

# Functional Overview

The ADACDC utility

- takes as input one or more sequential protection logs; and
- produces as output a delta of all changes made to the database over the period covered by the input protection logs.

"Delta of changes" means that the last change to each ISN in a file that was altered during this period appears on the primary output file.

This output may be used on a regular basis as input for data warehousing population procedures so that the delta of changes to a database is applied to the data warehouse database rather than a copy of the entire database. This affords more frequent and less time consuming updates to the data warehouse, ensuring greater accuracy of the information stored there.

In order to run the ADACDC utility

- an external sorter must be available and installed as the standard sorter in the operating system. See Operating System Considerations for more information.
- the external sorter must have access to the database's Associator containing the FDTs of the files for which records are to be processed.

This chapter covers the following topics:

- Phases of Operation and Resulting Files
- Primary Input Data
- Primary Output Data
- Transaction File

---

## Phases of Operation and Resulting Files

ADACDC processes sequential protection logs in two phases. You can execute phase 1 and phase 2 separately, or both at once (the default):

- If phase 2 is being run separately or both phases are being completed together, the data is decompressed and written to the *primary output file*.
- If only phase 1 is being executed, the data is written to an *extract file*. This extract file may then be processed multiple times by a phase 2 operation to decompress the records and write to primary output files.

The extract file contains data records in compressed format whereas the primary output file contains records in decompressed format. Refer to the section *ADACMP (Compress - Decompress)* in this documentation for more information about these formats.

The primary output file and the extract file are standard operating-system-dependent files that can handle variable length records.

## Phase 1 and the Extract File

During phase 1, updates from the protection logs are analyzed and prefixed with a standard structure called the CDCE. The format of each record on the file is a constant CDCE prefix followed by the compressed record information. These records are passed to an external SORT routine to establish the most recent update for each ISN on a file. Only the last change for a given file and ISN combination is written.

The extract file created when phase 1 is run separately makes it possible to process the PLOG data once and then optionally produce multiple primary output files from it based, for example, on file selection criteria. The option is useful if different file changes are required for different purposes.

When the phase 1 process is being run, the extract file is opened for output. As records are output from the SORT processing, the latest update for each file and ISN combination is written to the extract file if

- the update was performed by an ET user and belongs to a completed transaction; or
- the update was performed by an EXU user and belongs to a completed command; or
- NOET is specified.

All other updates for the file and ISN combination for that period are discarded if there are no controlled utility operations against that file (see Checkpoints Written to the Primary Output File).

### Note:

It is possible to have duplicate file and ISN combinations on the file if the ADACDC user exit (described later) adds records with file and ISN combination that already exist. A record added or modified by the user exit is so marked in the CDCE structure.

## Phase 2 or Both and the Primary Output File

The primary output file is used when both stages of ADACDC are run together, or for phase 2 processing only.

- If both phases are run together, the primary output file is opened and created directly using the output from the SORT processing. In this case, processing occurs as for the extract file for phase 1 processing.
- If only phase 2 is run, the primary output file is created using input from the extract file.

The format of each record on the file is a constant CDCO prefix followed by the decompressed record information. If for some reason the record cannot be decompressed, a warning message is issued and the compressed record is written to the primary output dataset. A flag in the CDCO structure informs a user program when decompression for the record has failed.

### Note:

It is possible to have duplicate file and ISN combinations on the file if the ADACDC user exit (described later) adds records with file and ISN combinations that already exist. A record added or modified by the user exit is so marked in the CDCO structure.

## Checkpoints Written to the Primary Output File

The primary objective of the ADACDC utility is to provide an output dataset containing the most recent changes for each ISN in a file that has been modified for the period concerned.

Apart from simple changes to a file, some utility operations executed against a file may fundamentally affect its contents. For example, if the file is deleted, simply providing the last updates for ISNs in the file does not accurately reflect the state of the file since all ISNs have been deleted.

For this reason, the following checkpoints are recorded and written to the primary output file as appropriate with the associated indication in the output record:

ADASAV RESTORE FILE	File created
ADAORD STORE FILE	File created
ADALOD LOAD FILE	File created
ADALOD UPDATE FILE	File updated
ADADEBS DELETE FILE	File deleted
ADADEBD REFRESH FILE	File deleted

Because these operations can fundamentally impact a file and its appearance, the checkpoint is written to the primary output dataset when it occurs relative to the other updates.

