

L3 and L6 Commands: Read Logical Sequential

The L3 and L6 commands are used to read a file in logical sequential order, based on the sequence of the values for a given descriptor.

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: L3 and L6 Commands
 - ACBX Interface Direct Call: L3 and L6 Commands
 - Buffers
 - Additional Considerations
-

Function and Use

Specify the:

- file to be read;
- search descriptor whose values are to control the read sequence (phonetic descriptors and descriptors contained within or derived from a field within a periodic group may not be used);
- field or fields within each record for which values are to be returned.

Each L3 or L6 command returns the requested values for one record's field or fields in the record buffer. Repeating the command returns the values of each field in order of the search descriptor values.

The ascending and descending options specify whether the records are read in ascending or descending order. The ascending, descending, and value start options may be used to position to a given value before sequential reading begins or to position to a given value during sequential reading. This allows you to position to a specific record in the file without having to read each record.

The multifetch or prefetch options allow prior accessing of one or more sequential records, reducing overall operating time and eliminating the time needed for single-record fetches. Multifetching and prefetching can be enabled by specifying "M" or "O" (for multifetching) in the Command Option 1 field. Refer to the section *Using the Multifetch/Prefetch Feature* for more information.

If the format buffer contains "C.", the field values for each record read are returned in compressed format; otherwise, they are returned in uncompressed format.

Processing for the L3 command uses an internal table (the table of sequential commands) to keep track of the most recently returned value and ISN. *Normal* and *expanded* files are operationally diverse and for performance reasons, the value and ISN for each are handled differently in the table. Specifically, the entries for normal files keep the value and ISN that *was returned* on the last call. The entries for expanded files keep the value and ISN that must be returned to the application on the *next* call. This means that for normal files where concurrent updates are taking place, the L3 command returns a newly inserted value/ISN combination as long as it is *greater than or equal to* the last value/ISN returned. For expanded files where concurrent updates are taking place, the L3 command returns a newly inserted value/ISN combination as long as it is *greater than or equal to* the value/ISN that is kept in the table of sequential commands entry as the *next* value/ISN to be returned.

The L6 command performs the same function as the L3 command, but places each record read in hold status. If a record to be read and held is currently being held by another user, the read operation stops and the user is placed in wait status until the record becomes available or an operation timeout occurs. If the Command Option 1 field is set to either "O" or "R" and a requested record is already being held, the L6 command returns response code 145.

- Repositioning to a Particular Value
- Changing the Direction of the Read

Repositioning to a Particular Value

For information about specifying a starting value for the sequence-controlling descriptor, read about the ISN control block field in *ISN: Optional Start ISN/Record Read ISN* in either *ACB Interface Direct Call: L3 and L6 Commands* or *ACBX Interface Direct Call: L3 and L6 Commands*. The following procedure is used to *reposition* to a particular value during a sequential pass:

1. Set the last six positions of the Additions 1 field (the descriptor controlling the read sequence) to blanks.
2. Insert in the value buffer the value to which the read sequence is to reposition.
3. Set the ISN field to zero.
4. Set the Command Option 2 field to "A", "D", or "V".

Changing the Direction of the Read

Within a logical read sequence, the direction of the read can be changed at any time from ascending to descending or back without repositioning by specifying "D" or "A" for Command Option 2.

This option is not supported if prefetch or multifetch processing is also requested.

ACB Interface Direct Call: L3 and L6 Commands

This section describes ACB interface direct calls for L3 and L6 commands. 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	binary	--	--
Command Code	3-4	alphanumeric	F	U
Command ID	5-8	alphanumeric	F	U
File Number	9-10	binary	F	U
Response Code	11-12	binary	--	A
ISN	13-16	binary	F	A
ISN Lower Limit	17-20	binary	F	U
	21-24	--	--	--
Format Buffer Length	25-26	binary	F	U
Record Buffer Length	27-28	binary	F	U
Search Buffer Length	29-30	binary	F *	U
Value Buffer Length	31-32	binary	F *	U
ISN Buffer Length **	33-34	binary	F	U
Command Option 1	35	alphanumeric	F	U
Command Option 2	36	alphanumeric	F	U
Additions 1	37-44	alphanumeric	F	A
Additions 2	45-48	binary / binary	--	A
Additions 3	49-56	alphanumeric	F	A
Additions 4	57-64	alphanumeric	F	A
Additions 5	65-72	alphanumeric	F	U
Command Time	73-76	binary	--	A
User Area	77-80	--	--	U

