

## **Adabas Vista**

### **Adabas Vista Programming Guidelines**

Version 7.4.2

September 2009

This document applies to Adabas Vista Version 7.4.2 and to all subsequent releases.

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

Copyright © Software AG 2009. All rights reserved.

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

## Table of Contents

1 Adabas Vista Programming Guidelines .....	1
2 Using Adabas Vista with Adabas .....	3
Adabas Vista ISN .....	4
Adabas Commands .....	5
Adabas Parameters .....	7
Adabas Security and Ciphering .....	8
Adabas SQL Server .....	8
3 Additional Programming-Related Topics .....	9
Command Limit .....	10
Error Handling .....	10
Multiclient File Processing .....	11
Shared Partitions .....	11
Single Partition Focus .....	11
Updating the Partitioning Field .....	12
4 Adabas Vista Application Programming Interface .....	13
Executing APIs with Natural and 3GL Programs .....	14
API Function Overview .....	18
API Function Descriptions .....	16
5 Using Adabas Vista with Adabas Cluster Services .....	19
Clustered Applications .....	20
Adabas Cluster Services .....	20
Adabas System Coordinator .....	21
Job Parameters for Clustered Applications .....	21
6 Adabas Vista User Memory Requirements .....	23
Index .....	27



# 1 Adabas Vista Programming Guidelines

---

This document provides information related to programming applications for use with Adabas Vista.

The following topics are provided:

- [Using Adabas Vista with Adabas](#)
- [Additional Programming-Related Topics](#)
- [Adabas Vista Application Programming Interface](#)
- [Using Adabas Vista with Cluster Services](#)
- [Adabas Vista User Memory Requirements](#)



# 2 Using Adabas Vista with Adabas

---

- Adabas Vista ISN ..... 4
- Adabas Commands ..... 5
- Adabas Parameters ..... 7
- Adabas Security and Ciphering ..... 8
- Adabas SQL Server ..... 8

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

## Adabas Vista ISN

---



If an Adabas file is defined as an 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 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.



---

## 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.

### ET Command

An ET (END TRANSACTION) command commits all outstanding updates for the current transaction.

The Natural END TRANSACTION statement accomplishes this by issuing ET commands to all databases involved in the current transaction. For multiple databases, Natural issues an ET serially to each database.

3GL programs should also adopt this logic.

Adabas Vista uses the same principle. On receipt of an ET command, serial ET commands are issued to all appropriate databases involved in the user transaction.

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

### ET and CL Commands with Command Option 'E'

ET and CL commands may use Command Option 'E' to indicate that user data is to be stored in an Adabas system file. When Adabas Vista encounters this command, it may issue multiple commands to different databases and will always issue the command with Command Option 'E' last.

### S5 Command (Complex Find)

The S5 command is only available for use with Adabas Vista partitioned files when the [Single Partition Focus](#) feature is enabled.

## 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`.

## Prefetch/Multifetch

Adabas Vista performs its own read-ahead processing. For an Adabas Vista application which uses the command options `P`, `M` or `O` to invoke prefetch/multifetch, Adabas Vista will return an appropriate number of records as per Adabas.

## Read Sequential Direction Changes

When the Adabas Vista read-ahead feature is enabled, a direction change during a sequential read is not allowed within the following command constructs:

- `L3` command with no accompanying value re-position; and
- `L9` command.

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 initial `L3` 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 <math>n</math> files within the same database, it may obtain as many as <math>n</math> 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"> <li>■ a significant number of new partitions are added to the database and hold-based commands are issued because of distributed access.</li> <li>■ the Adabas Vista read-ahead feature is used with hold-based commands.</li> </ul>
NQCID	Similar to the LI 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	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

Adabas Parameter	Description
	<p>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 TNA A , TNA E , 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.

## Adabas SQL Server

---

Adabas Vista partitioning can be used without restriction for normal SQL statements that refer to data within a table. Adabas Vista partitioning is not supported for the Adabas files that comprise the SQL catalog. SQL statements that perform physical management of a table are not supported for Adabas Vista partitions. Physical management of Adabas Vista partitions must be performed by Adabas utilities.

