

L9 Command: Read Descriptor Values

The L9 command reads the values of a specified descriptor.

This chapter covers the following topics:

- Function and Use
 - Command: L9
 - Control Block
 - Buffers
 - Additional Considerations
 - Examples
-

Function and Use

The L9 command is used to determine the range of values present for a descriptor and the number of records that contain each value.

The user specifies the file containing the descriptor, the descriptor for which the values are to be returned, and the value at which processing is to begin. Each L9 call returns the next value for the descriptor in the record buffer, and the count of records containing that value in the ISN quantity field. The values may be either positive or negative. Null values for descriptors defined with the null value suppression (NU) option are not returned.

The multifetch and prefetch options improve performance by reading several descriptor values at a time. Multi-/prefetching can be enabled by specifying "M" (for multifetching) or "P" (for prefetching) in the command option 1 field. Refer to the section *Using the Multifetch/Prefetch Feature* for more information.

The "I" option specified in the command option 2 field returns the ISNs of each value in the record buffer. The L9 command reads the Associator inverted lists only; no Data Storage access is required.

The "A" and "D" options in the command option 2 field specify whether the descriptor values are read in ascending or descending order, respectively.

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.

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 by specifying "D" or "A" for command option 2.

This option is not supported with prefetch or multifetch.

Command: L9

User Control Block

Field	Position	Format	Before Adabas Call	After Adabas Call
	1-2	--	--	--
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	--	A
ISN LOWER LIMIT	17-20	binary	F	A
ISN QUANTITY	21-24	binary	--	A
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	U
ADDITIONS 2	45-48	binary	--	A
ADDITIONS 3	49-56	alphanumeric	F	A
	57-64	--	--	--
ADDITIONS 5	65-72	alphanumeric	F	--
COMMAND TIME	73-76	binary	--	A
USER AREA	77-80	--	--	U

User Buffer Areas

Buffer	Before Adabas Call	After Adabas Call
FORMAT BUFFER	F	U
RECORD BUFFER	--	A
SEARCH BUFFER	F	U
VALUE BUFFER	F	U
ISN BUFFER *	--	A

where:

- F Filled in by user before Adabas Call
- A Filled in by Adabas
- U Unchanged after Adabas call
- * The ISN buffer and length required only if the multi-/prefetch option is specified
- Not used

Control Block

Command Code

L9

Command ID

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

This field must not be modified during a given sequential pass of the file.

The first byte of this field may not be set to hexadecimal 'FF'.

File Number

Specify the binary number of the file to be read in this field. For the 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

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* documentation.

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

ISN: Periodic Group Occurrence

If the descriptor for which values are to be returned is contained within a periodic group, Adabas returns in this field the occurrence number in which the value being returned is located. The occurrence number is provided in binary format in the two low-order bytes.

If the prefetch option is specified, any occurrence for prefetched values is returned in the header preceding the value in the ISN buffer.

ISN Lower Limit: Lowest ISN in Record Buffer

Adabas returns values in this field as follows:

Cmd Op 1	Cmd Op 2	The L9 command . . .
blank	I	returns the ISNs for each value in the record buffer; no value is returned in the ISN lower limit field.
	not I	places the first ISN of the returned ISN list in the ISN lower limit field.
P(prefetch)	I	returns the first ISN in the record buffer and all prefetched descriptor values in the ISN buffer, preceded by a 16-byte header; no value is returned in the ISN lower limit field.
	not I	places the first ISN of the last value prefetched in the ISN lower limit field.
M(ultifetch)	I	returns the group of multifetched records in the record buffer and a description of these records in the caller's ISN buffer; no value is returned in the ISN lower limit field.
	not I	places the first ISN of the last value multifetched in the ISN lower limit field.

If command option 1 is set to "M" (multifetch option), you can set

- a non-zero value in the ISN lower limit field to limit the number of values to be multifetched.
- zero in the ISN lower limit field to multifetch all values.

Refer to the section *Using the Multifetch/Prefetch Feature* for more information.