Buffer Areas

Buffer	Before Adabas Call	After Adabas Call
Format	F ***	U
Record	--	A
Search *	F	U
Value *	F	U
ISN **	F	U

where:

- F Supplied by user before Adabas call
- A Supplied by Adabas
- U Unchanged after Adabas call
- * Required only if value start option is being used
- ** The ISN buffer and length is required only if the multifetch or prefetch option is specified.
- *** May contain compress option control characters "C."
- 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)

L3 or L6

Command ID (ACBCID)

This field must be set to a non-blank, non-zero value. The value provided is used by Adabas to provide the records in the correct sequence and to avoid repetitive interpretation of the format buffer.

The first byte of this field may not be set to hexadecimal 'FF', and must not be changed during a given sequential pass of a file.

If the command ID value is X'FFFFFFFF', the command ID is automatically generated. The Adabas nucleus generates command ID values beginning with X'00000001', and increments the value by 1 for each L3 or L6 call.

When specifying user-defined command IDs, the user must ensure that each command ID is unique.

See also the Additions 5 field for separate format ID and/or global format ID usage.

File Number (ACBFNR)

Specify the binary number of the file to be read in this field. For physical direct calls, specify the file number as follows:

- For a one-byte file number, enter the file number in the rightmost byte (10); the leftmost byte (9), should be set to binary zero (B'0000 0000').
- For a two-byte file number, use both bytes (9 and 10) of the field.

Note:

When using two-byte file numbers and database IDs, a X'30' must be coded in the first byte of the control block.

Response Code (ACBRSP)

Adabas returns the response code for the command in this field. Response code 0 indicates that the command was executed successfully. Non-zero response codes, which can also have accompanying subcodes returned in the rightmost half of the Additions 2 field, are described in the *Adabas Messages and Codes Manual* documentation.

Response code 3 indicates that an end-of-file condition has been detected.

ISN: Optional Start ISN/Record Read ISN (ACBISN)

For initial or subsequent positioning to a specific location in the logical read sequence when the last six bytes of the Additions 1 field are set to blanks, the ISN field is used together with the starting descriptor value specified in the value buffer to determine the position where reading starts or restarts.

If an existing read sequence is to be continued without repositioning when the last six bytes of the Additions 1 field are not changed from the most recent read command of the sequence, the ISN field is ignored.

Adabas returns in the ISN field the ISN of the record that was read.

When positioning or repositioning in the logical read sequence occurs, the ISN field is not used for positioning if the comparator for the starting descriptor value is set to:

- GT (greater than) for an ascending read sequence; or
- LT (less than) for a descending read sequence.

Otherwise, the ISN field plays a role and the following rules apply:

Rules for Reading Forward (Ascending Option)

If the starting descriptor value specified in the value buffer *is* present in the logical read sequence, specifying:

- an ISN of zero results in positioning to the *first (lowest)* ISN with that descriptor value;
- a non-zero ISN results in positioning to the *lowest ISN greater than* the specified ISN with that descriptor value, if such an ISN exists. Otherwise, positioning is to the *first ISN of the next higher* descriptor value.

If the starting descriptor value specified in the value buffer is *not* present in the logical read sequence, positioning is always to the *first ISN of the next higher* descriptor value, regardless of the specified ISN.

Rules for Reading Backward (Descending Option)

If the starting descriptor value specified in the value buffer *is* present in the logical read sequence, specifying:

- an ISN of zero results in positioning to the *last (highest)* ISN with that descriptor value;
- a non-zero ISN results in positioning to the *highest ISN less than* the specified ISN with that descriptor value, if such an ISN exists. Otherwise, positioning is to the *last ISN of the next lower* descriptor value.

If the starting descriptor value specified in the value buffer is *not* present in the logical read sequence, positioning is always to the *last ISN of the next lower* descriptor value, regardless of the specified ISN.

ISN Lower Limit: Multifetch Record Count (ACBISL)

If either "M" or "O" (multifetching option) is specified in the Command Option 1 field, a non-zero value in this field determines the maximum number of records to be multifetched. If this value is zero, the number of records to be multifetched is limited by the record and ISN buffer lengths. Refer to the section *Using the Multifetch/Prefetch Feature* for more information.

Format Buffer Length (ACBFBL)

The format buffer length (in bytes). The format buffer area defined in the user program must be at least as large as the length specified.

