Adabas Utilities

Utility Function	Use to...
ADAACK ACCHECK	Check the address converter for a specific file or range of files or for a specific ISN or range of ISNs.
ADACDC	Produce a file containing the delta of all changes made to the database over the period covered by the input protection logs. You can also use this utility to produce a file that lists each individual insert and delete transaction made to the database over the period covered by the input protection logs (without producing a delta of these changes).
ADACMP COMPRESS	Edit and compress data records that are to be loaded into the database.
ADACMP DECOMPRESS	Decompress individual files in the Adabas database.
ADACNV CONVERT	Convert a database to higher Adabas version.
ADACNV REVERT	Revert a database to a lower Adabas version.
ADADBS ADD	Add a new data set to the Associator or Data Storage.
ADADBS ADDCLOG	Dynamically add a command log (CLOG) data set
ADADBS ADDPLOG	Dynamically add a protection log (PLOG) data set
ADADBS ALLOCATE	Allocate an extent of a specific size for an address converter, Data Storage, or normal or upper index.
ADADBS CHANGE	Change the standard length or format of a field.
ADADBS CVOLSER	Print the Adabas file extents on a given volume.
ADADBS DEALLOCATE	Deallocate a file extent for an address converter, Data Storage, or normal index or upper index.
ADADBS DECREASE	Decrease the size of the last data set currently being used for the Associator or Data Storage.
ADADBS DELCLOG	Dynamically delete a command log (CLOG) data set
ADADBS DELCP	Delete checkpoint records.
ADADBS DELDE	Logically delete descriptors.
ADADBS DELETE	Delete a file from a database.
ADADBS DELFN	Logically delete fields from a database file.
ADADBS DELPLOG	Dynamically delete a protection log (PLOG) data set
ADADBS DSREUSE	Activate or deactivate the reuse of Data Storage blocks for a file.
ADADBS ENCODEF	Change the file encoding of a file.
ADADBS EXPFILE	Insert or remove files in an expanded file chain.
ADADBS INCREASE	Increase the size of the last data set currently being used for the Associator or Data Storage.
ADADBS ISNREUSE	Reuse ISNs in a database.

Utility Function	Use to...
ADADBS MODFCB	Modify various parameters for a non-system Adabas file.
ADADBS MUPEX	Set the maximum count for MU and PE fields in a database.
ADADBS NEWFIELD:	Add one or more fields to a file.
ADADBS ONLINVERT	Start an online invert process.
ADADBS ONLREORFASSO	Start an online process to reorder the Associator of specified files.
ADADBS ONLREORFDATA	Start an online process to reorder the Data Storage of specified files.
ADADBS ONLREORFILE	Start an online process to reorder the Associator and Data Storage of specified files.
ADADBS OPERCOM	Issue operator commands to the Adabas nucleus.
ADADBS PRIORITY	Set or change the Adabas priority of a user.
ADADBS REACTLOG	Reactivate command logging.
ADADBS RECORDSPANNING	Enable or disable record spanning for a file.
ADADBS RECOVER	Recover allocated space by rebuilding the free space table (FST).
ADADBS REFRESH	Set a file to empty status.
ADADBS REFRESHSTATS	Reset the statistical values maintained by the Adabas nucleus for its current session.
ADADBS RELEASE	Release a descriptor from the descriptor space.
ADADBS RENAME	Rename a file or database.
ADADBS RENUMBER	Change the number of a file in the database.
ADADBS REPLICATION	Activate or deactivate replication for an Adabas database file.
ADADBS REPTOR	Activate, deactivate, open, or close an Event Replicator for Adabas resource.
ADADBS RESETDIB	Reset entries in the active utility list.
ADADBS RESETPPT	Resets the PPT blocks on the Associator data set.
ADADBS SPANCOUNT	Count and display the number of records in a file that are spanned.
ADADBS TRANSACTIONS	Suspend and resume update transaction processing for the database.
ADADBS UNCOUPLE	Uncouple files in the database.
ADADBS UNDELDE	Undelete logically deleted descriptors.
ADADBS UNDELFN	Logically undelete fields from a database file.
ADADCK DSCHECK	Check Data Storage and the Data Storage space table (DSST) of specified files in the database.
ADADEF DEFINE	Define a database and checkpoint file.

Utility Function	Use to...
ADADEF MODIFY	Change field encodings.
ADADEF NEWWORK	Define a Work file.
ADAFRM	Format Adabas database components.
ADAICK ACCHECK	Check the address converter of a specified database file.
ADAICK ASSOPRINT	Print and dump Associator blocks.
ADAICK BATCH	Set the printout width to 132 characters per line.
ADAICK DATAPRINT	Print and dump Data Storage blocks for specified RABNs.
ADAICK DSCHECK	Print and dump the content of Data Storage records in a file.
ADAICK DUMP	Request that blocks be printed in dump format.
ADAICK FCBPRINT	Print and dump the file control block (FCB) of a file.
ADAICK FDTPRINT	Print and dump the field definition table of a file.
ADAICK GCBPRINT	Print and dump the general control blocks (GCBs).
ADAICK ICHECK	Check the index against the address converter.
ADAICK INT	Request that blocks be printed in interpreted print format.
ADAICK NIPRINT	Print and dump the normal index for a file.
ADAICK NOBATCH	Set the print width to 80 characters per line.
ADAICK NODUMP	Suppress dump print format.
ADAICK NOINT	Suppress interpreted print format.
ADAICK PPTPRINT	Print and dump the parallel participant table (PPT).
ADAICK UIPRINT	Print and dump the upper index for a file.
ADAINV COUPLE	Define one descriptor for each of two files to be coupled.
ADAINV INVERT	Create descriptors, subdescriptors, superdescriptors, hyperdescriptors, phonetic descriptors or collation descriptors for existing fields in a file.
ADALOD LOAD	Load a file into the database.
ADALOD UPDATE	Add or delete a large number of records (ISNs) to or from an existing file.
ADAMER	Run the ADAM estimation. This utility produces statistics that indicate the number of Data Storage accesses required to find and read a record when using an ADAM descriptor.
ADAORD REORASSO	Reorder the Associator.
ADAORD REORDATA	Reorder Data Storage.
ADAORD REORDB	Reorder the database.
ADAORD REORFASSO	Reorder the Associator for a single file.
ADAORD REORFDATA	Reorder Data Storage for a single file.

Utility Function	Use to...
ADAORD REORFILE	Reorder a file.
ADAORD RESTRUCTUREDB	Restructure the database.
ADAORD RESTRUCTUREF	Restructure single files.
ADAORD STORE: Store Files	Load one or more files into an existing database using output produced by the ADAORD RESTRUCTURE functions.
ADAPLP	Print data protection records from the protection log (PLOG) or Work data sets.
ADAPRI	Print the contents of a block (or range of blocks) contained in the Associator, Data Storage, Work, temp, sort, multiple data set command log, multiple data set protection log, or the recovery log data set.
ADARAI CHKDB	Check that the recovery status of the nucleus is the status specified by the CHKDB function (active or inactive).
ADARAI DISABLE	Disable recovery logging by setting the RLOG table (control block) to inactive status.
ADARAI LIST	View the recovery log contents in table form.
ADARAI PREPARE	Initialize and start recovery logging to the recovery log.
ADARAI RECOVER	Build a recovery job stream for recovering the Adabas database or selected database files.
ADARAI REMOVE	Disable recovery logging by updating the Associator GCBs to indicate that recovery logging (that is, the Recovery Aid) is no longer active in the database, and that information will no longer be recorded in the recover log (RLOG).
ADAREP	Request a database status report.
ADARES BACKOUT SIBA	Back out updates applied between two specified checkpoints using the sequential protection log (SIBA).
ADARES BACKOUT DPLOG or MPLOG	Back out updates applied between two checkpoints contained on the same dual or multiple protection log.
ADARES CLCOPY	Copy the dual command log.
ADARES COPY	Copy a sequential protection log or save tape.
ADARES MERGE CLOG	Merge nucleus cluster command logs.
ADARES PLCOPY	Copy protection log data to a sequential data set.
ADARES REGENERATE	Reapply all the updates performed between two checkpoints.
ADARES REPAIR	Repair one or more Data Storage blocks, using the protection log and the output of the ADASAV utility.
ADASAV RESTONL	Restore a database from a database SAVE data set created while the Adabas nucleus was active.

Utility Function	Use to...
ADASAV RESTONL FILES	Restore files to their original RABNs in an existing database. Files are restored from a file or database SAVE data set created while the Adabas nucleus was active.
ADASAV RESTONL FMOVE	Restore files to any free space in an existing database. Files are restored from a file or database SAVE data set created while the Adabas nucleus was active.
ADASAV RESTONL GCB	Restore the general control blocks (GCBs), Associator RABNs 3-30 of the database, the checkpoint file, the security file (if present), and all specified files. These file structures are restored from a database SAVE data set created while the Adabas nucleus was active.
ADASAV RESTORE	Restore a database from a database SAVE data set created while the Adabas nucleus was inactive.
ADASAV RESTORE FILES	Restore files to their original RABNs in an existing database. Files are restored from a file or database SAVE data set created while the Adabas nucleus was inactive.
ADASAV RESTORE FMOVE	Restore files to any free space in an existing database. Files are restored from a file or database SAVE data set created while the Adabas nucleus was inactive.
ADASAV RESTORE GCB	Restore the general control blocks (GCBs), Associator RABNs 3-30 of the database, the checkpoint file, the security file (if present), and all specified files. These file structures are restored from a database SAVE data set created while the Adabas nucleus was inactive.
ADASAV RESTPLOG	Restore changes contained in the protection log (PLOG) to the already restored database or files.
ADASAV SAVE	Save the contents of the database to a sequential data set.
ADASAV SAVE FILES	Save the contents of one or more files to a sequential data set.
ADASEL	Select and decompress information in the Adabas sequential (SIBA) or dual/multiple (PLOG) protection log and write it to a print data set (DDDRUCK/ DRUCK) or a user-specified output data set.
ADAULD UNLOAD FILE	Unload a specified Adabas file.
ADAVAL	Validate any or all files within an Adabas database except the checkpoint and security files.
ADAWRK	Produce reports from records in the autorestart area of Work part 1. This information can be used when the database autostart fails and the database will not come up. It can help you determine how to handle database recovery.
ADAZAP	Display (in hexadecimal format) and optionally change the contents of the Associator, Data Storage, or Work data sets.
ADAZIN	Print Adabas maintenance and SVC information.
Field Definition Statements	Define or modify a field definition. Field definition statements can be specified in various Adabas utility functions.

ADAACK ACCHECK: Check Address Converter Against Data Storage

```
ADAACK  ACCHECK  [FILE= { file | file1 - filex } ]
                 [ISN= isn1 - isnx ]
                 [NOOPEN]
                 [NOUSERABEND]
```

ADACDC: Capture Delta Changes

```
ADACDC  [FILE= filelist ]
        [IGNORESPANNED]
        [ISN]
        [MAXLOGRECLEN = { n | 1048576 } ]
        [NOET]
        [PHASE={ 1 | 2 | BOTH ]
        [RESETTXF]
        [SPANREC]
```

ADACMP COMPRESS: Compress an Adabas File

```
ADACMP  COMPRESS  {field-definition-statements | FDT = file-number}
                  [CODE = cipher-code ]
                  [DATADEVICE = device-type ]
                  [DEVICE = device-type-list ]
                  [FACODE = file-alpha-EBCDIC-key ]
                  [FILE = {file-number | 0 } ]
                  [FWCODE = file-wide-key ]
                  [FUWCODE = wide-key ]
                  [FORMAT = format ]
                  [HEADER = { YES | NO } ]
                  [LOBDEVICE = device-type-list ]
                  [LOBVALUES = { YES | NO } ]
                  [MAXLOGRECLEN = buffer-size ]
                  [MUPECOUNT = 1 | 2 ]
                  [MUPEX]
                  [NOUSERABEND]
                  [NUMREC = number-of-records]
                  [PASSWORD = "password" ]
                  [RECFM = { F | FB | V | VB | U } ]
                      [,LRECL = record-length } ]
                  [SPAN]
                  [TZ = ' timezone-name ' [,DST] ]
                  [ { USERISN | MINISN = { start-isn | 1 } } ]
                  [UACODE = userdata-alpha-key ]
                  [UARC = {userdata-architecture-key | 2 } ]
                  [UWCODE = userdata-wide-key ]
```

ADACMP DECOMPRESS: Decompress Adabas Files

```
ADACMP  DECOMPRESS  [CODE = cipher-code ]
                    [FORMAT = output-record-format-definition ]
                    [HEADER = { YES | NO } ]
                    [INFILE = file-number ]
                       [ETID = owner-id ]
                       [LPB = prefetch-buffer-size ]
                       [PASSWORD = ' password ' ]
                       [SORTSEQ = { descriptor [,NU] | ISN } ]
                       [UTYPE = { EXF | EXU } ]
                    [ISN]
                    [LOBVALUES = { YES | NO } ]
                    [MAXLOGRECLEN = buffer-size ]
                    [NOUSERABEND]
                    [NUMREC = number-of-records ]
                    [TRUNCATE]
                    [TZ = ' timezone-name ' [,DST] ]
                    [UACODE = userdata-alpha-key ]
                    [UARC = {architecture-key | 2 } ]
                    [UWCODE = userdata-wide-key ]
```

Field Definition Statements

The following syntax must be followed when entering field definitions. A minimum of one and a maximum of 3214 definitions may be specified.

Statement Type	Syntax
Field and Group	**FNDEF = '***level*, *name* **[** , *length*, *format* **] [, MU [(***occurrences***)]] [** , *option* **] ... '**
Periodic Group	**FNDEF = '***level*, *name* **[, PE [(** *occurrences* **)]]'**
Collation descriptor	**COLDE = '***number*, *name* **[, UQ [, XI]] =** *parent-field***'**
Hyperdescriptor	**HYPDE = '***number*, *name*, *length*, *format* **[{** , *option***} ...] = {** *parent-field* **} , ...'**
Phonetic descriptor	**PHONDE = '** *name* **(***field***)'**
Subdescriptor	**SUBDE = '***name* **[, UQ [XI]] =** *parent-field* **(***begin*, *end***)'**
Subfield	**SUBFN = '** *name* **=** *parent-field* **(***begin*, *end***)'**
Superdescriptor	**SUPDE = '** *name* **[, UQ [XI]] = {** *parent-field* **(***begin*, *end***) } , ...'**
Superfield	**SUPFN = '***name* **=** *parent-field* **(***begin*, *end***)[,** *parent-field* **(** *begin* , *end* **)]...'**

User comments may be entered to the right of each definition. At least one blank must be present between a definition and any user comments.

