

About Adabas Commands

This section provides an overview of the Adabas commands categorized by function: database query, database modification, Data Storage read, Associator read, logical transaction processing, and special commands.

In addition, Adabas command facilities related to data protection, recovery, and user restart are described. The *transaction* concept is introduced and ET logic operations are explained. Competitive updating is discussed for ET logic (record hold/release and avoidance of resource deadlock) and exclusive control users; nonactivity timeouts are described for all user types.

This chapter covers the following topics:

- Command Types
 - Transactions and ET Logic
 - Competitive Updating
 - Non-Activity Time Limit
-

Command Types

The commands can be categorized into the following functions:

- Database Query Commands (Sx)
- Data Storage Read Commands (L1-L6)
- Associator Read Commands (L9, LF)
- Database Modification Commands (A1, E1, N1/N2)
- Logical Transaction Control Commands (ET/BT)
- Special Commands

Database Query Commands (Sx)

Database query commands (S1/S4, S2, S5) search for and return the ISNs of specified records or record groups according to specified search criteria. Other commands in this category (S8, S9) sort the resulting ISN lists in preparation for later operations.

The ISN lists resulting from any Sx command may be saved on the Adabas Work data set for later retrieval during your user session.

In most cases, these commands do not actually read the database; ISNs are read directly from the Associator's inverted lists. Options allow the ISN's record to be placed in hold status to prevent its being updated by other programs until the record is released; if desired, additional field values contained in the first ISN's record can be read from Data Storage.

Note:

The behavior of nondescriptor searches in Adabas databases differs between mainframe and open systems in regards to null suppression in the fields. In open systems, nondescriptor searches do not return records with null values in a field if the field is null-suppressed (NU); on mainframe systems, the null-suppression (NU) of fields is ignored during nondescriptor searches. At this time, to resolve this problem, we recommend that you remove the null suppression option (NU) for open systems fields, if the fields must be used for a nondescriptor search.

This section covers the following database query commands:

- S1/S4
- S2
- S5
- S8
- S9

S1/S4

The S1/S4 command selects the records that satisfy given search criteria.

- If only descriptors are used, the query is resolved using the inverted list alone without accessing Data Storage.
- If no descriptors are included in the search criteria, the S1/4 command reads each record in Data Storage to resolve the query.
- If both descriptors and nondescriptors are used within the search criteria, Adabas first searches the inverted list for the descriptors and then reads the Data Storage for all matching ISNs to check the nondescriptors.

S1 and S4 commands return the count of records that satisfy the search criterion, and a list of the ISNs for those records. An option permits the record identified by the first ISN in the resulting ISN list to be read from Data Storage.

The S4 command may be used to place the record identified by the first ISN in the ISN list in hold status. This prevents another user from updating the record until it is released.

S2

The S2 command is equivalent to the S1 command except that the ISNs of the records selected are returned in the sort sequence of a user-specified descriptor (or descriptors). One to three descriptors may be used. Ascending or descending sequence may be specified.

S5

The S5 command is used to select the ISNs of records in one file which are coupled to a record with a given ISN in another file.

You specify the file and ISN for which coupled ISNs are to be returned and the file in which the coupled records are to be selected. Adabas returns the number of records coupled to the ISN and the list of the coupled ISNs.

S8

The S8 command performs logical (AND, OR, or NOT) operations on two ISN lists previously created by an S1/S4, S5, S8, or S9 command.

- AND results in a single ISN list containing ISNs present in both source lists.
- OR results in a single ISN list containing ISNs present in either source list.
- NOT results in a single ISN list containing ISNs present in the first list but not in the second list.

S9

The S9 command sorts an ISN list created previously by a S1/S4, S2, S5, S8 or S9 command.

The ISN list may be sorted by ISN (ascending sequence only), or by one to three user-specified descriptors (ascending or descending sequence).

Data Storage Read Commands (L1-L6)

The L1 through L6 commands are used to read actual records from Data Storage. Depending on the specified command and its options, records are read individually in the sequence in which they are stored, in the order of an ISN list created by one of the database query commands, or in logical sequence according to a user-specified descriptor.

A hold option allows the database records to be locked until released by a separate command or at transaction end.

This section covers the following data storage read commands:

- L1/L4
- L2/L5
- L3/L6

L1/L4

The L1 command reads a single record from Data Storage. You specify the file number, ISN of the record to be read, and the fields for which values are to be returned. Adabas returns the requested field values.

The L4 command is the same as the L1 command except that the record is placed in hold status. This prevents other users from updating the record until it is released.

The multifetch/prefetch option prefetches records on either a session or command basis. This can reduce the overhead for multiple record fetches. The multifetch option is platform-independent.

The GET NEXT option may be used to read one or more records identified by ISNs contained in an ISN list without requiring that you specify each ISN. Usually, the ISN list is created by a previous Sx command.

The response code option issues a response code 145 (ADARSP145) if the L4 command cannot place a record in hold status because it is being held by another user. Otherwise, you are placed in wait status until the ISN (and record) are freed or the waiting user's transaction is timed out.

The ISN sequence option may be used to read records in ISN sequence. The ISN you specified is read, unless it is not present, in which case the record which has the next higher ISN is read.

L2/L5

The L2 command reads the records from a file in the sequence in which they are physically stored in Data Storage. You specify the file to be read and the fields for which values are to be returned. Adabas returns the requested field values.

The L5 command is the same as the L2 command except that the record read is placed in hold status. This prevents other users from updating the record until it is released. The multifetch/prefetch and response code options (see the L1/L4 command description, above) also apply to the L2/L5 commands.

L3/L6

The L3 command reads records from Data Storage in the logical sequence of a user-specified descriptor. You specify the file to be read, the descriptor to be used for sequence control, and the fields for which values are to be returned. Adabas returns the requested field values.

Command Option 2 for the L3/L6 command specifies whether the records are read in ascending or descending order. In addition, Command Option 2 can be used to specify a start value.

The L6 command is the same as the L3 command except that the record read is placed in hold status. This prevents other users from updating the record until it is released.

In addition to the multifetch/prefetch and response code options (see the L1/L4 command description, above), the start value option permits reading to begin at a user-specified value or ISN.

Associator Read Commands (L9, LF)

The L9 and LF commands read information directly from the Associator inverted lists or field definition tables (FDTs), returning either the inverted list values for a specified descriptor or the field definitions for a specified file in the database.

This section covers the following associator read commands:

- L9
- LF

L9

The L9 command returns each value contained in the inverted list for a given descriptor, and the number of records in which the value is contained.

You specify the file and descriptor for which values are to be returned, the value at which the command is to begin, and whether the values are returned in ascending or descending sequence.

LF

The LF command returns the field definitions for a file. You specify the file for which the field definitions are to be returned.

The field definitions for all the fields in the file are returned. Each field definition consists of the field name, level number, standard format, standard length, and definition options.