ADACDC retains the last change to all ISNs before each of the above checkpoints. This means that a file and ISN combination could appear multiple times on the primary output file if one or more checkpoints were written to it. This is useful for the many data warehouse packages that may wish to complete their view of a file and maintain a copy of it prior to deletion, re-creation, or mass update.

## Primary Input Data

The primary input data comprises sequential protection logs produced either

- by the database directly; or
- by the ADARES PLCOPY of nonsequential protection logs.

ADACDC processes this data to ensure that

- when a new PLOG block is read and the PLOG number is the same, the PLOG block number is 1 greater than the previous PLOG block number.
- when the PLOG number itself changes, the new PLOG number is higher than the previous PLOG number and the new PLOG block number is 1.

### Note:

When the PLOG number changes and the difference between the PLOG numbers is greater than 1, a warning message is issued and processing continues as this can legitimately happen if online saves are used.

If any of these checks fail, the utility execution terminates.

## Primary Output Data

The primary output data is a sequential file comprising all database records that were added, updated, or deleted during the period covered by the input protection logs.

If a record was changed several times, only its last change appears in the output file. ADACDC employs a SORT process to identify multiple changes to the same record.

Each primary output record comprises a fixed-length record prefix followed by the database record in decompressed form. The decompressed data corresponds in format to the output of the ADACMP DECOMPRESS function.

The primary output record prefix is described by the CDCO DSECT. It has the following structure:

Bytes	Description
0-1	record length (binary)
2-3	set to zeros
4-7	constant 'CDCO'
8-9	database ID
10-11	file number
12-15	ISN of the updated record
16-19	length of the decompressed data in bytes
20-47	28-byte communication ID of the last user who updated the record
48	change indicator: X'04'            record added X'08'            record updated X'0C            record deleted X'10'            file created X'14'            file updated X'18'            file deleted or refreshed
49	flags (independent bit settings): X'80'    record added by user exit X'40'    record modified by user exit X'20'    record still compressed; decompression failed
50	database version indicator
51-67	reserved for future use
68-...	decompressed record data

## Transaction File

To maintain input data checking over multiple runs of the utility, ADACDC stores information on the transaction file in a transaction control record containing the last database ID, the PLOG number, and the PLOG block number processed. This information is used to verify the latest input (unless the RESETTXF option is specified - see section *RESETTXF : Reset Input Transaction File* in the *ADACDC Optional Parameters* section).

ADACDC actually recognizes two different transaction files: input and output. Both transaction files are standard operating-system-dependent files that can handle variable length records.

## **Input Transaction File Processing**

During the input processing stage, ADACDC processes the input transaction file to the SORT program.

Following the control record on the input transaction file, 0 or more records may be found. These are database updates related to transactions not completed during the last run of the utility. These records are processed again as part of the input as their transactions will normally have been completed in the next sequential protection logs provided to the utility. This is the reason the sequence of protection logs is so important: updates may remain outstanding forever if the correct sequence is not maintained.

The transaction file also records whether the NOET option was specified during the last phase 1 run of the utility. When ADACDC detects that this option has changed from one utility execution to the next, it uses the information from the control record on the input transaction file; however, all transactional information in the other records is ignored. This is due to the fact that changing this option may cause inconsistent data to be written to the primary output file or extract file, as appropriate. ADACDC issues a warning when this happens.

## **Output Transaction File Processing**

Once output processing from the SORT program starts, the input transaction file is closed and the output transaction file is opened. The control record is written to the output transaction file followed by any updates that relate to incomplete transactions or, in the case where the NOET option is in effect or an EXU user is in control, to incomplete commands. The output transaction file is closed once processing is complete.

## **Using a Single Transaction File**

It is possible to use the same file as both the input and output transaction file; however, if the utility fails while writing to the output transaction file (that is, at any time during the output processing of the SORT utility), the input transaction file will no longer exist and therefore, rerunning the utility will yield a different result.

For this reason, the transaction file must be backed up prior to the utility run so that it can be restored in the event of a failure.

Alternatively, you could use a facility on your operating system (if available) that produces a new version of a file whenever a program updates the file.