

Adabas Utility Execution

The Adabas Utilities documentation contains detailed information about the function and execution of each Adabas utility.

You can use Adabas Online System to execute various utility functions online.

This chapter covers the following topics:

- Condition and Termination Codes
 - File and Database Usage
 - Disk Accesses by Utility and Function
 - Possible Resource Conflicts
 - Repetitive Utility Execution Under VM/ESA or z/VM
-

Condition and Termination Codes

Refer to the Adabas Messages and Codes documentation for detailed descriptions of the condition and termination code meanings for each utility.

The following condition codes can be set by an Adabas utility:

Code	The utility operation ...
0	was successful.
4	was successful but encountered a warning condition.
8	encountered an error condition but then continued.
16	was successful in its main function but then encountered an error condition.
20	could not be completed. The operation encountered an error condition but did not terminate abnormally because NOUSERABEND was specified.

For errors that occur during the execution of a utility, the following termination codes are returned:

Code	The utility terminated . . .
34	with a dump.
35	without a dump.
36	with a dump. Termination was caused by an I/O error while writing to DDDRUCK or DDPRINT. Check the JCL.

File and Database Usage

The table in this section shows the type of file/database usage required for each utility, the database status required, whether the Adabas nucleus must be active while an Adabas utility is running, and whether the function creates an entry in the data integrity block (DIB).

Any utility that requires a nucleus can be used in single-user mode, provided that the nucleus JCL statements are specified.

The following codes are used in the table:

Code	Meaning
n/a	Does not apply.
	File/Database Usage
ACC	Access-only usage: other users may access and update the file/database.
EXF	Exclusive file control: other users may not access or update the file.
EXU	Exclusive file update: other users may not update the file.
UTI	Utility update control: other users may not access or update the file.
UTS	Utility SAVE control: other users may access and update the file.
*	Exclusive database control: no other database activity is permitted.
**	Exclusive database control if the checkpoint or security file is involved; otherwise, read-only access is allowed during the read phase of utility operation.
***	Exclusive database control if the CONTINUE parameter is specified.
UPD	Update/ET user status does not apply to utilities. Utilities cannot be timed out.
	Database Status
NP	Other activity against the database is not permitted.
P	Activity against other database files is permitted.
	Adabas Nucleus Status
A	Adabas nucleus must be active.
C	Adabas nucleus may be active.
N	Adabas nucleus must not be active.

Utility	Function	File/DB Usage	Database Activity	Adabas Nucleus	DIB Entry Created
ADAACK	All	EXU	P	C	NO
ADACDC	All	n/a	P	C	NO

Utility	Function	File/DB Usage	Database Activity	Adabas Nucleus	DIB Entry Created
ADACMP	Compress FDT=...	ACC	P	A	NO
	Compress data	n/a	P	C	NO
	Decompress data	n/a		C	NO
	Decompress INFILE=...	EXU(EXF)		A	NO
ADACNV	All	*	NP	N	YES
	With TEST parameter	ACC	P	C	NO
ADADBS	All, except:	ACC/UTI	P (1)	A	NO
	RECOVER	UTI	(2)	A	NO
	RESETDIB	n/a	P	C	NO
	OPERCOR DDIB	n/a	P	C	NO
ADADCK	All	EXU	P	C	NO
ADADEF	All	*	NP	N	NO
ADAFRM	Formatting other than TEMP/SORT dataset	n/a	NP	N	NO
	Formatting TEMP/SORT dataset	n/a	P	C	NO
	Formatting single blocks	n/a	P	C	NO
ADAICK	ICHECK	EXU	P	C	NO
ADAINV	All	EXU/UTI	P	A	YES
ADALOD	All	UTI	P	C	YES
ADAMER	ADAM estimation	n/a	P	C	NO
ADAORD	REORDER A/D/DB	*	NP	A	YES
	REORDER FILE (A)	EXU/UTI	P	A	YES
	REORDER FILE (D)	**	P	A	YES
	REORDER FILE	EXU/UTI	P	A	YES
	RESTRUCTURE DB	**	NP	A	YES
	RESTRUCTURE FILE	EXU/UTI	P	A	YES
	STORE	**	P	A	YES
		* EXU ** UTI			
ADAPLP	All	n/a	P	C	NO
ADAPRI	Maint print	n/a	P	C	NO
ADARAI	DISABLE, PREPARE, REMOVE CHKDB, LIST RECOVER				

Utility	Function	File/DB Usage	Database Activity	Adabas Nucleus	DIB Entry Created
ADAREP	Database reporting CPLIST/CPEXLIST	n/a	P	C	NO
		ACC	P	A	NO
ADARES	Regenerate file,	UTI ***	P	A	NO
	Backout file	*	NP	A	NO
	Regenerate,	n/a	P	A/C	NO
	Backout	n/a	P	C	NO
	Copy	UTI	P	A	NO
	Copy dual/multi log Repair				
ADASAV	Save file	UTS	P	C	YES
	Save database	UTS	P ⁽³⁾	C	YES
	Restore file	UTI	P	C	YES
	Restore database	*	NP	N	YES
	Restore protection log	*	NP	N	YES
ADASEL	Select protection log data	n/a	P	C	NO
ADAULD	Unload file Unload file from save tape	EXU(EXF) n/a	P P	A C	NO NO
ADAVAl	Validate	EXU	P	A	NO
ADAZAP	All	*	NP	N	NO

Notes:

1. When completed, ADADBS ADD/INCREASE end the nucleus session to permit reallocation and formatting.
2. ADADBS RECOVER cannot run when other utilities are running.
3. When running ADASAV SAVE with an active nucleus ("online"), no other utilities can be run which update the database.