Database Modification Commands (A1, E1, N1/N2)

Database modification commands (A1, E1, and N1/N2) add, change, or delete database records and update the related Associator lists accordingly. You can assign ISNs to new records or they can be assigned by Adabas.

This section covers the following database modification commands:

- A1
- E1
- N1/N2

A1

The A1 command updates the contents of one or more fields within a record. You specify the file and ISN of the record to be updated, together with the fields to be updated and the values to be used for updating.

Adabas performs all necessary modifications to the Associator and Data Storage. Associator updating is required only if one or more descriptors are updated.

A hold option is available for the purpose of placing the record to be updated in hold status prior to the update.

E1

The E1 command deletes a record or refreshes a file. You specify the file and ISN of the record to be deleted, or specify the file only (without an ISN and command ID) to refresh the file.

Adabas performs all necessary modifications to the Associator and Data Storage.

A hold option is available for the purpose of placing the record to be deleted in hold status prior to the deletion.

N1/N2

The N1/N2 command adds a new record to a file. You specify the file to which the record is to be added together with the fields and field values to be used. Adabas performs all necessary modifications to the Associator and Data Storage.

If the N1 command is used, the ISN for the new record is assigned by Adabas. If N2 is used, you must provide the ISN.

Logical Transaction Control Commands (ET/BT)

An Adabas logical transaction defines the logical start (BT) and end (ET) of the database operation being performed. If the user operation or Adabas itself terminates abnormally, these commands make it possible to restart a user, beginning with the last unsuccessfully processed transaction. ET/BT commands define the transaction start and end, restore pretransaction conditions if a situation occurs that prevents successful completion of the transaction, and read program-specified user data written during the transaction sequence.

Programs that use these commands are called *ET logic programs*. Although not required, Software AG recommends that you use ET logic.

This section covers the following logical transaction control commands:

- BT
- ET

BT

The BT command backs out the current transaction being processed.

All modifications resulting from updates performed during the transaction are removed, and all records placed in hold status during the transaction are released (unless kept in hold status by the multifetch option; see the section *Multifetch Operation Processing*).

ET

The ET command indicates the end of a logical transaction.

An ET command causes Adabas to physically store all data protection information related to the transaction. This information is used to apply all the updates performed during the transaction at the start of the next Adabas session if the current session is terminated before these updates are physically applied to the database.

The ET command releases all the records that have been placed in hold status during the transaction (unless kept in hold status by the multifetch option; see the section *Multifetch Operation Processing*). The ET command may also be used to store user data in an Adabas system file. This data may be retrieved with an OP or RE command.

Special Commands

Special commands perform many of the housekeeping functions required for maintaining the Adabas database environment. Commands in this group allow you to perform the following functions:

- Open and close a session (but not to control a transaction).
- Write data protection information and checkpoints.
- Set and release record hold status.

This section covers the following special commands:

- CL
- C1
- C3
- C5
- HI
- OP
- RC
- RE
- RI

CL

The CL command terminates a user session, releasing all records held for that user.

- It physically writes all current data protection information to the data protection log.
- It releases all records currently in hold status for the user.
- It releases all the command IDs and corresponding ISN lists currently assigned the user.
- It stores user data in an Adabas system file (optional).

C1

The C1 command causes a checkpoint to be taken.

The C1 command physically writes all current data protection information to the data protection log, and writes a checkpoint entry to the data protection log and the system checkpoint file. This checkpoint entry may be used as a reference point for subsequent removal or reapplication of updates. An option allows the C1 command to initiate a buffer flush.

C3

The C3 command, issued only by exclusive control and update users (who are not using ET logic), writes a SYNX-03 checkpoint in the Adabas checkpoint file.

- The checkpoint contains the current data protection log and block number.
- The checkpoint may be used to restore the database (or certain files) to the status in effect at the time the checkpoint was taken. This may be necessary before a program performing exclusive control updating can be rerun or restarted.

If Command Option 2 is specified, the C3 command also stores user data in the Adabas checkpoint file for restart purposes. The stored data may be subsequently read with an OP or RE command.

C5

The C5 command writes user data to the Adabas data protection log.

The data can be read subsequently using the ADASEL utility.

HI

The HI command places a record in hold status. Specify the file and ISN of the record to be placed in hold status.

A record placed in hold status cannot be updated by another user until it is released.

OP

An OP command is mandatory when any of the following apply:

- The nucleus is run with ADARUN parameter OPENRQ=YES
- Exclusive file control (EXF) is to be performed
- User data that was stored in an Adabas system file by a previous ET command is to be read
- User data is to be stored in an Adabas system file, using a C3, CL, or ET command
- You are assigned a special processing priority
- You are an access-only user (no update commands permitted).
- A transaction time limit or a non-activity time limit is to be set for you that differs from that specified by ADARUN parameters TT or TNAX, respectively. Your setting must conform to the maximum setting set by the ADARUN parameters MXTT and MXTNA, respectively.
- Special data encoding or architecture is to be specified for your user session.

It may also be used to set the maximum number of records that you can place in hold status at the any given time, and the maximum number of command IDs you may have active at the same time.

RC

The RC command may be used to release one or more command IDs currently assigned to you, or to delete one or all global format IDs.

RE

The RE command reads user data previously stored in an Adabas system file by CL or ET commands.

RI

The RI command releases a record from hold status.

Specify the file and ISN of the record to be released. You may also request that all records currently held by you are released. The RI command should be issued by non-ET users only.

Transactions and ET Logic

This section describes the concept of a *transaction* and an *ET logic user*; it explains ET logic operations including normal and abnormal transaction termination processing and the storage and retrieval of user (ET) data.

This section covers the following topics:

- What Is A Logical Transaction?
- Transaction Sequence Number
- ET Transaction Time Limit

- Back Out Transaction (BT) Command
- Autobackout
- End Transaction (ET) Command
- User (ET) Data
- Adabas User ID

What Is A Logical Transaction?

A *logical transaction* is the smallest unit of work (as defined by the user) that must be performed in its entirety to ensure the logical consistency of the information in the database. Users who use logical transaction commands are referred to as ET logic users.

A logical transaction comprises one or more Adabas commands that read or update the database as required to complete a logical unit of work. A logical transaction begins with the first command that places a record in hold status and ends when an ET, BT, CL, or OP command for the same user is issued.

The RE (read ET data) command can be used to retrieve user restart data stored by the ET or CL command.

Transaction Sequence Number

When a program issues an ET or CL command, Adabas returns a transaction sequence number in the command ID field of the ET or CL command's control block. The transaction sequence number is a count of the total number of ET commands issued thus far during the user session.

The transaction sequence number is set to 1 for the first ET command issued by the user. The first ET command following the OP command returns transaction sequence number 2 in the Command ID field or the Adabas control block. Each subsequent OP command returns the transaction sequence number of the last ET command issued by that same user.

ET Transaction Time Limit

Adabas provides a transaction time limit for programs that use ET logic. The time measurement for a transaction begins when the first command places a record in hold status, and ends when the program issues an ET, BT, or CL command.

