

Record Buffers

A record buffer defines an area in storage to which Adabas can return data or in which you supply data for processing. When a record buffer is required, a corresponding format buffer is expected as well. If a format buffer is not provided, Adabas will create a dummy format buffer (with length zero) to pair with the record buffer. For complete information about the relationships between the different types of ABD or buffer specifications, read *Understanding the Different Buffer Types*.

When using the ACBX direct call interface, multiple record buffers can be specified for an Adabas direct call.

Record buffers are used primarily with read, search, and update commands:

- For read commands, the values of the fields specified in the format buffer are returned by Adabas in the record buffer. They are returned in the order specified by the format buffer. Here is an example of a format buffer and its corresponding record buffer.

Format Buffer	AA, BB	Name of the fields to be updated
Record Buffer	value-AA value-BB	Field values provided by user

Each value is returned in the standard length and format defined for the field unless a length or format override was specified in the format buffer. If the value is a null value, it is returned in the format that is in effect for the field, as follows:

Field Type	Null value represented by . . .
Alphanumeric (A)	blanks (hex '40') or blank of user override encoding
Binary (B)	binary zeros (hex '00')
Fixed (F)	binary zeros (hex '00')
Floating Point (G)	binary zeros (hex '00')
Packed (P)	decimal packed zeros with sign (hex '00' followed by '0A', '0B', '0C', '0D' or '0F' in the rightmost, low-order byte)
Unpacked (U)	decimal unpacked zeros with sign (hex 'F0' followed by 'C0' or 'D0' in the rightmost, low-order byte)
Wide-character (W)	Unicode blanks (hex '20') or blank of user override encoding

Note:

SQL-compatible null values in NC/NN option fields require the additional null value and significance indicator. See *Specifying and Reading the SQL Null Indicator in Record Buffers*, and *SQL Significance Indicator (S)*.

Adabas returns the number of bytes equal to the combined lengths (standard or overridden) of all requested fields.

- For add or update commands, the new values for the fields specified in the format buffer are provided by the user in the record buffer.

Format Buffer	XX, YY	Name of the fields to be updated
Record Buffer	value-XX value-YY	Field values provided by user

When updating a record, you must specify the new value in the record buffer. If a null value is being provided, it must be provided according to the field type in effect, as described above.

- The record buffer is also used to transfer information between the user program and Adabas in the following commands:

Command	Data Provided	Data Returned
OP	Files to update and the operation type (ET, exclusive control) The user's individual time zone	User data (optional)
LF	-	Field definitions for the file
RE	-	User data stored in system file
C5	Protection log user data	-
ET/CL	User data (optional)	-

For the OP command, the record buffer indicates the type of user and the files to be used and optionally, the time zone of the user..

The record buffer is also used for user data (OP, RE, CL, ET commands).

This chapter covers the following topics:

- Specifying and Reading the SQL Null Indicator in Record Buffers
- Specifying Field Lengths of LA (Long Alpha) Fields in Record Buffers
- Specifying Field Lengths of LOB (Large Object) Fields in Record Buffers
- Specifying the Daylight Savings Time Offset in Record Buffers

Specifying and Reading the SQL Null Indicator in Record Buffers

To support Adabas SQL Gateway (ACE) and other structured query languages (SQLs), fields defined with the NC/NN (not-counted/null-not-allowed) options indicate an SQL-significant null with a two-byte binary null indicator in the record buffer.

Whether a field's zero value is significant or an irrelevant null (unspecified) depends on the null indicator specified in the record buffer when the value is entered or changed, or returned in the record buffer when the value is read.

In addition to specifying or reading the value itself, either:

- set the null indicator into the record buffer position that corresponds to the field's designation in the format buffer for an update operation, or
- ensure that your program examines the null indicator (if any) returned in the record buffer position corresponding to the field's position in the format buffer for a read operation.

The null indicator is always two bytes long and has fixed-point format, regardless of the data format.

