

UPDATE: Add/Delete Records



Warning:

If ADALOD UPDATE ends abnormally (due to insufficient space, for example), updates made to the file before the abnormal ending cannot be backed out. Software AG therefore recommends that you perform ADASAV SAVE on the file before you run ADALOD UPDATE.

The UPDATE function adds or deletes a large number of records (ISNs) to or from an existing file. A single UPDATE operation can both add and delete ISNs.

Records to be added must be in compressed (ADACMP or ADAULD output) form and be in the DDEBAND/EBAND input data set.

ISNs to be deleted must be specified by either or both of the DDISN and DELISN parameters.

Notes:

1. The UPDATE function cannot be used with an Adabas system file if the Adabas nucleus is active, and cannot be used to change the checkpoint or security files.
2. A multiclient file cannot be made part of an expanded file, and an expanded file cannot be converted to a multiclient file.

```

ADALOD UPDATE  FILE = file-number
                 SORTSIZE = size
                 TEMPSIZE = size
                 [DDISN]
                 [DELISN = isn-list ]
                 [DSREUSE = { YES | NO } ]
                 [ETID = multiclient-file-owner-id ]
                 [ISNREUSE = { YES | NO } ]
                 [LIP = { isn-pool-size | 2000 } ]
                 [LWP = { work-pool-size | 1048576 } ]
                 [MAXISN = number
                 [ACRABN = starting-rabn ]
                 [ASSOVOLUME = ' Associator-extent-volume ' ] ]
                 [MAXISN2 = number
                 [AC2RABN = starting-rabr ]
                 [ASSOVOLUME = ' Associator-extent-volume ' ] ]
                 [NOUSERABEND]
                 [NUMREC = number ]
                 [PASSWORD = ' password ' ]
                 [RESTART]
                 [RPLLOAD = { YES | NO } ]
                 [SKIPREC = { number | 0 } ]
                 [SORTDEV = device-type ]
                 [TEMPDEV = device-type ]
                 [TEST]
                 [USERISN = { YES | NO } ]

```

This chapter covers the following topics:

- Essential Parameters
 - Optional Parameters and Subparameters
 - Examples
 - Formats for Specifying ISNs
 - UPDATE Data and Space Requirements
 - Mass Updates of Expanded Files
-

Essential Parameters

FILE: File Number

FILE specifies the number of the file to be updated. If a component file of an Adabas expanded file is specified, only that component file is updated; the other component files must be updated in separate UPDATE operations.

SORTSIZE: Sort Size

SORTSIZE is the number of blocks or cylinders available for the sort data set.

TEMPSIZE: Temporary Storage Size

TEMPSIZE is the number of blocks or cylinders available for the temp data set.

Optional Parameters and Subparameters

ACRABN: Starting RABN for Address Converter

ACRABN causes additional space allocation for the address converter to begin at the specified RABN. ACRABN is effective only if MAXISN specifies an increase for the file's address converter.

AC2RABN: Starting RABN for Secondary Address Converter

AC2RABN causes additional space allocation for the secondary address converter to begin at the specified RABN. AC2RABN is effective only if MAXISN2 specifies an increase for the file's secondary address converter. The secondary address converter is used to map the secondary ISNs of secondary spanned records to the RABNs of the Data Storage blocks where the secondary records are stored. There is no default value.

ASSOVOLUME: Associator Extent Volume

Note:

The value for ASSOVOLUME must be enclosed in apostrophes.

ASSOVOLUME is effective only if MAXISN or MAXISN2 specify an increase for the file's address converter.

ASSOVOLUME specifies the volume on which the file's address converter extents is to be allocated. If the requested number of blocks cannot be found on the specified volume, ADALOD retries the allocation while disregarding the ASSOVOLUME parameter.

If ACRABN or AC2RABN is specified, ADALOD ignores the ASSOVOLUME value when allocating the address converter extent type. If ASSOVOLUME is not specified, the file's Associator space is allocated according to ADALOD's default allocation rules.

DDISN: Read ISNs to be Deleted from Sequential Data Set

If DDISN is specified, ISNs to be deleted are read from the DDISN/ISN sequential data set. If both the DDISN and DELISN parameters are specified, the ISNs from the two lists are merged. The DDISN/ISN data set must have variable or variable blocked records. See the section Formats for Specifying ISNs for more information.

