

Adabas Buffer Descriptions (ABDs)

If an Adabas call using the ACBX interface is made that requires buffer specifications, Adabas buffer descriptions (ABDs) *must* be used. ABDs *must not* be used when specifying an Adabas call using the classic ACB interface; if an Adabas call using the ACB interface is made that requires buffer specifications, specify the buffers or pointers to the buffers directly in the Adabas call itself. For more information about the ACBX and ACB interface direct calls, read *Calling Adabas*.

Since Adabas 8, through its ACBX interface, supports segmented buffers (multiple pairs of format and record buffers, or multiple triplets of format, record, and multifetch buffers), the total number of buffers in an ACBX call is not fixed and limited. The individual buffers are no longer described by fields in the ACBX itself (in the way the buffer lengths are defined in the ACB); instead, each buffer has its own Adabas buffer description (ABD) structure that describes what kind of buffer it is, where it is located, what size it is, and other pertinent information.

In mainframe system applications, the addresses of ABDs are specified directly in the Adabas call; in open system applications, the addresses of ABDs are specified in the ABD list associated with the call.

This chapter describes the structure of an ABD and ABD lists. For information on the defining the buffers themselves, read *Defining Buffers*.

- Available ABD Types
 - ABD Structure
 - ABD Field Descriptions
 - ABD DSECT
 - ABD Lists
-

Available ABD Types

Using ABDs in an ACBX interface direct call, the buffers used in a direct call can be contiguous or discontinuous. With Adabas 8, you can define ABDs for eight different types of buffers:

- Format buffers
- Record buffers
- Multifetch buffers
- Search buffers
- Value buffers
- ISN buffers
- User buffers

- Performance buffers (reserved for use by Adabas Review only).

Each Adabas buffer segment is represented by a single ABD, although you can define multiple ABDs of a given type in the same program. Offset 4 (ABDID) in each ABD identifies the type of buffer defined by the ABD.

In an ACBX interface call, there is a one-to-one correspondence between ABD and buffer specifications; each buffer you want to specify must have a corresponding ABD. The buffer can be specified in the ABD itself or referenced by indirect reference.

ABDs can be specified in any sequence in an ACBX interface direct call. However, if an ABD requires a matching ABD of another type, Adabas will match them sequentially. For example, if three format buffer ABDs and three record buffer ABDs are included in the call, the first format buffer ABD in the call is matched with the first record buffer ABD in the call, the second format buffer ABD is matched with the second record buffer ABD, and the third format buffer ABD is matched with third record buffer ABD.

If unequal numbers of match-requiring ABDs are specified, Adabas will generate a dummy ABD (with a buffer length of zero) for the missing ABD. For example, if three format buffer ABDs are specified, but only two record buffer ABDs are specified, a dummy record buffer ABD is created for use with the third format buffer ABD. If you would prefer that the dummy record buffer ABD be used for the second format buffer ABD instead, you must specify the dummy record buffer ABD yourself prior to the record buffer ABD to be used by the third format buffer ABD.

For commands where data in the record buffer is *not* described by a format specification in the format buffer, no format buffer segments need be specified; if any are specified, they are ignored. This applies to only a few commands; the most prominent of them is OP.

For information about the relationships between different buffer types, read *Understanding the Different Buffer Types*.

ABD Structure

The following table describes the structure of the ABD.

DSECT Field Name	Field	Control Block Position	Offset	Length (in bytes)	Format
ABDXLEN	ABD length	1-2	00	2	binary
ABDXVER	Version indicator	3-4	02	2	binary
ABDXID	Buffer Type ID	5	04	1	alphanumeric
ABDXRSV1	Reserved 1	6	05	1	binary
ABDXLOC	Buffer location flag	7	06	1	alphanumeric/binary
ABDXRSV2	Reserved 2	8	07	1	binary
ABDXRSV3	Reserved 3	9	08	4	binary
ABDXALET	ALET for buffer (if ABDXLOC=C'D')	13	0C	4	binary
ABDXSIZE	Buffer size (allocated length)	17-24	10	8	binary
ABDXSEND	Data length to send from client to the nucleus	25-32	18	8	binary
ABDXRECV	Data length received by the client from the nucleus	33-40	20	8	binary
ABDXADRG	64-bit indirect address pointer (if ABDXLOC=C'I' or C'D')	41-44	28	4	binary
ABDXADR	34-bit indirect address pointer (if ABDXLOC=C'I' or C'D')	45-48	2C	4	alphanumeric
---	Buffer (if ABDXLOC=C' ' or X'00')	49-n	30	user-defined	not applicable

ABD Field Descriptions

Each of the fields in the ABD is described in this section, in the order they appear in the ABD structure.

ABD Length (ABDXLEN)

Required. Use this field to specify the length of the ABD. Currently, the value of this field must be 48.

Version Indicator (ABDXVER)

Required. This field identifies the version of the ABD structure. A value of C'G2' in this field indicates that the buffer definition is in the new, extended ABD structure.

