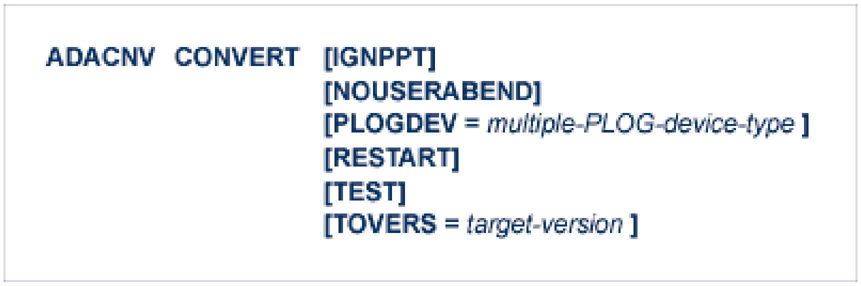
CONVERT: Convert Database to Higher Version

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



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

Specify IGNPPT to continue processing in spite of the uncopied/unmerged PLOGs.

Note:
If DDPLOG*x* statements have been specified in the JCL in addition to the DDPLOG*x* data sets in the PPT, the specified data sets must be empty or the error will still be received. IGNPPT only pertains to the PPT processing. If PLOG data sets are supplied in the JCL, they must be empty.

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). When NOUSERABEND is specified, it must be specified as the first parameter (before all other parameters) for the utility function.

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 data sets 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 two-character 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" or "81".

Note:
There may be files in a database that are not loaded but that have a Field Definition Table (FDT) stored in the FDT blocks. If ADACNV encounters such FDTs while converting a database to Version 8, they are deleted as part of its cleanup processing.

Conversion Considerations

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

All Versions

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

From Version 6.1 to 6.2

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

From Version 6.2 to 7.1

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

Any Version to Version 8

There may be files in a database that are not loaded but that have a Field Definition Table (FDT) stored in the FDT blocks. If ADACNV encounters such FDTs while converting a database to Version 8, they are deleted as part of its cleanup processing.

Example

```
ADACNV CONVERT TOVERS=81
```

The version of Adabas selected in the last nucleus session is to be converted to a version 8.1 database.