ADACNV CONVERT: Convert a Database to Higher Version

```
ADACNV  CONVERT  [IGNPPT]
                 [NOUSERABEND]
                 [PLOGDEV = multiple-PLOG-device-type ]
                 [RESTART]
                 [TEST]
                 [TOVERS = target-version ]
```

ADACNV REVERT: Revert a Database to Lower Version

```
ADACNV  REVERT  [TOVERS = target-version ]
                [IGNPPT]
                [NOUSERABEND]
                [PLOGDEV = multiple-PLOG-device-type ]
                [RESTART]
                [TEST]
```

ADADBS ADD: Add Data Set

```
ADADBS  ADD  { ASSOSIZE = size [ ASSODEV = device-type ] |
               DATASIZE = size [ DATADEV = device-type ] }
             [NOUSERABEND]
             [TEST]
```

ADADBS ADDCLOG: Dynamically Add CLOG Data Sets

```
ADADBS  ADDCLOG NUMBER = clog-ds-number
                [NOUSERABEND]
                [NUCID = nucid ]
                [CLOGDEV = device-type ]
                [TEST]
```

ADADBS ADDPLOG: Dynamically Add PLOG Data Sets

```
ADADBS  ADDPLOG NUMBER = plog-ds-number
                [NOUSERABEND]
                [NUCID = nucid ]
                [PLOGDEV = device-type ]
                [TEST]
```

ADADBS ALLOCATE: Allocate File Extent

```
ADADBS  ALLOCATE    FILE = file-number
                    { ACSIZE | AC2SIZE | DSSIZE | NISIZE | UISIZE } = size
                    [DEVICE = device-type ]
                    [NOUSERABEND]
                    [PASSWORD = ' password ' ]
                    [STARTRABN = start-rabn ]
                    [TEST]
```

ADADBS CHANGE: Change Standard Length or Format of a Field

```
ADADBS CHANGE FILE = file-number
                    { FNDEF = ' Adabas-field-definition '
                    { FIELD = ' field-name ' { FORMAT=P | LENGTH= new-length | OPTION={ LA | NOUQ } } }
                    [ NOUSERABEND ]
                    [ PASSWORD = ' password ']
                    [TEST]
```

ADADBS CVOLSER: Print Adabas Extents on Given Volume

```
ADADBS  CVOLSER  VOLSER = volume-serial-number
                 [NOUSERABEND]
                 [TEST]
```

ADADBS DEALLOCATE: Deallocate File Extent

```
ADADBS  DEALLOCATE  FILE = file-number
                    { ACSIZE | AC2SIZE | DSSIZE | NISIZE | UISIZE } = size
                    [NOUSERABEND]
                    [PASSWORD = ' password ' ]
                    [STARTRABN = start-rabn ]
                    [TEST]
```

ADADBS DECREASE: Decrease Last Associator or Data Storage Data Set Size

```
ADADBS  DECREASE   { ASSOSIZE | DATASIZE } = sizeB
                   [NOUSERABEND]
                   [TEST]
```

ADADBS DELCLOG: Dynamically Delete CLOG Data Sets

```
ADADBS  DELCLOG  NUMBER = clog-ds-number
                 [NOUSERABEND]
                 [NUCID = nucid ]
                 [TEST]
```

ADADBS DELCP: Delete Checkpoint Records

```
ADADBS  DELCP  TODATE = yyyymmdd
               [NOUSERABEND]
               [TEST]
```

ADADBS DELDE: Logically Delete a Descriptor

```
ADADBS  DELDE  FILE = file-number
               DESCRIPTOR = descriptor-name
               [NOUSERABEND]
               [PASSWORD = password ]
               [TEST]
```

ADADBS DELETE: Delete File

```
ADADBS  DELETE   { FILE = fnr  [KEEPFDT] [PASSWORD = 'password' ] }
                 [NOUSERABEND]
                 [TEST]
```

ADADBS DELFN: Logically Delete Fields

```
ADADBS DELFN    FILE = file-number
                FIELDLIST = ' field-list '
                [PASSWORD = password ]
                [NOUSERABEND]
                [TEST]
```

ADADBS DELPLOG: Dynamically Delete PLOG Data Sets

```
ADADBS  DELPLOG  NUMBER = plog-ds-number
                 [NOUSERABEND]
                 [NUCID = nucid ]
                 [TEST]
```

ADADBS DEVENTLOG: Display Adabas Event Log

```
ADADBS  DELPLOG  NUMBER = plog-ds-number
                 [NOUSERABEND]
                 [NUCID = nucid ]
                 [TEST]
```

ADADBS DSREUSE: Reuse Data Storage Blocks

```
ADADBS  DSREUSE  FILE = file-number
                 MODE = { ON | OFF }
                 [NOUSERABEND]
                 [PASSWORD = 'password ' ]
                 [RESET]
                 [TEST]
```

ADADBS ENCODEF: Change File Encoding

```
ADADBS  ENCODEF  FILE = file-number
                 FACODE =  alpha-key | UWCODE = wide-key
                 [NOUSERABEND]
                 [TEST]
```

ADADBS EXPFILE: Insert or Remove Files in Expanded File Chains

```
ADADBS EXPFILE  FILE = file-number
                { INSERT | REMOVE }
                ANCHOR = file-number
                [PASSWORD = password ]
                [NOUSERABEND]
                [TEST]
```

ADADBS INCREASE: Increase Last Associator or Data Storage Data Set Size

```
ADADBS  INCREASE  { ASSOSIZE | DATASIZE } = size
                  [ NOUSERABEND ]
                  [TEST]
```

ADADBS ISNREUSE: Reuse ISNs

```
ADADBS  ISNREUSE  FILE = file-number
                  MODE = { ON | OFF }
                  [NOUSERABEND]
                  [PASSWORD = 'password ' ]
                  [RESET]
                  [TEST]
```

ADADBS MODFCB: Modify File Parameters

```
ADADBS MODFCB  FILE = file-number
               [ASSOPFAC = new-padding-factor ]
               [DATAPFAC = new-padding-factor ]
               [FILEREADONLY = { YES | NO } ]
               [LOBFILE = file-number ]
               [MAXDS = maximum-secondary-allocation ]
               [MAXNI = maximum-secondary-allocation ]
               [MAXUI = maximum-secondary-allocation ]
               [MAXRECL = maximum-compressed-record-length ]
               [NOUSERABEND]
               [PASSWORD = ' password ' ]
               [PGMREFRESH = { YES | NO } ]
               [RPLUPDATEONLY = {YES | NO} ]
               [SYFMAXUV = nn ]
               [TEST]
```

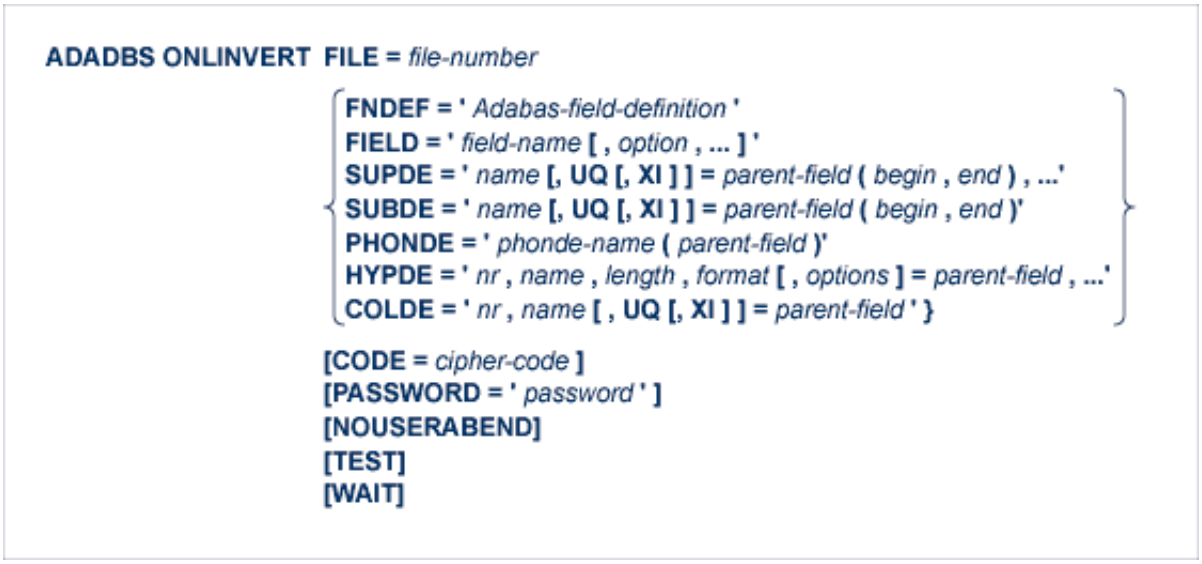
ADADBS MUPEX: Set Maximum Count for MU and PE Fields

```
ADADBS  MUPEX  FILE = file-number
               MUPECOUNT = { 1 | 2 }
               [PASSWORD = 'password' ]
               [NOUSERABEND]
               [TEST]
```

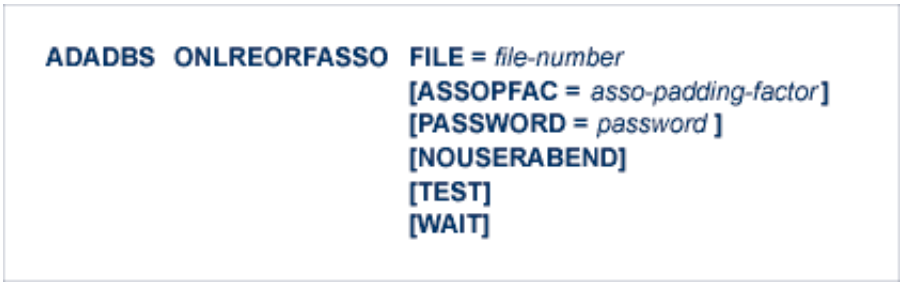
ADADBS NEWFIELD: Add New Field

```
ADADBS  NEWFIELD  FILE = file-number
                  [FNDEF = 'Adabas-field-definition ' ]
                  [NOUSERABEND]
                  [PASSWORD = 'password ' ]
                  [SUBFN = 'name  = parent-field ( begin , end )' ]
                  [SUPFN = 'name  = { parent-field ( begin , end ) } ...' ]
                  [TEST]
```

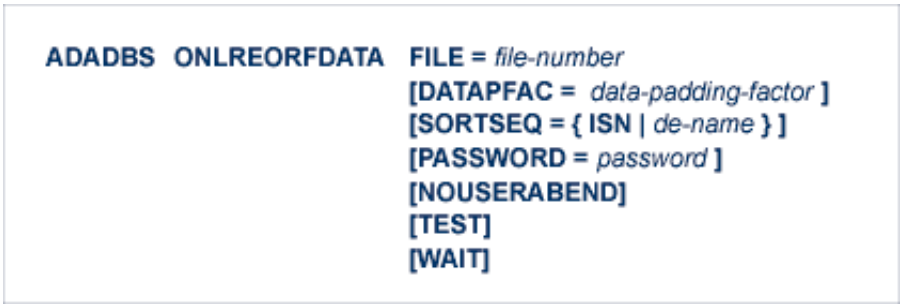
ADADBS ONLINVERT: Start Online Invert Process



ADADBS ONLREORFASSO: Start Online Reorder Associator for Files



ADADBS ONLREORFDATA: Start Online Reorder Data for Files



ADADBS ONLREORFILE: Start Online Reorder Associator and Data for Files

```
ADADBS  ONLREORFILE     FILE = file-number
                        [ASSOPFAC = asso-padding-factor]
                        [DATAPFAC = data-padding-factor ]
                        [SORTSEQ = { ISN | de-name } ]
                        [PASSWORD = password ]
                        [NOUSERABEND]
                        [TEST]
                        [WAIT]
```

ADADBS OPERCOM: Issue Adabas Operator Commands

```
ADADBS  OPERCOM  operator-command
                 [NOUSERABEND]
                 [NUCID = {nuc-id | 0 } ]
                 [TEST]
```

Operator Commands Supported by ADADBS OPERCOM

The following Adabas operator commands can be used in the ADADBS OPERCOM utility function:

- ```
  ADAEND [, GLOBAL ]
  ```

  This command terminates an Adabas session normally. No new users are accepted after this command has been issued. ET logic updating is continued until the end of the current logical transaction for each user. After all activity has been completed as described above, the Adabas session is terminated.

  In nucleus cluster environments, the GLOBAL option can be used to terminate the Adabas session in all active cluster nuclei.

- ```
  ALOCKF = file-number
  ```

  Lock a file in advance to ensure that an EXU, EXF, or UTI user will have exclusive control of the specified file. The advance-lock prevents new transactions from using the file. Once all current users have stopped using the file, the exclusive-control user has the lock. Until then, the exclusive-control user must wait.

To remove the advance lock without running the utility, see the RALOCKF command.

This command is not available in single user mode or for a read-only nucleus. It is available in cluster and non-cluster environments.

The following key points should be noted about advance-locks on files:

1. An advance-lock can be set while a file is being used.
2. A command requesting exclusive control (UTI, EXF, or EXU) over an advance-locked file will wait in the command queue until all other users stop using the file before it starts processing the file.
3. Advance-locks are automatically removed when a user gets exclusive control over the file. However, if a file is locked (via the LOCKF, LOCKU, or LOCKX commands), the locks are not removed when a user gets exclusive control over the file. (Locks must be explicitly removed, whereas advance-locks are automatically removed.)
4. Adabas will reject an advance-lock on a file that is already locked (via the LOCKF, LOCKU, or LOCKX commands) but will accept a lock request on an advance-locked file.
5. To ensure you have uninterrupted exclusive control over a file in a situation where you have multiple steps to run that require uninterrupted exclusive control while all steps have been processed, use a combination of advance-locking the file (ALOCKF), stopping all users of the file (STOPF), and locking the file (LOCKU). An example of this is given later in this section.
6. In the case of expanded files, an ALOCKF command is applied to the anchor file (representing the entire expanded file chain).
7. In a cluster environment, advance-locks are effective in all nuclei of the cluster.

Simple Example

In the following example, issuing the ALOCKF request to advance-lock file 32 ensures that file 32 will be available so the ADALOD UPDATE function can take exclusive control (via a UTI request) of the file for its processing:

```
ADADBS OPERCOM ALOCKF=32
ADALOD UPDATE FILE=32
```

Adabas processing proceeds in the following manner for these utility functions:

1. When the ADADBS OPERCOM ALOCKF request is submitted, file 32 is marked as advance-locked.
2. If there are any active users of file 32, the ADALOD UTI request cannot be granted immediately and will wait for the active users to end their transactions or sessions. Active users continue to issue commands against file 32. However, requests by new users for file 32 are rejected because of the advance-lock on the file.

3. When all active users of file 32 have ended their transactions or sessions, the ADALOD UTI request for exclusive control can be granted. Once exclusive control is established, ADALOD UPDATE processing can occur.

   As part of the successful execution of the ADALOD UTI request, the advance-lock is removed from the file. However, because ADALOD processing now has exclusive control of file 32, other users still cannot access it.

To accelerate the process and limit the wait time for the ADALOD UTI request, you can simply stop all active users of the file by force using the STOPF operator command:

```
ADADBS OPERCOM ALOCKF=32
ADADBS OPERCOM STOPF=32
ADALOD UPDATE FILE=32
```

In this case, the STOPF command will cause the nucleus to back out and stop users of file 32 before the ADALOD UTI request is granted. In addition, the advance-lock request specified by the ALOCKF command will prevent new users from accessing the file until the ADALOD UTI request for exclusive control is processed.

More Complex Example

An advance-lock set by ALOCKF cannot guarantee that multiple job steps in a series get uninterrupted exclusive control over a file, as the advance-lock is removed when the first step obtains exclusive control. Suppose an installation wants to run the following utility sequence:

```
ADAULD UNLOAD FILE=45
ADADBS REFRESH FILE=45
ADALOD UPDATE FILE=45
```

An ALOCKF request to advance-lock file 45 in this case would only work for the ADAULD UNLOAD function, because the ADAULD EXU request for exclusive control of file 45 would remove the advance-lock. If there are active users who try to issue commands against file 45, there is a chance that one of them will execute a command between the UNLOAD and REFRESH steps, or between the REFRESH and UPDATE steps. Such a user may also prevent the REFRESH or UPDATE step from obtaining exclusive control of file 45.

To ensure you have uninterrupted exclusive control over the file in this situation, use a combination of advance-locking the file (ALOCKF), stopping all users of the file (STOPF), and locking the file (LOCKU):

```
ADADBS OPERCOM ALOCKF=45
ADADBS OPERCOM STOPF=45
ADADBS OPERCOM LOCKU=45
ADAULD UNLOAD FILE=45
ADADBS REFRESH FILE=45
ADALOD UPDATE FILE=45
ADADBS OPERCOM UNLOCKU=45
```

In this example, Adabas processing proceeds in the following manner:

1. When the ADADBS OPERCOM ALOCKF request is submitted, file 45 is marked as advance-locked.

   The ADADBS OPERCOM STOPF request causes the nucleus to back out and stop users of file 45. (This step is optional.)

