

# A1 Command: Update Record

The A1 command updates records with a hold option.

This chapter covers the following topics:

- Function and Use
  - ACB Interface Direct Call: A1 Command
  - ACBX Interface Direct Call: A1 Command
  - Buffers
  - Additional Considerations
- 

## Function and Use

The A1 command is used to change the value of one or more fields in a record. The record containing the field (or fields) to be updated is identified by the file number in which it is contained and its ISN. Specify the fields to be updated in the format buffer and provide the updating values for these fields in the record buffer. Only the fields specified are modified. All other fields in the record remain unchanged.

All necessary updating to the Associator and Data Storage is performed by Adabas.

A hold option is available to place the record in hold status before it is updated.

If the user is operating in multiuser mode, the A1 command will be executed only if the record to be updated is in hold status for the user.

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

**Note:**

The A4 command from previous Adabas releases is executed as an A1 command.

## ACB Interface Direct Call: A1 Command

This section describes ACB-interface direct calls for the A1 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	alphanumeric / binary	F	U
File Number	9-10	binary	F	U
Response Code	11-12	binary	--	A
ISN	13-16	binary	F	U
ISN Lower Limit	17-20	binary	--	A <sup>1</sup>
ISN Quantity	21-24	binary	--	A <sup>1</sup>
Format Buffer Length	25-26	binary	F	U
Record Buffer Length	27-28	binary	F	U
	29-34	--	--	--
Command Option 1 / 2	35-36	alphanumeric	F	U
	37-44	--	--	--
Additions 2	45-48	alphanumeric / 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

### Notes

1. These fields are used and not reset by Adabas if coupled files are used.

### Buffer Areas

Buffer	Before Adabas Call	After Adabas Call
Format	F	U
Record	F	U

where:

F	Supplied by user before Adabas call
A	Supplied by Adabas
U	Unchanged after Adabas call
--	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)

A1

### Command ID (ACBCID)

If a series of records is to be updated by using a series of A1 calls, and the same fields are specified in the format buffer for each call (such as when updating a set of records resulting from a find command), this field should be set to a non-blank, non-zero value. If the A1 command is used in conjunction with an L1/L4, L2/L5, or L3/L6 command, and the same fields within each record are read and updated, the same command ID used for the read command should be used for the A1 calls. In both cases, this reduces the time needed to process each successive A1 call.

If only a single record is to be updated with a single A1 call, or the format buffer is modified between A1 calls, this field should be set to blanks.

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

### 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 (ADARSP000) 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.

### ISN (ACBISN)

The ISN of the record to be updated.

### ISN Quantity/Lower Limit (ACBISQ and ACBISL)

These fields are set to nulls following completion of the A1 operation unless coupled files or partial large object (LOB) fields are used.

If coupled files are used, these fields are used by A1 processing and are not reset.

If a partial LOB field is requested and the Command Option 2 field is set to L, the ACBISL field is used to keep track of the current position in the LOB value when multiple A1 calls are made for the field.

### Format Buffer Length (ACBFBL)

The format buffer length (in bytes). The format buffer area defined in the user program must be as large as (or larger than) the length specified.

### Record Buffer Length (ACBRBL)

The record buffer length (in bytes). The record buffer area defined in the user program must be as large as (or larger than) the length specified.

### Command Option 1 and Command Option 2 (ACBCOP1 and ACBCOP2)

Option	Description
H	An "H" in either the Command Option 1 or Command Option 2 field places the record in hold status before it is updated. If the record is currently being held by another user, the command is placed in wait status until either the record becomes available or the transaction times out-unless option "R" is also specified.
L	An L in the Command Option 2 field indicates that the position within a large object (LOB) field be tracked in the ACBISL (ISN lower limit) field. This option is useful when multiple A1 commands are being issued in a series for a LOB field. This option can only be used if a LOB field is specified in the format buffer using LOB segment notation with asterisk notation in the <i>bytenum</i> specification.
R	If this option is specified, it must be specified in the Command Option 1 field and option "H" must be specified in the Command Option 2 field. Option "R" causes Adabas to return response code 145 (ADARSP145) if the record to be held is not available. The command is <i>not</i> placed in wait status.

### Additions 2 - Length of Compressed 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;

- If the A1 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 defined 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.

If the accessed file is password protected, Adabas sets this field to blanks during command processing to protect the integrity of the password.

### **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)**

This field may be used to provide a separate format ID to identify the internal format buffer used for this command, or to provide a global format ID.