When the UPDATE function is executed, all ISNs are first read and stored in the ISN pool in the order they occur. The size of the ISN pool (specified by LIP) must be large enough to store all data read from DDISN/ISN. The records are then sorted in ascending order. Overlapping ranges and duplicate ISNs are not allowed. ISNs not found during processing are ignored.

When deleting ISNs from an Adabas expanded file, you can specify the complete ISN list for all component files; the UPDATE function automatically selects only the ISNs that are appropriate for the component file being processed.

Note:

This parameter cannot be specified in an ADALOD UPDATE operation on a *LOB file*.

DELISN: ISNs to be Deleted

DELISN specifies a list of the ISNs of records to be deleted. If both DDISN and DELISN are specified, the ISNs from the two lists are merged. A range list may be specified as:

```
DELISN=10-80,90,100-110
```

Overlapping ranges and duplicate ISNs are not allowed. You can specify, at most, 32 single ISNs or ISN ranges. When deleting ISNs from an Adabas expanded file, you can specify the complete list for all component files. The UPDATE function selects the appropriate ISNs from the list and deletes them from the component file.

Note:

This parameter cannot be specified in an ADALOD UPDATE operation on a *LOB file*.

DSREUSE: Data Storage Reusage

DSREUSE indicates whether or not Data Storage space that becomes available as a result of a record deletion is to be reused.

This parameter is in effect for the execution of the UPDATE function only. The permanent setting of DSREUSE is not changed. That permanent setting is the default if this value is not specified.

ETID: Multiclient File Owner ID

The ETID parameter assigns a new owner ID to all records being added to an existing multiclient file. The owner ID is automatically adjusted to the length for owner IDs specified by LOWNERID when the multiclient file was last loaded. If no ETID is specified, all loaded records keep their owner IDs specified on the input source.

The ETID parameter must be specified if the existing file is multiclient and the input file was not unloaded from a multiclient file. ETID must not be specified if the existing file is a non-multiclient file.

Note:

If the ETID parameter is used, the ADALOD utility requires an active nucleus. The nucleus will translate the ETID value into the internal owner ID value.

ISNREUSE: ISN Reusage

ISNREUSE indicates whether the ISN for a deleted record can be reassigned to a new record.

This ISNREUSE setting is in effect only during execution of the UPDATE function. The permanent ISNREUSE setting is unchanged. The permanent setting is the default if this value is not specified.

LIP: ISN Work Pool Size

LIP specifies the size of the work pool for containing ISNs to be deleted. Four bytes per ISN and eight bytes per ISN range are required in this pool. The value may be specified in bytes as a numeric value ("2048") or in kilobytes as a value followed by a "K" ("2K"). The default for LIP is 2000 bytes.

LWP: Work Pool Size

LWP specifies the size of the work pool to be used for descriptor value sorting. The value can be specified in bytes or kilobytes followed by a "K". If no value is specified, the default is 1048576 bytes (or 1024K); however, to shorten ADALOD run time for files with very long descriptors or an unusually large number of descriptors, set LWP to a higher value. To avoid problems with the Sort data set, a smaller LWP value should be specified when updating relatively small files.

The minimum work pool size depends on the sort data set's device type:

Sort Device	Minimum LWP	Minimum LWP
	Bytes	Kilobytes
2000	106496	104K
2314	090112	88K
3375	131072	128K
3380	139264	136K
3390	159744	156K

MAXISN: Highest ISN to be Allocated to the File

The MAXISN parameter may be used to specify a new ISN setting for the file. This parameter should be used if the current record count plus the number of ISNs (records) to be added exceeds the current MAXISN setting. The specified larger value determines the additional space required for the address converter, and causes ADALOD to allocate a new extent. A smaller MAXISN value causes no change in the address converter space.

Note:

The MAXISN setting for a file cannot be increased if the file was last loaded with NOACEXTENSION active.

The MAXISN setting should be increased by an amount suitable for all planned expansion; this avoids using up the address converter extent too quickly, and alleviates the need to either unload and reload the file or run the ADAORD REORFASSO utility because the maximum number of address converter extents has been allocated.

With the optional ACRABN parameter, the beginning of the new address converter extent can be set to a specific RABN number. See the ACRABN parameter description for more information.