Record Buffer Length (ACBRBL)

The record buffer length (in bytes). The record buffer area defined in the user program must be at least as large as the length specified.

Search Buffer Length (ACBSBL)

The search buffer length (in bytes). The search buffer area defined in the user program must be at least as large as the length specified.

Value Buffer Length (ACBVBL)

The value buffer length (in bytes). The value buffer area defined in the user program must be at least as large as the length specified.

ISN Buffer Length (ACBIBL)

The ISN buffer length (in bytes).

When the Command Option 1 field specifies "P", the L3 and L6 commands use the ISN buffer to hold prefetched descriptor values. The ISN buffer must be large enough to hold the largest descriptor value plus a 16-byte header preceding each value. The actual ISN buffer area defined in the user program must be at least as large as the length specified.

Command Option 1 (ACBCOP1)

Option	Description
M (multifetching)	Multifetch processing is activated.
O (multifetching with the R option)	Multifetch processing as well as option R (see below) processing is activated.
P (prefetching)	Prefetch processing is activated.
R (return)	Returns response code 145 if the record to be read and held by an L6 command is not available.

Specifying "M", "O", or "P" indicates that the (command-level) prefetch or multifetch option is to be used. The multifetch/prefetch option can improve performance by allowing prior access to one or more sequential records, reducing overall operating time and eliminating the time needed for single-record fetches. For more information, see the section *Using the Multifetch/Prefetch Feature*.

Command Option 2 (ACBCOP2)

This option specifies the sequence in which the descriptor's entries are to be processed.

Option	Description
A (ascending)	Processes entries for the descriptor in ascending sequence with an optional specification of a starting value. A starting descriptor value can be specified in the value buffer. In addition, a non-null value in the ISN field can be used to position within multiple matching descriptor value entries. If the search buffer and value buffer are omitted (SBL and VBL are set to zero (0)), all entries for the descriptor are processed.
D (descending)	Processes entries for the descriptor in descending sequence with an optional specification of a starting value. A starting descriptor value can be specified in the value buffer. In addition, a non-null value in the ISN field can be used to position within multiple matching descriptor value entries. If the search buffer and value buffer are omitted (SBL and VBL are set to zero (0)), all entries for the descriptor are processed.
V (value start)	Specifies an <i>ascending</i> sequential pass to begin (or continue) at a user-specified value rather than with the lowest (or next) value of the descriptor. The user-specified value <i>must</i> be in the value buffer, and the value's length and format <i>must</i> be specified in the search buffer. A non-null value specified in the ISN field can further qualify the read operation. <i>This option was used in Adabas prior to version 6; it is maintained to support existing programs.</i>
blank	Specifies an <i>ascending</i> sequential pass without a starting value; all values present are processed. The search buffer and value buffer are ignored when present. <i>This option was used in Adabas prior to version 6; it is maintained to support existing programs.</i>

Additions 1: Descriptor Used for Sequence Control (ACBADD1)

The descriptor to be used to control the read sequence must be specified in this field. A descriptor, subdescriptor, or superdescriptor may be specified.

- Phonetic descriptors and descriptors contained within or derived from a field contained within a periodic group may not be specified.
- A descriptor which is a multiple-value field may be specified, in which case the same record may be read several times (once for each different value within a given record) during the sequential pass through the file.
- If the descriptor specified is defined with the null value suppression (NU) option, any records containing a null value for the descriptor will not be read.
- The descriptor name is specified in the first two positions of this field. All remaining positions must be set to blanks on the initial call.

The Additions 1 field should not be modified between L3 or L6 calls. The exception is when the user wishes to reposition to a particular value during a sequential pass. This is done by setting the last six positions of this field to blanks, inserting the value to which repositioning is to occur in the value buffer, setting the ISN field to zero, and setting the Command Option 2 field to "A", "D", or "V".

Additions 2: Length of Compressed and Decompressed Record (ACBADD2)

If the command is processed successfully, the following information is returned in this field:

- If the record buffer contains at least one valid field value, the leftmost two bytes contain the length (in binary form) of the compressed record accessed;
- The rightmost two bytes contain the length (in binary form) of the decompressed fields selected by the format buffer and accessed.

Note:

This length information is not returned when the prefetch feature is being used.

If the L3 or L6 command returns a non-zero response code, the rightmost two bytes may contain a subcode defining the exact response code meaning. Response codes and their subcodes are described in the *Adabas Messages and Codes Manual* documentation.