As long as the high order bit of the first byte of the Additions 5 field is not set to 1, the value provided in the command ID field will be 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 ID, Format ID, Global Format ID* for more information and examples.

## **ACB Examples**

For the Adabas file definitions used in all the examples in this section, see *File Definitions Used in Examples*.

- Example 1
- Example 2

### **Example 1**

ISN 4 of file 1 is to be updated with the following values:

Field AA 1234

Field AB 20

**Control Block**

<b>Command Code</b>	A1	
<b>Command ID</b>	<i>bbbb</i> (blanks)	only 1 record is to be updated
<b>File Number</b>	1	
<b>ISN</b>	4	
<b>Format Buffer Length</b>	10	or larger
<b>Record Buffer Length</b>	10	or larger
<b>Additions 3</b>	<i>bbbbbbb</i> (blanks)	file is not security protected
<b>Additions 4</b>	<i>bbbbbbb</i> (blanks)	file is not ciphered

**Buffer Areas**

<b>Format Buffer</b>	AA,AB,2,U.
<b>Record Buffer</b>	X'F1F2F3F4404040F2F0'

**Example 2**

A set of records (previously identified with a FIND command) in file 2 are to be updated with the following values:

Field RA ABCD

Field XB 80

Field XC 0

**Control Block**

<b>Command Code</b>	A1	
<b>Command ID</b>	ABCD	a non-blank, non-zero command ID is recommended because the same fields in a series of records are being updated
<b>File Number</b>	2	
<b>ISN</b>	n	each ISN resulting from the previous FIND command will be inserted in this field before each call
<b>Format Buffer Length</b>	9	or larger
<b>Record Buffer Length</b>	16	or larger
<b>Additions 3</b>	password	file 2 is security protected
<b>Additions 4</b>	<i>bbbbbbb</i> (blanks)	file is not ciphered

**Buffer Areas**

<b>Format Buffer</b>	RA, XB, XC.
<b>Record Buffer</b>	X'C1C2C3C4404040080CF0F0F0F0F0F0'

The A1 call is repeated for each ISN which resulted from the previous find command.

## ACBX Interface Direct Call: A1 Command

This section describes ACBX-interface direct calls for the A1 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	binary	---	---
Version Indicator	3-4	binary	F	U
	5-6	binary	---	---
Command Code	7-8	alphanumeric	F	U

Field	Position	Format	Before Adabas Call	After Adabas Call
	9-10	binary	---	---
Response Code	11-12	binary	---	U
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	U
	33-36	---	---	---
ISN Lower Limit	37-40	binary	---	A <sup>1</sup>
	41-44	---	---	---
ISN Quantity	45-48	binary	---	A <sup>1</sup>
Command Option 1	49	alphanumeric	F	U
Command Option 2	50	alphanumeric	F	U
	51-64	---	---	---
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
	117-128	---	---	---
Compressed Record Length	129-136	binary	---	A
Decompressed Record Length	137-144	binary	---	A
Command Time	145-152	binary	---	A
User Area	153-168	not applicable	---	U
	169-193	---	---	---



## Notes

1. These fields are used and not reset by Adabas if coupled files are used.

## ABDs and Buffers

ABD and Buffer	Before Adabas Call	After Adabas Call
Format	F	U
Record	F	U

where:

- F        Supplied by user before Adabas call
- A        Supplied by Adabas
- U        Unchanged after Adabas call
- 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)

A1

### 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)

If a series of records is to be updated by using a series of A1 calls, and the same fields are specified in the format buffer for each call (such as when updating a set of records resulting from a find command), this field should be set to a non-blank, non-zero value. If the A1 command is used in conjunction with an L1/L4, L2/L5, or L3/L6 command, and the same fields within each record are read and updated, the same command ID used for the read command should be used for the A1 calls. In both cases, this reduces the time needed to process each successive A1 call.

If only a single record is to be updated with a single A1 call, or the format buffer is modified between A1 calls, this field should be set to blanks.

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

**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 (ACBXISN)**

Use this field to specify the ISN of the record to be updated.

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.

**ISN Lower Limit (ACBXISL)**

These fields are set to nulls following completion of the A1 operation unless coupled files or partial large object (LOB) fields are used.

If coupled files are used, these fields are used by A1 processing and are not reset.

If a partial LOB field is requested and the Command Option 2 field is set to L, the ACBXISL field is used to keep track of the current position in the LOB value when multiple A1 calls are made for the field.