The time limit is set with the ADARUN TT parameter; a transaction time limit that overrides this general limit for a specific user can be set with the OP command; this limit is controlled by the ADARUN MXTT parameter.

If a transaction exceeds the prescribed limit, Adabas generates a BT (back out transaction) command. The BT command removes all the updates performed during the transaction and releases all held records.

Adabas returns response code 9 (ADARSP009) for the next command issued by the user indicating that the current transaction has been backed out. The user can either repeat the backed out transaction from the beginning or begin another transaction.

Back Out Transaction (BT) Command

The BT command removes all updates made during the transaction currently being processed. This may be necessary because of a program error, a timeout, or when Adabas determines that the transaction cannot be completed successfully.

A BT command also performs an implied ET command, which releases all the records held during the transaction unless otherwise specified by the multifetch option; for more information, see the section *Multifetch Operation Processing*.

For example, the command sequence

```
FIND (S4)
  UPDATE (A1) (modify field XX to value 20)
  FIND (S4)
  UPDATE (A1) (modify field YY to value 50)
  END TRANSACTION (ET)
  FIND (S4)
  UPDATE (A1) (modify field XX to value 10)
  BACKOUT TRANSACTION (BT)
```

results in the field values of XX = 20 and YY = 50. The second update to field XX is removed by the BT command.

Autobackout

Autobackout, which is performed only for ET logic users, backs out transactions automatically in sessions that end abnormally. Adabas performs an internal BT (back out transaction) followed by autorestart, and then returns response code 9 (ADARSP009) to indicate that the last transaction has been backed out.

Autobackout occurs at the end of any nucleus session that was terminated with HALT.

Autobackout also occurs at the beginning of the next Adabas session to remove any updates that were performed in partially completed transactions by ET logic users during the previous terminated session.

When response code 9 (ADARSP009) indicates that the transaction was backed out, the user has the option of either reissuing the transaction from the start or beginning another transaction.

To restart the backed-out transaction, a terminal operator may need to reenter the data for the transaction, or an internal restart may have to be performed from the beginning of the "update phase" of the transaction; the "update phase" of a transaction begins with the first command that places a record in hold status.

End Transaction (ET) Command

The ET command must be issued at the end of each logical transaction. Successful execution of an ET command ensures that all the updates performed during the transaction will be physically applied to the database regardless of subsequent user or Adabas session interruption.

Updates performed within transactions for which ET commands have not been successfully executed will be backed out by the autobackout routine (see *Autobackout*).

Unless otherwise specified by the multifetch option, the ET command releases all records held by the user during the transaction. Adabas returns a unique transaction sequence number (see *Transaction Sequence Number*) that can be used to identify the transaction for auditing or restart purposes.

The ET command may also be used to store user data in an Adabas system file. This data may be used for user restart purposes, and may be read with an OP or RE command.

User Program	Adabas
FIND (S4), UPDATE (A4)	record updated in Adabas buffer but not necessarily written to the database
FIND (S1), HOLD ISN (HI), UPDATE (A4)	record updated in Adabas buffer but not necessarily written to the database
END TRANSACTION (ET)	data protection information for the transaction is written to the Adabas Work and data protection log, ending the transaction
FIND (S4), UPDATE (A4)	record updated in Adabas buffer but not necessarily written to the database
FIND (S1), HOLD ISN (HI), UPDATE (A4)	record updated in Adabas buffer but not necessarily written to the database
. . . Adabas or user session interruption . . .	

When the next Adabas session is started, or when the user is timed out, both updates for transaction 1 are physically written to the database (if they were not previously written). The updates for transaction 2 are not physically written (or will be backed out) because no ET command was processed for this transaction.

User (ET) Data

User data (ET data) may be stored with an ET or CL command and read with an OP or RE (read ET data) command.

One record of user data is kept for each user ID. The data is maintained until the next ET or CL command is issued in which user data is provided.

Each user data record is also written to the Adabas data protection log with each checkpoint written by the transaction. This data may be subsequently read with the ADASEL utility.

User data is primarily used to perform the following functions:

- Store data needed to restart an operation (e.g., input message data, transaction identification data, transaction summary data).
- Store intermediate data to be used by subsequent transactions (e.g., for audit trail purposes).
- Communicate with other users. The data stored may be read by other users provided that the ID of the user who stored the data is supplied in the OP command.

Software AG also suggests using the user data facility to perform the following functions:

- Establish an installation standard for the user data record format and store data for each update transaction with the ET command.
- Read the user data and display it in a standard message format when the user logs on to an application. The user is thus informed of the last successful updating activity which corresponds to his user ID.

Software AG recommends that you keep ET commands that provide user data separate from those that do not. Combining the two types of ET command may cause significant additional overhead by forcing Adabas to repeatedly copy user data from the Adabas Work data set (where it is temporarily stored) to the protection log.

Adabas User ID

The user ID is an identifier used by Adabas to store and retrieve user (ET) data and to assign a special processing priority to a user. The user ID is specified in the Additions 1 field of the OP command.

A user ID provided by the user must meet both of the following criteria:

- It must begin with the character A (X'C1') through 9 (X'F9')
- It must be unique to ensure that the user (ET) data is related to the user regardless of the terminal used. User (ET) data is maintained until the user's next ET or CL command in which user (ET) data is provided.

If the user either does not provide a user ID or provides an invalid user ID, Adabas establishes a default user ID for the user session, and any user (ET) data stored by the user is deleted when the current session is terminated.

To avoid later limitations, Software AG recommends that you always specify a user ID.

Competitive Updating

Competitive updating is in effect when two or more users (in multiuser mode) are updating the same Adabas file(s).

This section describes the Adabas facilities used to ensure data integrity in a competitive updating environment. These include record hold and release processing and the avoidance of resource deadlock, exclusive control updating, and shared hold updating.

This section covers the following topics:

- ET User Record Hold and Release
- Avoiding Resource Deadlock
- Shared Hold Status

- Exclusive Control Updating

ET User Record Hold and Release

The record hold facility allows the ET user to place a record in hold status for updating without allowing interim updating of the record by another user. Adabas *holds* a record by placing the ISN of the record in the hold queue; as a result, record hold is also called an *ISN hold*.

Record hold is applicable for all ET logic users; Adabas does not place an ISN in hold status for EXU (exclusive control updating) users. Read *Exclusive Control Updating*, for more information.

This section covers the following topics:

- Record Hold Commands
- Record Update Using the Hold Option
- Record Release

Record Hold Commands

A record is held when you use the find-with-hold command (S4), the read-with-hold commands (L4, L5, L6), an A1 or E1 command in which the hold option is specified, or a hold record (HI) command. An N1/N2 command issued by an ET logic user also places the added record in hold status.

The successful completion of any of these commands places the record (ISN) in hold status. If the record is already being held by another user, the user issuing the record hold command is placed in a wait status until the record becomes available, at which time Adabas reactivates the command.