If the MAXISN parameter is omitted, ADALOD allocates new address converter extents only if the old MAXISN value is exceeded.

MAXISN2: Highest ISN to be Allocated in the Secondary Address Converter for the File

The MAXISN2 parameter is optional, regardless of whether or not spanned records exist in the ADALOD input file or not. Use this parameter to specify the desired size of the secondary address converter (AC2) in ISNs. The secondary address converter is used to map secondary ISNs of secondary spanned records to the RABNs of the Data Storage blocks where the secondary records are stored.

ADALOD determines the number of ISN mappings for which to allow space in the secondary AC using the following method:

1. If a secondary AC is not yet allocated, one secondary AC block is allocated.
2. If the current MAXISN2 setting is greater than or equal to four times the MAXISN value (the current maximum primary ISNs expected), the same algorithm is used to determine the number of secondary ISNs as is used to allocate additional primary ISNs.

3. If none of the above conditions is met, then secondary AC space is allocated as the smaller of the following two calculations:
 - The product of 10 times the old MAXISN2 setting (10 x oldMAXISN2)
 - The sum of the old MAXISN2 setting and the MAXISN setting (oldMAXISN2 + MAXISN)

There is no default value.

NOUSERABEND: Termination without Abend

When a parameter error or a functional error occurs while this utility function is running, the utility ordinarily prints an error message and terminates with user abend 34 (with a dump) or user abend 35 (without a dump). If NOUSERABEND is specified, the utility will *not* abend after printing the error message. Instead, the message "*utility* TERMINATED DUE TO ERROR CONDITION" is displayed and the utility terminates with condition code 20.

Note:

When NOUSERABEND is specified, we recommend that it be specified as the first parameter of the utility function (before all other parameters). This is necessary to ensure that its parameter error processing occurs properly.

NUMREC: Limit Number of Records to Be Added

NUMREC limits the number of records to be added. If NUMREC is specified, ADALOD processing terminates after adding the number of records specified (unless an end-of-file condition on the input data set has already caused ADALOD termination). If this parameter is omitted, *all* input records are added.

If the input data set contains more records than specified by NUMREC, ADALOD adds the number of records specified by NUMREC and then terminates with condition code 4.

Note:

This parameter cannot be specified in an ADALOD UPDATE operation on a *LOB file*.

PASSWORD: File Password

If the file to be updated is password-protected, the parameter must be used to provide a valid password. There is no default for PASSWORD.

RESTART: Restart Interrupted ADALOD Execution

RESTART forces an interrupted ADALOD run to be restarted, beginning with the last *restart point* reached before the interruption. The *restart point* is the latest point of execution that can be restored from the Temp data set.

If ADALOD is interrupted by a defined error condition, ADALOD issues a message indicating whether or not a restart is possible.

When restarting the ADALOD operation, the following parameters may be changed:

- TEMPSIZE can be increased to make the temp data set larger. Note, however, that the temp data set contents *must not be changed* because it contains information necessary for the restart operation;
- The SORTSIZE and SORTDEV parameters and the sort data set can be changed.

No other parameters can be changed. The DDEBAND/EBAND, DDFILEA/FILEA and DDISN/ISN data sets must remain the same.

RPLLOAD: Replicate Update Data

The ADALOD UPDATE RPLLOAD parameter is an Event Replicator for Adabas parameter that can be specified for the Adabas database files when using replication. It is only allowed for ADALOD UPDATE if replication is already turned on for a database.

Note:

The version of Event Replicator specified in an ADALOD UPDATE utility job that specifies RPLLOAD=YES must match the version of Event Replicator used by the Event Replicator Server. In addition, the Adabas version used by the Event Replicator Server must be greater than or equal to the Adabas version used in an ADALOD UPDATE utility job that specifies RPLLOAD=YES.

The RPLLOAD parameter indicates whether or not inserts and deletes to the Adabas database via the ADALOD UPDATE utility will be replicated to the Event Replicator Server. Valid values are "YES" and "NO"; the default is "NO".

When RPLLOAD=YES is specified, ADALOD UPDATE inserts and deletes to the Adabas database will be replicated to the Event Replicator Server. When RPLLOAD=NO is specified, ADALOD UPDATE inserts and deletes are *not* replicated.