# 3 Additional Programming-Related Topics

---

- Command Limit ..... 10
- Error Handling ..... 10
- Multiclient File Processing ..... 11
- Shared Partitions ..... 11
- Single Partition Focus ..... 11
- Updating the Partitioning Field ..... 12

This section provides additional information related to programming applications for use with Adabas Vista:

## Command Limit

---

Each target category for a source profile contains an optional command limit, in addition to the target information. Response code 249, subcode 129 is returned on the first command which exceeds this limit. A command limit of 0 (zero) means that access to this particular file is unlimited.

## Error Handling

---

Adabas Vista uses the default Adabas response code 249 to indicate that an Adabas Vista processing error has occurred. This response code can be modified using the job parameter Error Response Code.

To qualify the error, a non-zero binary value is placed in the subcode field (the low-order 2-bytes of the Adabas Control Block Additions 2 field). Possible values for this subcode are provided in section Messages and Codes.

The application programmer can convert the response subcodes to error message text using the ERRM function of the Adabas Vista API (see [Application Programming Interface](#)).

When using Natural, the subcode may be retrieved using the USR0610N subprogram in library SYSEXT. This enables the following construct for an ON ERROR block:

```
0250 READ AVI1 LOGICAL BY NAME
0260     DISPLAY FIRST-NAME LAST-NAME SEX
0270 END-READ
0280 ON ERROR
0290 IF *ERROR-NR = 3249
0300     CALLNAT 'USR0610N' DB_ERR_STR
0310     WRITE 'AVI error, subcode:' DB_SUBCODE
0320 END-IF
0330 END-ERROR
0340 END
```

It is strongly recommended that error handling is implemented so that error conditions can be identified and handled immediately.

Refer to the job parameter `Error Reporting` for additional error reporting options.

## Multiclient File Processing

---

Adabas Vista is able to process multiclient files. There are a few considerations that should be made when using this type of file with Adabas Vista:

- Adabas Vista does not support superuser Owner IDs with distributed access. The sequence of records returned will not be the same as for a normal Adabas multiclient file. These superusers are most likely to be used for utilities, which are not normally run with Adabas Vista.
- Adabas Vista enforces the same ETID setting across all databases used simultaneously for a client session.
- Adabas Vista cannot use the internal Owner ID as partition criteria. See the file parameter `Partitioning Field` for valid field types.

## Shared Partitions

---

When processing Adabas Vista files that have some partitions shared, extra processing is required to determine the correct partition for a particular record, and to ensure that records are returned to the application in the most appropriate sequence for the command. This extra processing may have an adverse effect on performance for these command sequences.

For example, an `L1` command sequence processes all records in a file that is defined to Adabas Vista with three partitions, the first and third partitions sharing a single file. Adabas Vista will scan through the first file and extract all records that logically belong to the first partition. The second file will be processed as normal, and Adabas Vista will then scan through the first file again to extract the records that logically belong to the third partition.

When issuing an `L9` command against a shared partition where the descriptor used is not the partitioning field, although the count returned is correct, the ISN returned in the ISN Lower Limit field may indicate an incorrect partition. It is not possible to differentiate between the partitions in a shared file for an `L9` command.

## Single Partition Focus

---

The Single Partition Focus feature can be used to define temporarily a partitioned file as having only one partition. Any command restrictions that may apply to partitioned files are lifted. Alternatively, you can set one of the partitions to `FULL` access and the others to `NONE` to achieve the same result.

You may enable single partition focus as the default for a partitioned file by using Adabas Vista Online Services. You may enable it for a single session by using the provided Adabas Vista API (example program API004UP).



**Caution:** If single partition focus is specified for a partition of a partitioned file, the validation of the partitioning field value against the defined criteria for that partition is no longer performed. This means that records whose partitioning field value does not correspond to the defined partitioning criteria may be updated or stored in that partition. If single partition focus is subsequently lifted and focused access performed against this partitioned file, such records may not be found. For this reason, it is recommended to restrict the use of single partition focus to those partitioned files for which no focused access is performed.