   The ADADBS OPERCOM LOCKU request locks the file more permanently than the ALOCKF request since the LOCKU lock will stay in effect until it is explicitly released.

2. If there are any active users updating file 45, the ADAULD EXU request cannot be granted immediately and will wait for the update users to end their transactions or sessions. Active users may continue to issue commands against file 45. However, requests by new users for file 45 are rejected because of the advance-lock on the file.

3. When all active users of file 45 have ended their transactions or sessions, the ADAULD request for exclusive-update (EXU) control can be granted. Once exclusive-update control is established, ADAULD UNLOAD processing can occur.

   As part of the successful execution of the ADAULD EXU request, the advance-lock is removed from the file. When ADAULD completes processing, it releases the EXU control of file 45. However, during and after the ADAULD execution, the LOCKU lock ensures that other users still cannot access the file.

4. The ADADBS utility will issue a UTI request for exclusive control of file 45, which will be granted. ADADBS REFRESH processing will then occur. When it completes, ADADBS will release exclusive control of file 45. However, the LOCKU lock ensures that other users still cannot access it.

5. The ADALOD utility will issue a UTI request for exclusive control of file 45, which will be granted. ADALOD UPDATE processing will then occur. When it completes, ADALOD will release exclusive control of file 45. However, the LOCKU lock ensures that other users still cannot access it.

6. The ADADBS OPERCOM UNLOCKU request explicitly unlocks file 45, making it available for other users.

- **CANCEL [, GLOBAL]**

  Cancel the Adabas session immediately. All command processing is immediately suspended. A pending autorestart is in effect which in turn causes the autorestart routine to be executed during the initialization of the next Adabas session.

  In nucleus cluster environments, the GLOBAL option can be used to cancel the Adabas session in all active cluster nuclei.

-

CLOGMRG = { YES | NO }

Switches automatic command log merging (ADARUN CLOGMRG parameter value) on or off in nucleus cluster environments.

The CLOGMRG command is global by definition and affects all nuclei in the cluster. If a NUCID is specified, it is ignored.

- **CT = *timeout-limit***

  Dynamically override the ADARUN CT parameter value; that is, the maximum number of seconds that can elapse from the time an Adabas command has been completed until the results are returned to the user through interregion communication (which depends on the particular operating system being used). The minimum setting is 1; the maximum is 16777215.

  In nucleus cluster environments, the CT command is global by definition and affects all nuclei in the cluster. If a NUCID is specified, it is ignored.

- **DAUQ**

  Display the user queue element (UQE) of each user who has executed at least one Adabas command within the last 15 minutes.

  In nucleus cluster environments, the NUCID=0 option can be used to display information for all active cluster nuclei. Information is displayed for each nucleus, one after the other.

- **DCQ**

  Display all posted command queue elements (CQEs). Each CQE’s user ID, job name, and buffer length is displayed.

  In nucleus cluster environments, the NUCID=0 option can be used to display information for all active cluster nuclei. Information is displayed for each nucleus, one after the other.

- **DDIB**

Display data integrity block (DIB). This block contains entries indicating which Adabas utilities are active and the resources being used by each utility. The DDIB function can be performed with either an active or an inactive nucleus.

In nucleus cluster environments, the information displayed by the DDIB command is global; the command can be run on any nucleus.

- **DDSF**

  Display Adabas Delta Save Facility (DSF) status. The Adabas nucleus displays the DSF status on the operator console as well as in the ADADBS job protocol.

  This function is only available if the nucleus is run with the parameter ADARUN DSF=YES.

  In nucleus cluster environments, the information displayed by the DDSF command is global; the command can be run on any nucleus.

- **DFILES= { *n* | *n1* ,..., *n5* }**

  Displays the number of access, update, EXU, and UTI users for the specified files. User types are totaled for each file, and are listed by file. Up to five files can be specified in this command. Up to 798 users are displayed.

  In nucleus cluster environments, the NUCID=0 option can be used to display information for all active cluster nuclei. Information is displayed for each nucleus, one after the other.

- **DFILUSE = *file-number***

  Displays the count of commands processed for the specified file so far during the current session.

  In nucleus cluster environments, the NUCID=0 option can be used to display information for all active cluster nuclei. Information is displayed for each nucleus, one after the other.

- **DHQA**

Display up to 1000 hold queue elements.

- **DLOCKF**

  Display locked files.

  In nucleus cluster environments, the NUCID=0 option can be used to display information for all active cluster nuclei. Information is displayed for each nucleus, one after the other.

- **DNC**

  Display the number of posted command queue elements (CQEs) waiting to be selected.

  In nucleus cluster environments, the NUCID=0 option can be used to display information for all active cluster nuclei. Information is displayed for each nucleus, one after the other.

- **DNH**

  Display the number of ISNs currently in the hold queue.

  In nucleus cluster environments, the NUCID=0 option can be used to display information for all active cluster nuclei. Information is displayed for each nucleus, one after the other.

- **DNU**

  Display the number of current users.

  In nucleus cluster environments, the NUCID=0 option can be used to display information for all active cluster nuclei. Information is displayed for each nucleus, one after the other.

- **DONLSTAT**

Note:
Not currently available for use with Adabas Parallel Services cluster nuclei.

Display status of each active reorder or invert online process together with the process ID.

In nucleus cluster environments, the NUCID=0 option can be used to display information for all active cluster nuclei. Information is displayed for each nucleus, one after the other.

-

  Display the Adabas session parameters currently in effect.

  Note:
  Additional Adabas add-on products and nucleus components will display additional parameters from those shown for a classic nucleus. For a sample of this report, please review the Adabas operator command documentation.

-

  Display the allocated pool space and the highest use level (’high water mark’) reached so far during the current session by record count and by percent for the following resources:

  - attached buffers (AB)
  - command queue (CQ)
  - format pool (FP)
  - hold queue (HQ)
  - pool for the table of ISNs (TBI)
  - pool for the table of sequential commands (TBQ or TBLES)
  - user queue (UQ)
  - unique descriptor pool (DUQPOOL)
  - security pool
  - user queue file list pool
  - work pool (WP)

- pool for global transaction IDs (XIDs; nonzero only with Adabas Transaction Manager)
- cluster block update redo pool (nonzero only for a cluster nucleus with ADARUN LRDP greater than zero)
- Work part 1 area (WKP1)

  Note:
  The maximum pool value of Work part 1 is derived from the LP parameter. It corresponds to the maximum number of blocks a transaction can spend on Work Part 1 before Adabas decides to back it out.
- Work part 2 area (WKP2)
- Work part 3 area (WKP3)

The actual values are displayed in nucleus message ADAN28.

-

  Display the current Adabas nucleus operating status.
-

  Display thread status.
-

  Display up to five active and inactive user queue elements.
-

  Display all user queue elements (UQEs).

  In nucleus cluster environments, the NUCID=0 option can be used to display information for all active cluster nuclei. Information is displayed for each nucleus, one after the other.

- **DUQE = X'*user-id*'**

  Display a user queue element for the specified Adabas-assigned user ID as follows:

  ```
  DUQE=X'A3CF2'
  ```

  The user ID must be entered in hexadecimal format. Do not use a job name for the user ID.

  In nucleus cluster environments, NUCID must always be specified because the user ID is not unique to the cluster.

- **DUUQE**

  Display utility user queue elements (UQEs).

  In nucleus cluster environments, the NUCID=0 option can be used to display information for all active cluster nuclei. Information is displayed for each nucleus, one after the other.

- **FEOFCL [, GLOBAL]**

  Close the current dual or multiple Command log and switch to the other dual or another multiple Command log. This command is valid only if dual or multiple command logging is in effect.

  In nucleus cluster environments, the GLOBAL option can be used to switch the dual or multiple command log in all cluster nuclei at the same time.

- **FEOFPL [, GLOBAL]**

  Close the current dual or multiple data Protection log and switch to the other dual or another multiple Protection log. This command is valid only if dual or multiple data protection logging is in effect.

  In nucleus cluster environments, the GLOBAL option can be used to switch the dual or multiple protection log in all cluster nuclei at the same time.

- **HALT [, GLOBAL]**

  Stop Adabas session. A BT (backout transaction) command is issued for each active ET logic user. The Adabas session is then terminated; no dumps are produced.

  In nucleus cluster environments, the GLOBAL option can be used to halt the Adabas session in all active cluster nuclei.

- **LOCKF =** *file-number*

  Lock the specified file. The specified file will be locked at all security levels.

- **LOCKU =** *file-number*

  Lock the specified file for all non-utility use. Adabas utilities can use the file normally.

- **LOCKX =** *file-number*

  Lock the specified file for all users except EXU or EXF users. EXU and EXF users can use the file normally. The lock is released automatically when an EXU user issues an OP command.

- **LOGGING**

  Start command logging.

- **LOG***xx*

Begin logging as indicated by *xx* for each command logged where *xx* is one of the following:

- CB - the Adabas control block
- FB - the Adabas format buffer
- IB - the Adabas ISN buffer
- IO - Adabas I/O activity
- RB - the Adabas record buffer
- SB - the Adabas search buffer
- UX - user data passed in the seventh parameter of the Adabas parameter list
- VB - the Adabas value buffer
- VOLIO - the extended I/O list for CLOGLAYOUT=5 and CLOGLAYOUT=8

- **LOGWARN = { *seconds* | 0 }**

  Use the LOGWARN command to specify how often the PLOG and CLOG status is checked and resulting alert messages are produced. Valid values range from zero (0) through 2147483647 seconds. The default is 0, indicating that no PLOG or CLOG status checking occurs and no corresponding alert messages are produced. If a non-zero value is specified for LOGWARN, a valid user exit 2 or user exit 12 must also be specified.

- **NOLOGGING**

  Stop or prevent command logging.

- **NOLOG*xx***

  Stop or prevent logging of *xx* where *xx* is one of the following:

  - CB - the Adabas control block
  - FB - the Adabas format buffer

- IB - the Adabas ISN buffer
- IO - Adabas I/O activity
- RB - the Adabas record buffer
- SB - the Adabas search buffer
- UX - user data passed in the seventh parameter of the Adabas parameter list
- VB - the Adabas value buffer
- VOLIO - the extended I/O list for CLOGLAYOUT=5 and CLOGLAYOUT=8

- **ONLRESUME = X'***identifier*'

  Note:
  Not currently available for use with Adabas Parallel Services cluster nuclei.

  Resume a previously suspended online reorder or invert process.

  In a cluster environment, NUCID must always be specified because the online process ID is not unique to the cluster.

- **ONLSTOP = X'***identifier*'

  Note:
  Not currently available for use with Adabas Parallel Services cluster nuclei.

  Stop an online reorder or invert process cleanly. The process continues up to its next interrupt point in order to produce a consistent state, and then terminates after performing all necessary cleanup.

  In a cluster environment, NUCID must always be specified because the online process ID is not unique to the cluster.

- **ONLSUSPEND = X'***identifier*'

  Note:
  Not currently available for use with Adabas Parallel Services cluster nuclei.

  Suspend an online reorder or invert process. The process continues up to its next interrupt point in order to produce a consistent state, performs a command throwback, and enters a state where it cannot be selected for processing. This command is useful if the online process is consuming too much of the nucleus resources.

In a cluster environment, NUCID must always be specified because the online process ID is not unique to the cluster.

- RALOCKF = *n*

  Note:
  Not currently available for use with Adabas Parallel Services cluster nuclei.

  Remove the advance lock on the specified file (see ALOCKF command) without running the utility.

  This command is available in cluster and non-cluster environments.

- RALOCKFA

  Note:
  Not currently available for use with Adabas Parallel Services cluster nuclei.

  Remove the advance lock on all files for which it has been set (see ALOCKF command) without running the utility.

  This command is available in cluster and non-cluster environments.

- RDUMPST

  Terminate online dump status. This command is normally used if online execution of the ADASAV utility has terminated abnormally.

- READONLY = { YES | NO }

  Note:
  Not currently available for use with Adabas Parallel Services cluster nuclei.

  Switches READONLY status on or off.

  In nucleus cluster environments, the READONLY command is global by definition and affects all nuclei in the cluster. If a NUCID is specified, it is ignored.

- **REVIEW = { NO | LOCAL | *hub-id* }**

  Note:
  Not currently available for use with Adabas Parallel Services cluster nuclei.

  Deactivate Adabas Review; change from hub mode to local mode; specify or change the Adabas Review hub with which a nucleus communicates.

- **STOPF = *file-number* [, PURGE]**

  Stop all users who are using the specified file. Any open transactions of the stopped users are backed out. Unless PURGE is also specified, a stopped user who returns (by sending a command) receives response code 9 (ADARSP009).

  If the optional PURGE parameter is specified, the stopped users are also deleted (their user queue elements are removed from the user queue).

  This command does not stop EXF or UTI users.

  The following is an example of using the PURGE parameter:

  ```
  ADADBS OPERCOM STOPF=5,PURGE
  ```

  Caution:
  If Adabas is running with ADARUN OPENRQ=NO (specifying that users are not required to issue an OP as the first command of the session), run the STOPF command with PURGE only if you are certain that the users to be deleted are no longer active. If a user with an open transaction is deleted, but then returns (by sending a command), no indication is given about the transaction backout. If the user continues the transaction, logical inconsistencies in the database could occur.

- **STOPI = *time* [, PURGE]**

  Stop all users who have not executed a command during the specified time interval (in seconds). Any open transactions of the stopped users are backed out. Unless PURGE is also specified, a stopped user who returns (by sending a command) receives response code 9 (ADARSP009).

  This command does not stop EXF or UTI users.

If the optional PURGE parameter is specified, the stopped users are also deleted (their user queue elements are removed from the user queue).

The following is an example of using the PURGE parameter:

```
ADADBS OPERCOM STOPI=3600,PURGE
```

Caution:
If Adabas is running with ADARUN OPENRQ=NO (specifying that users are not required to issue an OP as the first command of the session), run the STOPI command with PURGE only if you are certain that the users to be deleted are no longer active. If a user with an open transaction is deleted, but then returns (by sending a command), no indication is given about the transaction backout. If the user continues the transaction, logical inconsistencies in the database could occur.

- **STOPU = { X'***user-id***' | *job-name* }**

  Stop and delete the user with the Adabas-assigned user ID (in the form shown in the display commands), or stop and delete all users with the specified job name (*job-name*). Any open transaction by the stopped users will be backed out.

  Caution:
  If Adabas is running with ADARUN OPENRQ=NO (specifying that users are not required to issue an OP as the first command of the session), run the STOPU command only if you are certain that the users to be deleted are no longer active. If a user with an open transaction is deleted, but then returns (by sending a command), no indication is given about the transaction backout. If the user continues the transaction, logical inconsistencies in the database could occur.

  Note:
  The STOPU=X'*userid*' command is not allowed for online reorder or invert processes. See the ONLSTOP=X'*identifier*' command instead.

  The user ID must be specified in hexadecimal format; for example:

  ```
  STOPU=X'1CF2'
  ```

  In a cluster environment, NUCID must always be specified because the user ID is not unique to the cluster.

- **SYNCC**

Force resynchronization of all ET users on the nucleus. The nucleus waits for all ET users to reach ET status before continuing.

- **TNA***u* = *time*

  Set non-activity time limit (in seconds) for users where *u* is one of the following:

  - A - for access-only (ACC) users
  - E - for ET logic users
  - X - for exclusive control (EXF/EXU) users

  If specified, *time* must be a value greater than zero; it overrides the ADARUN value.

  In nucleus cluster environments, the TNAu commands are global by definition and affect all nuclei in the cluster. If a NUCID is specified, it is ignored.

- **TT** = *time*

  Set transaction time limit (in seconds) for ET logic users. If specified, this value must be greater than zero; it overrides the ADARUN value. In nucleus cluster environments, the TT command is global by definition and affects all nuclei in the cluster. If a NUCID is specified, it is ignored.

- **UNLOCKF** = *file-number*

  Unlock the specified file and restore its usage to the prelocked status.