Warning:

If ADALOD UPDATE ends abnormally (due to insufficient space, for example), updates made to the file before the abnormal ending cannot be backed out; there is no automated recovery for the updated data or for the replicated data. Software AG therefore recommends that you perform an ADASAV SAVE on the file before you run ADALOD UPDATE.

SAVEDREC: Save Deleted Records on a Sequential File

SAVEDREC indicates that deleted records are to be written to a sequential data set. The format of the data set is identical to that created by the ADAULD utility with the MODE=SHORT option.

Note:

This parameter cannot be specified in an ADALOD UPDATE operation on a *LOB file*.

SKIPREC: Number of Records to Be Skipped

SKIPREC is the number of input records to be skipped before beginning to process updates. The default is 0 (no records are skipped).

Note:

This parameter cannot be specified in an ADALOD UPDATE operation on a *LOB file*.

SORTDEV: Sort Device Type

ADALOD uses the sort data set to sort descriptor values. The SORTDEV parameter indicates the device type to be used for this data set. This parameter is required only if the device type to be used is different from that specified by the ADARUN DEVICE parameter.

TEMPDEV: Temporary Storage Device Type

ADALOD uses the temp data set to store intermediate data. The TEMPDEV parameter indicates the device type to be used for this data set. This parameter is required only if the device type to be used is different from the standard device type assigned to Temp by the ADARUN DEVICE parameter.

TEST: Test Syntax

The TEST parameter tests the operation syntax without actually performing the operation. Only the syntax of the specified parameters can be tested; not the validity of values and variables.

USERISN: User ISN Assignment

USERISN=YES indicates that the USERISN option for the file is to be in effect, and that the ISN for each new record is being supplied by the user in the input data. If USERISN=NO, Adabas assigns the ISN for each new record.

The specified USERISN setting is effective only while the UPDATE function is executing. The permanent USERISN setting is not changed, and is the default if this parameter is not specified.

When performing an ADALOD UPDATE function on a file with a hyperdescriptor for which the hyperdescriptor exit changed the ISNs of descriptor values, USERISN=YES is no longer required for the add/load operation.

When adding records *from* a non-USERISN=YES file, the ADALOD parameter USERISN=NO must be specified and the file to be updated must have the USERISN option. This feature is useful for Adabas Text Retrieval (TRS).

Examples

Example 1:

```
ADALOD  UPDATE  FILE=6,MAXISN=18000
ADALOD           TEMPSIZE=10, SORTSIZE=5
```

Records are to be added to file 6. The MAXISN for the file is to be increased to 18,000.

Example 2:

```
ADALOD  UPDATE  FILE=7,TEMPSIZE=10,
ADALOD           ETID=USER3, SORTSIZE=5
```

Records with user's owner ID of USER3 are to be added to multiclient file 7.

Example 3:

```
ADALOD  UPDATE  FILE=8,DELISN=1000-1999,5000-5999
ADALOD           TEMPSIZE=10, SORTSIZE=5
```

The records with ISNs 1,000 to 1,999 and 5,000 to 5,999 are to be deleted from file 8. If an input data set is provided, records are to be added.

Example 4:

```
ADALOD  UPDATE  FILE=6
ADALOD           DDISN,SAVEDREC
ADALOD           TEMPSIZE=10, SORTSIZE=5
```

Records are to be deleted from file 6. The ISNs of the records to be deleted are contained in an input data set. The deleted records are to be saved on an output data set.

Example 5:

```
ADALOD  UPDATE  FILE=6,DDISN,LIP=20K,SKIPREC=500
ADALOD           TEMPSIZE=5, SORTSIZE=10
```

Records are to be added and deleted from file 6. The ISNs which identify the records to be deleted are contained in an input data set (DDISN). The size of the ISN pool is set to 20K. The first 500 records on the input data set are to be skipped.

Formats for Specifying ISNs

There are two formats for specifying ISNs in the DDISN or ISN data set. The first format can be used in all cases where only 31-bit ISNs are specified. A record can contain a mix of single ISNs and ranges of ISNs.

The second format supports 32-bit ISNs and can only be used with Adabas version 6 and above. Each record can specify *either* single ISNs (indicated by X'00000000' in the first fullword) *or* ranges of ISNs (indicated by X'FFFFFFFF' in the first fullword).