Additions 3: Password (ACBADD3)

This field is used to provide an Adabas security password. If the database, file, or fields are security protected, the user must provide a valid security password. Adabas sets the Additions 3 field to blanks during command processing to enhance password integrity.

Additions 4: Cipher Code (ACBADD4)

This field is used to provide a cipher code. If the file is ciphered, the user must provide a valid cipher code. If the file is not ciphered, this field should be set to blanks.

Adabas sets any cipher code to blanks during command processing, and returns a version code and database ID in the rightmost (low-order) three bytes of this field. For more information, see the section *Control Block Fields*.

Additions 5: Format ID, Global Format ID (ACBADD5)

Use this field to specify a separate format ID that identifies the internal format buffer used for this command, or to provide a global format ID allowing use of the internal format buffer by all users.

As long as the leftmost bit of the Additions 5 field is set to 0, the value provided in the command ID field is used as the format ID as well.

If, however, this bit is set to 1, the fifth through eighth bytes of the Additions 5 field are used as the format ID.

If the two high-order (leftmost) bits of the first byte of Additions 5 field are set to one (B'11'), all eight bytes of the Additions 5 field are used as a global format ID (that is, the format ID can be used by several users at the same time).

See the section *Command, Format, and Global Format IDs* for more information and examples.

ACB Examples**Example 1**

File 2 is to be read in logical sequential order. The descriptor RB is to be used for sequence control. The values for the fields RA and RB are to be returned. The entire file is to be read.

Control Block

Command Code	L3	
Command ID	EX01	a non-blank CID is required
File Number	2	
ISN	0	all records are to be read
Format Buffer Length	6	or larger
Record Buffer Length	18	or larger
Search Buffer Length	0	
Value Buffer Length	0	
Command Option 1	b	response code 145 or prefetch options not used
Command Option 2	A	Ascending order with no search or value buffer specified
Additions 1	RB bbbbbb	descriptor RB to be used for sequence control, where <i>bbbbbb</i> represents blanks.
Additions 3	password	file is security-protected
Additions 4	<i>bbbbbb</i> (blanks)	file is not ciphered

Buffer Areas

Format Buffer	RA,RB.
----------------------	--------

The L3 call is repeated to obtain each successive record. The CID and Additions 1 field must not be modified between L3 calls. Response code 3 will be returned when all records have been read.

Example 2: Read Logical Sequential with Record Hold

File 1 is to be read in logical sequential order. The descriptor AA is to be used for sequence control. The values for fields AA and AB are to be returned. Each record read is to be placed in hold status.

Control Block

Command Code	L6	
Command ID	EX02	non-blank CID required
File Number	1	
ISN	0	all records to be read
Format Buffer Length	3	or larger
Record Buffer Length	10	or larger
Search Buffer Length	0	
Value Buffer Length	0	
Command Option 1	b	response code 145 or prefetch options not used
Command Option 2	b	Ascending order with no search or value buffer specified
Additions 1	AA bbbbbb	descriptor AA to be used for sequence control, where <i>bbbbbb</i> represents blanks.
Additions 3	<i>bbbbbb</i> (blanks)	file is not security-protected
Additions 4	<i>bbbbbb</i> (blanks)	file is not ciphered

Buffer Areas

Format Buffer	GA.
----------------------	-----

To complete logical transactions and release held records, ET logic users should issue an ET command. Non-ET users should release held records with an A4, E4, or RI command.

Example 3: ASCENDING Option

The same requirement as example 1 except that reading is to begin with the value "N".

Control Block

Command Code	L3	
Command ID	EX03	non-blank CID required
File Number	2	
ISN	0	all records are to be read
Format Buffer Length	6	or larger
Record Buffer Length	18	or larger
Search Buffer Length	7	or larger
Value Buffer Length	1	or larger
Command Option 1	b	response code 145 option not used
Command Option 2	A	ascending option with start value supplied
Additions 1	RB bbbbbb	RB is to be used for sequence control, where <i>bbbbbb</i> represents blanks.
Additions 3	password	file is security-protected
Additions 4	<i>bbbbbb</i> (blanks)	file is not ciphered

Buffer Areas

Format Buffer	RA, RB.	
Search Buffer	RB, 1, A.	
Value Buffer	N	reading to begin with value N

The initial L3 call will result in the return of the first record which contains the value "N" for the descriptor RB. If no records exist with this value, the first record which contains a value greater than "N" (such as "NA") will be returned.