- **UNLOCKU** = *file-number*

  Unlock the specified file for utility use and restore it to its prelocked status for non-utility users.

- **UNLOCKX** = *file-number*

Unlock the specified file and restore its usage to the prelocked status.

-

```
UTIONLY = { YES | NO }
```

Note:
Not currently available for use with Adabas Parallel Services cluster nuclei.

Switch UTIONLY status on or off.

In nucleus cluster environments, the UTIONLY command is global by definition and affects all nuclei in the cluster. If a NUCID is specified, it is ignored.

ADADBS PRIORITY: Change User Priority

```
ADADBS  PRIORITY  USERID = 'user-id'
                  [NOUSERABEND]
                  [PRTY = { n | 255 } ]
                  [TEST]
```

ADADBS REACTLOG: Reactivating Command Logging

```
ADADBS REACTLOG  [NOUSERABEND]
                 [TEST]
```

ADADBS RECORDSPANNING: Enable or Disable Record Spanning

```
ADADBS  RECORDSPANNING  FILE = file-number
                        MODE = { ON | OFF }
                        [PASSWORD = 'password' ]
                        [NOUSERABEND]
                        [TIMELIMIT = { 60 | timelimit } ]
                        [TEST]
```

ADADBS RECOVER: Recover Space

```
ADADBS  RECOVER  [NOUSERABEND]
                 [TEST]
```

ADADBS REFRESH: Set File to Empty Status

```
ADADBS  REFRESH  FILE = file-number
                 [NOUSERABEND]
                 [PASSWORD = 'password ' ]
                 [TEST]
```

ADADBS REFRESHSTATS: Reset Statistical Values

```
ADADBS  REFRESHSTATS [ALL]
                     [CMDUSAGE]
                     [COUNTERS]
                     [FILEUSAGE]
                     [NUCID = nucid ]
                     [NOUSERABEND]
                     [POOLUSAGE]
                     [THREADUSAGE]
```

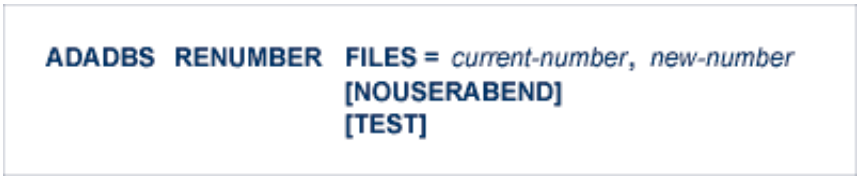
ADADBS RELEASE: Release Descriptor

```
ADADBS  RELEASE  FILE = file-number
                 DESCRIPTOR = 'name'
                 [NOUSERABEND]
                 [PASSWORD = 'password ' ]
                 [TEST]
```

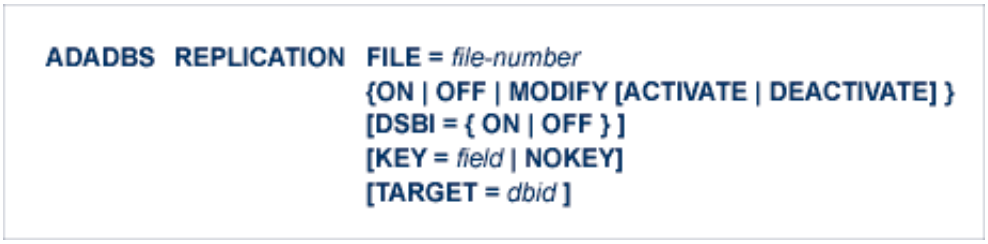
ADADBS RENAME: Rename a File or Database

```
ADADBS  RENAME   NAME = 'name'
                 [FILE = file-number ]
                 [NOUSERABEND]
                 [PASSWORD = 'password ' ]
                 [TEST]
```

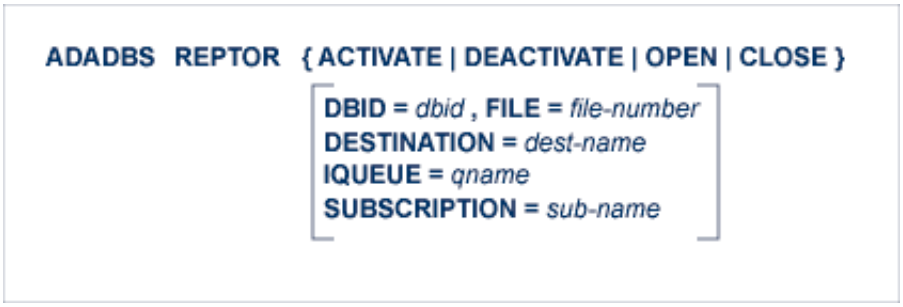
ADADBS RENUMBER: Change File Number



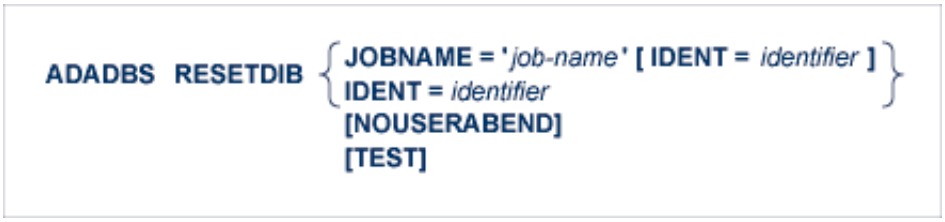
ADADBS REPLICATION: Activating or Deactivating Replication



ADADBS REPTOR: Activating, Deactivate, Open or Close an Event Replicator for Adabas Resource

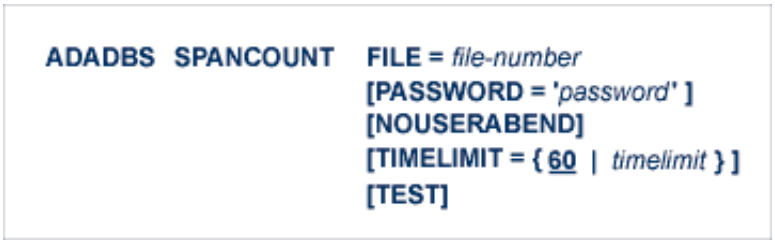


ADADBS RESETDIB: Reset Entries in Active Utility List

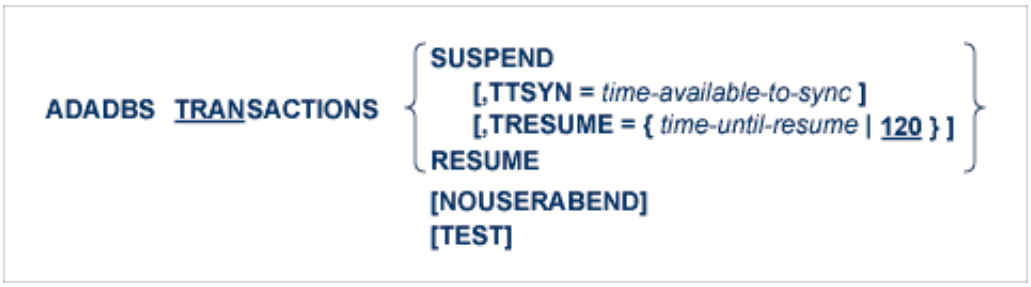


ADADBS RESETPPT: Reset PPT Blocks

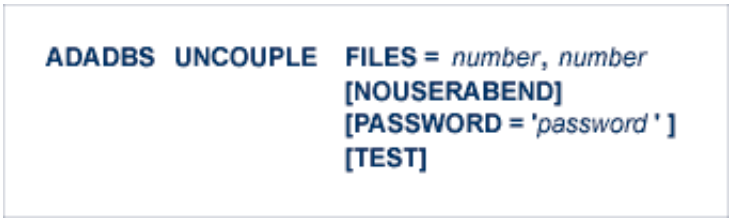
ADADBS SPANCOUNT: Count Spanned Records



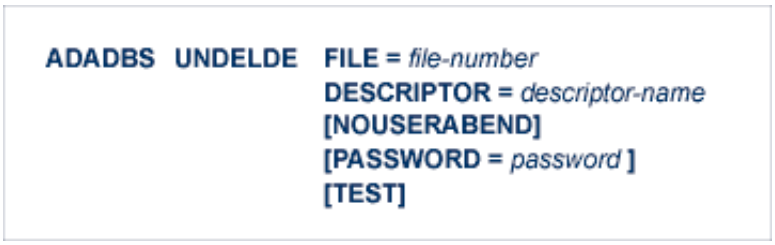
ADADBS TRANSACTIONS: Suspend and Resume Update Transaction Processing



ADADBS UNCOUPLE: Uncouple Files



ADADBS UNDELDE: Undelete a Logically Deleted Descriptor



ADADBS UNDELFN: Logically Undelete Fields

```
ADADBS UNDELFN  FILE = file-number
                FIELDLIST = ' field-list '
                [PASSWORD = password ]
                [NOUSERABEND]
                [TEST]
```

ADADCK DSCHECK: Check Data Storage and DSST

```
ADADCK DSCHECK  [FILE = { file [FROMRABN = DS-blknum ] [TORABN = DS-blknum ] | file - file } ]
                [MAXPISN = { num | 1000 } ]
                [NOOPEN]
                [NOUSERABEND]
                [REPAIR]
                [USAGE]
```

ADADEF DEFINE: Define a Database and Checkpoint File

ADADEF MODIFY: Change File Encodings

```
ADADEF  MODIFY  [FACODE = alpha-EBCDIC-key ]
                [FWCODE = wide-key ]
                [NOUSERABEND]
                [REPTOR = YES | NO ]
                [UACODE = alpha-ASCII-key ]
                [UES = { YES | NO } ]
                [UWCODE = wide-key ]
```

ADADEF NEWWORK: Define a Work File

```
ADADEF  NEWWORK  WORKSIZE = size
                 [NOUSERABEND]
                 [WORKDEV = device-type-list ]
```

ADAFRM: Format Adabas Database Components

To format the Associator (ASSO) or Data Storage (DATA) data sets, use this syntax:

```
ADAFRM  { ASSOFRM | DATAFRM }  SIZE = size
                               [DEVICE = device-type ]
                               [ { FROMRABN = {starting-rabn | NEXT } |
                                   NUMBER = {dataset-number | 1 } } ]
                               [NOUSERABEND]
```

To format the Work (WORK), command log (CLOG), protection log (PLOG), or sort (SORT) data sets, use this syntax:

```
ADAFRM  { WORKFRM | CLOGFRM | PLOGFRM | SORTFRM }
                    SIZE = size
                    [DEVICE = device-type ]
                    [ { FROMRABN = starting-rabn  |  NUMBER = {dataset-number | 1 } } ]
                    [NOUSERABEND]
```

To format the recovery log (RLOG), use this syntax:

```
ADAFRM  RLOGFRM  SIZE = size
                 [DEVICE = device-type ]
                 [NOUSERABEND]
```

To format the Temp (TEMP) data set, use this syntax:

```
ADAFRM  TEMPFRM  SIZE = size
                 [DEVICE = device-type ]
                 [FROMRABN =  starting-rabn ]
                 [NOUSERABEND]
```

To reformat blocks of an existing Associator, Data Storage, or Work data set, use this syntax:

```
ADAFRM  { ASSORESET | DATARESET | WORKRESET }
                    SIZE = size
                    FROMRABN = start-rabn
                    [NOUSERABEND]
```

ADAICK ACCHECK: Check Address Converter

```
ADAICK ACCHECK  FILE = file-number
                [NOOPEN]
                [NOUSERABEND]
```

ADAICK ASSOPRINT: Print/Dump Associator Blocks

```
ADAICK   ASSOPRINT   RABN = { rabn  |  rabn-rabn}
                     [NOUSERABEND]
```

ADAICK BATCH: Set Printout Width to 132 Characters Per Line

```
ADAICK   BATCH  [NOUSERABEND]
```

ADAICK DATAPRINT: Print/Dump Data Storage Blocks

```
ADAICK   DATAPRINT   RABN = { rabn  |  rabn-rabn}
                     [NOUSERABEND]
```

ADAICK DSCHECK: Print/Dump Content of Data Storage Record

```
ADAICK   DSCHECK  FILE = file-number
                  [ISN = isn-of-record ]
                  [NOOPEN]
                  [NOUSERABEND]
```

ADAICK DUMP: Activate Dump Print Format

```
ADAICK   DUMP [NOUSERABEND]
```

ADAICK FCBPRINT: Print/Dump File Control Block

```
ADAICK FCBPRINT FILE = file-number
                [NOOPEN]
                [NOUSERABEND]
```

ADAICK FDTPRINT: Print/Dump Field Definition Table

```
ADAICK FDTPRINT  FILE = file-number
                 [NOUSERABEND]
```

ADAICK GCBPRINT: Print/Dump General Control Blocks

```
ADAICK   GCBPRINT  [NOUSERABEND]
```

ADAICK ICHECK: Check Index Against Address Converter

```
ADAICK   ICHECK   FILE = {file-number | file-number-file-number }
                  [NOOPEN]
                  [NOUSERABEND]
```

ADAICK INT: Activate Interpreted Print Format

```
ADAICK INT [NOUSERABEND]
```

ADAICK NIPRINT: Print/Dump Normal Index

```
ADAICK   NIPRINT   FILE = file-number
                   [NOUSERABEND]
```

ADAICK NOBATCH: Set Print Width to 80 Characters Per Line

```
ADAICK   NOBATCH   [NOUSERABEND]
```

ADAICK NODUMP: Suppress Dump Print Format

```
ADAICK   NODUMP [NOUSERABEND]
```

ADAICK NOINT: Suppress Interpreted Format

```
ADAICK NOINT [NOUSERABEND]
```

ADAICK PPTPRINT: Print/Dump Parallel Participant Table

```
ADAICK   PPTPRINT   [NOUSERABEND]
```

ADAICK UIPRINT: Print/Dump Upper Index

```
ADAICK   UIPRINT   FILE = file-number
                   [NOUSERABEND]
```

ADAINV COUPLE: Define File-Coupling Descriptors

```
ADAINV  COUPLE  FILES = file-number1 , file-number2
                DESCRIPTOR = ' fieldname , fieldname '
                SORTSIZE = size
                TEMPSIZE = size
                [LPB = prefetch-buffer-size ]
                [LWP = { workpool-size | 1048576 } ]
                [NOUSERABEND]
                [PASSWORD = ' password ' ]
                [SORTDEV = device-type ]
                [TEMPDEV = device-type ]
                [TEST]
```

ADAINV INVERT: Create Descriptors

```
ADAINV  INVERT  FILES = file-num
                SORTSIZE = size
                TEMPSIZE = size
                [FIELD = ' field-name [ , option ]... ' ] ...
                [COLDE = ' num , name [ , UQ [, XI ] ] = parent-field ' ]
                [HYPDE = ' num , name , length , format [ , option ]... = parent-field , ...' ]
                [PHONDE = ' name ( field-name )' ]
                [SUBDE = ' name [, UQ [, XI ] ] = parent-field ( begin , end )' ]
                [SUPDE = ' name [, UQ [, XI ] ] = { parent-field ( begin , end ) } , ...' ]
                [CODE = cipher-code ]
                [LPB = prefetch-buffer-size ]
                [LWP = { workpool-size | 1048576 } ]
                [NOUSERABEND]
                [PASSWORD = ' password ' ]
                [SORTDEV = device-type ]
                [TEMPDEV = device-type ]
                [TEST]
```