### Example

A file is defined with two partitions and the partitioning field high value is set to M and Z, respectively. Single partition focus is specified for the first partition.

As a result, a record with the value T is added to the first partition.

When single partition focus is removed, a subsequent focused access for all records with the value T is directed to the second partition only.

## Updating the Partitioning Field

---

A partitioning field can only be updated if there is no requirement to relocate the record from the Adabas file in which it is stored. If this is not the case, a response code is returned. If such an update is required, a `DELETE` and `STORE` operation must be performed.



# 4 Adabas Vista Application Programming Interface

---

- Executing APIs with Natural and 3GL Programs ..... 14
- API Function Overview ..... 18
- API Function Descriptions ..... 16

Adabas Vista makes as much information as possible available to the application programmer through the Adabas Vista application programming interface (API).

All API functions are available for Natural and most API functions are also available for 3GL programs.

The Adabas Vista installation INPL dataset includes example programs that invoke each of the available API functions. These functions can be executed by the application for whom the information is relevant.

The necessary Natural objects must be installed according to the instructions provided in the section Installation.

## Executing APIs with Natural and 3GL Programs

---

Any API invoked changes only remain active for the duration of a user's session.

In a Natural environment, if you log on to the Adabas Vista library to execute these API functions the change may be lost when you log back on to your application library. This is due to Natural's session handling.

To avoid losing such changes, make the API example programs available without the need to log on to another library:

- If Natural Security is in use, include the Adabas Vista library *SYSMVors* in the steplib of the application library.
- If Natural Security is not in use, copy the appropriate API program to the Natural SYSTEM library along with the objects *API0000P* and *API00nVP* where *n* is the API function number.

In a 3GL environment, the supplied AVIPAPI program enables a 3GL program to utilise these API functions. Each API function available for use by a 3GL program has an example program in the Adabas Vista source library. These programs are named *APIASMnn*, where *nn* corresponds to the Natural example program number in the table below. For example, *APIASM09* is the sample 3GL API program for the CATEGORY function.

## API Function Overview

The following table lists the available API functions:

Function	Option	Description	Example
CATEGORY	UPDATE	Modifies the user's current target category.	API009UP
	EXTRACT	Extracts the user's current target category.	
CLOG		Traces user commands (only if job parameter Trace is active).	API001UP / AVICLOG
CONTEXT	UPDATE	Modifies the user's current context. Also allows the temporary suspension of the user's current context. Possible values:  S: single context M: multiple contexts (suspend the current context).	API010UP
	EXTRACT	Extracts the user's current context. Possible values:  ' ': current context L: list all user contexts	
	DELETE	Deletes a specified suspended context, or all suspended contexts for the user. Possible values:  ' ': specified context A: all suspended contexts.	
CONVISN	ADATOAVI	Converts an ISN from Adabas format to Adabas Vista format (including Partition ID).	API007UP
	AVITOADDA	Converts an ISN from Adabas Vista format to Adabas format.	
CRITREP		Lists partitions identified as unavailable for the specified partition file.	API005UP
ERRM		Translates a supplied Adabas Vista subcode into an error text message.	API002UP
PARTLIST		Lists all partitions for the specified partitioned file.	API006UP
PARTOPTS	UPDATE	Modifies the user's current Access and Critical parameter values for the specified partitioned file and partition.	API004UP
	EXTRACT	Extracts the user's current Access and Critical parameter values for the specified partitioned file and partition.	
PROFID	UPDATE	Modifies the user's current Profile ID.	API003UP
	EXTRACT	Extracts the user's current Profile ID.	
TRTARG	UPDATE	Modifies a specific translate rule for a user.	API008UP
	EXTRACT	Extracts the details of a user's current translation rule for a given source profile.	



**Note:** In a Natural environment, the use of multiple applications may mean that Natural issues close (CL) commands. Therefore, any API calls must be re-issued at each application logon.

