

# CL Command: Close User Session

The CL command ends an ET session and updates the database.

We recommend that you set unused ACB and ACBX fields to binary zeros before the direct call is initiated.

This chapter covers the following topics:

- Function and Use
  - ACB Interface Direct Call: CL Command
  - ACBX Interface Direct Call: CL Command
  - Buffers
- 

## Function and Use

The CL command is used to terminate a user session. Software AG recommends that all user programs issue a CL command upon completion of database processing. User programs operating in single-user mode which are performing database updating must issue a CL command to ensure that all updates have been written to the database.

A CL command:

- issues an implicit ET command (ET-logic users only);
- stores user data in an Adabas system file (optional);
- releases all records currently in hold status for the user as well as all command ID entries (and corresponding ISN lists) assigned to the user;
- transfers the user's ET data from the Adabas Work to an Adabas system file. This is done only if a user ID was provided with the OP command; otherwise, any ET data stored during the session is lost.

A CL command issued by a user operating in single-user mode causes a physical close of the database (Associator, Data Storage, Work, and the data protection log). It is therefore not possible for a single-user mode user to follow a CL command with another command (for example, an OP command).

## ACB Interface Direct Call: CL Command

This section describes ACB interface direct calls for the CL command. It covers the following topics:

- Control Block and Buffer Overview
- Control Block Field Descriptions

- ACB Examples

## Control Block and Buffer Overview

### Control Block

| Field                | Position | Format       | Before Adabas Call | After Adabas Call |
|----------------------|----------|--------------|--------------------|-------------------|
|                      | 1-2      | --           | --                 | --                |
| Command Code         | 3-4      | alphanumeric | F                  | U                 |
| Command ID           | 5-8      | binary       | --                 | A                 |
| File Number ***      | 9-10     | binary       | F                  | U                 |
| Response Code        | 11-12    | binary       | --                 | A                 |
| ISN                  | 13-16    | binary       | --                 | A                 |
| ISN Lower Limit      | 17-20    | binary       | --                 | A                 |
| ISN Quantity         | 21-24    | binary       | --                 | A                 |
|                      | 25-26    | --           | --                 | --                |
| Record Buffer Length | 27-28    | binary       | F *                | U                 |
|                      | 29-35    | --           | --                 | --                |
| Command Option 2     | 36       | alphanumeric | F                  | U                 |
|                      | 37-72    | --           | --                 | --                |
| Command Time         | 73-76    | binary       | --                 | A                 |
| User Area            | 77-80    | --           | --                 | U                 |

### Buffer Areas

| Buffer   | Before Adabas Call | After Adabas Call |
|----------|--------------------|-------------------|
| Format * | **                 | --                |
| Record * | F                  | U                 |

where:

|     |   |
|-----|---|
| F   | Supplied by user before Adabas call   |
| A   | Supplied by Adabas  |
| U   | Unchanged after Adabas call   |
| *   | Required only if user data is to be stored  |
| **  | Not used but must be included in parameter list of call statement if user data to be stored   |
| *** | A database ID is only necessary if you are accessing a database other than the application's default database (read in by ADARUN DBID parameter, provided in the loaded link globals table, or linked with the link routine). |
| --  | Not used  |

## Control Block Field Descriptions

We recommend that you set unused ACB fields to binary zeros before the direct call is initiated.

### Command Code (ACBCMD)

CL

### Command ID (ACBCID)

For ET logic users, Adabas returns the transaction sequence number of the user's last successfully executed transaction in this field. The number is provided in binary format.

Because the CL command includes an ET command (see *Function and Use*), it also increments the transaction sequence number by one (regardless of whether it completes a transaction with or without update commands). The incremented transaction sequence number is then returned.

The CL command returns binary zeros if it ends the session of an ET user without update commands.

### File Number (ACBFNR)

A database ID is only necessary if you are accessing a database other than the application's default database (read in by ADARUN DBID parameter, provided in the loaded link globals table, or linked with the link routine).