ADALOD LOAD: Load a File

```
ADALOD LOAD  FILE = file-number [ , filetype ]
             DSSIZE = size
             MAXISN = max-number-of-records [MAXISN2 = max-number-of-secondary-spanned-records ]
             SORTSIZE = size
             TEMPSIZE = size
             [ACRABN = starting-rabn ] [AC2RABN = starting-rabn ]
             [ADAMFILE ADAMDE = { field | ISN } [ADAMOFLOW = size ] [ADAMPARM = { number | 0 } ] ]
             [ALLOCATION = { FORCE | NOFORCE } ]
             [ANCHOR = file-number MINISN = lowest-allocated-isn , NOACEXTENSION ]
             [ASSOPFAC = { padding-factor | 10 } ] [ASSOVOLUME = ' Associator-extent-volume ' ]
             [{BASEFILE | LOBFILE} = file-number ]
             [DATAFRM = { YES | NO }]
             [DATAPFAC = { padding-factor | 10 } ] [DATAVOLUME = ' Data-Storage-extent-volume ' ]
             [DSDEV = device-type ] [DSRABN = start-rabn ] [DSREUSE = { YES | NO }]
             [ETID = owner-id ]
             [IGNFDT]
             [INDEXCOMPRESSION = { YES | NO }]
             [ISNREUSE = { YES | NO ] [ISNSIZE = { 3 | 4 } ]
             [LIP = { isn-pool-size | 2000 } ]
             [LOWNERID = { owner-id-length | 0 } ]
             [LWP = { work-pool-size | 1048576 } ]
             [MAXDS = max-DS-secondary-allocation ]
             [MAXNI = max-NI-secondary-allocation ]
             [MAXRECL = max-compressed-record-length ]
             [MAXUI= max-UI-secondary-allocation ]
             [MINISN = { lowest-allocated-isn | 1 } ]
             [MIXDSDEV]
             [NAME = { name | TESTFILE } ]
             [NIRABN = start-rabn ] [NISIZE = size ]
             [NOACEXTENSION]
             [NOUSERABEND]
             [NUMREC = max-number-of-records-to-load ]
             [PGMREFRESH = { YES | NO } ]
             [READONLY = { YES | NO } ]
             [REPLICATOR]
             [RESTART]
               RPLTARGETID = ' reptor-target-id '
                  [RPLDSBI]
                  [RPLERRORDEACTFILE = { NO | YES} ]
                  [RPLINITERROR = { FAIL | CONTINUE} ]
                  [RPLKEY = ' primary-key-for-replication ' ]
                  [RPLLOAD = { YES | FILE | NO } ]
             [RPLUPDATEONLY = { YES | NO } ]
             [SKIPREC = { number | 0 } ]
             [SLOG]
             [SORTDEV = device-type ]
             [SYFMAXUV = nn ]
             [TEMPDEV = device-type ]
             [TEST]
             [UIRABN = start-rabn ] [UISIZE= size ]
             [UQDE = descriptor-list ]
             [USERISN = { YES | NO } ]
             [VERSION = { 4 | 5 | 6 | 7 } ]
```

ADALOD UPDATE: Add/Delete Records

```
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-rabn]
                     [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 } ]
```

ADAMER Utility: ADAM Estimation

```
ADAMER  ADAMDE = { descriptor | ISN }
        MAXISN = maximum-number-of-records
        [BITRANGE = { minimum | 0 } {, maximum | 18 } {, increment | 2 } ]
        [DATADEV = device-type ]
        [DATAPFAC = padding-factor ]
        [DATASIZE = minimum , maximum [, increment ] ]
        [NOUSERABEND]
        [NUMREC = number-of-records ]
```

ADAORD REORASSO: Reorder Associator

```
ADAORD  REORASSO  [DBINDEXCOMPRESSION = { YES | NO } ]
                  [FILE = file-number ]
                      [ACRABN = starting-rabn ]
                      [AC2RABN = starting-rabn ]
                      [ALLOCATION = { FORCE | NOFORCE } ]
                      [ASSOPFAC = padding-factor  ]
                      [ASSOVOLUME = 'Associator-extent-volume ' ]
                      [INDEXCOMPRESSION = { YES | NO } ]
                      [ISNSIZE = { 3 | 4 } ]
                      [MAXISN = highest-isn ]
                      [MAXISN2 = highest-isn ]
                      [NIRABN = starting-rabn ]
                      [NIRELEASE]
                      [NISIZE = size ]
                      [UIRABN = starting-rabn ]
                      [UIRELEASE]
                      [UISIZE = size ]
                  [LPB = prefetch-buffer-size ]
                  [MAXFILES = maximum-number-files ]
                  [NEWDBID = database-identifier ]
                  [NEWDBNAME = database-name]
                  [NOUSERABEND]
                  [RAID]
                  [RPLUPDATEONLY = { YES | NO ]
                  [TEST]
```

ADAORD REORDATA: Reorder Data Storage

```
ADAORD  REORDATA  [FILE = file-number ]
                      [ALLOCATION = { FORCE | NOFORCE } ]
                      [DATAPFAC = padding-factor  ]
                      [DATAVOLUME = 'Data-Storage-extent-volume' ]
                      [DSDEV = device-type ]
                      [DSRABN = starting-rabn ]
                      [DSRELEASE]
                      [DSSIZE =size ]
                      [MAXRECL = record-length ]
                      [SORTSEQ = { descriptor | ISN ]
                  [LIP = { isn-pool-size  |  16384 } ]
                  [LPB = prefetch-buffer-size ]
                  [NOUSERABEND]
                  [RAID]
                  [TEST]
```

ADAORD REORDB: Reorder Database

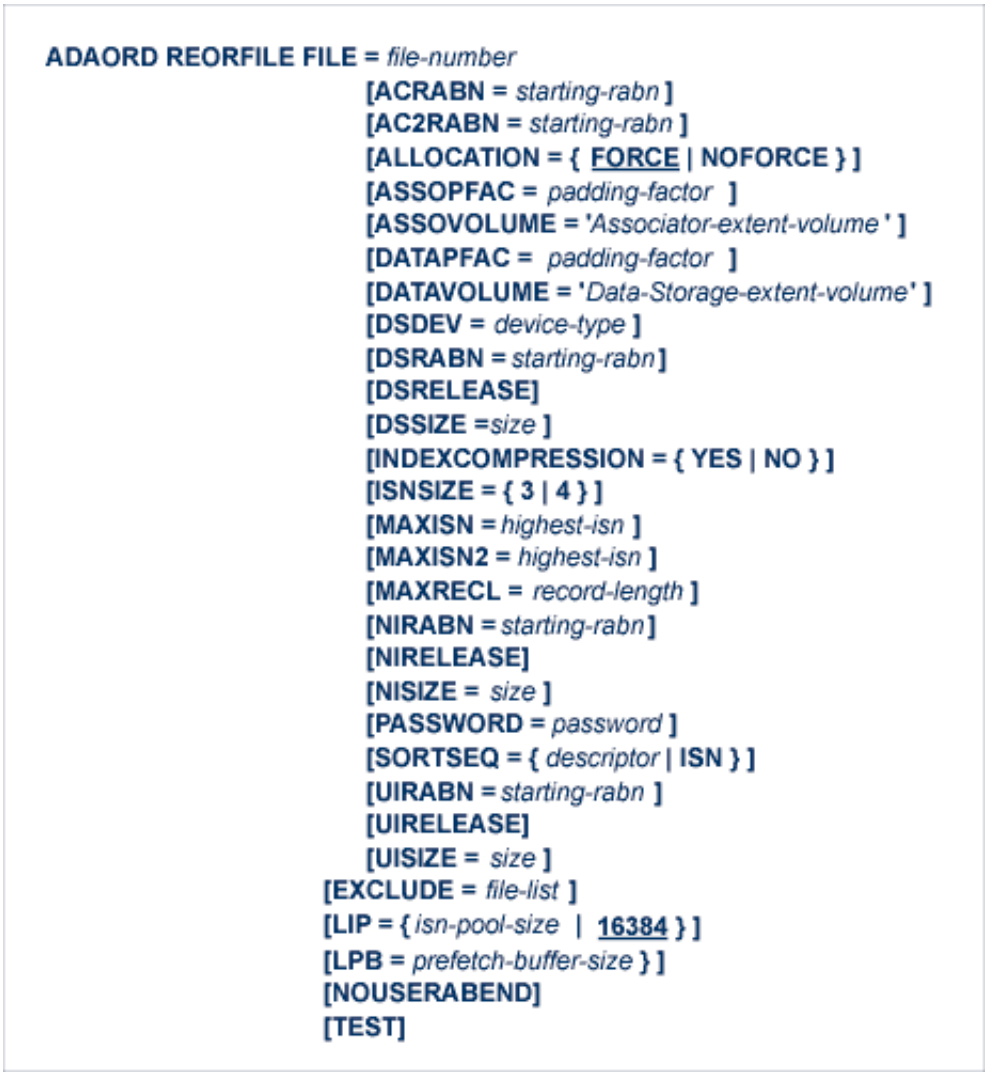
ADAORD REORFASSO: Reorder Associator for a Single File

```
ADAORD REORFASSO FILE = file-number
                        [ACRABN = starting-rabn ]
                        [AC2RABN = starting-rabn ]
                        [ALLOCATION = { FORCE | NOFORCE } ]
                        [ASSOPFAC = padding-factor ]
                        [ASSOVOLUME = 'Associator-extent-volume ' ]
                        [INDEXCOMPRESSION = { YES | NO } ]
                        [ISNSIZE = { 3 | 4 } ]
                        [MAXISN = highest-isn ]
                        [MAXISN2 = highest-isn ]
                        [NIRABN = starting-rabn ]
                        [NIRELEASE]
                        [NISIZE = size ]
                        [PASSWORD = password ]
                        [UIRABN = starting-rabn ]
                        [UIRELEASE]
                        [UISIZE = size ]
                    [EXCLUDE = file-list ]
                    [LPB = prefetch-buffer-size ]
                    [NOUSERABEND]
                    [TEST]
```

ADAORD REORFDATA: Reorder Data Storage for a Single File

```
ADAORD REORFDATA FILE = file-number
                        [ALLOCATION = { FORCE | NOFORCE } ]
                        [DATAPFAC = padding-factor ]
                        [DATAVOLUME = 'Data-Storage-extent-volume' ]
                        [DSDEV = device-type ]
                        [DSRABN = starting-rabn ]
                        [DSRELEASE]
                        [DSSIZE =size ]
                        [MAXRECL = record-length ]
                        [PASSWORD = password ]
                        [SORTSEQ = { descriptor | ISN } ]
                    [EXCLUDE = file-list ]
                    [LIP = { isn-pool-size | 16384 } ]
                    [LPB = prefetch-buffer-size ]
                    [NOUSERABEND]
                    [TEST]
```

ADAORD REORFILE: Reorder File



ADAORD RESTRUCTUREDB: Restructure Database

```
ADAORD  { RESTRUCTUREDB | REDB }
                [DBASSODEV = { device-type ]
                [DBDATADEV = { device-type ]
                [DBINDEXCOMPRESSION = { YES | NO } ]
                [FILE = file-number ]
                    [ASSOPFAC = padding-factor ]
                    [ASSODEV = device-type ] [DATADEV = device-type ]
                    [DATAPFAC = padding-factor ]
                    [INDEXCOMPRESSION = { YES | NO } ]
                    [ISNSIZE = { 3 | 4 } ]
                    [SORTSEQ = { descriptor | ISN } ]
                [LPB = prefetch-buffer-size ]
                [NOUSERABEND]
                [TEST]
```

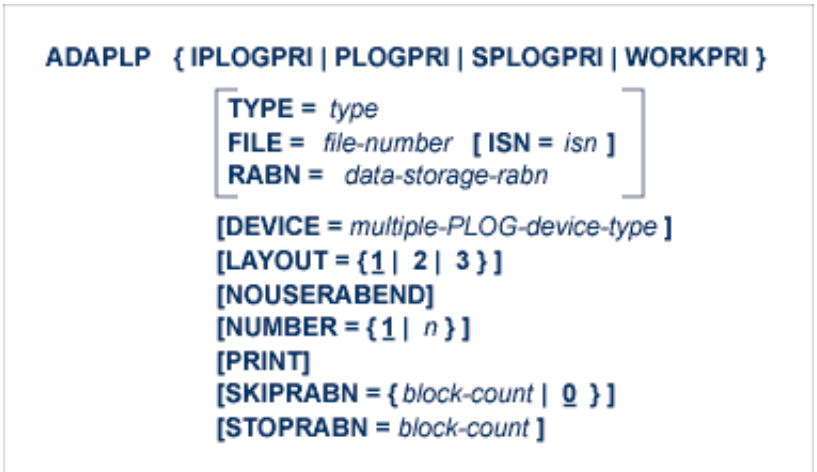
ADAORD RESTRUCTUREF: Restructure Single Files

```
ADAORD  { RESTRUCTUREF | REF }
                FILE = file-number
                    [ASSODEV = device-type ]
                    [ASSOPFAC = padding-factor ]
                    [DATADEV = device-type ]
                    [DATAPFAC = padding-factor ]
                    [DBASSODEV = device-type ]
                    [DBDATADEV = device-type ]
                    [INDEXCOMPRESSION = { YES | NO } ]
                    [ISNSIZE = { 3 | 4 } ]
                    [PASSWORD = password ]
                    [SORTSEQ = { descriptor | ISN } ]
                [LPB = prefetch-buffer-size } ]
                [NOUSERABEND]
                [TEST]
```

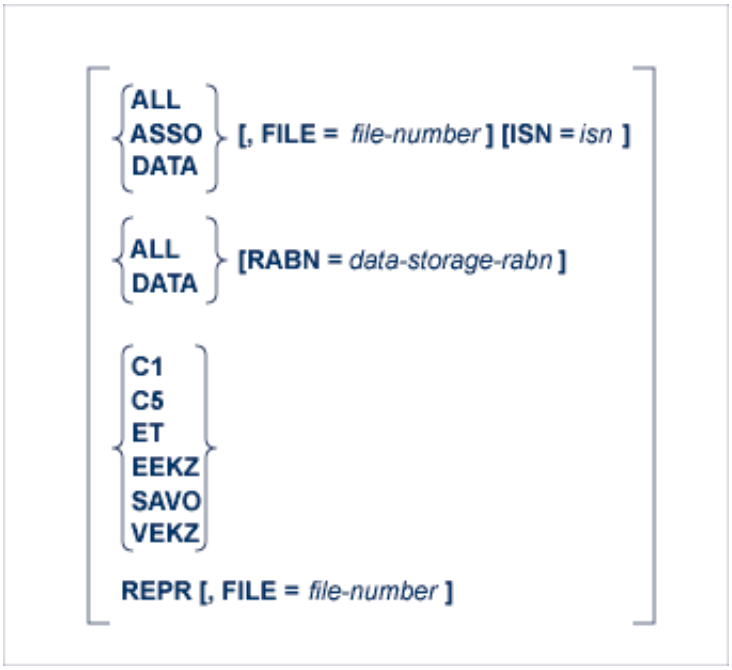
ADAORD STORE: Store Files

```
ADAORD STORE  [ALLFILES]
              [CHECKPOINT]
              [FILE = file-number ]
                 [ACRABN = starting-rabn ]
                 [AC2RABN = starting-rabn ]
                 [ALLOCATION = { FORCE | NOFORCE } ]
                 [ASSOPFAC = padding-factor ]
                 [ASSOVOLUME = ' Associator-extent-volume ' ]
                 [DATAPFAC = padding-factor ]
                 [DATAVOLUME = ' Data-Storage-extent-volume ' ]
                 [DSDEV = device-type ]
                 [DSRABN = starting-rabn ]
                 [DSRELEASE]
                 [DSSIZE = size ]
                 [INDEXCOMPRESSION = { YES | NO } ]
                 [ISNSIZE = { 3 | 4 } ]
                 [MAXISN = highest-isn ]
                 [MAXISN2 = highest-isn ]
                 [MAXRECL = record-length ]
                 [NIRABN = starting-rabn ]
                 [NIRELEASE]
                 [NISIZE = size ]
                 [READONLY = { YES | NO } ]
                 [RPLDSBI = { YES | NO } ]
                 [RPLKEY = { primary-key | OFF } ]
                 [RPLTARGETID = { target-ID | OFF | 0 } ]
                 [RPLUPDATEONLY = { YES | NO } ]
                 [UIRABN = starting-rabn ]
                 [UIRELEASE]
                 [UISIZE = size ]
              [EXCLUDE = file-list ]
              [LIP = { isn-pool-size | 16384 } ]
              [NOUSERABEND]
              [OVERWRITE]
              [REPLICATOR]
              [SECURITY]
              [SLOG]
              [TEST]
              [TRIGGER]
```