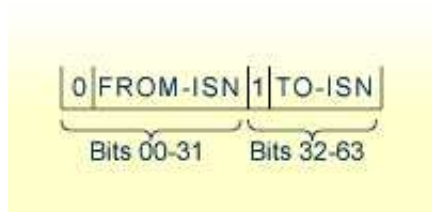
If the first fullword in a record contains a value other than X'00000000' or X'FFFFFFFF', it is assumed to be the 31-bit format. The DDISN/ISN data set can contain records in both formats.

Format 1: 31-Bit Format

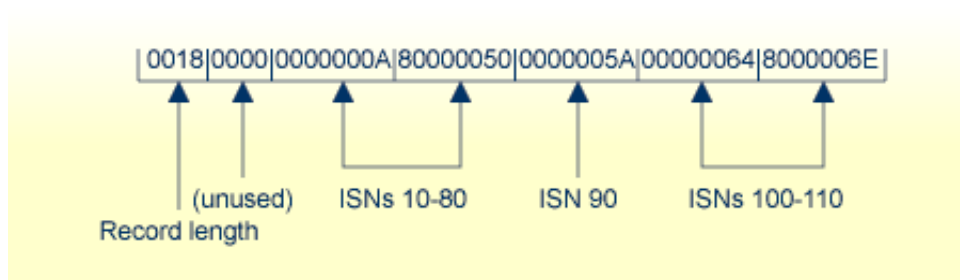
A single ISN requires 4 bytes. Set the high-order bit to 0 and specify the ISN in bits 01-31:



A range of ISNs requires 8 bytes. In the first four bytes, specify the first ISN in the range as a single ISN; in the next four bytes, set the high-order bit to 1 and specify the last ISN:

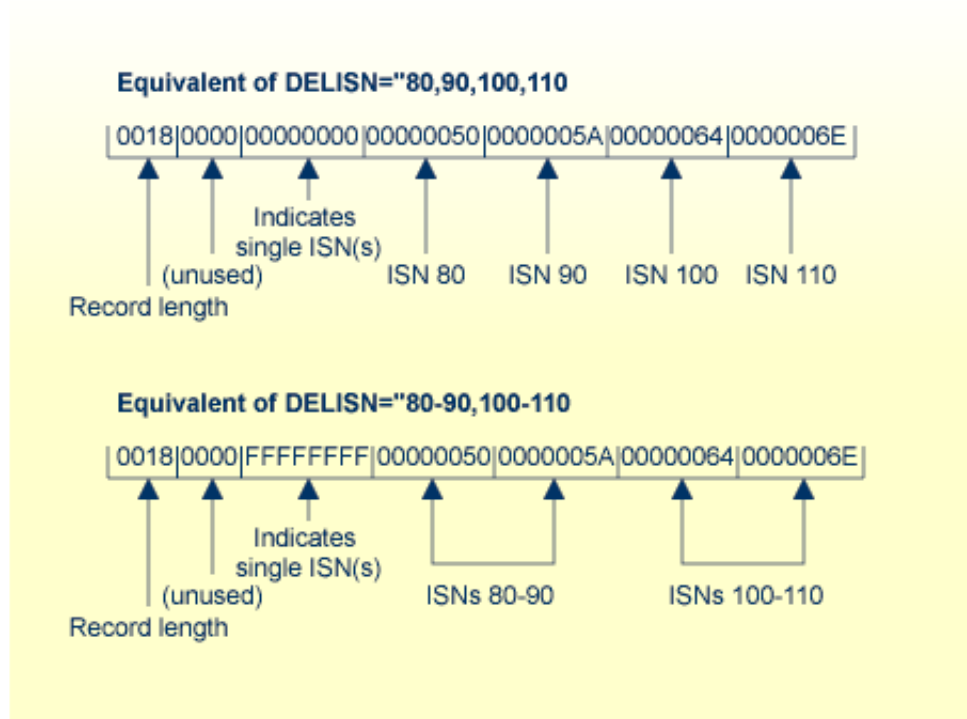


The following example shows a variable-length record containing the equivalent of DELISN=10-80,90,100-110:



Format 2: 32-Bit Format

In the 32-bit format, the first fullword in each record indicates whether the record contains single ISNs or ranges of ISNs. To indicate single ISNs, put zero in the first fullword (X'00000000'); to indicate ranges of ISNs, put -1 (X'FFFFFFFF'). In the following example, the first record contains single ISNs; the second record contains ranges. The two records are identical except for the indicator in the first fullword.