Disk Accesses by Utility and Function

The following table lists the datasets that each utility function accesses on disk:

Utility	Function	Disk Accessed Directly
ADAACK	all	ASSO, DATA
ADACDC	all	ASSO
ADACMP	all	n/a
ADACNV	all	ASSO, DATA, WORK, dual PLOG if specified
ADADBS	OPERCOM DDIB, RESETDIB others	ASSO, if nucleus is down n/a
ADADCK	all	ASSO, DATA

Utility	Function	Disk Accessed Directly
ADADEF	DEFINE NEWWORK	ASSO, DATA, WORK ASSO, WORK
ADAFRM	all	the dataset to be formatted
ADAICK	DATAPRINT, DSCHECK others	ASSO, DATA ASSO
ADAINV	all	ASSO
ADALOD	all	ASSO, DATA, WORK
ADAMER	all	n/a
ADAORD	REORDB, REORFILE, REORDATA, REORFDATA, STORE REORASSO, REORFASSO, RESTRUCTUREDB, RESTRUCTUREF	ASSO, DATA ASSO
ADAPLP	WORKPRI PLOGPRI others	WORK Dual PLOG n/a
ADAPRI	all	ASSO + dataset to be printed
ADARAI	PREPARE, REMOVE all others	ASSO n/a
ADAREP	all report from SAVETAPE*	ASSO; DATA is not required but is recommended because ADAREP tries to print the volume numbers. n/a
ADARES	CLCOPY, PLCOPY all others	Dual CLOG, PLOG; ASSO is recommended but not required ASSO
ADASAV	SAVE DB, SAVE FILE, RESTORE FILE	ASSO, DATA; also WORK if nucleus is down
ADASEL	all	ASSO
ADAULD	all	n/a
ADAVAL	all	n/a
ADAZAP	all ADAZAP is to DATA ADAZAP is to WORK	ASSO DATA WORK

* Adabas makes no attempt to open DATA and ASSO for an ADAREP from a SAVETAPE. For a regular ADAREP, an attempt is made to open DATA.

Possible Resource Conflicts

Note:

See *Program Synchronization Using Operating System Services* for information about how Adabas 7 uses operating system services to ensure that the start and end of nucleus and utility jobs are synchronized.

An Adabas utility which requires exclusive control of one or more Adabas files will be terminated with an error message under any of the following conditions:

1. The file has been locked using the operator command LOCKF (or Adabas Online System).

In this case, you must determine why the file was locked and whether it can now be unlocked. The UNLOCKF operator command (or Adabas Online System) may be used to unlock a locked file.

2. The file is marked as "in use" by another Adabas utility.

Adabas maintains a list of the files used by each Adabas utility in the DIB block (block 3 of the Associator). The DDIB operator command (or Adabas Online System) may be used to display this block to determine which job is using the file in question.

It is important to note that Adabas will remove entries in the DIB only when a utility terminates normally or with an error message. If a utility terminates abnormally, the files used by that utility remain "in use". The DBA may explicitly release any such files with the RESETDIB function of the ADADBS utility. For example, if the job name of the utility using the file is "JOBUT1", the following ADADBS parameters would be entered:

```
ADADBS RESETDIB JOBNAME=JOBUT1
```

If the jobname is not unique, the IDENT parameter must be used to specify the job name qualifier for the abnormally ended utility. The DDIB command (or Adabas Online System) can be used to display the identifier:

```
ADADBS RESETDIB IDENT=identifier
```

The DIB may also be reset using Adabas Online System.



Warning:

The DIB should be reset only if the utility in question actually terminated abnormally. Resetting the DIB for an active utility will cause unpredictable results.

3. The file is marked as in use by another application (user).

Adabas creates a user queue element (UQE) when the first Adabas command is received from a user program or Adabas utility. The UQE contains user information (user ID, user type, etc.) and a file list. The file list identifies each file the user is using and the type of file usage. The file-usage type is one of the following:

Type	The file is opened for . . .
ACC	access only
EXU	exclusive update
EXF	exclusive use
UPD	update (ET logic user)

See the table in the section *File and Database Usage* for information about file usage by Adabas utilities.

The file list is deleted as a result of the STOPU operator command (or using Adabas Online System), when an automatic timeout occurs, or when the user issues an Adabas CL (close) command. ET/BT users file lists are released when the user is at ET status.

Utility functions that modify the database cannot run if there is a pending autorestart, *and* the nucleus is not active.

A pending autorestart condition can only be recovered by starting the Adabas nucleus. If the nucleus cannot execute the autorestart successfully, the only valid restart method is to run

- ADASAV RESTORE (DATABASE); and then
- ADARES REGENERATE.

Repetitive Utility Execution Under VM/ESA or z/VM

To run the same utility (ADAREP, for example) more than once from an EXEC, the use of EXECOS is strongly recommended; otherwise, utility-dependent program checks can occur. The following example shows how the ADAREP utility could be run under EXECOS:

```
'EXECOS EXEC ADAREP REPCPLST 1'
'NUCXDROP ADARUN'
'EXECOS EXEC ADAREP REPCPLST 1'
```