If the R (return code) or O (multifetch/return code) option is used with any of the record hold commands and the record to be held is already being held by another user, Adabas returns response code 145 (ADARSP145) instead of placing the user in a wait status.

If you issue a find (S1) or read (L1, L2, L3) command, you can access the record regardless of the fact that the record is in hold status for another user.

Record Update Using the Hold Option

You can update or delete any records they have in hold status by issuing an A1 or E1 command.

An A1 command is executed only if the record is either in hold status for the requesting user, or free from hold status and the you specify that the record should be held. If the record is currently held by another user, your hold request is either placed in wait status or, if you request, you will receive response code 145 (ADARSP145). If the record is not in hold status, response code 144 (ADARSP144) is returned, indicating that the record was placed in hold status.

If an E1 command (without the hold option) is issued for a record that is not in hold status for you, Adabas places the record in hold status for you provided that the record is not in hold status for another user. If you do not place a record to be deleted in hold status, there is no guarantee that the record will not be updated or deleted by another user before the E1 command is executed.

If an N1/N2 command is issued and there is no available space in the hold queue, response code 145 (ADARSP145) is returned.

Record Release

An ET logic user releases records from hold status with the ET command following each logical transaction, and with a close (CL) command at the end of the Adabas session. Programs using ET logic should not release records with the release record (RI) command if any updating has been performed during the current transaction, since this could result in a loss of data integrity.

For example, a record is updated and then released with an RI command by ET user 1, and the transaction continues. In the meantime, the same record is updated by user 2, who then ends the session with an ET command. If user 1's transaction is subsequently backed out due to an error or transaction timeout, the updates performed by both users 1 and 2 are removed even though user 2's transaction completed successfully with an ET command.

Therefore, user programs that employ ET logic should only release records at the completion of a logical transaction with the ET command. Non-ET logic users should use A1 or E1 commands to update or delete records and then release them.

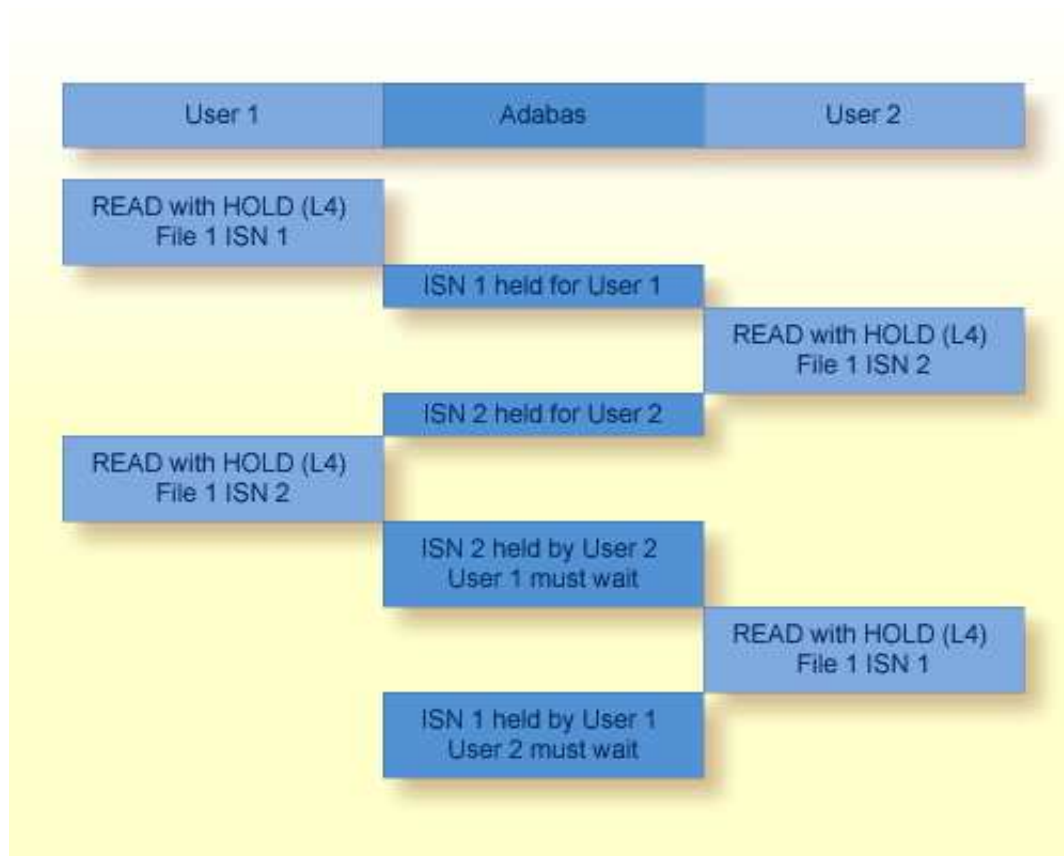
The multifetch option allows records (and their ISNs) to be released from hold status selectively if a non-zero count and the file number/ISN is specified in the ISN buffer when the ET (or BT) command is issued. See the section *Multifetch Operation Processing*.

The CL command releases all records in hold status for the issuing user, whether an ET user or not.

Avoiding Resource Deadlock

A resource deadlock can occur if two users are placed in wait status because each had requested a record that was currently in hold status for the other user.

For example:



Resource Deadlock Example

Adabas protects against such a user deadlock situation by detecting the potential deadlock and returning a response code (ADARSP145) to User 2 after putting the first user in wait status. This occurs if the two users are serviced by the same nucleus (a non-cluster nucleus or the same cluster nucleus). But even in a single nucleus, a deadlock might not be detected if an ISN involved in the deadlock is being held as a shared resource by more than one user. For more information, read about shared hold status.

Putting more records in hold status (including shared hold status) may decrease the possible parallelism of transaction processing (and thus, the performance of multiuser applications) and increase the likelihood of deadlocks between transactions (where each transaction holds a record that the other wants).

Shared Hold Status

Shared hold status allows you to lock data records in shared mode, rather than in exclusive mode. This allows your database users to read the same record in parallel transactions, but ensures that no one can update the record concurrently.

Using shared hold status, your users can protect large object values from concurrent updates without locking out other users who may need to read the same LOB value or other LOB values in the same record. It also allows your users to protect the records they read against concurrent updates for specific periods of time:

- For the duration of the read command;

- When the next record in a sequence is read;
- When the user's transaction ends;
- Indefinitely.

Shared hold updating is controlled by four command options for the BT, ET, HI, L4 - L6, RI, and S4 commands, although all four command options do not necessarily apply to all of these commands. These command options are specified in the command option 3 field of ACBX only calls for the associated command. Each command option specifies a different shared hold time period.

- Option C puts the record in shared hold status for the duration of the read operation. It ensures that the version of the record being read has been committed by the last updater. This option is available for the L4, L5, L6 and S4 commands.
- Option Q puts the record in shared hold status until the next record in the read sequence is read or the read sequence or transaction is terminated, whichever happens first. It ensures that the record being read cannot be updated concurrently until the next record in the sequence is read (or the transaction is terminated). This option is available for the L4 (when command option 2 is set to "N"), L5, L6 and S4 commands.