Example 4: DESCENDING Option with Repositioning

The same requirement as example 3 except that a value repositioning is to be performed. The sequential read process is to continue with the value "Q".

Control Block

Command Code	L3	
Command ID	EX03	the CID field may not be changed when repositioning
File Number	2	
ISN	0	this field must be set to zeros for value repositioning
Format Buffer Length	6	or larger
Record Buffer Length	18	or larger
Search Buffer Length	7	or larger
Value Buffer Length	1	or larger
Command Option 1	b	response code 145 option not used
Command Option 2	D	descending option with value repositioning
Additions 1	RB bbbbbb	the last six positions of this field must be set to blanks for value repositioning
Additions 3	password	file 2 is security-protected
Additions 4	bbbbbb (blanks)	file 2 is not ciphered

Buffer Areas

Format Buffer	RA,RB.	
Search Buffer	RB,1,A,LE.	
Value Buffer	Q	reposition to value "Q"

ACBX Interface Direct Call: L3 and L6 Commands

This section describes ACBX interface direct calls for L3 and L6 commands. 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/ binary	F	U
Database ID	17-20	numeric	F	U
File Number	21-24	numeric	F	U
	25-28	---	---	---
ISN	29-32	binary	F	A
	33-36	---	---	---
ISN Lower Limit	37-40	binary	F	U
	41-48	---	---	---
Command Option 1	49	alphanumeric	F	U
Command Option 2	50	alphanumeric	F	U
	51-56	---	---	---
Additions 1	57-64	alphanumeric/ binary	F	A
Additions 2	65-68	binary	---	A
Additions 3	69-76	alphanumeric/ binary	F	A
Additions 4	77-84	alphanumeric	F	A
Additions 5	85-92	alphanumeric/ binary	F	U
	93-114	---	---	---
Error Subcode	115-116	binary	---	A
	117-144	---	---	---

Field	Position	Format	Before Adabas Call	After Adabas Call
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	F ***	U
Record	--	A
Search *	F	U
Value *	F	U
Multifetch **	F	U

where:

- F Supplied by user before Adabas call
- A Supplied by Adabas
- U Unchanged after Adabas call
- * Required only if value start option is being used
- ** The multifetch buffer is required only if the multifetch option is specified.
- *** May contain compress option control characters "C."
-

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)

L3 or L6

Response Code (ACBXRSP)

Adabas returns the response code for the command in this field. Response code 0 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.

Response code 3 indicates that an end-of-file condition has been detected.

Command ID (ACBXCID)

This field must be set to a non-blank, non-zero value. The value provided is used by Adabas to provide the records in the correct sequence and to avoid repetitive interpretation of the format buffer.

The first byte of this field may not be set to hexadecimal 'FF', and must not be changed during a given sequential pass of a file.

If the command ID value is X'FFFFFFFF', the command ID is automatically generated. The Adabas nucleus generates command ID values beginning with X'00000001', and increments the value by 1 for each L3 or L6 call.

When specifying user-defined command IDs, the user must ensure that each command ID is unique.

See also the Additions 5 field for separate format ID and/or global format ID usage.

Database ID (ACBXDBID)

Use this field to specify the database ID. 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.

File Number (ACBXFNR)

Use this field to specify the number of the file to which the Adabas call should be directed.

This field is a four-byte binary field, but the file number should be specified in the low-order part (rightmost bytes) of the field, with leading binary zeros.

ISN: Optional Start ISN/Record Read ISN (ACBXISN)

For initial or subsequent positioning to a specific location in the logical read sequence when the last six bytes of the Additions 1 field are set to blanks, the ISN field is used together with the starting descriptor value specified in the value buffer to determine the position where reading starts or restarts.

If an existing read sequence is to be continued without repositioning when the last six bytes of the Additions 1 field are not changed from the most recent read command of the sequence, the ISN field is ignored.

The ACBXISN field is a four-byte binary field embedded in the eight-byte ACBXISNG field, which is not yet used. Set the high-order part of the ACBXISNG field to binary zeros.

Adabas returns in the ISN field the ISN of the record that was read.

When positioning or repositioning in the logical read sequence occurs, the ISN field is not used for positioning if the comparator for the starting descriptor value is set to:

- GT (greater than) for an ascending read sequence; or
- LT (less than) for a descending read sequence.

