The oncheck Utility

Use the **oncheck** utility to check specified disk structures for inconsistencies, repair

inconsistent index structures, and display information about disk structures.

The **oncheck** utility requires sort space when examining an index. The amount of sort space required is the same as that needed to build the index. If you receive the error "no free disk space for sort," you must estimate the amount of temporary space needed and make that space available.

You can use SQL administration API commands that are equivalent to some **oncheck** commands.

# oncheck Check-and-Repair

The **oncheck** utility repairs disk structures.

The **oncheck** utility can repair the following types of disk structures:

* Partition page statistics
* Bitmap pages
* Partition blobpages
* Blobspace blobpages
* Indexes
* Sbspace pages
* Metadata partitions for sbspaces

If **oncheck** detects inconsistencies in other structures, messages alert you to these inconsistencies, but **oncheck** cannot resolve the problem.

# What Does Each Option Do?

The **oncheck** options fall into three categories: check, repair, and display.

The display or print options (those prefixed with the letter **p**) are identical in function to the **-c** options, except that the**-p** options display additional information about the data that is being checked as the **oncheck** utility executes. You cannot combine **oncheck** option flags except as the following paragraphs describe.

In general, the **-c**options check for consistency and display a message on the screen only if they find an error or inconsistency.

Any user can execute the check options. On UNIX platforms, you must be user **informix** or **root** to display database data or initiate repair options.

On Windows, you must be a member of the **Informix-Admin** group to display database data or initiate repair options.

[Table 1](https://www.ibm.com/support/knowledgecenter/SSGU8G_14.1.0/com.ibm.adref.doc/ids_adr_0371.htm?view=kc#ids_adr_0371__sii-03a-11470) associates **oncheck** options with their function. It also shows the SQL administration API *command* strings that are equivalent to the **oncheck -c** options.

| **Object** | **Check** | **SQL administration API *command* string** | **Repair** | **Display** |
| --- | --- | --- | --- | --- |
| Blobspace simple large objects |  |  |  | **-pB** |
| System catalog tables | **-cc** |  |  | **-pc** |
| Data rows, no simple large objects or smart large objects | **-cd** |  |  | **-pd** |
| Data rows, simple large objects but no smart large objects | **-cD** |  |  | **-pD** |
| Table with a user-defined access method | **-cd, -cD** | CHECK DATA |  |  |
| Chunks and extents | **-ce** | CHECK EXTENTS |  | **-pe** |
| Index (key values) | **-ci, -cix** |  | **-ci -y -pk -y, -pkx -y** | **-pk** |
| Index (keys plus rowids) | **-cI, -cIx** |  | **-cI -y -pK -y, -pKx -y** | **-pK** |
| Index with a user-defined access method | **-ci, -cI** |  |  |  |
| Index (leaf key values) |  |  | **-pl -y, -plx -y** | **-pl** |
| Index (leaf keys plus rowids) |  |  | **-pL -y, -pLx -y** | **-pL** |
| Pages (by table or fragment) |  |  |  | **-pp** |
| Pages (by chunk) |  |  |  | **-pP** |
| Root reserved pages | **-cr, -cR** |  |  | **-pr, -pR** |
| Metadata for smart large objects | **-cs, -cS** |  |  | **-ps, -pS** |
| Space usage (by table or fragment) |  | CHECK PARTITION  PRINT PARTITION |  | **-pt** |
| Space usage (by table, with indexes) |  |  |  | **-pT** |
| *Table 1. oncheck Options and Their Function* | | | | |

# Using the -y Option to Perform Repairs

Use the **-y** option to instruct **oncheck** to perform repairs automatically.

If you do not use the **-y** option, **oncheck** prompts you when it encounters an inconsistency and allows you to request a repair. If you specify option **-n**, **oncheck** does not prompt you because this option instructs **oncheck** to not perform repairs.

The following examples show automatic repair commands for the **oncheck** utility:

oncheck -cd -y

oncheck -cD -y

oncheck -ci -y

oncheck -cI -y

# Repairing Indexes in Sbspaces and External Spaces

The **oncheck** utility can repair an index in an sbspace or external space if the index is created using an access method that supports the **oncheck -y** option.

Although the **oncheck** utility does not repair fragmented indexes, user-defined access methods can repair them.

# Locking and oncheck

The **oncheck** utility places a shared lock on a table, so no other users can perform updates, inserts, or deletes until the check has completed.

The **oncheck** utility places a shared lock on a table during the following operations:

* When it checks data
* When it checks indexes (with **-ci**, **-cI**, **-pk**, **-pK**, **-pl**, or **-pL**) and the table uses page locking
* When you specify the**-x** option with **-ci**, **-cI**, **-pk**, **-pK**, **-pl**, or **-pL** and the table uses row locking

If the table does not use page locking, the database server does not place a shared lock on the table when you check an index with the **oncheck** **-ci**, **-cI**, **-pk**,**-pK**,**-pl**, or **-pL** options. When no shared lock is on the table during an index check, other users can update rows during the check.

By not placing a shared lock on tables using row locks during index checks, the **oncheck** utility cannot be as accurate in the index check. For absolute assurance of a complete index check, you can execute **oncheck** with the **-x** option. With the **-x** option, **oncheck** places a shared lock on the table, and no other users can perform updates, inserts, or deletes until the check has completed.

The **oncheck** utility returns unreliable results when run on secondary servers in a high-availability cluster.

The **oncheck** utility places a shared lock on system catalog tables when they are checked. It places an exclusive lock on a table when it executes repair options.