For the S4 command, a command ID must be given and the ISN buffer length must be set to 4. The record returned by the S4 remains in shared hold status until the next record is retrieved by a subsequent S4 or L4/N command with the same command ID.

- Option S puts the record in shared hold status until the end of the transaction. It ensures that the record being read cannot be updated concurrently until the transaction is terminated. This option is available for the HI, L4, L5, L6, RI and S4 commands.

When specified in an HI request, the record is placed in shared hold status for the user. If a second user also issues an HI request, the record is put in shared hold status for both users.

When specified in an RI request, if the record is in exclusive hold status and has not been updated in the current transaction, it is placed in shared hold status; if it is in exclusive hold status and has been updated, it is not placed in shared hold status and the exclusive hold remains in effect.

- Option H keeps a record in shared hold status indefinitely (until the next ET or BT command). This option is available only for the BT and ET commands. Records in shared hold status at the time of the BT or ET command are kept in shared hold status beyond the end of the transaction until another ET or BT command is issued (without this H option or the prefetch or multifetch options). Any records in exclusive control are also changed to shared hold status beyond the end of the transaction.

You cannot use option H if the multifetch or prefetch options are used (or, in Adabas on open systems, if all resources owned by the user are to be released via a command option 1 "T" setting).

If the same record is placed in shared hold status more than once (using the C or S options or the Q option for different read sequences), it stays in shared hold status until all of the specified hold lifetimes have expired.

Putting more records in hold status (including shared hold status) may decrease the possible parallelism of transaction processing (and thus, the performance of multiuser applications) and increase the likelihood of deadlocks between transactions (where each transaction holds a record that the other wants).

Using shared hold status affects the ADARUN NH and NISNHQ parameter settings. Each shared hold request with a different command ID (CID), as well as a (shared or exclusive) hold request without a CID, is counted against the NISNHQ and NH limits. This affects application programs that make use of the new Q option for sequential reads. For example, if a program reads records with the Q option and then updates every record, the shared hold operations from the use of the Q option and the exclusive hold operations from the update commands are counted separately. Such a program might need NISNHQ and NH limits set two times larger than when the Q option is not specified.

For more information about these commands and command options, read about the individual commands in *Commands*. Detailed description of shared hold status processing and rules are provided in this section.

This section covers the following topics:

- Duration (Lifetimes) of Shared Hold Status
- Rules
- Processing

Duration (Lifetimes) of Shared Hold Status

The following table summarizes how long records remain in shared hold status, depending on which commands are used to place them in and release them from hold status. The key to the codes used in the table are given below the table. Each cell in the table describes the hold status after the command in the top row was used to place the record in hold status and the command in the leftmost column was used to release the record from hold status. If the same record is put in hold status more than once in different ways, it remains in hold status until it has been released from hold status for each way.

Note:

In this table, the notation *xx/y* indicates the command (*xx*) and the command option (*y*).

Command used to....													
Release record from hold status...	Place record in hold status...												
	BT/H	ET/H	Lx	Lx/C	Lx/Q	Lx/S	HI	HI/S	RI/S	S4	S4/C	S4/S	S4/Q
BT	R	R	R	---	R	R	R	R	R	R	---	R	R
BT/H	S	S	S	---	S	S	S	S	S	S	---	S	S
BT/M	R ^M	R ^M	R ^M	---	R ^M	R ^M	R ^M	R ^M	R ^M	R ^M	---	R ^M	R ^M
BT/P ¹	R ^P	R ^P	R ^P	---	R ^P	R ^P	R ^P	R ^P	R ^P	R ^P	---	R ^P	R ^P
BT/S ²	R ^S	R ^S	R ^S	---	R ^S	R ^S	R ^S	R ^S	R ^S	R ^S	---	R ^S	R ^S
ET	R	R	R	---	R	R	R	R	R	R	---	R	R
ET/H	S	S	S	---	S	S	S	S	S	S	---	S	S
ET/M	R ^M	R ^M	R ^M	---	R ^M	R ^M	R ^M	R ^M	R ^M	R ^M	---	R ^M	R ^M
ET/P ¹	R ^P	R ^P	R ^P	---	R ^P	R ^P	R ^P	R ^P	R ^P	R ^P	---	R ^P	R ^P
ET/S ²	N	N	N	---	N	N	N	N	N	N	---	N	N
Lx/C	S	S	X	S&R	S	S	X	S	S	X	S&R	S	S
Lx/Q	S	S	X	---	R&S	S	X	S	S	X	---	S	R&S
RC	S	S	X	---	R	S	X	S	S	X	---	S	R
RI	R	R	R ^U	---	R	R	R ^U	R	R	R ^U	---	R	R
RI/S	S	S	S ^U	---	S	S	S ^U	S	S	S ^U	---	S	S
S4/C	S	S	X	S&R	S	S	X	S	S	X	S&R	S	S
S4/Q	S	S	X	---	R&S	S	X	S	S	X	---	S	R&S

Key:

Code	Description
---	The record was not in hold status at the end of the first command.
N	The hold status of the record has not been changed.
R	The record has been released from hold status.
R ^M	The record has been released from hold status if it was specified in the ISN buffer; otherwise, it is kept in its current (shared or exclusive) hold status.
R ^P	The record has been released from hold status if it was <i>not</i> specified in the ISN buffer; otherwise, it is kept in its current (shared or exclusive) hold status.
R ^S	The record has been released or downgraded to the hold status it had at the time of the specified target savepoint. Caution: The record has <i>not</i> been put in hold status or upgraded to the hold status it had at the time of the target savepoint.
R ^U	The record has been released from hold status; however, if it was updated earlier in the same transaction, it is kept in exclusive hold status.
R&S	The record is released from shared hold status and the next record in the read sequence is placed in shared hold status.
S	The record is in shared hold status.
S ^U	The record has been downgraded to shared hold status; however, if it was updated earlier in the same transaction, it is kept in exclusive hold status.
S&R	The record is placed in and released from shared hold status in the course of the same command.
X	The record is in exclusive hold status.

Notes:

1. The prefetch option (P) for ET and BT commands is available only for Adabas on mainframe systems.
2. The savepoint option (S) for ET and BT commands is planned or available in Adabas 6.2 on open systems.

Corollaries:

1. A command that uses the C option has no impact on and is not impacted by the hold status operations of other commands from the same user.
2. A command that uses the Q option, as well as an RC command, may have an impact only on the one record from the read sequence that was previously put in shared hold status using the Q option.

3. An updated record is never released from exclusive hold status before the update has been committed or backed out.
4. An ET or BT command without the H, M, P, or S options always releases all records from hold status, whether they are shared or exclusive.
5. An ET command with the S option has no effect on the hold status conditions of records.
6. A BT command with the S option only releases or downgrades records from hold status. It does not reverse the effects of RI commands. If the application does not use RI commands, a BT command with the S option reinstates the hold status of all records held by the user at the time of the target savepoint.