Otherwise, the ISN field plays a role and the following rules apply:

Rules for Reading Forward (Ascending Option)

If the starting descriptor value specified in the value buffer *is* present in the logical read sequence, specifying:

- an ISN of zero results in positioning to the *first (lowest)* ISN with that descriptor value;
- a non-zero ISN results in positioning to the *lowest ISN greater than* the specified ISN with that descriptor value, if such an ISN exists. Otherwise, positioning is to the *first ISN of the next higher* descriptor value.

If the starting descriptor value specified in the value buffer is *not* present in the logical read sequence, positioning is always to the *first ISN of the next higher* descriptor value, regardless of the specified ISN.

Rules for Reading Backward (Descending Option)

If the starting descriptor value specified in the value buffer *is* present in the logical read sequence, specifying:

- an ISN of zero results in positioning to the *last (highest)* ISN with that descriptor value;
- a non-zero ISN results in positioning to the *highest ISN less than* the specified ISN with that descriptor value, if such an ISN exists. Otherwise, positioning is to the *last ISN of the next lower* descriptor value.

If the starting descriptor value specified in the value buffer is *not* present in the logical read sequence, positioning is always to the *last ISN of the next lower* descriptor value, regardless of the specified ISN.

ISN Lower Limit: Multifetch Record Count (ACBXISL)

If either "M" or "O" (multifetching option) is specified in the Command Option 1 field, a non-zero value in this field determines the maximum number of records to be multifetched. If this value is zero, the number of records to be multifetched is limited by the record and multifetch buffer lengths. Refer to the section *Using the Multifetch/Prefetch Feature* for more information.

The ACBXISL field is a four-byte binary field embedded in the eight-byte ACBXISLG field, which is not yet used. Set the high-order part of the ACBXISLG field to binary zeros.

Command Option 1 (ACBXCOP1)

Option	Description
M (multifetching)	Multifetch processing is activated.
O (multifetching with the R option)	Multifetch processing as well as option R (see below) processing is activated.
R (return)	Returns response code 145 if the record to be read and held by an L6 command is not available.

Specifying the M or O options indicates that the (command-level) multifetch option is to be used. The multifetch option can improve performance by allowing prior access to one or more sequential records, reducing overall operating time and eliminating the time needed for single-record fetches. For more information, see the section *Using the Multifetch/Prefetch Feature*.

Command Option 2 (ACBXCOP2)

This option specifies the sequence in which the descriptor's entries are to be processed.

Option	Description
A (ascending)	Processes entries for the descriptor in ascending sequence with an optional specification of a starting value. A starting descriptor value can be specified in the value buffer. In addition, a non-null value in the ISN field can be used to position within multiple matching descriptor value entries. If the search buffer and value buffer are omitted (SBL and VBL are set to zero (0)), all entries for the descriptor are processed.
D (descending)	Processes entries for the descriptor in descending sequence with an optional specification of a starting value. A starting descriptor value can be specified in the value buffer. In addition, a non-null value in the ISN field can be used to position within multiple matching descriptor value entries. If the search buffer and value buffer are omitted (SBL and VBL are set to zero (0)), all entries for the descriptor are processed.
V (value start)	Specifies an <i>ascending</i> sequential pass to begin (or continue) at a user-specified value rather than with the lowest (or next) value of the descriptor. The user-specified value <i>must</i> be in the value buffer, and the value's length and format <i>must</i> be specified in the search buffer. A non-null value specified in the ISN field can further qualify the read operation. <i>This option was used in Adabas prior to version 6; it is maintained to support existing programs.</i>
blank	Specifies an <i>ascending</i> sequential pass without a starting value; all values present are processed. The search buffer and value buffer are ignored when present. <i>This option was used in Adabas prior to version 6; it is maintained to support existing programs.</i>

Additions 1: Descriptor Used for Sequence Control (ACBXADD1)

The descriptor to be used to control the read sequence must be specified in this field. A descriptor, subdescriptor, or superdescriptor may be specified.

- Phonetic descriptors and descriptors contained within or derived from a field contained within a periodic group may not be specified.
- A descriptor which is a multiple-value field may be specified, in which case the same record may be read several times (once for each different value within a given record) during the sequential pass through the file.
- If the descriptor specified is defined with the null value suppression (NU) option, any records containing a null value for the descriptor will not be read.
- The descriptor name is specified in the first two positions of this field. All remaining positions must be set to blanks on the initial call.