### Response Code (ACBRSP)

In this field, Adabas returns the response code for the command. Response Code 0 indicates that the command was executed successfully. If the CL command returns a nonzero response code, the rightmost two bytes of the Adabas control block, Additions 2 field (bytes 47 and 48) may contain a subcode defining the exact response code meaning. Response codes and their subcodes are defined in the *Adabas Messages and Codes Manual* documentation.

### ISN: Number of I/Os (ACBISN)

Adabas returns the number of I/O operations resulting from this session's Adabas calls in this field.

### ISN Lower Limit: Number of Commands (ACBISL)

In this field, Adabas returns the number of commands issued by the user during the user session.

### ISN Quantity: CPU Time (ACBISQ)

In this field, Adabas returns an estimate of the amount of processor time used by this user for Adabas command processing.

The time is provided in units of 1.048576 seconds.

### Record Buffer Length (ACBRBL)

If user data is to be stored in an Adabas system file, the length of the record buffer must be specified in this field. The length specified determines the number of bytes of user data to be stored.

The maximum length which may be specified is 2000 bytes.

If no user data is to be stored, this field is not used.

### Command Option 2: Store User Data (ACBCOP2)

| Option | Description  |
|--------|--|
| E      | Indicates that user data is to be stored in an Adabas system file. |

## ACB Examples

### Example 1

The user program has completed all database activity and issues the CL command. No user data is to be stored.

#### Control Block

|                  |    |                              |
|------------------|----|------------------------------|
| Command Code     | CL |                              |
| Command Option 2 | b  | no user data is to be stored |

### Example 2

The user program issues a CL command and provides user data to be stored in an Adabas system file.

#### Control Block

|                      |    |                                    |
|----------------------|----|------------------------------------|
| Command Code         | CL |                                    |
| Record Buffer Length | 17 | 17 bytes of user data to be stored |
| Command Option 2     | E  | user data is to be stored          |

**Buffer**

|                      |                   |
|----------------------|-------------------|
| <b>Record Buffer</b> | USER 7 NORMAL END |
|----------------------|-------------------|

**ACBX Interface Direct Call: CL Command**

This section describes ACBX interface direct calls for the CL command. It covers the following topics:

- Control Block and Buffer Overview
- Control Block Field Descriptions

**Control Block and Buffer Overview****Control Block**

| Field              | Position | Format                  | Before Adabas Call | After Adabas Call |
|--------------------|----------|-------------------------|--------------------|-------------------|
|                    | 1-2      | ---                     | ---                | ---               |
| Version Indicator  | 3-4      | binary                  | F                  | U                 |
|                    | 5-6      | ---                     | ---                | ---               |
| Command Code       | 7-8      | alphanumeric            | F                  | U                 |
|                    | 9-10     | ---                     | ---                | ---               |
| Response Code      | 11-12    | binary                  | ---                | A                 |
| Command ID         | 13-16    | alphanumeric/<br>binary | ---                | A                 |
| Database ID****    | 17-20    | numeric                 | F                  | U                 |
|                    | 21-28    | ---                     | ---                | ---               |
| Number of I/Os     | 29-32    | binary                  | ---                | A                 |
|                    | 33-36    | ---                     | ---                | ---               |
| Number of Commands | 37-40    | binary                  | ---                | A                 |
| CPU Time           | 41-48    | binary                  | ---                | A                 |
|                    | 49       | ---                     | ---                | ---               |
| Command Option 2   | 50       | alphanumeric            | F                  | U                 |
|                    | 51-114   | ---                     | ---                | ---               |
| Error Subcode      | 115-116  | binary                  | ---                | A                 |
|                    | 117-144  | ---                     | ---                | ---               |
| Command Time       | 145-152  | binary                  | ---                | A                 |
| User Area          | 153-168  | not applicable          | ---                | U                 |
| ---                | 169-193  | do not touch            | ---                | ---               |