Rules

The following rules apply to shared hold status and the associated C, Q, S, and H command options:

1. These new command options can only be issued in ACBX calls in the command option 3 field; they are not valid in ACB calls.
2. Only ET users can put records in shared hold status. Access-only (ACC) users and exclusive-update/exclusive-file (EXU/EXF) users who are not also ET users cannot put records in shared hold status. Users with EXU/EXF control on a file cannot run in parallel with users who put records of that file in shared hold status (which must be ET users).
3. Multiple users can have the same record in shared hold status at the same time, whereas only one user at a time can have a record in exclusive hold status.
4. You can, in the course of several commands, put the same record in shared or exclusive hold status more than once. The record is kept in hold status throughout the longest lifetime (duration) of any of the hold status requests.
5. Use of the Q option requires that a command ID be given and, for S4 commands, that the ISN buffer length be set to 4. The Q option is not allowed if the read is in ISN sequence (L4/I) or if the read sequence is performed with multifetch or prefetch (command options M, O, or P).
6. Command options C, Q, and S are not allowed together with the prefetch command option (P).
7. A request to keep held records in shared hold status via the H command option on an ET or BT command applies to all records that were ever put in hold status during the transaction and that have not been released since.

Processing

This section describes the processing behaviors that apply to shared hold status and the associated C, Q, S, and H command options in different situations.

1. A request to put a record in hold status cannot go forward if:
 1. one user has the record in exclusive hold status and another user requests putting it in exclusive hold status, too (existing logic);

2. one user has the record in exclusive hold status and another user requests putting it in shared hold status; or
 3. one or more users have the record in shared hold status and one of them, or another user, requests putting it in exclusive hold status.
2. If a request to put a record in hold status cannot go forward, the following processing occurs, depending on other command option settings:
1. If the return option has been specified (command option 1 = "R"), the command is returned immediately with response code 145 (ADARSP145).
 2. If the multifetch return option has been specified (command option 1 = "O"), the records already read and put in hold status by this command are returned. If no record has already been read and put in hold status, the command is returned with response code 145 (ADARSP145).
 3. If the multifetch option has been specified (command option 1 = "M"), the records already read and put in hold status by this command are returned. If no record has already been read and put in hold status, the command is put in wait status until all other users have released the record from hold status, unless putting the command in wait status will cause a deadlock between users (read point e in this list for more information about deadlock handling).
 4. If none of these command options has been specified, the command is put in wait status until all other users have released the record from hold status, unless putting the command in wait status will cause a deadlock between users (read point e in this list for more information about deadlock handling).
 5. The command is not put in wait status if Adabas detects that this new wait condition would create a deadlock between users who are holding records and waiting to put another one in hold status.

A deadlock involving multiple users who are holding the same record shared, or (with Adabas Cluster Services or Adabas Parallel Services) who are serviced by different nuclei in a cluster, may not be detected. In this case, the hold commands involved are delayed until one of the waiting users incurs a transaction timeout (TT parameter).

3. While a command is waiting for other users to release a record from shared hold status so that it can put that record in exclusive hold status, further requests by additional users to also put the record in shared hold status will not go forward either.

A request to upgrade a record from shared to exclusive hold status can go forward in the following scenario:

- Only one user has the record in shared hold status.
- Another user is waiting to put the record in exclusive hold status, without first getting it in shared hold status.
- The first user requests to upgrade the hold status from shared to exclusive. This upgrade request is granted, even though another user is already waiting, because it does not increase the set of users the second user is waiting for.

4. A record in shared hold status can be updated without specifying command option 3 as "H" in the A1 command. If an A1 or E1 command specifies a record that the user has put in shared hold status, the hold status is upgraded to exclusive prior to the update or deletion of the record. (The same applies to an N1 or N2 command if the user has already put the record in shared hold status via a previous HI command.)

Similarly, a record in shared hold status is upgraded to exclusive if it is read in an L4, L5, L6, or S4 command without the command option 3 settings "S", "Q", and "C", or if it is specified in an HI command without the command option 3 setting of "S".

5. If a request to upgrade a record from shared to exclusive hold status cannot go forward, the following processing occurs, depending on other command option settings:
 1. If the return option has been specified (command option 1 = "R"), the command is returned immediately with response code 145 (ADARSP145).
 2. If the multifetch return option has been specified (command option 1 = "O"), the records already read and put in hold status by this command are returned. If no record has already been read and put in hold status, the command is returned with response code 145 (ADARSP145).
 3. If the multifetch or prefetch option has been specified (command option 1 = "M" or "P"), the records already read and put in hold status by this command are returned. If no record has already been read and put in hold status by this command, the command is put in wait status until all other users have released the record from shared hold status, unless the command cannot be put into wait status because another user is already waiting to upgrade the same record from shared to exclusive hold status (read item point e in this list for more information).
 4. If none of these command options has been specified, the command is put in wait status until all other users have released the record from shared hold status, unless the command cannot be put into wait status because another user is already waiting to upgrade the same record from shared to exclusive hold status (read item point e in this list for more information)..
 5. The command is not put in wait status if another user is already waiting to upgrade the same record from shared to exclusive hold status. In this case, the new command is returned with response code 145 (ADARSP145) with a new subcode, even though the return option has not been specified.
6. If command option 3 setting "C" is specified for an L4, L5, L6, or S4 command:
 1. The target record is put in shared hold status (unless it was already in hold status for the user prior to the command).
 2. The record is read from the database.
 3. The record is released from shared hold status, unless it had already been in hold status for the user prior to the command with 'C' option.
 4. The record read from the database is returned to the user.

This procedure ensures that the returned data will not be the result of an uncommitted update by another user that may still be backed out.

7. If command option 3 setting "Q" is specified for an L4/N, L5, L6, or S4 command:
 1. The target record is put in shared hold status (unless it was already in hold status for the user prior to the command).
 2. The record is read from the database.
 3. The record read from the database is returned to the user.
 4. Unless it is specified in another command that the record be kept in hold status longer, the record is released from shared hold status when one of the following occurs:
 - the next read command in the sequence is issued, or
 - the read sequence is terminated (by releasing the associated command ID).
 5. Furthermore, the record is released from shared hold status when:
 - the user's transaction ends (e.g., via an ET command) without specifying that the record be kept in hold status (e.g., via the option H), or
 - it is explicitly released from hold status (via an RI command).
 6. If the record is updated, it remains in exclusive hold status until the end of the transaction.

This procedure ensures that the returned record will not be changed by another user before the user reads the next record in the read sequence or terminates the read sequence or the current transaction.

8. If command option 3 setting "S" is specified for an L4, L5, L6, or S4 command:
 1. The target record is put in shared hold status (unless it was already in hold status for the user prior to the command).
 2. The record is read from the database.
 3. The record read from the database is returned to the user.
 4. The record is released from shared hold status when:
 - the user's transaction ends (e.g., via an ET command), or
 - it is explicitly released from hold status (via an RI command).
 5. If the record is updated, it remains in exclusive hold status until the end of the transaction.

