

# DSCHECK: Check Data Storage

```
ADADCK DSCHECK [FILE = { file [FROMRABN = DS-blknum ] [TORABN = DS-blknum ] | file - file } ]  
                [MAXPISN = { num | 1000 } ]  
                [NOOPEN]  
                [NOUSERABEND]  
                [REPAIR]  
                [USAGE]
```

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

- Optional Parameters and Subparameters
- Examples

---

## Optional Parameters and Subparameters

### **FILE: Files to Be Checked**

The file (or a single range of files) to be checked. If omitted, all files in the database are checked.

### **FROMRABN: Data Storage Block Number**

The RABN of the Data Storage block where the check is to start. This parameter is applicable only if a single file is to be checked. In other words, only one FROMRABN/TORABN range can be specified in a single ADADCK run.

If more than one FROMRABN/TORABN range is specified in an ADADCK request for multiple files, only the last range is used in the run. In addition, if the range specified by the FROMRABN/TORABN parameters is outside the range for any given file DS extent, ADADCK will not check the blocks in the extent. Consequently, if more than one FROMRABN/TORABN range is specified, or if the range is outside the range for any given file DS extent, Adabas issues a warning message.

If this parameter is omitted, the check starts at the beginning of the first allocated Data Storage extent for the file.

### **MAXPISN**

The maximum number of primary ISNs that will be checked for a spanned Data Storage file. The default is 1000. If the file contains more primary ISNs than the MAXPISN setting, execution will continue but the following warning message will be displayed.

\*\*\* Warning \*\*\*

More than MAXPISN primary spanned ISNs. Only the first MAXPISN ISNs will be checked. Run ADADCK again specifying a different FROMRABN to check the remaining RABNs or specify a higher MAXPISN value. Any errors reported after this warning may be due to the table limitation.

### **NOOPEN: Prevent Open Synchronization**

When starting, ADADCK 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 ADADCK from issuing the open call and blocking file usage for other users.

### **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.

### **REPAIR: Repair the Data Storage Space Table**

If ADADCK finds any invalid Data Storage Space Table (DSST) elements, it automatically repairs the table if this parameter is supplied.

### **TORABN: Ending Data Storage Block Number**

The RABN of the Data Storage block where the check is to end. This parameter is applicable only if a single file is to be checked. In other words, only one FROMRABN/TORABN range can be specified in a single ADADCK run.

If more than one FROMRABN/TORABN range is specified in an ADADCK request for multiple files, only the last range is used in the run. In addition, if the range specified by the FROMRABN/TORABN parameters is outside the range for any given file DS extent, ADADCK will not check the blocks in the extent. Consequently, if more than one FROMRABN/TORABN range is specified, or if the range is outside the range for any given file DS extent, Adabas issues a warning message.

If this parameter is omitted, the check ends at the end of the last allocated Data Storage extent for the file.

### **USAGE: Print Data Storage Block Usage**

If USAGE is specified, ADADCK prints a bar graph that shows the number of bytes used in each Data Storage block, the block size, and the percentage of blocks used.

## Examples

Check Data Storage and its DSST for file 20, print a bar graph of the Data Storage block utilization and repair the space table if required.

```
ADADCK DSCHECK FILE=20, USAGE, REPAIR
```

Check Data Storage and its DSST for the files 8 through 12.

```
ADADCK DSCHECK FILE=8-12
```

Check Data Storage and its DSST for file 12 in the RABN range 878 through 912.

```
ADADCK DSCHECK FILE=12,  
FROMRABN=878, TORABN=912
```