UPDATE Data and Space Requirements

The following general information describes data requirements for UPDATE operation, and how ADALOD UPDATE allocates space. For more information about space allocation, refer to the *Adabas DBA Reference* documentation.

Input Data for UPDATE Operations

Records to be added must be in the form of compressed data records produced by the ADACMP or ADAULD utility. The field definitions used for the ADACMP run must agree with the definitions for the file to which the records will be added as contained in the field definition table (FDT).

Note:

Records being added to a ciphered file must already be encrypted using the same cipher code as was used for the records already in the file.

The ISNs of records to be deleted may be provided with the DELISN parameter and/or in an input data set. If provided in an input data set, each ISN must be provided as a 4-byte binary number. The data set must have the record format VARIABLE BLOCKED. If desired, all ISNs to be added to or deleted from an Adabas expanded file can be specified; the UPDATE function selects the appropriate ISNs for the component file being processed.

UPDATE Space Allocation

If records are to be added and a larger MAXISN value has been specified, an additional address converter extent will be allocated by ADALOD. The size of the new extent is based on the difference between the new MAXISN and the previous MAXISN setting. If either insufficient space is available for the new extent or the maximum number of extents has already been allocated, processing ends with an error message.

If an additional Data Storage extent is required, ADALOD allocates an additional extent equal to approximately 25 percent of the total size of the Data Storage extents currently allocated to the file. As for the address converter, processing ends with an error message if either sufficient space is not available for the added extent or the maximum number of extents has already been allocated.

Generating UPDATE Descriptor Information

When adding records, ADALOD UPDATE generates a list of all descriptor values and the corresponding ISNs of the new records, and writes this information to the temp data set.

Associator Updating with UPDATE

Before processing the input, ADALOD UPDATE copies the file's existing normal index to the temp data set, but removes the descriptor values of any ISNs to be deleted.

ADALOD sorts the information written to temp during the input phase and merges the sorted values with the current normal index. The normal index is reordered during this process, and the Associator block padding factor is reestablished for each block. A new upper index is then created.

Empty space in partially filled blocks resulting from descriptor updating is reused. This can increase the number of empty blocks at the end of the index. Although one or more normal index and/or upper index extents may become empty as the result of the reorder process, ADALOD does not condense, delete, or change the size of these extents.

If new free space is needed for the normal index or upper index, ADALOD allocates an additional extent (or extents). Each additional extent allocated is equal to approximately 25 percent of the total current space allocated to the index. If insufficient space is available for the additional extent or if the maximum number of extents has already been allocated, ADALOD terminates with an error message.

Mass Updates of Expanded Files

Using ADALOD UPDATE for a mass update to an expanded file, records must be added to or deleted from each component file individually. However, each component file can be processed using the same ADALOD commands.

When deleting a record with DELISN or DDISN, the complete list of ISNs to be deleted from all component files can be supplied. ADALOD automatically selects only the ISN values from the specified range that is appropriate for the component file currently being processed.

The same is true when adding new records with USERISN=YES.

When new expanded file records are being added with USERISN=NO but no free ISN is found, the loader cannot allocate a new address converter extent since the ISN range cannot be increased (NOACEXTENSION is active for all component files). Instead, ADALOD creates the index as though end-of-file had been reached. The remaining records not loaded may be added later to another component file using the SKIPREC parameter.

ADALOD does not check for unique descriptor values across component file boundaries.

Example:

The following is an example for performing a mass update to an expanded file (only the relevant parts of the complete jobs are shown):

```
.  
.
//DDEBAND DD DSN=MOREDATA.LOAD.PART1-2,...
//DDKARTE DD *
ADALOD UPDATE FILE=40,USERISN=YES
ADALOD DELISN=9000001-9500000,12000001-14000000
ADALOD SORTSIZE=...,TEMPSIZE=...
.
.
//DDEBAND DD DSN=MOREDATA.LOAD.PART1-2,...
//DDKARTE DD *
ADALOD LOAD FILE=41,USERISN=YES
ADALOD DELISN=9000001-9500000,12000001-14000000
ADALOD SORTSIZE=...,TEMPSIZE=...
.
-
```