

ACCHECK: Check Address Converter Against Data Storage

```
ADAACK ACCHECK [FILE= { file | file1 - filex } ]  
                [ISN= isn1 - isnx ]  
                [NOOPEN]  
                [NOUSERABEND]
```

This chapter describes the syntax and parameters of the ACCHECK function.

- Optional Parameters
 - Examples
-

Optional Parameters

FILE: Files to be Checked

The file, single range of files, or all files to be checked. By default, all files in the database are checked.

ISN: ISN Range to be Checked

A range of ISNs or all ISNs to be checked. By default, the entire range MINISN through TOPISN is checked.

If spanned records are in use, ADAACK assumes that any ISNs passed to it are primary ISNs, and performs its processing accordingly. If an ISN is the primary ISN of a spanned Data Storage record, Adabas will automatically check the appropriate segments records for the spanned record in the secondary address converter (AC2).

When printing error information about a particular ISN, the ADAACK utility will now indicate whether the problem is with a primary or secondary ISN, if the record is spanned.

NOOPEN: Prevent Open Resynchronization

When starting, ADAACK normally performs a utility open call to the nucleus to assure that no blocks of the affected file or files are still in the nucleus buffer pool. However, this also locks the file for other users. Specifying NOOPEN prevents ADAACK from issuing the open call.

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.

Examples

Example 1:

```
ADAACK ACCHECK
```

Check all files in the database.

Example 2:

```
ADAACK ACCHECK FILE=12, ISN=1-8000
```

Check ISNs 1 through 8000 for file 12.

Example 3:

```
ADAACK ACCHECK FILE=8-10
```

Check all ISNs in files 8 through 10.