

For example, the format of an Adabas Vista ISN for a standard partitioned file defined with a `Maximum Number of Partitions` set to 255 is as follows:

| Offset | Length | Contents | Meaning |
|--------|--------|-----------------------|--|
| 00 | 1 | X'01' - X'FF' | The ID of the partition from which the record was retrieved. |
| 01 | 3 | X'000000' - X'FFFFFF' | The Adabas ISN of the record within the physical partition |

Application programs that manipulate ISNs must be capable of handling the full 4-byte binary value returned. For Natural applications, any variable assigned to an ISN value should be defined with a minimum of P10. A value lower than this reduces the number of usable partitions and may result in a NAT1305 error code.

Caution:

Special care is required when an application uses an ISN value as data. If a partitioned file is re-partitioned, the stored ISN may well become inaccurate.

Caution:

When converting an Adabas file to a partitioned file, any ISNs for the file that may have been stored as data in other files must go through a conversion process to change the ISN values stored into correct Adabas Vista ISN values.

Note:

Adabas Vista partitioned files may be defined as "extreme", supporting > 4 gig ISNs. See Using extreme files for more information.

Adabas Commands

E1 Command (Refresh File)

If an E1 command is issued with an ISN value of zero and with a blank Command ID, the file is refreshed. This is also true for a partitioned file.

Note:

The E1 command will be issued to each partition in turn. Any error which occurs when a later partition is refreshed (for example, Adabas response code 114 because the partition was not loaded with PGMREFRESH=YES specified) will result in a partially refreshed partitioned file.

S5 Command

The S5 command is not supported for Partitioned Files.

Record Hold Logic

When using Adabas Vista, it is recommended to use the GET (L4) command to hold records immediately before modifying them. This is particularly relevant to those programs that would otherwise use the READ (L6) and FIND (S4) commands against a partitioned file.

Refer also to the section Distributed Lock Mode.

Read Sequential Direction Changes

When issuing an L9 command sequence, a second direction change is not allowed for distributed access.

When processing a direction change during an L9 command sequence, Adabas Vista places restrictions on format and length overrides in the search buffer.

Release Global Format IDs

A RC command which uses the C, L or O command options in order to release specific or all global format IDs will only be issued to the database specified in the command. In addition, the command options will always be changed to C to ensure that all global format IDs generated by Adabas Vista are also released by Adabas.

Optional ISN Positioning

Any program that accesses a partitioned file and requires the use of ISN positioning (for example, a starting ISN on an initialL3 command, or a minimum ISN value on an initial Sx command) must ensure that the supplied ISN is in Adabas Vista format. Adabas Vista then targets the appropriate partitions of the partitioned file.

Command ID within a Search Expression

Adabas Vista does not support a Command ID value enclosed in parentheses as a search expression for partitioned file access.

Adabas Parameters

The following Adabas ADARUN parameters may need to be tuned for use with Adabas Vista.

| Adabas Parameter | Description |
|----------------------|---|
| LI | <p>If a single file is partitioned into n files within the same database, it may obtain as many as n TBI elements per user as required for Adabas Vista distributed access.</p> <p>The increase in TBI elements depends on the amount of this distributed access and the number of partitions used in the average request. Therefore, the value of the LI parameter may need to be increased.</p> |
| LQ | The LQ parameter is similar to the LI parameter above. The LQ pool may need to be increased. |
| LU | The LU parameter value should be set to 65535 whether or not Adabas Vista is used. This requires a minimum NABparameter setting of 16. |
| NAB | The NAB parameter value may need to be increased if the Adabas Vista read-ahead feature is used and/or a significant number of partitions are defined to the database. |
| NC | If one or more partitions are added to a database for the first time, the value for the NC parameter may need to be increased. |
| NH | See the NISNHQ parameter below. |
| NISNHQ | <p>Along with the NH parameter, this parameter may need to be increased if</p> <ul style="list-style-type: none"> ● a significant number of new partitions are added to the database and hold-based commands are issued because of distributed access. ● the Adabas Vista read-ahead feature is used with hold-based commands. |
| NQCID | Similar to theLI parameter. The value for NQCID may need to be increased if a significant number of new partitions are added to the database and significant use of TBQ elements occurs because of distributed access. |
| NU | If one or more partitions are added to a database for the first time, the NU parameter value may need to be increased. |
| PREFETCH | This parameter is needed to control the Adabas Vista read-ahead feature at the job level. No other Adabas prefetch parameter is necessary. |
| TNAA, TNAE, TNAX, TT | <p>Where partitions span more than one database, it is possible that a user may, during a series of related accesses, request numerous consecutive records from the same database within a distributed access. This may result in Adabas timing the user out on the other databases involved in the distributed access, which produces a response code 9 when Adabas Vista tries to continue the sequence on these databases.</p> <p>The values for the parameters TNAA , TNAE , TNAX, and TT should therefore be aggregated to the highest values for all databases containing partitions.</p> |

Adabas Security and Ciphering

When using Adabas Vista partitioned files, Adabas passwords and cipher codes must be the same for each partition.