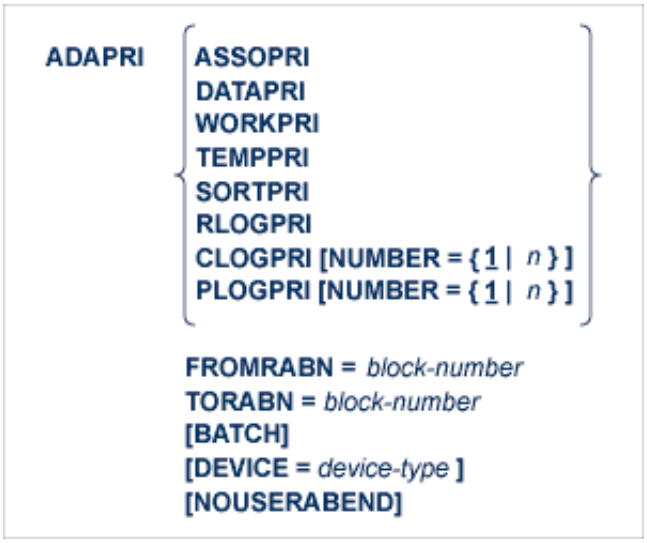
ADAPLP: Print Data Protection Records from PLOG/Work



where *type* is one of the following:



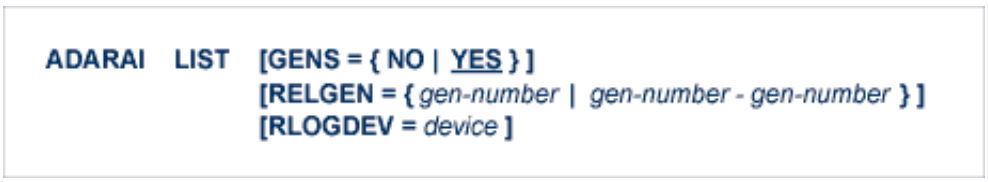
ADAPRI: Print Selected Adabas Blocks



ADARAI CHKDB: Check the Database Recovery Status

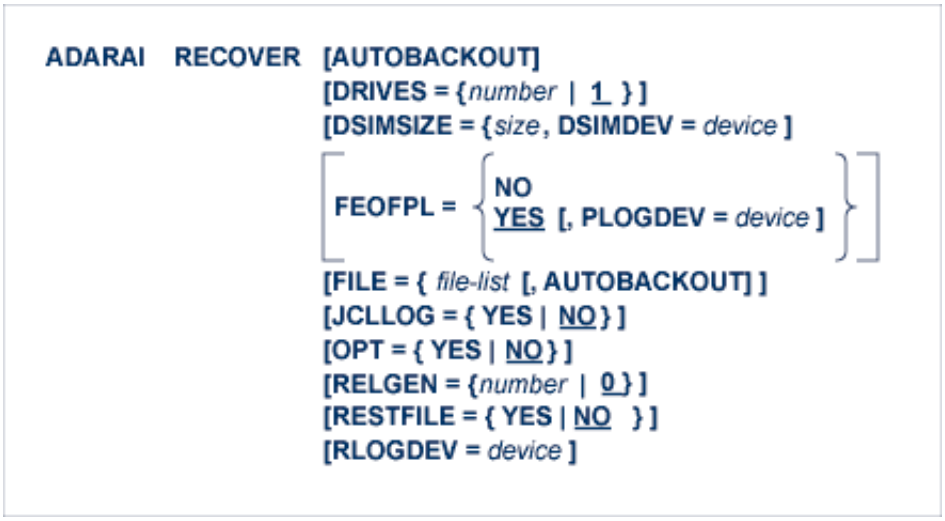
ADARAI DISABLE: Deactivate Recovery Logging

ADARAI LIST: Display Current RLOG Generations



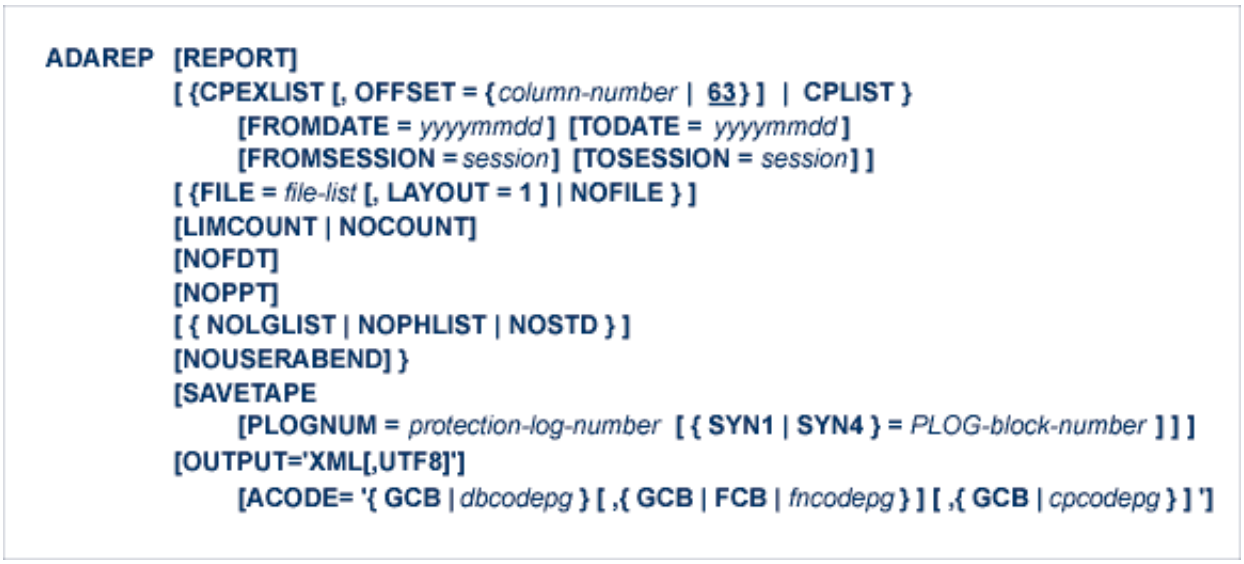
ADARAI PREPARE: Initialize and Start the RLOG

ADARAI RECOVER: Build a Recovery Job Stream

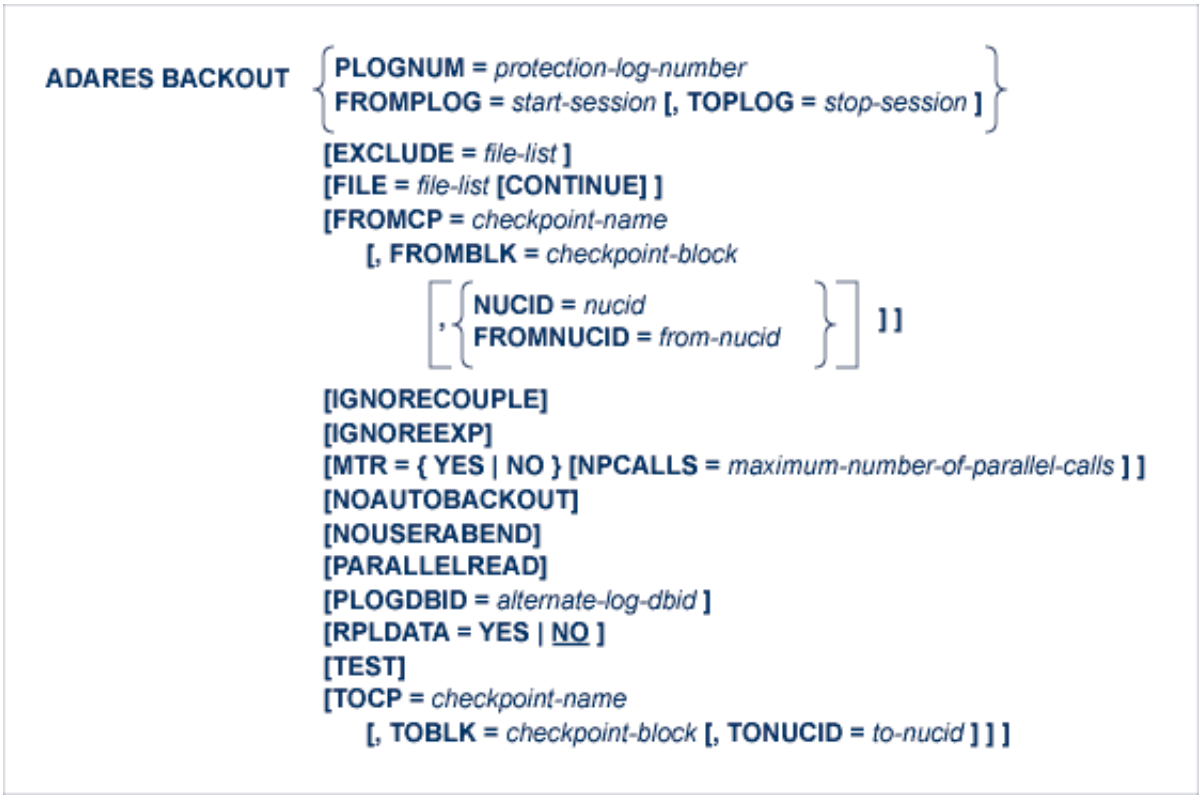


ADARAI REMOVE: Remove the Recovery Aid

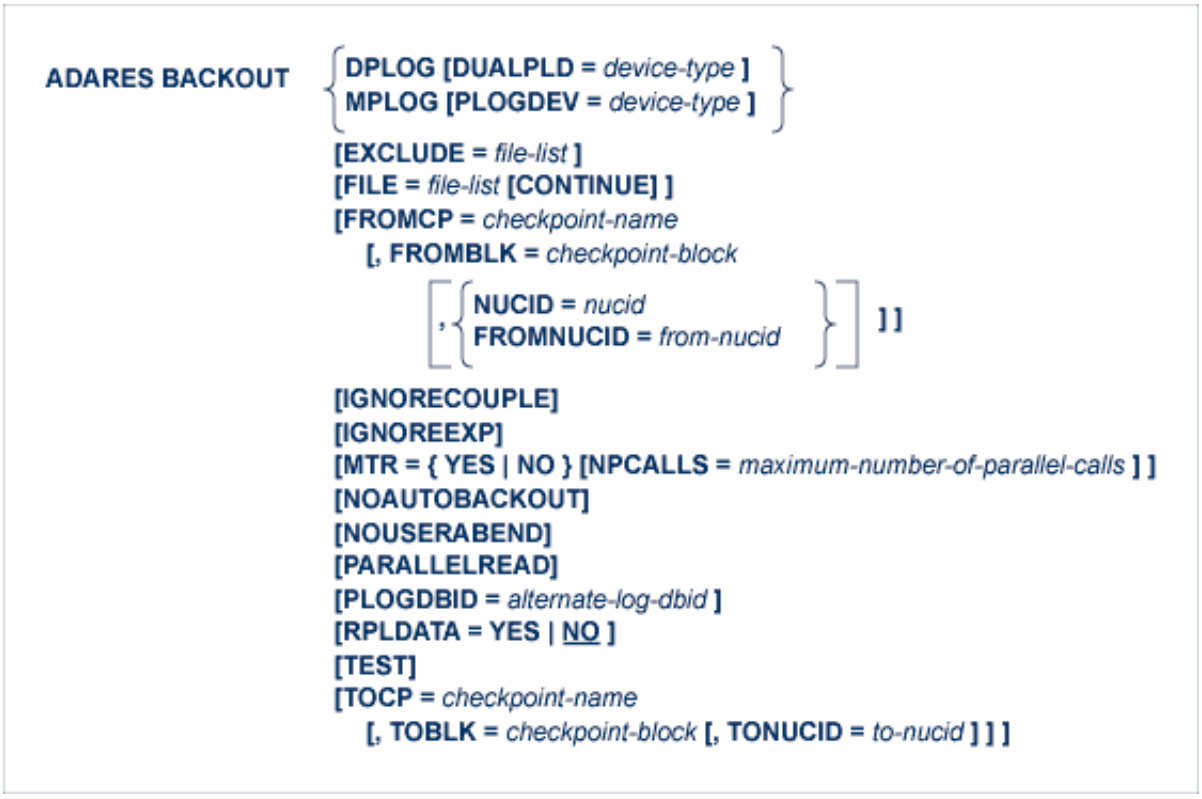
ADAREP: Request Database Status Report



ADARES BACKOUT SIBA: Back Out Updates Using the Sequential Protection Log (SIBA)



ADARES BACKOUT DPLOG or MPLOG: Back Out Updates Using the Dual or Multiple Protection Log



ADARES CLCOPY: Copy Dual Command Log

```
ADARES  CLCOPY  [DUALCLD = device-type ]
                [NOUSERABEND]
                [OPENOUT]
                [TEST]
                [TWOCOPIES]
```

ADARES COPY: Copy a Sequential Protection Log or Save Tape

```
ADARES  COPY    [PLOGNUM = protection-log-number
                FROMPLOG = start-session  [, TOPLOG = stop-session ]]
                [NOUSERABEND]
                [OPENOUT]
                [RLOGDEV = device-type ]
                [TEST]
                [TWOCOPIES]
                [UTICPLIST]
```

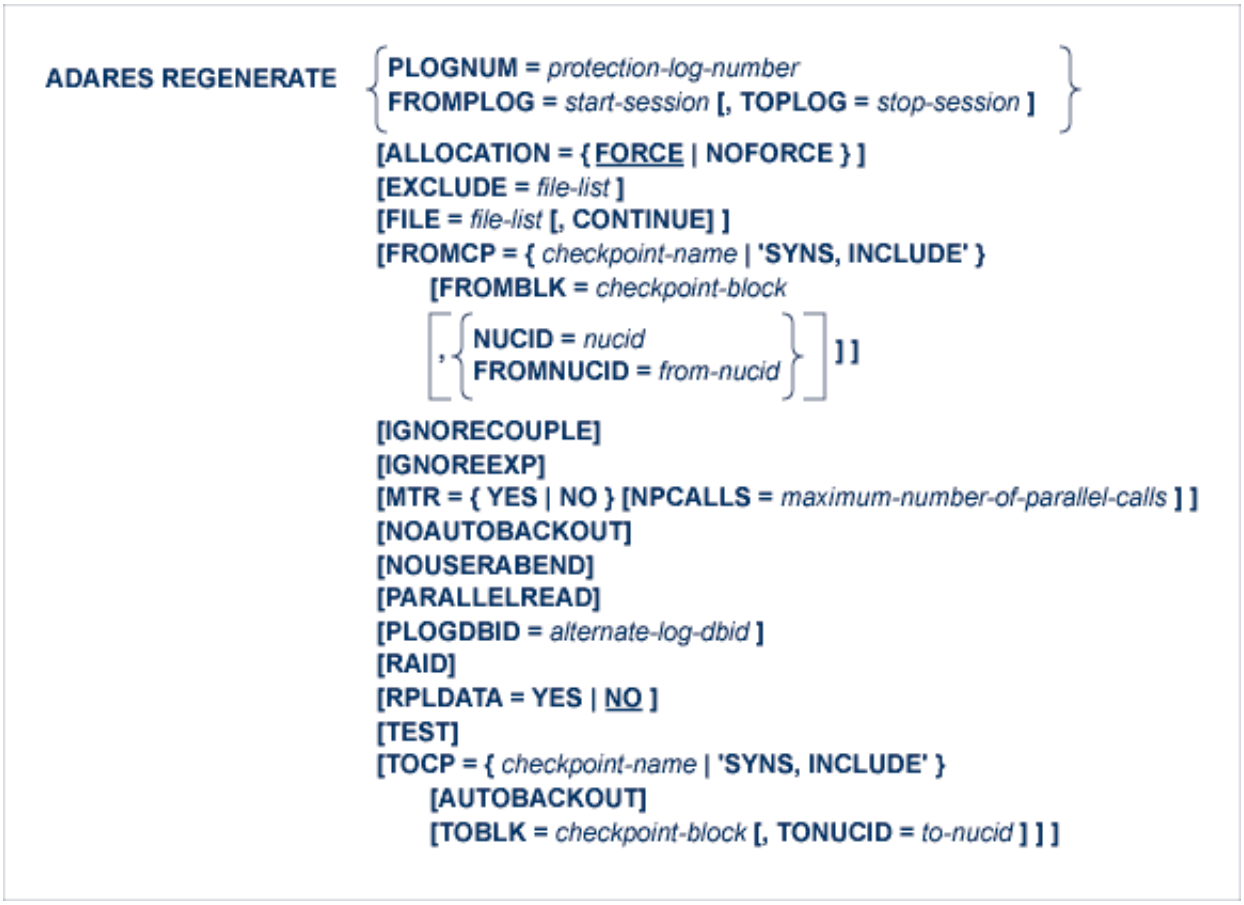
ADARES MERGE CLOG: Merge Nucleus Cluster Command Logs

```
ADARES  MERGE  CLOG, NUMLOG = nn
```