This procedure ensures that the returned record will not be changed by another user before the user ends the current transaction. If the user reads the record again directly (via its ISN) prior to transaction end, the record will still exist and be unchanged. If the user attempts to read the record again using the same selection criterion (e.g., in an L4/I, L5, L6, or S4 command), the command might possibly return a different record, which did not satisfy the selection criterion earlier but does satisfy it now.

9. If command option 3 setting S is specified for an HI command:
 1. The target record is put in shared hold status (unless it was already in hold status for the user prior to the command).
 2. The record is released from shared hold status when:
 - the user's transaction ends (e.g., via an ET command), or
 - it is explicitly released from hold status (via an RI command).
 3. If the record is updated, it remains in exclusive hold status until the end of the transaction.

This procedure ensures that the specified record will not be changed by another user before the user ends the current transaction.

10. By default, all shared and exclusive hold status conditions of records being held by a user are released when the user's current transaction ends. In particular, they are released when the user issues a BT or ET command without command option 1 set to "M", "P" or "O", or command option 3 is set to "H"; or when Adabas decides for internal reasons to back out the transaction.

For a BT or ET command with multifetch option (command option 1 = "M"), only the shared and exclusive hold status conditions for the records specified in the multifetch buffer are released; all others are kept as is.

For a BT or ET command with prefetch option (command option 1 = "P"), only the shared and exclusive hold status conditions for the records not specified in the ISN buffer are released. The current (shared or exclusive) hold status conditions for the records specified in the ISN buffer are kept as is. Any additional records specified in the ISN buffer are put in exclusive hold status (with response code 2 (ADARSP002) issued if that is not immediately possible).

11. If command option 3 setting "H" is specified for a BT or ET command, Adabas keeps in shared hold status all records that were put in (or downgraded to) shared hold status earlier in the transaction (or kept in shared hold status at the end of the preceding transaction) and not released since then. Furthermore, Adabas downgrades all records to shared hold status that were put in (or upgraded to) exclusive hold status earlier in the transaction (or kept in exclusive hold status at the end of the preceding transaction) and not released since then.

When combined with command option 3 setting "Q" for the L4/N, L5, L6 and S4 commands, the "H" option can be used to perform multiple transactions in the course of one read sequence, without losing, at transaction end, the protection of the current record in the sequence against concurrent updates by other users.

When combined with command option 3 setting "S" for the HI, L4-L6 and S4 commands, the H option can be used to perform multiple transactions in the course of one multi-step, user-defined operation, without losing, at transaction end, the protection of the held records against concurrent updates by other users.

12. If command option 3 setting "S" is specified for an RI command, the target record is downgraded from exclusive to shared hold status, provided it is in exclusive hold status for the user and has not been updated in the current transaction. If the target record is not in exclusive hold status for the user, the RI command has no effect. If the record has been updated in the current transaction, response code 113 (ADARSP113) is returned (if ISN=0 was specified, response code 2, ADARSP002, is

returned).

For an RI command without command option "S", the target record is released from shared or exclusive hold status, provided it is in hold status for the user and has not been updated in the current transaction. If the target record is not in hold status for the user, the RI command has no effect. If the record has been updated in the current transaction, response code 113 (ADARSP113) is returned (if ISN=0 was specified, response code 2, ADARSP002, is returned).

13. If you put the same record in shared or exclusive hold status more than once, the record is kept in hold status throughout the longest lifetime of any of the hold status requests. In particular:
 1. If a record has been updated, it stays in exclusive hold status until the end of the transaction. If the end of the transaction is triggered by a BT or ET command with the "M" (multifetch) or "P" (prefetch) options, you can retain the record in exclusive hold status beyond the end of the transaction. If the end of the transaction is triggered by a BT or ET command with the "H" option, the record is downgraded to and retained in shared hold status beyond the end of the transaction.
 2. If a record has been put in exclusive hold status using an HI, L4-L6 or S4 command and has not been updated, it stays in exclusive hold status until the end of the transaction or until its release via an RI command, whichever occurs first. If the end of the transaction is triggered by a BT or ET command with the "M" (multifetch) or "P" (prefetch) options, you can retain the record in exclusive hold status beyond the end of the transaction. If the end of the transaction is triggered by a BT or ET command with the "H" option, the record is downgraded to and retained in shared hold status beyond the end of the transaction.
 3. If a record has been put in shared hold status using an HI, L4-L6 or S4 command with the "S" option, it stays in shared hold status at least until the end of the transaction or until its release via an RI command, whichever occurs first. If the end of the transaction is triggered by a BT or ET command with the "H" or "M" or "P" options, you can retain the record in shared hold status beyond the end of the transaction.
 4. If a record has been put in shared hold status using an L4/N, L5, L6 or S4 command with the "Q" option, it stays in shared hold status at least until the next read command in the read sequence, the termination of the read sequence (e.g., via an RC command), the end of the transaction, or until its release via an RI command, whichever occurs first.

If the end of the transaction is triggered by a BT or ET command with "H" or "M" or "P" option, the user can retain the record in shared hold status beyond the end of the transaction.

5. If a record has been put in shared hold status using an L4-L6 or S4 command with the "C" option, it stays in shared hold status only until the end of the command, unless it had already been in hold status prior to the command.

A record remains in exclusive hold status until no reason exists anymore to keep it in this status according to items a or b in the list above. A record remains in shared hold status until no reason exists anymore to keep it in this status, according to rules c, d or e.

14. If you end a transaction using a BT or ET command without the "H", "M" or "P" options, all records held by your processing are released from their shared or exclusive hold status at that time.

If you end a transaction using a BT or ET command with the "H" option, all records that were put in shared hold status earlier (using the "Q" or "S" option) and not released since then are kept in shared hold status for the next transaction. Records that were put in exclusive hold status and not released since then are downgraded to shared hold status.

15. When you put a record in either shared or exclusive hold status, a new transaction starts for you, if one is not already active. The transaction is subject to the usual transaction timeout logic.

If you keep records in shared or exclusive hold at the end of a transaction (via command option "H", "M", or "P" on the BT or ET command), a new transaction starts immediately.

Caution:

If an application uses the command option "H" too liberally, one user can prevent a second user from putting a record in exclusive hold status indefinitely, across multiple transactions of the first user, until the transaction of the second user incurs a timeout and is backed out.

If an RI command releases the only record in (shared or exclusive) hold status for the user, the user's transaction ends. (Note that the record cannot have been updated).