For a read (Lx) or find with read (Sx with format buffer entry) command, the null indicator value returns one of the following (hexadecimal) null indicator values, according to the actual value that the selected field contains:

Hex Value	Description
FFFF	A null value in the field is not significant.
0000	A null value in the field is a significant value; that is, a true zero or blank.
xxxx	The field is truncated. The null indicator value contains the length (xxxx) of the entire value as stored in the database record if the length is less than 32,768.
0001	The field is significant and the value is truncated, and the length of the value does not fit into the S element because it is greater than 32,767.

For an update (Ax) or add (Nx) command, the (hexadecimal) null indicator value in the record buffer must be set to one of the following values:

Hex Value	Description
FFFF	The field value is set to "undefined", an insignificant null; the field's contents in the record buffer are irrelevant when set to binary zero or blank characters.
0000	If either no value is specified in the record buffer, or binary zero or blanks are specified, the field contains a significant null value.

For an *add* command, if no value for the field is supplied in the record buffer for a field defined with the NC option, the field is treated as a null field. The following example shows how a null would be represented in a two-byte Adabas binary field AA defined with the NC option:

Field definition: 01,AA,2,B,NC

	For a nonzero value	For a blank	For null
Null Value indicator in Record Buffer	0 (binary value is significant)	0 (binary null is significant)	FFFF (binary null is not significant)
Data	0005	0000 (zero)	not relevant
Adabas internal representation	0205	0200	C1

For an *update* (A1/N1) command, the field value is always significant whenever the field is defined with the NC option; the field is treated as if a hexadecimal null indicator value of "0000" has been specified.

For a *read* command, if the null indicator is not specified for an NC option field, the field value is returned in the record buffer whenever there is a significant value in the record. If the Data Storage record contains a "not significant" (FFFF) indicator value for the field, response code 55 (ADARSP055) will be returned when the record is read.

Specifying Field Lengths of LA (Long Alpha) Fields in Record Buffers

The LA option is normally used with variable-length data. If the length is not explicitly specified in the format buffer then the length of an alphanumeric field with the LA option can be specified in the record buffer or will be returned by Adabas.

The field value is preceded by a two-byte length field containing the length of the value, plus 2 (inclusive length).

Format Buffer	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin-bottom: 10px;">BA,</div> or <div style="border: 1px solid black; padding: 5px; width: fit-content; margin-bottom: 10px;">BA,0,</div>	Name of the fields to be read without explicit length specifications.
Record Buffer	X'0005C1C1C1'... or X'2712...(10,000 character	Field values specified for Adabas or returned by Adabas in hexadecimal. In the first case the value is three bytes long; in the second case the value is 10,000 bytes long.

Specifying Field Lengths of LOB (Large Object) Fields in Record Buffers

The LB option is normally used with variable-length data. If the length is not explicitly specified in the format buffer then the length of an alphanumeric field with the LB option can be specified in the record buffer or will be returned by Adabas.

The field value is preceded by a four-byte length field containing the length of the value, plus 4 (inclusive length).

Format Buffer	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin-bottom: 10px;">L1,</div> or <div style="border: 1px solid black; padding: 5px; width: fit-content; margin-bottom: 10px;">L1,0,</div>	Name of the fields to be read or updated without explicit length specifications.
Record Buffer	X'00000007C1C1C1'... or X'000186A4...(100,000 character	Field values specified for Adabas or returned by Adabas in hexadecimal. In the first case the value is three bytes long; in the second case the value is 100,000 bytes long.

Specifying the Daylight Savings Time Offset in Record Buffers

The daylight savings offset from standard time (in seconds) is specified in the record buffer associated with the format buffer in the following format:

`H' nnnn'`

The value *nnnn* represents the number of seconds that the stored time should be offset from standard time to calculate daylight savings time. A value of zero indicates that standard time should be used; any value other than zero indicates that daylight savings time should be adjusted for and specifies the offset for that adjustment.

Suppose the definition of field AA in the FDT is:

```
"1,AA,14,U,TZ,DT-E(DATETIME)"
```

A valid format buffer might be:

```
"AA,14,U,AAD,2,F."
```

The corresponding record buffer might be:

```
"20080814120000",H'3600'
```

In this example the daylight savings offset for field AA is 1 hour (3600 seconds) from standard time.

Read *Daylight Savings Indicator (D)* for more information about daylight savings indicator usage in format and search buffers.