## API Function Descriptions

---

- [CATEGORY Function](#)
- [CONTEXT Function](#)
- [PARTOPTS Function](#)
- [PROFID Function](#)
- [TRTARG Function](#)

### CATEGORY Function

During application logon, each client may identify its own target category to Adabas Vista using the CATEGORY function. Adabas Vista then uses this Target Category ID together with the Profile ID, source database, and file number to determine the appropriate target database and file number to which commands issued by the application are to be directed.

This function can be executed any number of times during a session to enable dynamic file translation.

The use of the update option of this function will cause an internal backout of all access by this application. Adabas Vista will issue BT and CL commands to all relevant databases to effect this backout.

### CONTEXT Function

During application logon, the current context is defined by the Profile ID and Target Category parameters defined for the current job. Each parameter may be dynamically modified by use of the appropriate API function. The CONTEXT function extends this processing.

This function may be used to suspend temporarily the current context for the client and continue processing in a new context. The client application controls the use of such contexts and determines at which point a suspended context should be resumed.

The DELETE option enables the application to terminate any context that is currently suspended. Adabas Vista will issue RC commands at this point.

## PARTOPTS Function

The PARTOPTS function is used to update or extract the user's current `Access` and `Critical` partition parameter values for the specified partitioned file and partition.

The following `Access` values are available:

Value	Description
F	Set the <code>Access</code> partition parameter value to FULL.
R	Set the <code>Access</code> partition parameter value to READ.
N	Set the <code>Access</code> partition parameter value to NONE.
O	Enable the single partition focus feature for this partition.
X	Reset all <code>Access</code> partition parameter values to the defaults.

The following `Critical` values are available:

Value	Description
Y	Set the <code>Critical</code> partition parameter value to YES.
N	Set the <code>Critical</code> partition parameter value to NO.

## PROFID Function

During application logon, each client may identify its own Profile ID to Adabas Vista using the PROFID function. Adabas Vista then matches this ID with the Profile ID of a translation rule or partition definition that uses the same source database and file number to re-direct commands issued by the client to the appropriate target database and file number.

This function can be executed any number of times during a session to enable dynamic file translation.

The use of the update option will cause an internal backout of all access by this client. Adabas Vista will issue `BT` and `CL` commands to all relevant databases to effect this backout.



**Note:** The `CATEGORY` and `PROFID` functions may be used in conjunction with this function to enable dynamic file translation.

## TRTARG Function

This function can be used to extract or update a specific target database and file number. The client application provides the source database and file number, which is combined with the client's current Profile ID. Using this source profile and the client's current target category, the appropriate details are extracted or updated as required. The update action is only available for translation rules.

The use of the update option will cause an internal backout of access to the specified file. Adabas Vista will issue RC command(s) to effect this backout.



**Note:** This API can only be used to extract/update details for a 'file' translation profile - it cannot be used to extract/update details for an 'all-files' profile.

# 5 Using Adabas Vista with Adabas Cluster Services

---

- Clustered Applications ..... 20
- Adabas Cluster Services ..... 20
- Adabas System Coordinator ..... 21
- Job Parameters for Clustered Applications ..... 21

A clustered operating system is a collection of independent operating system images working together as one. The objective is to spread work around the cluster to achieve better load balancing, throughput, and availability.

Clustered applications appeared before their operating system counterpart. Application clusters operated over multiple processes within a single operating system image. Now clustered applications are enhanced to operate throughout an operating system cluster.

Adabas Cluster Services allows multiple instances of the Adabas server to operate against the same database across multiple operating system images, which enhances the general clustering approach for high-end systems.

## Clustered Applications

---

A clustered application is a collection of independent jobs operating together as a single service.

Each implementation of a clustered application allows client sessions to migrate from one job to another across one or more images in an operating system cluster. The memory associated with a particular client session must be migrated along with the client session itself.