Exclusive Control Updating

You may request exclusive control of one or more Adabas files to prevent other users from updating the file during the execution of your user session. Adabas exclusive file control occurs in one of the following ways:

- **UTI** control is required by long-running, external utilities (such as the ADALOD utility) as well as by short-running utility functions that are executed by the nucleus (for example, using the Adabas Online System or ADADBS functions). It is not available to your application programs. UTI file control is incompatible with any other use of the file; other users may not access or update the file while it is under UTI file control.
- **EXF** control can be specifically requested by your application programs. It is requested when "EXF=*file-number*" is specified in the record buffer of an OP command and it is relinquished by a CL command. EXF file control is incompatible with any other use of the file (except for online save requests); other users may not access or update the file while it is under EXF file control.
- **EXU** control is required by external utilities (such as the ADAULD utility) and can be specifically requested by your application programs. It is requested when "EXU=*file-number*" is specified in the record buffer of an OP command and it is relinquished by a CL command. EXU file control is incompatible with any other update access requests for the file (except for online save requests); other users may access, but not update the file while it is under EXU file control.

When requested, EXF or UTI control is given only for a file if the file is not already in use by another user or utility; EXU control is given only if the file is not already opened for update by another user. If the exclusive control request is denied, Adabas returns response code 48 (ADARSP048).

In addition to preventing competitive updating of a file, exclusive control may be used to simplify recovery procedures, in that the files may be restored regardless of other user activity.

The record hold commands need not (but may) be used for files for which the user has exclusive control. Adabas disables hold logic processing for files being updated under exclusive file control.

Exclusive control users are assumed to be non-ET logic users, and therefore not controlled by the ET timeout restrictions. However, if a non-ET user issues an ET command, that user automatically becomes an ET logic user and is subject to transaction timeout restrictions if records are put in hold status for any reason. That user retains exclusive control until either finished or timed out.

Users performing exclusive control updating can use the C1 command to request that a checkpoint be taken. The C1 checkpoint acts as a reference point to either remove updates that have been applied after the checkpoint, or to reapply updates that were applied before the checkpoint.

Note that a user (application) is accessing a file if the file is listed in the user's User Queue Element (UQE) file list. The file list identifies each file the user (application) is using and the type of file usage (ACC, EXU, EXF, or UPD). A file is added to the UQE file list when:

- it is requested for UTI use by a utility or Adabas Online System;
- it is specified in the record buffer of an OP command following one of the ACC, EXF, or EXU keywords;
- it is specified in the record buffer of an OP command with Command Option 1 set to "R" following the UPD keyword; or
- it is the target of a command without hold (ACC use of the file) or a command with hold (UPD use of the file) by a user who did not start the Adabas session with an OP command specifying Command Option 1 set to "R".

A file is deleted from the UQE file list when:

- a utility or Adabas Online System function with UTI control over the file ends;
- the user session ends; or
- the user's current transaction ends (e.g., in the course of an ET or BT command) and the file had not been added to the file list by the user's OP command.

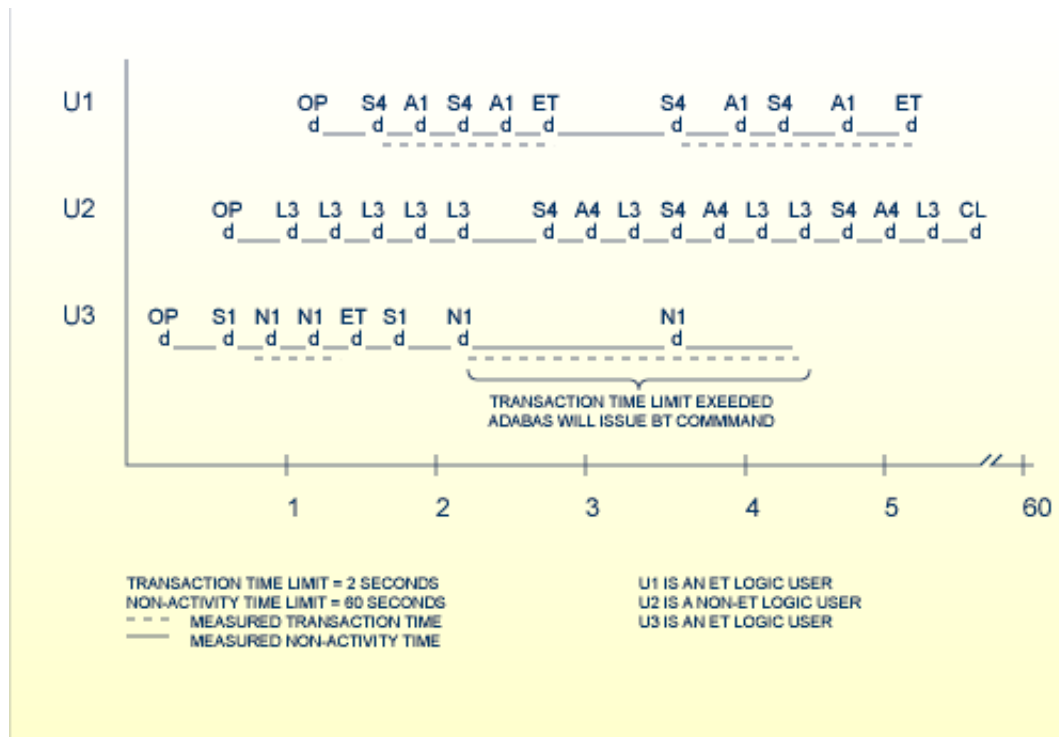
For more information about file usage by utilities and possible resource conflicts, read *Utility Usage of Files and Databases* and *Possible Resource Conflicts*.

Non-Activity Time Limit

All users (access-only, ET logic, or exclusive control) are subject to a non-activity time limit. Different non-activity time limits may be defined for each user type with the ADARUN TNAA, TNAE, and TNAX parameters (for more information, read *Adabas Initialization (ADARUN Statement)*).

A user-specific non-activity time limit may also be set with the OP command; the maximum is controlled with the ADARUN MXTNA parameter. For more information, read *OP Command: Open User Session*.

The following diagram summarizes the actions taken by Adabas when a user exceeds the non-activity time limit.



Transaction and Non-Activity Time Limits

This section describes these Adabas actions in the following topics:

- For an ET Logic User
- For an Exclusive File Control User (EXF)
- For an Access Only User

For an ET Logic User

For an ET logic user, Adabas performs the following processing:

- It issues a backout transaction (BT) command for the current user transaction (if necessary).
- It releases all records held during the transaction.
- It deletes the user's file list in the UQE.
- It deletes all command IDs for the user.

Adabas returns response code 9 (ADARSP009) to the user when the next command is issued if one of the following conditions occur:

- The user was not in ET status when the timeout occurred.
- The user was in ET status when the timeout occurred and provided a non-blank user ID in the OP command.

If the user was at ET status when the timeout occurred and did *not* provide a non-blank user ID in the OP command, the user's UQE is deleted.

For an Exclusive File Control User (EXF)

For an exclusive file control user, Adabas deletes the user's file list in the UQE. As a result, the user loses exclusive control of the file or files for which exclusive control was in effect.

In addition, all command IDs for the user are released and the user is changed to an access-only user.

For an Access Only User

For an access-only user, Adabas deletes the user's file list in the UQE.