Buffer Type ID (ABDXID)

Required. Use this field to identify the type of buffer described by the ABD, as shown in the following table:

ID Setting	Type of Buffer
C'F'	Format
C'I'	ISN
C'M'	Multifetch
C'P'	Performance (reserved for use by Adabas Review only)
C'R'	Record
C'S'	Search
C'U'	User
C'V'	Value

Reserved 1 (ABDXRSV1)

This field is reserved and must be set to binary zeros.

Buffer Location Flag (ABDXLOC)

Required. Use this field to identify whether the location of the buffer is defined at an indirect address, is qualified by an ALET, or is defined at the end of the ABD itself.

If this field is set to "I" (C'I'), Adabas assumes indirect addressing is specified and will use the address specified in the indirect address pointer field (ABDADDR or ABDXADRG). In this case the buffer must reside in 31-bit or 64-bit addressable storage, in the current primary address space.

If this field is set to "D" (C'D'), Adabas assumes an ALET has been provided in ABDXALET and indirect addressing is implied. Adabas will use the address specified in the indirect address pointer field (ABDXADDR or ABDXADRG). In this case, the buffer must reside in 31-bit or 64-bit addressable storage, addressable using the ALET.

If this field is blank (C' ') or contains hexadecimal zeros, the buffer must immediately follow the ABD in the primary address space.

Reserved 2 (ABDXRSV2)

This field is reserved and must be set to binary zeros.

Reserved 3 (ABDXRSV3)

This field is reserved and must be set to binary zeros.

ALET (ABDXALET)

If ABDXLOC is set to "D" (C'D'), this field must contain an ALET suitable to access the buffer. The ALET must be on the dispatchable unit access list (DU-al) and may relate to a dataspace or an address space.

Some special ALET values are supported. A value of zero ("0") specifies the current primary address space (PASID) and is equivalent to specifying ABDXLOC=C'I'. A value of "1" specifies the current secondary address space (SASID) and is not allowed, resulting in response code 253 (ADARSP253), subcode 14. Under z/OS, a value of "2" specifies the home address space (HASID) and is equivalent to specifying ABDXLOC=C'I' if PASID=HASID when the command is issued.

Buffer Size (ABDXSIZE)

Required. Use this field to specify the size of the buffer (in bytes), as it is allocated. A size of zero indicates a dummy buffer, which is treated as if it was not specified at all. More than 32 KB of data can now be specified in an Adabas buffer.

Data Length to Send (ABDXSEND)

Required. Use this field to specify the length of the data (in bytes) to be sent to Adabas. The maximum value of this field cannot exceed the value set for the buffer size field (ABDXSIZE). A buffer is sent to Adabas only if it is an input buffer for the type of command being issued.

Note:

At this time, you must specify the same value for this field as you specify for the maximum buffer size (ABDXSIZE field). This is a temporary limitation of Adabas 8, that will be resolved in a future release.

Data Length Received (ABDXRECV)

This field specifies the length of the data (in bytes) returned to Adabas. The Adabas router sets this value at the end of call processing. The maximum value of this field will not exceed the value set for the buffer size field (ABDXSIZE). A buffer is received from Adabas only if it is an output buffer for the type of command being issued.

64-Bit Indirect Address Pointer (ABDXADRG)

If you set the buffer location flag field (ABDLOC) to C'I' (indirect buffer) or to C'D' (ALET-qualified buffer), specify the 64-bit address of the actual buffer in this 8-byte field.

31-Bit Indirect Address Pointer (ABDXADR)

This four-byte field is an alias for the second four bytes of ABDXADRG (the 64-bit address pointer). If you set the buffer location flag field (ABDLOC) to C'I' (indirect buffer) or to C'D' (ALET-qualified buffer), specify the 31-bit address of the actual buffer in this field and ensure that the preceding four bytes (in ABDXADRG) are zero.

Actual Buffer

If you set the buffer location flag field (ABDLOC) to C' ' (blanks), this field should contain the actual buffer. For complete information on defining buffers, read *Defining Buffers*.

ABD DSECT

The ABD DSECT can be found in member ADABDX of the distributed Adabas SRCE library.

ABD Lists

An *ABD list* is a file containing a list of pointer references to the Adabas buffer descriptions (ABDs) used for a direct call. ABD lists are used only for open systems ACBX direct calls. In the list, one ABD pointer is required for every buffer segment that is needed for the direct call.

ABD lists can include pointers to the ABDs for eight different types of buffers: format, record, multifetch, search, value, ISN, user, and performance buffers. Multiple ABDs of the same type can be specified in an ABD list.

ABDs can be specified in the list in any sequence. However, if an ABD requires a matching ABD of another type, Adabas will match them sequentially. For example, if three format ABDs and three record ABDs are included in the list, the first format ABD in the list is matched with the first record ABD in the list, the second format ABD is matched with the second record ABD, and the third format ABD is matched with third record ABD. If unequal numbers of match-requiring ABDs are listed (for example, if three format ABDs are listed, but only two record ABDs), Adabas will generate a dummy ABD for the missing ABD (in this case a dummy record ABD will be created).

For complete information about the relationships between the different types of ABD or buffer specifications, read *Understanding the Different Buffer Types*.