ISN Quantity: Record Count

Except when the "return ISNs" option (I) is specified, Adabas returns in this field the number of records containing the value returned in the record buffer.

If the prefetch option is specified, the count of prefetched values is in the header that precedes the value in the ISN buffer.

Format Buffer Length

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

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

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

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: Only with Command-Level Multifetch/Prefetch Option

The ISN buffer length (in bytes).

When the command option 1 field specifies "P", the L9 command uses 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: Command-Level Multifetch/Prefetch Option

Specifying one of these options indicates that the (command-level) prefetch or multifetch option is to be used. The multifetch/prefetch option can improve performance reading several descriptor values at a time thereby eliminating the time needed for single-record fetches. For more information, see the section *Using the Multifetch/Prefetch Feature*.

Option	Description
M	(multifetching) In conjunction with the "M" option, you can set <ul style="list-style-type: none"> • a non-zero value in the ISN lower limit field to limit the number of values to be multifetched. • zero in the ISN lower limit field to multifetch all values. The format of the ISN buffer data with the "M" option reflects the standard record descriptor data format.
P	(prefetching) The L9 command puts all prefetched descriptor values in the ISN buffer, preceded by a 16-byte header. The ISN buffer must be large enough to hold the largest descriptor value plus 16 bytes. The actual ISN buffer area defined in the user program must be at least as large as the ISN buffer length specified.

Command Option 2: Return ISNs Option

Option	Description
I	returns the ISNs for each value in the record buffer. The L9 command reads the Associator inverted lists only; no Data Storage access is required.
A	the descriptor's entries are processed in ascending order.
D	the descriptor's entries are processed in descending order.

Additions 1: Descriptor Name

If both the search and value buffer lengths are set to zero, a value in the additions 1 field is the name of the descriptor for which values are to be returned. The name must be the same as the descriptor name specified in the format buffer.

In this case, L9 processes all values for the specified descriptor from the beginning of the file.

The descriptor name is specified in the first two positions of this field. The remaining positions must be set to blanks.

Additions 2: Response Subcodes

If the L9 command returns a nonzero response code, the rightmost two bytes of this field may contain a subcode defining the exact response code meaning. Response codes and their subcodes are defined in the *Adabas Messages and Codes* documentation.

Additions 3: Password

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

Additions 5: Format ID, Global Format ID

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 the leftmost bit is set to 1, the contents of bytes 5 through 8 of the additions 5 field (additions 5 + 4(4)) are interpreted 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'), the value in all eight bytes of the additions 5 field (additions 5 + 0(8)) is 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 ID, Format ID, Global Format ID* for more information and examples.

Buffers

Format Buffer

Note:

Commas separate elements in the format buffer; a period terminates the syntax statement. For more information about format buffer syntax, see the discussion beginning on page .

The format in which the values are to be returned must be specified in this buffer.

The syntax of the format buffer for L9 operation is

```
name [ ,length ] [ ,format ] .
```

where:

- | | |
|--------|--|
| name | is the name of the descriptor for which values are to be returned. A phonetic descriptor or hyperdescriptor may not be specified. A collation descriptor may only be specified if the decode option has been specified in its user exit: the value returned for it is not the index but the original field value. Subdescriptors, superdescriptors and descriptors defined as a multiple-value field may be specified. |
| length | is the length in which the value is to be returned. If length is not specified, the value will be returned using the standard length of the descriptor. |
| format | is the format in which the value is to be returned. The format specified must be compatible with the standard format of the descriptor. If no format is specified, the value will be returned using the standard format of the descriptor. |

Record Buffer

The descriptor value for the descriptor specified in the search and value buffers is returned in this field. A different value is provided with each L9 call. If the descriptor is defined with the null value suppression option, no value for the descriptor will be returned. If the command option 2 field specifies "I", Adabas returns a list of ISNs containing the requested value as well as the value itself in this buffer. The ISNs are provided in ascending sequence.