ETID

Software AG strongly recommends that the use of ETID is consistent (identical) across all databases used for a client session.



Warning:

It is possible to dynamically change Vista translation rules (etc.) in the configuration file and apply the changes to a running system. Vista accommodates this as best as possible in running applications. For example, all active cursors (read loops, find loops, etc.) and active transactions must be allowed to complete using the former set of rules (on a per session basis) before dynamic switchover to the new rules takes place. This covers most but not all situations. For example, there can be implications for handling ET data if rule changes alter the location where ET data are stored; unpredictable results are possible - especially if the rarely used situation of using ET data without setting ETID is used by the application. However, altering the location of the ET data store using only standard Adabas (without Vista) is very problematic since applications need to be fully coordinated to all look to the new location (only after all ET data has been moved), otherwise unpredictable results can occur, this is no different when Vista is used.

Transaction Directives

Adabas Vista generates transaction directives in accordance with standard distributed transactions making it compliant with the use of Adabas Transaction Manager. Correct distributed transaction processing is enforced by Adabas Vista (that is, an application is expected to be well-formed in the way it pursues transaction outcome). To explain...where multiple databases are modified the application is expected to pursue transaction outcome to all the changed databases, in series. For example, if two databases are modified then Adabas Vista expects to see two ET commands (one to each database) to cause a positive transaction outcome; or two BT commands (one to each database) for a negative transaction outcome. If an application tries to issue a transactional command without having fully issued all ET/BT commands for all databases in the previous transaction then RSP249 sub-code 496 will be issued. This behavior is compliant with distributed transaction processing standards, as implemented by Adabas Transaction Manager.

The runtime control `ET Data Database Number` is used to specify the location for transaction data.

Extended Hold

The P and M command options allow a program to commit (ET) or back out (BT) a transaction while leaving some records in hold status. Adabas Vista supports these options using the `Extended Hold` runtime control. Extended hold processing proceeds as follows:

- `Extended Hold = Minimum` (default)
 - When Adabas Vista receives the first transaction directive, it has no knowledge of any P or M options that are to be applied to the databases involved in the transaction except those specified on this first directive.

- Adabas Vista therefore honors the P or M option specified for the source database number of the first directive. It then generates the necessary directives for all other source database numbers without regard to P or M options. Thus, the client's transaction is committed or rolled back as requested, P or M options on the first directive honored and all other ISNs released off hold.
- Extended Hold = Maximum
 - When Adabas Vista receives the first transaction directive, it has no knowledge of any P or M options that are to be applied to the databases involved in the transaction except those specified on this first directive.
 - Adabas Vista therefore honors the P or M option specified for the source database number of the first directive, but ensures that held records put on hold using other source databases are not released during the commit or roll-back process. Thus, the client's transaction is committed or rolled back as requested, but at this stage some of the transaction's held records may still be in hold status.

As each subsequent directive (in the sequence) is issued, Adabas Vista honours the P or M option specified for those source database numbers. When the Vista proxy detects "end of directive sequence", the proxy generates the necessary directives for each source database number that was not party to the client's directive sequence.

- If the application program fails to terminate the sequence of directives, records may unintentionally remain in hold status in one or more target databases. In this case, it is necessary to intervene manually by operator command or Adabas Online System to stop the client and free its resources; otherwise, the resources are freed when the client's session times out.

CID Handling

Translation and partitioning features mean that CID commands may be re-directed to other databases/files which can cause inadvertent clashing of CIDs inside Adabas databases. Consequently the CID handling in Adabas Vista has now been strengthened to avoid this happening.

ET Data

If an ET or CL command with ET data is issued by a client and, as a result of this, Vista has to generate a series of such commands, then Vista will always issue the ET data on the first ET or CL of the series. This is in accordance with standard distributed transaction protocols making it compliant with the use of Adabas Transaction Manager.

The runtime control Database Number for ET Data is used to specify the location for transaction data.

**Warning:**

It is possible to dynamically change Vista translation rules (etc.) in the configuration file and apply the changes to a running system. Vista accommodates this as best as possible in running applications. For example, all active cursors (read loops, find loops, etc.) and active transactions must be allowed to complete using the former set of rules (on a per session basis) before dynamic switchover to the new rules takes place. This covers most but not all situations. For example, there can be implications for handling ET data if rule changes alter the location where ET data are stored; unpredictable results are possible - especially if the rarely used situation of using ET data without setting ETID is used by the application. However, altering the location of the ET data store using only standard Adabas (without Vista) is very problematic since applications need to be fully coordinated to all look to the new location (only after all ET data has been moved), otherwise unpredictable results can occur, this is no different when Vista is used.

Collation Descriptors

Adabas Vista does not support the use of collation descriptors for partitioned files.

Read Sequential Direction Changes

Changing the direction of an L3 or L9 sequence when the sequence has been previously subjected to multifetch is not allowed against a Partitioned File. This is consistent with the behaviour of ADARUN PREFETCH (refer to RSP021 sub-code 9 in the *Adabas Messages and Codes* documentation).