**ISN Quantity (ACBXISQ)**

This field is set to nulls following completion of the A1 operation, unless hard-coupled files are used. If coupled files are used, this field is used during A1 operation and is not reset.

**Command Option 1 and Command Option 2 (ACBXCOP1 and ACBXCOP2)**

Option	Description
H	An "H" in either the Command Option 1 or Command Option 2 places the record in hold status before it is updated. If the record is currently being held by another user, the command is placed in wait status until either the record becomes available or the transaction times out-unless option "R" is also specified.
L	An L in the Command Option 2 field indicates that the position within a large object (LOB) field be tracked in the ACBXISL (ISN lower limit) field. This option is useful when multiple A1 commands are being issued in a series for a LOB field. This option can only be used if a LOB field is specified in the format buffer using LOB segment notation with asterisk notation in the <i>bytenum</i> specification.
R	If specified, this setting must be specified in the Command Option 1 field and option "H" must be specified in the Command Option 2 field. Option "R" causes Adabas to return response code 145 (ADARSP145) if the record to be held is not available. The command is <i>not</i> placed in wait status.

### 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.

If the accessed file is password protected, Adabas sets this field to blanks during command processing to protect the integrity of the password.

### 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)

This field may be used to provide a separate format ID to identify the internal format buffer used for this command, or to provide a global format ID.

As long as the high order bit of the first byte of the Additions 5 field is not set to 1, the value provided in the command ID field will be 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 ID, Format ID, 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.

### **Compressed Record Length (ACBXLCMP)**

This field returns the compressed record length when a record was read or written. This is the length of the compressed data processed by the successful Adabas call. If the logical data storage record spans multiple physical data records, the combined length of all associated physical records may not be known. In this case, Adabas returns high values in the low-order word of this field.

### **Decompressed Record Length (ACBXLDEC)**

This field returns the decompressed record length. This is the length of the decompressed data processed by the successful call. If multiple record buffer segments are specified, this reflects the total length across all buffer segments.

## **Buffers**

The following buffers should be specified with an A1 command:

- Format Buffer
- Record Buffer

### **Format Buffer**

The fields to be updated must be specified in this buffer. When performing an A1 command, the format buffer *cannot* contain any of the following:

- Field selection criteria;
- An edit mask element;
- A reference to a subdescriptor or superdescriptor field;
- The same field specified more than once (except a multiple-value field);
- An "-N" type of specification for an multiple-value field, or for fields in a periodic group (for example, ABN or AB1-N).

Any of the above in the format buffer will cause a nucleus response of 44 for an A1 command.

Descriptions of the syntax and examples of format buffer construction are provided in *Format Buffers*.

## Record Buffer

The values to be used for updating are provided in this buffer according to the length and format as specified in the format buffer. For more information, read *Record Buffers*.

## Additional Considerations

The following additional considerations are applicable for the A1 command:

1. Subdescriptors, superdescriptors, and phonetic descriptors may not be directly updated. To update any of these descriptors, the fields used to derive them must be updated. All corresponding subdescriptor, superdescriptor, or phonetic descriptor values will then be updated by Adabas.
2. Theoretically, the maximum record length permitted is 32767 bytes before compression. The actual maximum is limited by block size restrictions. It is also smaller depending on the size of the LU parameter specified for the Adabas session; the maximum is (LU - format buffer length - 108). The maximum record length after compression is equal to the smaller of either the Data Storage block size - 4 bytes, or the Work block size - 110 bytes.
3. A descriptor value cannot be larger than 253 bytes.
4. If a field is updated using a length override that exceeds the standard length (not permitted if the field is defined with the Fixed Storage option), all subsequent references to this field should specify the length that was used. If a subsequent reference uses the standard length, value truncation of an alphanumeric fields, a response code 55 (ADARSP055) for a numeric field may occur.
5. A multiple-value field or a field within a periodic group may be specified more than once in the format buffer. A multiple-value field may be specified without an index several times, or with different indices. A periodic field must always have different indices.
6. A multiple-value or periodic group count field specified in the format buffer will be ignored by Adabas. The corresponding value in the record buffer will also be ignored. A literal in the format buffer will be ignored by Adabas, and the corresponding positions in the record buffer will also be ignored.
7. If a multiple-value field is updated and the field count must be changed, Adabas updates the multiple-value field count according to the following rules:
  - For a multiple-value field defined with the null suppression (NU) option, the count field is adjusted to reflect the current number of existing non-null values. Null values are completely suppressed.