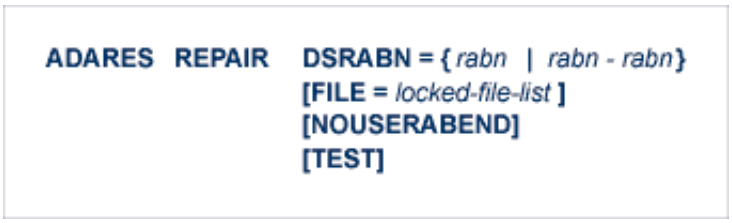
ADARES PLCOPY: Copy Protection Log to Sequential Data Set

```
ADARES  PLCOPY  [PLOGDEV = device-type ]
                [NOPPT]
                [NOUSERABEND]
                [OPENOUT]
                [RLOGDEV = device-type ]
                [SBLKNUM = starting-block-num ]
                [TEST]
                [TWOCOPIES]
                [UTICPLIST]
```

ADARES REGENERATE: Regenerate Updates



ADARES REPAIR: Repair Data Storage Blocks



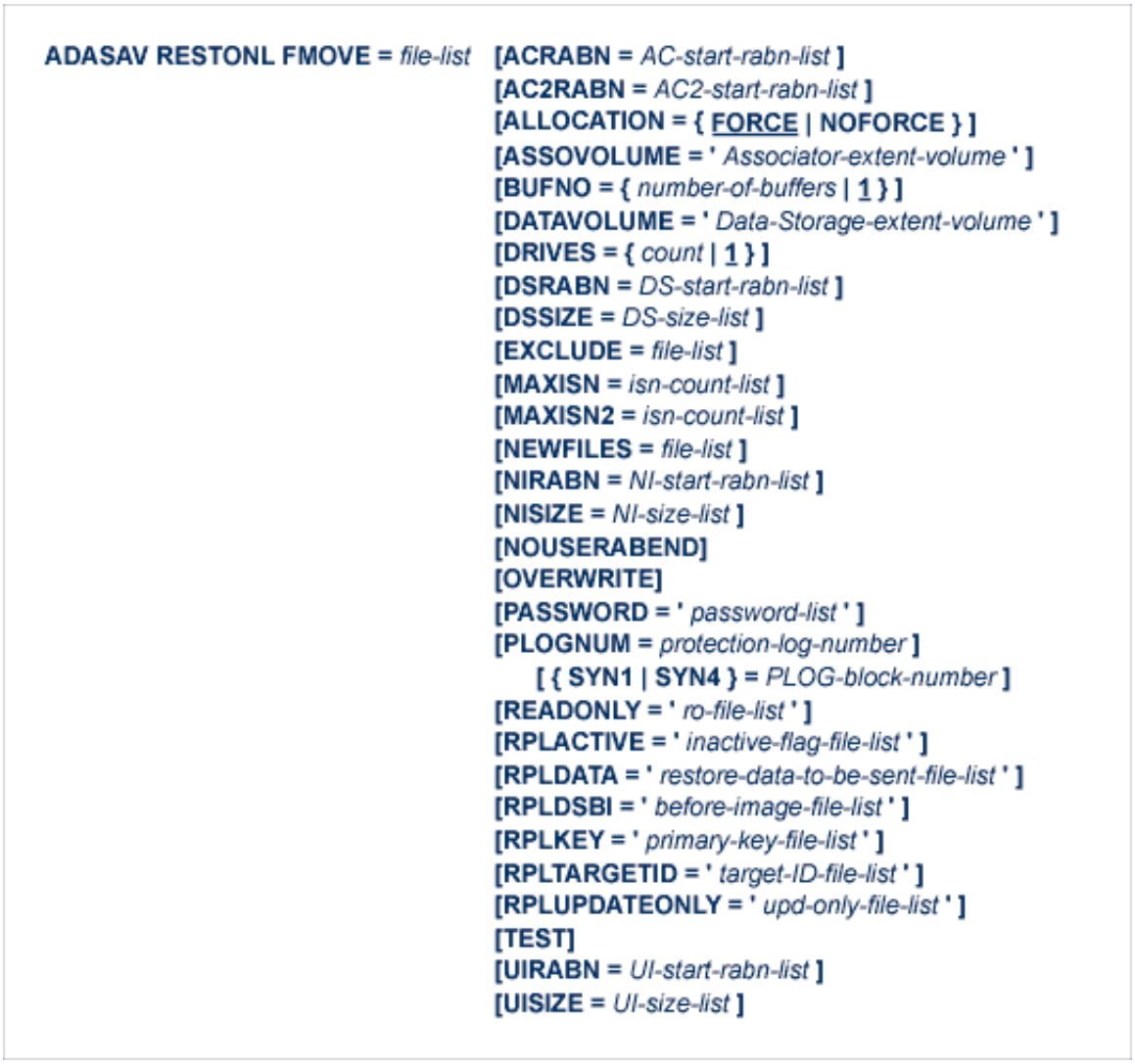
ADASAV RESTONL: Restore Database from Online Source

```
ADASAV RESTONL  [BUFNO = { number-of-buffers | 1 } ]
                [CLOGDEV = CLOG1-device-type ]
                [DRIVES = { count | 1 } ]
                [EXCLUDE = file-list ]
                [NEWDBID = new-database-id ]
                [NEWDBNAME = new-database-name ]
                [NOUSERABEND]
                [OVERWRITE]
                [PLOGDEV = PLOG-device-type ]
                [PLOGNUM = PLOG-number [, SYN1 = PLOG-block-number ] ]
                [TEST]
```

ADASAV RESTONL FILES: Restore Files to Original RABNs from Online Source

```
ADASAV RESTONL FILES = file-list  [ALLOCATION = { FORCE | NOFORCE } ]
                                  [BUFNO = { number-of-buffers | 1 } ]
                                  [DRIVES = { count | 1 } ]
                                  [EXCLUDE = file-list ]
                                  [NEWFILES = file-list ]
                                  [NOUSERABEND]
                                  [OVERWRITE]
                                  [PASSWORD = ' password-list ' ]
                                  [PLOGNUM = protection-log-number ]
                                      [ { SYN1 | SYN4 } = PLOG-block-number ]
                                  [READONLY = ' ro-file-list ' ]
                                  [RPLACTIVE = ' inactive-flag-file-list ' ]
                                  [RPLDATA = ' restore-data-to-be-sent-file-list ' ]
                                  [RPLDSBI = ' before-image-file-list ' ]
                                  [RPLKEY = ' primary-key-file-list ' ]
                                  [RPLTARGETID = ' target-ID-file-list ' ]
                                  [RPLUPDATEONLY = ' upd-only-file-list ' ]
                                  [TEST]
```

ADASAV RESTONL FMOVE: Restore Files to Any RABNs from Online Source



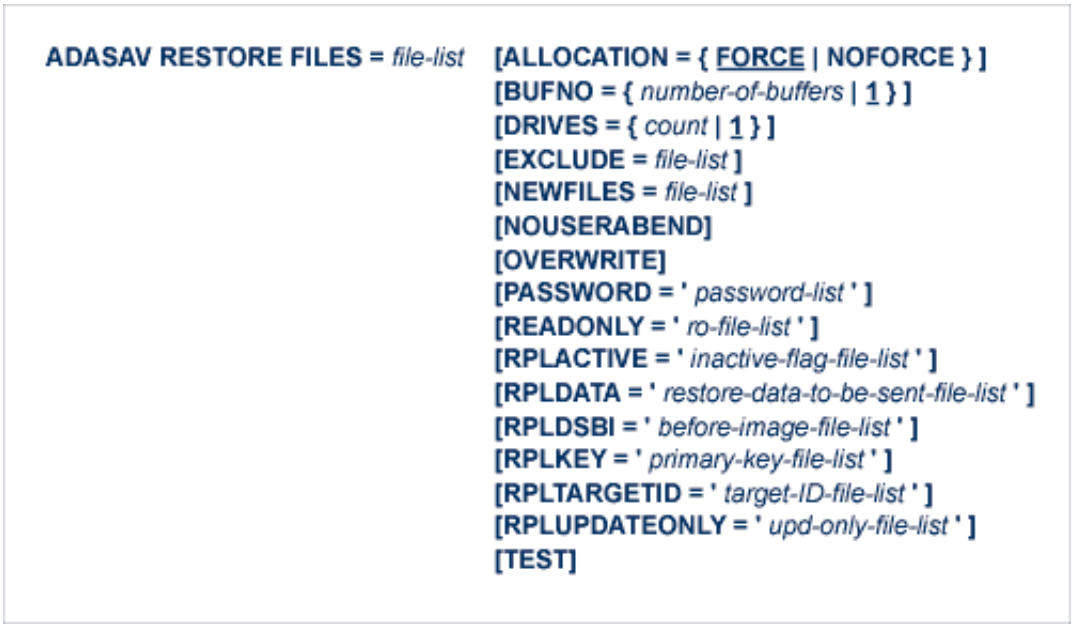
ADASAV RESTONL GCB: Restore Database Incremental from Online Source

```
ADASAV RESTONL GCB  [BUFNO = { number-of-buffers | 1 } ]
                    [CLOGDEV = CLOG1-device-type ]
                    [DRIVES = { count | 1 } ]
                    [EXCLUDE = file-list ]
                    [FILES = file-list ]
                    [NEWDBID = new-database-id ]
                    [NEWDBNAME = new-database-name ]
                    [NOUSERABEND]
                    [OVERWRITE]
                    [PLOGDEV = PLOG-device-type ]
                    [PLOGNUM = PLOG-number [, SYN1 = PLOG-block-number ] ]
                    [READONLY = ' ro-file-list ' ]
                    [RPLACTIVE = ' inactive-flag-file-list ' ]
                    [RPLDATA = ' restore-data-to-be-sent-file-list ' ]
                    [RPLDSBI = ' before-image-file-list ' ]
                    [RPLKEY = ' primary-key-file-list ' ]
                    [RPLTARGETID = ' target-ID-file-list ' ]
                    [RPLUPDATEONLY = ' upd-only-file-list ' ]
                    [TEST]
```

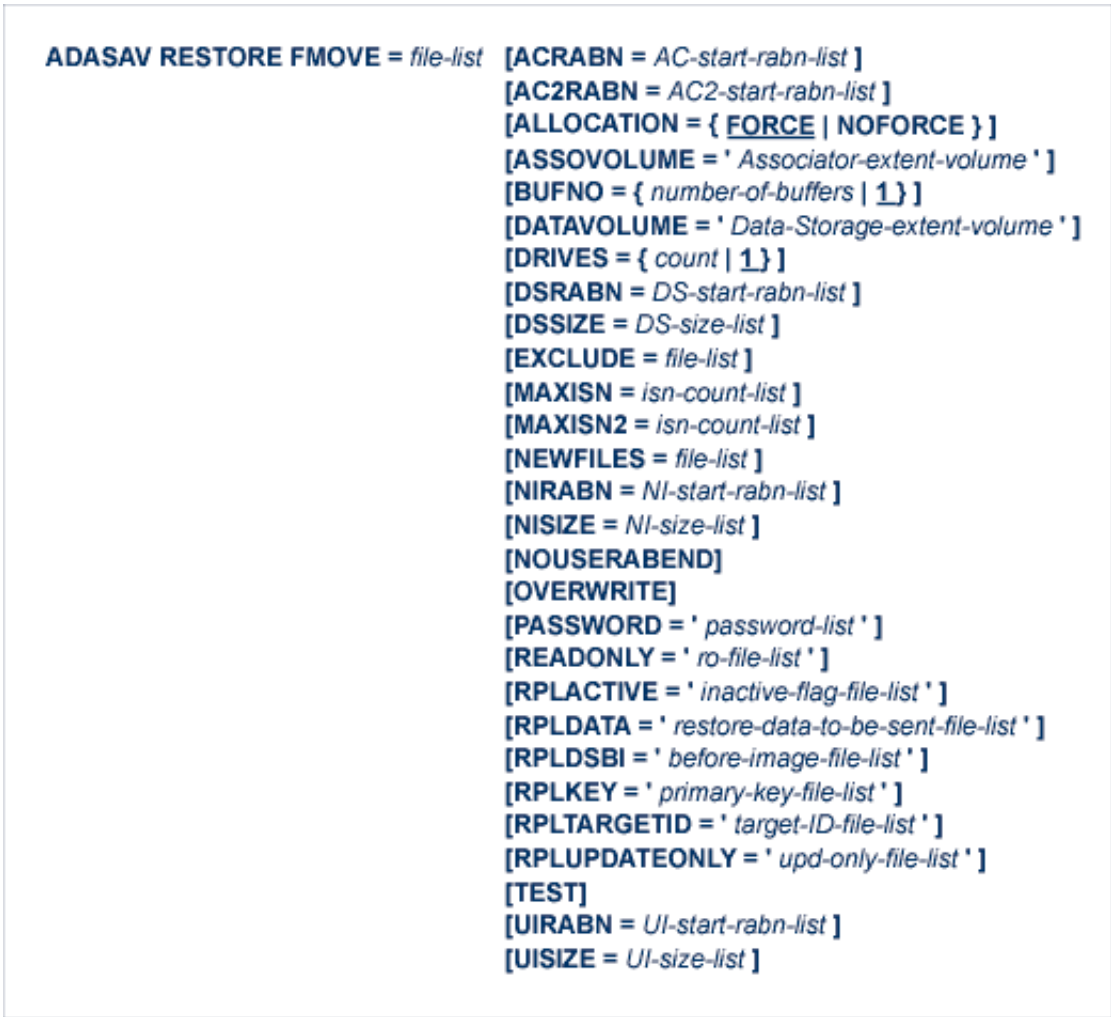
ADASAV RESTORE: Restore Database from Offline Source

```
ADASAV RESTORE  [BUFNO = { number-of-buffers | 1 } ]
                [CLOGDEV = CLOG1-device-type ]
                [DRIVES = { count | 1 } ]
                [EXCLUDE = file-list ]
                [NEWDBID = new-database-id ]
                [NEWDBNAME = new-database-name ]
                [NOUSERABEND]
                [OVERWRITE]
                [PLOGDEV = PLOG-device-type ]
                [TEST]
```