The Additions 1 field should not be modified between L3 or L6 calls. The exception is when the user wishes to reposition to a particular value during a sequential pass. This is done by setting the last six positions of this field to blanks, inserting the value to which repositioning is to occur in the value buffer, setting the ISN field to zero, and setting the Command Option 2 field to "A", "D", or "V".

Additions 2: Length of Compressed and Decompressed Record (ACBXADD2)

If the command is processed successfully, the following information is returned in this field:

- If the record buffer contains at least one valid field value, the leftmost two bytes contain the length (in binary form) of the compressed record accessed;
- The rightmost two bytes contain the length (in binary form) of the decompressed fields selected by the format buffer and accessed.

Note:

This length information is not returned when the prefetch feature is being used.

Additions 3: Password (ACBXADD3)

This field is used to provide an Adabas security password. If the database, file, or fields are security protected, the user must provide a valid security password. Adabas sets the Additions 3 field to blanks during command processing to enhance password integrity.

Additions 4: Cipher Code (ACBXADD4)

This field is used to provide a cipher code. If the file is ciphered, the user must provide a valid cipher code. If the file is not ciphered, this field should be set to blanks.

Adabas sets any cipher code to blanks during command processing, and returns a version code and database ID in the rightmost (low-order) three bytes of this field. For more information, see the section *Control Block Fields*.

Additions 5: Format ID, Global Format ID (ACBXADD5)

Use this field to specify a separate format ID that identifies the internal format buffer used for this command, or to provide a global format ID allowing use of the internal format buffer by all users.

As long as the leftmost bit of the Additions 5 field is set to 0, the value provided in the command ID field is used as the format ID as well.

If, however, this bit is set to 1, the fifth through eighth bytes of the Additions 5 field are used as the format ID.

If the two high-order (leftmost) bits of the first byte of Additions 5 field are set to one (B'11'), all eight bytes of the Additions 5 field are used as a global format ID (that is, the format ID can be used by several users at the same time).

See the section *Command, Format, and Global Format ID* for more information and examples.

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 L3 and L6 commands:

- Format Buffer
- Record Buffer
- Search Buffer
- Value Buffer
- Search and Value Buffer Examples
- ISN Buffer
- Multifetch Buffer

Format Buffer

The fields for which values are to be returned must be specified in this buffer. Descriptions of the syntax and examples of format buffer construction are provided in *Defining Buffers*.

Specifying "C." in the first two positions indicates that the compressed option is in effect. This causes the record to be returned in compressed instead of decompressed format.

Record Buffer

Adabas returns the requested field values in this buffer. The field values are returned according to the standard format and length of each field, unless the user has requested a different format and/or length in the format buffer.

Search Buffer

Note:

Commas separate elements in the search buffer; a period terminates the syntax statement.

The search buffer length (SBL) must be set in the control block. If the search buffer is not used, SBL must be set to zero (0).

If the value start (V) option is used for Command Option 2, the starting value of the descriptor used for sequence control *must* be specified in the value buffer and the value's length and format must be specified in the search buffer.

If the ascending (A) or descending (D) option is used for Command Option 2, a search buffer may be used but is not required. If either the search or value buffer is omitted, all values for a given descriptor are processed. If a starting value, ending value, or both are specified in the value buffer, the search buffer *must* be used to limit the number of descriptor entries retrieved.

The length and format of the descriptor value as provided in the value buffer must be specified in the search buffer if different from the standard length and/or format of the named descriptor.

When a single value is provided in the value buffer (that is, the starting value with option A or the ending value with option D), the syntax for the search buffer is

```
name [ ,length ] [ ,format ] [ ,comparator ]
```

When two values are provided in the value buffer (that is, a starting and an ending value), the syntax for the search buffer is

```
name [ ,length ] [ ,format ] ,S ,name [ ,length ] [ ,format ]
```