The following clustered applications may be implemented within a single operating system image:

- CICS/MRO with dynamic transaction routing in z/OS, OS/390 and VSE/ESA
- IMS/TM in z/OS and OS/390
- UTM in BS2000

The following clustered applications are available for operating system clusters:

- CICSplex in z/OS and OS/390
- IMSplex in z/Os and OS/390

## Adabas Cluster Services

---

Adabas Cluster Services allows you to use multiple images of the Adabas server across an operating system cluster in order to provide truly scalable throughput.



## Adabas System Coordinator

Support for clustered applications within a single system image or across multiple system images in an operating system cluster is enabled by the Adabas System Coordinator when running one or more of the following Adabas add-on products:

- Adabas Fastpath
- Adabas Vista
- Adabas Transaction Manager
- Adabas SAF Security

Adabas Fastpath also uses the Adabas System Coordinator to host the Asynchronous Buffer Manager service.

## Job Parameters for Clustered Applications

When defining a job parameter for CICS/MRO with dynamic transaction routing or for CICSplex, select the job type CICS Cluster.

Clustered application job types have user context information allocated from shared memory rather than from local (Job) memory.

Additional fields are available to support clustered applications:



**Note:** For non-clustered applications, leave these fields empty.


Field	Used to...
System Coordinator Group Name	link the clustered application to the relevant Adabas System Coordinator group.
Clustered Application Service Name	group together all instances of the application to the Adabas System Coordinator.
Manage Terminal Sessions or Manage All Sessions	determine the type of sessions managed by the Adabas System Coordinator daemon.

Refer to the *Adabas System Coordinator* documentation for more information.



## 6 Adabas Vista User Memory Requirements

The following table lists the main memory requirements for an Adabas Vista user.

 **Note:** This table is not intended to be an exhaustive list of the memory requirements for an Adabas Vista user. It is recommended that you use this table during the initial analysis of a system's memory requirements. Once implemented, Adabas System Coordinator Online Services may be used to more accurately monitor the amount of real memory in use.

Memory requirement (per user) for each...		Size (in bytes)
user session area (one only)		512
database accessed		+ 160
file accessed	Adabas file	+ 80
	Adabas Vista translation file	+ 80 + 64 + 48
	Adabas Vista partitioned file	+ 80 + 64 + (48 + partitioning field length + user partition concurrency * 5) * number of partitions
Command ID issued against	an Adabas file	+ 80
	an Adabas Vista translation file	+ 36 + 80
	an Adabas Vista partitioned file	+ (256 + 80 * number of involved partitions) + (length of Adabas buffers * number of involved partitions)
record held (within a transaction) against	an Adabas file	+ 64
	an Adabas Vista translation file	+ 64
	an Adabas Vista partitioned file	+ 64 * number of involved partitions

**Example**

1. Estimate the per user memory requirements for an application which has the following access profile:
  - two databases
  - five Adabas files
  - two Adabas Vista translation files
  - one Adabas Vista partitioned file
    - Partitioning field length is eight bytes
    - User partition concurrency is eight
    - Number of partitions is three
    - Majority of access is distributed (all three partitions involved)
    - Average length of Adabas buffers is 1000 bytes
  - Maximum Command ID concurrency of two (one Adabas file and one Adabas Vista partitioned file)
  - Average number of successful hold commands between ET/BT is 20
    - 18 against Adabas files
    - two against Adabas Vista partitioned file

**Result**

User session area:		512
Database requirements:	$(2 * 160)$	320
File requirements:	$(5 * 80) + (2 * 192) + 1 * 432$	1216
Command ID requirements:	$80 + 3496$	3576
Held record requirements:	$18 * 64 + 2 * (64 * 3)$	1536
	Total (per user)	7160

2. Multiply this value by the estimated number of application clients to estimate the application memory requirements.
3. If a single TP system is host to multiple applications, repeat this process for each application to provide an overall estimate of the TP memory requirements.

The required memory is requested from either

- local TP memory using the host TP services; or
- shared memory using the Adabas System Coordinator daemon.

The use of shared memory depends on whether the job is defined as a clustered application using job types such as CICS Cluster, IMS, or UTM.



**Note:** It is strongly recommended that you discuss the use of such job types with your system programmer before you implement them.



## Index

---