ADASAV RESTORE FILES: Restore Files to Original RABNs from Offline Source



ADASAV RESTORE FMOVE: Restore Files to Any RABNs from Offline Source



ADASAV RESTORE GCB: Restore Database Incremental from Offline Source

```
ADASAV RESTORE GCB  [BUFNO = { number-of-buffers | 1 } ]
                    [CLOGDEV = CLOG1-device-type ]
                    [DRIVES = { count | 1 } ]
                    [EXCLUDE = file-list ]
                    [FILES = file-list ]
                    [NEWDBID = new-database-id ]
                    [NEWDBNAME = new-database-name ]
                    [NOUSERABEND]
                    [OVERWRITE]
                    [PLOGDEV = PLOG-device-type ]
                    [READONLY = ' ro-file-list ' ]
                    [RPLACTIVE = ' inactive-flag-file-list ' ]
                    [RPLDATA = ' restore-data-to-be-sent-file-list ' ]
                    [RPLDSBI = ' before-image-file-list ' ]
                    [RPLKEY = ' primary-key-file-list ' ]
                    [RPLTARGETID = ' target-ID-file-list ' ]
                    [RPLUPDATEONLY = ' upd-only-file-list ' ]
                    [TEST]
```

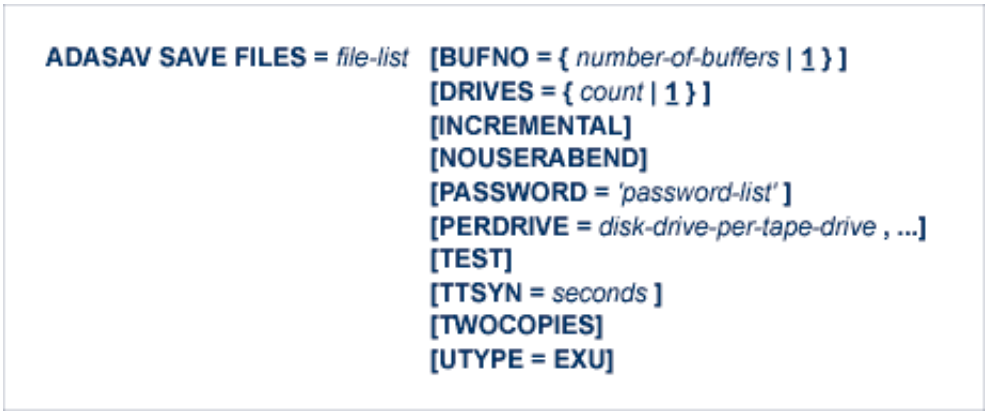
ADASAV RESTPLOG: Restore Protection Log Only

```
ADASAV RESTPLOG  PLOGNUM = protection-log-number
                 { SYN1 | SYN4 } = starting-block-number
                 [FILES = file-list ]
                     [NEWFILES = file-list ]
                 [NOUSERABEND]
                 [TEST]
```

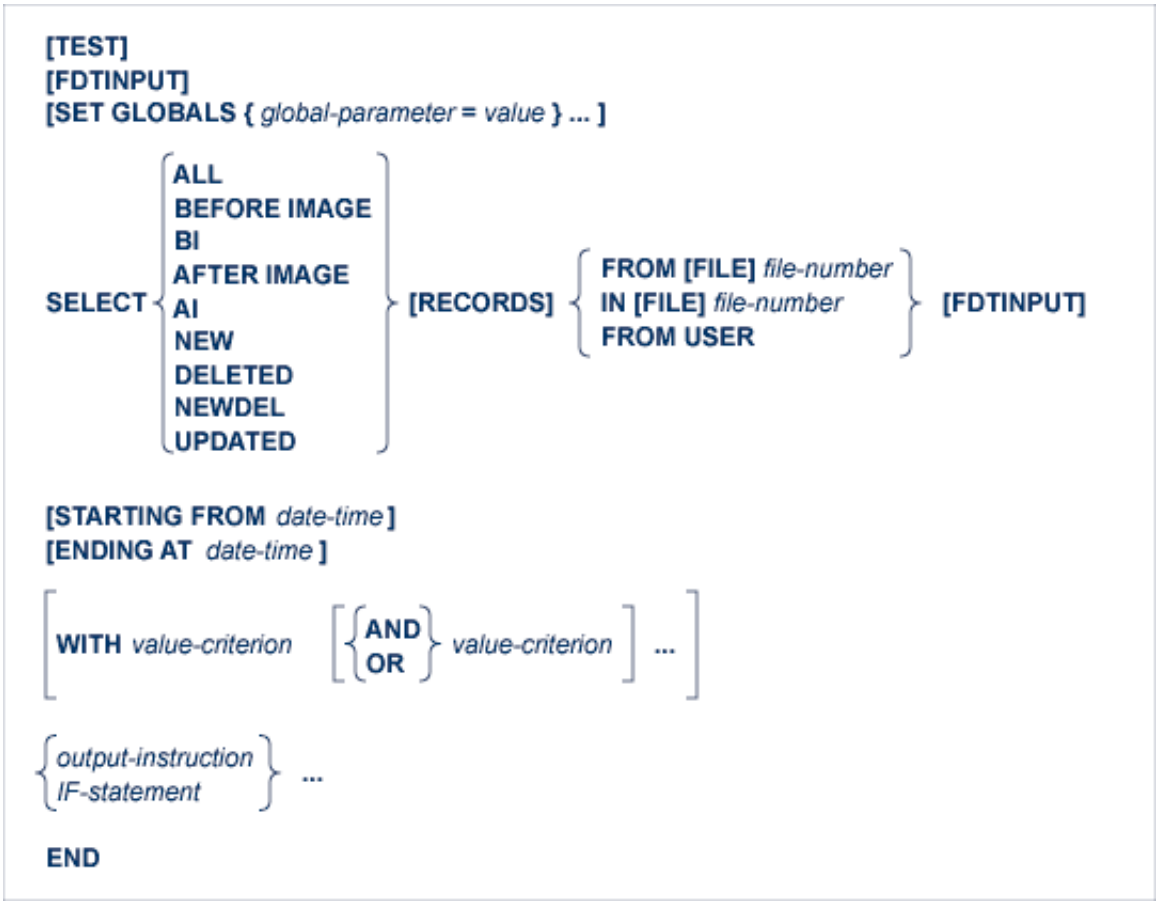
ADASAV SAVE: Save Database

```
ADASAV SAVE  [BUFNO = { number-of-buffers | 1 } ]
             [DRIVES = { count | 1 } ]
             [INCREMENTAL]
             [NOUSERABEND]
             [PERDRIVE = disk-drive-per-tape-drive , ...]
             [TTSYN = seconds ]
             [TWOCOPIES]
             [TEST]
```

ADASAV SAVE FILES: Save Specified Files

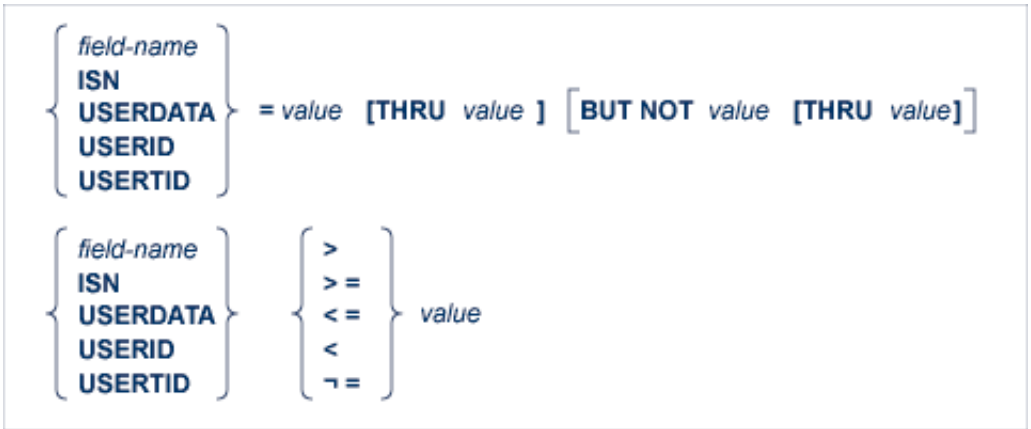


ADASEL Utility: Select Protection Data

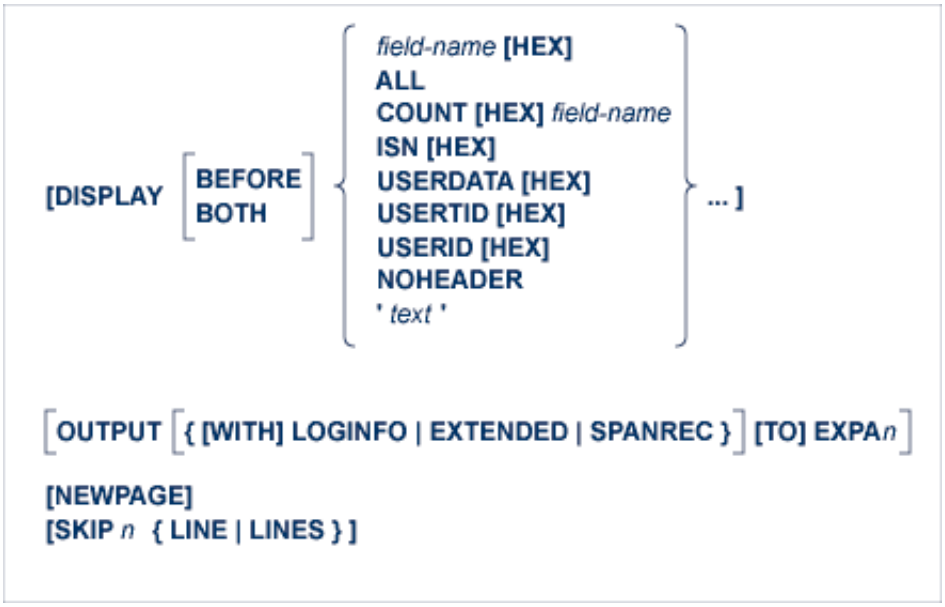


- Value Criterion
- Output Instruction
- IF Processing

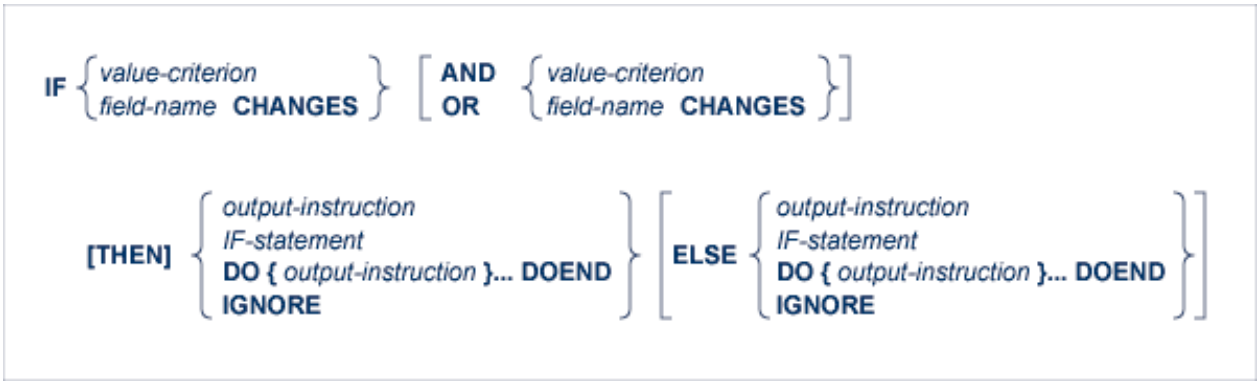
Value Criterion



Output Instruction



IF Processing



ADAULD UNLOAD FILE: Unload Specified File

```
ADAULD [UNLOAD] FILE= file-number
                [CODE = cipher-key ]
                [DDISN]
                [ERRLIM = { error-threshold-count | 1 } ]
                [ETID = multiclient-file-owner-id ]
                [LPB = prefetch-buffer-size ]
                [LRECL = { maximum-compressed-length | 4000 } ]
                [MODE = SHORT]
                [NOUSERABEND]
                [NUMOUT = { 1 | 2 } ]
                [NUMREC = number ]
                [PASSWORD = file-password ]
                [SAVETAPE]
                    [PLOGNUM = plog-number , { SYN1 | SYN4 } = plog-block-number ]
                    [TEMPDEV = device-type ]
                    [TEMPSIZE = size ]
                [SELCRIT = ' selection-criteria ', SELVAL = ' values-for-selection-criteria ' ]
                    [STARTISN = value ]
                [ SORTSEQ = { descriptor [,MU] [,NU]
                            { ISN [, STARTISN = value ] } ]
                [TEST]
                [UTYPE = { EXU | EXF } ]
```

ADAVAL Utility: Validate the Database

```
ADAVAL VALIDATE FILE= file-list
                SORTSIZE= size
                TEMPSIZE= size
                [CODE = cipher-key ]
                [DESCRIPTOR = ' descriptor-list ' ]
                [LPB = prefetch-buffer-length ]
                [LRECL = { record-buffer-length | 4000 } ]
                [LWP = { work-pool-size | 1048576 } ]
                [NOUSERABEND]
                [PASSWORD = password ]
                [SORTDEV = device-type ]
                [TEMPDEV = device-type ]
```

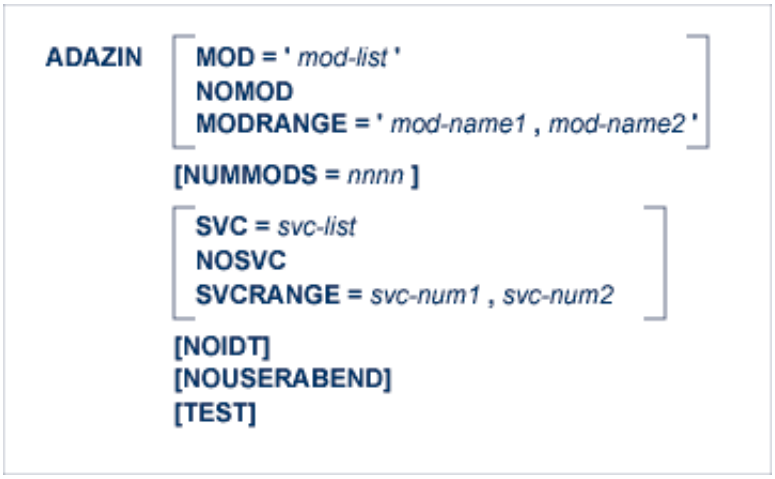
ADAWRK Utility: Work Area Recovery Reports

```
ADAWRK [ABEND34]
       [CHECKPOINT = {YES | NO }]
       [CMID = id1 [, id2 ] ... [, id24 ] ]
       [ETID = etid1 [, etid2 ] ... [, etid32 ] ]
       [FILES = fn1 [, fn2 ] ... [, fn64 ] ]
       [FORCE = {YES | NO }]
       [LWP = nnnK | 1024K ]
       [NOPPT]
       [NOUSERABEND]
       [REPLICATION = { NO | YES | DETAIL | FULL }]
       [REPORTFILE = {YES | NO }]
       [SUMMARY = {NO | YES }]
       [TEST]
       [TIMEZONE = {MACHINE | LOCAL | {+ | -} nn } ]
       [TRANSACTIONS = { NO | YES | DETAIL | FULL} ]
       [USERID = id1 [, id2 ] ... [, id24 ] ]
```

ADAZAP Utility: Display or Modify Asso, Data, and Work Data Sets

```
ADAZAP  MCCODE = master-code
        { ASSO | DATA | WORK }
        [LENGTH = length-of-data ]
        [OFFSET = { offset-from-RABN-start | 0 } ]
        [RABN = { rabn-number | 1 } ]
        [REP = replace-data ]
        [VER = verify-data ]
```

ADAZIN Utility: Print Adabas Maintenance and SVC Information


ADAZIN
MOD = ' mod-list '
NOMOD
MODRANGE = ' mod-name1 , mod-name2 '
[NUMMODS = nnnn]
SVC = svc-list
NOSVC
SVCRANGE = svc-num1 , svc-num2
[NOIDT]
[NOUSERABEND]
[TEST]