

CONVERT: Convert Database to Higher Version

The CONVERT function starts from the Adabas version of the last nucleus session.

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

This chapter covers the following topics:

- Optional Parameters
 - Conversion Considerations
 - Example
-

Optional Parameters

IGNPPT: Ignore Parallel Participant Table PLOG Entries

When converting from a version of Adabas that uses the parallel participant table (PPT) structure to a higher version of Adabas, an error is printed and conversion fails if the system detects one or more protection logs (PLOGs) from the current version that have not been copied/merged.

Use IGNPPT=YES to continue processing in spite of the uncopied/unmerged PLOGs.

NOUSERABEND: Termination Without Abend

When an error is encountered while the function is running, the utility prints an error message and terminates with user abend 34 (with a dump) or user abend 35 (without a dump).

If NOUSERABEND is specified, the utility will *not* abend after printing the error message. Instead, the message "utility TERMINATED DUE TO ERROR CONDITION" is displayed and the utility terminates with condition code 20.

PLOGDEV: Multiple PLOG Device Type

PLOGDEV specifies the physical device type on which the multiple protection log datasets to be converted are contained. If PLOGDEV is not specified, the device type specified by the ADARUN DEVICE parameter is used.

RESTART: Rerun after Point of No Return

If ADACNV terminates abnormally after the "point of no return", that is, after all changed blocks have been written to DD/FILEA, the RESTART parameter instructs ADACNV to begin its run by reading the contents of DD/FILEA and continue by writing them to the database.

TEST: Test Conversion

The TEST parameter tests the feasibility of the conversion operation without actually writing any changes to the database.

TOVERS: Target Version

The version of Adabas database (version and revision level) to achieve at the end of the ADACNV run. If the TOVERS parameter is

- specified, it must be a version higher than the source version.
- not specified, ADACNV uses its own version as the target version.

The version format is *vr* indicating the version and revision level; for example, 74.

Conversion Considerations

The following is an overview of the conversion steps performed by ADACNV.

All Versions

- The data protection area on the Work dataset and the multiple PLOG datasets (if supplied) are cleared to binary zeros.

From Version 5.2 to 5.3

- The new checkpoint file FDT is installed.
- For a security file, any search-by-value criteria are adjusted to the new internal search structure.

From Version 5.3 to 6.1

- The free space table (FST) is converted from 3- to 4-byte RABN. If an FST RABN overflow occurs, the smallest FST extent is removed. This is repeated until the FST fits into the ASSO block. An appropriate message is printed.
- Unused RABN chains are converted from 3- to 4-byte RABNs for each loaded file.
- If a block of unreadable blocks (BUB) exists, it is converted from 3- to 4-byte RABN structure.
- The new security file FDT is installed.
- Any Delta Save Facility DLOG area header is set to the correct version. If the Delta Save Facility logging status is "enabled", it is set to "disabled" and an appropriate message is printed.

From Version 6.1 to 6.2

- Any Delta Save Facility DLOG area header is set to the correct version.

From Version 6.2 to 7.1

- Any Delta Save Facility DLOG area header is set to the correct version.

Example

```
ADACNV CONVERT TOVERS=71
```

The version of Adabas selected in the last nucleus session is to be converted to a version 7.1 database.