### ABDs and Buffers

| ABD and Buffer | Before Adabas Call | After Adabas Call |
|----------------|--------------------|-------------------|
| Format *       | **                 | --                |
| Record *       | F                  | U                 |

where:

|     |   |
|-----|---|
| F   | Supplied by user before Adabas call   |
| A   | Supplied by Adabas  |
| U   | Unchanged after Adabas call   |
| *   | Required only if user data to be stored   |
| **  | Not used but should be included in Adabas call or one will be automatically generated.  |
| *** | A database ID is only necessary if you are accessing a database other than the application's default database (read in by ADARUN DBID parameter, provided in the loaded link globals table, or linked with the link routine). |
| --  | Not used  |

## Control Block Field Descriptions

We recommend that you set unused ACBX fields to binary zeros before the direct call is initiated.

### Version Indicator (ACBXVER)

F2

### Command Code (ACBXCMD)

CL

### Response Code (ACBXRSP)

Adabas returns the response code for the command in this field. Response code 0 (ADARSP000) indicates that the command was executed successfully. Non-zero response codes, which can also have accompanying subcodes returned in the Error Subcode (ACBXERRC) field, are described in the *Adabas Messages and Codes Manual* documentation.

### Command ID (ACBXCID)

For ET logic users, Adabas returns the transaction sequence number of the user's last successfully executed transaction in this field. The number is provided in binary format.

Because the CL command includes an ET command (see *Function and Use*), it also increments the transaction sequence number by one (regardless of whether it completes a transaction with or without update commands). The incremented transaction sequence number is then returned.

The CL command returns binary zeros if it ends the session of an ET user without update commands.

### Database ID (ACBXDBID)

Use this field to specify the database ID only if you are accessing a database other than the application's default database (read in by ADARUN DBID parameter, provided in the loaded link globals table, or linked with the link routine). The Adabas call will be directed to this database.

This field is a four-byte binary field, but at this time only two-byte database IDs are supported. Therefore, the database ID should be specified in the low-order part (rightmost bytes) of the field, with leading binary zeros.

If this field is set to binary zeros, the Adabas API uses either the database ID from the ADARUN cards provided in DDCARD input data or the default database ID value provided in the LNKGBLS module linked with or loaded by the link routine.

#### **ISN: Number of I/Os (ACBXISN)**

Adabas returns the number of I/O operations resulting from this session's Adabas calls in this field.

#### **ISN Lower Limit: Number of Commands (ACBXISL)**

In this field, Adabas returns the number of commands issued by the user during the user session.

#### **ISN Quantity: CPU Time (ACBXISQG)**

In this field, Adabas returns an estimate of the amount of processor time used by this user for Adabas command processing.

The time is provided in units of 1/4096 microseconds; therefore, the high-order word of this field contains the estimated processor time in units of 1.048576 seconds.

#### **Command Option 2: Store User Data (ACBXCOP2)**

| <b>Option</b> | <b>Description</b>   |
|---------------|--|
| E             | Indicates that user data is to be stored in an Adabas system file. |

#### **Error Subcode (ACBXERRC)**

If the command returns a nonzero response code, this field contains a subcode defining the exact response code meaning. Response codes and their subcodes are defined in the *Adabas Messages and Codes Manual* documentation.

## **Buffers**

The following buffers apply to a CL command:

- Format Buffer
- Record Buffer

### **Format Buffer**

A format buffer is not used by the CL command, but should be included in the Adabas call. If this is an ACB interface direct call and a format buffer is not specified, a processing error will occur; ACB interface direct calls expect buffers to be specified in a set sequence. If this is an ACBX interface direct call and a format buffer is not specified, one will be automatically generated.

### **Record Buffer**

User data output from the command is stored in this buffer. The number of bytes actually stored is determined by the value specified in the record buffer length field. The data is retained only if you have issued an OP command in which a nonblank, unique user ID was provided. If so, the data is retained until



you issue the next ET or CL command in which user data is provided. If a nonblank user ID was not provided, the data cannot be retrieved in a subsequent session.