# Modify Storage Parameters

**Jochen Hein**

$Id: modify_storage.xml,v 1.1 2002/08/22 09:24:55 jhein Exp $



## Table of Contents



## Modify Storage Parameters

If you are creating a database with a custom load or perform a database migration you probably need to change some storage parameters. SAP suggests to change the SAP dictionary (Transaction SE11) temporarily, and go back after the migration. This has two problems: It is not easy to know the needed settings and it is tendious to change the parameters.

I propose to change the parameters at export time in the EXT-file for the initial extent and in the STR-files for the next extent. One pro is that you can change these files easily even when importing (beware that you might need to drop the table and must manipulate the logfile). Once you had imported all tables for a test run, you know exactly what must be changed for the productive migration. To eliminate all chances for errors I've written the script **modify_storage.sh**.

The basic problem is that ...

To run the script, change into the export directory, from there to `DATA` and run the script. You should backup all the STR- and EXT-files before you start.

```
$ modify_storage.sh
...
```

## The code

The code has been written in **ksh**, a Unix scripting language, and **sed**, the stream editor. As you will see, nothing magic hase been done here.

The main code of the program is structured as follows:

```
Main Listing

#!/bin/ksh
#
Modify table parameters
Modify index parameters
Move table into another tablespace
Usage examples
```

First part of the script is a function to modify_table. This function changes the initial extent for the table and the primary index in the EXT-file and the next extents in the STR-file. This functions reads all STR files, so it may need some seconds to finish. If you change many tables or indices this is not ideal performance wise, but it runs once for a migration and probably in the online time before the export starts. So some minutes runtime won't hurt.

The function must be called with five parameters:

1. The name of the table to change
2. Size of the initial extent of the table in bytes
3. Next extent size (as describes in DDLORA.TPL) for the table
4. Size of the initial extent of the primary index in bytes
5. Next extent size (as describes in DDLORA.TPL) for the primary index

```
Modify table parameters

#-
# Hilfsfunktion zum Manipulieren einer Tabelle
# Parameter:
# $1: Tabellenname
# $2: Size des Initial-Extents der Tabelle in Bytes
# $3: Size des Next-Extents der Tabelle (Verweis auf DDLORA.TPL)
# $4: Size des Initial-Extents des Primaryindex in Bytes
# $5: Size des Next-Extents des Primaryindex (Verweis auf DDLORA.TPL)
modify_table()
{
  if [ $# -ne 5 ]; then
    echo "Uah"
    exit 1
  fi

  for i in SAP*.STR; do
    mv $i $i.TMP
    # Manipulieren der STR-Datei (Next-Extents)
    sed  -e "/^tab: $1\$/,/^att:/s/\(^att: [A-Z][A-Z0-9]* \)[0-9]*\( .* \)[0-9][0-9]*\( \$\)/\1$3\2$5

  done
  for i in ../DB/ORA/SAP*.EXT; do
    mv $i $i.TMP
    # Manipulieren der EXT-Datei (Initial-Extent)
    sed -e "s/^\($1     [       ]*\)[0-9][0-9]*/\1$2/" > $i.TMP > $i.WRK
    sed -e "s/^\($1~0   [       ]*\)[0-9][0-9]*/\1$4/" > $i.WRK > $i
  done
}
```

The next helper function modifies the secondary indices. These are store in the STR file and are pretty similar to the table definition. The idea and structure of the regular expression is therefor also similar to the `modify_table` function.

```
Modify table parameters
```

```
modify_index()
{
  for i in SAP*.STR; do
    mv $i $i.BAK
    sed  -e "/^ind: $1/,/^att:/s/\(^att: [A-Z][A-Z0-9]* *[A-Z][A-Z0-9]* \)[0-9]*\( .*\)/\1$2\2/g" > $

  done
}
```

Another often needed function is moving a table into another tablespace. In theory, the SAP data dictionary should be updated when the table is moved (for exmple with **sapdba**). In practice, this is often not done. Another reason is that you'd like to move a table when migrating. The system is offline anyway and it doesn't really matter in which tablespace the import writes from a performance point of view.

```
Move Table into another tablespace

echo "Sind die *.STR und *.EXT-Dateien gesichert? (j/n)"
read antwort
if [ "$antwort" != "j" ]; then exit 1 ; fi
```

```
Modify Table Parameters

# Pruefen: TST03,

#            Tabelle    Initial      Data  Initial     Index
modify_table APQD       300000000    8     100000000   9
modify_table BALC       500000000    8     100000000   9

modify_index COEP~1   11
modify_index COEP~UBR 11
```

# Example files

Give some examples hers.

```
tab: COEP
att: APPL1 4  ?N T all    COEP~0                          APPL1 4
fld: MANDT                          CLNT     3     0   0 not_null  1
fld: KOKRS                          CHAR     4     0   0 not_null  2
fld: BELNR                          CHAR    10     0   0 not_null  3
fld: BUZEI                          NUMC     3     0   0 not_null  4
fld: PERIO                          NUMC     3     0   0 not_null  0
fld: WTGBTR                         CURR     8    15   2 not_null  0
fld: WOGBTR                         CURR     8    15   2 not_null  0
fld: WKGBTR                         CURR     8    15   2 not_null  0
...


ind: COEP~1
att: COEP                           APPL1 4  not_unique
fld: MANDT
fld: LEDNR
fld: OBJNR
fld: GJAHR
...


ind: COEP~UBR
att: COEP                           APPL1 4  not_unique
fld: BUKRS
fld: PERIO
...
```