<b>Field Definition</b>	01,MF,5,A,MU,NU
<b>MF values before update</b>	XXXXX,YYYYY
<b>Format Buffer</b>	MF4
<b>Record Buffer</b>	<i>ZZZZZ</i>
<b>Result after update</b>	XXXXX,YYYYY, <i>ZZZZZ</i> MF count = 3
<b>MF values before update</b>	XXXXX,YYYYY, <i>ZZZZZ</i>
<b>Format Buffer</b>	MF2
<b>Record Buffer</b>	<i>bbbb</i> (blanks)
<b>Result after update</b>	XXXXX, <i>ZZZZZ</i> MF count = 2
<b>MF values before update</b>	XXXXX, <i>ZZZZZ</i>
<b>Format Buffer</b>	MF1-2
<b>Record Buffer</b>	<i>bbbbbbbb</i> (blanks)
<b>Result after update</b>	Values suppressed MF count = 0

- For a multiple-value field defined without the NU option, the count is adjusted to reflect the current number of existing values (including null values).

<b>Field Definition</b>	01,MF,5,A,MU
<b>MF values before update</b>	XXXXX,YYYYY
<b>Format Buffer</b>	MF4
<b>Record Buffer</b>	DDDD
<b>Result after update</b>	XXXXX,YYYYY,b(blank),DDDD MF count = 4
<b>MF values before update</b>	XXXXX,YYYYY, <i>ZZZZZ</i>
<b>Format Buffer</b>	MF3
<b>Record Buffer</b>	<i>bbbb</i> (blanks)
<b>Result after update</b>	XXXXX,YYYYY,b (blank) MF count = 3

A maximum of 191 values is permitted for a multiple-value field.

An exception to the rules is when the format buffer contains a multiple-value field *without an index specification*. In this case, only the new values specified are used and all other values are deleted regardless of the content of the value.

<b>Field Definition</b>	01,MF,5,A,MU,NU
<b>MF values before update</b>	XXXXX,YYYYY
<b>Format Buffer</b>	MF
<b>Record Buffer</b>	AAAAA
<b>Result after update</b>	AAAAA MF count = 1
<b>MF values before update</b>	XXXXX,YYYYY
<b>Format Buffer</b>	MF1
<b>Record Buffer</b>	AAAAA
<b>Result after update</b>	AAAAA,YYYYY MF count = 2
<b>MF values before update</b>	XXXXX,YYYYY, <u>ZZZZZ</u>
<b>Format Buffer</b>	MF
<b>Record Buffer</b>	<i>bbbb</i> (blanks)
<b>Result after update</b>	value suppressed MF count = 0

8. If one or more fields contained in a periodic group are updated, Adabas updates the periodic group count, if necessary, according to the following rule:

The count is adjusted to reflect the highest occurrence number referenced in the format buffer (provided that this occurrence is higher than the existing highest occurrence number).

<b>Field Definitions</b>	01,GB,PE 02,BA,1,B,DE,NU 02,BB,5,P,NU
<b>GB values before update</b>	GB (1st occurrence) BA = 5 BB = 20 GB (2nd occurrence) BA = 6 BB = 25 GB count = 2
<b>Format Buffer</b>	GB4
<b>Record Buffer</b>	X'08000000500F'
<b>Result after update</b>	GB (1st occurrence) BA = 5 BB = 20 GB (2nd occurrence) BA = 6 BB = 25 GB (3rd occurrence) BA = 0 BB = 0 GB (4th occurrence) BA = 8 BB = 500 GB count = 4
<b>GB values before update</b>	GB (1st occurrence) BA = 5 BB = 20 GB (2nd occurrence) BA = 6 BB = 25 GB count = 2
<b>Format Buffer</b>	GB1
<b>Record Buffer</b>	X'00000000000F'
<b>Result after update</b>	GB (1st occurrence) BA = 0 BB = 0 GB (2nd occurrence) BA = 6 BB = 25 GB count = 2

A maximum of 191 occurrences is permitted for a periodic group.

- If a field defined with variable length (no standard length) is specified in the format buffer, the corresponding value in the record buffer must be preceded by a one-byte binary length value that includes the length byte itself.

<b>Field Definitions</b>	01,AA,3,A 01,AB,A
<b>Format Buffer</b>	AA,AB
<b>Record Buffer</b>	X'F1F2F306F1F2F3F4F5'



Fields AA and AB are to be updated. The new value for AA is "123". The new value for AB (which is a variable-length field) is "12345".