The descriptor value is provided as follows:

<code>length</code>	<code>value</code>	<code>count</code>	<code>ISN-list</code>
---------------------	--------------------	--------------------	-----------------------

where:

- | | |
|-----------------------|---|
| <code>length</code> | is a one-byte binary value which is the length of the value being provided. If the value has a standard length according to the descriptor type, this field is zero. |
| <code>value</code> | is a descriptor value. |
| <code>count</code> | is the number of values present in the file for the specified descriptor. This number may be returned in one or two bytes. If in one byte, the format is X'cc' where "cc" is the count; if in two bytes, the format is X'8ccc' where "ccc" is the count. |
| <code>ISN-list</code> | <p>If the command option 2 field specified "I", the rest of the record buffer contains ISNs of the records containing this value. One record buffer will contain only the ISNs from one NI (normal index) block. Therefore:</p> <ul style="list-style-type: none"> • the record buffer should be large enough to contain an entire NI block; and • the same value may appear several times with ascending ISNs. |

Search Buffer

Note:

Commas separate elements in the search buffer; a period terminates the syntax statement. For more information about search buffer syntax, see the discussion beginning on page .

If both the search and value buffer lengths are set to zero, a value in the additions 1 field is the name of the descriptor for which values are to be returned. In this case, L9 processes all values for the specified descriptor from the beginning of the file. The search and value buffers are not used.

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.

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 when command option 2 is set to 'A' or the ending value when command option 2 is set to 'D'), the syntax for the search buffer is

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

When two values are provided in the value buffer, the syntax for the search buffer is

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

The elements used in the search buffer syntax statements are as follows:

name	is the name of the descriptor for which values are to be returned. The name must be the same as that specified in the format buffer.								
i	is a one- to three-digit occurrence number subscript appended to the descriptor name if the descriptor is contained within a periodic group and only values within that particular occurrence are to be returned.								
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 on page 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 on page 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: 40px;">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								

Value Buffer

If both the search and value buffer lengths are set to binary zeros, a value in the additions 1 field is the name of the descriptor for which values are to be returned. In this case, L9 processes all values for the specified descriptor in the sequence specified in the command option 2 field: from the beginning (A option) or end (D option) of the file. The search and value buffers are not used.

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 search buffer comparator is GE or GT and a single value is provided in the value buffer, it represents the starting value when command option 2 is set to A or the ending value when command option 2 is set to D.

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.

If a value specified is not present in the file, Adabas finds the next higher or lower value, depending on the command option 2 setting of A or D. In this case, search buffer comparator LE is equivalent to LT and GE is equivalent to GT. If the value specified is not present and no higher (or lower) value is present, response code 3 is returned.

ISN Buffer

The information held in the ISN buffer following an L9 command results from specifying the M (multifetch) or P (prefetch) option in the command option 1 field. The format of the ISN buffer data depends on which option was specified.

Data Format for the Multifetch Option (M)

See section *READ (Lx) Multifetch Processing* for the record descriptor data format.

Data Format for the Prefetch Option (P)

The ISN buffer holds the optionally prefetched descriptor values, each preceded by a 16-byte header. The 16-byte header preceding each value has the following format:

Byte	Usage
1-2	length of descriptor (including this header)
3-4	nucleus response code
5-8	nucleus internal ID
9-12	periodic group occurrence (see the ISN field description)
13-16	record count (see the ISN quantity field description)

Additional Considerations

The following additional considerations are applicable for the L9 command:

1. The command ID used with the L9 command is saved internally and used by Adabas. It is released by Adabas when an end-of-file condition is detected, an RC or CL command is issued, or the Adabas session is terminated. Until it is released, the command ID may not be used for another command.
2. If another user is updating the file being read with an L9 command, it is possible that the user reading with the L9 command will not receive one or more values in the file.
3. 3. Records can be updated in or deleted from a file being read by an L9 command. Adabas attempts to keep track of the last and the next value to be provided to the L9 command, and to provide the correct next value despite any interim update or deletion. However, if the record about to be accessed by the L9 command changes for some reason (it is updated or deleted by another user, for example), the L9 command continues processing as though no change occurred. In other words, a record deleted just before its inverted list entry is accessed by the L9 command is still considered a valid entry by the L9 command.
4. 4. An internal format buffer used by an L9 command must have been created by a previous L9 command. Non-L9 commands cannot use internal format buffers created by L9 commands.

