

# 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 DDPLOGx statements have been specified in the JCL in addition to the DDPLOGx 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 a parameter error or a functional error occurs while this utility function is running, the utility ordinarily 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.

**Note:**

When NOUSERABEND is specified, we recommend that it be specified as the first parameter of the utility function (before all other parameters). This is necessary to ensure that its parameter error processing occurs properly.

**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 release 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

In general, you will need to run ADACNV on your database whenever the Adabas version or release number (*v.r*) has changed, but not for modification (SM) levels. For example, ADACNV should be run to convert your database from 7.4 to 8.1 and from 8.1 to 8.2. However, there is usually no need to run ADACNV to convert your database from 8.2.2 to 8.2.3.

The following is an overview of special conversion considerations for 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 Facility DLOG area header is set to the correct version.

**From Version 6.2 to 7.1**

- Any Adabas Delta Save Facility 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**

The following example indicates that the database should be converted to the version of Adabas of which the ADACNV utility is part.

```
ADACNV CONVERT
```