The elements used in the search buffer syntax statements are

name	is the name of the descriptor to be used for sequence control. The name specified must be the same as that specified in the Additions 1 field.								
length	is the length of the value provided in the value buffer. If the length is not specified, it is assumed that the value is being provided using the standard length of the descriptor. See the section <i>Length and Data Format</i> for the allowed length settings.								
format	is the format of the value provided in the value buffer. If the format is not specified, it is assumed that the value is being provided using the standard format of the descriptor. See the section <i>Length and Data Format</i> for the allowed format settings.								
comparator	identifies the scope of the read sequence: <table border="0" style="width: 100%;"> <tr> <td style="padding-right: 20px;">GE</td> <td>greater than or equal to the value to/from the highest value (the default)</td> </tr> <tr> <td>GT</td> <td>greater than the value to/from the highest value</td> </tr> <tr> <td>LE</td> <td>less than or equal to the value to/from the lowest value</td> </tr> <tr> <td>LT</td> <td>less than the value to/from the lowest value</td> </tr> </table>	GE	greater than or equal to the value to/from the highest value (the default)	GT	greater than the value to/from the highest value	LE	less than or equal to the value to/from the lowest value	LT	less than the value to/from the lowest value
GE	greater than or equal to the value to/from the highest value (the default)								
GT	greater than the value to/from the highest value								
LE	less than or equal to the value to/from the lowest value								
LT	less than the value to/from the lowest value								
S	A FROM-TO range (inclusive) that involves two search expressions. The same descriptor must be used in both expressions: <table border="0" style="width: 100%;"> <tr> <td style="padding-right: 20px;">AA,S,AA.</td> <td>valid</td> </tr> <tr> <td>AA,S,AB.</td> <td>invalid</td> </tr> </table>	AA,S,AA.	valid	AA,S,AB.	invalid				
AA,S,AA.	valid								
AA,S,AB.	invalid								

See *Example 3: Overview of Sequence Options* for an overview of sequence options resulting from the choice of A or D for Command Option 2 and various search and value buffer options.

Value Buffer

The value buffer length (VBL) must be set in the control block. If the value buffer is not used, VBL must be set to zero (0).

If the value start option is to be used, or if value repositioning is to be performed, the value with which reading is to start (or continue) must be specified in this buffer.

If the ascending (A) or descending (D) option is used for Command Option 2, a value buffer may be used but is not required. If either the search or value buffer is omitted, all values for a given descriptor are processed. If a starting value, ending value, or both are specified in the value buffer, the search buffer is required in order to limit the number of descriptor entries retrieved.

If the ascending (A) option is used for Command Option 2, reading starts (or continues) with the first record containing the value specified. If there are no records with keys equal to a start or reposition (continue) value, reading begins with the first record containing the next higher value.

When two values are provided in the value buffer, the first specifies the lower limit of a range and the second specifies the upper limit. Each value is either the starting or ending value depending on the Command Option 2 setting of A or D.

In addition, if the ISN field is set to a non-zero value, reading will start (or continue) or end with the first record whose ISN is present for the value specified, provided that the ISN is greater than the ISN in the ISN field. If no such ISN exists, the first record with the next higher (or lower) value is read.

Search and Value Buffer Examples

Example 1: Ascending Option with Optional Search and Value Buffer

Inverted list for the descriptor used for sequence control:

Value	ISN List
A	1, 4
B	2
D	3, 5

Initial L3 or L6 command with Command Option 2 set to "A" (ascending option), or subsequent L3 or L6 command with Command Option 2 set to "A" and the last six bytes of Additions 1 reset to blanks:

User-Supplied Value	User-Supplied ISN	ISN Where Reading Starts or Continues
A	0	1
A	1	4
A	2	4
A	4	2
A	5	2
B	0	2
B	1	2
B	2	3
B	3	3
BABC	1	3
C	0	3
D	0	3
D	3	5
D	4	5
D	5	response code 3
E-Z	-	response code 3

Example 2: Descending Option with Optional Search and Value Buffer

Inverted list for the descriptor used for sequence control:

Value	ISN List
A	1, 9, 25
B	3, 18, 21
C	7, 8, 11

Initial L3 or L6 command with Command Option 2 set to "D" (descending option), ISN=0 for optional start, search buffer comparator set to "LT", and value buffer containing value "C". The start position is the *highest* ISN of the next *lower* descriptor value: in this example, ISN 21 of descriptor value "B".

Example 3: Overview of Sequence Options

The following table illustrates the possibilities for using the ascending/descending option in conjunction with various search buffer and value buffer contents. The following applies to the file being read and ISN=0 (no further positioning within matching start values):

2. You are permitted to update or delete records from a file that you are reading with an L3 or L6 command. If the inserted record or records are to be read, you must reposition after having inserted the record or records.
3. If any user is updating the file being read with an L3 or L6 command, it is possible that you may not receive one or more records in the file while reading it with the L3 or L6 command. In addition, you may receive the same record more than once during a sequential pass of the file.

Note:

You should avoid updating any descriptor field that controls the read sequence with a value greater than that which it currently contains.