Examples

Example 1

All values for the descriptor RB in file 2 are to be returned.

Control Block

Command Code	L9	
Command ID	L901	a non-blank CID is required
File Number	2	
Format Buffer Length	3	or larger
Record Buffer Length	10	or larger
Search Buffer Length	5	or larger
Value Buffer Length	1	or larger
Additions 3	password	file 2 is security-protected

Buffer Areas

Format Buffer	RB.	the values are to be returned using standard length and format
Search Buffer	RB,1.	the values for descriptor RB are to be returned, and the starting value is being provided with standard format and length equal to 1
Value Buffer	b	processing is to begin with the first value for RB equal or greater than 'b'

Each successive L9 call will result in the return of the next value (values are provided in ascending order). The number of records containing the value is returned in the ISN quantity field.

Example 2

The values for descriptor AB in file 1 are to be returned. Only values which are equal to or greater than 20 are to be returned.

Control Block

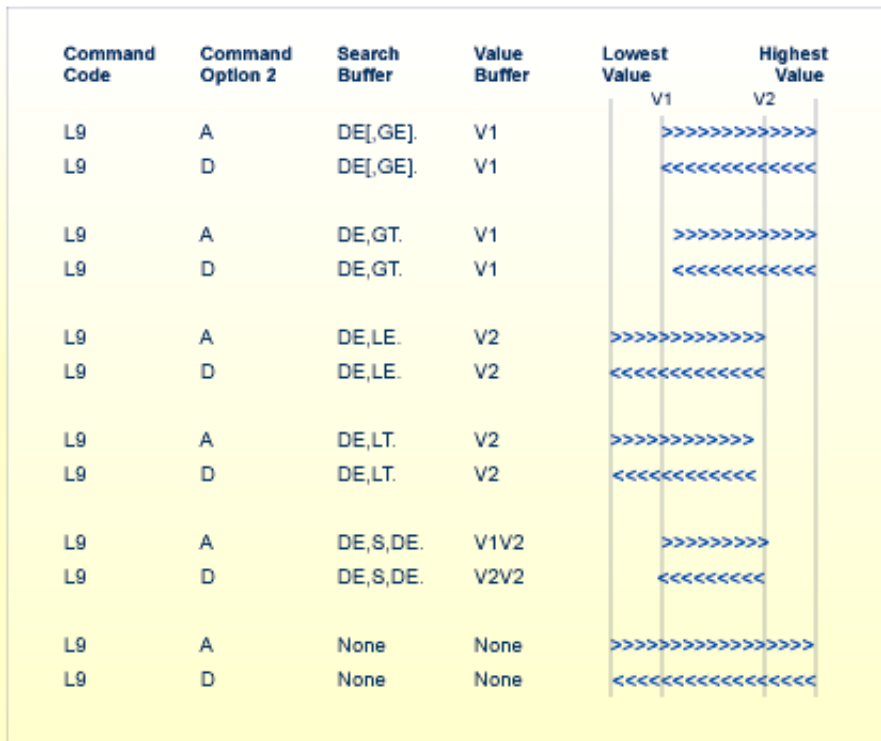
Command Code	L9	
Command ID	L902	a non-blank CID is required
File Number	1	
Format Buffer Length	7	or larger
Record Buffer Length	3	or larger
Search Buffer Length	7	or larger
Value Buffer Length	2	or larger
Additions 3	bbbbbb	file 1 is not security-protected

Buffer Areas

Format Buffer	AB,3,U.	the values are to be returned with length=3 and with format=unpacked
Search Buffer	AB,2,U.	the values for the descriptor AB are to be returned, and the starting value is to be provided as a 2-byte unpacked number
Value Buffer	X'F2F0'	processing is to begin with the first value for AB that is equal to or greater than 20

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:



Overview of Sequence Options